

A Gauss Pseudospectral Transcription for Optimal Control

by

David Benson

M.S. Aerospace Engineering, University of Colorado, 2001

B.S. Aerospace Engineering, University of Colorado, 2000

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

February 2005

© David Benson, MMV. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Author

Department of Aeronautics and Astronautics

October 21, 2004

Certified by

Tom Thorvaldsen

Distinguished Member of the Technical Staff, Draper Laboratory

Technical Supervisor

Certified by

Steven R. Hall

Professor of Aeronautics and Astronautics

Thesis Supervisor

Certified by

Jonathan P. How

Associate Professor of Aeronautics and Astronautics

Thesis Advisor

Certified by

Eric M. Feron

Associate Professor of Aeronautics and Astronautics

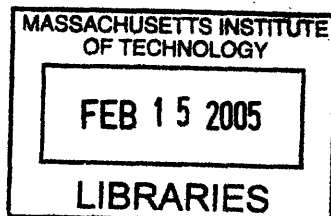
Thesis Advisor

Accepted by

Jaime Peraire

Professor of Aeronautics and Astronautics

Chair, Committee on Graduate Students



ARCHIVES

[This page intentionally left blank.]

A Gauss Pseudospectral Transcription for Optimal Control

by

David Benson

Submitted to the Department of Aeronautics and Astronautics
on October 21, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

A pseudospectral method for solving nonlinear optimal control problems is proposed in this thesis. The method is a direct transcription that transcribes the continuous optimal control problem into a discrete nonlinear programming problem (NLP), which can be solved by well-developed algorithms. The method is based on using global polynomial approximations to the dynamic equations at a set of Gauss collocation points. The optimality conditions of the NLP have been found to be equivalent to the discretized optimality conditions of the continuous control problem, which is not true of other pseudospectral methods. This result indicates that the method can take advantage of the properties of both direct and indirect formulations, and allows for the costates to be estimated directly from the Lagrange multipliers of the NLP.

The method has been shown empirically to have very fast convergence (exponential) in the states, controls, and costates, for problems with analytic solutions. This convergence rate of the proposed method is significantly faster than traditional finite difference methods, and has been demonstrated with many example problems.

The initial costate estimate from the proposed method can be used to define an optimal feedback law for real time optimal control of nonlinear problems. The application and effectiveness of this approach has been demonstrated with the simulated trajectory optimization of a launch vehicle.

Technical Supervisor: Tom Thorvaldsen
Title: Distinguished Member of the Technical Staff, Draper Laboratory

Thesis Supervisor: Steven R. Hall
Title: Professor of Aeronautics and Astronautics

Thesis Advisor: Jonathan P. How
Title: Associate Professor of Aeronautics and Astronautics

Thesis Advisor: Eric M. Feron
Title: Associate Professor of Aeronautics and Astronautics

[This page intentionally left blank.]

Acknowledgments

During my time at MIT, many people contributed to my success. One of the most significant contributions came from the Charles Stark Draper Laboratory and its staff. Without the help of Tom Thorvaldsen, Anil Rao, Matt Bottkol, and Ron Proulx, this research could not have been accomplished. Not only did you provide significant help, insight, and guidance, but also the weekly motivation to keep going. While I got a lot of support from MIT and its faculty, I would like to particularly thank professors Steve Hall, Jon How, and Eric Feron for being on my thesis committee and helping me through the doctoral program. I would also like to thank professor John Deyst for taking the time to help with my general exam and thesis defence.

To my officemate Geoff, thank you for all the good times. How would I have survived without the theme parties, taco nights, and poker? MIT is a lot more fun when you are around. I would like to give special thanks to Christine for the beers, pool, and friendship. Thursday night dinners will never be the same. Although most of you are long gone, thanks to Kim, Daveed, Steve, and Chris for helping make Thursday nights fun. Next, I would like to thank Jen for the Friday breakfasts and the candy bowl. Draper fellows past and present, thanks for the breakfasts, snacks, FYLF (forget your lunch Fridays), and all the good times.

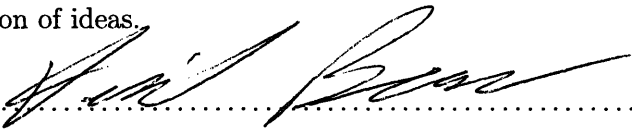
I would also like to thank you Dad for everything that you have given me. Thank you so much for being there to support me. So much of what I have accomplished is because of you.

Last (and certainly not least), I want to thank Leslie. Although the years apart were difficult, it is worth it in the end. Certainly I am glad we have been able to weather this storm. Keep up the hard work in medical school, and I will try to support you as much as you have supported me. Every time I felt like giving up, you were there to encourage me. You are the reason I was able to get through this and I love you so much.

So in case I forgot anybody else, thank you to everyone who helped make this possible.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Research and Development.

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.

David Benson.....

[This page intentionally left blank.]

[This page intentionally left blank.]

Contents

1	Introduction	19
1.1	Thesis Overview	24
2	Mathematical Background	27
2.1	Numerical Integration	27
2.1.1	Polynomial interpolation	28
2.1.2	Quadrature	29
2.2	Spectral Methods	33
2.2.1	Orthogonal Polynomials	34
2.2.2	Tau, Galerkin, Collocation	35
2.3	Static Optimization	37
2.3.1	Unconstrained Optimization	37
2.3.2	Equality Constrained Optimization	38
2.3.3	Inequality Constrained Optimization	40
2.4	Optimal Control	42
2.4.1	Calculus of Variations	42
2.4.2	Necessary Conditions	42
2.4.3	Pontryagin's Maximum Principle	46
3	Direct Transcription Methods	49
3.1	Euler Transcription	49
3.2	Runge-Kutta Transcription	51
3.3	Legendre Pseudospectral Method	52

3.3.1	Problem Formulation	53
3.3.2	KKT Conditions	55
4	Integral Gauss Pseudospectral Method	61
4.1	Analytical Necessary Conditions	62
4.1.1	Pontryagin's Maximum Principle	69
4.2	Discrete Transformation	71
4.2.1	Time intervals	72
4.2.2	Integration Approximation Matrix	72
4.2.3	Pseudospectral Transcription	80
4.3	KKT Conditions	83
4.3.1	Pontryagin	88
4.3.2	Free Time	92
4.3.3	Path Constraints	98
4.4	Costate Estimates	100
4.4.1	Initial Costate	102
4.5	Multiple Phases	103
4.5.1	Implementation	104
4.5.2	Costate Estimate	106
4.6	Polynomial Approximation	107
4.7	Summary	109
5	Differential Gauss Pseudospectral Method	111
5.1	Discretization	112
5.2	Differential - Integral Relation	115
5.3	Optimal Control Problem	117
5.4	KKT Conditions	118
5.5	Summary	123
6	Example Problems	125
6.1	LQR Problem	126

6.1.1	Euler Transcription	128
6.1.2	Runge-Kutta Transcription	130
6.1.3	Legendre Pseudospectral Method	131
6.1.4	Gauss Pseudospectral Method	132
6.1.5	LQR Problem Convergence	133
6.2	Nonlinear Example	135
6.2.1	Legendre Pseudospectral Method	136
6.2.2	Gauss Pseudospectral Method	137
6.2.3	Nonlinear Problem Convergence	140
6.3	Bang-Bang Control Problem	141
6.3.1	Legendre Pseudospectral Method	142
6.3.2	Gauss Pseudospectral Method	144
6.3.3	Bang-Bang Problem Convergence	147
6.3.4	Multiphase Solution	149
6.4	State Path Constraint Example	153
6.4.1	Gauss Pseudospectral Method	158
6.4.2	State Path Constraint Problem Convergence	159
6.4.3	Multiphase Solution	159
6.5	Brachistochrone Problem	164
6.6	Singular Arc Example	170
6.7	Multiple Solution Example	175
6.8	Low-Thrust Orbit Transfer Problem	177
6.8.1	Legendre Pseudospectral Method	178
6.8.2	Gauss Pseudospectral Method	179
6.8.3	Low-Thrust Orbit Problem Convergence	184
6.8.4	Differential and Integral Relation	186
6.9	Summary	189
7	Real Time Optimal Control	191
7.1	Examples	193

7.1.1	LQR Problem	193
7.1.2	Low-Thrust Orbit Problem	198
8	Launch Vehicle Example	203
8.1	Problem Description	203
8.1.1	Vehicle Properties	204
8.1.2	Dynamic Model	204
8.1.3	Optimal Control Formulation	206
8.2	Gauss Pseudospectral Implementation	208
8.2.1	Numerical Values	210
8.2.2	Results	212
8.3	Real Time Control	213
8.3.1	Results	216
8.4	Summary	218
9	Conclusion	219
A	MATLAB Code	223
A.1	Gauss Points	223
A.2	Gauss Weights	223
A.3	Integration Approximation Matrix	224
A.4	Differential Gauss Approximation Matrix	225
B	LQR Pseudospectral Solution	227
B.1	Indirect Solution	227
B.2	KKT Conditions	229
C	Orbital Elements	233
D	Launch Vehicle Costate Dynamics	235

List of Figures

2-1	Lagrange Interpolating Polynomials	29
2-2	Gauss Point Distribution	32
2-3	LGL Point Distribution	32
4-1	Integration Approximation Matrix Convergence	76
4-2	Initial Value Problem Convergence	80
4-3	Direct/Indirect Pseudospectral Solutions	87
4-4	Gauss Pseudospectral Discretization	88
6-1	LQR Exact Solution	129
6-2	LQR Approximate State Solution	131
6-3	LQR Approximate Control Solution	132
6-4	LQR Control Convergence	133
6-5	LQR Control Error vs. CPU Time	134
6-6	Nonlinear Approximate State Solution	137
6-7	Nonlinear Approximate Costate Solution	138
6-8	Nonlinear Problem State/Control Convergence	139
6-9	Nonlinear Problem Costate Convergence	140
6-10	Bang-Bang Problem, State 1 Solution	143
6-11	Bang-Bang Problem, State 2 Solution	143
6-12	Bang-Bang Control Solution	145
6-13	Bang-Bang Costate 1 Solution	145
6-14	Bang-Bang Costate 2 Solution	146
6-15	Bang-Bang Problem State/Control Convergence	147

6-16 Bang-Bang Problem Costate/Final Time Convergence	148
6-17 Bang-Bang State/Control Multiphase Solution	152
6-18 Bang-Bang Costate Multiphase Solution	153
6-19 Path Constraint, State 1 Solution	155
6-20 Path Constraint, State 2 Solution	156
6-21 Path Constraint, Control Solution	156
6-22 Path Constraint, Costate 1 Solution	157
6-23 Path Constraint, Costate 2 Solution	157
6-24 Path Constraint, Convergence	160
6-25 Path Constraint, Multiphase Convergence	163
6-26 Path Constraint, State 1 Multiphase Solution	163
6-27 Brachistochrone Problem	165
6-28 Brachistochrone Convergence	167
6-29 Brachistochrone Costate y Error	169
6-30 Brachistochrone Control Error	169
6-31 Singular Problem State Solution	172
6-32 Singular Problem Control Solution	173
6-33 Singular Problem Costate Solution	173
6-34 Singular Problem Convergence	174
6-35 Multiple Solution 1 State/Control	176
6-36 Multiple Solution 2 State/Control	176
6-37 Orbit Problem States	180
6-38 Orbit Problem Control	180
6-39 Orbit Problem Costates	182
6-40 Orbit Problem State Convergence	185
6-41 Orbit Problem Costate Convergence	185
6-42 Orbit Problem Control Convergence	186
6-43 Differential Form Sparsity Pattern	188
6-44 Integral Form Sparsity Pattern	189
6-45 Differential and Integral CPU Time	190

7-1	Real Time Control	192
7-2	LQR Exact Solution	194
7-3	LQR Pseudospectral Convergence	195
7-4	LQR Control Error, Open Outer Loop	196
7-5	LQR Propagated State	197
7-6	LQR Propagated Costate	197
7-7	Orbit Problem Real Time States, 5 Nodes	200
7-8	Orbit Problem Real Time Costates, 5 Nodes	200
7-9	Orbit Problem Real Time Control, 5 Nodes	201
7-10	Orbit Problem Real Time States, 10 Nodes	201
7-11	Orbit Problem Real Time Costates, 10 Nodes	202
7-12	Orbit Problem Real Time Control, 10 Nodes	202
8-1	Delta III Control Direction	211
8-2	Delta III Vehicle Altitude Profile	212
8-3	Delta III Vehicle Ground Track	213
8-4	Delta III Real Time Altitude Profile	216
8-5	Delta III Real Time Altitude Profile with Disturbance	217

[This page intentionally left blank.]

List of Tables

6.1	Low Thrust Orbit Optimality	184
7.1	Orbit Problem Real Time Results	199
8.1	Delta III Properties	204
8.2	Delta III Trajectory Errors	217
8.3	Delta III Trajectory Errors with Disturbance	218

[This page intentionally left blank.]

Chapter 1

Introduction

Optimal control problems arise in a wide range of applications, particularly in the aerospace industry. Optimal control problems can be defined for trajectory optimization, attitude control, and missile guidance, among others. The objective of optimal control theory is to determine the control (or controls) that will cause a system to meet a set of physical constraints and at the same time minimize (or maximize) some performance criterion. Acceptable criteria can be defined by any number of things, including time, fuel expenditure, or various trajectory parameters.

The solution to general optimal control problems can be found by applying the calculus of variations ([10] [45] [48]) and Pontryagin's maximum principle [53] to determine the first order necessary conditions for a solution. The necessary conditions reduce the optimal control problem to a two-point boundary value problem. The resulting boundary value problem is difficult or impossible to solve analytically for most problems, therefore numerical techniques are required to determine an approximation to the continuous solution. Many numerical techniques have been developed to solve optimal control problems. These numerical methods generally fall into the categories of indirect methods and direct methods ([6] [64]).

Indirect methods involve approximating the solution to the continuous necessary conditions. Some of these methods are multiple shooting ([54] [51]), finite difference [43], quasi-linearization [3], and collocation ([18] [58]), among others. The primary advantages of indirect methods is the high accuracy and assurances the solution satisfies

the necessary optimality conditions. However, indirect methods have significant disadvantages. First, the necessary optimality conditions must be derived analytically, for most problems this derivation is non-trivial. Second, the radius of convergence is typically small, therefore requiring a good initial guess. Third, a guess for the costates must be provided, which is difficult because the costate does not represent a physical entity such as the states and controls. Finally, the constrained and unconstrained arcs, or switching structure, for problems with path constraints must be known a priori.

Direct methods overcome some of the deficiencies of indirect methods by transforming the continuous optimal control problem into a nonlinear programming problem (NLP), which can be solved by well-developed algorithms. Direct methods have the advantages that the optimality conditions do not need to be derived; they have a large radius of convergence, therefore they do not need a good initial guess; there is no need for a guess of the costates; and finally the switching structure does not need to be known. Direct methods are not as accurate as indirect methods and many do not give any costate information. There are many types of direct methods. One approach, direct shooting, is to parameterize the controls and use explicit numerical integration to satisfy the differential constraints ([69] [46] [40] [8] [66] [59]). These methods suffer from the computationally expensive numerical integration that is required. Another approach is to parameterize both the states and controls. In these methods, piecewise polynomials can be used to approximate the differential equations at collocation points ([35] [7] [46] [21] [65] [33]). The states and controls can also be parameterized using global polynomials ([72] [73] [22] [23] [19] [20]). These methods are based on spectral methods which were extensively used to solve fluid dynamics problems and typically have faster convergence rates than traditional methods ([13] [26] [68]).

Spectral methods were largely developed in the 1970's for the numerical solution of partial differential equations. These methods were used as an extension to finite-element and finite-difference methods which were mostly explored during the previous two decades. Spectral methods were developed from a class of discretization schemes

for differential equations known generally as the method of weighted residuals [24].

Spectral representations have been used for studies of differential equations since Fourier in 1822 [27] and the idea of using them for numerical solutions of differential equations goes back to Lanczos in 1938 [47]. The spectral collocation method was first used by Slater [62] and Kantorovic [42] in 1934, but it was the work of Lanczos that established that a proper choice of trial functions and distribution of collocation points was crucial to the accuracy of the solution. The method was revived by Clenshaw [14], Norton [15], and Wright [70] in the late 1950's and early 1960's for Chebyshev polynomial expansions to initial value problems. In the 1970's the first unifying mathematical theory of spectral methods was compiled by Gottlieb and Orszag [31]. The application of spectral methods to problems in fluid dynamics and meteorology made the method famous, with the primary work done by Gottlieb, Orszag, Canuto, Hussaini, Quarteroni, and Zang ([13] [32] [52]). Orszag was first to use the term pseudospectral when referring to spectral collocation methods in 1972. Further development of spectral methods was done by Funaro [29], Fornberg [26], and Trefethen [68].

Spectral methods were applied to optimal control problems by Vlassenbroeck and Van Dooren ([72] [73]) in the late 1980's using Chebyshev polynomials and later Elnagar, Kazemi, and Razzaghi ([19] [20]), developed the Legendre pseudospectral method using Lagrange polynomials and collocation at Legendre-Gauss-Lobatto (LGL) points. An extension of the Legendre pseudospectral method was done by Fahroo and Ross ([22] [55]) to generate costate estimates.

The costate estimates are important for verifying the optimality of the solutions, mesh refinement, sensitivity analysis, and real time optimization. Costates can be determined for many direct methods. Some of these estimates are based on solving an approximation to the costate dynamics in post processing ([37] [49]). Other estimates are based on relationships between the Karush-Kuhn-Tucker (KKT) multipliers of the NLP and the continuous costates found by a sensitivity analysis [60], or relating the KKT conditions of the NLP to the continuous costate dynamics ([65] [34]). Recently costate estimates have been developed for pseudospectral methods. A costate map-

ping principle has been derived by Fahroo and Ross to define the relation between the KKT multipliers and costate estimates ([22] [55] [56]). However this principle, based on the relation between the KKT conditions of the NLP and the continuous costate dynamics, does not hold at the boundary points. This deficiency is a result of the defects in the discretization when using LGL points. The resulting costate estimates at the boundaries do not satisfy the costate dynamics or boundary conditions, but only a linear combination of the two.

The method proposed in this thesis is a direct transcription method using parameterization of the states and controls by global polynomials collocated at Gauss points. The method differs from other pseudospectral methods in the fact that the dynamic equations are not collocated at the boundary points. This approach is advantageous because the KKT conditions from the resulting NLP are exactly equivalent to the discretized form of the first-order necessary conditions of the optimal control problem. This means that the optimality conditions for the discretized problem are consistent with the continuous optimality conditions of the optimal control problem, which is not true for other pseudospectral methods. Only a special class of Runge-Kutta methods have been shown to have this property [33]. The consistency of the KKT conditions allow for an estimate of the costate based on the KKT multipliers from the NLP that does not suffer from a problem at the boundary points. This property allows for a costate estimate that is more accurate than other methods.

Furthermore, the costate at the initial time can be estimated very accurately using the Gauss pseudospectral method. The initial costate, along with the initial state, defines the entire solution of the optimal control problem by reducing the two point boundary value problem resulting from the necessary conditions to an initial value problem, where the control can be found by applying Pontryagin's maximum principle. By having a very good estimate of the initial costate, a dynamic optimization problem can be reduced to a series of static optimization problems [38]. Therefore, the initial costate can be used to define a feedback law resulting in real time optimal control for nonlinear systems.

The Gauss pseudospectral method has been shown to satisfy the optimality con-

ditions for a large class of problems. The KKT conditions resulting from problems with free initial or final times, problems with state and control path constraints, and problems involving multiple phases, have been shown to be consistent with the continuous first-order necessary conditions. The proposed method has also been shown to satisfy Pontryagin's maximum principle. This analysis indicates that the Gauss pseudospectral method can be used to solve a large set of problems, shown with many numerical examples.

The Gauss pseudospectral method was originally developed from the integral form of the optimal control problem. In this form, the optimal control problem is discretized using pseudospectral approximations of the integral of the differential dynamic equations. This was done to facilitate the discretization at Gauss points while still enforcing the boundary conditions. However, the differential form of the optimal control problem can also be discretized at Gauss points in a way that allows the boundary conditions to be enforced. It has been shown that this differential form of the Gauss pseudospectral method is mathematically equivalent to the integral form of the Gauss pseudospectral method. The KKT conditions of the resulting NLP from the differential and integral forms of the Gauss method have been shown to be consistent with the continuous first-order necessary conditions. This property is a result of the fact that the dynamic equations are collocated only at Gauss points, which do not include the boundary points. The differential form of the method has the advantage that the resulting NLP is more sparse than the NLP from the integral form. Therefore, numerical solvers can take advantage of this sparsity to solve the NLP with less computation time for large problems.

The Gauss pseudospectral method derived in this thesis has many advantages over other numerical methods for solving optimal control problems. The first and most important, is that the KKT conditions for the NLP are exactly equivalent to the discretized first-order necessary conditions. This property indicates that the solution to the NLP is mathematically equivalent to the solution to the discrete optimality conditions, meaning that the NLP solution is guaranteed to satisfy the pseudospectral approximation to the optimality conditions. This distinction signifies that the direct

and indirect solutions are the same. Therefore, the Gauss pseudospectral method can take advantage of the properties of both types of methods. An accurate solution can be found using well-developed sparse NLP solvers with no need for an initial guess on the costates or derivation of the necessary conditions. The method is robust and can handle path constraints without knowledge of the switching structure. Another consequence of the consistency of the KKT conditions is that the costates can be estimated directly from the KKT multipliers of the NLP. Second, the Gauss method have been shown to be consistent for a large class of problems including those with path constraints, free time, and requiring Pontryagin's maximum principle. The final advantage of the Gauss pseudospectral method, is that they take advantage of the very fast exponential convergence typical of spectral methods. This convergence rate has been shown empirically for many example problems. The fast convergence rate indicates that an accurate solution to the optimal control problem can be found using fewer nodes and hence less computation time. The rapid solution of the problems along with an accurate estimate for the costate can be used to realize real time optimal control for nonlinear systems.

1.1 Thesis Overview

Chapter 2 describes the basic mathematical background for the formulation and analysis of pseudospectral methods for optimal control. The concepts of numerical integration and polynomial interpolation are explored, as well as a brief introduction to the theory of spectral methods. Also included in this chapter is a discussion of optimization methods. Static optimization involving both inequality and equality constraints are examined, including the derivation of the Karush-Kuhn-Tucker (KKT) conditions, which define the solution to the optimization problem. Finally optimal control theory is described including the derivation of the first order necessary conditions and Pontryagin's maximum principle.

In Chapter 3, several direct transcription methods are described including an Euler, Runge-Kutta, and the Legendre pseudospectral method. The KKT conditions

for the Legendre pseudospectral method are derived to show how the costate mapping principle is defined as well as the defects in the costate estimates.

Chapter 4 describes the integral formulation of the continuous optimal control problem as well as the derivation of the first order necessary conditions. The pseudospectral transcription at Gauss points of the integral form of the problem is explored, which leads to the derivation of the KKT conditions for the proposed method. The KKT conditions are shown to be consistent with the first order necessary conditions, for a large class of problems. Finally the method for the costate estimates, including the initial costate, is described.

In Chapter 5 the Gauss pseudospectral transcription of the differential form of the problem is derived. It is shown that the Gauss pseudospectral transcription is mathematically equivalent to the integral pseudospectral transcription. It is also shown that the KKT conditions from the Gauss pseudospectral transcription are consistent with the first-order necessary conditions, and a costate estimate from the KKT multipliers can be defined.

In Chapter 6 many example problems of different types are solved by several numerical methods to demonstrate the advantages and disadvantages of the Gauss pseudospectral method. The types of example problems solved are linear, nonlinear, those which have discontinuous solutions in the control and costates, problems with singular arcs, and problems with multiple local minimum. A comparison of both the integral and differential forms of the method is made on a common orbit transfer problem.

Chapter 7 tests the use of the initial costate for real time optimal control. Several examples demonstrating the method are included for linear and nonlinear problems.

Chapter 8 describes a launch vehicle trajectory optimization problem. The vehicle dynamics are described as well as the formulation and solution of the resulting optimal control problem. The real time control algorithm described in Chapter 7 is used to demonstrate the real time control of the launch vehicle in the presence of a disturbance. Finally Chapter 9 includes the final summary and concluding remarks.

[This page intentionally left blank.]

Chapter 2

Mathematical Background

In this chapter, we review many mathematical concepts that aid in the understanding of the proposed pseudospectral method. The first and most important is the idea of polynomial interpolation using a basis of Lagrange polynomials. Polynomial interpolation is used in the development of quadratures, for numerical integration.

Another important concept is the application of spectral methods for the solution of differential equations. A brief outline of the most common types of spectral methods is included, as well as the background for spectral collocation or pseudospectral methods, which are the most important for this thesis.

Also included in this chapter is the development and solution methods of static optimization problems or nonlinear programs (NLP). Unconstrained, equality constrained, and inequality constrained problems are considered, along with the derivation of the necessary conditions or Karush-Kuhn-Tucker (KKT) conditions for a solution of each.

Finally a review of basic optimal control theory is included, along with the derivation of the first-order necessary conditions and Pontryagin's maximum principle.

2.1 Numerical Integration

Explicit closed form solutions cannot be found for integrals of most functions. Therefore numerical procedures must be used to approximate the value of the integral. The

methods of numerical integration considered in this thesis are based on polynomial interpolation.

2.1.1 Polynomial interpolation

The interpolation formula of Lagrange [63] is based on the fact that given N arbitrary support points of the function $f(t)$, on the interval $t_i \in [a, b]$,

$$(t_i, f_i), \quad i = 1, \dots, N, \quad t_i \neq t_k, \quad i \neq k, \quad (2.1)$$

there exists a unique polynomial $P(t)$, of degree $N - 1$ so that

$$P(t_i) = f_i, \quad i = 1, \dots, N. \quad (2.2)$$

The unique polynomial can be found using the Lagrange interpolation formula, so that

$$P(t) = \sum_{i=1}^N f_i \cdot L_i(t), \quad (2.3)$$

where $L_i(t)$ are the Lagrange interpolation polynomials [17]. These polynomials can be found using the formula,

$$L_i(t) = \prod_{k=1, k \neq i}^N \frac{(t - t_k)}{(t_i - t_k)}. \quad (2.4)$$

These Lagrange polynomials have the property that they are one at the i th support, and zero at all the others, so that

$$L_i(t_k) = \delta_{ik} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}. \quad (2.5)$$

An example of several Lagrange polynomials are shown in Fig. 2-1. The error in the Lagrange interpolation formula for functions in which N derivatives exist is known

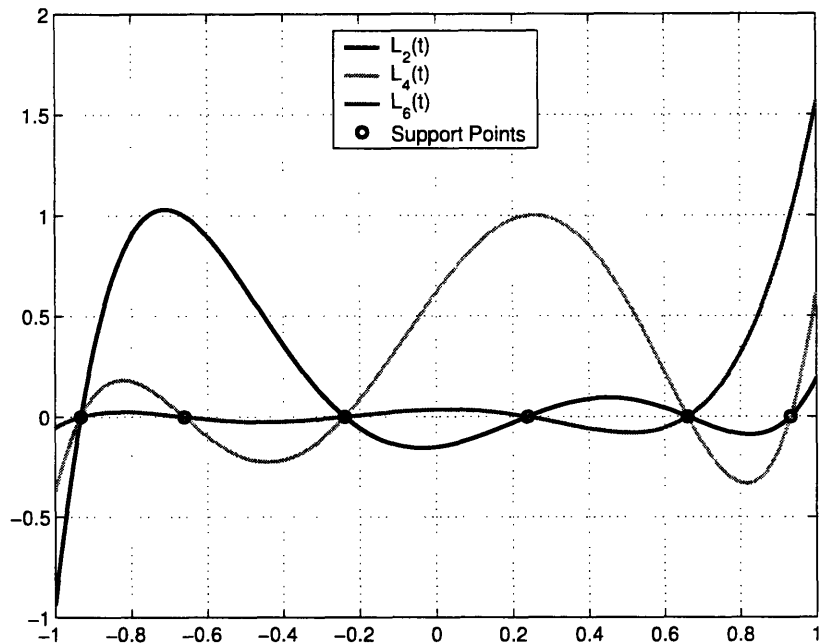


Figure 2-1: Lagrange Interpolating Polynomials

to be [41]

$$R(t) \doteq f(t) - P(t) = \frac{1}{N!} \prod_{i=1}^N (t - t_k) \cdot \frac{d^{(N)}f}{dt^{(N)}}(\zeta), \quad (2.6)$$

where $\zeta \in [a, b]$. This equation shows rapid convergence for functions whose derivatives are bounded.

2.1.2 Quadrature

Quadratures are a common approach to the numerical approximation of integrals. A numerical quadrature is an approximation to a definite integral in the form,

$$\int_a^b f(t) dt \approx \sum_{i=1}^N \alpha_i \cdot f(t_i), \quad (2.7)$$

where α_i are the quadrature weights and t_i are the quadrature points or nodes. An interpolatory quadrature formula can be created for arbitrary support points by ap-

proximating the integrand by Lagrange polynomials [41], so that

$$\int_a^b f(t) dt \approx \int_a^b \sum_{i=1}^N L_i(t) \cdot f(t_i) dt . \quad (2.8)$$

The quadrature weights can be easily determined as

$$\alpha_i = \int_a^b L_i(t) dt . \quad (2.9)$$

Quadrature formula of this type have a degree of precision of $N - 1$, which means they are exact for polynomials of degree $N - 1$ or less.

The quadrature formula with the maximum degree of precision is the Gauss quadrature formula, which is exact for polynomials of degree $2N - 1$ or less. The Gauss formula is found by choosing the weights w_i and points t_i which make the formula exact for the highest degree polynomial possible [16]. The points and weights are determined so that

$$\int_{-1}^1 f(t) dt = \sum_{i=1}^N w_i \cdot f(t_i) + E_N , \quad (2.10)$$

and the error, E_N , is zero for a polynomial, $f(t)$, of degree $2N - 1$. The Gauss points are determined as the zeros of the N th degree Legendre polynomial [12] and the weights are the integrals of the resulting Lagrange interpolating polynomials, so that

$$w_i = \int_{-1}^1 \prod_{k=1, k \neq i}^N \frac{(t - t_k)}{(t_i - t_k)} dt , \quad i = 1, \dots, N , \quad (2.11)$$

where t_i are the zeros of the N th degree Legendre polynomial. The Gauss weights can also be determined using the formula [36],

$$w_i = \frac{2}{1 - t_i^2} \left[\dot{P}_N(t_i) \right]^2 , \quad (2.12)$$

where $\dot{P}_N(t)$ is the derivative of the Legendre polynomial of degree N . The Gauss points are all interior to the interval $[-1, 1]$ and tend to be more dense near the

boundaries. A sample distribution of Gauss points for several numbers of nodes are shown in Fig. 2-2. The error in the Gauss quadrature formula is proportional to the $(2N)$ th derivative of the integrand ([44] [36]), so that

$$E_N = \frac{1}{(2N)!} \frac{d^{2N} f}{dt^{2N}}(\zeta) \int_a^b \left(\prod_{i=1}^N (t - t_i) \right)^2 dt = \frac{2^{2N+1} (N!)^4}{(2N+1) [(2N)!]^3} \frac{d^{2N} f}{dt^{2N}}(\zeta), \quad \zeta \in [a, b]. \quad (2.13)$$

Tables of Gauss points and weights can be found in [36].

The Gauss-Lobatto (LGL) quadrature formula is similar to the Gauss formula, except the boundary points are fixed at -1 and 1 . The formula is created by choosing weights w_i and $N - 2$ remaining points t_i , to integrate the highest degree polynomial possible with zero error, so that

$$\int_{-1}^1 f(t) dt = \sum_{i=1}^N w_i \cdot f(t_i) + E_N, \quad t_1 = -1, \quad t_N = 1. \quad (2.14)$$

Because two degrees of freedom have been removed, the LGL quadrature formula is exact for polynomials of degree $2N - 3$. The LGL points are determined to be the zeros of the derivative of the Legendre polynomial of degree $N - 1$, $\dot{P}_{N-1}(t)$, plus the two end points, $-1, 1$ [16]. The weights are determined as

$$w_i = \frac{2}{N(N-1)[P_{N-1}(t_i)]^2}, \quad t_i \neq \pm 1. \quad (2.15)$$

The weights at the boundary points are $w_1, w_N = \frac{2}{N(N-1)}$. The error in the LGL quadrature formula is

$$E_N = \frac{-N(N-1)^3 2^{2N+1} [(N-2)!]^4}{(2N-1) [(2N-2)!]^3} \frac{d^{(2N-2)} f}{dt^{(2N-2)}}(\zeta), \quad \zeta \in [a, b]. \quad (2.16)$$

Tables of LGL points and weights can be found in [36]. A plot of the distribution of LGL points is shown in Fig. 2-3.

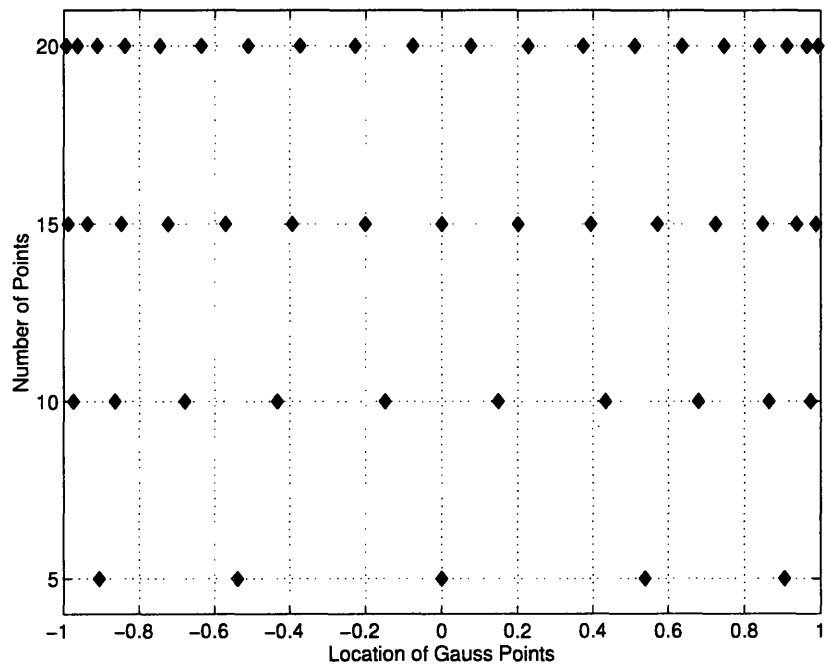


Figure 2-2: Gauss Point Distribution

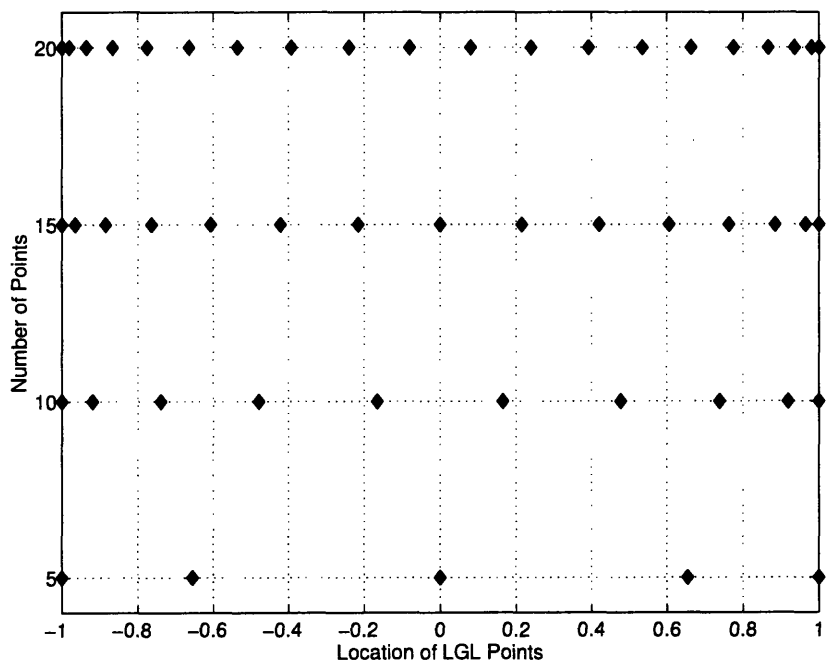


Figure 2-3: LGL Point Distribution

2.2 Spectral Methods

Spectral methods are used to numerically approximate the solution to partial or ordinary differential equations, and are characterized by the trial functions and test functions. The trial functions are used as a basis for the truncated series expansion of the solution, while the test functions are used to ensure that the differential equation is satisfied as closely as possible.

The choice of trial functions is what distinguishes spectral methods from finite-difference and finite-element methods. Finite-difference methods approximate derivatives using local approximations. In the case of finite-element methods, the domain is divided into small elements with a trial function for each element. The resulting trial functions are therefore local in character. Trial functions for spectral methods are infinitely differentiable global functions [26], so that

$$x(t) \approx \sum_{k=1}^N a_k \cdot \phi_k(t), \quad (2.17)$$

where the basis functions, $\phi_k(t)$, are for example orthogonal polynomials or trigonometric functions. The global nature of the trial functions is the primary distinguishing characteristic of spectral methods.

The test functions used distinguish the three most common types of spectral methods, Galerkin, tau, and collocation. In the Galerkin approach, the test functions are the same as the trial functions, which are infinitely smooth functions that satisfy the boundary conditions. The differential equation is enforced by requiring that the residual, the error in the differential equation produced by using the truncated expansion, be orthogonal to all the test functions. Tau methods are similar to the Galerkin methods in the way the differential equation is enforced, but the test functions do not satisfy the boundary conditions so an extra set of equations must be used to apply the boundary conditions. In the collocation approach, the test functions used are Dirac delta functions centered at a set of collocation points. This approach requires the differential equation to be satisfied exactly at the collocation points.

Spectral methods have the notable strength that for analytic functions, errors typically decay exponentially as N increases, rather than the much slower polynomial rates typical of finite difference methods [31]. For periodic problems, trigonometric basis functions are generally used because of the rapid convergence to smooth periodic functions. For non-periodic problems, orthogonal polynomials of Jacobi type have proven to be the most useful, with Chebyshev and Legendre polynomials as the most important special cases.

2.2.1 Orthogonal Polynomials

Orthogonal polynomials of Chebyshev and Legendre, are commonly used in spectral methods. The polynomials belong to a larger class of Jacobi polynomials. The class of Jacobi polynomials are all the eigenfunctions of a singular Sturm-Liouville problem on the interval $[-1, 1]$. The special case for the Legendre polynomials are eigenfunctions of the singular Sturm-Liouville problem [13],

$$\frac{d}{dt} \left((1 - t^2) \cdot \frac{dP_N}{dt}(t) \right) + N(N + 1) \cdot P_N(t) = 0, \quad (2.18)$$

with the normalization $P_N(1) = 1$. Legendre polynomials are orthogonal with respect to the inner product,

$$\langle f(t), g(t) \rangle = \int_{-1}^1 f(t) \cdot g(t) dt, \quad (2.19)$$

and satisfy the three term recursion

$$(N + 1) \cdot P_{N+1}(t) = (2N + 1) \cdot t \cdot P_N(t) - N \cdot P_{N-1}(t), \quad P_0(t) = 1, \quad P_1(t) = t. \quad (2.20)$$

Chebyshev polynomials of the first kind are the eigenfunctions of the singular Sturm-Liouville problem,

$$\frac{d}{dt} \left(\sqrt{1 - t^2} \cdot \frac{dT_N}{dt}(t) \right) + \frac{N^2}{\sqrt{1 - t^2}} \cdot T_N(t) = 0. \quad (2.21)$$

Using the normalization $T_N(1) = 1$ leads to the polynomials of

$$T_N(t) = \cos(N \cdot \theta), \quad \theta = \arccos t, \quad (2.22)$$

which are orthogonal with respect to the inner product

$$\langle f(t), g(t) \rangle = \int_{-1}^1 \frac{f(t) \cdot g(t)}{\sqrt{1-t^2}} dt. \quad (2.23)$$

2.2.2 Tau, Galerkin, Collocation

There are three fundamental types of spectral methods, tau, Galerkin, and collocation [13]. They differ in the choice of test functions that keep the residual, or error in the differential equation small.

Consider a general differential equation for $\mathbf{x}(t) \in \mathbb{R}^n$, so that

$$\mathbf{F}(\dot{\mathbf{x}}(t), \mathbf{x}(t), t) = \mathbf{0}, \quad t \in [-1, 1], \quad (2.24)$$

where $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$. The boundary conditions are, $\beta(\mathbf{x}(-1), \mathbf{x}(1)) = \mathbf{0}$, with $\beta : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^q$. The solution $\mathbf{x}(t)$ is approximated using a basis of orthogonal polynomials, $\phi_k(t)$, as trial functions, so that

$$\mathbf{x}(t) \approx \mathbf{x}_N(t) = \sum_{k=1}^N \mathbf{a}_k \cdot \phi_k(t). \quad (2.25)$$

Using this approximation, the residual of the differential equation is

$$\mathbf{R}_N(t) = \mathbf{F}(\dot{\mathbf{x}}_N(t), \mathbf{x}_N(t), t), \quad (2.26)$$

where $\mathbf{R}_N(t)$ is the residual.

The objective of the tau method [26] is to determine the coefficients a_k so that the approximate solution satisfies the boundary conditions, $\beta(\mathbf{x}_N(-1), \mathbf{x}_N(1)) = \mathbf{0}$,

and the residual is orthogonal to as many basis functions as possible, so that

$$\langle \mathbf{R}_N, \phi_k \rangle = \mathbf{0}, \quad k = 1, \dots, N - q. \quad (2.27)$$

In the Galerkin method [31], the basis functions, $\phi_k(t)$, are combined to form a new set of basis functions, $\tilde{\phi}_k(t)$, which span the space of polynomials that satisfy the boundary conditions, $\beta = 0$. The solution is then approximated using the new set of basis functions,

$$\mathbf{x}(t) \approx \mathbf{x}_N(t) = \sum_{k=1}^N \mathbf{b}_k \cdot \tilde{\phi}_k(t), \quad (2.28)$$

where the objective is to find the coefficients, \mathbf{b}_k , so that the residual \mathbf{R}_N is orthogonal to as many basis functions as possible, so that

$$\langle \mathbf{R}_N, \tilde{\phi}_k \rangle = \mathbf{0}, \quad k = 1, \dots, N. \quad (2.29)$$

Note that the boundary conditions are automatically satisfied because the polynomial $\mathbf{x}_N(t)$ is a linear combination of the basis functions $\tilde{\phi}_k(t)$ that span the space of polynomials that satisfy the boundary conditions.

Finally, in the collocation or pseudospectral method [26], the solution $\mathbf{x}(t)$ is approximated using the basis of polynomials, $\phi_k(t)$,

$$\mathbf{x}(t) \approx \mathbf{x}_N(t) = \sum_{k=1}^N \mathbf{a}_k \cdot \phi_k(t), \quad (2.30)$$

where the objective is to determine the coefficients \mathbf{a}_k so that the boundary conditions are satisfied, $\beta(\mathbf{x}_N(-1), \mathbf{x}_N(1)) = 0$, and the residual is zero at a set of collocation points. In this method the test functions are chosen as Dirac delta functions, so that

$$\mathbf{R}_N(t_i) = \mathbf{0}, \quad i = 1, \dots, N - q. \quad (2.31)$$

The pseudospectral technique can be used with the basis of Lagrange polynomials

formed from the collocation points, so that

$$\mathbf{x}(t) \approx \mathbf{x}_N(t) = \sum_{k=1}^N \mathbf{a}_k \cdot L_k(t) . \quad (2.32)$$

This approach is advantageous because the coefficients of the Lagrange polynomials are equal to the value of the approximating polynomial at the collocation points,

$$\mathbf{x}_N(t_k) = \mathbf{a}_k . \quad (2.33)$$

This section describes the fundamental differences between the most common types of spectral methods. The collocation or pseudospectral method is the most important technique that is applied in this thesis.

2.3 Static Optimization

Another important concept in the understanding of the method proposed in this thesis, is the development and solution to static optimization problems, or nonlinear programs (NLP). Static optimization problems are a class of parameter optimization problems where time is not a parameter. The objective is to find the set of parameters that minimize some cost function that could be subject to a set of algebraic equality or inequality constraints.

2.3.1 Unconstrained Optimization

The objective of unconstrained optimization is to find the set of parameters, $x \in \mathbb{R}^n$, that give a minimum value to a scalar function, $J(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ [48]. A local minimum occurs when any arbitrary increment of the variable x , produces an increase in the function, J , so that

$$J(x + dx) \geq J(x) , \quad \forall dx . \quad (2.34)$$

The conditions for a solution are determined by first writing the Taylor series expansion for an increment in J . The expansion is

$$dJ = J_x^T \cdot dx + \frac{1}{2} dx^T \cdot J_{xx} \cdot dx + \text{higher order terms} , \quad (2.35)$$

where the higher order terms are of order three and higher. The gradient of the function, J , with respect to x is a vector and the Hessian of the function with respect to x is a matrix, so that

$$J_x \doteq \frac{\partial J}{\partial x} \in \mathbb{R}^n \quad J_{xx} \doteq \frac{\partial^2 J}{\partial x^2} \in \mathbb{R}^{n \times n} \quad (2.36)$$

For the increment in J to be positive for any increment of x , two conditions must be met. First, the gradient of the function must be zero. Second, the Hessian matrix must be positive definite. These conditions are

$$J_x = 0 , \quad J_{xx} > 0 . \quad (2.37)$$

These conditions define the sufficient conditions for a local minimum. The gradient condition is a set of n conditions to determine the n unknown variables x . The gradient condition by itself is a necessary condition (or optimality condition) of an extremal and could define local minimum, maximum, or saddle points. There are many numerical algorithms for solving unconstrained nonlinear problems, such as gradient methods, Newton's method, and conjugate direction methods [4].

2.3.2 Equality Constrained Optimization

The objective of equality constrained optimization is to minimize a scalar function, $J(x)$, that is subject to a set of $q < n$ constraints, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^q$, so that

$$f(x) = 0 . \quad (2.38)$$

A classical approach to finding the solution is to define the Lagrangian by augmenting the cost with a set of Lagrange multipliers [7], so that

$$L(x, \lambda) = J(x) - \lambda^T \cdot f(x) = J(x) - \sum_{k=1}^q \lambda_k \cdot f_k(x), \quad (2.39)$$

where $\lambda \in \mathbb{R}^q$ is a vector of Lagrange multipliers. Note that the constraints are subtracted from the cost in the Lagrangian. This convention of subtraction is used throughout this thesis. The conditions for a minimum are found by taking a first-order expansion of an increment in $L(x, \lambda)$, so that

$$dL = L_x \cdot dx + L_\lambda \cdot d\lambda. \quad (2.40)$$

For a point (x^*, λ^*) to be a minimum, the gradient of L with respect to the variables x and λ must be zero. These two conditions result in $n + q$ constraints to determine the $n + q$ unknowns of x and λ . These constraints are

$$\begin{aligned} L_x &= J_x - G^T \cdot \lambda, & G &\doteq \frac{\partial f}{\partial x} \in \mathbb{R}^{q \times n}, \\ L_\lambda &= -f(x). \end{aligned} \quad (2.41)$$

The partial derivative of the constraints, G , is called the Jacobian. This result is from the Lagrange Multiplier Theorem [4]. These conditions do not distinguish between a point that is a maximum, minimum, or a saddle point. For the stationary point to be a minimum, conditions on the curvature of the Lagrangian must be met. The Hessian of the Lagrangian is defined to be

$$H_L = \frac{\partial^2 L}{\partial x^2} = \frac{\partial^2 J}{\partial x^2} - \sum_{k=1}^q \lambda_k \frac{\partial^2 f_k}{\partial x^2}. \quad (2.42)$$

Therefore, a sufficient condition for a minimum is that

$$v^T \cdot H_L \cdot v > 0, \quad (2.43)$$

for any vector $v \in \mathbb{R}^n$ in the constrained tangent space. This requirement is such that any feasible variation, v , of the unknown variables, x , must satisfy the constraint, $f(x^* + v) = 0$.

2.3.3 Inequality Constrained Optimization

The final case of static optimization problems is when the constraints of the problem are inequality constraints. The problem is then to find the variables, x , to minimize a function, J , subject to a set of r inequality constraints, $m(x) : \mathbb{R}^n \rightarrow \mathbb{R}^r$, so that

$$m(x) \leq 0. \quad (2.44)$$

In contrast to equality constraint problems, the number of inequality constraints can exceed the number of unknown variables, x . Any set of points that satisfy all the constraints is called feasible. The set of all feasible points is called the feasible region. At the optimal solution, x^* , some of the constraints may be satisfied as equality constraints,

$$m_i(x^*) = 0, \quad \forall i \in \mathbb{A}, \quad (2.45)$$

and some constraints may be strictly satisfied,

$$m_i(x^*) < 0, \quad \forall i \in \mathbb{A}'. \quad (2.46)$$

\mathbb{A} is called the active set while \mathbb{A}' is called the inactive set. The active set of constraints can be treated exactly as equality constraints and the inactive set can be ignored. The problem then is to determine which of the constraints are active and which ones are inactive. The conditions for a solution to an inequality constrained problem are found by applying the Karush-Kuhn-Tucker optimality conditions [4].

Consider a problem with both equality and inequality constraints. The objective is to minimize a cost function, $J(x)$, subject to a set of equality constraints, $f(x)$,

and inequality constraints $m(x)$. The problem is then

$$\begin{aligned} & \text{minimize} && J(x) \\ & \text{subject to} && f(x) = 0, \\ & && m(x) \leq 0. \end{aligned} \tag{2.47}$$

The Karush-Kuhn-Tucker (KKT) conditions are found by first defining the Lagrangian, or augmented cost function, L , so that

$$L(x, \lambda, \mu) = J(x) - \sum_{k=1}^q \lambda_k \cdot f_k(x) - \sum_{i=1}^r \mu_i \cdot m_i(x), \tag{2.48}$$

where λ_k, μ_i are the Lagrange multipliers or KKT multipliers. Note that both the equality and inequality constraints were subtracted from the cost to keep with the convention of subtraction. The gradient of the Lagrangian defines the conditions for an optimal solution, x^* . If J, f , and m are continuously differentiable functions then there exists Lagrange multipliers λ^* and μ^* such that

$$\begin{aligned} \frac{\partial L}{\partial x}(x^*, \lambda^*, \mu^*) &= 0, \\ \mu_i^* &\leq 0, \quad \forall i, \\ \mu_i^* \cdot m_i(x^*) &= 0, \quad \forall i. \end{aligned} \tag{2.49}$$

These conditions along with $f(x^*) = 0$ and $m(x^*) \leq 0$ make up the KKT conditions. The KKT conditions are necessary for local stationary points but do not distinguish between minimum, maximum, or saddle points.

There are many well-developed algorithms and software packages that have been created to solve optimization problems. The software used in this research was primarily SNOPT [30], which uses a sequential quadratic programming algorithm (SQP) to solve large sparse optimization problems.

2.4 Optimal Control

Optimal control problems arise in a wide range of applications. The analytic solution to optimal control problems can be found by using the calculus of variations to determine the first-order optimality conditions.

2.4.1 Calculus of Variations

The calculus of variations is a useful tool in solving minimization problems that are dependent on continuous functions of time. The minimization of a functional, $J(\mathbf{x}(t))$, is found by requiring that the variation of the cost on an extreme path \mathbf{x}^* vanish for all admissible variations $\delta\mathbf{x}$ [45], so that

$$\delta J(\mathbf{x}^*, \delta\mathbf{x}) = 0. \quad (2.50)$$

This is used along with Lagrange theory to determine the conditions for a minimum of a constrained cost functional.

2.4.2 Necessary Conditions

For optimal control problem the calculus of variations is used to determine the necessary conditions for a local minimum.

Consider the optimal control problem where the objective is to find the states $\mathbf{x}(t) \in \mathbb{R}^n$ and controls $\mathbf{u}(t) \in \mathbb{R}^m$ on the interval $t \in [t_o, t_f]$ that minimizes a cost in Bolza form. The cost is

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_{t_o}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt, \quad (2.51)$$

where $\Phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ is the terminal cost, and $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}$ is the integrated cost. The states are subject to the differential dynamic constraints,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2.52)$$

where $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$. The states are also subject to the boundary conditions in their most general form,

$$\phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) = \mathbf{0}, \quad (2.53)$$

where $\phi : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^q$.

The first-order necessary conditions are found by applying the calculus of variations to the Lagrangian or augmented cost function. The augmented cost function is created by adjoining the costate, $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$, and Lagrange multipliers, $\boldsymbol{\nu} \in \mathbb{R}^q$, to the differential and boundary constraints, so that

$$\begin{aligned} J_a = & \Phi(\mathbf{x}(t_f), t_f) - \boldsymbol{\nu}^T \cdot \phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) \\ & + \int_{t_o}^{t_f} \left[g(\mathbf{x}(t), \mathbf{u}(t), t) - \boldsymbol{\lambda}^T(t) \cdot \left(\frac{d\mathbf{x}}{dt} - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \right) \right] dt. \end{aligned} \quad (2.54)$$

Taking the first-order variation of the Lagrangian with respect to all free variables results in

$$\begin{aligned} \delta J_a = & \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \delta \mathbf{x}_f + \frac{\partial \Phi}{\partial t_f} \delta t_f - \delta \boldsymbol{\nu}^T \phi - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_o)} \delta \mathbf{x}_o - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} \delta t_o \\ & - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \delta \mathbf{x}_f - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} \delta t_f + (g - \boldsymbol{\lambda}^T \cdot (\dot{\mathbf{x}} - \mathbf{f})) \Big|_{t=t_f} \delta t_f \\ & - (g - \boldsymbol{\lambda}^T \cdot (\dot{\mathbf{x}} - \mathbf{f})) \Big|_{t=t_o} \delta t_o + \int_{t_o}^{t_f} \left[\frac{\partial g}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial g}{\partial \mathbf{u}} \delta \mathbf{u} \right. \\ & \left. - \delta \boldsymbol{\lambda}^T \cdot (\dot{\mathbf{x}} - \mathbf{f}) + \boldsymbol{\lambda}^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} + \boldsymbol{\lambda}^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} - \boldsymbol{\lambda}^T \delta \dot{\mathbf{x}} \right] dt. \end{aligned} \quad (2.55)$$

Integrating the last term in the integrand by parts removes the variation with respect to the derivative of the states, so that

$$\int_{t_o}^{t_f} -\boldsymbol{\lambda}^T \delta \dot{\mathbf{x}} dt = -\boldsymbol{\lambda}^T(t_f) \delta \mathbf{x}(t_f) + \boldsymbol{\lambda}^T(t_o) \delta \mathbf{x}(t_o) + \int_{t_o}^{t_f} \dot{\boldsymbol{\lambda}}^T \delta \mathbf{x} dt. \quad (2.56)$$

It is important to note that there is a difference between the variation of the final state, $\delta \mathbf{x}_f$, and the variation of the state at the final time, $\delta \mathbf{x}(t_f)$, because the final time is free. The relation between the two can be found using the derivative of the

state and the variation of the final time. The same is true of the initial state, so that

$$\begin{aligned}\delta \mathbf{x}_f &= \delta \mathbf{x}(t_f) + \dot{\mathbf{x}}(t_f) \cdot \delta t_f, \\ \delta \mathbf{x}_0 &= \delta \mathbf{x}(t_0) + \dot{\mathbf{x}}(t_0) \cdot \delta t_0.\end{aligned}\tag{2.57}$$

Substituting the results of (2.57) into (2.56) and then into (2.55) and combining terms, defines the necessary conditions for optimality. The variation of the augmented cost is

$$\begin{aligned}\delta J_a &= \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\lambda}^T(t_f) \right) \delta \mathbf{x}_f + \left(-\boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_0)} + \boldsymbol{\lambda}^T(t_0) \right) \delta \mathbf{x}_0 \\ &\quad - \delta \boldsymbol{\nu}^T \phi + \left(-\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_0} - g(t_0) - \boldsymbol{\lambda}^T(t_0) \cdot \mathbf{f}(t_0) \right) \delta t_0 \\ &\quad + \left(\frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + g(t_f) + \boldsymbol{\lambda}^T(t_f) \cdot \mathbf{f}(t_f) \right) \delta t_f + \int_{t_0}^{t_f} \left[-\delta \boldsymbol{\lambda}^T \cdot (\dot{\mathbf{x}} - \mathbf{f}) \right. \\ &\quad \left. + \left(\frac{\partial g}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \dot{\boldsymbol{\lambda}} \right) \delta \mathbf{x} + \left(\frac{\partial g}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} \right] dt.\end{aligned}\tag{2.58}$$

The optimality conditions define a stationary point so that any arbitrary variations in any of the free variables results in no change in the total cost. The optimality conditions are

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \\ -\dot{\boldsymbol{\lambda}}^T(t) &= \frac{\partial g}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}}.\end{aligned}\tag{2.59}$$

The resulting boundary value problem is defined as a set of nonlinear (in general) coupled first-order differential equations. Therefore, there will be $2n$ integration constants needed to find the solution. The total number of unknown constants is $2n+q+2$. There are $2n$ integration constants, q Lagrange multipliers, and the initial and final time.

The last function that has to be defined is the control. It is assumed in this case that the control is uniquely defined by the relation,

$$\mathbf{0} = \frac{\partial g}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}}.\tag{2.60}$$

The boundary conditions define the q Lagrange multipliers,

$$\phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) = \mathbf{0}, \quad (2.61)$$

and the boundary conditions on the costate, $\boldsymbol{\lambda}(t)$, define the $2n$ integration constants,

$$\begin{aligned} \boldsymbol{\lambda}^T(t_o) &= \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_o)}, \\ \boldsymbol{\lambda}^T(t_f) &= \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)}. \end{aligned} \quad (2.62)$$

Finally the conditions that define the initial and final times are

$$\begin{aligned} -g(t_o) - \boldsymbol{\lambda}^T(t_o) \cdot \mathbf{f}(t_o) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} &= 0, \\ g(t_f) + \boldsymbol{\lambda}^T(t_f) \cdot \mathbf{f}(t_f) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} &= 0. \end{aligned} \quad (2.63)$$

The conditions (2.59) - (2.63) make up the first-order necessary conditions for a local minimum ([10],[45],[48]). These conditions can be simplified using the definition of the Hamiltonian,

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) = g(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^T(t) \cdot \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (2.64)$$

so that the resulting necessary conditions are

$$\begin{aligned} \dot{\mathbf{x}}^T(t) &= \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}, \\ -\dot{\boldsymbol{\lambda}}^T(t) &= \frac{\partial \mathcal{H}}{\partial \mathbf{x}}, \\ \mathbf{0} &= \frac{\partial \mathcal{H}}{\partial \mathbf{u}}, \\ -\mathcal{H}(t_o) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} &= 0, \\ \mathcal{H}(t_f) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} &= 0, \end{aligned} \quad (2.65)$$

along with the boundary conditions on the state and costate (2.61 - 2.62).

2.4.3 Pontryagin's Maximum Principle

For some problems, the control can not be determined with the equation (2.60). For these problems, Pontryagin's maximum principle must be used to determine the control.

Consider a general nonlinear optimal control problem (2.51)-(2.53) with a general control constraint,

$$\mathbf{m}(\mathbf{u}(t), t) \leq \mathbf{0}, \quad (2.66)$$

where $\mathbf{m} : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^d$. This function defines the region of feasible control, $\mathbf{u}(t) \in \mathbb{U}$. The control \mathbf{u}^* that gives a local minimum value of the cost J is by definition [45],

$$J(\mathbf{u}) - J(\mathbf{u}^*) = \Delta J(\mathbf{u}^*, \mathbf{u}) \geq 0, \quad (2.67)$$

for all admissible control \mathbf{u} sufficiently close to \mathbf{u}^* . If \mathbf{u} is defined as $\mathbf{u} = \mathbf{u}^* + \delta\mathbf{u}$ then the change in the cost can be expressed as

$$\Delta J(\mathbf{u}^*, \delta\mathbf{u}) = \delta J(\mathbf{u}^*, \delta\mathbf{u}) + \text{higher order terms}. \quad (2.68)$$

If $\delta\mathbf{u}$ is sufficiently small then the higher order terms approach zero and the cost has a local minimum if

$$\delta J(\mathbf{u}^*, \delta\mathbf{u}) \geq 0. \quad (2.69)$$

By using the definition of the Hamiltonian, \mathcal{H} (2.64), the first-order variation of

the augmented cost is (assuming fixed initial/final time)

$$\begin{aligned}
\delta J_a(\mathbf{u}^*, \delta \mathbf{u}) &= \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{x}_f} - \boldsymbol{\lambda}^T(t_f) \right) \cdot \delta \mathbf{x}_f - \delta \boldsymbol{\nu}^T \cdot \boldsymbol{\phi} \\
&+ \left(\boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{x}(t_o)} + \boldsymbol{\lambda}^T(t_o) \right) \cdot \delta \mathbf{x}_o \\
&\int_{t_o}^{t_f} \left[\left(\dot{\boldsymbol{\lambda}}^T(t) + \frac{\partial \mathcal{H}}{\partial \mathbf{x}}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) \right) \cdot \delta \mathbf{x}(t) \right. \\
&+ \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) - \dot{\mathbf{x}}(t) \right) \cdot \delta \boldsymbol{\lambda}(t) \\
&\left. + \left(\frac{\partial \mathcal{H}}{\partial \mathbf{u}}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) \right) \cdot \delta \mathbf{u}(t) \right] dt. \tag{2.70}
\end{aligned}$$

At the optimal solution, \mathbf{x}^* , \mathbf{u}^* , $\boldsymbol{\lambda}^*$, $\boldsymbol{\nu}^*$, the differential equations along with the boundary conditions are satisfied, therefore all the coefficients of the variation terms are zero except the control term. This leaves the variation of the cost as

$$\delta J_a(\mathbf{u}^*, \delta \mathbf{u}) = \int_{t_o}^{t_f} \left[\frac{\partial \mathcal{H}}{\partial \mathbf{u}}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \cdot \delta \mathbf{u}(t) \right] dt. \tag{2.71}$$

The variation of the cost is the integral of the first-order approximation to the change in the Hamiltonian caused by a change in the control alone. The first-order approximation of the change in the Hamiltonian is by definition

$$\begin{aligned}
\frac{\partial \mathcal{H}}{\partial \mathbf{u}}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \cdot \delta \mathbf{u}(t) &\doteq \\
\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t) + \delta \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t) - \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t). \tag{2.72}
\end{aligned}$$

The variation of the cost for all admissible and sufficiently small $\delta \mathbf{u}$ becomes

$$\delta J_a(\mathbf{u}^*, \delta \mathbf{u}) = \int_{t_o}^{t_f} [\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t) + \delta \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t) - \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t)] dt. \tag{2.73}$$

In order for $\delta J_a(\mathbf{u}^*, \delta \mathbf{u})$ to be non-negative for any admissible variation in the control, the Hamiltonian must be greater than the optimal Hamiltonian for all time, so that

$$\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t) + \delta \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t) \geq \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t). \tag{2.74}$$

Therefore, the optimal control is the admissible control that minimizes the Hamiltonian. Pontryagin's maximum principle is stated as

$$\mathbf{u}^*(t) = \arg \min_{\mathbf{u} \in \mathbf{U}} [\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t)] . \quad (2.75)$$

Note that in Pontryagin's original work, a different sign was used in the Hamiltonian. This difference resulted in the optimal control being defined as the control that maximized the Hamiltonian.

In this chapter the most important concepts for the understand of the proposed pseudospectral method has been introduced. The most important of these are polynomial interpolation and numerical integration. Also included was a brief outline of the most common types of spectral methods for the solution of differential equations. Finally the derivation of the first-order optimality conditions for both static optimization problems as well as optimal control problems was included, along with the derivation of Pontryagin's maximum principle.

Chapter 3

Direct Transcription Methods

A common numerical approach to solving optimal control problems, falls into a class of direct methods. These methods rely on a discretization scheme to transcribe the continuous optimal control problem into a discrete nonlinear programming problem (NLP). The resulting NLP can then be solved by one of the many well developed nonlinear optimization algorithms.

In this chapter, two finite difference based methods, Euler and Runge-Kutta [7], are explored. The final method considered in this chapter is the Legendre pseudospectral method [19]. This method has many advantages over the finite difference methods, the most important being the convergence rate. However, the Legendre pseudospectral method suffers from a defect in the optimality conditions at the boundary points. Correcting this defect is the primary motivation for this thesis.

3.1 Euler Transcription

Some direct methods are based on methods for solving initial value problems [7]. The methods are extended to optimal control problems by including an approximation of the cost. The independent variable of time is discretized at a set of nodes or intervals, so that

$$t_o = t_1 < \dots < t_N = t_f . \tag{3.1}$$

Note that the discretization nodes may or may not be at equal intervals.

A direct transcription method can be formulated based on the simple Euler method for solving initial value problems ([12] [41] [63]). Euler's method is derived from Taylor's theorem. Defining $h_i = t_{i+1} - t_i$, $i = 1, \dots, N - 1$, as the distance between two nodes, the state $x(t) \in \mathbb{R}^n$ can be expanded,

$$x(t_{i+1}) = x(t_i + h_i) = x(t_i) + h_i \cdot \frac{dx}{dt}(t_i) + \frac{h_i^2}{2} \cdot \frac{d^2x}{dt^2}(\zeta_i), \quad (3.2)$$

for some $\zeta_i \in [t_i, t_{i+1}]$. Since the state satisfies the differential equation (2.52), the approximation made is

$$\frac{x(t_{i+1}) - x(t_i)}{h_i} = f(x(t_i), u(t_i), t_i), \quad (3.3)$$

for sufficiently small h_i . The cost (2.51) can be approximated by noting that the cost is

$$J = \Phi(x(t_f), t_f) + z(t_f), \quad (3.4)$$

where $z(t)$ satisfies the differential equation,

$$\frac{dz}{dt}(t) = g(x(t), u(t), t). \quad (3.5)$$

The cost term $z(t_f)$, is approximated in the same way as the state, resulting in the cost of

$$J = \Phi(x(t_N), t_N) + \sum_{k=1}^{N-1} g(x(t_k), u(t_k), t_k) \cdot h_k, \quad (3.6)$$

where $t_f = t_N$ is the final time.

The objective of the NLP is to find the state variables $x(t_i) = x_i \in \mathbb{R}^n$, $i = 1, \dots, N$ and control variables $u(t_k) = u_k \in \mathbb{R}^m$, $k = 1, \dots, N - 1$ that minimize the cost (3.6), subject to the set of dynamic equation approximations (3.3), $i = 1, \dots, N - 1$, and boundary conditions

$$\phi(x_1, t_1, x_N, t_N) = 0. \quad (3.7)$$

This method is relatively simple to understand and use, but it generally does not have very good accuracy. The accuracy is improved by using a better approximation to the differential equations, such as the Runge-Kutta method.

3.2 Runge-Kutta Transcription

An improvement over the Euler method can be made by using a better discretization scheme. The method outlined here is the classic Runge-Kutta method of order four ([12] [41] [63]). The differential equation (2.52) is approximated using the relations,

$$x(t_{i+1}) = x(t_i) + \frac{1}{6}(s_{1k} + 2 \cdot s_{2k} + 2 \cdot s_{3k} + s_{4k}), \quad (3.8)$$

where the stages are defined as

$$\begin{aligned} s_{1i} &= h_i \cdot f(x(t_i), u(t_i), t_i), \\ s_{2i} &= h_i \cdot f\left(x(t_i) + \frac{1}{2}s_{1i}, u\left(t_i + \frac{h_i}{2}\right), t_i + \frac{h_i}{2}\right), \\ s_{3i} &= h_i \cdot f\left(x(t_i) + \frac{1}{2}s_{2i}, u\left(t_i + \frac{h_i}{2}\right), t_i + \frac{h_i}{2}\right), \\ s_{4i} &= h_i \cdot f\left(x(t_i) + s_{3i}, u(t_{i+1}), t_{i+1}\right). \end{aligned} \quad (3.9)$$

Note that additional variables must be added to the NLP to approximate the control at the midpoints of each interval, $\tilde{u}_{i+1} = u\left(t_i + \frac{h_i}{2}\right)$, $i = 1, \dots, N-1$, [7]. The cost is approximated in the same way as (3.4), so that the Runge-Kutta approximation to the cost is

$$J = \Phi(x(t_N), t_N) + \sum_{k=1}^{N-1} \frac{1}{6} (\bar{s}_{1k} + 2 \cdot \bar{s}_{2k} + 2 \cdot \bar{s}_{3k} + \bar{s}_{4k}), \quad (3.10)$$

with the stages defined as

$$\begin{aligned}
\bar{s}_{1k} &= h_k \cdot g(x(t_k), u(t_k), t_k) , \\
\bar{s}_{2k} &= h_k \cdot g \left(x(t_k) + \frac{1}{2} s_{1k}, \tilde{u}_{k+1}, t_k + \frac{h_k}{2} \right) , \\
\bar{s}_{3k} &= h_k \cdot g \left(x(t_k) + \frac{1}{2} s_{2k}, \tilde{u}_{k+1}, t_k + \frac{h_k}{2} \right) , \\
\bar{s}_{4k} &= h_k \cdot g \left(x(t_k) + s_{3k}, u(t_{k+1}), t_{k+1} \right) .
\end{aligned} \tag{3.11}$$

Note that the stages used in the approximation to the cost integral, require the same stages used to satisfy the differential equation. This fact is a result of satisfying the differential equations for the state and cost $z(t)$, as a coupled differential equation.

The objective of the NLP is then to find the variables for the state, $x_i = x(t_i) \in \mathbb{R}^n$, $i = 1, \dots, N$, and variables for the control $u_i = u(t_i) \in \mathbb{R}^m$, $i = 1, \dots, N$, and controls at intermediate points, $\tilde{u}_{k+1} = u(t_k + \frac{h_k}{2}) \in \mathbb{R}^m$, $k = 1, \dots, N - 1$, that minimize the cost (3.10) subject to the constraints (3.8), $i = 1, \dots, N - 1$ and boundary conditions

$$\phi(x_1, t_1, x_N, t_N) = 0 . \tag{3.12}$$

While the Runge-Kutta method is an improvement over the Euler method in accuracy, it is significantly more complicated to implement. An improvement over both these finite difference methods is made by applying the ideas of spectral methods to optimal control problems.

3.3 Legendre Pseudospectral Method

The Legendre pseudospectral method ([19] [22] [55]), is a direct transcription method that converts a continuous optimal control problem into a discrete nonlinear programming problem. The resulting NLP can be solved by many well developed computational algorithms. The method uses a set of Lagrange-Gauss-Lobatto (LGL) points for collocation of the differential dynamic constraints of the optimal control problem.

3.3.1 Problem Formulation

The LGL points lie on the interval from -1 to 1 , so the first step in the Legendre pseudospectral transcription is to change the time interval of the optimal control problem from $t \in [t_o, t_f]$ to $\tau \in [-1, 1]$. This is done using the mapping

$$t = \frac{(t_f - t_o)}{2} \tau + \frac{(t_f + t_o)}{2} . \quad (3.13)$$

The mapping is used to replace the optimal control problem (2.51) - (2.53), with the problem of minimizing the cost

$$J = \Phi(\mathbf{x}(1), t_f) + \frac{(t_f - t_o)}{2} \int_{-1}^1 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau , \quad (3.14)$$

subject to the dynamic constraints

$$\frac{2}{(t_f - t_o)} \cdot \frac{d\mathbf{x}}{d\tau} = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) , \quad (3.15)$$

and boundary conditions

$$\phi(\mathbf{x}(-1), t_o, \mathbf{x}(1), t_f) = \mathbf{0} . \quad (3.16)$$

To discretize the dynamic constraints, the states and controls are approximated using a set of Lagrange interpolating polynomials at the N LGL points, so that

$$\begin{aligned} \mathbf{x}(t) &\approx \mathbf{X}(t) = \sum_{i=1}^N \mathbf{x}(t_i) \cdot L_i(t) , \\ \mathbf{u}(t) &\approx \mathbf{U}(t) = \sum_{i=1}^N \mathbf{u}(t_i) \cdot L_i(t) , \end{aligned} \quad (3.17)$$

where $t_i, i = 1, \dots, N$, are the LGL points and $L_i(t)$ are the Lagrange polynomials of degree $N - 1$. From a property of the Lagrange polynomials, $L_i(t_k) = \delta_{ki}$ (Kronecker

delta), it follows that

$$\begin{aligned}\mathbf{X}(t_i) &= \mathbf{x}(t_i), \\ \mathbf{U}(t_i) &= \mathbf{u}(t_i).\end{aligned}\tag{3.18}$$

The derivative of the state is approximated as the exact derivative of the interpolating polynomial. Evaluating the derivative at the LGL points results in

$$\frac{d\mathbf{x}}{dt}(t_k) \approx \frac{d\mathbf{X}}{dt}(t_k) = \sum_{i=1}^N \mathbf{x}(t_i) \cdot D_{ki}, \quad k = 1, \dots, N.\tag{3.19}$$

This defines the derivative matrix, D , as the derivative of the Lagrange polynomials at the LGL points, so that

$$D_{ki} = \frac{dL_i}{dt}(t_k).\tag{3.20}$$

The derivative matrix allows the dynamic equations to be collocated at the LGL points. The resulting algebraic constraints are

$$\frac{2}{(t_f - t_o)} \sum_{i=1}^N D_{ki} \cdot X_i = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k), \quad k = 1, \dots, N.\tag{3.21}$$

The boundary constraints are enforced using the boundary points of the approximating polynomial for the state, X_1, X_N , so that

$$\phi(\mathbf{X}_1, t_o, \mathbf{X}_N, t_f) = 0.\tag{3.22}$$

The integration in the cost function is discretized using the Gauss-Lobatto quadrature rule, so that

$$J = \Phi(\mathbf{X}_N, t_f) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N g(\mathbf{X}_k, \mathbf{U}_k, t_k) \cdot w_k,\tag{3.23}$$

where w_k are the LGL weights.

The continuous optimal control problem is discretized to a NLP where the objective is to find the variables $\mathbf{X}_k \in \mathbb{R}^n$, $\mathbf{U}_k \in \mathbb{R}^m$, $k = 1, \dots, N$, and $t_o, t_f \in \mathbb{R}$ that minimize the cost (3.23), subject to the constraints (3.21) and (3.22). The solution

to the NLP is defined by the resulting KKT conditions.

3.3.2 KKT Conditions

Analysis of the Kurush-Kuhn-Tucker (KKT) conditions is aided by several properties of the derivative matrix, which can be derived by looking at the integration by parts formula for two functions $p(t), q(t) \in C^1[-1, 1]$. The formula is

$$\int_{-1}^1 \dot{p}(t) \cdot q(t) dt = p(t) \cdot q(t) \Big|_{-1}^1 - \int_{-1}^1 p(t) \cdot \dot{q}(t) dt . \quad (3.24)$$

If $p(t), q(t)$ are polynomials of degree $N - 1$, the product $\dot{p}(t) \cdot q(t)$ and $p(t) \cdot \dot{q}(t)$ are polynomials of degree $2 \cdot N - 3$. The integrals in (3.24) are replaced exactly by a Gauss-Lobatto quadrature using N LGL points. Using the derivative matrix to find the exact derivative of the polynomials at the LGL points results in

$$\sum_{k=1}^N \sum_{i=1}^N D_{ki} \cdot p(t_i) \cdot q(t_k) \cdot w_k = p(t) \cdot q(t) \Big|_{-1}^1 - \sum_{k=1}^N p(t_k) \cdot \sum_{i=1}^N D_{ki} \cdot q(t_i) \cdot w_k . \quad (3.25)$$

Since this must be true for all polynomials of degree $N - 1$, it must be true for the sets of Lagrange interpolation polynomials at the LGL points,

$$p(t) = L_l(t), \quad q(t) = L_j(t), \quad l, j = 1, \dots, N . \quad (3.26)$$

Because these polynomials have the property $L_l(t_j) = \delta_{lj}$, this implies

$$D_{jl} \cdot w_j = D_{lj} \cdot w_l, \quad (l, j) \neq (1, 1), (N, N) . \quad (3.27)$$

Also since $w_k \neq 0$, it also implies

$$\begin{aligned} D_{11} \cdot w_1 &= -\frac{1}{2}, \\ D_{NN} \cdot w_N &= \frac{1}{2}. \end{aligned} \quad (3.28)$$

The derivation of the KKT conditions is simplified by using the relations (3.27) and (3.28).

The solution to the NLP derived in Section 3.3.1 can be found using the KKT conditions. The KKT conditions are determined from the augmented cost function or Lagrangian, so that

$$\begin{aligned}
J_a = & \Phi(\mathbf{X}_N, t_f) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N g_k \cdot w_k - \boldsymbol{\nu}^T \cdot \phi(\mathbf{X}_1, t_o, \mathbf{X}_N, t_f) \\
& - \sum_{k=1}^N \tilde{\boldsymbol{\lambda}}_k \cdot \left(\frac{2}{(t_f - t_o)} \sum_{i=1}^N D_{ki} \cdot \mathbf{X}_i - \mathbf{f}_k \right), \tag{3.29}
\end{aligned}$$

where $g_k = g(\mathbf{X}_k, \mathbf{U}_k, t_k)$ and $\mathbf{f}_k = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k)$. The KKT multipliers, $\tilde{\boldsymbol{\lambda}}_k \in \mathbb{R}^n$, are associated with the approximation to the dynamic constraints (3.21), and the Lagrange multipliers, $\boldsymbol{\nu} \in \mathbb{R}^q$, are associated with the boundary constraints (3.22). The KKT conditions result from taking the partial derivatives of the Lagrangian with respect to the control and state variables and setting equal to zero, so that

$$\frac{\partial J_a}{\partial \mathbf{X}_k} = 0, \quad \frac{\partial J_a}{\partial \mathbf{U}_k} = 0, \quad k = 1, \dots, N. \tag{3.30}$$

Taking the partial derivatives of the Lagrangian with respect to the interior state variables \mathbf{X}_k , $k = 2, \dots, N - 1$, results in

$$\frac{\partial J_a}{\partial \mathbf{X}_k} = \frac{(t_f - t_o)}{2} \frac{\partial g_k}{\partial \mathbf{X}_k} w_k + \tilde{\boldsymbol{\lambda}}_k \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} - \frac{2}{(t_f - t_o)} \sum_{i=1}^N \tilde{\boldsymbol{\lambda}}_i \cdot D_{ik} = 0, \tag{3.31}$$

where $\frac{\partial g_k}{\partial \mathbf{X}_k} = \frac{\partial g}{\partial \mathbf{x}}(\mathbf{X}_k, \mathbf{U}_k, t_k)$ and $\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{X}_k, \mathbf{U}_k, t_k)$. Multiplying through by $\frac{2}{(t_f - t_o) \cdot w_k}$ results in

$$\frac{\partial g_k}{\partial \mathbf{X}_k} + \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\boldsymbol{\lambda}}_k}{w_k} \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} - \frac{2}{(t_f - t_o)} \sum_{i=1}^N \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\boldsymbol{\lambda}}_i}{w_k} \cdot D_{ik} = 0. \tag{3.32}$$

Using the property $w_i \cdot D_{ik} = -w_k \cdot D_{ki}$ on the interior nodes $k \neq 1, N$ (3.27), the

equation is simplified to

$$\frac{\partial g_k}{\partial \mathbf{X}_k} + \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_k}{w_k} \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{(t_f - t_o)} \sum_{i=1}^N \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_i}{w_i} \cdot D_{ki} = 0. \quad (3.33)$$

This result leads to the costate mapping principle, [22], between the KKT multipliers $\tilde{\lambda}_k$ and the estimate for the costate $\lambda_k \in \mathbb{R}^n$. This principle is

$$\lambda_k = \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_k}{w_k}. \quad (3.34)$$

Using the costate mapping principle, the KKT condition (3.31), is equivalent to the discretized form of the continuous first-order necessary conditions for the costate dynamics (2.59), at the interior LGL points. These resulting conditions are

$$\frac{d\lambda}{dt}(t_k) \approx \frac{2}{(t_f - t_o)} \sum_{i=1}^N D_{ki} \cdot \lambda_i = -\frac{\partial g_k}{\partial \mathbf{X}_k} - \lambda_k \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k}, \quad k = 2, \dots, N-1. \quad (3.35)$$

This relation however, is not true for the boundary points.

The KKT conditions from the two boundary points are explored by taking the derivative of the Lagrangian (3.29) with respect to the boundary states, $\mathbf{X}_1, \mathbf{X}_N$. The result is

$$\frac{\partial J_a}{\partial \mathbf{X}_1} = \frac{(t_f - t_o)}{2} \cdot \frac{\partial g_1}{\partial \mathbf{X}_1} \cdot w_1 - \nu^T \cdot \frac{\partial \phi}{\partial \mathbf{X}_1} - \frac{2}{(t_f - t_o)} \sum_{k=1}^N \tilde{\lambda}_k \cdot D_{k1} + \tilde{\lambda}_1 \cdot \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} = 0. \quad (3.36)$$

Using the property that $D_{k1} = -(w_1/w_k) \cdot D_{1k}$, $k = 2, \dots, N$ (3.27) and adding $2 \cdot \frac{2}{(t_f - t_o)} D_{11} \cdot \tilde{\lambda}_1$ to both sides of the equation results in

$$\frac{(t_f - t_o)}{2} \cdot \frac{\partial g_1}{\partial \mathbf{X}_1} w_1 + \frac{2}{(t_f - t_o)} \sum_{k=1}^N w_1 \cdot \frac{\tilde{\lambda}_k}{w_k} \cdot D_{1k} + \tilde{\lambda}_1 \cdot \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} = 2 \cdot \frac{2}{(t_f - t_o)} D_{11} \cdot \tilde{\lambda}_1 + \nu^T \cdot \frac{\partial \phi}{\partial \mathbf{X}_1}. \quad (3.37)$$

Multiplying through by $\frac{2}{(t_f - t_o) \cdot w_1}$ and using the property $2 \cdot D_{11} = -1/w_1$ (3.28),

results in

$$\begin{aligned} \frac{\partial g_1}{\partial \mathbf{X}_1} + \frac{2}{(t_f - t_o)} \sum_{k=1}^N \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_k}{w_k} \cdot D_{1k} + \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_1}{w_1} \cdot \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} \\ = \frac{2}{(t_f - t_o) \cdot w_1} \left(-\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_1}{w_1} + \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}_1} \right). \end{aligned} \quad (3.38)$$

Finally using the costate mapping principle (3.34), the equation is simplified to

$$\frac{\partial g_1}{\partial \mathbf{X}_1} + \frac{2}{(t_f - t_o)} \sum_{k=1}^N \lambda_k \cdot D_{1k} + \lambda_1 \cdot \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} = \frac{2}{(t_f - t_o) \cdot w_1} \left(-\lambda_1 + \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}_1} \right). \quad (3.39)$$

The left-hand side of the equation is the discretized form of the continuous costate dynamics at the initial LGL point (t_1). The right-hand side of the equation is the discrete form the costate boundary condition (2.62). This KKT condition (3.39) shows that at the initial LGL point, the costate estimate λ_1 , does not satisfy the boundary condition or the differential equation, but a linear combination of the two.

The KKT condition derived from the final state can be used with the properties $D_{kN} = -(w_N/w_k) \cdot D_{1N}$, $k = 1, \dots, N-1$ (3.27) and $2 \cdot D_{NN} = 1/w_N$ (3.28) to show that

$$\begin{aligned} \frac{\partial g_N}{\partial \mathbf{X}_N} + \frac{2}{(t_f - t_o)} \sum_{k=1}^N \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_k}{w_k} \cdot D_{Nk} + \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_N}{w_N} \cdot \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_N} \\ = \frac{2}{(t_f - t_o) \cdot w_N} \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\lambda}_N}{w_N} - \frac{\partial \Phi}{\partial \mathbf{X}_N} + \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}_N} \right). \end{aligned} \quad (3.40)$$

Again using the the costate mapping principle (3.34), the equation is simplified to

$$\frac{\partial g_N}{\partial \mathbf{X}_N} + \frac{2}{(t_f - t_o)} \sum_{k=1}^N \lambda_k \cdot D_{Nk} + \lambda_N \cdot \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_N} = \frac{2}{(t_f - t_o) \cdot w_N} \left(\lambda_N - \frac{\partial \Phi}{\partial \mathbf{X}_N} + \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}_N} \right). \quad (3.41)$$

This condition also indicates that the costate estimate at the final LGL point, λ_N , does not satisfy the boundary condition or the differential equation, but a linear combination of the two. The correction of these errors is the primary motivation for

this thesis.

Taking the partial derivatives of the Lagrangian (3.29) with respect to the control variables, \mathbf{U}_k , results in the second set of KKT conditions. These conditions are

$$\frac{\partial J_a}{\partial \mathbf{U}_k} = \frac{(t_f - t_o)}{2} \frac{\partial g_k}{\partial \mathbf{U}_k} w_k + \tilde{\boldsymbol{\lambda}}_k \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} = 0. \quad (3.42)$$

Multiplying both sides of the equation by $\frac{2}{(t_f - t_o) \cdot w_k}$ gives the equation in the desired form, so that

$$\frac{\partial g_k}{\partial \mathbf{U}_k} + \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\boldsymbol{\lambda}}_k}{w_k} \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} = 0 \quad (3.43)$$

This equation is simplified using the costate mapping principle (3.34), resulting in

$$\frac{\partial g_k}{\partial \mathbf{U}_k} + \boldsymbol{\lambda}_k \cdot \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} = 0. \quad (3.44)$$

This result is the discretized form of the continuous necessary condition for the control (2.60).

The KKT conditions for the Legendre pseudospectral method, define the set of conditions that approximate the continuous costate dynamics. The costate can be estimated from the KKT conditions using the costate mapping principle (3.34). The costate estimate however, does not satisfy the discrete form of the costate dynamics at the boundaries. These defects in the costate equations lead to significant errors in the costate estimates, especially at the boundary points. The Legendre pseudospectral method works well on many problems, but has the deficiency that the KKT conditions which define the solution to the transcribed NLP are not the same as the discretized first-order necessary conditions from the continuous optimal control problem. Improving the Legendre pseudospectral method is the most important motivation for this thesis.

[This page intentionally left blank.]

Chapter 4

Integral Gauss Pseudospectral Method

The previous chapter outlined the deficiencies of the Legendre pseudospectral method. To correct these deficiencies, a pseudospectral transcription is proposed, which is based on the integral form of the optimal control problem. It is shown that this Gauss pseudospectral method does not suffer from the deficiencies of the Legendre pseudospectral method and will return more accurate solutions.

The Legendre pseudospectral method transcribes a continuous optimal control problem in differential form by discretizing the problem and collocating at a set of Gauss-Lobatto points. These points have the property that they include the boundaries of the interval between -1 and 1 . To remove these end points and collocate at a set of Gauss points, which are all interior to the interval $(-1, 1)$, the dependence on the boundary points must be removed or expressed in a different form. The transformation is done by reformulating the optimal control problem using an equivalent form of the state dynamics. This equivalent form is found by integrating the differential dynamics and adding in the initial condition.

In this chapter, it is shown that the continuous optimal control problem in integral form satisfies the same optimality conditions as the differential form of the problem. This result allows a pseudospectral transcription to be defined in terms of the integral form of the continuous optimal control problem. The discretization is accomplished

at a set of Gauss points using the integration approximation matrix, which allows the optimal control problem to be transcribed into a nonlinear program (NLP).

It is also shown that the Karush-Kuhn-Tucker (KKT) conditions of the NLP from the integral method are exactly equal to the discretized first-order optimality conditions. This relation indicates that solving the optimal control problem using the integral pseudospectral method is consistent with the continuous optimal control problem, and the direct and indirect formulations are the same. This result is not true of other pseudospectral methods. This consistency allows the Gauss pseudospectral method to return more accurate solutions than other methods.

The Gauss pseudospectral method has also been shown to be consistent for a large class of problems, including free time, multiple phase, path constraints, and those requiring Pontryagin's maximum principle.

4.1 Analytical Necessary Conditions

Consider the general optimal control problem, (2.51) - (2.53), but with the differential dynamic constraints defined as integrals, so that

$$\mathbf{x}(t) = \mathbf{x}(t_o) + \int_{t_o}^t \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau, \quad (4.1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state, $\mathbf{u}(t) \in \mathbb{R}^m$ is the control and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$ is the right hand side of the differential equation describing the dynamics.

The cost function remains the same as the differential form, and is stated most generally in Bolza form as the sum of a terminal cost, $\Phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$, and an integrated cost, $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}$. The cost is then

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_{t_o}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt. \quad (4.2)$$

The boundary conditions are stated in their most general form as

$$\phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) = 0, \quad (4.3)$$

where $\phi : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^q$. The final state, $\mathbf{x}(t_f)$, occurring in the expressions for the final cost, Φ , and in the boundary conditions, ϕ , is defined in terms of the states and controls at the interior points of the problem, so that

$$\mathbf{x}(t_f) = \mathbf{x}(t_o) + \int_{t_o}^{t_f} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) dt. \quad (4.4)$$

This relation is necessary to allow for the discretization at Gauss points (see Section 4.2.3). Both the initial time, t_o , and the final time, t_f , can be free. The optimal control problem in this form will be referred to as the integral Bolza problem.

The solution to the integral Bolza problem is the same as the solution to the differential form of the problem. This equivalence can be seen by looking at the first-order necessary conditions for optimality.

Theorem 4.1.1. *The first-order optimality conditions for the integral Bolza problem are equivalent to the necessary conditions for the Bolza problem in differential form (2.59). The costates from the differential and integral forms can be related by*

$$\boldsymbol{\lambda}^T(t) = \int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)}. \quad (4.5)$$

Proof. The necessary conditions for the integral Bolza problem are derived using the calculus of variations on the augmented cost function or Lagrangian. The augmented cost function is created by adjoining the dynamic constraints (4.1) with costates, $\mathbf{p}(t) \in \mathbb{R}^n$, and the boundary conditions (4.3) by Lagrange multipliers, $\boldsymbol{\nu} \in \mathbb{R}^q$. The augmented cost is then

$$\begin{aligned} J_a = & \Phi(\mathbf{x}(t_f), t_f) - \boldsymbol{\nu}^T \cdot \phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) \\ & + \int_{t_o}^{t_f} \left[g(\mathbf{x}, \mathbf{u}, t) - \mathbf{p}^T \cdot \left(\mathbf{x} - \mathbf{x}(t_o) - \int_{t_o}^t \mathbf{f}(\mathbf{x}, \mathbf{u}, \tau) d\tau \right) \right] dt. \end{aligned} \quad (4.6)$$

Note that the adjoints \mathbf{p} are not the same as the costates $\boldsymbol{\lambda}$ from the differential formulation (see Section 2.4). According to Lagrange theory, the stationary point of the constrained cost J (4.2), is equal to the stationary point of the unconstrained augmented cost J_a (4.6).

Before applying the calculus of variations, a more convenient form of the augmented cost is found. Taking advantage of a property of the double integral allows for the independent variables in the integrands to be switched. For example, in the double integral

$$\int_{t_o}^{t_f} \left(f(t) \int_{t_o}^t g(\tau) d\tau \right) dt = \int_{t_o}^{t_f} \left(\int_t^{t_f} f(\tau) d\tau \right) g(t) dt, \quad (4.7)$$

the independent variables of integration, t and τ , are switched between the left and right sides of the equation. Applying the integration substitution (4.7) to the augmented cost (4.6) allows the independent variables, t and τ , in the double integral to be switched resulting in

$$J_a = \Phi(\mathbf{x}(t_f), t_f) - \nu^T \cdot \phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) + \int_{t_o}^{t_f} \left[g(\mathbf{x}, \mathbf{u}, t) - \mathbf{p}^T \cdot (\mathbf{x} - \mathbf{x}(t_o)) + \int_t^{t_f} \mathbf{p}^T d\tau \cdot \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \right] dt. \quad (4.8)$$

Now the first-order variation is taken with respect to all free variables. Note that when taking the variation of the cost with respect to the final time the form of (4.6) is used while the variation of the cost with respect to the initial time the form of (4.8) is used. These forms are used because the integrand in (4.6) does not depend on the final time and the integrand in (4.8) does not depend on the initial time. This procedure is valid because both forms of the cost, (4.6) and (4.8), are equivalent. The

first-order variation of the augmented cost becomes

$$\begin{aligned}
\delta J_a &= \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \delta \mathbf{x}_f + \frac{\partial \Phi}{\partial t_f} \delta t_f - \delta \nu^T \phi - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_o)} \delta \mathbf{x}_o \\
&\quad - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} \delta t_o - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \delta \mathbf{x}_f - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} \delta t_f \\
&\quad + \left(g - \mathbf{p}^T \cdot \left(\mathbf{x} - \mathbf{x}(t_o) - \int_{t_o}^t \mathbf{f} d\tau \right) \right) \Big|_{t=t_f} \delta t_f \\
&\quad - \left(g - \mathbf{p}^T \cdot (\mathbf{x} - \mathbf{x}(t_o)) + \int_t^{t_f} \mathbf{p}^T d\tau \cdot \mathbf{f} \right) \Big|_{t=t_o} \delta t_o \\
&\quad + \int_{t_o}^{t_f} \left[\frac{\partial g}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial g}{\partial \mathbf{u}} \delta \mathbf{u} - \delta \mathbf{p}^T \cdot \left(\mathbf{x} - \mathbf{x}(t_o) - \int_{t_o}^t \mathbf{f} \cdot d\tau \right) - \mathbf{p}^T \delta \mathbf{x} \right. \\
&\quad \left. + \int_t^{t_f} \mathbf{p}^T d\tau \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} + \int_t^{t_f} \mathbf{p}^T d\tau \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} + \mathbf{p}^T \delta \mathbf{x}_o \right] dt.
\end{aligned} \tag{4.9}$$

There are many things that can be done to simplify the expression. First, the variation of the final states is expressed in terms of the variations of the initial states and intermediate points. This expression is used because the variation of the final state is not independent, but depends on the variations of the other independent variables based on the relation between the final state and the interior states (4.4). The first-order variation of the final state is

$$\delta \mathbf{x}_f = \delta \mathbf{x}_o + \int_{t_o}^{t_f} \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} \right] dt + \mathbf{f}(t_f) \delta t_f - \mathbf{f}(t_o) \delta t_o. \tag{4.10}$$

The final state variation (4.10) is substituted into the variation of the augmented cost (4.9) and like terms can be grouped. Note that the terms

$$\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \quad \text{and} \quad \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)}, \tag{4.11}$$

do not depend on time, and therefore can be moved inside the integral. Also in the final term of the integrand in (4.9), the variation of the initial state $\delta \mathbf{x}_o$, does not depend on time and can be moved outside the integral. The simplified variation is

then

$$\begin{aligned}
\delta J_a &= -\delta \boldsymbol{\nu}^T \boldsymbol{\phi} + \left(-\boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_o)} + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} + \int_{t_o}^{t_f} \mathbf{p}^T dt \right) \delta \mathbf{x}_o \\
&+ \left(g - \mathbf{p}^T \cdot \left(\mathbf{x} - \mathbf{x}(t_o) - \int_{t_o}^t \mathbf{f} d\tau \right) \right) \Big|_{t=t_f} \delta t_f \\
&+ \left(\frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial t_f} + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \mathbf{f}(t_f) - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \mathbf{f}(t_f) \right) \delta t_f \\
&+ \left(-g + \mathbf{p}^T \cdot (\mathbf{x} - \mathbf{x}(t_o)) - \int_{t_o}^{t_f} \mathbf{p}^T d\tau \cdot \mathbf{f} \right) \Big|_{t=t_o} \delta t_o \\
&+ \left(-\boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial t_o} - \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \mathbf{f}(t_o) + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \mathbf{f}(t_o) \right) \delta t_o \\
&+ \int_{t_o}^{t_f} \left[\left(\frac{\partial g}{\partial \mathbf{x}} - \mathbf{p}^T + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \delta \mathbf{x} \right. \\
&\quad \left. - \delta \mathbf{p}^T \cdot \left(\mathbf{x} - \mathbf{x}(t_o) - \int_{t_o}^t \mathbf{f} d\tau \right) \right. \\
&\quad \left. + \left(\frac{\partial g}{\partial \mathbf{u}} \delta \mathbf{u} + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} \right] dt .
\end{aligned} \tag{4.12}$$

The necessary conditions for stationary point are defined as the conditions so that for any arbitrary variation of the independent variables the variation of the cost is zero. The necessary conditions for the optimal solution are therefore found by setting the coefficients of the independent increments, $\delta \boldsymbol{\nu}$, $\delta \mathbf{x}_o$, δt_o , δt_f , $\delta \mathbf{x}$, $\delta \mathbf{p}$, and $\delta \mathbf{u}$, equal to zero. The resulting necessary conditions are

$$\begin{aligned}
\mathbf{x}(t) &= \mathbf{x}(t_o) + \int_{t_o}^t \mathbf{f}(\mathbf{x}, \mathbf{u}, \tau) d\tau , \\
\mathbf{p}^T &= \frac{\partial g}{\partial \mathbf{x}} + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}} , \\
0 &= \frac{\partial g}{\partial \mathbf{u}} + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} , \\
\phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) &= 0 , \\
\int_{t_o}^{t_f} \mathbf{p}^T dt + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} &= \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_o)} .
\end{aligned} \tag{4.13}$$

The condition for the terminal time

$$g(t_f) - \mathbf{p}^T(t_f) \left(\mathbf{x}(t_f) - \mathbf{x}(t_o) - \int_{t_o}^{t_f} \mathbf{f} dt \right) + \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \mathbf{f}(t_f) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} = 0, \quad (4.14)$$

is reduced by setting the term $\mathbf{x}(t_f) - \mathbf{x}(t_o) - \int_{t_o}^{t_f} \mathbf{f} \cdot dt$ equal to zero. This term must be zero to satisfy the definition of the final state relation (4.4). The terminal time condition is then

$$g(t_f) + \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \mathbf{f}(t_f) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} = 0. \quad (4.15)$$

The condition for the initial time

$$-g(t_o) + \mathbf{p}^T(t_o) (\mathbf{x}(t_o) - \mathbf{x}(t_o)) - \int_{t_o}^{t_f} \mathbf{p}^T dt \cdot \mathbf{f}(t_o) - \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \mathbf{f}(t_o) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} = 0, \quad (4.16)$$

is simplified to

$$-g(t_o) - \left(\int_{t_o}^{t_f} \mathbf{p}^T dt + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \mathbf{f}(t_o) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} = 0. \quad (4.17)$$

If there exists a solution to the necessary conditions for the differential problem (2.59), then the conditions (4.13), (4.15), and (4.17) define an equivalent solution if the costate, $\boldsymbol{\lambda}$, is defined as

$$\boldsymbol{\lambda}^T(t) = \int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)}. \quad (4.18)$$

The relation (4.18), indicates that the differential equation for the costate, $\boldsymbol{\lambda}(t)$, (second equation in (2.59)), is the same as the integral equation involving $\mathbf{p}(t)$, (second equation in (4.13)) because of the derivative of (4.18) is

$$\dot{\boldsymbol{\lambda}}^T(t) = -\mathbf{p}^T(t). \quad (4.19)$$

The relation (4.18) also defines the same boundary conditions of the costate, $\lambda(t)$. The boundary at the final time comes from evaluating the relation (4.18) at the final time. The boundary condition at the initial time comes from evaluating the relation (4.18) at the initial time, and substituting into the last necessary condition in (4.13). Therefore, the boundary conditions

$$\begin{aligned}\lambda(t_f)^T &= \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \nu^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)}, \\ \lambda(t_o)^T &= \int_{t_o}^{t_f} \mathbf{p}^T dt + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \nu^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} = \nu^T \frac{\partial \phi}{\partial \mathbf{x}(t_o)},\end{aligned}\tag{4.20}$$

are equivalent to (2.62). Using the boundary conditions (4.20), it can be seen that the conditions for the initial and final times, (4.15) and (4.17), can be expressed as

$$\begin{aligned}-g(t_o) - \lambda^T(t_o) \cdot \mathbf{f}(t_o) - \nu^T \frac{\partial \phi}{\partial t_o} &= 0, \\ g(t_f) + \lambda^T(t_f) \cdot \mathbf{f}(t_f) - \nu^T \frac{\partial \phi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} &= 0,\end{aligned}\tag{4.21}$$

which are the same as (2.63).

This derivation shows that first-order necessary conditions for the integral Bolza problem (4.1)-(4.3) are equivalent to the necessary conditions for the differential optimal control problem (2.51) - (2.53). It also indicates that the costates from the integral Bolza problem and the differential form of the problem can be related by (4.5) □

Theorem 4.1.1 shows that the necessary conditions are equivalent for the integral and differential forms of the problem. Therefore, any stationary points of the problem in integral form will be exactly the same as the stationary points of the differential problem, and the global minimum for both problems will also be the same.

4.1.1 Pontryagin's Maximum Principle

In Section 4.1, it was assumed that the control, $\mathbf{u}(t) \in \mathbb{R}^m$, was uniquely determined from the algebraic equation

$$\mathbf{0} = \frac{\partial g}{\partial \mathbf{u}}(t) + \boldsymbol{\lambda}^T(t) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(t) = \frac{\partial H}{\partial \mathbf{u}}(t). \quad (4.22)$$

For some problems, such as those with control path constraints, it may not be possible to determine the control from (4.22). For these problems, Pontryagin's maximum principle is applied to determine the control (see Section 2.4.3). It can be shown that the solution to the integral Bolza problem also satisfies Pontryagin's maximum principle.

Consider a general nonlinear optimal control problem with fixed initial and final times. The cost is

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_{t_o}^{t_f} g(\mathbf{x}, \mathbf{u}, t) dt, \quad (4.23)$$

and the dynamics are written in integral form, so that

$$\mathbf{x}(t) = \mathbf{x}(t_o) + \int_{t_o}^t \mathbf{f}(\mathbf{x}, \mathbf{u}, \tau) d\tau. \quad (4.24)$$

The boundary conditions are

$$\boldsymbol{\phi}(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) = \mathbf{0}, \quad (4.25)$$

with a control path constraint

$$\mathbf{u}(t) \in \mathbb{U}, \quad (4.26)$$

where \mathbb{U} defines the set of admissible controls.

Theorem 4.1.2. *The optimal control of the integral Bolza problem satisfies Pontryagin's maximum principle.*

Proof. In order for the control to be optimal, the variation of the cost at the optimal solution for sufficiently small admissible variations in the control, must be non-

negative ([10], [45]), so that

$$\begin{aligned} \delta J(\mathbf{u}^*, \delta \mathbf{u}) &\geq 0, \\ \forall \delta \mathbf{u} \quad \text{s.t.} \quad \mathbf{u}^* + \delta \mathbf{u} &\in \mathbb{U}. \end{aligned} \tag{4.27}$$

The integral Bolza problem can be shown to satisfy Pontryagin's maximum principle, by applying the condition (4.27) to the augmented cost function. The first step is to form the augmented cost function and find the first-order variation. The variation (from Section 4.1) is

$$\begin{aligned} \delta J_a &= \delta \boldsymbol{\nu}^T \cdot \boldsymbol{\phi} + \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_o)} + \int_{t_o}^{t_f} \mathbf{p}^T dt \right) \delta \mathbf{x}(t_o) \\ &+ \int_{t_o}^{t_f} \left[\left(\frac{\partial g}{\partial \mathbf{x}} - \mathbf{p}^T(t) + \left(\int_t^{t_f} \mathbf{p}(\tau)^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \delta \mathbf{x} \right. \\ &\quad - \left(\mathbf{x}(t) - \mathbf{x}(t_o) - \int_{t_o}^t \mathbf{f} d\tau \right) \delta \mathbf{p} \\ &\quad \left. + \left(\frac{\partial g}{\partial \mathbf{u}} + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} \right] dt \end{aligned} \tag{4.28}$$

At the optimal solution, the state and costate necessary conditions (4.13) along with the boundary constraints must be satisfied. Simplifying the variation of the cost at the optimal solution results in

$$\delta J_a = \int_{t_o}^{t_f} \left[\left(\frac{\partial g}{\partial \mathbf{u}} + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} \right] dt. \tag{4.29}$$

The first-order change of the integrand with respect to changes in the control alone can be expressed as

$$\begin{aligned} &\left(\frac{\partial g}{\partial \mathbf{u}} + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} = \\ &g(\mathbf{u}^* + \delta \mathbf{u}) + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \mathbf{f}(\mathbf{u}^* + \delta \mathbf{u}) \\ &- g(\mathbf{u}^*) - \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(t_f)} \right) \cdot \mathbf{f}(\mathbf{u}^*). \end{aligned} \tag{4.30}$$

Note that because of the definition of the costate relation (4.5) and the definition of the Hamiltonian, $\mathcal{H} = g + \lambda \cdot f$, the relation (4.30) is equivalent to (2.72), which is

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}}(\mathbf{u}^*) \cdot \delta \mathbf{u} \doteq \mathcal{H}(\mathbf{u}^* + \delta \mathbf{u}) - \mathcal{H}(\mathbf{u}^*) . \quad (4.31)$$

For the variation of the cost to be non-negative for any admissible control variation, it must be true that

$$\begin{aligned} g(\mathbf{u}^* + \delta \mathbf{u}) + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \cdot \mathbf{f}(\mathbf{u}^* + \delta \mathbf{u}) \geq \\ g(\mathbf{u}^*) + \left(\int_t^{t_f} \mathbf{p}^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \cdot \mathbf{f}(\mathbf{u}^*) . \end{aligned} \quad (4.32)$$

Therefore, the optimal control is the admissible control that minimizes

$$g(\mathbf{x}^*, \mathbf{u}, t) + \left(\int_t^{t_f} (\mathbf{p}^*)^T d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right) \cdot \mathbf{f}(\mathbf{x}^*, \mathbf{u}, t) , \quad (4.33)$$

for all time. This equation is the same condition as (2.74) because of the costate relation (4.5), and the definition of the Hamiltonian.

This derivation demonstrates that the solution to the integral Bolza problem satisfies Pontryagin's maximum principle. \square

4.2 Discrete Transformation

The previous sections demonstrated that the solution to the integral Bolza problem is equivalent to the solution to the continuous problem in differential form. To find an approximate solution to the continuous optimal control problem, the integral Bolza problem is discretized.

To discretize the integral Bolza problem several steps must be taken. The first is to change the time interval from arbitrary bounds $t \in [t_o, t_f]$ to the interval $[-1, 1]$, which is more convenient for the discretization. The next step is to discretize the integral form of the dynamic constraints using an integration approximation matrix, and finally the cost can be approximated using the Gauss quadrature rule. These

steps allow the continuous optimal control problem to be transcribed into a discrete NLP.

4.2.1 Time intervals

The first step in defining the approximate solution of the integral Bolza problem is to map the independent variable of time to a new interval. Both the Gauss quadrature rule and the integration approximation matrix are applied on the time interval $\tau \in [-1, 1]$. General problems on the interval $t \in [t_o, t_f]$ can be scaled to $[-1, 1]$ using the relation

$$t = \frac{(t_f - t_o) \cdot \tau + (t_f + t_o)}{2}. \quad (4.34)$$

This relation allows an integral on a general interval to be rescaled and then approximated by the Gauss quadrature rule, so that

$$\int_{t_o}^{t_f} f(t) dt = \frac{(t_f - t_o)}{2} \int_{-1}^1 f(\tau) d\tau \approx \frac{(t_f - t_o)}{2} \sum_{k=1}^N f(t_k) \cdot w_k. \quad (4.35)$$

The same time scaling can be done for the integral form of the dynamics, so that

$$\int_{t_o}^t f(\hat{\tau}) d\hat{\tau} = \frac{(t_f - t_o)}{2} \int_{-1}^{\tau} f(\hat{\tau}) d\hat{\tau}. \quad (4.36)$$

4.2.2 Integration Approximation Matrix

The discrete transformation of the continuous optimal control problem relies on an approximation of the dynamic constraints. The integral form of the differential dynamics,

$$\mathbf{x}(t) = \mathbf{x}(t_o) + \int_{t_o}^t \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau, \quad (4.37)$$

is approximated using an integration approximation matrix [2]. The integration approximation matrix approximates the integral of a function by using the exact integral of an interpolating polynomial fit to the function.

The integration approximation matrix is derived from the Lagrange interpolating

polynomials at a set of collocation points, $-1 \leq t_1 < \dots < t_N \leq 1$. The associated Lagrange polynomials [63] are

$$L_k(t) \doteq \frac{\omega(t)}{(t - t_k)\omega'(t_k)}, \quad (4.38)$$

where t_1, \dots, t_N are the roots of the polynomial $\omega(t)$, where

$$\omega(t) \doteq \prod_{i=1}^N (t - t_i). \quad (4.39)$$

The Lagrange polynomials satisfy the property at the collocation points

$$L_k(t_i) = \delta_{ki} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}, \quad (4.40)$$

so that the Lagrange interpolation formula approximates a function by a polynomial (see Section 2.1.1),

$$f(t) \approx \sum_{k=1}^N f(t_k) \cdot L_k(t). \quad (4.41)$$

Definition 4.2.1. *The integration approximation matrix, $A \in \mathbb{R}^{N \times N}$, is a matrix which approximates the integral of a function from -1 to t_i , where t_i is the i th collocation point, so that*

$$\int_{-1}^{t_i} f(\tau) d\tau \approx \sum_{k=1}^N A_{ik} \cdot f(t_k). \quad (4.42)$$

The entries in the matrix A are found by integrating the Lagrange interpolation formula (4.41), so that

$$\begin{aligned} \int_{-1}^{t_i} f(\tau) d\tau &\approx \int_{-1}^{t_i} \sum_{k=1}^N f(t_k) \cdot L_k(\tau) d\tau = \sum_{k=1}^N f(t_k) \int_{-1}^{t_i} L_k(\tau) d\tau, \\ A_{ik} &= \int_{-1}^{t_i} L_k(\tau) d\tau. \end{aligned} \quad (4.43)$$

The elements in the matrix A for Gauss points can also be approximated by using

Axelsson's algorithm [2],

$$A_{ik} = \frac{w_i}{2} \left(1 + t_i + \sum_{v=1}^{n-2} P_v(t_k) [P_{v+1}(t_i) - P_{v-1}(t_i)] + P_{N-1}(t_k) [P_N(t_i) - P_{N-2}(t_i)] \right), \quad (4.44)$$

where w_i is the i th Gauss weight, and P_j is the j th Legendre polynomial.

Using the integration approximation matrix A is then equivalent to finding the Lagrange interpolating polynomial and taking the exact integral from -1 to t_i . The Lagrange interpolating polynomial is exact for polynomials of degree $N-1$, therefore the integration approximation matrix is also exact for polynomials of degree $N-1$, so that

$$\int_{-1}^{t_i} g(\tau) d\tau = \sum_{k=1}^N A_{ik} \cdot g(t_k), \quad \forall g(t) \in \mathbb{P}^{N-1}. \quad (4.45)$$

Convergence

Consider the integration approximation matrix, $A \in \mathbb{R}^{N \times N}$, which integrates a function from -1 to t_i exactly for polynomials of degree $N-1$ or less. For general functions there will be some error involved in the approximate integration, so that

$$\int_{-1}^{t_i} f(\tau) d\tau = \sum_{k=1}^N A_{ik} \cdot f(t_k) + E_i. \quad (4.46)$$

Theorem 4.2.1. *For functions $f(t) \in C^N[-1, 1]$, the error in the pseudospectral integration at the i th Gauss point is bounded by*

$$|E_i| \leq \left| \frac{d^N f(\zeta)}{dt^N} \right| \frac{2}{N!}, \quad \zeta \in [-1, 1], \quad (4.47)$$

and the pseudospectral integral will converge for any $f(t)$ whose derivatives are bounded, as the number of nodes used approaches infinity, so that

$$\lim_{N \rightarrow \infty} \left| \int_{-1}^{t_i} f(\tau) d\tau - \sum_{k=1}^N A_{ik} \cdot f(t_k) \right| \rightarrow 0, \quad \forall i. \quad (4.48)$$

Proof. The error in the integration using the approximation matrix can be bounded

by looking at the error of the Lagrange interpolating polynomial. The error in the Lagrange interpolating polynomial, $g(t) \doteq \sum_{k=1}^N f(t_k) \cdot L_k(t)$, is known to be [28]

$$f(t) - g(t) = \frac{d^N f}{dt^N}(\zeta) \cdot \frac{1}{N!} \cdot \omega(t) \doteq e(t), \quad (4.49)$$

for infinitely differentiable functions $f(t) \in C^\infty[-1, 1]$, where $\omega(t)$ is defined as (4.39), and ζ is a function of t and lies on the interval, $[-1, 1]$. The integral of $f(t)$ is

$$\int_{-1}^{t_i} f(\tau) d\tau = \int_{-1}^{t_i} g(\tau) d\tau + \int_{-1}^{t_i} e(\tau) d\tau. \quad (4.50)$$

The polynomial, $g(t)$, is integrated exactly by the integration approximation matrix because it is of degree $N - 1$, so that

$$\int_{-1}^{t_i} f(\tau) d\tau = \sum_{k=1}^N A_{ik} f(t_k) + \int_{-1}^{t_i} e(\tau) d\tau. \quad (4.51)$$

Therefore, the maximum error of integration is bounded by the integral of the absolute value of interpolation error, so that

$$\left| \int_{-1}^{t_i} f(\tau) d\tau - \sum_{k=1}^N A_{ik} \cdot f(t_k) \right| = \left| \int_{-1}^{t_i} e(\tau) d\tau \right|. \quad (4.52)$$

The integral of the interpolation error can then be bounded by

$$\begin{aligned} \left| \int_{-1}^{t_i} e(\tau) d\tau \right| &\leq \int_{-1}^{t_i} \left| \frac{d^N f(\zeta)}{dt^N} \frac{1}{N!} \cdot \omega(\tau) \right| d\tau, \\ &\leq \int_{-1}^1 \left| \frac{d^N f(\zeta)}{dt^N} \frac{1}{N!} \cdot \omega(\tau) \right| d\tau, \\ &\leq \int_{-1}^1 \left| \frac{d^N f(\zeta)}{dt^N} \frac{1}{N!} \right| \cdot |\omega(\tau)| d\tau, \\ &\leq \left| \frac{d^N f(\zeta)}{dt^N} \frac{1}{N!} \right| \int_{-1}^1 |\omega(\tau)| d\tau, \\ &\leq \left| \frac{d^N f(\zeta)}{dt^N} \right| \frac{2}{N!}. \end{aligned} \quad (4.53)$$

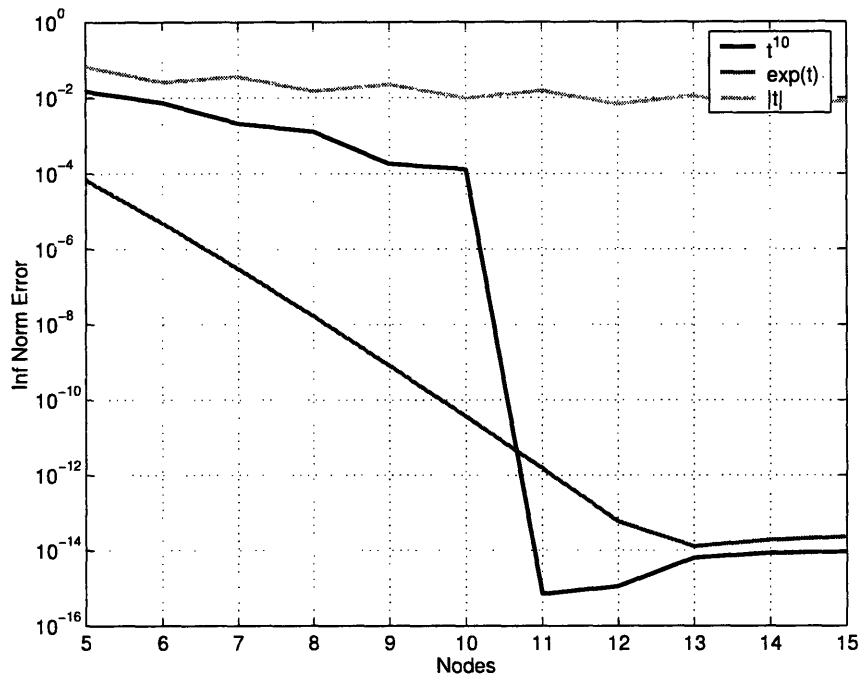


Figure 4-1: Integration Approximation Matrix Convergence

If the derivatives of $f(t)$ are bounded, $\left| \frac{d^k f(\zeta)}{dt^k} \right| \leq C, \forall k$, then the error in the approximate integration will approach zero as the number of nodes, N , approaches infinity, so that

$$\lim_{N \rightarrow \infty} \left| \int_{-1}^{t_i} f(\tau) d\tau - \sum_{k=1}^N A_{ik} \cdot f(t_k) \right| \rightarrow 0. \quad (4.54)$$

The error in the approximate integration is bounded by the error in the polynomial interpolation (4.52), therefore the approximate integral will converge for any function that can be approximated by a polynomial. \square

The convergence properties of the integration approximation matrix were tested on three different functions, t^{10} , $\exp(t)$, and $|t|$. The infinity norm error between the approximate integral and the exact integral is plotted as a function of the number of nodes used in Fig. 4-1. The integration of the first function t^{10} is exact, within the numerical round-off error, when using 11 nodes. This result is as expected for a 10-th degree polynomial. The error in the integration of $\exp(t)$, which is infinitely

differentiable, tends to zero very quickly. Finally, the error in the integration of $|t|$, which has a discontinuous first derivative, is tending to zero but at a much slower rate.

These results indicate that the error in the approximate integration of functions will converge to zero very quickly (exponentially) if the function is infinitely differentiable.

Integration Approximation Matrix Adjoint

In the analysis of the integral pseudospectral method, it is convenient to use the adjoint of the integration matrix A .

Lemma 4.2.1. *The adjoint of A is*

$$A_{ik}^\dagger = \frac{w_k}{w_i} A_{ki}. \quad (4.55)$$

Proof. The adjoint, A^\dagger , is a generalization of a transpose with a non-Euclidean inner product. If the discrete inner product used is

$$\langle f, g \rangle = \sum_{k=1}^N w_k \cdot f_k \cdot g_k, \quad (4.56)$$

then the adjoint of A must satisfy

$$\sum_{k=1}^N w_k \cdot f_k \cdot \sum_{i=1}^N A_{ki} \cdot g_i = \sum_{i=1}^N w_i \cdot \sum_{k=1}^N A_{ik}^\dagger \cdot f_k \cdot g_i, \quad (4.57)$$

for every $f, g \in \mathbb{R}^N$, where w_k is the Gauss quadrature weights. \square

It can be shown that the adjoint of A approximates the integral from t_i to 1, so that

$$\int_{t_i}^1 f(\tau) d\tau \approx \sum_{k=1}^N A_{ik}^\dagger \cdot f(t_k), \quad i = 1, \dots, N, \quad (4.58)$$

$$A_{ik}^\dagger = \int_{t_i}^1 L_k(\tau) d\tau.$$

Ordinary Differential Equations

The integration approximation matrix can be used to solve initial value problems [2]. The ordinary differential equation

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}(t), t), \\ \mathbf{x}(t_o) &= \mathbf{x}_o,\end{aligned}\tag{4.59}$$

can be integrated into the form

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{x}(t_o) + \int_{t_o}^t \mathbf{f}(\mathbf{x}(\tau), \tau) d\tau, \\ \mathbf{x}(t_o) &= \mathbf{x}_o,\end{aligned}\tag{4.60}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$. The dynamic equation (4.60) can then be approximated by the integration approximation matrix A , using the approximate state $\mathbf{X}(t) \approx \mathbf{x}(t)$. The approximate state at the i th Gauss point is

$$\begin{aligned}\mathbf{X}(t_i) &= \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}(t_k), t_k), \\ \mathbf{X}(t_o) &= \mathbf{x}_o.\end{aligned}\tag{4.61}$$

This set of $N + 1$ equations (N equations for the state dynamics and one for the boundary condition) define the approximate state at $N + 1$ points (the boundary point plus the N Gauss points).

Lemma 4.2.2. *The pseudospectral solution to the initial value problem satisfies the boundary condition and differential equation at the collocation points, so that*

$$\begin{aligned}\mathbf{X}(t_o) &= \mathbf{x}_o, \\ \frac{d\mathbf{X}}{dt}(t_i) &= \mathbf{f}(\mathbf{X}(t_i), t_i), \quad i = 1, \dots, N.\end{aligned}\tag{4.62}$$

Proof. The approximation of the state $\mathbf{X}(t)$ is the integral of the polynomial approximation to the function $\mathbf{f}(\mathbf{X}(t_k), t_k)$ based on the definition of the integral approxi-

mation matrix, so that

$$\mathbf{X}(t) = \mathbf{X}(t_o) + \int_{t_o}^t \sum_{k=1}^N \mathbf{f}(\mathbf{X}(t_k), t_k) \cdot L_k(\tau) d\tau , \quad (4.63)$$

where $L_k(t)$, $k = 1, \dots, N$, are the Lagrange interpolating polynomials. Differentiating the approximation of the state (4.63) results in

$$\frac{d\mathbf{X}}{dt}(t) = \sum_{k=1}^N \mathbf{f}(\mathbf{X}(t_k), t_k) \cdot L_k(t) . \quad (4.64)$$

Using the property of the Lagrange polynomials (4.40), the derivative of the approximate state at the collocation points becomes

$$\frac{d\mathbf{X}}{dt}(t_i) = \mathbf{f}(\mathbf{X}(t_i), t_i), \quad i = 1, \dots, N . \quad (4.65)$$

This result means that the approximate state $\mathbf{X}(t)$ satisfies the initial condition and the differential equation at all the collocation points. \square

The approximate state can be found by solving (4.61) to determine its value at the N collocation points. Using the initial condition, the value of the approximate state is known at a total of $N + 1$ points and a polynomial of degree N can be fit to these points. This fact implies that $\mathbf{X}(t)$ is a degree N polynomial, and is in agreement with (4.63), which indicates that the approximate state is the integral of a degree $N - 1$ polynomial.

An example of an initial value problem is

$$\dot{x}(t) = -2 \cdot x(t) + 3 \cdot e^t , \quad x(0) = -3 , \quad (4.66)$$

which has an exact solution of

$$x(t) = -4 \cdot e^{-2t} + e^t . \quad (4.67)$$

This problem can be solved using the pseudospectral integration approximation on

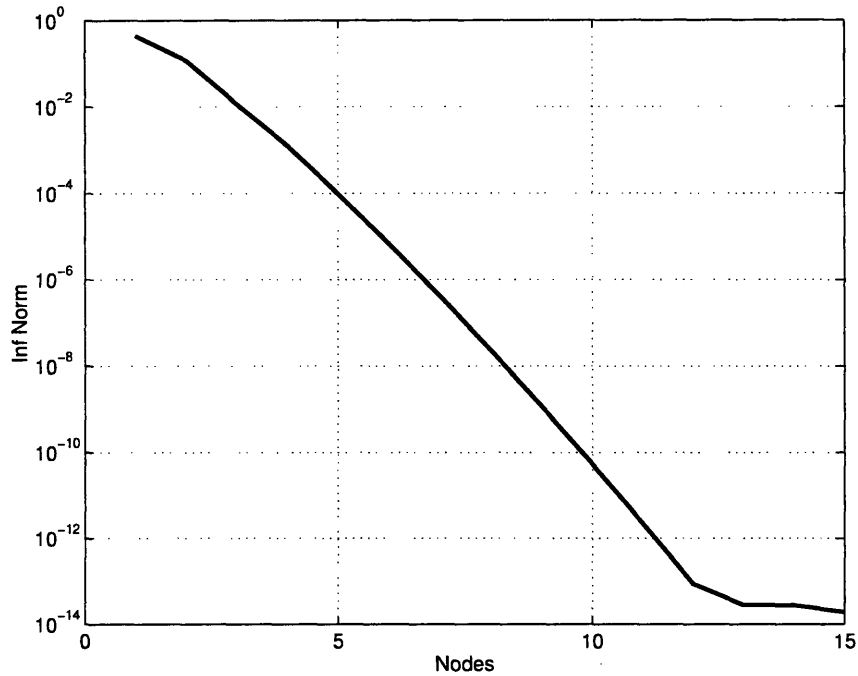


Figure 4-2: Initial Value Problem Convergence

the interval from $[0, 1]$ by solving the linear system for X_N .

$$X_N = X_N(0) + \frac{1}{2} \cdot A \cdot (-2 \cdot X_N + 3 \cdot e^{t_N}) \quad (4.68)$$

$X_N \in \mathbb{R}^N$ is the vector of the approximation to $x(t)$ at the N Gauss points, $X_N(0) = -3$ is the initial condition, and $t_N \in \mathbb{R}^N$ is the vector of times at the Gauss points. The infinity norm error between the pseudospectral solution and the true solution is shown as a function of the number of Gauss nodes used in Fig. 4-2. The convergence plot indicates that the pseudospectral approximation is converging very rapidly (exponentially) to the exact solution.

4.2.3 Pseudospectral Transcription

The integration approximation matrix along with the Gauss quadrature rule can now be used to discretize the integral Bolza problem into an NLP.

The pseudospectral transcription of the integral form of the continuous optimal control problem (4.1 - 4.4) is expressed as an NLP where the objective is to minimize an algebraic cost function subject to a set of algebraic constraint functions. The cost is

$$J = \Phi(\mathbf{X}(t_f), t_f) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N g(\mathbf{X}_k, \mathbf{U}_k, t_k) \cdot w_k, \quad (4.69)$$

which comes from approximating the integral in the continuous cost (4.2) with a Gauss quadrature. Note that a notation simplification has been made for the approximate states and controls at the Gauss points, so that

$$\begin{aligned} \mathbf{X}_i &= \mathbf{X}(t_i) \in \mathbb{R}^n, \\ \mathbf{U}_i &= \mathbf{U}(t_i) \in \mathbb{R}^m, \quad i = 1, \dots, N. \end{aligned} \quad (4.70)$$

The integral form of the dynamic constraints (4.1) are approximated using the integral approximation matrix, so that

$$\mathbf{X}_i = \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k), \quad i = 1, \dots, N. \quad (4.71)$$

The nonlinear boundary constraints,

$$\phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) = \mathbf{0}, \quad (4.72)$$

remain unchanged from the continuous form (4.3). Finally, the final states occurring in (4.69) and (4.72) are defined as

$$\mathbf{X}(t_f) = \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k), \quad (4.73)$$

which is a Gauss quadrature approximation to the continuous definition of the final states (4.4).

The algebraic cost (4.69), along with the nonlinear constraints (4.71 - 4.72), make up the nonlinear programming problem, which can then be solved by well-developed

NLP algorithms and software. Finding the approximate solution to the continuous optimal control problem in this way is referred to as the direct solution, because the continuous problem is approximated as an NLP and solved directly. An indirect solution to the continuous optimal control problem can be found by approximating the continuous first-order necessary conditions.

Indirect Pseudospectral Solution

A solution to the integral Bolza problem can be found by pseudospectral collocation of the continuous first-order necessary conditions (4.13). This discretization at the set of Gauss points converts the continuous necessary conditions into a set of nonlinear algebraic equations, so that

$$\begin{aligned}
\mathbf{X}_i &= \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k), \\
\mathbf{P}_i^T &= \frac{\partial g}{\partial \mathbf{X}_i} + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i}, \\
\mathbf{0} &= \frac{\partial g}{\partial \mathbf{U}_i} + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{U}_i}, \\
& i = 1, \dots, N,
\end{aligned} \tag{4.74}$$

$$\mathbf{X}(t_f) = \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k),$$

$$\phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) = \mathbf{0},$$

$$\frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} = \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)}.$$

The variables that satisfy this set of equations, $\mathbf{X}_i, \mathbf{U}_i, \mathbf{P}_i, i = 1, \dots, N$, and $\boldsymbol{\nu}$, make up an approximate solution of the integral Bolza problem. Solving the problem in this way is classified as an indirect approach to solving the problem.

4.3 KKT Conditions

The direct approach to approximate the solution to the integral Bolza problem is found by solving the nonlinear program defined by the cost (4.69) and constraints (4.71 - 4.72). The solution to the NLP is defined by the Karush-Kuhn-Tucker (KKT) conditions. These KKT conditions are the first-order optimality conditions for the NLP. It will be shown that the KKT conditions are equivalent to the discretized form of the continuous first-order necessary conditions for the integral Bolza problem (4.74). First consider the case with fixed initial and final times, control uniquely determined by the equation

$$\frac{\partial g}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 0, \quad (4.75)$$

and no state or control path constraints.

Theorem 4.3.1. *The KKT conditions from the resulting NLP of the pseudospectral discretization are exactly equivalent to the discretized form of the first-order optimality conditions (4.74). These conditions are equivalent by relating the KKT multipliers of the NLP, $\tilde{\mathbf{P}}_i$, to the approximate integral costates at the Gauss points, \mathbf{P}_i , using the integral costate mapping principle, so that*

$$\mathbf{P}_i = \frac{2}{w_i(t_f - t_o)} \tilde{\mathbf{P}}_i, \quad i = 1, \dots, N. \quad (4.76)$$

Proof. The KKT conditions for the NLP from discretizing the continuous problem (4.69 - 4.73), are found by applying Karush-Kuhn-Tucker theorem [4]. These conditions define the direct solution to the integral Bolza problem. The cost function is augmented with KKT multipliers $\tilde{\mathbf{P}} \in \mathbb{R}^n$ and Lagrange multipliers $\boldsymbol{\nu} \in \mathbb{R}^q$, so that

$$\begin{aligned} J_a = & \Phi(\mathbf{X}(t_f), t_f) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot g(\mathbf{X}_k, \mathbf{U}_k, t_k) \\ & - \sum_{i=1}^N \tilde{\mathbf{P}}_i^T \cdot \left(\mathbf{X}_i - \mathbf{X}(t_o) - \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k) \right) \\ & - \boldsymbol{\nu}^T \cdot \boldsymbol{\phi}(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f). \end{aligned} \quad (4.77)$$

The KKT conditions are found by setting the partial derivatives of the augmented cost with respect to all free variables equal to zero. The free variables in this case are the multipliers, $\tilde{\mathbf{P}}$ and $\boldsymbol{\nu}$, the state variables at the Gauss points, \mathbf{X} , the control variables at the Gauss points, \mathbf{U} , and finally the initial state, $\mathbf{X}(t_o)$.

The partial of the augmented cost with respect to the multipliers set equal to zero, return the original constraints, so that

$$\begin{aligned}\frac{\partial J_a}{\partial \tilde{\mathbf{P}}_i} = \mathbf{0} &= \mathbf{X}_i - \mathbf{X}(t_o) - \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k), \\ \frac{\partial J_a}{\partial \boldsymbol{\nu}} = \mathbf{0} &= \boldsymbol{\phi}(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f).\end{aligned}\quad (4.78)$$

Setting the partial of the augmented cost with respect to the states results in

$$\begin{aligned}\frac{\partial J_a}{\partial \mathbf{X}_i} = \mathbf{0} &= -\tilde{\mathbf{P}}_i^T + w_i \frac{(t_f - t_o)}{2} \cdot \frac{\partial g}{\partial \mathbf{X}_i} + \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ki} \cdot \tilde{\mathbf{P}}_k^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i} \\ &+ w_i \frac{(t_f - t_o)}{2} \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i}.\end{aligned}\quad (4.79)$$

Note that chain rule was applied to final cost and boundary constraint terms because the final state is expressed in terms of state at all collocation points (4.73). Taking advantage of the adjoint of A , Lemma 4.2.1, and dividing through by the Gauss weights w_i and $\frac{(t_f - t_o)}{2}$, results in

$$\begin{aligned}\mathbf{0} &= -\frac{2}{w_i(t_f - t_o)} \tilde{\mathbf{P}}_i^T + \frac{\partial g}{\partial \mathbf{X}_i} + \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i} \\ &+ \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \frac{2}{w_k(t_f - t_o)} \tilde{\mathbf{P}}_k^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i}.\end{aligned}\quad (4.80)$$

Simplification by the integral costate mapping principle defines the relation between the KKT multipliers $\tilde{\mathbf{P}}_i$, and the approximation for the integral costate $\mathbf{P}(t_i) = \mathbf{P}_i$, so that

$$\mathbf{P}_i = \frac{2}{w_i(t_f - t_o)} \tilde{\mathbf{P}}_i, \quad i = 1, \dots, N. \quad (4.81)$$

Using the integral costate mapping principle, the constraint (4.80) is reduced to

$$\mathbf{P}_i^T = \frac{\partial g}{\partial \mathbf{X}_i} + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i}. \quad (4.82)$$

This equation is the discrete form of the continuous costate dynamics for the integral Bolza problem.

The same simplification procedure can be applied for the partial derivatives of the augmented cost (4.77) with respect to the control. The result is

$$\mathbf{0} = \frac{\partial g}{\partial \mathbf{U}_i} + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{U}_i}. \quad (4.83)$$

This equation is the discrete form of the control equation for the continuous integral problem.

The final KKT condition is found by taking the partial of the augmented cost with respect to the initial state $\mathbf{X}(t_o)$, so that

$$\frac{\partial J_a}{\partial \mathbf{X}(t_o)} = \mathbf{0} = \sum_{k=1}^N \tilde{\mathbf{P}}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)}. \quad (4.84)$$

Applying the integral costate mapping principle (4.81) results in

$$\frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} = \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)}. \quad (4.85)$$

This equation is the discrete form of the costate boundary conditions for the integral problem.

The resulting KKT conditions (4.78, 4.82, 4.83, 4.85) define a set of nonlinear algebraic conditions, which define the solution to the NLP. This set of equations is exactly equivalent to the discretized form of the continuous necessary conditions (4.74), including the boundary points. The equivalence of the KKT conditions and the discretized necessary conditions implies that the KKT multipliers satisfy the discrete form of the integral costate dynamics and the integral costate mapping principle is

valid. This demonstrates the ability to estimate the costates of the integral Bolza problem directly from the KKT multipliers from the direct solution. \square

Numerical evidence has suggested that the costate estimates from the integral Bolza problem are significantly more accurate than the costate estimates of the Legendre pseudospectral method.

A summary of the direct and indirect relationship for both the Legendre pseudospectral method and the Gauss pseudospectral method is shown in Fig. 4-3. The chart shows the differential formulations on the left half and the integral formulations on the right. In the second column the original optimal control problem is at the top. Moving down the column is the indirect formulation of the differential form. The first step is the formulation of the continuous first-order necessary conditions which define the solution to the optimal control problem, indicated by the double arrow. The last step is the pseudospectral discretization of the necessary conditions resulting in a set of algebraic equations which approximate the solution, indicated by the single arrow. In the first column is the direct formulation of the problem. The continuous problem is discretized into a discrete NLP. The next step is the formulation of the KKT conditions which define the solution to the NLP. For the differential pseudospectral transcription, the KKT conditions are not the same as the discretized first-order necessary conditions (see Section 3.3.2). This error is due to the presence of the defects in the costate dynamics and boundary conditions resulting from the discretization at LGL points, and is not the case for the Gauss pseudospectral transcription.

The third column shows the indirect formulation of the optimal control problem in integral form. The first step is the formulation of the continuous first-order necessary conditions. These conditions have been shown to be equivalent to the conditions of the differential formulation based on the relationship between the two costates. The final step is the pseudospectral discretization of the necessary conditions at Gauss points. In the fourth column is the direct formulation of the problem. The continuous optimal control problem in integral form is transcribed to an NLP which has a solution defined by the KKT conditions. This set of algebraic KKT conditions is exactly equivalent to the discretized first-order necessary conditions. This relation-

Legendre Pseudospectral

Gauss Pseudospectral

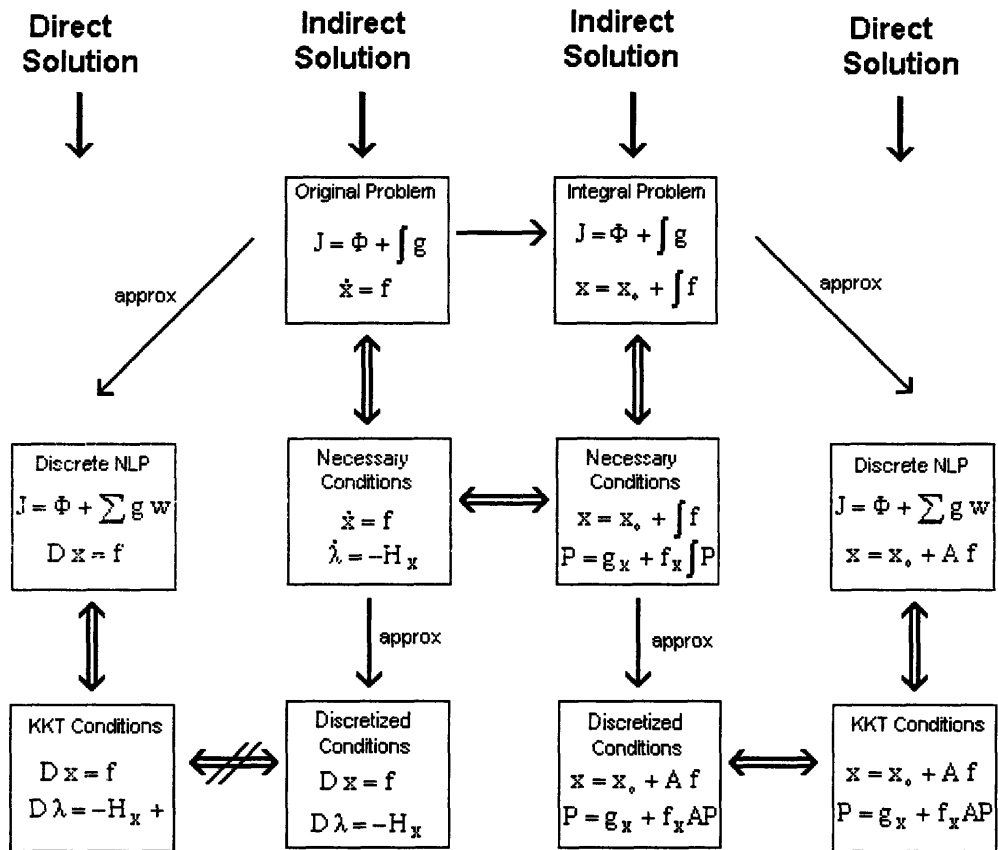


Figure 4-3: Direct/Indirect Pseudospectral Solutions

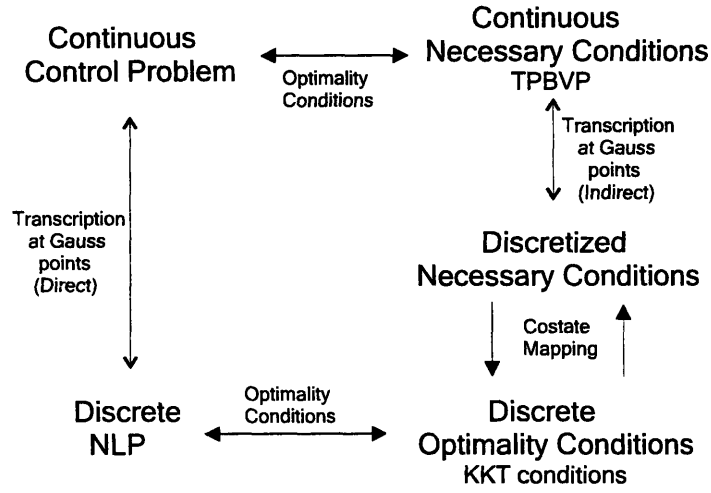


Figure 4-4: Gauss Pseudospectral Discretization

ship indicates that the KKT conditions are consistent with the continuous first-order necessary conditions. The discretization of the optimal control problem at Gauss points has no defects, so that the direct solution is exactly equivalent to the indirect solution. This result means that the operations of discretization and optimization commute, and the solution is the same whether you discretize then optimize, or optimize then discretize. This consistency is shown in Fig. 4-4, which is not true of other pseudospectral methods.

4.3.1 Pontryagin

Now consider the case where the control can not be determined from (4.75) and Pontryagin's maximum principle is required to determine the first-order necessary conditions (see Section 4.1.1).

Consider the discretized optimal control problem with a control inequality constraint. The states and controls are discretized with N Gauss collocation points, Gauss weights w_k , and integration approximation matrix A . The cost is

$$J = \Phi(\mathbf{X}(t_f), t_f) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot g(\mathbf{X}_k, \mathbf{U}_k, t_k), \quad (4.86)$$

with dynamic constraints in integral form,

$$\mathbf{X}_i = \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, t_i), \quad i = 1, \dots, N, \quad (4.87)$$

boundary constraints,

$$\phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) = 0, \quad (4.88)$$

and control path constraint,

$$\mathbf{m}(\mathbf{U}_i, t_i) \leq \mathbf{0}, \quad i = 1, \dots, N. \quad (4.89)$$

The control constraint, $\mathbf{m} : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^b$, defines the set of admissible controls at each Gauss point.

Theorem 4.3.2. *The pseudospectral transcription of the integral Bolza problem also satisfies Pontryagin's maximum principle at the collocation points.*

Proof. The pseudospectral transcription of the integral Bolza problem is shown to satisfy Pontryagin's maximum principle at the collocation points by deriving the KKT conditions from the augmented cost function.

The Lagrangian, or augmented cost function, is found by subtracting the constraints, multiplied by Lagrange multipliers, from the cost. The multipliers associated with the control constraints, $\boldsymbol{\mu}_i \in \mathbb{R}^b$, $i = 1, \dots, N$, are zero when the control is off the boundary of admissible controls and negative when the control is on the boundary of admissible controls. If

$$\boldsymbol{\mu}_i = \begin{bmatrix} \mu_{i1} \\ \vdots \\ \mu_{ib} \end{bmatrix}, \quad \mathbf{m}(\mathbf{U}_i, t_i) = \begin{bmatrix} m_1(\mathbf{U}_i, t_i) \\ \vdots \\ m_b(\mathbf{U}_i, t_i) \end{bmatrix}, \quad (4.90)$$

then

$$\begin{aligned} \mu_{ik} = 0 & : m_k(\mathbf{U}_i, t_i) < 0, \\ \mu_{ik} < 0 & : m_k(\mathbf{U}_i, t_i) = 0, \end{aligned} \quad (4.91)$$

for $i = 1, \dots, N$ and $k = 1, \dots, b$. This relation is a result of duality and complimentary slackness [4] and is shown in Section 2.3.3.

The augmented cost function of the discrete optimal control problem is

$$J_a = \Phi + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot g_k - \boldsymbol{\nu}^T \cdot \boldsymbol{\phi} - \sum_{i=1}^N \boldsymbol{\mu}_i^T \cdot \mathbf{m}(\mathbf{U}_i, t_i) - \sum_{k=1}^N \tilde{\mathbf{P}}_k^T \cdot \left(\mathbf{X}_k - \mathbf{X}(t_o) - \frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ik} \cdot \mathbf{f}_i \right), \quad (4.92)$$

where $g_k = g(\mathbf{X}_k, \mathbf{U}_k, t_k)$, $\mathbf{f}_k = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k)$, and $\boldsymbol{\nu} \in \mathbb{R}^q$, $\tilde{\mathbf{P}}_k \in \mathbb{R}^n$, $\boldsymbol{\mu}_k \in \mathbb{R}^b$ are Lagrange or KKT multipliers. The KKT conditions are found by setting the gradient of the augmented cost equal to zero, so that

$$\delta J_a = \frac{\partial J_a}{\partial \mathbf{X}} \cdot \delta \mathbf{X} + \frac{\partial J_a}{\partial \mathbf{U}} \cdot \delta \mathbf{U} + \frac{\partial J_a}{\partial \mathbf{X}(t_o)} \cdot \delta \mathbf{X}(t_o) + \frac{\partial J_a}{\partial \tilde{\mathbf{P}}} \cdot \delta \tilde{\mathbf{P}} + \frac{\partial J_a}{\partial \boldsymbol{\nu}} \cdot \delta \boldsymbol{\nu} + \frac{\partial J_a}{\partial \boldsymbol{\mu}} \cdot \delta \boldsymbol{\mu}. \quad (4.93)$$

At the optimal solution the gradient must be zero and therefore each of the coefficients associated with each variation must be zero. The partials of the augmented cost with respect to the control are

$$\begin{aligned} \frac{\partial J_a}{\partial \mathbf{U}_i} &= \frac{(t_f - t_o)}{2} w_i \cdot \left[\frac{\partial g_i}{\partial \mathbf{U}_i} + \left(\sum_{k=1}^N \tilde{\mathbf{P}}_k \cdot \frac{A_{ki}}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu} \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{U}_i} \right] \\ &\quad - \boldsymbol{\mu}_i \cdot \frac{\partial \mathbf{m}}{\partial \mathbf{U}_i} = 0, \end{aligned} \quad (4.94)$$

for $i = 1, \dots, N$. This equation can be simplified to

$$\frac{\partial J_a}{\partial \mathbf{U}_i} = \frac{(t_f - t_o)}{2} w_i \cdot \frac{\partial \mathcal{H}_i}{\partial \mathbf{U}_i} - \boldsymbol{\mu}_i \cdot \frac{\partial \mathbf{m}}{\partial \mathbf{U}_i}, \quad (4.95)$$

where \mathcal{H}_i is defined as

$$\mathcal{H}_i \doteq g_i + \left(\sum_{k=1}^N \tilde{\mathbf{P}}_k \cdot \frac{A_{ki}}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu} \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}(t_f)} \right) \cdot \mathbf{f}_i. \quad (4.96)$$

Therefore, the variation of the cost at the optimal solution with respect to changes

in the control at the i th collocation point is

$$\delta J_a = \left(\frac{(t_f - t_o)}{2} w_i \cdot \frac{\partial \mathcal{H}_i}{\partial \mathbf{U}_i} - \boldsymbol{\mu}_i \cdot \frac{\partial \mathbf{m}}{\partial \mathbf{u}_i} \right) \cdot \delta \mathbf{U}_i = 0. \quad (4.97)$$

Note that at the optimal solution all other necessary conditions are met, i.e. the gradient of the augmented cost with respect to all the other variables is zero. The total variation of the cost can be expressed as

$$\delta J_a = \frac{(t_f - t_o)}{2} w_i \cdot (\mathcal{H}_i(\mathbf{U}_i^* + \delta \mathbf{U}_i) - \mathcal{H}_i(\mathbf{U}_i^*)) - \boldsymbol{\mu}_i \cdot (\mathbf{m}(\mathbf{U}_i^* + \delta \mathbf{U}_i, t_i) - \mathbf{m}(\mathbf{U}_i^*, t_i)) = 0. \quad (4.98)$$

There are only two possible cases at the i th point, the control is off the boundary $\mathbf{m}(\mathbf{U}_i, t_i) < 0$, or the control is on the boundary, $m_j(\mathbf{U}_i, t_i) = 0$, for some $j \in [1, \dots, b]$. When the control is off the boundary $\boldsymbol{\mu}_i$ is equal to zero, so that the condition for the control (4.97) is reduced to

$$\frac{\partial \mathcal{H}_i}{\partial \mathbf{U}_i} = 0. \quad (4.99)$$

This condition is the same that is found if the control is unbounded (4.83). When the control is on the boundary the associated multiplier is negative, $\mu_{ij} < 0$, and the control constraint is zero, $m_j(\mathbf{U}_i^*, t_i) = 0$. Taking advantage of this, the condition for the control can be expressed

$$\mathcal{H}_i(\mathbf{U}_i^* + \delta \mathbf{U}_i) - \frac{2}{w_i \cdot (t_f - t_o)} \cdot \sum_j \mu_{ij} \cdot m_j(\mathbf{U}_i^* + \delta \mathbf{U}_i, t_i) = \mathcal{H}_i(\mathbf{U}_i^*), \quad (4.100)$$

where $j \in [1, \dots, b]$ are all of the tight or active constraints. This equation can also be expressed as

$$\begin{aligned} \mathcal{H}_i(\mathbf{U}_i^* + \delta \mathbf{U}_i) - \alpha &= \mathcal{H}_i(\mathbf{U}_i^*), \\ \alpha &= \frac{2}{w_i \cdot (t_f - t_o)} \cdot \sum_j \mu_{ij} \cdot m_j(\mathbf{U}_i^* + \delta \mathbf{U}_i, t_i), \end{aligned} \quad (4.101)$$

where α must be non-negative. This fact is because both the Gauss weight w_i and the

time interval $t_f - t_o$ are positive, μ_{ij} is negative and $m_j(\mathbf{U}_i^* + \delta\mathbf{U}_i, t_i)$ is non-positive. If $m_j(\mathbf{U}_i^* + \delta\mathbf{U}_i, t_i)$ were positive, the constraint would be violated, meaning this would not be a feasible variation of the control. Because α is non-negative, the function \mathcal{H}_i is always increased by a feasible variation of the control,

$$\mathcal{H}_i(\mathbf{U}_i^* + \delta\mathbf{U}_i) \geq \mathcal{H}_i(\mathbf{U}_i^*). \quad (4.102)$$

Therefore the optimal control at the i th collocation point, \mathbf{U}_i^* , is defined as the feasible control which minimizes \mathcal{H}_i . This relation must be true at all the collocation points.

The function \mathcal{H}_i can be shown to be the approximation of the Hamiltonian at the i th collocation point. The adjoint of the integration approximation matrix $A_{ik}^\dagger = \frac{w_k}{w_i} A_{ki}$, is used to simplify \mathcal{H}_i . Substituting into the definition of \mathcal{H}_i (4.96) yields

$$\mathcal{H}_i = g_i + \left(\sum_{k=1}^N \tilde{\mathbf{P}}_k \cdot \frac{A_{ik}^\dagger}{w_k} + \frac{\partial\Phi}{\partial\mathbf{X}(t_f)} - \boldsymbol{\nu} \cdot \frac{\partial\phi}{\partial\mathbf{X}(t_f)} \right) \cdot \mathbf{f}_i. \quad (4.103)$$

Further simplification is done by the integral costate mapping principle (4.76), to obtain

$$\mathcal{H}_i = g_i + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N \mathbf{P}_k \cdot A_{ik}^\dagger + \frac{\partial\Phi}{\partial\mathbf{X}(t_f)} - \boldsymbol{\nu} \cdot \frac{\partial\phi}{\partial\mathbf{X}(t_f)} \right) \cdot \mathbf{f}_i. \quad (4.104)$$

This relation is the discrete form of continuous Hamiltonian (4.33) at the i th collocation point. Therefore, the approximation to the optimal control satisfies Pontryagin's maximum principle, that is to say, the optimal control minimizes the discrete Hamiltonian at each collocation point. \square

4.3.2 Free Time

Now consider the problem with free initial and final times. For problems with free time, the initial or final times are determined by the conditions in terms of the

Hamiltonian (see Section 2.4), so that

$$\begin{aligned}
-\mathcal{H}(\mathbf{x}(t_o), \boldsymbol{\lambda}(t_o), \mathbf{u}(t_o), t_o) - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} &= 0, \\
\mathcal{H}(\mathbf{x}(t_f), \boldsymbol{\lambda}(t_f), \mathbf{u}(t_f), t_f) + \frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} &= 0.
\end{aligned} \tag{4.105}$$

For the pseudospectral approximation in integral form, the KKT conditions for free time are determined by finding the partial of the augmented cost (4.77), or Lagrangian, with respect to the initial and final times and setting equal to zero.

Final Time

The partial of the augmented cost function (4.77) with respect to the final time is found using the chain rule. The terminal cost and boundary constraints are functions of the final state, which is a function of the final time, based on the definition of the final state (4.73). Also, the cost function g and dynamics \mathbf{f} are functions of time (if time varying) which is also a function of the final time, based of the time transformation to $[-1, 1]$ (4.34). The partial of the augmented cost with respect to the final time is then

$$\begin{aligned}
\frac{\partial J_a}{\partial t_f} &= \frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{1}{2} \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \sum_{k=1}^N w_k \cdot \mathbf{f}_k \\
&+ \frac{(t_f - t_o)}{2} \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \sum_{k=1}^N w_k \cdot \frac{\partial \mathbf{f}_k}{\partial t} \cdot \frac{\tau_k + 1}{2} \\
&+ \frac{1}{2} \sum_{k=1}^N w_k \cdot g_k + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \frac{\partial g_k}{\partial t} \cdot \frac{\tau_k + 1}{2} \\
&+ \frac{1}{2} \sum_{i=1}^N \tilde{\mathbf{P}}_i^T \sum_{i=1}^N A_{ik} \cdot \mathbf{f}_k + \frac{(t_f - t_o)}{2} \sum_{i=1}^N \tilde{\mathbf{P}}_i^T \sum_{i=1}^N A_{ik} \cdot \frac{\partial \mathbf{f}_k}{\partial t} \cdot \frac{\tau_k + 1}{2}.
\end{aligned} \tag{4.106}$$

The expression can be simplified by using the integral costate mapping principle (4.76), and the adjoint of the integration approximation matrix, Lemma 4.2.1. The

result is

$$\begin{aligned}
0 = & \frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \left[\frac{\partial g_k}{\partial t} \right. \\
& + \left. \left(\frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ki}^\dagger \cdot \mathbf{P}_i^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_k}{\partial t} \right] \cdot \frac{\tau_k + 1}{2} \\
& + \frac{1}{2} \sum_{k=1}^N w_k \cdot \left[g_k + \left(\frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ki}^\dagger \cdot \mathbf{P}_i^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \mathbf{f}_k \right].
\end{aligned} \tag{4.107}$$

The expression can be further simplified using the definition of the approximation to the Hamiltonian (4.104). The discrete Hamiltonian and the partial of the Hamiltonian with respect to time are

$$\begin{aligned}
\mathcal{H}_k &= g_k + \left(\frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ki}^\dagger \cdot \mathbf{P}_i^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \mathbf{f}_k, \\
\frac{\partial \mathcal{H}_k}{\partial t} &= \frac{\partial g_k}{\partial t} + \left(\frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ki}^\dagger \cdot \mathbf{P}_i^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_k}{\partial t}.
\end{aligned} \tag{4.108}$$

Using these two equations the KKT condition for the final time can be simplified to

$$0 = \frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \frac{\partial \mathcal{H}_k}{\partial t} \cdot \frac{\tau_k + 1}{2} + \frac{1}{2} \sum_{k=1}^N w_k \cdot \mathcal{H}_k. \tag{4.109}$$

This equation does not appear to be the discrete form of the continuous final time condition (4.15). However, it can be shown to be the same by looking at the continuous version of the final time condition (4.109).

The summation terms in (4.109) are Gauss quadrature approximations to integrals. The continuous form of this condition can be expressed as

$$0 = \frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{(t_f - t_o)}{2} \int_{-1}^1 \frac{\partial \mathcal{H}(t)}{\partial t} \cdot \frac{\tau + 1}{2} d\tau + \frac{1}{2} \int_{-1}^1 \mathcal{H}(t) d\tau. \tag{4.110}$$

This equation can then be transformed back to the time interval $[t_o, t_f]$, noting that

$\frac{\tau + 1}{2} = \frac{t - t_o}{t_f - t_o}$, so that

$$0 = \frac{\partial \Phi}{\partial t_f} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{1}{(t_f - t_o)} \int_{t_o}^{t_f} \frac{\partial \mathcal{H}(t)}{\partial t} \cdot (t - t_o) dt + \frac{1}{(t_f - t_o)} \int_{t_o}^{t_f} \mathcal{H}(t) dt. \quad (4.111)$$

This equation is then the continuous form of the discrete condition (4.109), which is equivalent to the original final time condition (4.105). It can be shown that the sum of the integral terms in the above relation are equal to the Hamiltonian at the final time.

The first step is to show that at the optimal solution, the partial of the Hamiltonian is equivalent to the total derivative of the Hamiltonian. This relation can be shown by finding the total derivative of the Hamiltonian, so that

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial t} + \frac{\partial \mathcal{H}}{\partial \mathbf{x}} \cdot \dot{\mathbf{x}} + \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \cdot \dot{\mathbf{u}} + \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} \cdot \dot{\boldsymbol{\lambda}}. \quad (4.112)$$

The first-order necessary conditions (see Section 2.4),

$$\dot{\mathbf{x}} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}, \quad \dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}, \quad 0 = \frac{\partial \mathcal{H}}{\partial \mathbf{u}}, \quad (4.113)$$

reduce the right hand side so that $\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial t}$. Therefore, the integrals in (4.111) can be expressed as

$$\frac{1}{(t_f - t_o)} \left(\int_{t_o}^{t_f} \mathcal{H} dt + \int_{t_o}^{t_f} \frac{d\mathcal{H}}{dt} \cdot (t - t_o) dt \right). \quad (4.114)$$

The second term can be integrated by parts resulting in

$$\frac{1}{(t_f - t_o)} \left(\int_{t_o}^{t_f} \mathcal{H} dt + \mathcal{H}(t_f) \cdot t_f - \mathcal{H}(t_o) \cdot t_o - \int_{t_o}^{t_f} \mathcal{H} dt - \int_{t_o}^{t_f} \frac{d\mathcal{H}}{dt} \cdot t_o dt \right). \quad (4.115)$$

The last term can be integrated, so that

$$\frac{1}{(t_f - t_o)} \left(\int_{t_o}^{t_f} \mathcal{H} dt - \int_{t_o}^{t_f} \mathcal{H} dt + \mathcal{H}(t_f) \cdot t_f - \mathcal{H}(t_o) \cdot t_o - \mathcal{H}(t_f) \cdot t_o + \mathcal{H}(t_o) \cdot t_o \right). \quad (4.116)$$

The expression is then simplified to

$$\frac{1}{(t_f - t_o)} (\mathcal{H}(t_f) \cdot t_f - \mathcal{H}(t_f) \cdot t_o) , \quad (4.117)$$

which is the Hamiltonian at the final time, $\mathcal{H}(t_f)$.

This derivation shows that the pseudospectral transcription of the problem in integral form has a KKT condition for the final time (4.109) that is the discrete form of a condition (4.111) that is analytically equivalent to the final time condition for the original analytic control problem (4.105). The condition (4.111) can be derived from the continuous optimal control problem by finding the final time condition by calculus of variations, after transforming the time interval to $[-1, 1]$. In differential form the optimal control problem is

$$\begin{aligned} J &= \Phi(\mathbf{x}(t_f, t_f) + \frac{(t_f - t_o)}{2} \int_{-1}^1 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \quad d\tau , \\ &\frac{(t_f - t_o)}{2} \frac{d\mathbf{x}}{d\tau} = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) , \\ &\phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) = 0 . \end{aligned} \quad (4.118)$$

In integral form the optimal control problem is

$$\begin{aligned} J &= \Phi(\mathbf{x}(t_f, t_f) + \frac{(t_f - t_o)}{2} \int_{-1}^1 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \quad d\tau , \\ \mathbf{x}(\tau) &= \mathbf{x}(t_o) + \frac{(t_f - t_o)}{2} \int_{-1}^{\tau} \mathbf{f}(\mathbf{x}(s), \mathbf{u}(s), s) \quad ds , \\ &\phi(\mathbf{x}(t_o), t_o, \mathbf{x}(t_f), t_f) = 0 . \end{aligned} \quad (4.119)$$

Initial Time

The KKT condition for the initial time is found using the partial of the Lagrangian (4.77) with respect to the initial time. The chain rule must be used again because of

the dependence of the final state on the initial time, so that

$$\begin{aligned}
\frac{\partial J_a}{\partial t_o} = & -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_o} - \frac{1}{2} \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \sum_{k=1}^N w_k \cdot \mathbf{f}_k \\
& + \frac{(t_f - t_o)}{2} \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \sum_{k=1}^N w_k \cdot \frac{\partial \mathbf{f}_k}{\partial t} \cdot \frac{-\tau_k + 1}{2} \\
& - \frac{1}{2} \sum_{k=1}^N w_k \cdot g_k + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \frac{\partial g_k}{\partial t} \cdot \frac{-\tau_k + 1}{2} \\
& - \frac{1}{2} \sum_{i=1}^N \tilde{\mathbf{P}}_i^T \sum_{k=1}^N A_{ik} \cdot \mathbf{f}_k + \frac{(t_f - t_o)}{2} \sum_{i=1}^N \tilde{\mathbf{P}}_i^T \sum_{k=1}^N A_{ik} \cdot \frac{\partial \mathbf{f}_k}{\partial t} \cdot \frac{-\tau_k + 1}{2}.
\end{aligned} \tag{4.120}$$

The expression can be simplified by using the integral costate mapping principle (4.76), and the adjoint of the integration approximation matrix, Lemma 4.2.1, resulting in

$$\begin{aligned}
0 = & -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \left[\frac{\partial g_k}{\partial t} \right. \\
& \left. + \left(\frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ki}^\dagger \cdot \mathbf{P}_i^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_k}{\partial t} \right] \cdot \frac{-\tau_k + 1}{2} \\
& - \frac{1}{2} \sum_{k=1}^N w_k \cdot \left[g_k + \left(\frac{(t_f - t_o)}{2} \sum_{i=1}^N A_{ki}^\dagger \cdot \mathbf{P}_i^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \mathbf{f}_k \right].
\end{aligned} \tag{4.121}$$

Using the definition of the approximation to the Hamiltonian (4.108), the expression is simplified to

$$0 = -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \frac{\partial \mathcal{H}_k}{\partial t} \cdot \frac{-\tau_k + 1}{2} - \frac{1}{2} \sum_{k=1}^N w_k \cdot \mathcal{H}_k. \tag{4.122}$$

This condition for the initial time is the discrete form of the continuous relation,

$$0 = -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{(t_f - t_o)}{2} \int_{-1}^1 \frac{\partial \mathcal{H}(t)}{\partial t} \cdot \frac{-\tau + 1}{2} d\tau - \frac{1}{2} \int_{-1}^1 \mathcal{H}(t) d\tau, \tag{4.123}$$

which transformed back to the time interval $[t_o, t_f]$, noting that $\frac{-\tau + 1}{2} = \frac{t_f - t}{t_f - t_o}$, is

$$0 = -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{1}{(t_f - t_o)} \int_{t_o}^{t_f} \frac{\partial \mathcal{H}(t)}{\partial t} \cdot (t_f - t) dt - \frac{1}{(t_f - t_o)} \int_{t_o}^{t_f} \mathcal{H}(t) dt. \quad (4.124)$$

This equation is exactly equivalent to the necessary condition for the continuous problem (4.105). This relation can be shown by integration of the final terms.

The equation (4.124) can be expressed as

$$0 = -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} + \frac{1}{(t_f - t_o)} \left(\int_{t_o}^{t_f} \dot{\mathcal{H}} \cdot t_f dt + \int_{t_o}^{t_f} \dot{\mathcal{H}} \cdot t \cdot dt - \int_{t_o}^{t_f} \mathcal{H} dt \right). \quad (4.125)$$

After integration of the first term and integrating the second term by parts results in

$$0 = -\boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f} - \mathcal{H}(t_o). \quad (4.126)$$

This derivation shows that the KKT condition for the initial time (4.122) is the discrete form of the continuous necessary condition for the initial time (4.105).

The derivations of the KKT conditions with free initial and final times, indicates that these conditions are the discrete form of the continuous necessary conditions of the integral Bolza problem. This fact indicates that the pseudospectral transcription is consistent (Fig. 4-4) for problems with free time.

4.3.3 Path Constraints

Finally, consider the case involving a state and control path constraint,

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}, \quad (4.127)$$

where $\mathbf{C} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^b$. Direct formulations of the optimal control problems are well suited for solving problems with path constraints because the NLP solvers handle the constraints automatically.

For the integral pseudospectral transcription, the path constraint can be enforced

at the collocation points, so that

$$\mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, t_k) \leq \mathbf{0}, \quad k = 1, \dots, N. \quad (4.128)$$

It can be shown that the KKT conditions for the problem involving the path constraint are consistent with the continuous first-order necessary conditions. The KKT conditions for the discretized problem are found by formulating the Lagrangian with the added path constraint, so that

$$\begin{aligned} J_a = & \Phi + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot g_k - \boldsymbol{\nu}^T \cdot \boldsymbol{\phi} - \sum_{k=1}^N \tilde{\boldsymbol{\mu}}_k^T \cdot \mathbf{C}_k \\ & - \sum_{i=1}^N \tilde{\mathbf{P}}_i^T \cdot \left(\mathbf{X}_i - \mathbf{X}(t_o) - \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}_k \right), \end{aligned} \quad (4.129)$$

where $\tilde{\boldsymbol{\mu}}_k \in \mathbb{R}^b$ are the multipliers associated with the path constraint $\mathbf{C}_k = \mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, t_k)$, $k = 1, \dots, N$. It can be shown that the KKT conditions resulting from the augmented cost function (4.129) are the same as the conditions from (4.77), with the exception of the additional terms on the costate and control equations,

$$\begin{aligned} \mathbf{P}_i^T = & \frac{\partial g}{\partial \mathbf{X}_i} - \frac{2}{(t_f - t_o) \cdot w_i} \tilde{\boldsymbol{\mu}}_i^T \cdot \frac{\partial \mathbf{C}}{\partial \mathbf{X}_i} \\ & + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}_i}, \\ \mathbf{0} = & \frac{\partial g}{\partial \mathbf{U}_i} - \frac{2}{(t_f - t_o) \cdot w_i} \tilde{\boldsymbol{\mu}}_i^T \cdot \frac{\partial \mathbf{C}}{\partial \mathbf{U}_i} \\ & + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{U}_i}. \end{aligned} \quad (4.130)$$

By defining the estimate of the continuous Lagrange multiplier associated with the continuous path constraint (4.127), $\boldsymbol{\mu}(t) \in \mathbb{R}^b$, as

$$\boldsymbol{\mu}_i = \frac{2}{(t_f - t_o) \cdot w_i} \tilde{\boldsymbol{\mu}}_i^T, \quad (4.131)$$

the KKT conditions resulting from the Lagrangian (4.129) are exactly equal to the

discretized first-order necessary conditions for the optimal control problem with a path constraint [10].

It has been shown that the Karush-Kuhn-Tucker conditions for the nonlinear program, found from discretizing the integral Bolza problem, are consistent with the continuous first-order necessary conditions, Fig. 4-4. The consistency of the Gauss pseudospectral method has also been shown for many special cases including, problems requiring Pontryagin's maximum principle, free time problems, and problems with state and control path constraints. The consistency of the KKT conditions is not true for other pseudospectral methods, Fig. 4-3.

4.4 Costate Estimates

For many problems is it advantageous to have estimates for the costate from the differential problem, $\lambda(t)$. In the previous section the integral costate mapping principle (4.76), showed how to get an estimate for the costate from the integral problem, $\mathbf{p}(t)$, directly from the KKT multipliers of the NLP. It will be shown that estimates for the differential costate can also be found directly from the NLP of the integral form of the problem. Also an estimate of the initial costate, $\lambda(t_o)$, can be found from the KKT multipliers.

Theorem 4.4.1. *The costates of the differential Bolza problem can be found from the KKT multipliers of the pseudospectral transcription of the integral Bolza problem, so that*

$$\Lambda_i^T = \sum_{k=1}^N A_{ki} \cdot \frac{\tilde{\mathbf{P}}_k^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \quad (4.132)$$

Proof. The estimate for the differential costate can be found by using the integral costate mapping principle, along with the relation between the differential and integral costates.

It has been shown in Section 4.3 that an estimate for the integral costate, $\mathbf{p}(t) \in \mathbb{R}^n$, can be found directly from the KKT multipliers using the integral costate mapping

principle, Theorem 4.3.1, restated here as

$$\mathbf{p}(t_i) \approx \mathbf{P}_i = \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\mathbf{P}}_i}{w_i}. \quad (4.133)$$

It has also been shown in Section 4.1 that the integral costate is related to the differential costate, $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$, Theorem 4.1.1, so that

$$\boldsymbol{\lambda}^T(t) = \int_t^{t_f} \mathbf{p}^T(\tau) \cdot d\tau + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{x}(t_f)}. \quad (4.134)$$

The pseudospectral approximation of this relation can be expressed using the adjoint of the integration approximation matrix, Lemma 4.2.1, so that

$$\boldsymbol{\lambda}^T(t_i) \approx \boldsymbol{\Lambda}_i^T = \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \quad (4.135)$$

Combining (4.133) and (4.135) allows for the estimation of the differential costate directly from the KKT multipliers of the pseudospectral approximation in integral form. The expression can be simplified to

$$\begin{aligned} \boldsymbol{\Lambda}_i^T &= \sum_{k=1}^N A_{ik}^\dagger \cdot \frac{\tilde{\mathbf{P}}_k^T}{w_k} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}, \\ &= \sum_{k=1}^N A_{ki} \cdot \frac{\tilde{\mathbf{P}}_k^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \end{aligned} \quad (4.136)$$

This derivation shows that the differential costate can be estimated from the KKT multipliers of the integral pseudospectral method. \square

Using this estimate for the differential costate (4.136), it can be shown that the pseudospectral approximation to the Hamiltonian is

$$\mathcal{H}(t_i) = g_i + \left(\frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik}^\dagger \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \mathbf{f}_i, \quad (4.137)$$

and in terms of the KKT multipliers,

$$\mathcal{H}(t_i) = g_i + \left(\sum_{k=1}^N A_{ki} \cdot \frac{\tilde{\mathbf{P}}_k^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \mathbf{f}_i. \quad (4.138)$$

4.4.1 Initial Costate

The initial differential costate, $\boldsymbol{\lambda}(t_o)$, can also be determined directly from the KKT multipliers of the NLP.

Theorem 4.4.2. *The initial differential costate, $\boldsymbol{\lambda}(t_o)$, can be found in terms of the KKT multipliers of the integral Bolza problem, so that*

$$\boldsymbol{\Lambda}^T(t_o) = \sum_{k=1}^N \tilde{\mathbf{P}}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \quad (4.139)$$

Proof. The initial costate in terms of the continuous integral costate is

$$\boldsymbol{\lambda}^T(t_o) = \int_{t_o}^{t_f} \mathbf{p}^T(t) \cdot dt + \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{x}(t_f)}. \quad (4.140)$$

The pseudospectral approximation to $\boldsymbol{\Lambda}(t_o)$ is found using a Gauss quadrature, so that

$$\boldsymbol{\Lambda}^T(t_o) = \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}, \quad (4.141)$$

which can be expressed in terms of the KKT multipliers. The result is

$$\boldsymbol{\Lambda}^T(t_o) = \sum_{k=1}^N \tilde{\mathbf{P}}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \quad (4.142)$$

□

The estimate for the initial costate can also be found using the KKT condition for the initial state (4.85), which is

$$\frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{P}_k^T + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} = \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)}. \quad (4.143)$$

Therefore, using the equation (4.141) with the initial state KKT condition, the initial

costate estimate is equivalent to

$$\Lambda^T(t_o) = \nu^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)}. \quad (4.144)$$

This equation shows that the estimate for the initial differential costate can be found directly from the KKT multipliers of the NLP by using either (4.139) or (4.144).

Empirical evidence has suggested the the initial costate estimate is more accurate than the intermediate costate estimates. This fact is consistent with the results reported by Axelsson [2] for the boundary points of ODEs.

In this section, it was shown that an estimate for the differential costate, $\lambda(t)$, along with an estimate for the initial costate, $\lambda(t_o)$, can be found directly from the KKT multipliers of the NLP from the discretized integral Bolza problem. Information about the differential costate can be used for optimality verification, mesh refinement, sensitivity analysis, as well as real time control applications.

4.5 Multiple Phases

The Gauss pseudospectral method is well suited for solving optimal control problems with multiple phases. Multiple phase problems can arise if the problem has distinctive differences on two or more sections of the solution. These differences could be different dynamic relations on one segment or another, or a different cost function to be minimized on different segments. A multiple phase problem can be used to model a specific event, such as an intermediate point constraint, or a state discontinuity such as a mass drop. Another use for a multiple phase formulation is to model artificial events such as the entry or exit from a path constraint. This formulation can be particularly useful for pseudospectral approximations where a potentially discontinuous solution can be approximated using piecewise polynomials rather than a global polynomial. Linking phases of an optimal control problems is considered in [7] and [57].

4.5.1 Implementation

The pseudospectral implementation of a multiple phase problem is treated by defining each phase separately and then linking the phases with a set of phase boundary conditions. The problem defined with R distinct phases, has a cost on phase $r \in [1, \dots, R]$ of

$$J^{(r)} = \Phi^{(r)}(\mathbf{X}^{(r)}(t_o^{(r)}), t_o^{(r)}, \mathbf{X}^{(r)}(t_f^{(r)}), t_f^{(r)}) + \frac{(t_f^{(r)} - t_o^{(r)})}{2} \sum_{k=1}^{N(r)} w_k^{(r)} \cdot g^{(r)}(\mathbf{X}_k^{(r)}, \mathbf{U}_k^{(r)}, t_k^{(r)}), \quad (4.145)$$

and constraints

$$\mathbf{X}_i^{(r)} = \mathbf{X}^{(r)}(t_o^{(r)}) + \frac{(t_f^{(r)} - t_o^{(r)})}{2} \sum_{k=1}^{N(r)} A_{ik}^{(r)} \cdot \mathbf{f}^{(r)}(\mathbf{X}_k^{(r)}, \mathbf{U}_k^{(r)}, t_k^{(r)}), \quad i = 1, \dots, N(r), \quad (4.146)$$

where $\mathbf{X}_k^{(r)} \in \mathbb{R}^{n(r)}$, $\mathbf{U}_k^{(r)} \in \mathbb{R}^{m(r)}$ is the approximate state and control at the Gauss points on phase r , $k = 1, \dots, N(r)$. The cost function $g^{(r)} : \mathbb{R}^{n(r)} \times \mathbb{R}^{m(r)} \times \mathbb{R} \rightarrow \mathbb{R}$ and state dynamics function $\mathbf{f}^{(r)} : \mathbb{R}^{n(r)} \times \mathbb{R}^{m(r)} \times \mathbb{R} \rightarrow \mathbb{R}^{n(r)}$ may be different (and have different dimensions) on each phase. Note that the initial and final times of the phase are $t_o^{(r)}$ and $t_f^{(r)}$, and the initial and final states are $\mathbf{X}^{(r)}(t_o^{(r)})$ and $\mathbf{X}^{(r)}(t_f^{(r)})$. As in the single phase formulation (see Section 4.2.3), the final state must be related to the initial state on each phase using the relation

$$\mathbf{X}^{(r)}(t_f^{(r)}) = \mathbf{X}^{(r)}(t_o^{(r)}) + \frac{(t_f^{(r)} - t_o^{(r)})}{2} \sum_{k=1}^{N(r)} w_k^{(r)} \cdot \mathbf{f}^{(r)}(\mathbf{X}_k^{(r)}, \mathbf{U}_k^{(r)}, t_k^{(r)}). \quad (4.147)$$

Finally, the phase boundary constraints must be defined to link the states and times between phase r and phase $r + 1$, so that

$$\mathcal{L}_{(r)}^{(r+1)}(\mathbf{X}^{(r)}(t_f^{(r)}), t_f^{(r)}, \mathbf{X}^{(r+1)}(t_o^{(r+1)}), t_o^{(r+1)}) = \mathbf{0}. \quad (4.148)$$

In most cases, the initial time of phase $r + 1$ is equal to the final time of phase r ,

so that one of the phase boundary constraints is

$$t_f^{(r)} - t_o^{(r+1)} = 0. \quad (4.149)$$

If the states are continuous across the phase boundary the phase boundary constraint is defined as

$$\mathbf{X}(t_f^{(r)}) - \mathbf{X}(t_o^{(r+1)}) = \mathbf{0}. \quad (4.150)$$

However, the general phase boundary constraint (4.148) allows for discontinuities or even different numbers of states on each side of the phase boundary, such as if different coordinate frames are used on different phases.

This implementation shows that the multiple phase optimal control problem can be discretized into an NLP where the cost is the sum of the costs on each phase (4.145), so that

$$J = \sum_{r=1}^R J^{(r)}. \quad (4.151)$$

The dynamic constraints are approximated on each phase (4.146), $r = 1, \dots, R$, along with the conditions between the phases (4.148), $r = 1, \dots, R - 1$. The last constraints of the NLP are the boundary constraints of the problem, which are the boundary constraints at the beginning of the first phase and the end of the final phase, so that

$$\begin{aligned} \mathcal{L}_{(0)}^{(1)}(\mathbf{X}^{(1)}(t_o^{(1)}), t_o^{(1)}) &= \mathbf{0}, \\ \mathcal{L}_{(R)}^{(R+1)}(\mathbf{X}^{(R)}(t_f^{(R)}), t_f^{(R)}) &= \mathbf{0}. \end{aligned} \quad (4.152)$$

The NLP defined by the cost (4.151), and constraints (4.146, 4.148, 4.152), results in a pseudospectral transcription of the multiple phase problem in integral form. The method will retain the consistency property of the single phase problem, allowing for an accurate solution to the optimal control problem. The consistency of the method will also allow for accurate costate estimates for the multiple phase problem.

4.5.2 Costate Estimate

The costate estimate for the multiple phase problem is generated in the same way as the single phase problem (see Section 4.4). The costate is found from the KKT multipliers of the dynamic constraints along with the Lagrange multipliers of the boundary constraints. The costate estimates on each phase are

$$\begin{aligned} \left(\Lambda_i^{(r)}\right)^T &= \sum_{k=1}^{N(r)} A_{ki}^{(r)} \cdot \frac{\left(\tilde{\mathbf{P}}_k^{(r)}\right)^T}{w_i^{(r)}} + \frac{\partial\Phi^{(r)}}{\partial\mathbf{X}^{(r)}(t_f^{(r)})} - \left(\boldsymbol{\nu}^{(r)}\right)^T \cdot \frac{\partial\mathcal{L}_{(r)}^{(r+1)}}{\partial\mathbf{X}^{(r)}(t_f^{(r)})}, \\ &i = 1, \dots, N(r). \end{aligned} \quad (4.153)$$

The initial and final costate estimates on each phase are

$$\begin{aligned} \left(\Lambda^{(r)}(t_o^{(r)})\right)^T &= -\frac{\partial\Phi^{(r)}}{\partial\mathbf{X}^{(r)}(t_o^{(r)})} + \left(\boldsymbol{\nu}^{(r-1)}\right)^T \cdot \frac{\partial\mathcal{L}_{(r-1)}^{(r)}}{\partial\mathbf{X}^{(r)}(t_o^{(r)})}, \\ \left(\Lambda^{(r)}(t_f^{(r)})\right)^T &= \frac{\partial\Phi^{(r)}}{\partial\mathbf{X}^{(r)}(t_f^{(r)})} - \left(\boldsymbol{\nu}^{(r)}\right)^T \cdot \frac{\partial\mathcal{L}_{(r)}^{(r+1)}}{\partial\mathbf{X}^{(r)}(t_f^{(r)})}, \end{aligned} \quad (4.154)$$

with the relation between the two as

$$\Lambda^{(r)}(t_o^{(r)}) = \Lambda^{(r)}(t_f^{(r)}) + \sum_{k=1}^{N(r)} \tilde{\mathbf{P}}_k^{(r)}. \quad (4.155)$$

The costate estimate across a phase boundary $\Lambda^{(r)}(t_f^{(r)})$ to $\Lambda^{(r+1)}(t_o^{(r+1)})$ may or may not be continuous depending on the phase boundary constraints (and phase boundary costs).

The Gauss pseudospectral method is well suited for solving multiple phase optimal control problems. The consistency of the method allows for accurate solutions in the states, controls, and costates.

4.6 Polynomial Approximation

The Gauss pseudospectral method returns an estimate of the states, controls, and costates, at the Gauss collocation points. Because the method is using global polynomials as trial functions, the solutions for the states, controls, and costates must be polynomials. The derivation of the KKT conditions indicate the properties of the solution polynomials. Examination of these polynomials leads to a better understanding of how the Gauss pseudospectral method works.

The pseudospectral approximation to the differential dynamics indicates that the approximation to the states, $\mathbf{X}(t) \in \mathbb{R}^n$, is a polynomial of degree N that satisfies the state dynamics at the N Gauss collocation points and satisfies the boundary conditions (Lemma 4.2.2), such that

$$\begin{aligned} \frac{d\mathbf{X}}{dt}(t_k) &= \mathbf{f}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k), \quad k = 1, \dots, N, \\ \phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) &= \mathbf{0}. \end{aligned} \quad (4.156)$$

The estimate for the controls, $\mathbf{U}(t) \in \mathbb{R}^m$, is defined only at the N Gauss points, so that $\mathbf{U}(t)$ is a polynomial of degree $N - 1$. The discrete form for the integral costate dynamics (4.82) defines the costate estimate, $\mathbf{P}(t) \in \mathbb{R}^n$, at the N Gauss points so that it is also a polynomial of degree $N - 1$. The estimate for the differential costate, $\boldsymbol{\Lambda}(t) \in \mathbb{R}^n$, is the pseudospectral integration of the integral costate (4.135). Because the integral costate is a polynomial of degree $N - 1$, the pseudospectral integration is exact. This result defines the differential costate estimate as a polynomial of degree N , so that

$$\boldsymbol{\Lambda}(t) = \int_t^{t_f} \mathbf{P}(\tau) d\tau + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \quad (4.157)$$

The exact derivative of this costate is

$$\frac{d\boldsymbol{\Lambda}(t)}{dt} = -\mathbf{P}(t). \quad (4.158)$$

Using this relation with the approximation to the costate dynamics (4.82) shows

that the differential costate estimate satisfies the costate dynamics at the collocation points,

$$\frac{d\Lambda^T}{dt}(t_k) = -\frac{\partial g}{\partial \mathbf{X}}(t_k) - \Lambda^T(t_k) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(t_k), \quad k = 1, \dots, N. \quad (4.159)$$

Thus the polynomial approximation of the states, controls, and costates, satisfy the equations

$$\begin{aligned} \frac{d\mathbf{X}}{dt}(t_k) &= \mathbf{f}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k), \\ \frac{d\Lambda^T}{dt}(t_k) &= -\frac{\partial g}{\partial \mathbf{X}}(t_k) - \Lambda^T(t_k) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(t_k), \\ \mathbf{U}(t_k) &= \arg \min_{\mathbf{U}(t_k) \in \mathcal{U}} [\mathcal{H}(\mathbf{X}(t_k), \mathbf{U}(t_k), \Lambda(t_k), t_k)], \\ k &= 1, \dots, N, \\ \mathbf{X}(t_f) &= \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{i=1}^N w_i \cdot \mathbf{f}(\mathbf{X}(t_i), \mathbf{U}(t_i), t_i), \\ \phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) &= \mathbf{0}, \\ \Lambda^T(t_o) &= \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)}, \\ \Lambda^T(t_f) &= \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}, \\ \Lambda^T(t_f) &= \Lambda^T(t_o) - \frac{(t_f - t_o)}{2} \sum_{i=1}^N w_i \cdot \left(\frac{\partial g}{\partial \mathbf{X}}(t_i) + \Lambda^T(t_i) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(t_i) \right). \end{aligned} \quad (4.160)$$

These equations indicate that solving the NLP generated from the pseudospectral transcription of the optimal control problem in integral form is equivalent to finding polynomials of degree N for the states, $\mathbf{X}(t)$, and costates, $\Lambda(t)$, along with a polynomial of degree $N - 1$ for the controls, $\mathbf{U}(t)$, that satisfy the state and costate boundary conditions, and the state dynamics, costate dynamics, and Pontryagin's maximum principle at the N Gauss collocation points.

4.7 Summary

In this chapter, the Gauss pseudospectral transcription has been outlined and analyzed. The integral form of the continuous optimal control problem has been defined and shown to have equivalent first-order optimality conditions as the differential form of the problem. This equivalence indicates that a pseudospectral transcription can be defined in terms of the integral form of the continuous optimal control problem.

The discretization using the integration approximation matrix was introduced, which allows the optimal control problem to be transcribed into a nonlinear program (NLP), which then can be solved by well-developed algorithms. Unlike other pseudospectral methods that collocate at the boundary points, the Gauss pseudospectral method is collocating only at the Gauss points, which are all interior. The result of this difference is that the Karush-Kuhn-Tucker (KKT) conditions of the NLP from the Gauss method are exactly equal to the discretized first-order optimality conditions. This relation indicates that solving the optimal control problem using the Gauss pseudospectral method is consistent with the continuous optimal control problem, and the direct and indirect formulations are the same.

The Gauss pseudospectral method has been shown to be consistent for a large class of problems, including free time, multiple phase, path constraints, and those requiring Pontryagin's maximum principle. Finally, the relationship between the approximation polynomials was derived. This relationship indicates that the Gauss pseudospectral method is equivalent to finding the set of polynomials for the states, costates, and controls, that satisfy the state and costate dynamics and Pontryagin's maximum principle at the collocation points, along with the state and costate boundary conditions. This relation between the approximating polynomials does not hold for other pseudospectral methods. The approximating polynomials for the states, costates, and controls indicate that an equivalent discretization of the optimal control problem in differential form can be made at Gauss points. The derivation of this discretization is shown in Chapter 5.

[This page intentionally left blank.]

Chapter 5

Differential Gauss Pseudospectral Method

It has been shown (see Section 4.2.2) that the pseudospectral approximation of an ODE in integral form is equivalent to finding the polynomial of degree N that satisfies the differential equation at the N Gauss collocation points and satisfies the initial condition. It has also been shown (see Section 4.6) that the pseudospectral approximation of an optimal control problem with dynamics in integral form is equivalent to finding polynomials of degree N that satisfy the state and costate dynamic equations at the N Gauss collocation points, along with the boundary constraints. A pseudospectral transcription can be devised directly from the differential form of an ODE (or optimal control problem) that is exactly equivalent to the pseudospectral transcription of the integral form.

This differential approximation is different from the Legendre pseudospectral approximation in several ways. The first is that the differential equation is not collocated at the boundary points. The second is that the approximating polynomial, $\mathbf{X}(t)$, is of degree N , so that $N + 1$ points are required to determine its derivative. In the Legendre pseudospectral method, N points are used to determine the derivative of the approximating polynomial of degree $N - 1$.

5.1 Discretization

A pseudospectral approximation of the differential form of the continuous optimal control problem is made by transcribing the continuous optimal control problem into a nonlinear program. This transcription is done by approximating the states, $\mathbf{x}(t)$, with polynomials, $\mathbf{X}(t) \in \mathbb{R}^n$, of degree N , formed from a basis of $N + 1$ Lagrange interpolating polynomials [17] on the interval from $[-1, 1]$. These polynomials are

$$\mathbf{x}(t) \approx \mathbf{X}(t) = \sum_{k=0}^N \mathbf{x}(t_k) \cdot L_k(t) . \quad (5.1)$$

The interpolation points used are the boundary point, -1 , along with the N Gauss points, t_k , $k = 1, \dots, N$, which are all in the interior of the interval $[-1, 1]$.

Definition 5.1.1. *The differential approximation matrices $D \in \mathbb{R}^{N \times N}$ and $\bar{D} \in \mathbb{R}^N$, are found using the exact derivative of the Lagrange interpolating polynomials, $L_k(t)$, so that*

$$\dot{\mathbf{x}}(t_i) \approx \dot{\mathbf{X}}(t_i) = \mathbf{x}(t_o) \cdot \bar{D}_i + \sum_{k=1}^N \mathbf{x}(t_k) \cdot D_{ik} , \quad (5.2)$$

where $D_{ik} = \dot{L}_k(t_i)$ and $\bar{D}_i = \dot{L}_0(t_i)$ are the differential approximation matrices.

Theorem 5.1.1. *The approximate derivative found using the differential approximation matrices is exact for polynomials of degree N or less.*

Proof. The differential approximation matrices finds the approximate derivative by using the exact derivative of an interpolating polynomial at $N + 1$ distinct points. The Lagrange interpolating polynomial is exact for polynomials of degree N or less [17] and therefore, the differential operator is exact for polynomials of degree N or less. \square

In the remainder of this section, several properties of the differential approximation matrices are derived. These properties are important in the analysis of the transcription method.

Lemma 5.1.1. *The differential approximation matrix, D , is related to \bar{D} by the relation*

$$\bar{D}_i = - \sum_{k=1}^N D_{ik} . \quad (5.3)$$

Proof. The relationship between the differential approximation matrix, D , and \bar{D} , can be shown by examining the derivative of a constant function, $f(t) = c$. Applying the differential differential approximation matrices to find the derivative results in

$$c \cdot \bar{D}_i + c \cdot \sum_{k=1}^N D_{ik} = \dot{f}(t_i) = 0 .$$

This relation is exact for all values of N , because the function $f(t) = c$ is a polynomial of degree 0. The result can be simplified to (5.3). \square

A pseudospectral approximation can also be made using a slightly different basis of Lagrange polynomials. The $N + 1$ points used in this case are the N Gauss points along with the final point t_f , so that

$$\mathbf{x}(t) \approx \mathbf{X}(t) = \sum_{k=1}^{N+1} \mathbf{x}(t_k) \cdot L_k^\dagger(t) . \quad (5.4)$$

Another differential approximation matrix can be found by differentiation.

Definition 5.1.2. *The differential approximation matrices, $D^\dagger \in \mathbb{R}^{N \times N}$ and $\bar{D}^\dagger \in \mathbb{R}^N$, are found using the exact derivative of the Lagrange interpolating polynomials, $L_k^\dagger(t)$. These matrices are*

$$\dot{\mathbf{x}}(t_i) \approx \dot{\mathbf{X}}(t_i) = \sum_{k=1}^N \mathbf{x}(t_k) \cdot D_{ik}^\dagger + \mathbf{x}(t_f) \cdot \bar{D}_i^\dagger , \quad (5.5)$$

where $D_{ik}^\dagger = \dot{L}_k^\dagger(t_i)$ and $\bar{D}_i^\dagger = \dot{L}_{N+1}^\dagger(t_i)$ are the differential approximation matrices.

Lemma 5.1.2. *The relation between the differential operator, D^\dagger , and \bar{D}^\dagger is*

$$\bar{D}_i^\dagger = - \sum_{k=1}^N D_{ik}^\dagger . \quad (5.6)$$

Proof. The relation can be found by examining the derivative of a constant function (see Lemma 5.1.1). \square

Lemma 5.1.3. *The relation between the differential approximation matrix D and its adjoint D^\dagger is*

$$D_{ik}^\dagger = -\frac{w_k}{w_i} \cdot D_{ki}. \quad (5.7)$$

Proof. Consider the integration by parts formula for two polynomials of degree N , $f(t), g(t)$, so that

$$\int_{-1}^1 \dot{f}(t) \cdot g(t) dt = f(t) \cdot g(t) \Big|_{-1}^1 - \int_{-1}^1 f(t) \cdot \dot{g}(t) dt.$$

The polynomials $\dot{f}(t) \cdot g(t)$ and $f(t) \cdot \dot{g}(t)$ are of degree $2N - 1$, so that the integrals can be replaced exactly by a Gauss quadrature [16]. The derivatives of the polynomials at the Gauss points can be found exactly using the differential approximation (5.2) for $\dot{f}(t)$ and (5.5) for $\dot{g}(t)$. The integration by parts formula is replaced exactly by

$$\begin{aligned} \sum_{k=1}^N \left[f(-1) \cdot \bar{D}_k + \sum_{i=1}^N D_{ki} \cdot f(t_i) \right] g(t_k) \cdot w_k = \\ f(t) \cdot g(t) \Big|_{-1}^1 - \sum_{k=1}^N f(t_k) \left[g(1) \cdot \bar{D}_k^\dagger + \sum_{i=1}^N D_{ki}^\dagger \cdot g(t_i) \right] \cdot w_k, \end{aligned} \quad (5.8)$$

where w_k are the Gauss weights. Because this must be true for all polynomials, it must be true for the set of Lagrange interpolating polynomials generated from the points -1 plus the N Gauss points, $f(t) = L_l(t)$, $l = 0, \dots, N$ and the polynomials generated from the N Gauss points plus the end point 1, $g(t) = L_j^\dagger(t)$, $j = 1, \dots, N + 1$. These are the same Lagrange polynomials used to generate the differential approximations (5.2) and (5.5). If the Lagrange polynomials corresponding to $l = 1, \dots, N$ and $j = 1, \dots, N$ are used in (5.8), the expression is simplified because the polynomials L_l are zero at the points -1 and all but the l th Gauss point, and the polynomials L_j^\dagger are zero at 1 and all but the j th Gauss point. The non-zero terms result in the relation,

$$D_{jl} \cdot w_j = -D_{lj}^\dagger \cdot w_l. \quad \square$$

Note that the differential approximation matrices used here are not the same as the approximation matrix from the Legendre pseudospectral method (see Section 3.3.1).

5.2 Differential - Integral Relation

The differential approximation matrix is related to the integral approximation matrix (see Section 4.2.2). This relationship can be seen by looking at the discretized form of an ordinary differential equation (ODE).

The pseudospectral approximation of a differential equation in integral form is

$$\mathbf{X}(t_i) = \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N A_{ik} \cdot \mathbf{f}(\mathbf{X}(t_k), t_k), \quad i = 1, \dots, N, \quad (5.9)$$

with initial condition $\mathbf{X}(t_o) = \mathbf{X}_o$, which defines a polynomial of degree N , $\mathbf{X}(t) \in \mathbb{R}^n$. This polynomial satisfies the initial condition and the differential equation at the N collocation points, so that

$$\begin{aligned} \frac{d\mathbf{X}}{dt}(t_i) &= \mathbf{f}(\mathbf{X}(t_i), t_i), \quad i = 1, \dots, N, \\ \mathbf{X}(t_o) &= \mathbf{X}_o. \end{aligned} \quad (5.10)$$

Note that the collocation points are at the Gauss points, so the differential equation is not satisfied at the initial time, t_o .

The discretization of the ODE using the differential approximation matrices results in

$$\begin{aligned} \frac{2}{(t_f - t_o)} \cdot \mathbf{X}(t_o) \cdot \bar{D}_i + \frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik} \cdot \mathbf{X}(t_k) &= \mathbf{f}(\mathbf{X}(t_i), t_i), \quad i = 1, \dots, N, \\ \mathbf{X}(t_o) &= \mathbf{X}_o. \end{aligned} \quad (5.11)$$

The fraction, $\frac{2}{(t_f - t_o)}$, has been included to account for the transformation of the independent time variable from $t \in [t_o, t_f]$ to $\tau \in [-1, 1]$ (see Section 4.2.1). Note that

the differential approximation uses $N + 1$ points of the function $\mathbf{X}(t)$ to determine the derivative at the N collocation points. The pseudospectral approximation of the differential equation using the differential approximation matrices, also satisfies (5.10).

Lemma 5.2.1. *The differential and integration approximation matrices are related by*

$$D = A^{-1}. \quad (5.12)$$

Proof. The relation between the integration approximation matrix, A , and the differential approximation matrix, D , can be more easily seen by looking at a one dimensional initial value problem on the interval from -1 to 1 , so that

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), t), \\ x(-1) &= x_o. \end{aligned} \quad (5.13)$$

The pseudospectral approximation in integral form can be written

$$X = X(-1) + A \cdot F, \quad X(-1) = x_o, \quad (5.14)$$

where $X \in \mathbb{R}^N$ is the approximation to $x(t)$ at the N Gauss points, $X(-1) \in \mathbb{R}$ is the approximation to $x(t)$ at -1 , and $F \in \mathbb{R}^N$ is the function $f(x(t), t)$ evaluated at the N Gauss points. The pseudospectral approximation in differential form can be written

$$\bar{D} \cdot X(-1) + D \cdot X = F, \quad X(-1) = x_o. \quad (5.15)$$

The differential equation and boundary condition can be expressed together as

$$\begin{bmatrix} 1 & \mathbf{0} \\ \bar{D} & D \end{bmatrix} \begin{bmatrix} X(-1) \\ X \end{bmatrix} = \begin{bmatrix} x_o \\ F \end{bmatrix}. \quad (5.16)$$

The integral form (5.14) can be expressed in a similar manner as

$$\begin{bmatrix} X(-1) \\ X \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{1} & A \end{bmatrix} \begin{bmatrix} x_o \\ F \end{bmatrix}. \quad (5.17)$$

For both of these equations to be true, the matrices must be inverses of each other, so that

$$\begin{bmatrix} 1 & \mathbf{0} \\ \bar{D} & D \end{bmatrix} \cdot \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{1} & A \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}. \quad (5.18)$$

Note that the integration approximation matrix is non-singular. From this relation it follows that

$$\begin{aligned} A^{-1} &= D, \\ D \cdot \mathbf{1} &= -\bar{D}. \end{aligned} \quad (5.19)$$

Using these relations (5.19), the differential and integral pseudospectral approximations of the ODE can be shown to be the same. Multiplying the integral form (5.14) by the inverse of the integration approximation matrix results in

$$\begin{aligned} A^{-1} \cdot X &= A^{-1} \cdot \mathbf{1} \cdot x_o + A^{-1} \cdot A \cdot F, \\ D \cdot X &= -\bar{D} \cdot x_o + F, \\ \bar{D} \cdot x_o + D \cdot X &= F, \end{aligned} \quad (5.20)$$

which is equivalent to the differential form (5.15). \square

This derivation indicates that the differential form of the pseudospectral approximation can be found from the integral form by simply inverting the integration approximation matrix.

5.3 Optimal Control Problem

The differential Gauss pseudospectral method is a pseudospectral transcription of the differential form of the continuous optimal control problem at Gauss collocation

points. The NLP is formed by approximating the cost using a Gauss quadrature, so that

$$J = \Phi(\mathbf{X}(t_f), t_f) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot g(\mathbf{X}_k, \mathbf{U}_k, t_k). \quad (5.21)$$

The differential dynamic constraints are approximated using the differential approximation matrices, which are collocating the constraints at the Gauss points, so that

$$\frac{2}{(t_f - t_o)} \bar{D}_i \cdot \mathbf{X}(t_o) + \frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik} \cdot \mathbf{X}_k = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k), \quad i = 1, \dots, N. \quad (5.22)$$

The boundary constraints are expressed in their most general form as

$$\phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) = \mathbf{0}. \quad (5.23)$$

Finally, the terminal states, $\mathbf{X}(t_f)$, are defined in terms of the quadrature approximation to the dynamics. This relation is required to enforce boundary constraints at the final time. The final states are

$$\mathbf{X}(t_f) = \mathbf{X}(t_o) + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k). \quad (5.24)$$

The cost (5.21) and constraints (5.22 - 5.23) define an NLP, whose solution approximates the solution to the continuous optimal control problem.

5.4 KKT Conditions

The differential Gauss pseudospectral method can be shown to be consistent with the continuous first-order optimality conditions of the optimal control problem by looking at the KKT conditions.

Theorem 5.4.1. *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. Furthermore, a costate estimate can be found from the KKT multipliers, $\bar{\Lambda}_k$, and Lagrange multipliers, ν , that satisfy the pseudospectral approximation*

to the costate dynamics, so that

$$\Lambda_k^T = \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_k^T}{w_k} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \quad (5.25)$$

Proof. The Karush-Kuhn-Tucker (KKT) conditions of the NLP can be found from the augmented cost function, or Lagrangian [4]. The augmented cost is formed using the multipliers $\tilde{\Lambda}_i \in \mathbb{R}^n$, $i = 1, \dots, N$, and $\boldsymbol{\nu} \in \mathbb{R}^q$, so that

$$\begin{aligned} J_a = & \Phi + \frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot g_k - \boldsymbol{\nu}^T \cdot \phi \\ & - \sum_{i=1}^N \tilde{\Lambda}_i^T \cdot \left(\frac{2}{(t_f - t_o)} \bar{D}_i \cdot \mathbf{X}(t_o) + \frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik} \cdot \mathbf{X}_k - \mathbf{f}_i \right), \end{aligned} \quad (5.26)$$

where $g_k = g(\mathbf{X}_k, \mathbf{U}_k, t_k)$ and $\mathbf{f}_i = \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, t_i)$. The KKT conditions are found by setting the gradient of the Lagrangian to zero.

The partials with respect to the states are

$$\begin{aligned} \mathbf{0} = \frac{\partial J_a}{\partial \mathbf{X}_i} = & -\frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ki} \cdot \tilde{\Lambda}_k^T + \frac{(t_f - t_o)}{2} \cdot w_i \cdot \frac{\partial g_i}{\partial \mathbf{x}} \\ & + \left(\tilde{\Lambda}_i^T + \frac{(t_f - t_o)}{2} \cdot w_i \cdot \left[\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right] \right) \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}}. \end{aligned} \quad (5.27)$$

Note that the chain rule was used because the final states, $\mathbf{X}(t_f)$, are defined in terms of the interior points (5.24). Dividing (5.27) through by $\frac{2 \cdot w_i}{(t_f - t_o)}$, and using the adjoint of the differential approximation matrix (Lemma 5.1.3) results in

$$\begin{aligned} & -\frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik}^\dagger \cdot \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_k^T}{w_k} = \\ & \frac{\partial g_i}{\partial \mathbf{x}} + \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_i^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}}. \end{aligned} \quad (5.28)$$

The term

$$\frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik}^\dagger \cdot \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \quad (5.29)$$

can be added and subtracted to the left side of the equation resulting in

$$\begin{aligned}
& -\frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik}^\dagger \cdot \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_k^T}{w_k} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \\
& \quad + \frac{2}{(t_f - t_o)} \cdot \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \sum_{k=1}^N D_{ik}^\dagger \\
& = \frac{\partial g_i}{\partial \mathbf{x}} + \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_i^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}}.
\end{aligned} \tag{5.30}$$

Finally, this can be simplified using Lemma 5.1.2, resulting in

$$\begin{aligned}
& \frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik}^\dagger \cdot \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_k^T}{w_k} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \\
& \quad + \frac{2}{(t_f - t_o)} \cdot \left(\frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \bar{D}_i^\dagger \\
& = -\frac{\partial g_i}{\partial \mathbf{x}} - \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_i^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}}.
\end{aligned} \tag{5.31}$$

The right hand side of (5.31) is the approximation to the right hand side of the costate dynamics (2.59), and the left hand side is the approximate derivative of the costate. This result defines a costate mapping from the KKT multipliers, $\tilde{\Lambda}_k$, to the estimates for the costate, Λ_k , so that

$$\Lambda_k^T = \frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_k^T}{w_k} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}. \tag{5.32}$$

The costate mapping (5.32), along with the estimate for the final costate,

$$\Lambda^T(t_f) = \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}, \tag{5.33}$$

allows (5.31) to be expressed as

$$\frac{2}{(t_f - t_o)} \sum_{k=1}^N D_{ik}^\dagger \cdot \Lambda_k^T + \frac{2}{(t_f - t_o)} \cdot \Lambda^T(t_f) \cdot \bar{D}_i^\dagger = -\frac{\partial g_i}{\partial \mathbf{x}} - \Lambda_i^T \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}}, \tag{5.34}$$

which is the discrete form of the continuous costate dynamics (2.59).

The partial of the Lagrangian (5.26) with respect to the control can be simplified in the same manner to

$$\mathbf{0} = \frac{\partial g_i}{\partial \mathbf{u}} + \left(\frac{2}{(t_f - t_o)} \cdot \frac{\tilde{\Lambda}_i^T}{w_i} + \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} \right) \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}}. \quad (5.35)$$

Using the costate mapping (5.32), this becomes

$$\mathbf{0} = \frac{\partial g_i}{\partial \mathbf{u}} + \Lambda_i^T \cdot \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}}, \quad (5.36)$$

which is the discrete form of the continuous control equation (2.59).

Finally, the last KKT condition is found from the partial of the Lagrangian (5.26) with respect to the initial state. The partial is

$$\frac{\partial J_a}{\partial \mathbf{X}(t_o)} = \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)} - \frac{2}{(t_f - t_o)} \sum_{i=1}^N \tilde{\Lambda}_i^T \cdot \bar{D}_i = \mathbf{0}, \quad (5.37)$$

where the last term is the Gauss quadrature of the approximation to the derivative of the costate. This result can be shown by using the approximate derivative of the costate from (5.28). The Gauss quadrature of the derivative is

$$\frac{(t_f - t_o)}{2} \sum_{k=1}^N w_k \cdot \dot{\Lambda}_k^T = \frac{2}{(t_f - t_o)} \sum_{k=1}^N w_k \cdot \left(\sum_{i=1}^N D_{ki}^\dagger \cdot \frac{\tilde{\Lambda}_i^T}{w_i} \right). \quad (5.38)$$

The definition of the adjoint is used, Lemma 5.1.3, resulting in

$$\begin{aligned} &= \frac{2}{(t_f - t_o)} \sum_{k=1}^N w_k \cdot \left(\sum_{i=1}^N -D_{ik} \cdot \frac{\tilde{\Lambda}_i^T}{w_k} \right) \\ &= \frac{2}{(t_f - t_o)} \sum_{i=1}^N \tilde{\Lambda}_i^T \cdot \sum_{k=1}^N -D_{ik}, \end{aligned} \quad (5.39)$$

which can be simplified by the definition of \bar{D}_i , Lemma 5.1.1. The result is

$$\frac{2}{(t_f - t_o)} \sum_{i=1}^N \tilde{\Lambda}_i^T \cdot \bar{D}_i = \frac{(t_f - t_o)}{2} \sum_{i=1}^N w_i \cdot \dot{\Lambda}_i^T. \quad (5.40)$$

This derivation indicates that (5.37) is the relation between the initial and terminal costates, so that

$$\begin{aligned} \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)} &= \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)} + \frac{(t_f - t_o)}{2} \sum_{i=1}^N w_i \cdot \dot{\Lambda}_i^T, \\ \Lambda^T(t_f) &= \Lambda^T(t_o) + \frac{(t_f - t_o)}{2} \sum_{i=1}^N w_i \cdot \dot{\Lambda}_i^T. \end{aligned} \quad (5.41)$$

The KKT conditions found from the NLP defined by (5.21 - 5.24), are exactly equivalent to the set of conditions that are the discretized form of the continuous first-order necessary conditions. This result indicates that solving the NLP derived from the pseudospectral transcription of the optimal control problem in differential form is exactly equivalent to solving the discretized form of the continuous first-order necessary conditions. It also indicates that the differential costate mapping (5.32) is valid, so that the costate can be estimated directly from the KKT multipliers of the NLP. The initial and final costates can also be found from the Lagrange multipliers (5.41). \square

Theorem 5.4.1 indicates that solving the NLP derived from the pseudospectral transcription of the optimal control problem is exactly equivalent to solving the discretized form of the continuous first-order necessary conditions. The state dynamics are collocated at the Gauss points, and the resulting costate dynamics are also collocated at the Gauss points. This result indicates that solving the NLP from the discretized problem is mathematically equivalent to finding the polynomials of degree N for the states and costates, $\mathbf{X}(t), \Lambda(t) \in \mathbb{R}^n$, and polynomial of degree $N - 1$ for the control, $\mathbf{U}(t) \in \mathbb{R}^m$, that satisfy the boundary conditions and the control, state, and costate dynamics at the collocation points. This result is exactly equivalent to the result from the integral form of the optimal control problem (see Section 4.6).

The approximating polynomials satisfy

$$\begin{aligned}
\frac{d\mathbf{X}}{dt}(t_k) &= \mathbf{f}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k), \\
\frac{d\boldsymbol{\Lambda}}{dt}(t_k) &= -\frac{\partial g}{\partial \mathbf{x}}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k) - \boldsymbol{\Lambda}^T(t_k) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k), \\
\mathbf{0} &= \frac{\partial g}{\partial \mathbf{u}}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k) + \boldsymbol{\Lambda}^T(t_k) \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{X}(t_k), \mathbf{U}(t_k), t_k), \\
k &= 1, \dots, N,
\end{aligned} \tag{5.42}$$

$$\phi(\mathbf{X}(t_o), t_o, \mathbf{X}(t_f), t_f) = 0,$$

$$\boldsymbol{\Lambda}(t_o) = \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_o)},$$

$$\boldsymbol{\Lambda}(t_f) = \frac{\partial \Phi}{\partial \mathbf{X}(t_f)} - \boldsymbol{\nu}^T \cdot \frac{\partial \phi}{\partial \mathbf{X}(t_f)}.$$

It has been shown that the differential Gauss pseudospectral method is mathematically equivalent to the integral pseudospectral method. Both of these methods overcome the problems at the costate boundaries associated with the Legendre pseudospectral method. The differential formulation has the advantage over the integral method, in that the resulting NLP has a significantly more sparse Jacobian matrix. For software packages that take advantage of this sparsity, such as SNOPT [30], an improvement in computation time can be made for large problems by using the differential method.

5.5 Summary

In this chapter, the differential Gauss pseudospectral method was derived. This pseudospectral method transcribes the continuous optimal control problem in differential form to a nonlinear program (NLP). This transcription has significant differences from the Legendre pseudospectral method. In this case, the dynamic constraints are collocated at Gauss points, which do not include the boundary points.

It was shown that the Gauss pseudospectral method using the differential form is mathematically equivalent to the integral form. As a result, the Karush-Kuhn-Tucker

(KKT) conditions from this method are also consistent with the continuous first-order necessary conditions. This property indicates that the Gauss pseudospectral method using the differential form, can be used to find accurate solutions of the states, costates, and controls of optimal control problems.

The differential form of the Gauss pseudospectral method has the advantage that it yields an NLP that is more sparse than the integral formulation. Numerical solvers can take advantage of the sparsity and solve the NLP in less computation time. In the next chapter, many example problems are considered to show the advantages of the Gauss pseudospectral method.

Chapter 6

Example Problems

In this chapter, several example problems are considered to show the advantages and disadvantages of the proposed Gauss pseudospectral method. The first is a simple LQR problem, which is used to compare the convergence properties of the Gauss pseudospectral method with the Legendre pseudospectral method, as well as some common finite difference methods.

The second example is a simple nonlinear optimal control problem, which shows the advantages of using the Gauss method in the estimation of the initial costate.

The third example is a bang-bang problem. In this example, the exact solution for the control has a discontinuity. It is shown that this discontinuity presents a problem for the pseudospectral methods, and how this problem can be overcome by using the multiple phase approach.

The fourth example is one that has a state path constraint. This constraint introduces discontinuities that again present a problem for the Gauss pseudospectral method. The multiple phase approach, however, does not work as well on the example with a state path constraint.

The fifth example considered is the classic brachistochrone problem. In this problem, one of the costates is infinite at the initial time. This infinite point is shown to reduce the accuracy of the Gauss pseudospectral method. It is also shown that control problems having an infinite point near the solution interval also significantly affect the accuracy of the proposed method.

The sixth example is one that includes a singular arc. This example shows that the Gauss method is not well suited for solving problems that contain singular arcs.

The seventh example considered is one that has multiple local minimum. It is shown that special care must be taken in choosing an initial guess for the method in order to find the desired global minimum to the problem.

Finally, the last example considered is a more complicated nonlinear optimal control problem, involving a low thrust orbit transfer. It is shown in this example that the Gauss pseudospectral method returns a significantly better solution than the Legendre pseudospectral method.

Note that the example problems in this chapter were solved using the integral form of the Gauss pseudospectral method (see Chapter 4). This was done because the example problems were solved before the differential form of the Gauss pseudospectral method was developed. It has been shown in Chapter 5, that the differential and integral forms are mathematically equivalent and therefore, the solution using either formulation will be identical. A comparison of the differential and integral forms of the Gauss pseudospectral method is made on the low thrust orbit transfer problem.

6.1 LQR Problem

The first example considered is a linear quadratic regulator (LQR) problem. The problem is solved numerically using the Euler, Runge-Kutta, Legendre pseudospectral, and Gauss pseudospectral methods. The problem is to minimize a quadratic cost function with fixed initial and final times, so the cost is

$$J = \frac{1}{2}x(t_f)^T \cdot S \cdot x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x(t)^T \cdot Q \cdot x(t) + u(t)^T \cdot R \cdot u(t)) dt, \quad (6.1)$$

subject to linear system dynamics,

$$\frac{dx}{dt} = A \cdot x + B \cdot u, \quad (6.2)$$

and possible boundary conditions on some or all the states,

$$\begin{aligned}x(t_o) &= x_o, \\x(t_f) &= x_f.\end{aligned}\tag{6.3}$$

The number of states is n , $x(t) \in \mathbb{R}^n$, and the number of controls is m , $u(t) \in \mathbb{R}^m$, so that $S \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$.

The solution to the problem can be found by defining a two-point boundary value problem derived from the first-order optimality conditions ([10] [45]). The result is

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} A & -B \cdot R^{-1} \cdot B^T \\ -Q & -A^T \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix},\tag{6.4}$$

where $\lambda \in \mathbb{R}^n$ is the costate. The boundary conditions are

$$\begin{aligned}x(t_o) &= x_o, & \lambda(t_o) &= \nu_o, \\x(t_f) &= x_f, & \lambda(t_f) &= S \cdot x(t_f) - \nu_f,\end{aligned}\tag{6.5}$$

with Lagrange multipliers, $\nu_o, \nu_f \in \mathbb{R}^n$. The optimal control is defined to be

$$u(t) = -R^{-1} \cdot B^T \cdot \lambda(t).\tag{6.6}$$

The specific case considered here is for single state and single control, $n, m = 1$, so that the matrixes, S , Q , R , A , and B are scalars. The values for these scalars are chosen so that $S = 0$ (no terminal cost), and A , B , Q , and R are chosen to be one. The initial condition, x_o is one, and the final condition x_f is zero. The initial time, $t_o = 0$, and the final time, $t_f = 5$. The problem is reduced to solving for $x(t) \in \mathbb{R}$ and $\lambda(t) \in \mathbb{R}$, so that

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \doteq M \cdot \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix},\tag{6.7}$$

with boundary conditions,

$$\begin{aligned} x(t_o) &= 1, & \lambda(t_o) &= \nu_o, \\ x(t_f) &= 0, & \lambda(t_f) &= -\nu_f. \end{aligned} \tag{6.8}$$

The solution can be found by defining the eigenvectors, v_1 and v_2 , and eigenvalues, μ_1 and μ_2 , of the matrix $M \in \mathbb{R}^{2 \times 2}$. The state and costates are then linear combinations of eigenvalues and eigenvectors, so that

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = c_1 \cdot v_1 \cdot e^{\mu_1 t} + c_2 \cdot v_2 \cdot e^{\mu_2 t} \tag{6.9}$$

The integration constants, c_1 and c_2 , are determined from the boundary conditions on the state, $x(t_o)$ and $x(t_f)$. The final solution for the state, costate, and control is

$$\begin{aligned} x(t) &= 1.0000 \cdot e^{-\sqrt{2}t} - 7.2135 \times 10^{-7} \cdot e^{\sqrt{2}t}, \\ \lambda(t) &= 2.4124 \cdot e^{-\sqrt{2}t} - 2.9879 \times 10^{-7} \cdot e^{\sqrt{2}t}, \\ u(t) &= -\lambda(t), \end{aligned} \tag{6.10}$$

and the Lagrange multipliers are

$$\nu_o = 2.4124, \quad \nu_f = -0.0024. \tag{6.11}$$

The solution is plotted in Fig. 6-1. This solution will be referred to as the exact solution for comparison to the numerical approximations.

6.1.1 Euler Transcription

The one dimensional LQR problem is solved numerically by using an Euler transcription (see Section 3.1). The dynamic constraints are approximated at a uniform

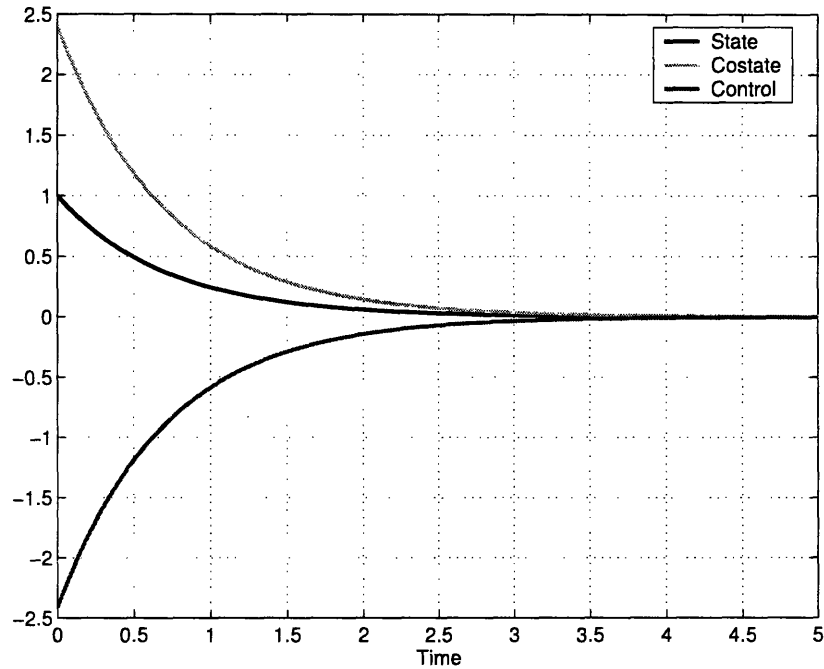


Figure 6-1: LQR Exact Solution

distribution of nodes, $0 = t_1 < \dots < t_N = 5$. $x_k \doteq x(t_k)$ and $u_k \doteq u(t_k)$, so that

$$x_{k+1} = x_k + h \cdot (x_k + u_k), \quad k = 1, \dots, N - 1, \quad (6.12)$$

where $h = \frac{1}{N-1}$. The cost function is approximated as

$$J = \frac{1}{2} \sum_{k=1}^{N-1} (x_k^2 + u_k^2) h. \quad (6.13)$$

This approximation then defines an NLP to find the N state variables, x_k , $k = 1, \dots, N$, and $N - 1$ control variables, u_k , $k = 1, \dots, N - 1$, that minimize J subject to the $N - 1$ equality constraints (6.12) and the two boundary conditions $x_1 = 1$ and $x_N = 0$. The approximate solution is plotted in Fig. 6-2 and Fig. 6-3.

6.1.2 Runge-Kutta Transcription

The LQR problem is solved using a four stage Runge-Kutta transcription (see Section 3.2) at the same set of equidistant points, $0 = t_1 < \dots < t_N = 5$. The state dynamics are approximated as

$$x_{i+1} = x_i + \frac{1}{6} (s_{1i} + 2 \cdot s_{2i} + 2 \cdot s_{3i} + s_{4i}), \quad i = 1, \dots, N-1, \quad (6.14)$$

where the stages are defined as

$$\begin{aligned} s_{1i} &= h \cdot (x_i + u_i), \\ s_{2i} &= h \cdot (x_i + 0.5 \cdot s_{1i} + \tilde{u}_{i+1}), \\ s_{3i} &= h \cdot (x_i + 0.5 \cdot s_{2i} + \tilde{u}_{i+1}), \\ s_{4i} &= h \cdot (x_i + s_{3i} + u_{i+1}), \end{aligned} \quad (6.15)$$

and \tilde{u}_{i+1} is the estimate for the control $u(t_i + h/2)$. The cost function is approximated as

$$J = \frac{1}{2} \sum_{k=1}^{N-1} \frac{1}{6} (\bar{s}_{1k} + 2 \cdot \bar{s}_{2k} + 2 \cdot \bar{s}_{3k} + \bar{s}_{4k}), \quad (6.16)$$

where the stages are defined as

$$\begin{aligned} \bar{s}_{1k} &= \frac{h}{2} (x_k^2 + u_k^2), \\ \bar{s}_{2k} &= \frac{h}{2} ((x_k + 0.5 \cdot s_{1k})^2 + \tilde{u}_{k+1}^2), \\ \bar{s}_{3k} &= \frac{h}{2} ((x_k + 0.5 \cdot s_{2k})^2 + \tilde{u}_{k+1}^2), \\ \bar{s}_{4k} &= \frac{h}{2} ((x_k + s_{3k})^2 + u_{k+1}^2). \end{aligned} \quad (6.17)$$

The cost and constraints along with the boundary conditions, $x_1 = 1$ and $x_N = 0$, define the NLP over the N state variables, $N-1$ control variables, and $N-1$ intermediate control variables. The approximate solution is plotted in Fig. 6-2 and Fig. 6-3.

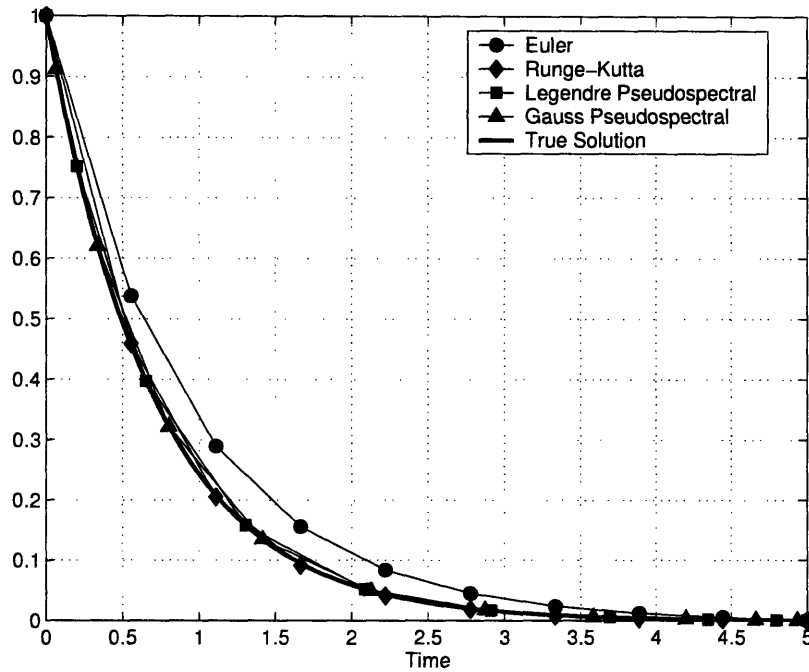


Figure 6-2: LQR Approximate State Solution

6.1.3 Legendre Pseudospectral Method

The LQR is solved using the Legendre pseudospectral method (see Section 3.3) by defining the state constraints in terms of the pseudospectral differentiation matrix, so that

$$\frac{2}{5}D \cdot \mathbf{x}_N = \mathbf{x}_N + \mathbf{u}_N, \quad (6.18)$$

where $\mathbf{x}_N \in \mathbb{R}^N$ is a vector of the approximation to the states at the Gauss-Lobatto points, and $\mathbf{u}_N \in \mathbb{R}^N$ is the approximation to the control. The fraction in the constraint is to take care of the time transformation from $[0, 5]$ to $[-1, 1]$. The cost is found using the Gauss-Lobatto quadrature rule, so that

$$J = \frac{5}{2} \sum_{k=1}^N \frac{1}{2} (x_k^2 + u_k^2) \cdot w_k, \quad (6.19)$$

where x_k is the k th component of the vector \mathbf{x}_N , and u_k is the k th component of the control vector. Also, w_k is the Gauss Lobatto weight. The boundary conditions

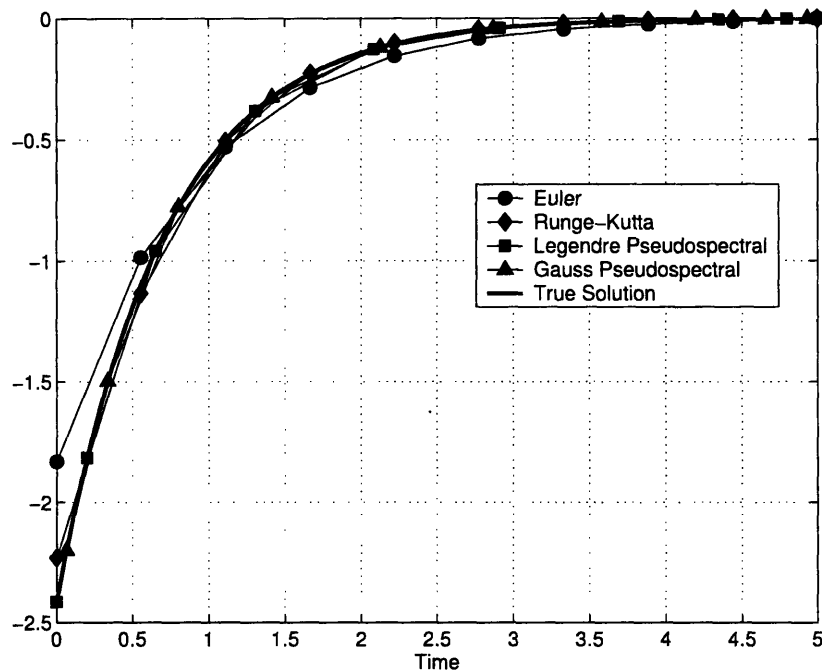


Figure 6-3: LQR Approximate Control Solution

are $x_1 = 1$ and $x_N = 0$, the first and last components of the state vector. These constraints and the cost function make up the NLP for the Legendre pseudospectral transcription. The approximate solution is plotted in Fig. 6-2 and Fig. 6-3.

6.1.4 Gauss Pseudospectral Method

Finally, the LQR problem is solved using the Gauss pseudospectral method by defining the state constraints in terms of the pseudospectral integration matrix, A , so that

$$\mathbf{x}_N = 1 + \frac{5}{2}A \cdot (\mathbf{x}_N + \mathbf{u}_N), \quad (6.20)$$

where \mathbf{x}_N and \mathbf{u}_N are vectors of the state and control at Gauss points. The cost function is found using the Gauss quadrature rule, so that

$$J = \frac{5}{2} \sum_{k=1}^N \frac{1}{2} (x_k^2 + u_k^2) \cdot w_k, \quad (6.21)$$

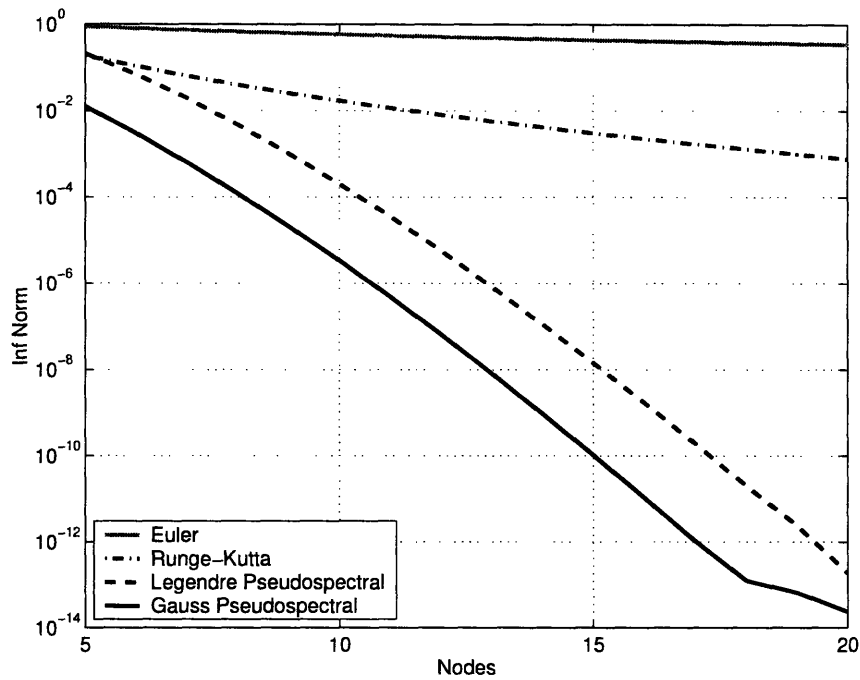


Figure 6-4: LQR Control Convergence

where x_k and u_k are components of the state and control vectors, and w_k is the Gauss quadrature weight. The initial condition is already included in (6.20), and the final condition is included using the integral of the state dynamics. This condition is

$$0 = 1 + \frac{5}{2} \sum_{k=1}^N (x_k + u_k) \cdot w_k . \quad (6.22)$$

These constraints and cost define the NLP for the Gauss pseudospectral method. The approximate solution is plotted in Fig. 6-2 and Fig. 6-3.

6.1.5 LQR Problem Convergence

The plots of the approximate solutions, Fig. 6-2 and Fig. 6-3, indicate that all four methods are in reasonable agreement with the exact solution. The rates of convergence of the control for the four transcription methods is shown in Fig. 6-4. These convergence plots are found by taking the infinity norm of the difference between the

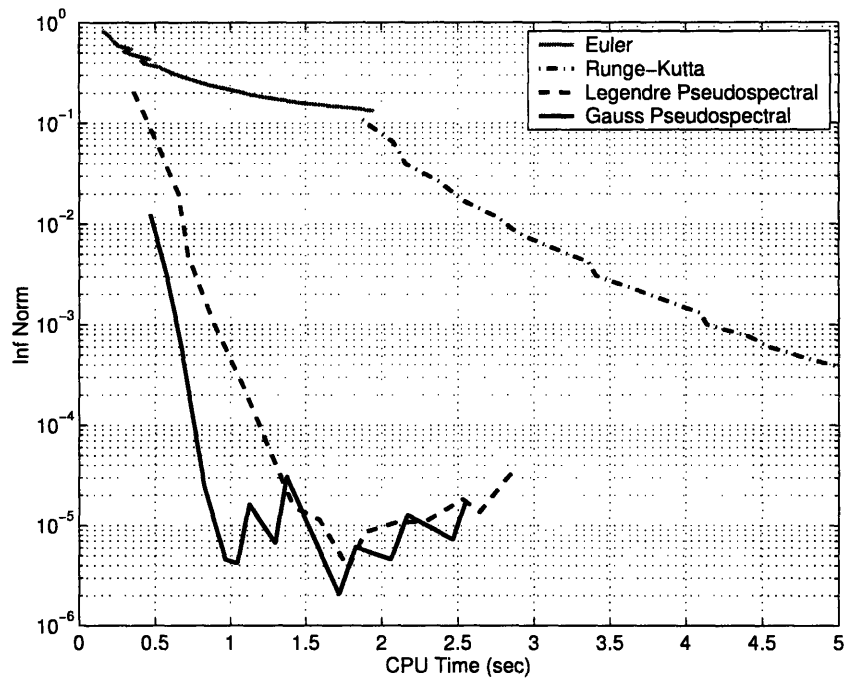


Figure 6-5: LQR Control Error vs. CPU Time

approximate solution for the control and the exact solution for the control. The error is then plotted as a function of the number of nodes used. While the Runge-Kutta problem converges at a faster rate than the Euler problem, neither one is nearly as fast as the pseudospectral transcriptions. The solution found using the Gauss pseudospectral method has the same convergence rate (the lines are parallel) as the Legendre pseudospectral method, but it is always approximately two orders of magnitude more accurate. This example shows that the Gauss pseudospectral method converges for the linear quadratic optimal control problem.

The time to solve the LQR problem for each of the four transcriptions using SNOPT [30] is shown in Fig. 6-5. All four methods were setup in an efficient manner and solved using the same initial guess. The figure shows that the pseudospectral transcriptions are significantly more accurate than the finite difference schemes for the same amount of computation time. Also the Gauss pseudospectral method is faster than the Legendre pseudospectral method for the same amount of accuracy. This

property is due to the fact that the Gauss method can achieve the same accuracy as the Legendre method using fewer nodes. Note that both the pseudospectral transcriptions were solved directly (see Appendix B) for the convergence plot (Fig. 6-4) but solved with SNOPT to compare CPU time (Fig. 6-5).

6.2 Nonlinear Example

The nonlinear problem considered here is a problem that was derived from a simple one-dimensional LQR problem. Formulating the problem in this way allows for the generation of a simple nonlinear problem with a known analytic solution. The problem was created by defining a new state $y(t) \in \mathbb{R}$, as the square of the original state $x(t) \in \mathbb{R}$, in the LQR problem (6.1 - 6.3). The transformation is valid because the relation $y(t) = x(t)^2$, is one-to-one and a surjection for $y(t), x(t) \geq 0$. The new cost function becomes

$$J = \frac{1}{2} \int_{t_o}^{t_f} (q \cdot y(t) + r \cdot u(t)^2) dt, \quad (6.23)$$

with state dynamics

$$\frac{dy}{dt} = 2 \cdot a \cdot y(t) + 2 \cdot b \cdot \sqrt{y(t)} \cdot u(t), \quad (6.24)$$

and boundary conditions

$$\begin{aligned} y(t_o) &= x_o^2, \\ y(t_f) &= x_f^2. \end{aligned} \quad (6.25)$$

In the example considered here, the scalars a , b , q , and r are set equal to one. The initial condition is $x_o = \sqrt{2}$, and the final condition is $x_f = 1$. The time interval considered is from $[0, 5]$. The solution of the nonlinear problem is found from the solution to the original LQR problem, which can be computed using the technique shown in Section 6.1. The solution of the LQR problem shown to five significant

figures is

$$\begin{aligned}
x(t) &= 1.4134 \cdot e^{-\sqrt{2}t} + 8.4831 \times 10^{-4} \cdot e^{\sqrt{2}t} , \\
\lambda_x(t) &= 3.4122 \cdot e^{-\sqrt{2}t} - 3.5138 \times 10^{-4} \cdot e^{\sqrt{2}t} , \\
u(t) &= -3.4122 \cdot e^{-\sqrt{2}t} + 3.5138 \times 10^{-4} \cdot e^{\sqrt{2}t} .
\end{aligned} \tag{6.26}$$

The solution to the nonlinear problem is found by squaring the state, the control is unchanged, and the costate can be found by the relation between the control and costate for the two problems. The solution to the continuous nonlinear optimal control problem (6.23 - 6.25) is

$$\begin{aligned}
y(t) &= x(t)^2 = \left(1.4134 \cdot e^{-\sqrt{2}t} + 8.4831 \times 10^{-4} \cdot e^{\sqrt{2}t} \right)^2 , \\
\lambda_y(t) &= \frac{\lambda_x(t)}{2 \cdot \sqrt{y(t)}} = \left(3.4122 \cdot e^{-\sqrt{2}t} - 3.5138 \times 10^{-4} \cdot e^{\sqrt{2}t} \right) / \left(2 \cdot \sqrt{y(t)} \right) , \\
u(t) &= -3.4122 \cdot e^{-\sqrt{2}t} + 3.5138 \times 10^{-4} \cdot e^{\sqrt{2}t} .
\end{aligned} \tag{6.27}$$

The solution for the state is shown in Fig. 6-6, and the solution for the costate is shown in Fig. 6-7.

6.2.1 Legendre Pseudospectral Method

The Legendre pseudospectral solution to the nonlinear optimal control problem is found by defining a NLP with a cost based on the Gauss-Lobatto quadrature rule, so that

$$J = \frac{5}{4} \cdot w^T \cdot (Y_N + U_N^2) , \tag{6.28}$$

where $w \in \mathbb{R}^N$ is the vector of LGL weights, and $Y_N, U_N \in \mathbb{R}^N$ are the vectors of the state and control at the LGL points respectively. The state dynamics are approximated using the differential matrix $D \in \mathbb{R}^{N \times N}$, so that

$$\frac{2}{5} \cdot D \cdot Y_N = 2 \cdot Y_N + 2 \cdot \sqrt{Y_N} \cdot U_N . \tag{6.29}$$

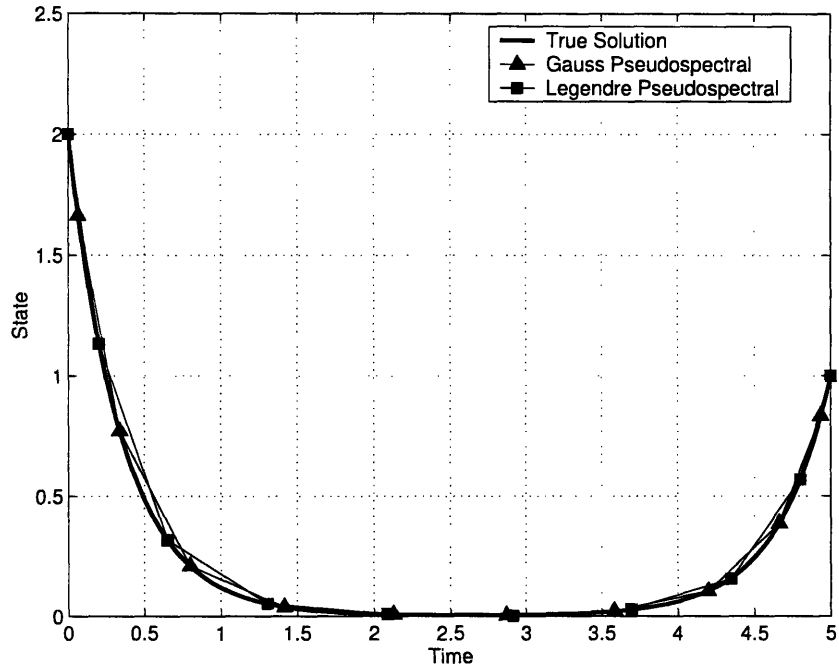


Figure 6-6: Nonlinear Approximate State Solution

Finally the boundary conditions are enforced by requiring the first entry in Y_N to be 2 and the final entry to be 1. The solution to this NLP was found using SNOPT [30]. The estimation for the costate was found using the costate mapping principle [22]. The costate is estimated to be

$$\Lambda_N = \frac{2}{5} \cdot W^{-1} \cdot \tilde{\lambda}_N, \quad (6.30)$$

where W is a square matrix with the LGL weights on the diagonal, and $\tilde{\lambda}_N$ is the vector of KKT multipliers. The approximate solution, using 10 nodes, for the state is shown in Fig. 6-6, and the costate in Fig. 6-7.

6.2.2 Gauss Pseudospectral Method

The Gauss pseudospectral solution to the nonlinear problem is found by defining the NLP based on the integral form of the optimal control problem at Gauss points. The

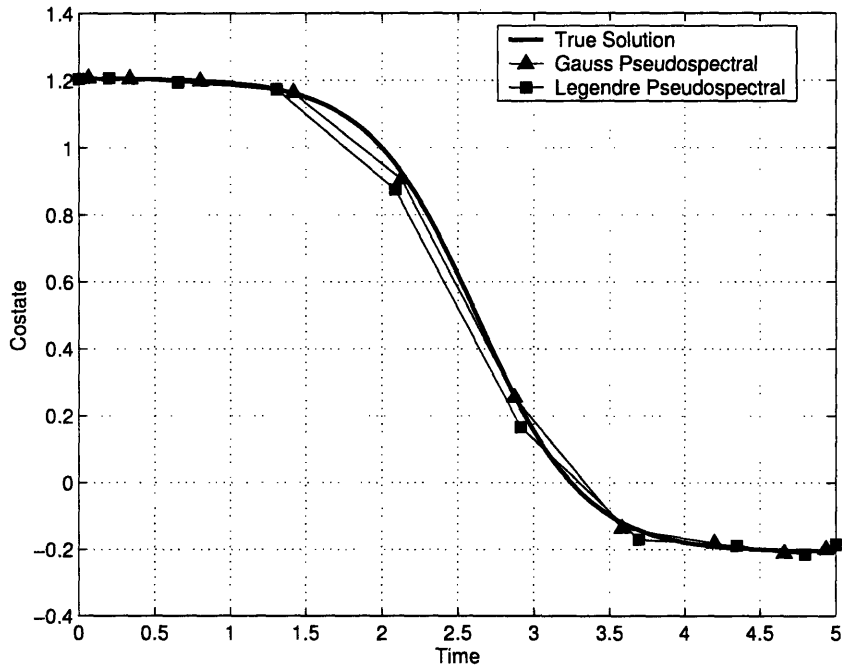


Figure 6-7: Nonlinear Approximate Costate Solution

cost function is defined in terms of a Gauss quadrature, so that

$$J = \frac{5}{4} \cdot w^T \cdot (Y_N + U_N^2) , \quad (6.31)$$

where $w \in \mathbb{R}^N$ is the vector of Gauss weights, and $Y_N, U_N \in \mathbb{R}^N$ are the vectors of the state and control at the Gauss points respectively. The integral form of the state dynamics are approximated using the integration approximation matrix $A \in \mathbb{R}^{N \times N}$, so that

$$Y_N = Y_o + \frac{5}{2} \cdot A \left(2 \cdot Y_N + 2 \cdot \sqrt{Y_N} \cdot U_N \right) . \quad (6.32)$$

The boundary constraints are enforced using the conditions $Y_o = 2$ and $Y_f = 1$, with the relation

$$Y_f = Y_o + \frac{5}{2} \cdot w^T \cdot \left(2 \cdot Y_N + 2 \cdot \sqrt{Y_N} \cdot U_N \right) . \quad (6.33)$$

The solution was again found using SNOPT [30], and the costate estimate was

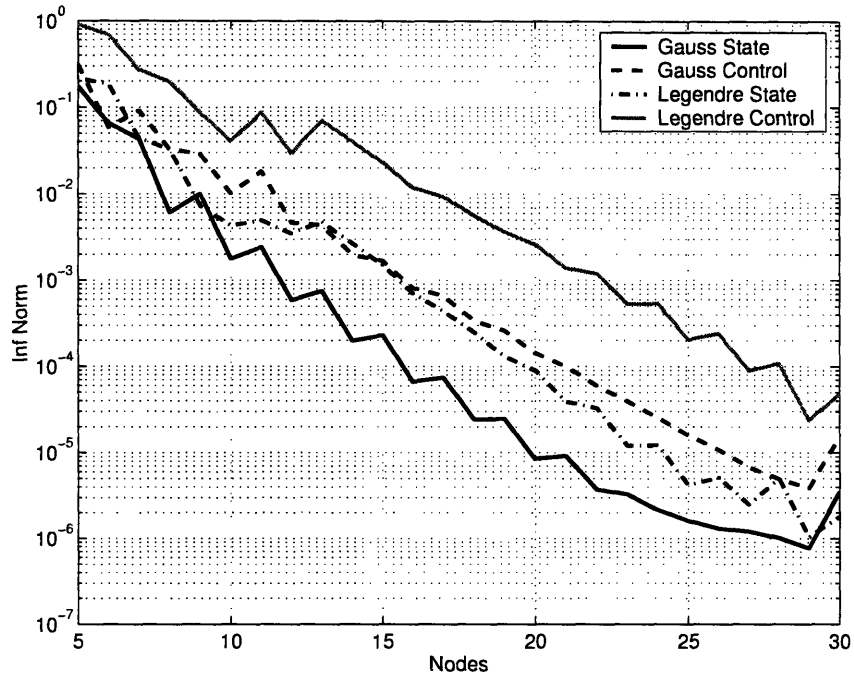


Figure 6-8: Nonlinear Problem State/Control Convergence

found using the integral costate mapping principle (4.76). The costate estimate is

$$\Lambda_N = W^{-1} \cdot A^T \cdot \tilde{P}_N - \nu_f, \quad (6.34)$$

where W is a square matrix with the Gauss weights on the diagonal, \tilde{P}_N is the vector of KKT multipliers associated to the dynamic constraints, and ν_f is the Lagrange multiplier associated to the final state constraint. The initial costate is estimated using the sum of the KKT multipliers, so that

$$\Lambda(0) = P_N - \nu_f. \quad (6.35)$$

The approximate solution for the state is shown in Fig. 6-6, and the costate in Fig. 6-7, again using 10 nodes.

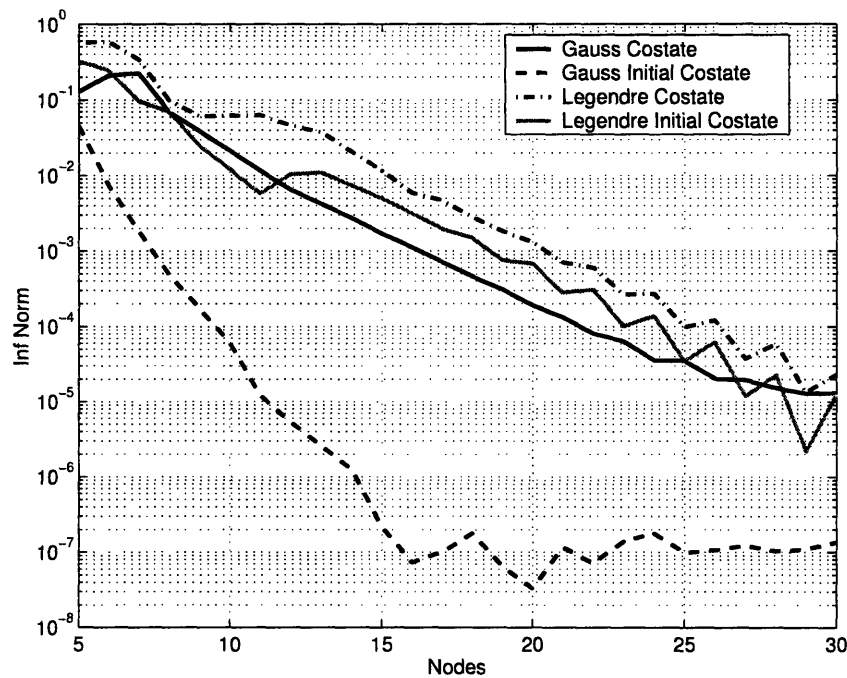


Figure 6-9: Nonlinear Problem Costate Convergence

6.2.3 Nonlinear Problem Convergence

Figures 6-6 and 6-7 show excellent agreement between the approximate pseudospectral solutions and the exact solution. The convergence of the Legendre pseudospectral method and the Gauss pseudospectral method on this nonlinear problem is examined in Figures 6-8 and 6-9. Fig. 6-8 shows the convergence of the state and control in the infinity norm as a function of the number of nodes used, for both the Legendre and Gauss methods. The figure indicates that both methods are converging very rapidly, but the Gauss method error is always slightly better than the Legendre method.

Fig. 6-9 shows the convergence for the costate and initial costate in the infinity norm, as a function of the number of nodes. The convergence rates for the costates are again the same, with the Gauss method costate always better than the Legendre costate. The convergence rates for the initial costates, however, are not the same. The convergence rate for the initial Legendre costate is the same as the convergence of the other costates, but the convergence rate for the initial costate from the Gauss

method is much faster. Note that the error in the initial costate is decreasing until it approximately reaches the tolerance of SNOPT. This result indicates that the initial costate estimate from the Gauss method is significantly more accurate than the costate estimates at the Gauss points and the costate estimates from the Legendre method. This example shows that the Gauss pseudospectral method can solve nonlinear optimal control problems with very fast convergence properties.

6.3 Bang-Bang Control Problem

The next example considered is one with a discontinuity in the control. The bang-bang control problem is a reorientation problem of a double integrator, from an arbitrary state to the origin in minimum time. The control is bounded in both the positive and negative directions by a maximum value. The cost is the final time,

$$J = t_f , \tag{6.36}$$

which is free. The state dynamics of the double integrator are

$$\begin{aligned} \frac{dx_1}{dt}(t) &= x_2(t) , \\ \frac{dx_2}{dt}(t) &= u(t) , \end{aligned} \tag{6.37}$$

with initial conditions

$$x_1(0) = x_{1o} , \quad x_2(0) = x_{2o} . \tag{6.38}$$

The feasible control is defined as

$$|u(t)| \leq u_{max} = 1 . \tag{6.39}$$

The exact solution to the bang-bang problem can be found by using Pontryagin's

maximum principle to determine the control. The control is

$$u(t) = \begin{cases} 1, & \lambda_2(t) < 0, \\ -1, & \lambda_2(t) > 0. \end{cases} \quad (6.40)$$

This control can be used to determine the switching curve in the state space. The exact solution for the case $x_{1o} = 1$, $x_{2o} = 3$ is

$$\begin{aligned} x_1(t) &= \begin{cases} -t^2/2 + x_{2o} \cdot t + x_{1o}, & t < t_1, \\ t^2/2 - t_f \cdot t + t_f^2/2, & t > t_1, \end{cases} \\ x_2(t) &= \begin{cases} -t + x_{2o}, & t < t_1, \\ t - t_f, & t > t_1, \end{cases} \\ u(t) &= \begin{cases} -1, & t < t_1, \\ 1, & t > t_1, \end{cases} \end{aligned} \quad (6.41)$$

where the switching time is $t_1 = x_{2o} + \sqrt{0.5 \cdot x_{2o}^2 + x_{1o}}$, and final time is $t_f = 2 \cdot t_1 - x_{2o}$.

The costates are

$$\lambda_1(t) = c_1, \quad (6.42)$$

$$\lambda_2(t) = -c_1 \cdot t + c_2,$$

where $c_1 = -1/(t_1 - t_f)$ and $c_2 = c_1 \cdot t_1$. The exact solution for the states is plotted in Fig. 6-10, 6-11, the control is plotted in Fig. 6-12, and the exact solution for the costates is plotted in Fig. 6-13, 6-14.

6.3.1 Legendre Pseudospectral Method

The Legendre pseudospectral solution to the bang-bang problem was found by defining the NLP in terms of the variables $X_{1N}, X_{2N}, U_N \in \mathbb{R}^N$, which are the approximations to the states and control at the LGL points. The final time $t_f \in \mathbb{R}$ is also a free variable to be estimated. The cost is to minimize the final time

$$J = t_f, \quad (6.43)$$

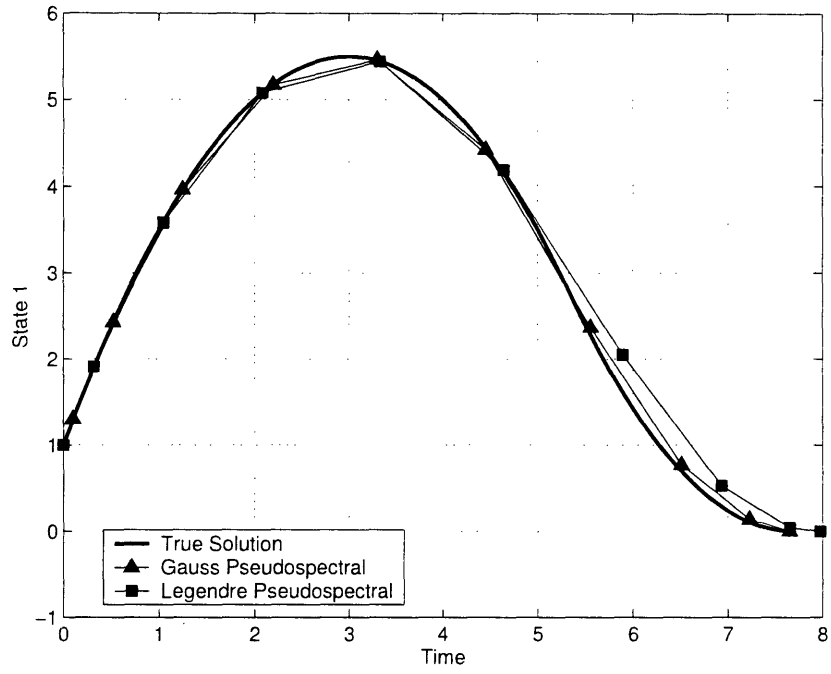


Figure 6-10: Bang-Bang Problem. State 1 Solution

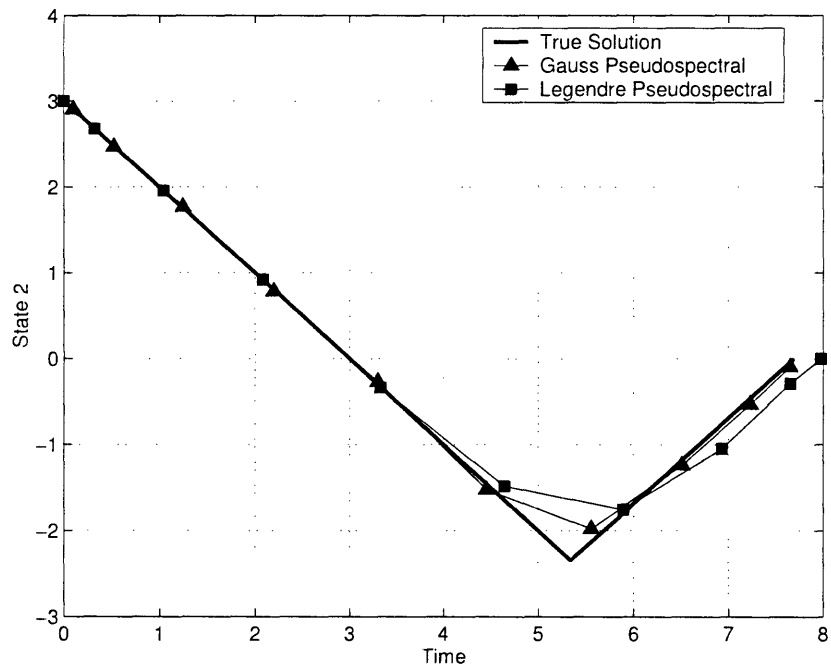


Figure 6-11: Bang-Bang Problem. State 2 Solution

subject to the set of dynamic constraints

$$\begin{aligned}\frac{2}{t_f}D \cdot X_{1N} - X_{2N} &= 0, \\ \frac{2}{t_f}D \cdot X_{2N} - U_N &= 0,\end{aligned}\tag{6.44}$$

and boundary conditions

$$\begin{aligned}X_{1N}[1] &= 1, & X_{2N}[1] &= 3, \\ X_{1N}[N] &= 0, & X_{2N}[N] &= 0,\end{aligned}\tag{6.45}$$

which are the first and last entries in the vectors. Finally the control constraint is enforced as

$$-1 \leq U_N \leq 1.\tag{6.46}$$

The NLP was solved using SNOPT [30], and the approximate solution for the states and control using 10 nodes is shown in Fig. 6-10,6-11, and 6-12.

The costate estimates were generated using the KKT multipliers from the NLP solver. The costate estimates are

$$\begin{aligned}\lambda_{1N} &= \frac{t_f}{2}W^{-1} \cdot \tilde{\lambda}_1, \\ \lambda_{2N} &= \frac{t_f}{2}W^{-1} \cdot \tilde{\lambda}_2,\end{aligned}\tag{6.47}$$

where W is a square matrix with the LGL weights on the diagonal. The costate estimates are shown in Fig. 6-13, 6-14.

6.3.2 Gauss Pseudospectral Method

The Gauss pseudospectral solution is found by defining an NLP in terms of the variables $X_{1N}, X_{2N}, U_N \in \mathbb{R}^N$, which are the approximations to the states and control at the Gauss collocation points. The final time, t_f , is minimized so that

$$J = t_f,\tag{6.48}$$

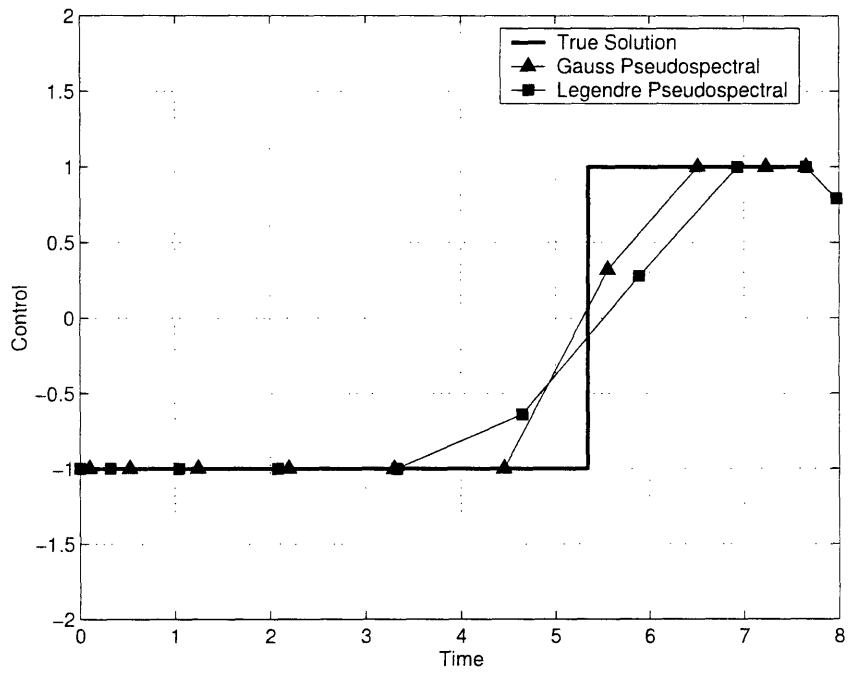


Figure 6-12: Bang-Bang Control Solution

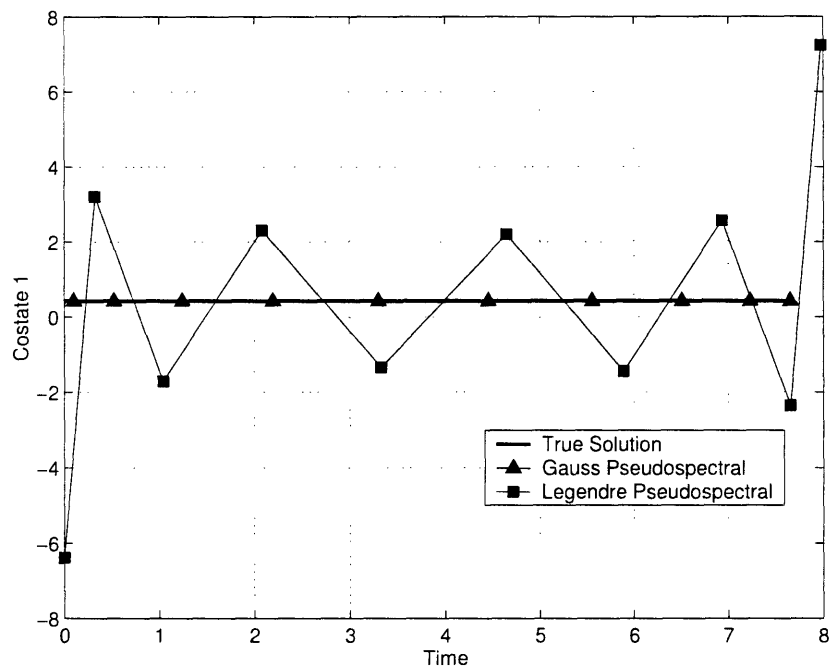


Figure 6-13: Bang-Bang Costate 1 Solution

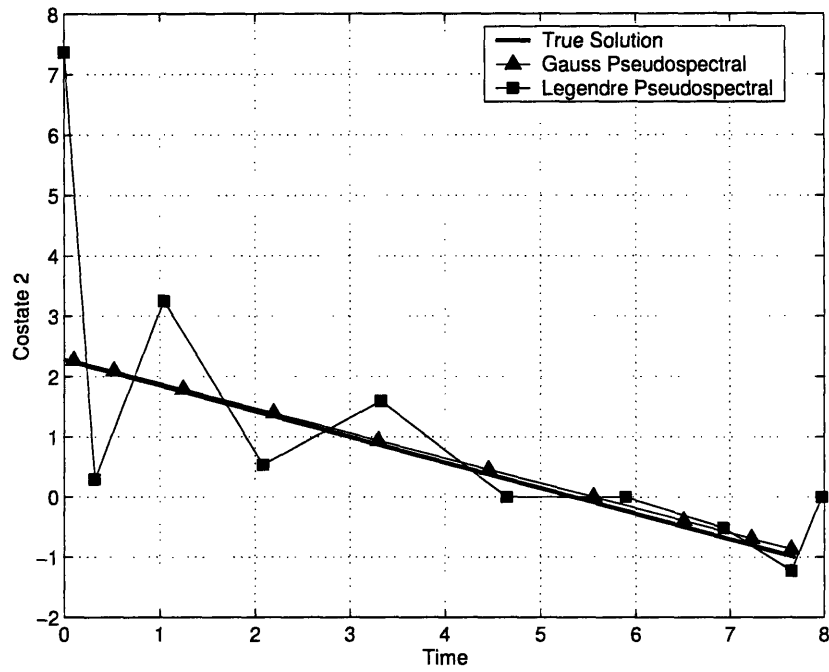


Figure 6-14: Bang-Bang Costate 2 Solution

subject to the dynamic constraints in integral form

$$\begin{aligned} X_{1N} - 1 - \frac{t_f}{2} A \cdot X_{2N} &= 0, \\ X_{2N} - 3 - \frac{t_f}{2} A \cdot U_N &= 0, \end{aligned} \quad (6.49)$$

with boundary constraints in terms of a Gauss quadrature

$$\begin{aligned} 1 + \frac{t_f}{2} w^T \cdot X_{2N} &= 0, \\ 3 + \frac{t_f}{2} w^T \cdot U_N &= 0. \end{aligned} \quad (6.50)$$

Finally, the control is constrained as

$$-1 \leq U_N \leq 1. \quad (6.51)$$

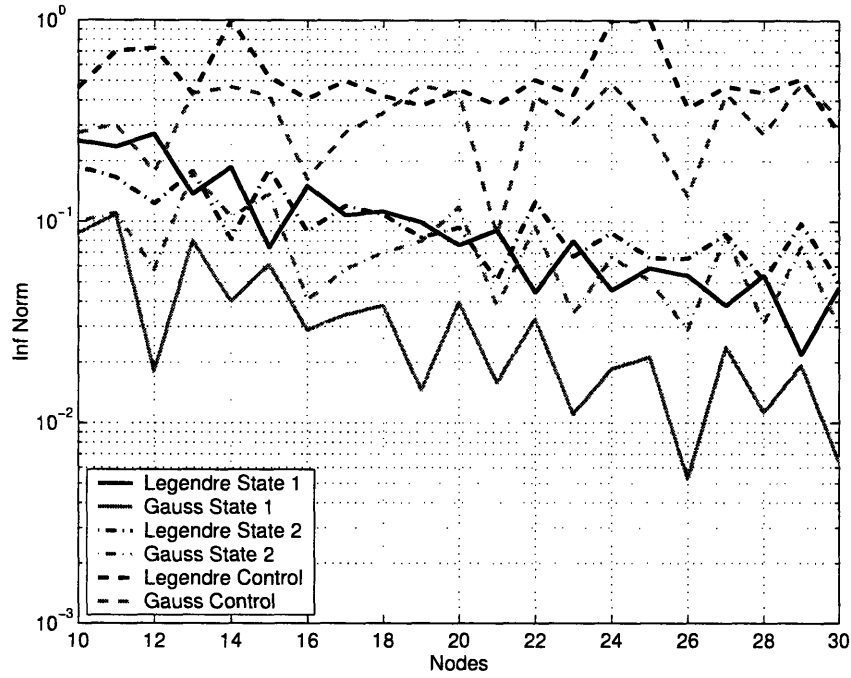


Figure 6-15: Bang-Bang Problem State/Control Convergence

The NLP was also solved using SNOPT [30] with 10 nodes and the solutions for the states and control are shown in Fig. 6-10, 6-11, and 6-12.

The costate estimate was found using the KKT multipliers and Lagrange multipliers from the NLP solver. The costates are

$$\begin{aligned}\lambda_{1N} &= W^{-1} \cdot A^T \cdot \tilde{P}_1 - \nu_1, \\ \lambda_{2N} &= W^{-1} \cdot A^T \cdot \tilde{P}_2 - \nu_2,\end{aligned}\tag{6.52}$$

where ν_1 and ν_2 are the Lagrange multipliers associated with the boundary constraints (6.50). The costate estimates are plotted in Fig. 6-13, 6-14.

6.3.3 Bang-Bang Problem Convergence

The plots of the approximate solutions show significant errors in the states and control, Fig. 6-10, 6-11, and 6-12, around the point of the control discontinuity. At the

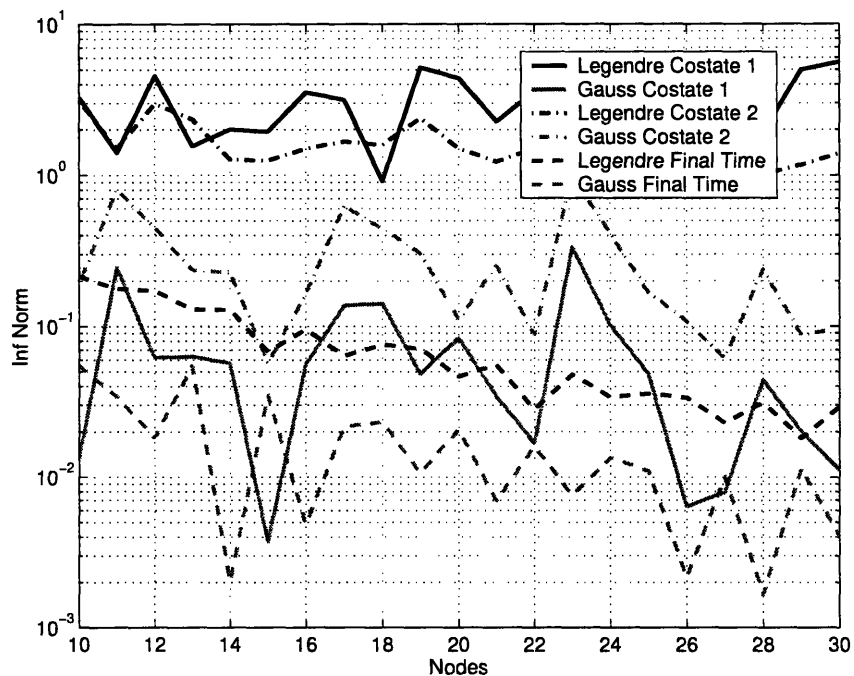


Figure 6-16: Bang-Bang Problem Costate/Final Time Convergence

switching point the control is discontinuous, the second state has a discontinuous first derivative, and the first state has a discontinuous second derivative. Both of the pseudospectral methods use polynomials to approximate these non-smooth functions. It can be seen in the figures that the polynomials do not do a very good job of approximating the exact solutions. The approximating polynomials exhibit a Gibbs phenomenon that is consistent with the results for polynomial interpolation of non-smooth functions [31].

The Gauss pseudospectral method, does however, have a significantly better costate estimate than the Legendre method as seen in Fig. 6-13, 6-14. The large errors in the costate for the Legendre method have been attributed to the problem with the defects in the costate equations (see Section 3.3.2). The Gauss method does not have these defects and therefore, results in a much improved costate estimate.

The convergence of the Legendre pseudospectral solution and the Gauss pseudospectral solution is investigated by looking at the infinity norm of the error between

the approximate solution and the exact solution for the states, costates, control, and final time. The plots for the error as a function of the number of nodes used for the states and control is shown in Fig. 6-15. The convergence plots for the costates and final time is shown in Fig. 6-16. These plots indicate that although the Gauss method seems to outperform the Legendre method, neither one shows much improvement as the number of nodes is increased. This result is due to the fact that the interpolation polynomials used in the pseudospectral methods are not well suited for approximating functions with discontinuities.

An improvement of the approximate solution can be found by using the multiphase approach to link two segments at the switching point.

6.3.4 Multiphase Solution

The multiphase solution to the bang-bang problem is formulated once the switching structure is known. The switching structure is estimated from the single phase solution (see Section 6.3.2), and for the initial conditions, $x_{1o} = 1$, $x_{2o} = 3$, the control starts at -1 and switches to 1 at some time t_1 , Fig. 6-12. The control is

$$u(t) = \begin{cases} -1, & t < t_1, \\ 1, & t > t_1. \end{cases} \quad (6.53)$$

The problem can be formulated so that the control is constrained to be -1 before the switch and 1 after the switch with the switching time as a free variable to be estimated.

The Gauss pseudospectral transcription of the multiphase problem is formulated with the variables $X_{1aN}, X_{2aN}, U_{aN} \in \mathbb{R}^N$, as the states and control on the first phase and $X_{1bN}, X_{2bN}, U_{bN} \in \mathbb{R}^N$, as the states and control on the second phase. Note that the number of nodes used does not have to be the same on all phases, but for this example the same number was used for simplicity. The states at the switching time, $X_{1bo}, X_{2bo} \in \mathbb{R}$ are used in the phase boundary constraints to link the two phases. The switching time, t_1 , and final time, t_f , are also free variables.

The cost of the NLP is again the final time,

$$J = t_f . \quad (6.54)$$

The dynamic constraints for the first phase are

$$\begin{aligned} X_{1aN} - 1 - \frac{t_1}{2} A \cdot X_{2aN} &= 0 , \\ X_{2aN} - 3 - \frac{t_1}{2} A \cdot U_{aN} &= 0 . \end{aligned} \quad (6.55)$$

The dynamic constraints on the second phase are

$$\begin{aligned} X_{1bN} - X_{1bo} - \frac{(t_f - t_1)}{2} A \cdot X_{2bN} &= 0 , \\ X_{2bN} - X_{2bo} - \frac{(t_f - t_1)}{2} A \cdot U_{bN} &= 0 . \end{aligned} \quad (6.56)$$

Note that different time transformations were used for the two different phases because the phases are not the same length of time. The final state constraints are expressed using a Gauss quadrature, so that

$$\begin{aligned} X_{1bo} + \frac{(t_f - t_1)}{2} w^T \cdot X_{2bN} &= 0 , \\ X_{2bo} + \frac{(t_f - t_1)}{2} w^T \cdot U_{bN} &= 0 . \end{aligned} \quad (6.57)$$

The phase boundary constraints which enforce continuity of the states are also expressed using a Gauss quadrature, so that

$$\begin{aligned} X_{1bo} - 1 - \frac{(t_1)}{2} w^T \cdot X_{2aN} &= 0 , \\ X_{2bo} - 3 - \frac{(t_1)}{2} w^T \cdot U_{aN} &= 0 . \end{aligned} \quad (6.58)$$

Finally, the control is constrained as

$$\begin{aligned} U_{aN} &= -1 , \\ U_{bN} &= 1 . \end{aligned} \quad (6.59)$$

These equations define the cost and the constraints of the NLP, which defines the solution to the multiphase bang-bang problem.

The costate estimate is found using the KKT and Lagrange multipliers from the NLP solver. The costates on the first phase are

$$\begin{aligned}\lambda_{1aN} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1aN} - \nu_{\mathcal{L}1} , \\ \lambda_{2aN} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2aN} - \nu_{\mathcal{L}2} ,\end{aligned}\tag{6.60}$$

where $\tilde{P}_{1aN}, \tilde{P}_{2aN} \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints (6.55), and $\nu_{\mathcal{L}1}, \nu_{\mathcal{L}2} \in \mathbb{R}$ are the Lagrange multipliers associated with the phase boundary constraints (6.58). The costates on the second phase are

$$\begin{aligned}\lambda_{1bN} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1bN} - \nu_1 , \\ \lambda_{2bN} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2bN} - \nu_2 ,\end{aligned}\tag{6.61}$$

where $\tilde{P}_{1bN}, \tilde{P}_{2bN} \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints (6.56), and $\nu_1, \nu_2 \in \mathbb{R}$ are the Lagrange multipliers associated with the boundary constraints (6.57). The final costates of the first phase are defined as

$$\begin{aligned}\lambda_{1a}(t_1) &= -\nu_{\mathcal{L}1} , \\ \lambda_{2a}(t_1) &= -\nu_{\mathcal{L}2} .\end{aligned}\tag{6.62}$$

The initial costates for the second phase are defined as

$$\begin{aligned}\lambda_{1b}(t_1) &= \sum \tilde{P}_{1bN} - \nu_1 , \\ \lambda_{2b}(t_1) &= \sum \tilde{P}_{2bN} - \nu_2 .\end{aligned}\tag{6.63}$$

The KKT condition that relates the two costate estimates (from (4.155) defines that the costates are continuous across the phase boundary, which is what is expected

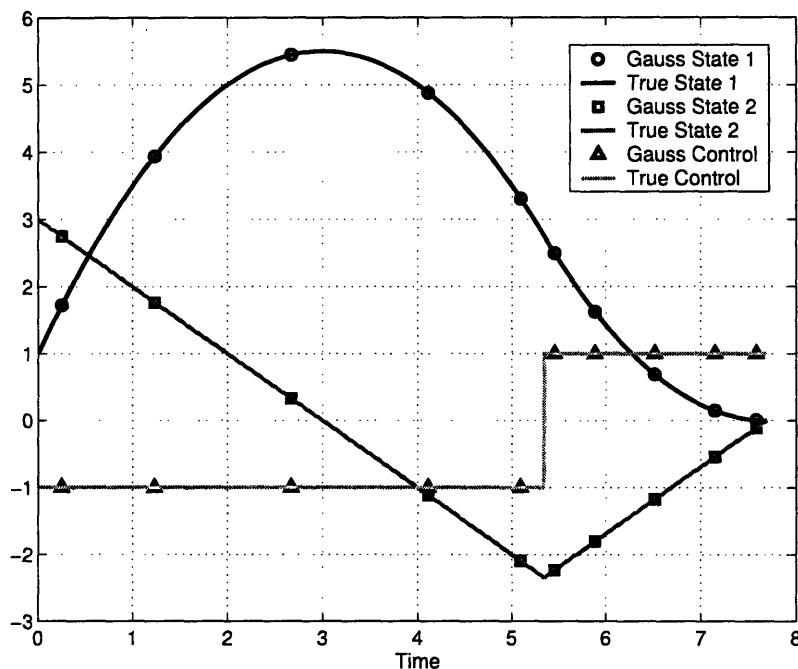


Figure 6-17: Bang-Bang State/Control Multiphase Solution

from the exact solution, so that

$$\begin{aligned} \lambda_{1a}(t_1) &= \lambda_{1b}(t_1) , \\ \lambda_{2a}(t_1) &= \lambda_{2b}(t_1) . \end{aligned} \tag{6.64}$$

The approximate solution for the states and control for the bang-bang multiphase problem, using 5 nodes per phase, along with the exact solution is plotted in Fig. 6-17. The approximate costates are shown in Fig. 6-18. The pseudospectral solution is exactly equivalent (within machine tolerances) to the true solution for the states, costates, control, and final/switching times, for any number of nodes used greater than two per phase, because the exact solution is a piecewise polynomial with a highest degree of two (in the first state), (6.41). Because the solution can be approximated exactly by polynomials, the pseudospectral solution can be exact for any number of nodes two or higher on each phase.

This example demonstrates that the integral pseudospectral method can be used

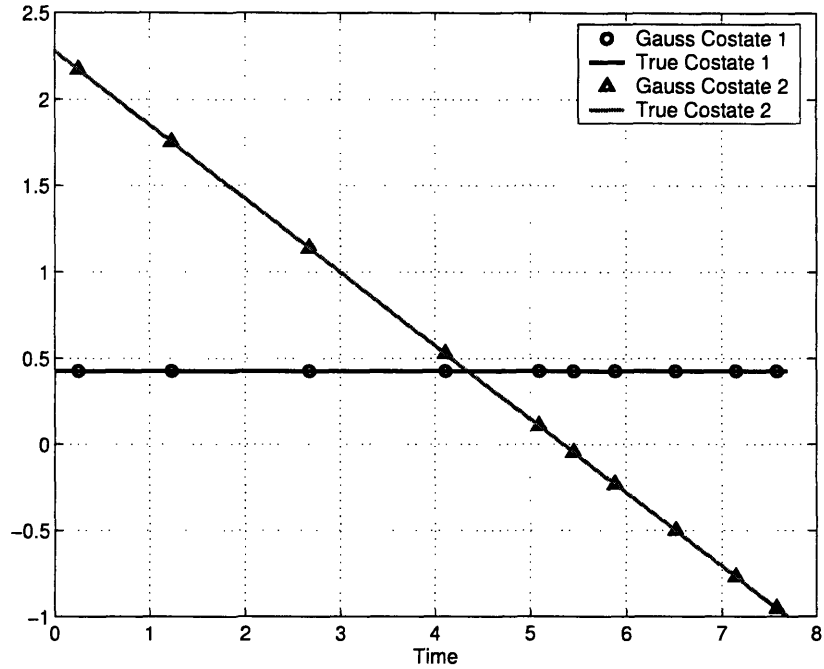


Figure 6-18: Bang-Bang Costate Multiphase Solution

on problems with control constraints to first determine the switching structure, and then with a multiphase formulation provide a much better result to the problem. This approach however, does not work as well for problems with state path constraints.

6.4 State Path Constraint Example

This example is a minimum energy problem with a second order state variable inequality constraint. This example appears in the text by Bryson and Ho [10]. The objective is to minimize the energy from the control,

$$J = \frac{1}{2} \int_0^1 u^2(t) dt, \quad (6.65)$$

subject to the dynamic constraints,

$$\begin{aligned}\dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= u(t),\end{aligned}\tag{6.66}$$

and boundary conditions,

$$\begin{aligned}x_1(0) &= 0, & x_1(1) &= 0, \\ x_2(0) &= 1, & x_2(1) &= -1,\end{aligned}\tag{6.67}$$

with the state path constraint,

$$x_1(t) \leq 0.1.\tag{6.68}$$

The solution to the optimal control problem for the states is determined to be

$$\begin{aligned}x_1(t) &= \begin{cases} 0.1 \left(1 - \left(1 - \frac{t}{3 \cdot 0.1} \right)^3 \right), & 0 \leq t \leq 0.3, \\ 0.1, & 0.3 \leq t \leq 0.7, \\ 0.1 \left(1 - \left(1 - \frac{1-t}{3 \cdot 0.1} \right)^3 \right), & 0.7 \leq t \leq 1, \end{cases} \\ x_2(t) &= \begin{cases} \left(1 - \frac{t}{3 \cdot 0.1} \right)^2, & 0 \leq t \leq 0.3, \\ 0, & 0.3 \leq t \leq 0.7, \\ \left(1 - \frac{1-t}{3 \cdot 0.1} \right)^2, & 0.7 \leq t \leq 1. \end{cases}\end{aligned}\tag{6.69}$$

The exact control is

$$u(t) = \begin{cases} -\frac{2}{3 \cdot 0.1} \left(1 - \frac{t}{3 \cdot 0.1} \right), & 0 \leq t \leq 0.3, \\ 0, & 0.3 \leq t \leq 0.7, \\ -\frac{2}{3 \cdot 0.1} \left(1 - \frac{1-t}{3 \cdot 0.1} \right), & 0.7 \leq t \leq 1. \end{cases}\tag{6.70}$$

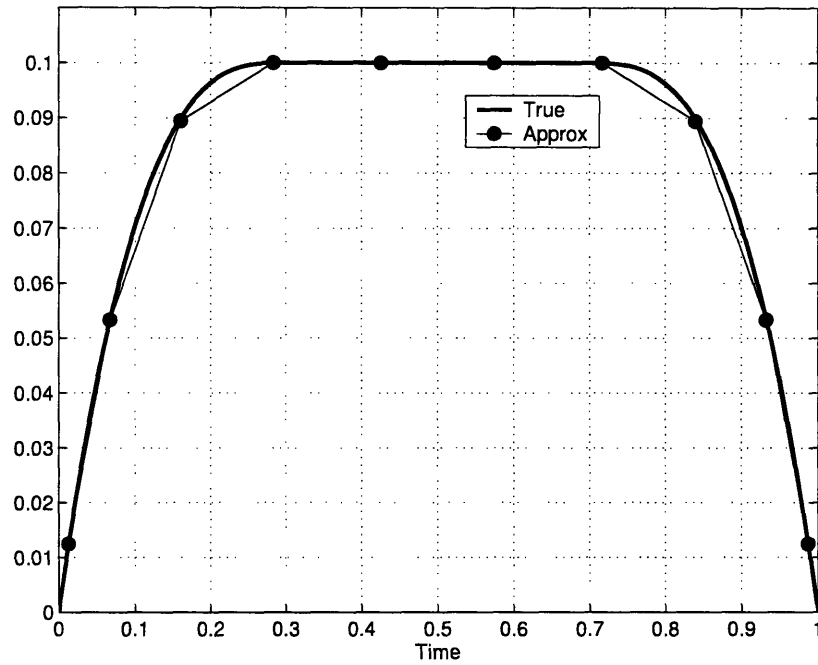


Figure 6-19: Path Constraint, State 1 Solution

The exact costates are

$$\lambda_1(t) = \begin{cases} \frac{2}{9 \cdot 0.01}, & 0 \leq t \leq 0.3, \\ 0, & 0.3 \leq t \leq 0.7, \\ -\frac{2}{9 \cdot 0.01}, & 0.7 \leq t \leq 1, \end{cases} \quad (6.71)$$

$$\lambda_2(t) = \begin{cases} \frac{2}{3 \cdot 0.1} \left(1 - \frac{t}{3 \cdot 0.1}\right), & 0 \leq t \leq 0.3, \\ 0, & 0.3 \leq t \leq 0.7, \\ -\frac{2}{3 \cdot 0.1} \left(1 - \frac{1-t}{3 \cdot 0.1}\right), & 0.7 \leq t \leq 1. \end{cases}$$

The plot of the exact states are shown in Fig. 6-19, 6-20, the control in Fig. 6-21, and costate in Fig. 6-22, 6-23.

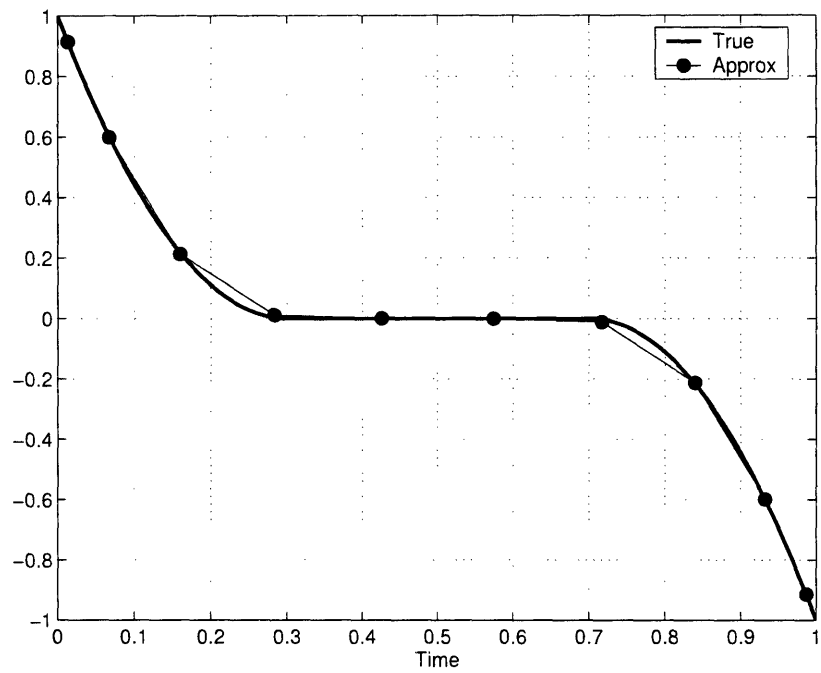


Figure 6-20: Path Constraint, State 2 Solution

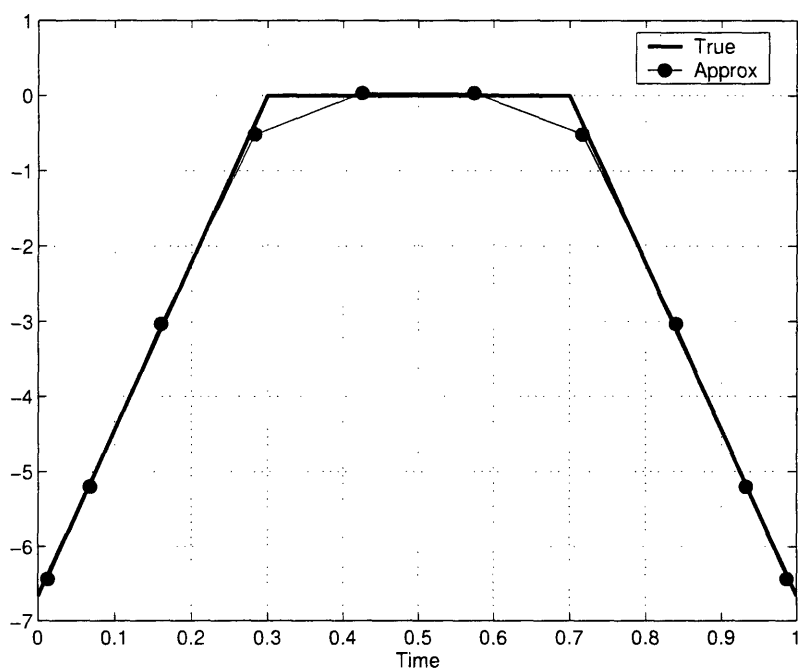


Figure 6-21: Path Constraint, Control Solution

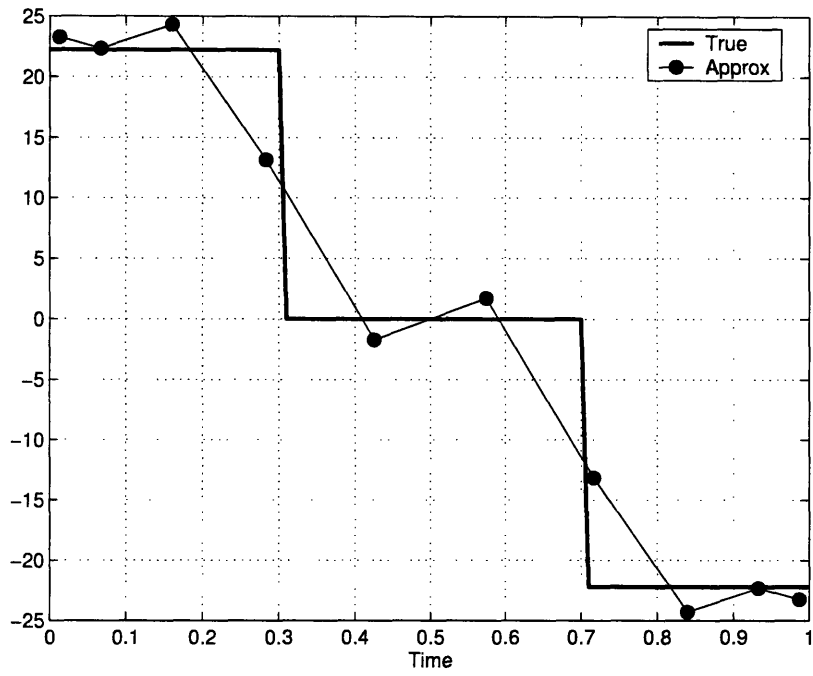


Figure 6-22: Path Constraint, Costate 1 Solution

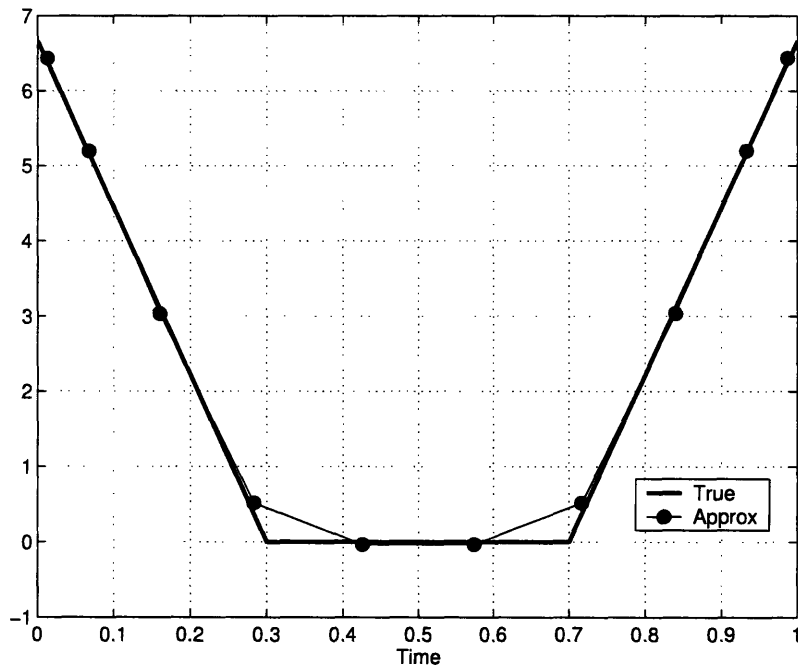


Figure 6-23: Path Constraint, Costate 2 Solution

6.4.1 Gauss Pseudospectral Method

The solution to the state path problem is found using the Gauss pseudospectral method using the variables $X_{1N}, X_{2N} \in \mathbb{R}^N$ for the states, and $U_N \in \mathbb{R}^N$ as the control at the Gauss points. The cost is to minimize

$$J = \frac{1}{4} w^T \cdot U_N, \quad (6.72)$$

subject to the constraints,

$$\begin{aligned} X_{1N} - \frac{1}{2} A \cdot X_{2N} &= 0, \\ X_{2N} - 1 - \frac{1}{2} A \cdot U_N &= 0, \end{aligned} \quad (6.73)$$

and terminal conditions,

$$\begin{aligned} \frac{1}{2} w^T \cdot X_{2N} &= 0, \\ 2 + \frac{1}{2} w^T \cdot U_N &= 0. \end{aligned} \quad (6.74)$$

The path constraint is enforced at all the collocation points.

$$X_{1N} \leq 0.1 \quad (6.75)$$

The resulting NLP was solved with 10 nodes using SNOPT [30], and the costates were estimated using the KKT multipliers. The costate estimates are

$$\begin{aligned} \lambda_{1N} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1N} - \nu_1, \\ \lambda_{2N} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2N} - \nu_2, \end{aligned} \quad (6.76)$$

where $\tilde{P}_{1N}, \tilde{P}_{2N} \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints (6.73), and $\nu_1, \nu_2 \in \mathbb{R}$ are the Lagrange multipliers associated with the terminal constraints (6.74). The approximate solution is shown in Fig. 6-19, 6-20 for the states, Fig. 6-21 for the control, and Fig. 6-22, 6-23 for the costates.

6.4.2 State Path Constraint Problem Convergence

The approximate solution for the states seems to be a good approximation to the exact states, Fig. 6-19, 6-20, but the approximate control and costates, Fig. 6-21, 6-22, 6-23, do not do a good job of approximating the exact solution, especially the first costate. Because the problem has a path constraint, the first costate is discontinuous, and the second costate and control have discontinuous derivatives. Consequently, the second state has a discontinuous second derivative and the first state a discontinuous third derivative. The approximating polynomials do not do as well in approximating functions that are discontinuous or have discontinuous derivatives.

A convergence plot of the infinity norm error of the approximate solution versus the number of nodes is shown in Fig. 6-24. The figure shows that the approximate solution does not improve much as the number of nodes is increased. This result is because of the discontinuities in the problem solution. The resulting approximating polynomials in the approximate solution exhibit a Gibbs like phenomenon, which causes them to converge very slowly to the exact solution. An attempt to correct for the deficiencies of the approximating polynomial is made by using the multiple phase approach.

6.4.3 Multiphase Solution

The multiple phase approach is used on the state path problem by breaking the problem into three phases. On the first phase, the state is off the path constraint, the second it is on the constraint, and finally the state is again off the constraint on the final phase. The variables used for the multiphase approach are $X_{1Na}, X_{2Na} \in \mathbb{R}^N$ for the states on the first phase, $X_{1Nb}, X_{2Nb} \in \mathbb{R}^N$ for the states on the second phase, and $X_{1Nc}, X_{2Nc} \in \mathbb{R}^N$ for the states on the final phase. The control for the three phases is $U_{Na}, U_{Nb}, U_{Nc} \in \mathbb{R}^N$, with $t_1, t_2 \in \mathbb{R}$ as the times of the phase boundaries. The states at the switching time, $X_{1bo}, X_{1co}, X_{2bo}, X_{2co} \in \mathbb{R}$ are used in the phase boundary constraints to link the phases.

The total cost to be minimized is the sum of the integrals on the three phases, st

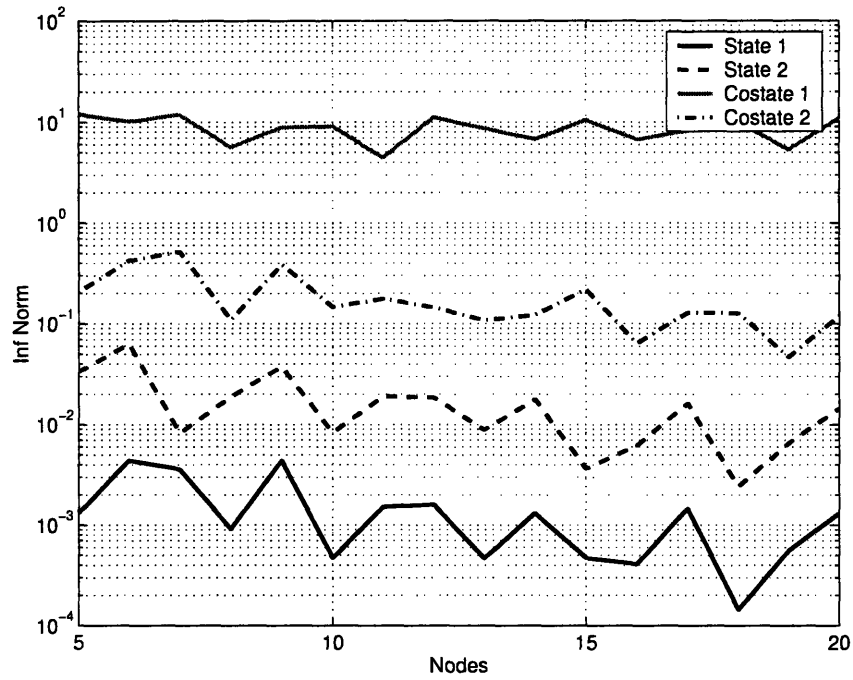


Figure 6-24: Path Constraint, Convergence

that

$$J = \frac{1}{2} \left(\frac{t_1}{2} w^T \cdot U_{Na} + \frac{(t_2 - t_1)}{2} w^T \cdot U_{Nb} + \frac{(1 - t_2)}{2} w^T \cdot U_{Nc} \right). \quad (6.77)$$

The dynamic constraints on the first phase are approximated as

$$\begin{aligned} X_{1Na} - \frac{t_1}{2} A \cdot X_{2Na} &= 0, \\ X_{2Na} - 1 - \frac{t_1}{2} A \cdot U_{Na} &= 0. \end{aligned} \quad (6.78)$$

On the second phase the dynamic constraints are

$$\begin{aligned} X_{1Nb} - X_{1bo} - \frac{(t_2 - t_1)}{2} A \cdot X_{2Nb} &= 0, \\ X_{2Nb} - X_{2bo} - \frac{(t_2 - t_1)}{2} A \cdot U_{Nb} &= 0, \end{aligned} \quad (6.79)$$

and on the final phase,

$$\begin{aligned} X_{1Nc} - X_{1co} - \frac{(1-t_2)}{2}A \cdot X_{2Nc} &= 0, \\ X_{2Nc} - X_{2co} - \frac{(1-t_2)}{2}A \cdot U_{Nc} &= 0. \end{aligned} \tag{6.80}$$

The terminal constraints are

$$\begin{aligned} X_{1co} + \frac{(1-t_2)}{2}w^T \cdot X_{2Nc} &= 0, \\ X_{2co} + \frac{(1-t_2)}{2}w^T \cdot U_{Nc} - 1 &= 0, \end{aligned} \tag{6.81}$$

with phase boundary conditions for the second and third phase,

$$\begin{aligned} X_{1bo} + \frac{(t_2-t_1)}{2}w^T \cdot X_{2Nb} - X_{1co} &= 0, \\ X_{2bo} + \frac{(t_2-t_1)}{2}w^T \cdot U_{Nb} - X_{2co} &= 0, \end{aligned} \tag{6.82}$$

and phase boundary conditions for the first and second phase,

$$\begin{aligned} \frac{t_1}{2}w^T \cdot X_{2Na} - X_{1bo} &= 0, \\ 1 + \frac{t_1}{2}w^T \cdot U_{Na} - X_{2bo} &= 0. \end{aligned} \tag{6.83}$$

The path constraints are enforced as inequalities on the first and third phases, so that

$$X_{1Na} \leq 0.1, \quad X_{1Nc} \leq 0.1, \tag{6.84}$$

and the path constraint is forced to be active on the second phase, so that

$$X_{1Nb} = 0.1. \tag{6.85}$$

The initial state on the second and third phases were also forced to be on the constraint, so that

$$X_{1bo} = 0.1, \quad X_{1co} = 0.1. \tag{6.86}$$

The costates for the multiphase solution were found using the KKT multipliers of the NLP. On the first phase they are

$$\begin{aligned}\lambda_{1Na} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1Na} - \nu_{1a} , \\ \lambda_{2Na} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2Na} - \nu_{2a} ,\end{aligned}\tag{6.87}$$

where $\tilde{P}_{1Na}, \tilde{P}_{2Na} \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints (6.78) and $\nu_{1a}, \nu_{2a} \in \mathbb{R}$ are the Lagrange multipliers associated with the phase boundary constraints (6.83). The costates on the second phase are

$$\begin{aligned}\lambda_{1Nb} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1Nb} - \nu_{1b} , \\ \lambda_{2Nb} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2Nb} - \nu_{2b} ,\end{aligned}\tag{6.88}$$

where $\tilde{P}_{1Nb}, \tilde{P}_{2Nb} \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints (6.79) and $\nu_{1b}, \nu_{2b} \in \mathbb{R}$ are the Lagrange multipliers associated with the phase boundary constraints (6.82). The costates on the final phase are

$$\begin{aligned}\lambda_{1Nc} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1Nc} - \nu_{1c} , \\ \lambda_{2Nc} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2Nc} - \nu_{2c} ,\end{aligned}\tag{6.89}$$

where $\tilde{P}_{1Nc}, \tilde{P}_{2Nc} \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints (6.80) and $\nu_{1c}, \nu_{2c} \in \mathbb{R}$ are the Lagrange multipliers associated with the terminal constraints (6.81).

The solution to the multiphase formulation of the state path constraint problem was closer to the exact solution than the single phase solution, but did not show the significant improvement of the bang-bang problem. The convergence plot of the solution errors (Fig. 6-25) shows improved convergence over the single phase formulation (Fig. 6-24) but it is not as quick as expected. The exact solution is a piecewise polynomial of highest degree of three. In theory, three nodes per phase should be enough to get the exact solution.

The reason for the lack of significant improvement can be seen by looking at the

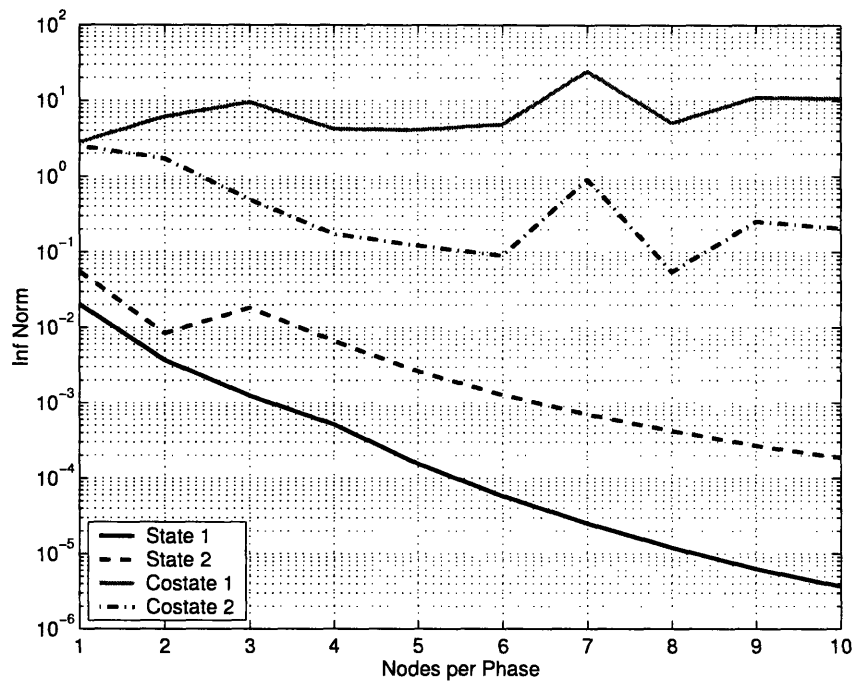


Figure 6-25: Path Constraint, Multiphase Convergence

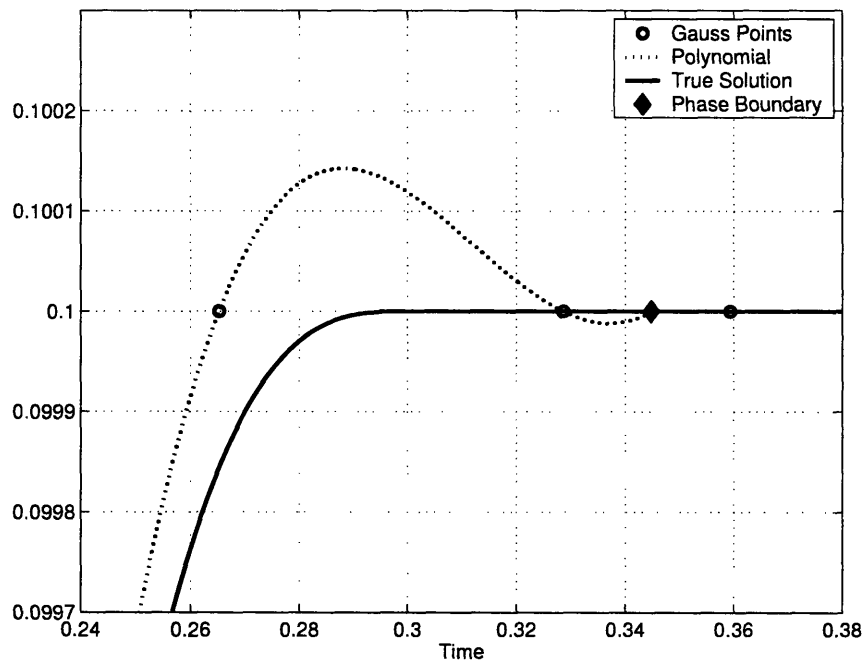


Figure 6-26: Path Constraint, State 1 Multiphase Solution

phase boundaries of the first state. A zoomed in plot of the approximate solution for the first state, using five nodes per phase, is shown in Fig. 6-26. The plot shows the approximate solution at the Gauss points, but also the interpolating polynomial derived from these points. The polynomial is

$$X_1(t) = X_1(0) \cdot L_0(t) + \sum_{k=1}^N X_1(t_k) \cdot L_k(t) . \quad (6.90)$$

The plot shows that the interpolating polynomial violates the state path constraint between the collocation points. Even though the path constraint is satisfied at the collocation points, the solver is able to “cheat” by exceeding the constraint between the points. The violation of the constraint allows the solver to find a solution that has a lower cost, 4.4405, than the true solution, 4.4444. As the number of nodes is increased, the resulting polynomial can not exceed the constraint in as many places so the approximate solution tends toward the true solution, as shown in the convergence plot Fig. 6-25.

This example shows that using the multiphase approach for solving problems that have state path constraints with active and inactive sections does not work as well as the multiphase approach for problems with bang-bang control.

6.5 Brachistochrone Problem

The next example considered is the brachistochrone problem. The brachistochrone problem was proposed and solved by Johann Bernoulli in 1696, and is one of the earliest applications of the calculus of variations [48]. The objective is to determine the optimal path that a bead would take sliding down a frictionless wire to reach a given horizontal position in the minimum possible time, Fig. 6-27.

The equations of motion are [10]

$$\dot{x} = \sqrt{2 \cdot g \cdot y} \cdot \cos \theta, \quad \dot{y} = \sqrt{2 \cdot g \cdot y} \cdot \sin \theta , \quad (6.91)$$

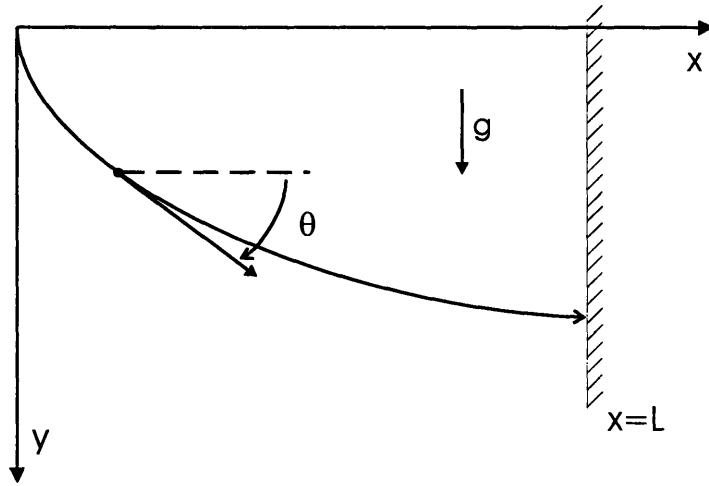


Figure 6-27: Brachistochrone Problem

with initial conditions,

$$x(0) = 0, \quad y(0) = 0, \quad (6.92)$$

and terminal condition,

$$x(t_f) = L. \quad (6.93)$$

The true solution for the control angle is

$$\theta(t) = \frac{\pi}{2} - \omega \cdot t, \quad (6.94)$$

where $\omega = \sqrt{\frac{\pi \cdot g}{4 \cdot L}}$ and the final time is $\sqrt{\frac{\pi \cdot L}{g}}$. The resulting states are

$$\begin{aligned} x(t) &= \frac{2 \cdot L}{\pi} \left(\omega \cdot t - \frac{\sin(2 \cdot \omega \cdot t)}{2} \right), \\ y(t) &= \frac{2 \cdot L}{\pi} (\sin(\omega \cdot t))^2. \end{aligned} \quad (6.95)$$

The costates for the brachistochrone problem are

$$\begin{aligned}\lambda_x &= \frac{-\omega}{g}, \\ \lambda_y &= \frac{-\omega}{g} \cdot \cot(\omega \cdot t).\end{aligned}\tag{6.96}$$

The difficulty in solving the brachistochrone problem numerically comes from the fact that the second costate is infinite at $t = 0$.

The brachistochrone problem is solved using the Gauss pseudospectral method by defining an NLP where the objective is to find the variables $X_N \in \mathbb{R}^N$, $Y_N \in \mathbb{R}^N$, $\theta_N \in \mathbb{R}^N$, and t_f to minimize the final time,

$$J = t_f,\tag{6.97}$$

subject to the discretized dynamic constraints in integral form,

$$\begin{aligned}X_N - \frac{t_f}{2}A \cdot \left(\sqrt{2 \cdot g \cdot Y_N} \cdot \cos \theta_N\right) &= 0, \\ Y_N - \frac{t_f}{2}A \cdot \left(\sqrt{2 \cdot g \cdot Y_N} \cdot \sin \theta_N\right) &= 0.\end{aligned}\tag{6.98}$$

Note that the square root, trigonometric functions, and multiplication within the parenthesis is performed term by term. The terminal condition is expressed using a Gauss quadrature, so that

$$\frac{t_f}{2}w^T \cdot \left(\sqrt{2 \cdot g \cdot Y_N} \cdot \cos \theta_N\right) - L = 0.\tag{6.99}$$

Note again that the operations with the parenthesis are performed term by term. The costate estimates are computed using the KKT multipliers $\tilde{P}_x, \tilde{P}_y \in \mathbb{R}^N$ associated with the dynamic constraints (6.98), and the Lagrange multiplier $\nu \in \mathbb{R}$ associated with the boundary condition (6.99) from the NLP solver. The costate estimates are

$$\begin{aligned}\Lambda_x &= W^{-1} \cdot A^T \cdot \tilde{P}_x + \nu, \\ \Lambda_y &= W^{-1} \cdot A^T \cdot \tilde{P}_y,\end{aligned}\tag{6.100}$$

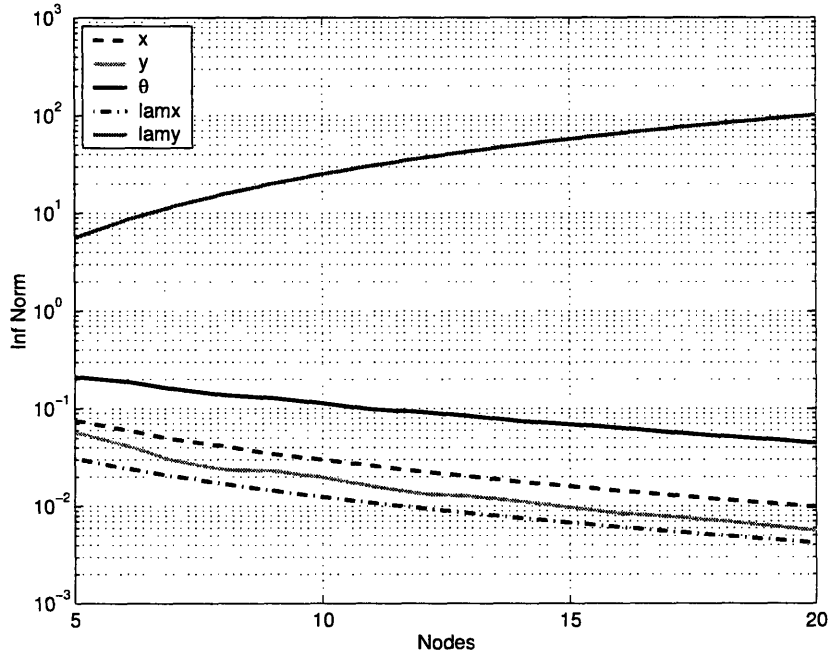


Figure 6-28: Brachistochrone Convergence

where $W \in \mathbb{R}^{N \times N}$ is a diagonal matrix of the Gauss weights.

The NLP was solved using the constants $g = L = 1$. The convergence properties of the approximate solution are shown in Fig. 6-28. The figure shows that the convergence rate on the states, control, and first costate is slow and it appears that the second costate does not converge at all. This result can be attributed to the fact that the exact solution for the second costate (6.96) is infinite at $t = 0$. Because the states and costates are approximated using interpolating polynomials, the error in the solution can be expected to be related to the error in the interpolating polynomial of the exact solution. The error in the Lagrange polynomial interpolating formula is

$$E(t) = \frac{1}{N!} \prod_{i=1}^N (t - t_k) \cdot \frac{d^{(N)} f}{dt^{(N)}}(\zeta), \quad (6.101)$$

where t_k are the Gauss collocation points and $\zeta \in [0, t_f]$. The N th derivative of the

second costate can be found to be approximately

$$\left| \frac{d^N \lambda_y}{dt^N}(\zeta) \right| \approx \frac{N! \cdot \omega^N}{g} (\cot(\omega \cdot \zeta))^N, \quad (6.102)$$

which could be very large especially if ζ is close to zero. This equation indicates that the error in the polynomial interpolation of the second costate could be growing as the number of nodes is increased which leads to the poor convergence of the states, costates, and control as shown in Fig. 6-28.

The effect of the infinite costate can be reduced by redefining the problem to start at a different point on the optimal path. This approach also shows the effect of having the infinite point nearby the solution time interval. The initial starting time can be chosen to be t_o , with the initial position of the bead defined by the exact solution of the states,

$$\begin{aligned} x(t_o) &= \frac{2 \cdot L}{\pi} \left(\omega \cdot t_o - \frac{\sin(2 \cdot \omega \cdot t_o)}{2} \right), \\ y(t_o) &= \frac{2 \cdot L}{\pi} (\sin(\omega \cdot t_o))^2. \end{aligned} \quad (6.103)$$

The problem is solved similarly to before with the addition of the non-zero initial conditions. The state dynamic constraints are approximated by

$$\begin{aligned} X_N - x(t_o) - \frac{(t_f - t_o)}{2} A \cdot \left(\sqrt{2 \cdot g \cdot Y_N} \cdot \cos \theta_N \right) &= 0, \\ Y_N - y(t_o) - \frac{(t_f - t_o)}{2} A \cdot \left(\sqrt{2 \cdot g \cdot Y_N} \cdot \sin \theta_N \right) &= 0, \end{aligned} \quad (6.104)$$

with the terminal constraint approximated as

$$x(t_o) + \frac{(t_f - t_o)}{2} w^T \cdot \left(\sqrt{2 \cdot g \cdot Y_N} \cdot \cos \theta_N \right) - L = 0. \quad (6.105)$$

The convergence rate of of the second costate using the approach is shown in Fig. 6-29 for several starting times t_o . It can be seen that as the starting time gets larger, the convergence rate of the costate gets better. This result makes intuitive sense because the error in the interpolating polynomial for the second costate will be smaller as the initial start time t_o gets large. The convergence rate for the control

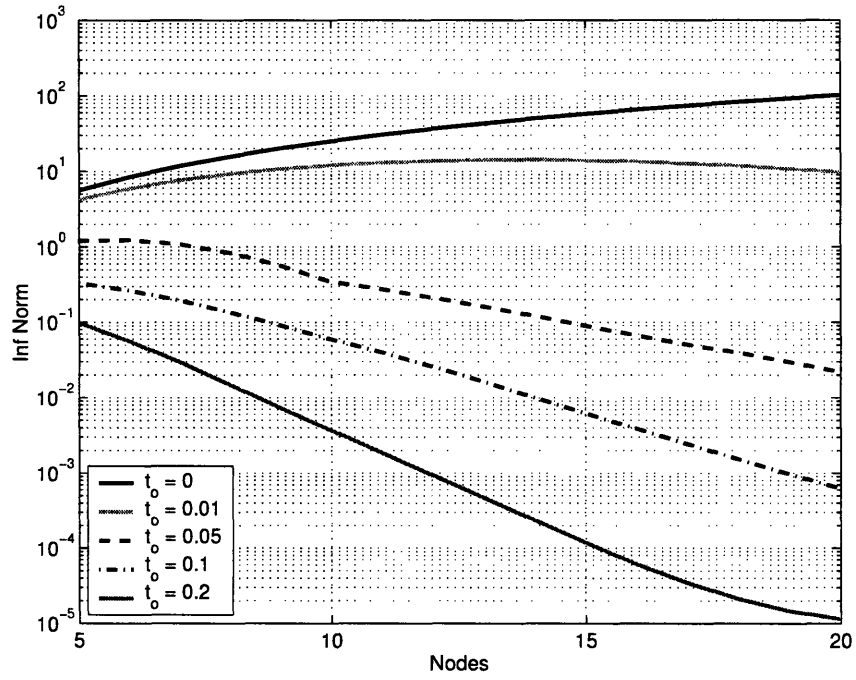


Figure 6-29: Brachistochrone Costate y Error

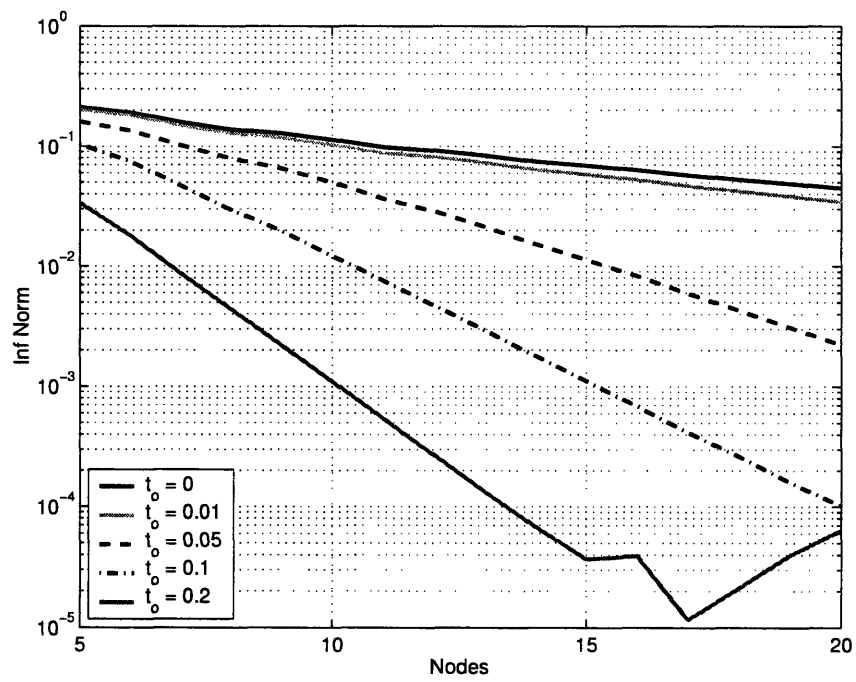


Figure 6-30: Brachistochrone Control Error

is also shown in Fig. 6-30. This figure also indicates that the convergence rate gets better as the problem gets further from the point where the costate is infinite.

This example shows that the convergence rate can be slow, if there is convergence at all, for problems whose exact solutions are infinite at any point near the time interval of the problem. The convergence rate for the approximate solution of the states, costates, and controls are all effected by the infinite point. This effect could potentially be a difficulty, as it may not be obvious that a costate could be infinite for some optimal control problems.

6.6 Singular Arc Example

The next example considered is a problem that contains a singular arc.

In an optimal control problem, if there is a finite interval of time in which Pontryagin's maximum principle (2.75) does not define the relation between the control, states, and costates, then the problem is said to be singular ([10] [45] [48]). One example of a singular problem is found in [45]. The problem is to find the states, $x_1(t), x_2(t) \in \mathbb{R}$, and control, $u(t) \in \mathbb{R}$, that minimize the quadratic cost with free final time. The cost is

$$J = \frac{1}{2} \int_0^{t_f} (x_1^2(t) + x_2^2(t)) dt, \quad (6.106)$$

with linear dynamics,

$$\begin{aligned} \frac{dx_1}{dt} &= x_2(t), \\ \frac{dx_2}{dt} &= u(t), \end{aligned} \quad (6.107)$$

and constrained control,

$$|u(t)| \leq 1. \quad (6.108)$$

The boundary conditions used for the example are

$$\begin{aligned} x_1(0) &= 0.8, & x_1(t_f) &= 0.01, \\ x_2(0) &= -0.8, & x_2(t_f) &= -0.01. \end{aligned} \quad (6.109)$$

The problem can be shown to be singular, and the exact solution is found by using the fact that the Hamiltonian must be zero. The solution is given by

$$\begin{aligned}x_1(t) &= 0.8 \cdot e^{-t}, \\x_2(t) &= -0.8 \cdot e^{-t},\end{aligned}\tag{6.110}$$

with control

$$u(t) = 0.8 \cdot e^{-t},\tag{6.111}$$

and costates

$$\begin{aligned}\lambda_1(t) &= 0.8 \cdot e^{-t}, \\ \lambda_2(t) &= 0.\end{aligned}\tag{6.112}$$

The final time is determined to be $t_f = -\ln(\frac{1}{80})$. The exact solution to the singular problem is shown in Fig. 6-31 for the states, Fig. 6-32 for the control, and Fig. 6-33 for the costates.

The pseudospectral solution to the singular problem is found by defining the states at the Gauss points, $X_{1N}, X_{2N} \in \mathbb{R}^N$, and control at the Gauss points, $U_N \in \mathbb{R}^N$. The cost is approximated using a Gauss quadrature, so that

$$J = \frac{1}{2} \cdot \frac{t_f}{2} \cdot w^T \cdot (X_{1N}^2 + X_{2N}^2).\tag{6.113}$$

The state dynamics are approximated in integral form as

$$\begin{aligned}X_{1N} - 0.8 - \frac{t_f}{2} A \cdot X_{2N} &= 0, \\ X_{2N} + 0.8 - \frac{t_f}{2} A \cdot U_N &= 0.\end{aligned}\tag{6.114}$$

The terminal conditions are enforced using a Gauss quadrature, so that

$$\begin{aligned}0.01 - 0.8 - \frac{t_f}{2} w^T \cdot X_{2N} &= 0, \\ -0.01 + 0.8 - \frac{t_f}{2} w^T \cdot U_N &= 0\end{aligned}\tag{6.115}$$

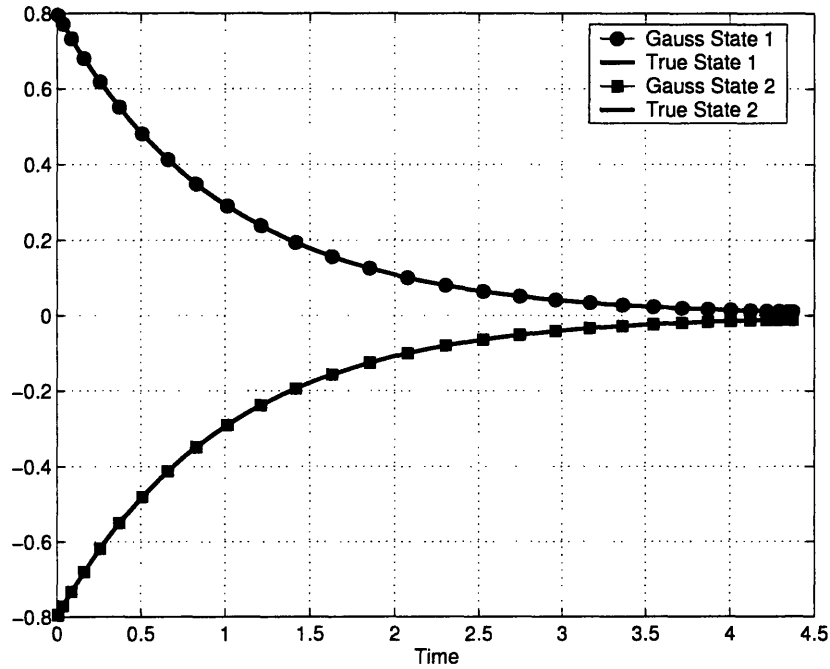


Figure 6-31: Singular Problem State Solution

Finally, the control constraint is enforced at the Gauss points using

$$-1 \leq U_N \leq 1. \quad (6.116)$$

The NLP is then solved for the states, control, and final time that minimize the cost (6.113) subject to the constraints (6.114 - 6.116).

The costates were estimated using the KKT multipliers, $\tilde{P}_{1N}, \tilde{P}_{2N} \in \mathbb{R}^N$, associated with the constraints (6.114), and Lagrange multipliers, $\nu_1, \nu_2 \in \mathbb{R}$, associated with the terminal constraints (6.115). The costate estimates are

$$\begin{aligned} \lambda_{1N} &= W^{-1} \cdot A^T \cdot \tilde{P}_{1N} - \nu_1, \\ \lambda_{2N} &= W^{-1} \cdot A^T \cdot \tilde{P}_{2N} - \nu_2. \end{aligned} \quad (6.117)$$

The approximate solution for the states, control, and costates using 30 nodes is shown in Fig. 6-31 - 6-33.

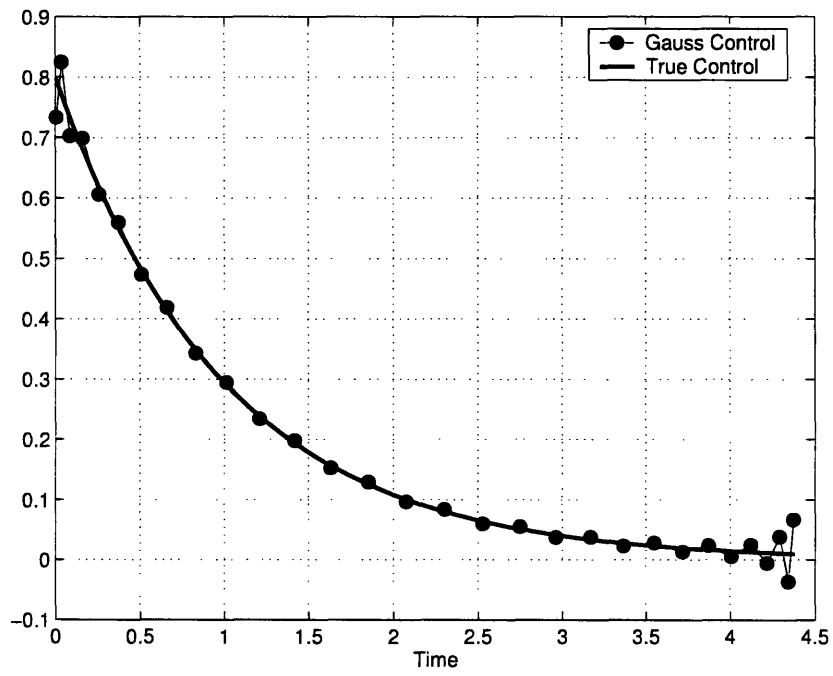


Figure 6-32: Singular Problem Control Solution

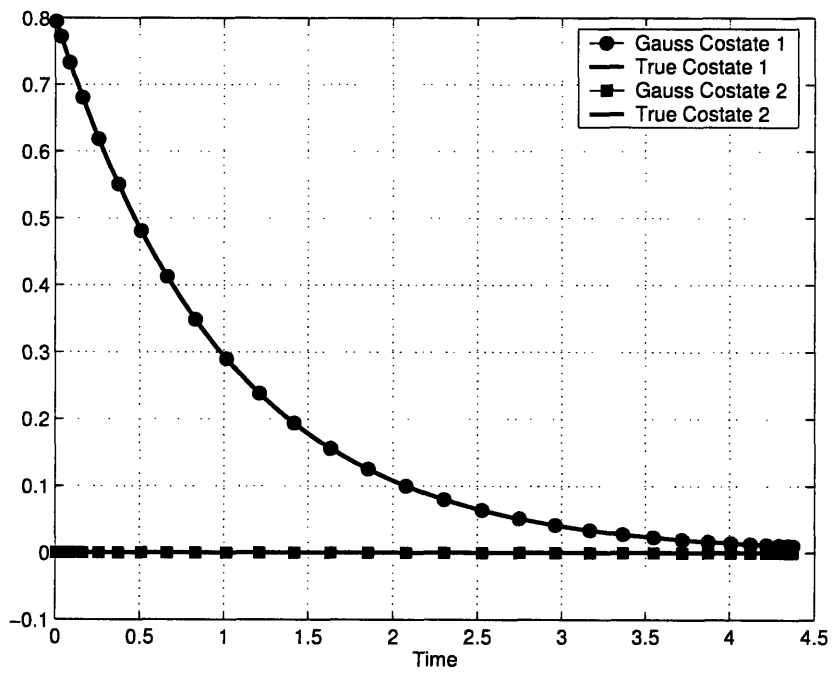


Figure 6-33: Singular Problem Costate Solution

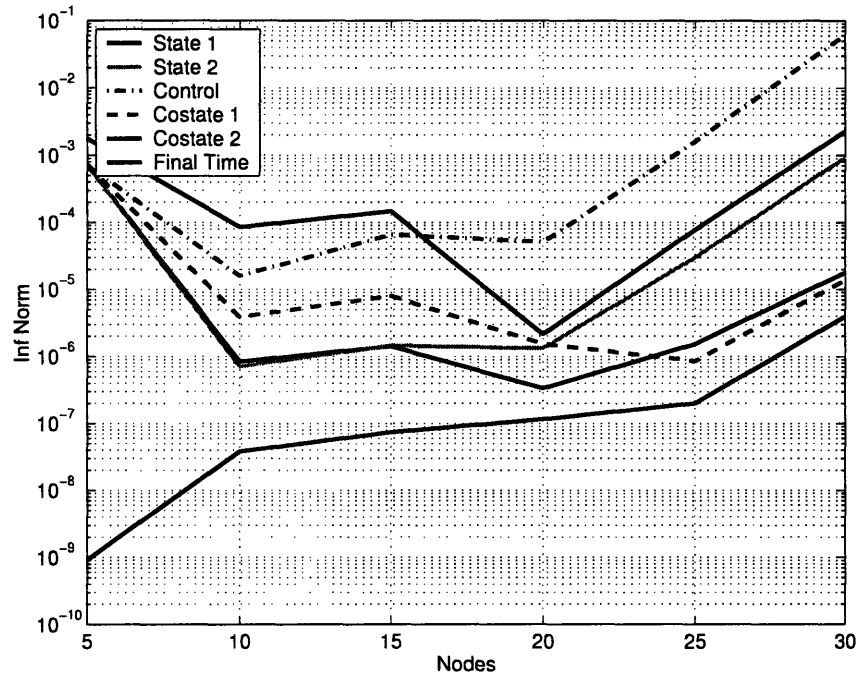


Figure 6-34: Singular Problem Convergence

The figures show that the approximate solution for the states and costates is a relatively good approximation to the exact solution. However, the error in the control is significant. The source of the error comes from the fact that the NLP solution is satisfying a set of conditions (the discrete form of Pontryagin's maximum principle) that do not give any information about the control.

A plot of the convergence of the pseudospectral solution is shown in Fig. 6-34. The convergence plot shows that the approximate solution is not getting closer to the exact solution as the number of nodes is increased. This example demonstrates that the Gauss pseudospectral method is not well suited for solving problem with singular arcs. The resulting NLP does not have enough information to accurately determine the control on a singular interval.

6.7 Multiple Solution Example

The problem considered here is an optimal control problem with two distinct local minimum [25]. The problem has one state and one control, $x(t), u(t) \in \mathbb{R}$, where the objective is to maximize the square of the final state, $x(t_f)$, or to minimize

$$J = -x^2(t_f) . \quad (6.118)$$

The dynamic constraints are

$$\frac{dx}{dt} = -x^2 + u , \quad (6.119)$$

with initial condition, $x(0) = 9$. The variables have bounds defined by

$$\begin{aligned} -5 &\leq u(t) \leq 5 , \\ -12 &\leq x(t) \leq 9 , \\ t &\in [0, 1] . \end{aligned} \quad (6.120)$$

The exact solution for the global minimum for the problem is shown in Fig. 6-35. Note that there also exists another local minimum shown in Fig. 6-36.

The Gauss pseudospectral solution was found by defining the NLP in terms of the state and control, $X_N, U_N \in \mathbb{R}^N$, at the Gauss points. The cost function to be minimized is

$$J = - \left[9 + \frac{1}{2} w^T \cdot (-X_N^2 + U_N) \right]^2 , \quad (6.121)$$

subject to the dynamic constraints,

$$X_N - 9 - \frac{1}{2} A \cdot (-X_N^2 + U_N) = 0 , \quad (6.122)$$

and variable bounds,

$$\begin{aligned} -5 &\leq U_N \leq 5 , \\ -12 &\leq X_N \leq 9 . \end{aligned} \quad (6.123)$$

There were two solutions to the NLP found, shown in Fig. 6-35, 6-36. The solution

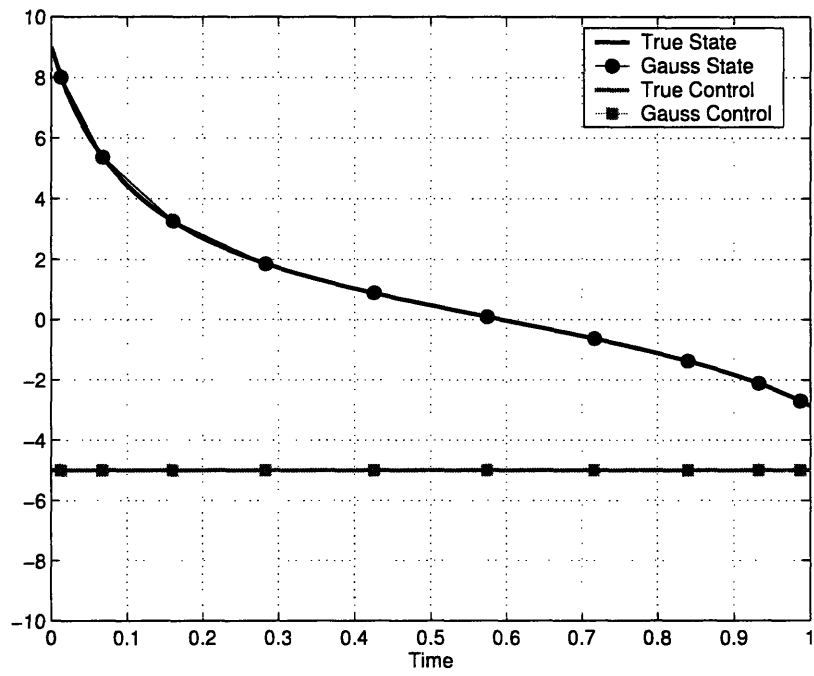


Figure 6-35: Multiple Solution 1 State/Control

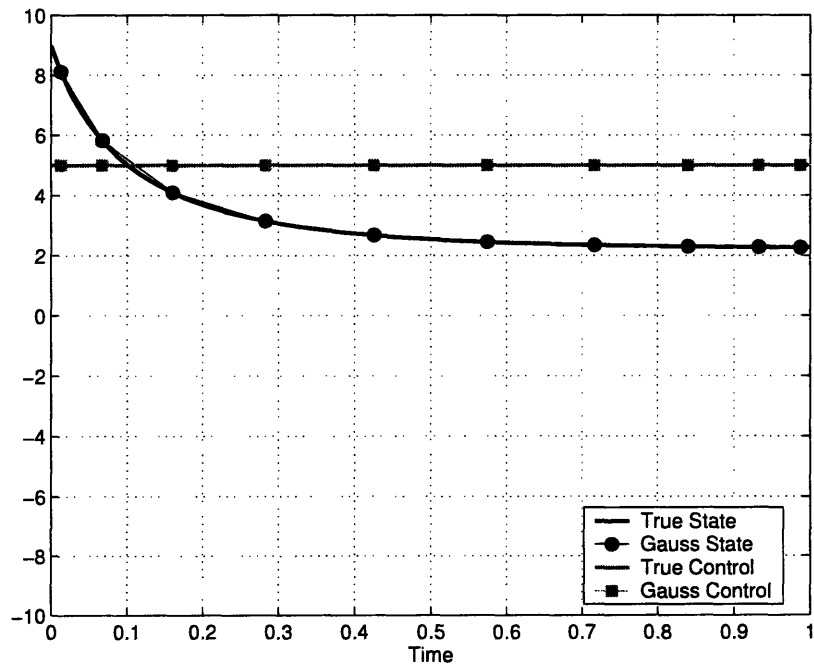


Figure 6-36: Multiple Solution 2 State/Control

found was highly dependent on the initial guess and number of nodes used. As seen in the plots, one of the solutions is approximating one of the local minimum, and the other is approximating the other local minimum.

This example illustrates that the NLP resulting from the integral pseudospectral method may have several local minimum. This result is because the solution approximates the first-order necessary conditions of an optimal control problem, which only defines local extremals. For control problems that may have several stationary points, the pseudospectral method can only identify local extremals and give no information about global results. Care must be taken in determining an initial guess for an NLP on problems that have several local solutions.

6.8 Low-Thrust Orbit Transfer Problem

The final example considered here is a low-thrust orbit transfer problem that first appears in [50]. The problem has been solved numerically in many places, ([10] [11] [22] [33] [67]). The objective is to transfer a spacecraft from a circular orbit using a low thrust engine to the largest possible circular orbit in fixed time. The problem is described using the states $r(t), u(t), v(t) \in \mathbb{R}$, for the radial position, radial velocity, and tangential velocity respectively. The control $\phi(t) \in \mathbb{R}$, is the angle between the thrust vector and tangential velocity. All motion is considered planar. The cost is to maximize the final radius, or to minimize

$$J = -r(t_f). \quad (6.124)$$

The dynamics describing the motion are

$$\begin{aligned} \frac{dr}{dt} &= u, \\ \frac{du}{dt} &= \frac{v^2}{2} - \frac{\mu}{r^2} + \frac{T \cdot \sin \phi}{m_o - |\dot{m}|t}, \\ \frac{dv}{dt} &= -\frac{u \cdot v}{r} + \frac{T \cdot \cos \phi}{m_o - |\dot{m}|t}, \end{aligned} \quad (6.125)$$

where μ is the gravitational parameter, T is the thrust magnitude, m_o is the initial spacecraft mass, and \dot{m} is the fuel consumption rate. The boundary conditions used to constrain the initial and final orbits to be circular are

$$\begin{aligned} r(0) &= r_o, \\ u(0) &= 0, & u(t_f) &= 0, \\ v(0) &= \sqrt{\frac{\mu}{r_o}}, & v(t_f) - \sqrt{\frac{\mu}{r(t_f)}} &= 0. \end{aligned} \tag{6.126}$$

The states and mass are normalized, and the constants used are

$$T = 0.1405, \quad \dot{m} = 0.0749, \quad m_o = r_o = \mu = 1, \quad t_f = 3.32. \tag{6.127}$$

6.8.1 Legendre Pseudospectral Method

The Legendre pseudospectral transcription was used to solve the orbit problem using the variables $R_N, U_N, V_N, \phi_N \in \mathbb{R}^N$ for the states and control at the LGL points. The cost to be minimized is the negative of the final radius, which is the last entry in the state vector. The cost is

$$J = -R_N[N]. \tag{6.128}$$

The dynamic constraints are approximated using the differential matrix. The right hand sides of the dynamics are calculated at the LGL points for simplification. The dynamics are

$$\begin{aligned} F_R &= U_N, \\ F_U &= \frac{V_N^2}{2} - \frac{\mu}{R_N^2} + \frac{T \cdot \sin \phi_N}{m_o - |\dot{m}|t_N}, \\ F_V &= -\frac{U_N \cdot V_N}{R_N} + \frac{T \cdot \cos \phi_N}{m_o - |\dot{m}|t_N}, \end{aligned} \tag{6.129}$$

where t_N is the vector of time at the LGL points. Note that all vector math in (6.129) is performed term by term. The dynamic constraints of the NLP are then

$$\begin{aligned}\frac{2}{t_f}D \cdot R_N - F_R &= 0, \\ \frac{2}{t_f}D \cdot U_N - F_U &= 0, \\ \frac{2}{t_f}D \cdot V_N - F_V &= 0,\end{aligned}\tag{6.130}$$

with the boundary constraints,

$$\begin{aligned}R_N[1] - r_o &= 0, \\ U_N[1] &= 0, & U_N[N] &= 0, \\ V_N[1] - \sqrt{\frac{\mu}{r_o}} &= 0, & V_N[N] - \sqrt{\frac{\mu}{R_N[N]}} &= 0.\end{aligned}\tag{6.131}$$

The solution for the states and control, Fig. 6-37, 6-38, was found by solving the resulting NLP with SNOPT [30].

The costate estimates, $\Lambda_R, \Lambda_U, \Lambda_V \in \mathbb{R}^N$ (Fig. 6-39), was found at the LGL points using the KKT multipliers, $\tilde{\Lambda}_R, \tilde{\Lambda}_U, \tilde{\Lambda}_V \in \mathbb{R}^N$, from the NLP solution. The costate estimates are

$$\begin{aligned}\Lambda_R &= \frac{2}{t_f}W^{-1} \cdot \tilde{\Lambda}_R, \\ \Lambda_U &= \frac{2}{t_f}W^{-1} \cdot \tilde{\Lambda}_U, \\ \Lambda_V &= \frac{2}{t_f}W^{-1} \cdot \tilde{\Lambda}_V.\end{aligned}\tag{6.132}$$

6.8.2 Gauss Pseudospectral Method

The Gauss pseudospectral transcription of the orbit problem is done using the variables $R_N, U_N, V_N, \phi_N \in \mathbb{R}^N$ for the states and control at Gauss points. For simplicity

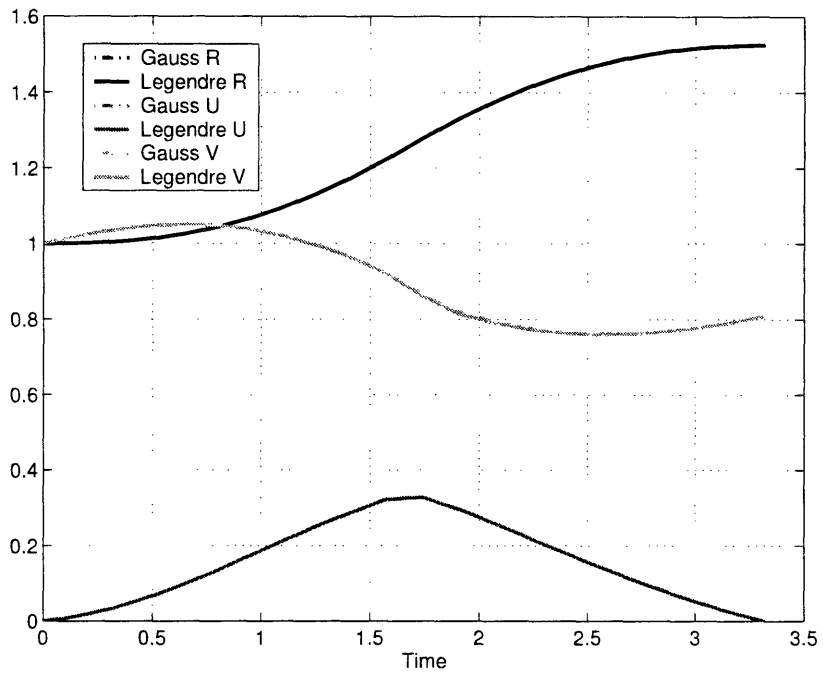


Figure 6-37: Orbit Problem States

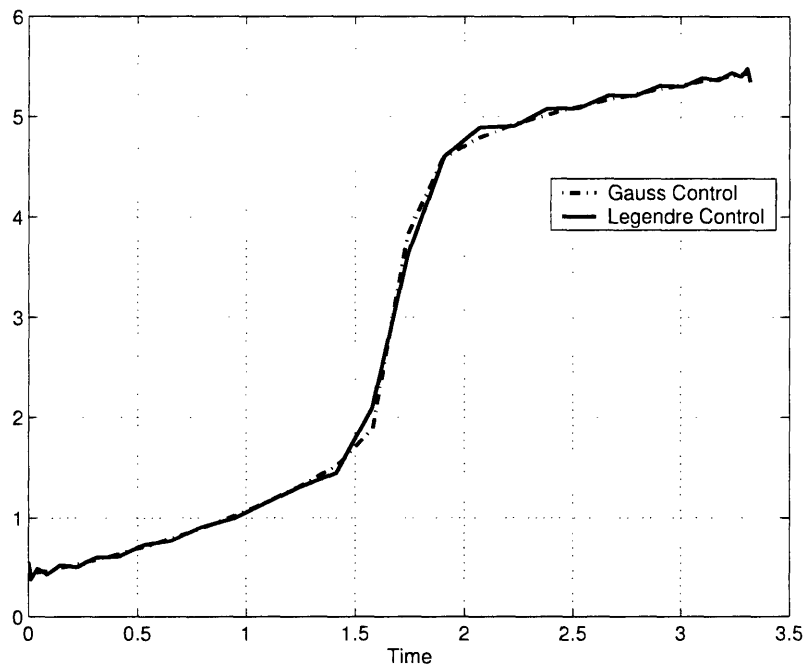


Figure 6-38: Orbit Problem Control

the right hand sides of the state dynamics are calculated at the Gauss points as

$$\begin{aligned}
F_R &= U_N, \\
F_U &= \frac{V_N^2}{2} - \frac{\mu}{R_N^2} + \frac{T \cdot \sin \phi_N}{m_o - |\dot{m}|t_N}, \\
F_V &= -\frac{U_N \cdot V_N}{R_N} + \frac{T \cdot \cos \phi_N}{m_o - |\dot{m}|t_N},
\end{aligned} \tag{6.133}$$

where t_N is the vector of time at the Gauss points. Note that all vector math in (6.133) is performed term by term. The cost to be minimized is the negative of the final radius, which is expressed in terms of a Gauss quadrature. The cost is

$$J = -r_o - \frac{t_f}{2} w^T \cdot F_R. \tag{6.134}$$

The dynamic constraints in integral form are

$$\begin{aligned}
R_N - r_o - \frac{t_f}{2} A \cdot F_R &= 0, \\
U_N - \frac{t_f}{2} A \cdot F_U &= 0, \\
V_N - \sqrt{\frac{\mu}{r_o}} - \frac{t_f}{2} A \cdot F_V &= 0.
\end{aligned} \tag{6.135}$$

The terminal constraints are also expressed using Gauss quadratures, so that

$$\begin{aligned}
\frac{t_f}{2} w^T \cdot F_U &= 0, \\
\sqrt{\frac{\mu}{r_o}} + \frac{t_f}{2} w^T \cdot F_V - \sqrt{\frac{\mu}{r_o + \frac{t_f}{2} w^T \cdot F_R}} &= 0.
\end{aligned} \tag{6.136}$$

The NLP was solved using SNOPT [30] for the states and control at the Gauss points, Fig. 6-37, 6-38.

The costate estimates, $\Lambda_R, \Lambda_U, \Lambda_V \in \mathbb{R}^N$, at the Gauss points were found using

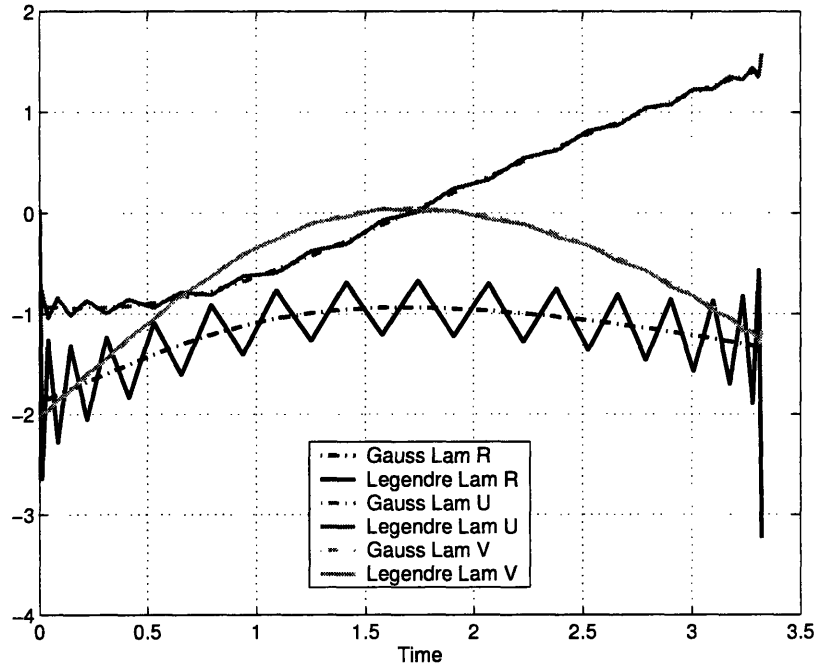


Figure 6-39: Orbit Problem Costates

the KKT multipliers of the dynamic constraints. The costate estimates are

$$\Lambda_R = W^{-1} \cdot A^T \cdot \tilde{P}_R - 1 - \frac{\nu_2 \sqrt{\mu}}{2 \left(r_o + \frac{t_f}{2} w^T \cdot F_R \right)^{3/2}}, \quad (6.137)$$

$$\Lambda_U = W^{-1} \cdot A^T \cdot \tilde{P}_U - \nu_1,$$

$$\Lambda_V = W^{-1} \cdot A^T \cdot \tilde{P}_V - \nu_2,$$

where $\tilde{P}_R, \tilde{P}_U, \tilde{P}_V \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints, and $\nu_1, \nu_2 \in \mathbb{R}$ are the Lagrange multipliers associated with the terminal constraints, Fig. 6-39.

Both the Gauss and Legendre pseudospectral solutions are in reasonable agreement with each other, and with the solution from other places ([10] [11] [22] [33] [67]). This result confirms that the integral pseudospectral solution to the orbit problem is approximately equivalent to the solution from other methods. The major differences

in the two solutions are seen in the costates, Fig. 6-39. While the costate estimates from the Legendre pseudospectral method exhibit the typical large variations, the estimates from the integral method are much smoother, especially in the first costate, λ_R . These variations in the costate also contribute to errors in the control as seen by the relatively large variation of the control from the Legendre method, Fig. 6-38. When using the integral method, the elimination of the relatively large variations in the costate is attributed to the elimination of the defects in the costate approximation equations. The fact that the KKT conditions that define the solution to the NLP are exactly equivalent to the discretized form of the first-order necessary conditions is why the integral pseudospectral method has an improved costate estimate.

The costate estimate along with the approximate solution for the states and control can be used to verify the optimality of the solution. The approximate state, costate and control, $\mathbf{X}^*(t), \boldsymbol{\Lambda}^*(t) \in \mathbb{R}^3$, and $U^*(t) \in \mathbb{R}$, can be formulated for the low thrust orbit problem using a set of interpolating polynomials for both the integral pseudospectral and Legendre pseudospectral solution. These functions can then be used to test how well they satisfy the first-order optimality conditions derived in [10] for the low thrust orbit problem. These conditions are

$$\begin{aligned}\dot{\mathbf{X}}^*(t) &= \mathbf{f}(\mathbf{X}^*(t), \mathbf{U}^*(t), t), \\ \dot{\boldsymbol{\Lambda}}^*(t) &= -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}(\mathbf{X}^*(t), \boldsymbol{\Lambda}^*(t), \mathbf{U}^*(t), t), \\ \mathbf{U}^*(t) &= \arg \min_{\mathbf{u}(t) \in \mathbf{U}} [\mathcal{H}(\mathbf{X}^*(t), \boldsymbol{\Lambda}^*(t), \mathbf{u}(t), t)].\end{aligned}\tag{6.138}$$

The error in satisfying these conditions was computed using the infinity norm, and the comparison of the integral method and the Legendre method is shown in Table 6.1. The table shows that the solution found using the Legendre method satisfies the state dynamics and Pontryagin's maximum principle very well but does not satisfy the costate dynamics. The integral method however satisfies all three conditions very well. This result indicates that the solution from the integral method can be said to be optimal with higher confidence. Because the solution from the Legendre method does not satisfy the optimality conditions, the verification of the control can not be

Table 6.1: Low Thrust Orbit Optimality

	Integral Gauss Pseudospectral	Legendre Pseudospectral
$\ \dot{\mathbf{X}}^*(t) - \mathbf{f}(\mathbf{X}^*(t), \mathbf{U}^*(t), t)\ _\infty$	0.0459	0.0197
$\ \dot{\boldsymbol{\Lambda}}^*(t) + \mathcal{H}_{\mathbf{x}}(\mathbf{X}^*(t), \boldsymbol{\Lambda}^*(t), \mathbf{U}^*(t), t)\ _\infty$	0.0019	572.65
$\ \mathbf{U}^*(t) - \arg \min_{\mathbf{u}(t) \in \mathbf{U}} [\mathcal{H}(\mathbf{X}^*(t), \boldsymbol{\Lambda}^*(t), \mathbf{u}(t), t)]\ _\infty$	0.3379	0.4047

done.

6.8.3 Low-Thrust Orbit Problem Convergence

An estimate of the convergence of both pseudospectral methods can be made by comparing the approximate solutions to a solution of the two point boundary value problem derived in [10]. The solution of the boundary value problem was found using the MATLAB function `bvp4c`, with a tolerance of 10^{-9} . The solution of the low thrust orbit problem using the integral pseudospectral method and 128 nodes was used for the initial guess.

The convergence in the infinity norm of the states for both the Gauss pseudospectral and Legendre pseudospectral methods are shown in Fig. 6-40. The figure shows that the convergence for the states is similar for both methods. However, the convergence of the costates is not the same, shown in Fig. 6-41. The error in the costates for the Legendre method is significantly higher than the integral method, which was also indicated in Fig. 6-39. Fig. 6-41 indicates that a significantly greater number of nodes for the Legendre method is required to achieve the same level of accuracy as the integral method for the costates. As expected, the larger errors in the costate affect the error in the control as shown in Fig. 6-42. The figure indicates that as many as 40 additional nodes would be required of the Legendre pseudospectral method to achieve the same accuracy in the control as the Gauss pseudospectral method.

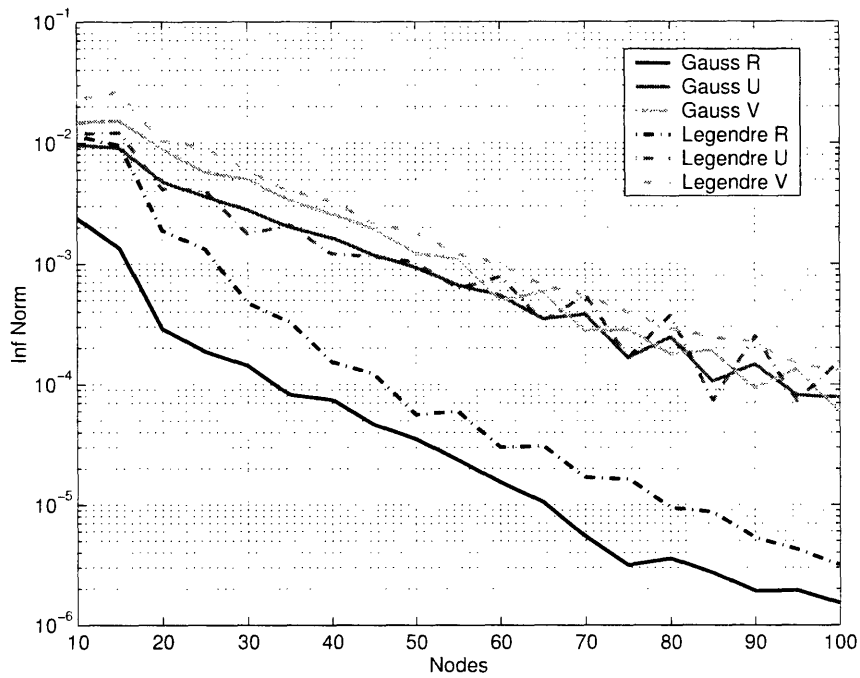


Figure 6-40: Orbit Problem State Convergence

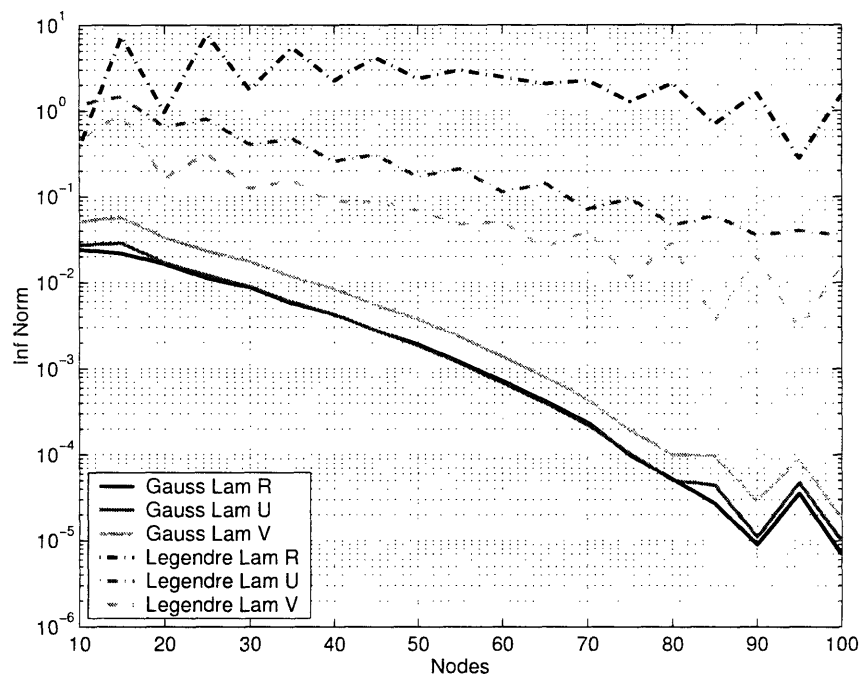


Figure 6-41: Orbit Problem Costate Convergence

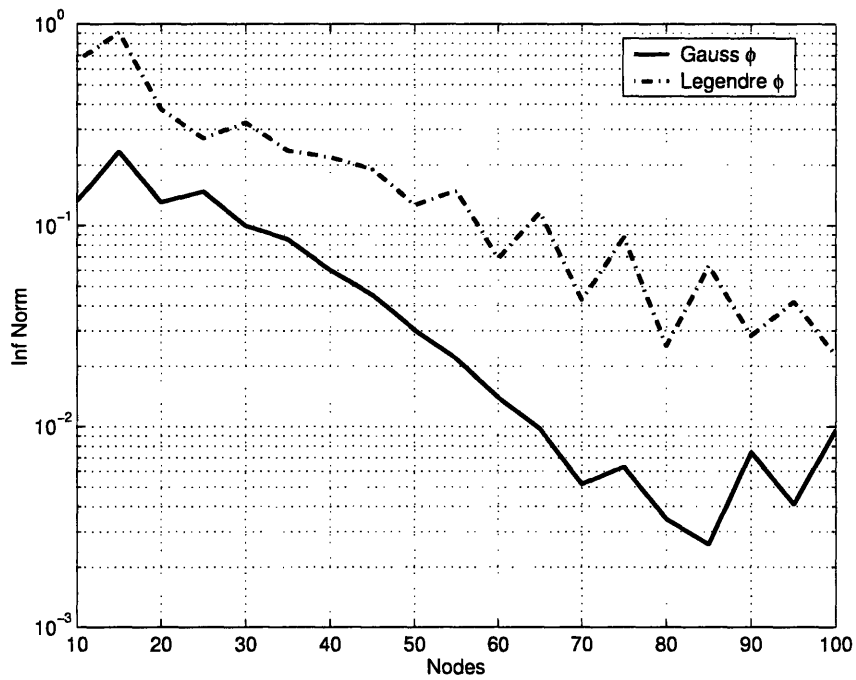


Figure 6-42: Orbit Problem Control Convergence

6.8.4 Differential and Integral Relation

In this section, the differential and integral formulations of the Gauss pseudospectral method are compared on the nonlinear orbit transfer problem.

The NLP for the differential form is found using the same NLP variables as the integral form (see Section 6.8.2) and the right hand sides of the differential equations (6.133).

The cost function to be minimized is the opposite sign of the final radius, so that

$$J = -r_o - \frac{t_f}{2} w^T \cdot F_R. \quad (6.139)$$

The dynamic constraints in differential form are

$$\begin{aligned}
\frac{2}{t_f} (\bar{D} \cdot r_o + D \cdot R_N) - F_R &= 0, \\
\frac{2}{t_f} D \cdot U_N - F_U &= 0, \\
\frac{2}{t_f} \left(\bar{D} \cdot \sqrt{\frac{\mu}{r_o}} + D \cdot V_N \right) - F_V &= 0,
\end{aligned} \tag{6.140}$$

Where D and \bar{D} make up the differential approximation matrix. The terminal constraints are expressed using Gauss quadratures, so that

$$\begin{aligned}
\frac{t_f}{2} w^T \cdot F_U &= 0, \\
\sqrt{\frac{\mu}{r_o}} + \frac{t_f}{2} w^T \cdot F_V - \sqrt{\frac{\mu}{r_o + \frac{t_f}{2} w^T \cdot F_R}} &= 0.
\end{aligned} \tag{6.141}$$

The NLP was solved using SNOPT [30] for the states and control at the Gauss points. The costates, $\Lambda_R, \Lambda_U, \Lambda_V \in \mathbb{R}^N$, at the Gauss points, were estimated using the costate mapping (Theorem 5.4.1), so that

$$\begin{aligned}
\Lambda_R &= \frac{2}{t_f} \cdot W^{-1} \cdot \tilde{\Lambda}_R - 1 - \frac{\nu_2 \sqrt{\mu}}{2 \left(r_o + \frac{t_f}{2} w^T \cdot F_R \right)^{3/2}}, \\
\Lambda_U &= \frac{2}{t_f} \cdot W^{-1} \cdot \tilde{\Lambda}_U - \nu_1, \\
\Lambda_V &= \frac{2}{t_f} \cdot W^{-1} \cdot \tilde{\Lambda}_V - \nu_2,
\end{aligned} \tag{6.142}$$

where W is the diagonal matrix of Gauss weight, $\tilde{\Lambda}_R, \tilde{\Lambda}_U, \tilde{\Lambda}_V \in \mathbb{R}^N$ are the KKT multipliers associated with the dynamic constraints, and $\nu_1, \nu_2 \in \mathbb{R}$ are the Lagrange multipliers associated with the terminal constraints. The resulting solution for the states, costates, and control are the same (within numerical tolerances of SNOPT) to the solution using the integral formulation (see Fig. 6-37, 6-38, 6-39).

The constraint Jacobian (see Section 2.3) is defined as the partial derivative of the nonlinear constraints with respect to the optimization variables. It is common

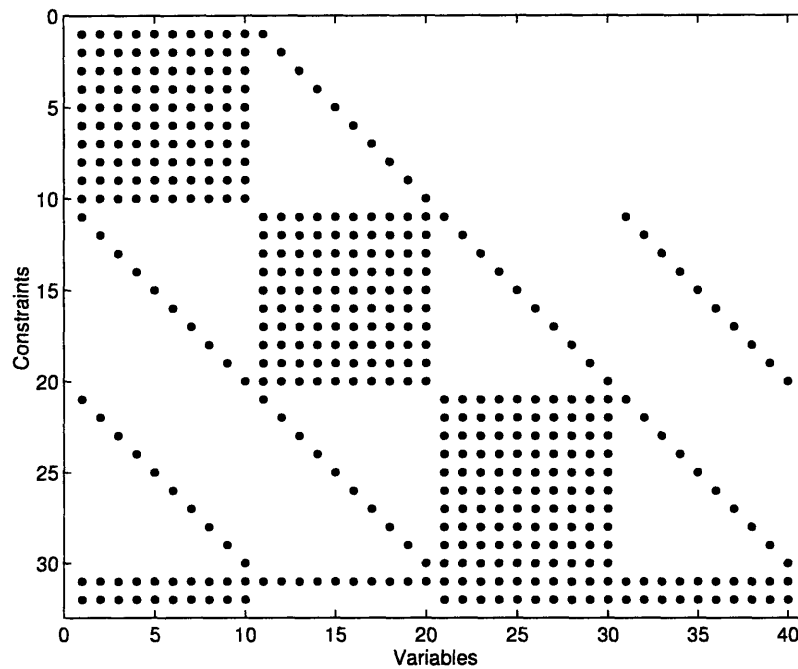


Figure 6-43: Differential Form Sparsity Pattern

for NLP solvers to estimate the Jacobian numerically. The sparsity pattern provides information as to which entries are non-zero.

The sparsity pattern of the Jacobian from the differential formulation is shown using 10 nodes in Fig. 6-43. The sparsity pattern indicates which nonlinear constraints (6.140 - 6.141), depend on which variables, R_N, U_N, V_N, ϕ_N . In this case there are 32 nonlinear constraints, 10 for each differential equation and 2 for the boundary conditions, and 40 variables, 10 for each state and 10 for the control. The figure indicates that most of the Jacobian is zero, and has a density of approximately 25%.

The sparsity pattern of the Jacobian from the integral formulation is much more dense (Fig. 6-44). The density of this Jacobian is approximately 67%, indicating the many more derivatives must be estimated. The more sparse NLP from the differential formulation leads to more efficient computation of the solution.

The time to find the solution of the NLP using the differential formulation, is compared to the integral formulation in Fig. 6-45. Note that in both cases the same

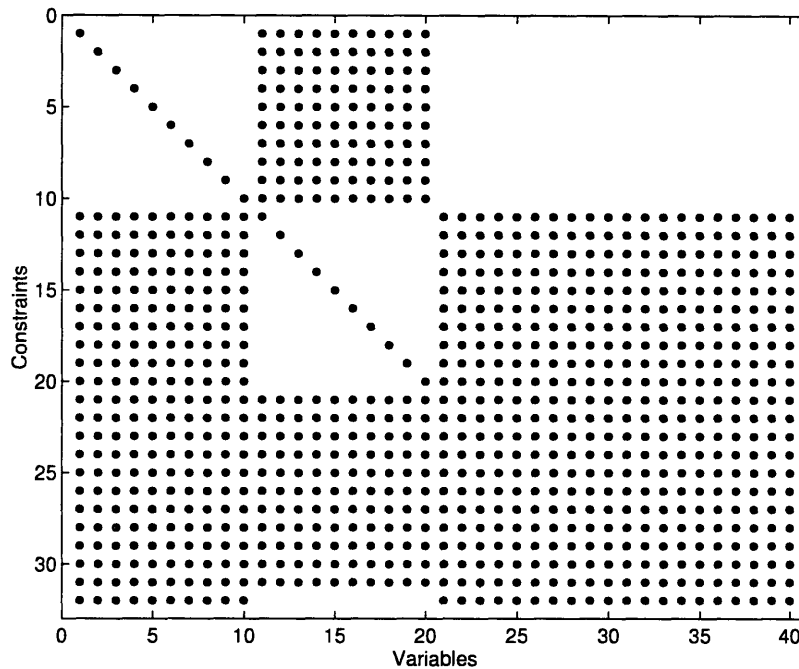


Figure 6-44: Integral Form Sparsity Pattern

initial guess was used. The figure indicates that less computation time is required to solve the NLP in the differential form. This result is a consequence of the more sparse NLP generated using the differential form.

6.9 Summary

These example problems demonstrate empirically that the Gauss pseudospectral solution is very well suited for solving a large class of problems. The method works well on linear and nonlinear problems whose solutions are infinitely differentiable. The method does not work as well on problems that contain discontinuities in the solution, problems with a singular arcs, or problems that have singularities near the solution interval. For problems with discontinuities in the solution or in the derivatives of the solution, using the multiple phase approach can dramatically improve the approximate solution for some, but not all, problems.

It has been shown in these examples that the Gauss pseudospectral method has

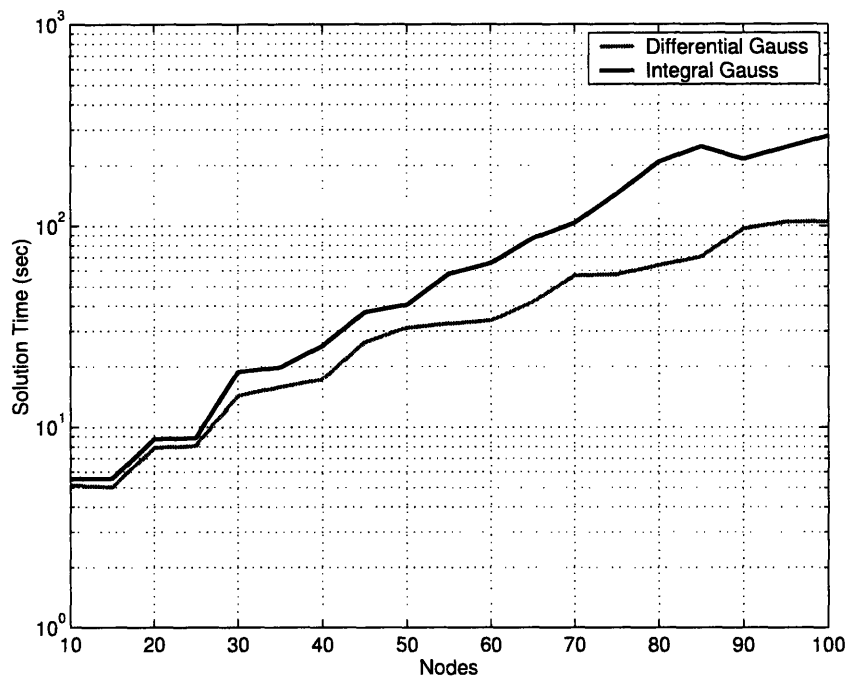


Figure 6-45: Differential and Integral CPU Time

significantly better convergence properties than finite difference methods and yields a better costate estimate than the Legendre pseudospectral method. These examples have also shown that the initial costate estimate from the Gauss method converges to the true solution faster than the states and costates at the Gauss points. This accurate initial costate is useful for real time control of nonlinear systems. The development of a real time control algorithm using the initial costate is derived in Chapter 7.

Chapter 7

Real Time Optimal Control

The Gauss pseudospectral method has the potential to be used for real time optimal control of complex nonlinear systems. This potential can be attributed to two properties of the transcription method. First, it has very fast convergence, as demonstrated in Chapter 6, which allows for an accurate solution to be generated with a relatively small number of nodes. The fewer nodes that are required results in the NLP being smaller, which can then be solved quickly. The second property is the availability of accurate costate information, particularly the initial costate. It has been empirically shown (see Chapter 6) that the initial costate estimate is more accurate than the costate estimate at the interior collocation points. The accurate initial costate information, along with the initial state information, can be used for real time control.

The initial costate can be used, with the initial state, to estimate the state and costate for all time. This estimation is done by integrating forward the state and costate dynamic equations,

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \\ \frac{d\boldsymbol{\lambda}}{dt} &= \frac{-\partial\mathcal{H}^T}{\partial\mathbf{x}}(\mathbf{x}(t), \boldsymbol{\lambda}(t), \mathbf{u}(t), t).\end{aligned}\tag{7.1}$$

This integration can be done quickly and accurately using single step numerical techniques such as a classic Runge-Kutta integration scheme ([41], [44], [63]). The optimal control can be found at any point by applying Pontryagin's maximum principle ([10],

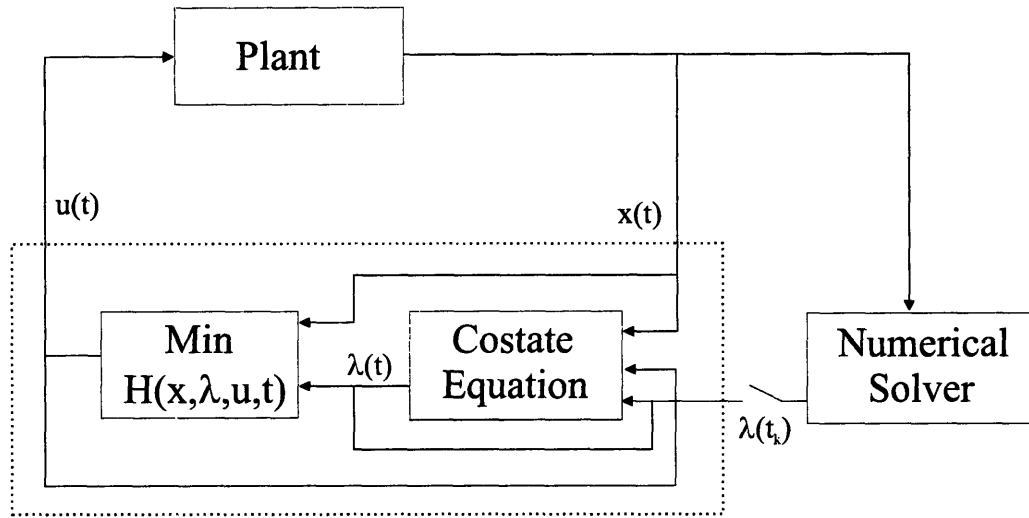


Figure 7-1: Real Time Control

[45], [53]), with the known state and costate, so that

$$\mathbf{u}(t) = \arg \min_{\mathbf{u} \in \mathbf{U}} [\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t)] . \quad (7.2)$$

Once the optimal control is known, it can be applied to the system. By continuing to integrate the state and costate dynamics and applying Pontryagin's maximum principle, the optimal control can be found for all future time. As time increases, the system could drift from the optimal path because of disturbances and modeling errors. The optimization problem can be resolved using the current state to find the new current costate. This re-initialization would compensate for any errors that have accumulated and the system would follow the new optimal path.

Figure 7-1 shows the block diagram of the proposed optimal controller. The diagram shows that the state dynamics do not need to be propagated numerically because the current state can be measured (or estimated) directly from the plant. The controller is then defined as the system from the state to the control, which requires periodic re-initialization of the initial costate from the Gauss pseudospectral method. The re-initialization rate as well as the accuracy of the initial costate required to adequately control a system in real time is highly dependent on the system to be

controlled.

The implementation and effectiveness of this real time control approach is shown with two examples below.

7.1 Examples

7.1.1 LQR Problem

The linear quadratic problem considered here is a one-dimensional, fixed-time problem with fixed initial and final states. The cost to be minimized is

$$J = \frac{1}{2} \int_0^5 (x^2(t) + u^2(t)) dt, \quad (7.3)$$

subject to the dynamic constraint

$$\frac{dx}{dt}(t) = x(t) + u(t), \quad (7.4)$$

and boundary constraints

$$x(0) = 1, \quad x(5) = 2. \quad (7.5)$$

The true solution for the state, control, and costate was calculated to be

$$\begin{aligned} x(t) &= 0.9983 \cdot e^{-\sqrt{2}t} + 1.6979 \cdot 10^{-3} \cdot e^{\sqrt{2}t}, \\ u(t) &= -2.4101 \cdot e^{-\sqrt{2}t} + 7.0331 \cdot 10^{-4} \cdot e^{\sqrt{2}t}, \\ \lambda(t) &= 2.4101 \cdot e^{-\sqrt{2}t} - 7.0331 \cdot 10^{-4} \cdot e^{\sqrt{2}t}, \end{aligned} \quad (7.6)$$

and is shown in Fig. 7-2. The pseudospectral solution to the LQR problem was found using a similar formulation to the LQR example in Section 6.1.4. The convergence of the approximate solution, infinity norm error versus number of nodes used, is shown in Fig. 7-3. Note that the error in the control is equal to the error in the costate. As expected, the initial costate converges to the exact solution at a faster rate than the state, control, and costate at the interior nodes. The accurate initial costate will be

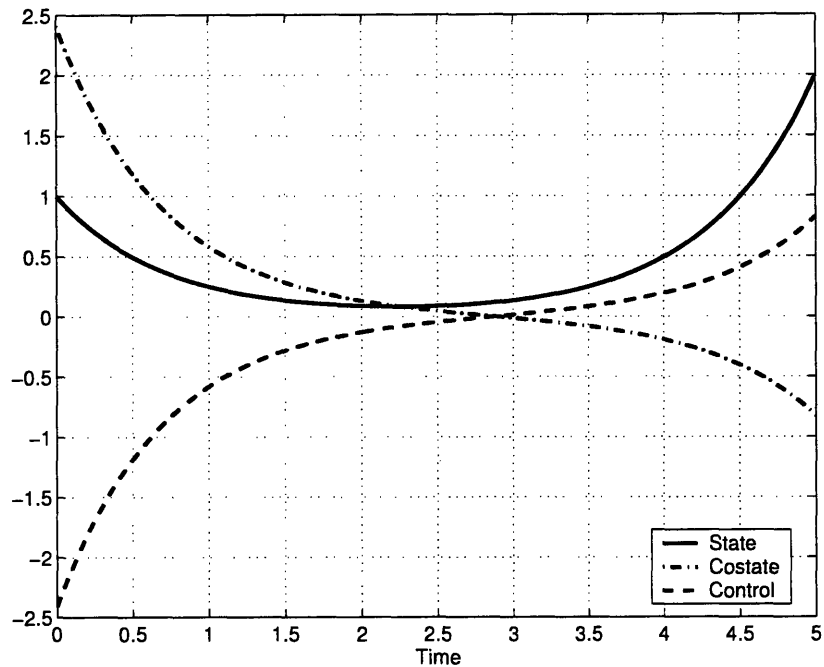


Figure 7-2: LQR Exact Solution

exploited to determine an accurate approximation to the control.

The state and costate dynamics for the problem can be found using the first-order necessary conditions and the control determined using Pontryagin's maximum principle (see Section 2.4). The control is found to be

$$u(t) = -\lambda(t), \quad (7.7)$$

and the state/costate dynamics are then

$$\begin{aligned} \dot{x}(t) &= x(t) - \lambda(t), \\ \dot{\lambda}(t) &= -x(t) - \lambda(t). \end{aligned} \quad (7.8)$$

Given the initial conditions for state and costate, $x(t_0)$, $\lambda(t_0)$, the exact solution for the system can be found using the eigenvalues and eigenvectors from the system matrix. By determining the exact state and costate resulting from any initial

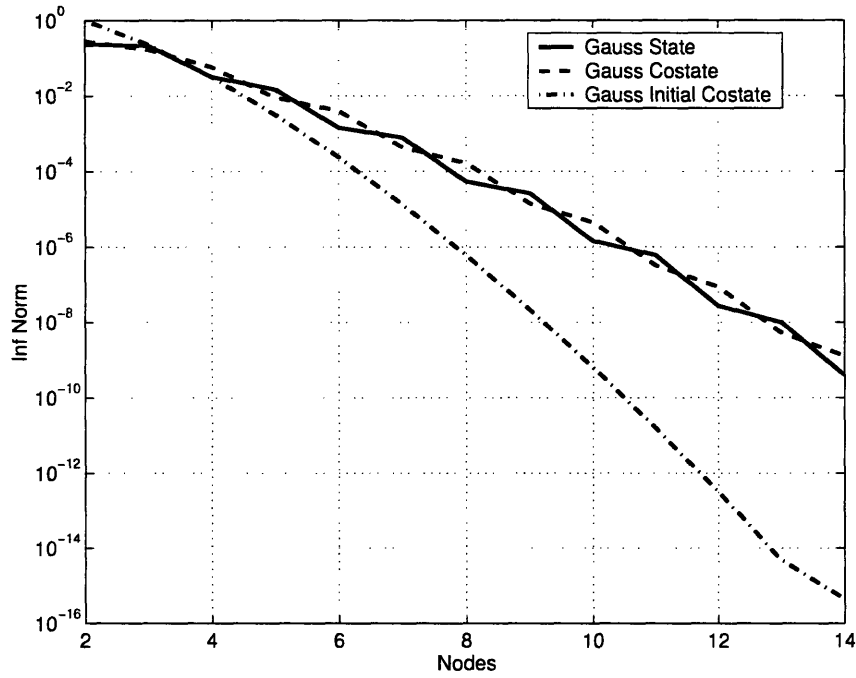


Figure 7-3: LQR Pseudospectral Convergence

conditions avoids any numerical integration techniques to approximate the resulting behavior of the system. The solution to the dynamics can be found to be

$$\begin{aligned} x(t) &= c_1 \cdot v_{11} \cdot e^{\mu_1 \cdot (t-t_0)} + c_2 \cdot v_{12} \cdot e^{\mu_2 \cdot (t-t_0)}, \\ \lambda(t) &= c_1 \cdot v_{21} \cdot e^{\mu_1 \cdot (t-t_0)} + c_2 \cdot v_{22} \cdot e^{\mu_2 \cdot (t-t_0)}, \end{aligned} \quad (7.9)$$

where $\mu_1 = -\sqrt{2}$, $\mu_2 = \sqrt{2}$ are the eigenvalues of the system and

$$\begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} -0.3827 \\ -0.9239 \end{bmatrix}, \quad \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix} = \begin{bmatrix} -0.3827 \\ 0.9239 \end{bmatrix},$$

are the eigenvectors. The coefficients, c_1, c_2 , are determined to be

$$\begin{aligned} c_1 &= \frac{-v_{12} \cdot \lambda(t_0) + v_{22} \cdot x(t_0)}{v_{11} \cdot v_{22} - v_{12} \cdot v_{21}}, \\ c_2 &= \frac{v_{11} \cdot \lambda(t_0) - v_{21} \cdot x(t_0)}{v_{11} \cdot v_{22} - v_{12} \cdot v_{21}}. \end{aligned} \quad (7.10)$$

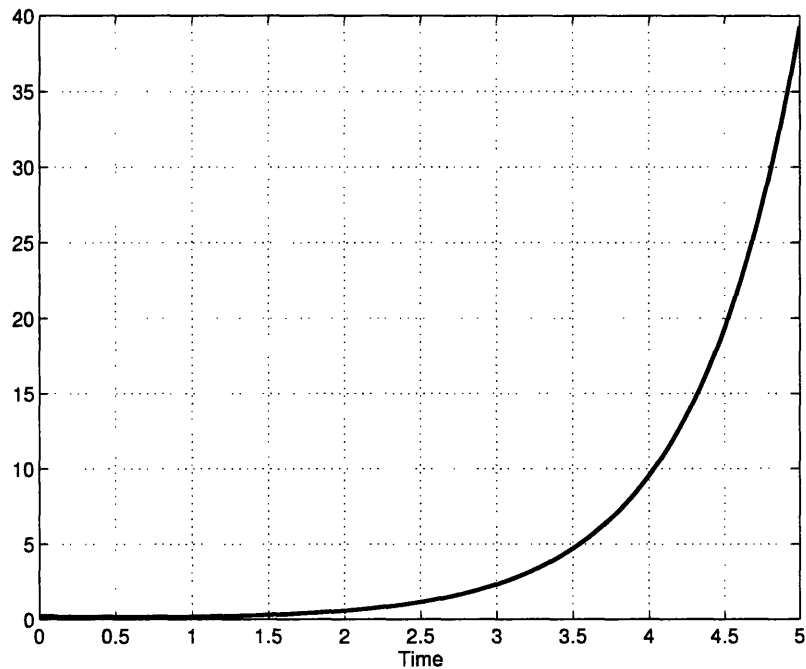


Figure 7-4: LQR Control Error, Open Outer Loop

The initial state, $x(0) = 1$, along with the estimate for the initial costate from the Gauss pseudospectral method was used to propagate the system forward in time using the dynamic relations (7.9). The system was propagated to the final time $t_f = 5$. The error in the resulting control, the propagated control subtracted from the exact control (7.6), is plotted in Fig. 7-4. The initial costate was found using 3 nodes. As expected, the error in the control starts at approximately 10^{-1} and grows exponentially with time. This large control error is because the Hamiltonian system is unstable. The error in the control leads to a significant error in the final state, $x(t_f) = -92.79$, which does not meet the terminal constraint. The solution can be improved by resolving the optimal control problem and updating the costate, which updates the control, several times as the system is propagated forward in time.

Resolving the system four times, once every time unit, the propagated state and costate is much closer to the exact solution, as shown in Fig. 7-5 and Fig. 7-6. Three nodes were used in the pseudospectral solution each time the problem was re-solved.

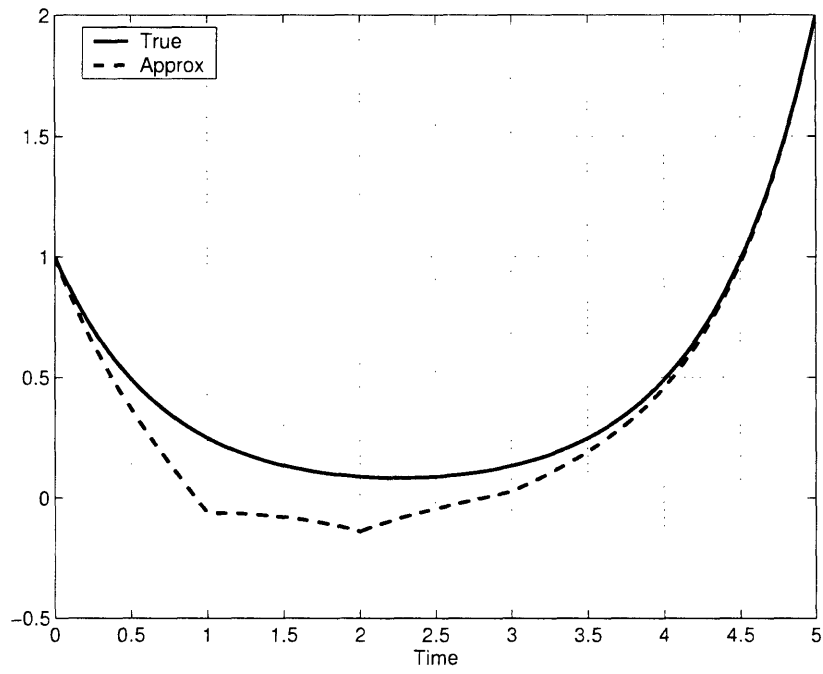


Figure 7-5: LQR Propagated State

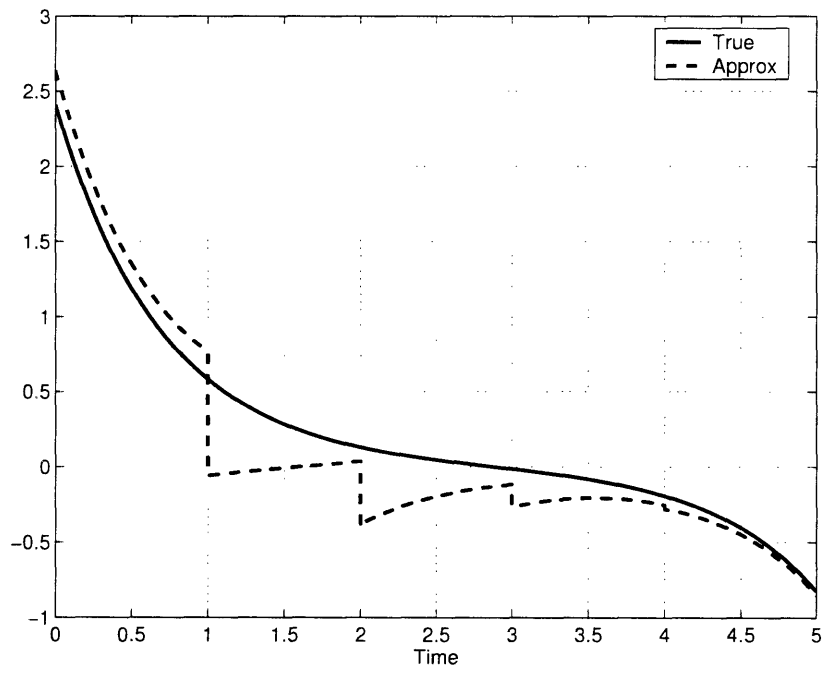


Figure 7-6: LQR Propagated Costate

The propagated state is within $2.5 \cdot 10^{-4}$ of the exact state at the final time, with a total cost of 2.2151 versus 2.0307 for the exact solution. As the number of system updates, or number of nodes used is increased, the propagated solution gets closer to the exact solution and the cost approaches the cost for the exact solution. This example demonstrates that the control can be found from the approximate initial costate, that causes the system to meet the required boundary conditions by continually resolving the optimal control problem to update the costate.

7.1.2 Low-Thrust Orbit Problem

The low-thrust orbit transfer problem [50] solved by the Gauss pseudospectral method in Section 6.8 was used to demonstrate the real time control approach using the initial costate. The state dynamics for the problem are

$$\begin{aligned} \frac{dr}{dt} &= u, \\ \frac{du}{dt} &= \frac{v^2}{2} - \frac{\mu}{r^2} + \frac{T \cdot \sin \phi}{m_o - |\dot{m}|t}, \\ \frac{dv}{dt} &= -\frac{u \cdot v}{r} + \frac{T \cdot \cos \phi}{m_o - |\dot{m}|t}. \end{aligned} \quad (7.11)$$

Using the calculus of variations and Pontryagin's maximum principle the costate dynamics can be shown to be [10]

$$\begin{aligned} \frac{d\lambda_r}{dt} &= -\lambda_u \cdot \left(-\frac{v^2}{r^2} + \frac{2\mu}{r^3} \right) - \lambda_v \cdot \left(\frac{u \cdot v}{r^2} \right), \\ \frac{d\lambda_u}{dt} &= -\lambda_r + \lambda_v \cdot \frac{v}{r}, \\ \frac{d\lambda_v}{dt} &= -\lambda_u \cdot \frac{2 \cdot v}{r} + \lambda_v \cdot \frac{u}{r}, \end{aligned} \quad (7.12)$$

with the control defined as

$$\tan \phi = \frac{\lambda_u}{\lambda_v}. \quad (7.13)$$

The Gauss pseudospectral method is used to estimate the initial costate. Using the initial costate along with the initial states, the state and costate dynamics are

Table 7.1: Orbit Problem Real Time Results

Nodes per Solution	Final Radius	Error in Final Radius	Final Eccentricity
5	1.5228	$2.5 \cdot 10^{-3}$	$7.73 \cdot 10^{-6}$
10	1.5252	$1.2 \cdot 10^{-4}$	$9.04 \cdot 10^{-7}$

integrated forward in time in MATLAB using the ode45 integrator. The dynamics are integrated for a fixed time then the optimal control problem is resolved to reestimate the costate at this time. In this case the dynamics are integrated for 0.0332 time units, which divides the problem into 100 intervals.

The NLP from the Gauss pseudospectral method was solved using SNOPT [30] and used five nodes for each solution. The solver required on average approximately 0.6 seconds to determine each solution. The plot of the approximate real time solution is shown in Fig. 7-7 for the states, Fig. 7-8 for the costates, and Fig. 7-9 for the control. The “true” solution that these are compared to is the solution of the two point boundary value problem using the MATLAB boundary value solver bvp4c. The figures indicate that although the costates and control seem have significant errors, the states satisfy the boundary conditions, the final orbit is very nearly circular, and the final orbit radius is within 0.16% of the maximum possible radius (Table 7.1). This performance is all achieved using only 5 nodes in the pseudospectral solution. The solution accuracy can be increased by using 10 nodes per solution as shown in Fig. 7-10 for the states, Fig. 7-11 for the costates, and Fig. 7-12 for the control. In this case the final states satisfy the boundary conditions and the final orbit radius is within 0.0079% of the maximum (Table 7.1).

These examples show how the initial costate can be applied with Pontryagin’s maximum principle for real time optimal control of nonlinear systems. By continually resolving the optimal control problem to update the costate, errors that accumulate from propagation, modeling errors, and disturbances, can be canceled out. The Gauss pseudospectral method is ideally suited to be used to estimate the initial costate. The method can quickly solve for a highly accurate initial costate because it needs only a few nodes to return a good solution.

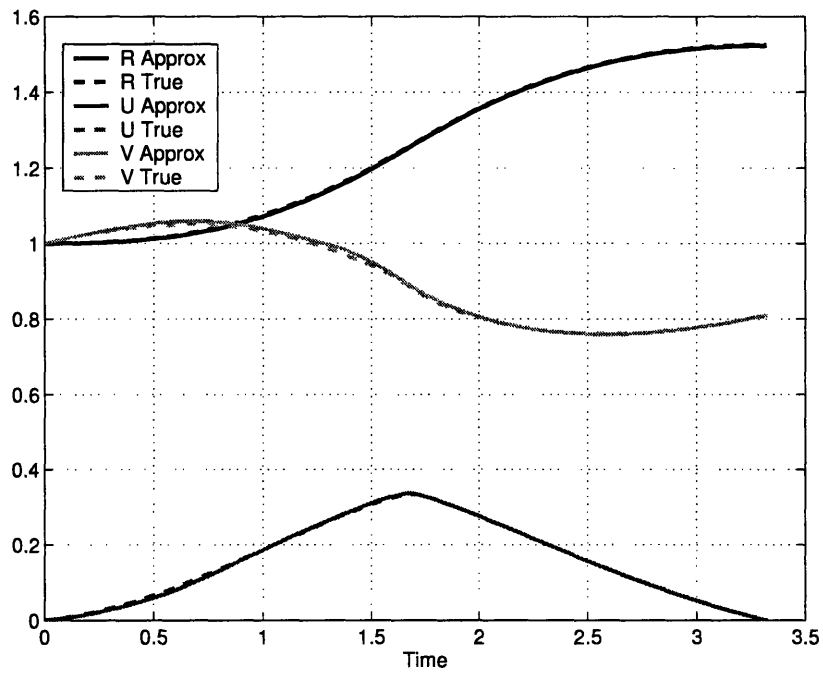


Figure 7-7: Orbit Problem Real Time States, 5 Nodes

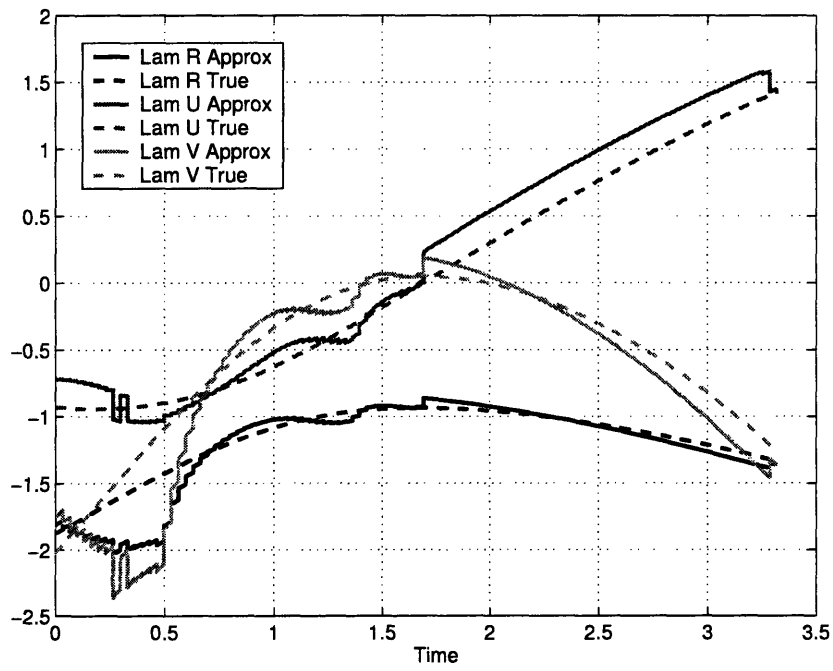


Figure 7-8: Orbit Problem Real Time Costates, 5 Nodes

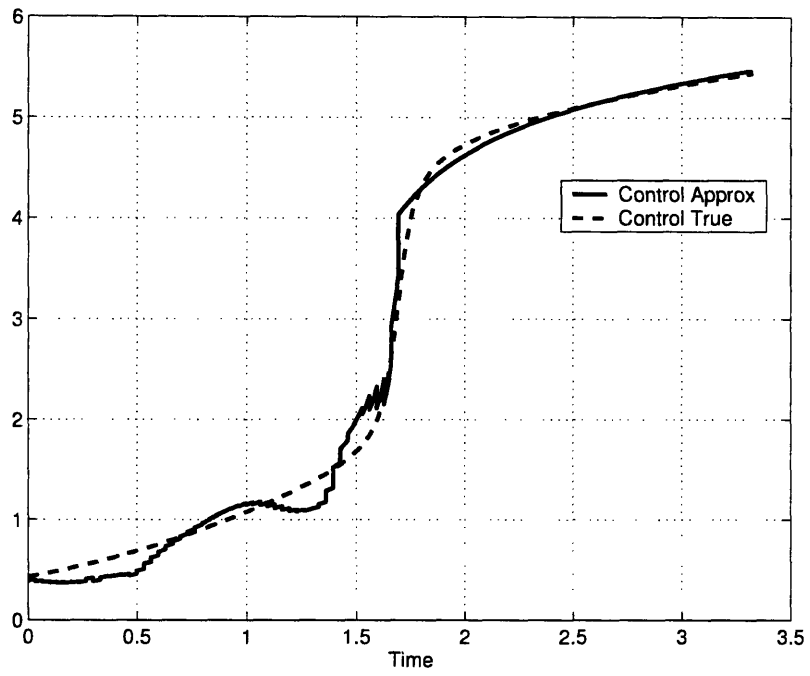


Figure 7-9: Orbit Problem Real Time Control, 5 Nodes

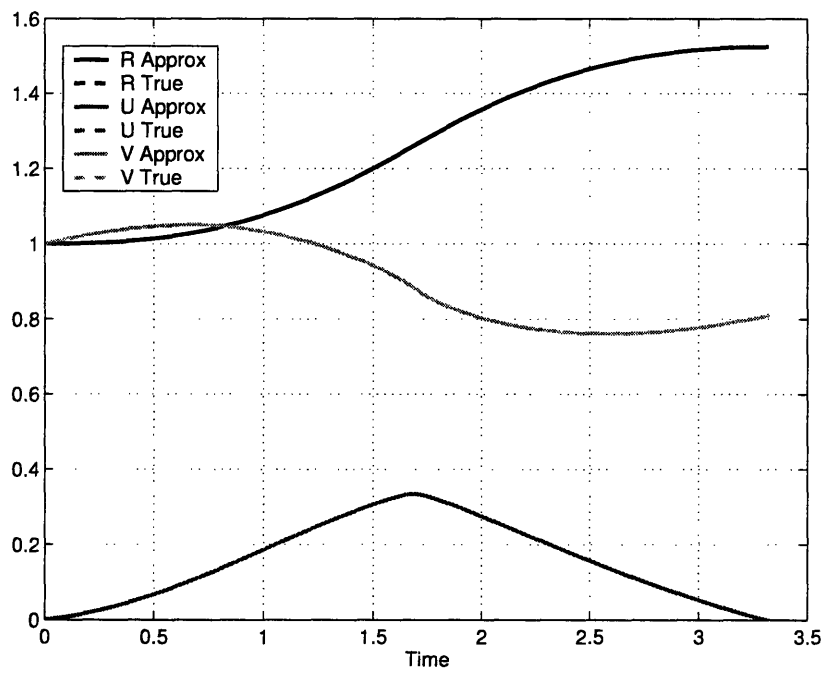


Figure 7-10: Orbit Problem Real Time States, 10 Nodes

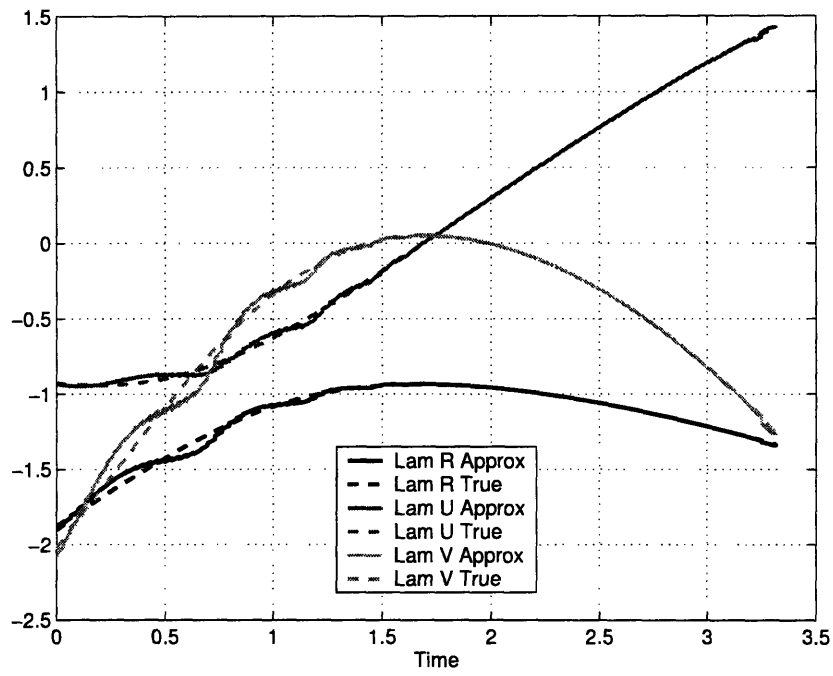


Figure 7-11: Orbit Problem Real Time Costates, 10 Nodes

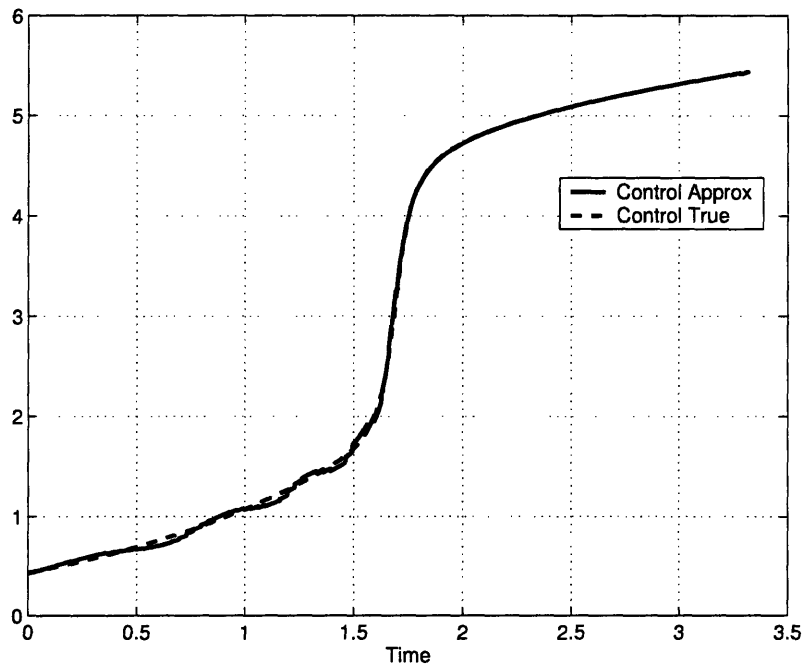


Figure 7-12: Orbit Problem Real Time Control, 10 Nodes

Chapter 8

Launch Vehicle Example

In this chapter, the trajectory optimization of a Delta III launch vehicle is presented. The dynamic model is formulated and the desired trajectory parameters are described. The resulting optimal control problem is then solved using the integral Gauss pseudospectral method.

Next, the real time control approach from Chapter 7 is used to simulate the real time control of the launch vehicle. The simulation was run with and without external disturbances to the vehicle. In both cases, it is shown that the real time control algorithm is able to correct for any errors in the system, and keep the vehicle on the optimal trajectory.

8.1 Problem Description

The problem considered here is the guidance for a Delta III launch vehicle. The objective is to get the spacecraft from the launch site into a predetermined target orbit, while maximizing the fuel remaining in the upper stage. This approach is used to maximize the amount of fuel available to correct the trajectory in flight. In this way, the control for the vehicle can account for any disturbances or uncertainties, during of the flight. The less fuel that is required to get the vehicle into the desired orbit, the larger the disturbance that can be tolerated.

Table 8.1: Delta III Properties

	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

8.1.1 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 broken into four distinct phases. The first phase starts on the ground where the main engine burns along with six of the solid boosters. When the boosters are depleted their remaining mass is dropped and the final three boosters are lit. This begins phase two where the main engine continues to burn with the three remaining solid boosters. When the final three boosters are depleted their mass is also dropped leaving only the main engine burning. Phase three continues until the main engine fuel has been exhausted (MECO). In the fourth and final phase the main engine is dropped and the second stage is ignited. The second stage burns until the target orbit has been reached (SECO), and the payload is separated from the launch vehicle. A summary of the Delta III vehicle characteristics [1] is found in Table 8.1.

8.1.2 Dynamic Model

The dynamics of the vehicle are expressed in an Earth centered inertial (ECI) frame as

$$\begin{aligned}
 \frac{d\mathbf{r}}{dt} &= \mathbf{v}, \\
 \frac{d\mathbf{v}}{dt} &= -\frac{\mu}{|r|^3} \cdot \mathbf{r} + \frac{T}{m} \cdot \mathbf{u} + \frac{\mathbf{D}}{m}, \\
 \frac{dm}{dt} &= -\frac{T}{g_o \cdot Isp},
 \end{aligned} \tag{8.1}$$

where $\mathbf{r} \in \mathbb{R}^3$ is the position, $\mathbf{v} \in \mathbb{R}^3$ is the velocity, μ is the gravitational parameter, T is the thrust, m is the mass, $\mathbf{u} \in \mathbb{R}^3$ is the unit vector in the direction of thrust, and $\mathbf{D} \in \mathbb{R}^3$ is the aerodynamic drag [61]. The drag is defined as

$$\mathbf{D} = -\frac{1}{2}C_d \cdot A_{ref} \cdot \rho \cdot |v_{rel}| \cdot \mathbf{v}_{rel} , \quad (8.2)$$

where C_d is the coefficient of drag, A_{ref} is the reference area, ρ is the atmospheric density, and \mathbf{v}_{rel} is the relative velocity with respect to the atmosphere. The relative velocity is computed to account for the fact that the Earth's atmosphere is rotating with the Earth, so that

$$\mathbf{v}_{rel} = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r} , \quad (8.3)$$

where $\boldsymbol{\omega}$ is the rotation rate of the Earth. The atmospheric density is modeled as exponential, so that

$$\rho = \rho_o \cdot \exp[-h/h_o] , \quad (8.4)$$

where ρ_o is sea level density, h is the altitude above the Earth's surface, and h_o is the scale height of the atmosphere.

The dynamic model used has many simplifying assumptions. First, the thrust from each engine is assumed to be the vacuum thrust. Therefore, the thrust magnitude does not depend on the atmospheric pressure. Second, the reference area and coefficient of drag are constant for the entire trajectory with no dependence on Mach number or angle of attack. Third, the drag is assumed to always be in the opposite direction of the relative velocity. There is no component of lift, and the drag has no dependence on the vehicle orientation. The final assumption is the the Earth is a sphere. This assumption is used when determining the position of the launch site and the altitude above the Earth. The spherical Earth also satisfies the point mass gravity model.

These assumptions cause the dynamics to be the same during each phase of the flight, and only the thrust magnitude and mass changes. Each phase is linked to the next phase by a set of boundary constraints that force the position and velocity to be continuous, and takes the mass drops into account.

8.1.3 Optimal Control Formulation

The objective of the launch vehicle problem is to determine the guidance, or thrust direction, of the vehicle to place the payload into the determined target orbit, while maximizing the remaining fuel in the second stage at orbit insertion. The initial time for the problem is t_0 , and t_1, t_2, t_3, t_4 are the end times of each of the phases. The times t_0 through t_3 are fixed and only t_4 is free.

The initial conditions are determined by the coordinates of the launch site and the initial mass of the vehicle, so that

$$\begin{aligned} \mathbf{r}(t_0) &= \mathbf{r}_o, \\ \mathbf{v}(t_0) &= \mathbf{v}_o, \\ m(t_0) &= m_o. \end{aligned} \tag{8.5}$$

The dynamic constraints for the four phases are expressed in integral form as

$$\begin{aligned} \mathbf{r}_i(t) &= \mathbf{r}(t_{i-1}) + \int_{t_{i-1}}^t \mathbf{v}_i d\tau, \\ \mathbf{v}_i(t) &= \mathbf{v}(t_{i-1}) + \int_{t_{i-1}}^t \left(-\frac{\mu}{|r_i|^3} \cdot \mathbf{r}_i + \frac{T_i}{m_i} \cdot \mathbf{u}_i + \frac{\mathbf{D}_i}{m_i} \right) d\tau, \\ m_i(t) &= m(t_{i-1}) + \int_{t_{i-1}}^t \left(-\frac{T_i}{g_o \cdot Isp_i} \right) d\tau, \\ i &= 1, \dots, 4. \end{aligned} \tag{8.6}$$

The phase boundary constraints are defined in terms of the states at the final time of each phase. This relation defines the initial condition for the next phase as

$$\begin{aligned} \mathbf{r}(t_i) &= \mathbf{r}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \mathbf{v}_i d\tau, \\ \mathbf{v}(t_i) &= \mathbf{v}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \left(-\frac{\mu}{|r_i|^3} \cdot \mathbf{r}_i + \frac{T_i}{m_i} \cdot \mathbf{u}_i + \frac{\mathbf{D}_i}{m_i} \right) d\tau, \\ m(t_i) &= m(t_{i-1}) + \int_{t_{i-1}}^{t_i} \left(-\frac{T_i}{g_o \cdot Isp_i} \right) d\tau - \Delta m_i, \\ i &= 1, \dots, 4, \end{aligned} \tag{8.7}$$

where Δm_i is the mass dropped at the end of phase i . The final conditions are imposed so that the final states $\mathbf{r}(t_4)$ and $\mathbf{v}(t_4)$ satisfy the required orbital elements,

$$\begin{aligned}
a(t_f) &= f_a(\mathbf{r}(t_4), \mathbf{v}(t_4)) = a_f , \\
e(t_f) &= f_e(\mathbf{r}(t_4), \mathbf{v}(t_4)) = e_f , \\
i(t_f) &= f_i(\mathbf{r}(t_4), \mathbf{v}(t_4)) = i_f , \\
\Omega(t_f) &= f_\Omega(\mathbf{r}(t_4), \mathbf{v}(t_4)) = \Omega_f , \\
\omega(t_f) &= f_\omega(\mathbf{r}(t_4), \mathbf{v}(t_4)) = \omega_f .
\end{aligned} \tag{8.8}$$

The orbital elements, a, e, i, Ω , and ω , are the semi-major axis, eccentricity, inclination, right ascension of ascending node (RAAN), and argument of perigee respectively. The functions, f_a, f_e, f_i, f_Ω , and f_ω (Appendix C) are the relations between the inertial position and velocity, and the respective orbital elements [71]. These constraints define the final position and velocity of the payload to lie in the designated orbit, but it does not constrain the location within the orbit, usually defined by the true anomaly ν .

A state path constraint is imposed so that the vehicle's altitude is always above zero, so that

$$|\mathbf{r}_i(t)| \geq R_o , \tag{8.9}$$

where R_o is the radius of the Earth. This constraint is to ensure that the path of the spacecraft does not pass through the Earth. A path constraint is imposed on the control to guarantee that the control vector is of unit length, so that

$$|\mathbf{u}_i(t)| = 1 . \tag{8.10}$$

Finally, the objective is to maximize the fuel remaining, therefore the cost function is to minimize

$$J = -m(t_4) . \tag{8.11}$$

This cost function (8.11), along with the dynamic constraints (8.1), boundary con-

straints (8.5, 8.8), path constraints (8.9 - 8.10), and phase boundary constraints (8.7), define the continuous optimal control problem to be solved.

8.2 Gauss Pseudospectral Implementation

The integral form of the Delta 3 launch problem is discretized at a set of Gauss collocation points on each of the four phases of the problem. The NLP variables to be solved for, are defined as the position $\mathbf{r}_i \in \mathbb{R}^{N \times 3}$, velocity $\mathbf{v}_i \in \mathbb{R}^{N \times 3}$, mass $m_i \in \mathbb{R}^N$, and control $\mathbf{u}_i \in \mathbb{R}^{N \times 3}$ at the N Gauss points on the interior of each phase $i = 1, \dots, 4$. The initial state values for each phase are, $\mathbf{r}(t_{i-1}) \in \mathbb{R}^3$, $\mathbf{v}(t_{i-1}) \in \mathbb{R}^3$, and $m(t_{i-1}) \in \mathbb{R}$. Finally the final state values, $\mathbf{r}(t_4) \in \mathbb{R}^3$, $\mathbf{v}(t_4) \in \mathbb{R}^3$, and $m(t_4) \in \mathbb{R}$, along with the final time, $t_4 \in \mathbb{R}$ are included as NLP variables. There are a total of $10 \cdot N + 7$ variables for each phase along with the 7 final state variables and the final time. Therefore, there are a total of $40 \cdot N + 36$ NLP variables that are solved for in the problem.

The objective of the NLP is to minimize

$$J = -m(t_4), \quad (8.12)$$

subject to a set of nonlinear constraints. The approximation to the dynamic constraints using the pseudospectral integration approximation matrix are

$$\begin{aligned} \mathbf{r}_i - \mathbf{r}(t_{i-1}) - \frac{(t_i - t_{i-1})}{2} A(\mathbf{v}_i) &= 0, \\ \mathbf{v}_i - \mathbf{v}(t_{i-1}) - \frac{(t_i - t_{i-1})}{2} A\left(-\frac{\mu}{|r_i|^3} \cdot \mathbf{r}_i + \frac{T_i}{m_i} \cdot \mathbf{u}_i + \frac{\mathbf{D}_i}{m_i}\right) &= 0, \\ m_i - m(t_{i-1}) - \frac{(t_i - t_{i-1})}{2} A\left(-\frac{T_i}{g_o \cdot Isp_i}\right) &= 0, \\ i &= 1, \dots, 4, \end{aligned} \quad (8.13)$$

where drag \mathbf{D}_i , is defined by (8.2). The path constraint (8.9) is enforced at each node

on all phases. The initial conditions are

$$\begin{aligned}
\mathbf{r}(t_0) &= \mathbf{r}_o, \\
\mathbf{v}(t_0) &= \mathbf{v}_o, \\
m(t_0) &= m_o,
\end{aligned} \tag{8.14}$$

where \mathbf{r}_o , \mathbf{v}_o correspond to the launch site inertial position and velocity, and m_o is the initial mass of the vehicle. The phase boundary constraints are

$$\begin{aligned}
\mathbf{r}(t_i) - \mathbf{r}(t_{i-1}) - \frac{(t_i - t_{i-1})}{2} w^T (\mathbf{v}_i) &= 0, \\
\mathbf{v}(t_i) - \mathbf{v}(t_{i-1}) - \frac{(t_i - t_{i-1})}{2} w^T \left(-\frac{\mu}{|r_i|^3} \cdot \mathbf{r}_i + \frac{T_i}{m_i} \cdot \mathbf{u}_i + \frac{\mathbf{D}_i}{m_i} \right) &= 0, \\
m(t_i) - m(t_{i-1}) - \frac{(t_i - t_{i-1})}{2} w^T \left(-\frac{T_i}{g_o \cdot Isp_i} \right) + \Delta m(t_i) &= 0, \\
i &= 1, \dots, 4,
\end{aligned} \tag{8.15}$$

where w is the vector of Gauss quadrature weights. The state and control path constraints are enforced on all phases, so that

$$\begin{aligned}
|\mathbf{r}_i| &\geq R_o, \\
|\mathbf{u}_i| - 1 &= 0.
\end{aligned} \tag{8.16}$$

Finally the terminal constraints are

$$\begin{aligned}
f_a(\mathbf{r}(t_4), \mathbf{v}(t_4)) - a_f &= 0, \\
f_e(\mathbf{r}(t_4), \mathbf{v}(t_4)) - e_f &= 0, \\
f_i(\mathbf{r}(t_4), \mathbf{v}(t_4)) - i_f &= 0, \\
f_\Omega(\mathbf{r}(t_4), \mathbf{v}(t_4)) - \Omega_f &= 0, \\
f_\omega(\mathbf{r}(t_4), \mathbf{v}(t_4)) - \omega_f &= 0.
\end{aligned} \tag{8.17}$$

The cost function along with all the constraint equations (8.12 - 8.17) define the NLP

to be solved.

8.2.1 Numerical Values

The inertial coordinates of the initial condition in an Earth centered Cartesian frame, is determined using the location of the Cape Canaveral launch site, at a latitude of 28.5 degrees, for the initial position and velocity of the vehicle. The initial position and velocity are

$$\mathbf{r}_o = \begin{bmatrix} 5605.2 \\ 0 \\ 3043.4 \end{bmatrix} \text{ km}, \quad \mathbf{v}_o = \begin{bmatrix} 0 \\ 0.4076 \\ 0 \end{bmatrix} \text{ km/s}. \quad (8.18)$$

The initial mass, $m_o = 301454$ kg, is the total mass of the two main stages, nine solid boosters, and a payload of 4164 kg. The magnitude of the thrust and mass flow rate for each phase is determined using the vehicle engine properties (Table 8.1). The mass dropped at the end of each phase is the dry mass of the expended engines. At the end of the first phase six solid boosters are dropped, $\Delta m(t_1) = 13680$ kg, in the second three boosters are dropped, $\Delta m(t_2) = 6840$ kg, in the third the main engine is dropped, $\Delta m(t_3) = 8830$ kg, and finally there is no mass dropped at the end of the final phase, $\Delta m(t_4) = 0$.

The aerodynamic characteristics of the launch vehicle are assumed to be constant over the entire trajectory. The reference area is chosen to be $A_{ref} = 4\pi$ m² based on the diameter of the vehicle, and the coefficient of drag is $C_d = 0.5$, which is independent of angle of attack and Mach number. This value is determined from a typical drag coefficient of a similar shaped projectile [9]. The exponential atmosphere model (8.4) is chosen with a sea level density of $\rho_o = 1.225$ kg/m³ and a scale height of $h_o = 7.2$ km.

The length of each phase is determined by the burn time of the engines. The phase boundary times are, $t_0 = 0$, $t_1 = 75.2$, $t_2 = 150.4$, and $t_3 = 261$, in seconds, where t_4 is free. Finally, the terminal conditions are to put the payload into a geosynchronous

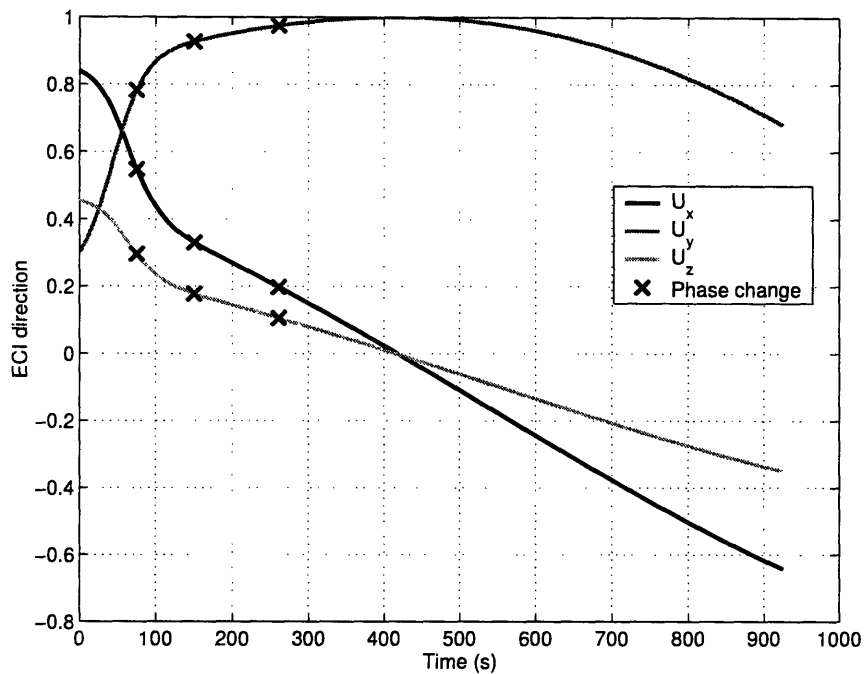


Figure 8-1: Delta III Control Direction

transfer orbit (GTO), which has the set of orbit elements

$$\begin{aligned}
 a_f &= 24361.14 \text{ km} , \\
 e_f &= 0.7308 , \\
 i_f &= 28.5 \text{ deg} , \\
 \Omega_f &= 269.8 \text{ deg} , \\
 \omega_f &= 130.5 \text{ deg} .
 \end{aligned}
 \tag{8.19}$$

The problem is scaled to change the magnitude of variables to be on the order of one. Distances are scaled by the radius of the Earth, 6378.14 km, velocities are scaled by the circular orbit velocity at Earth radius, 7.905 km/s, mass is scaled by the total vehicle mass 301454 kg, and forces are scaled by the vehicle weight 2956250 N. Using these scale factors, the resulting scale factor for time is 806.8 seconds.

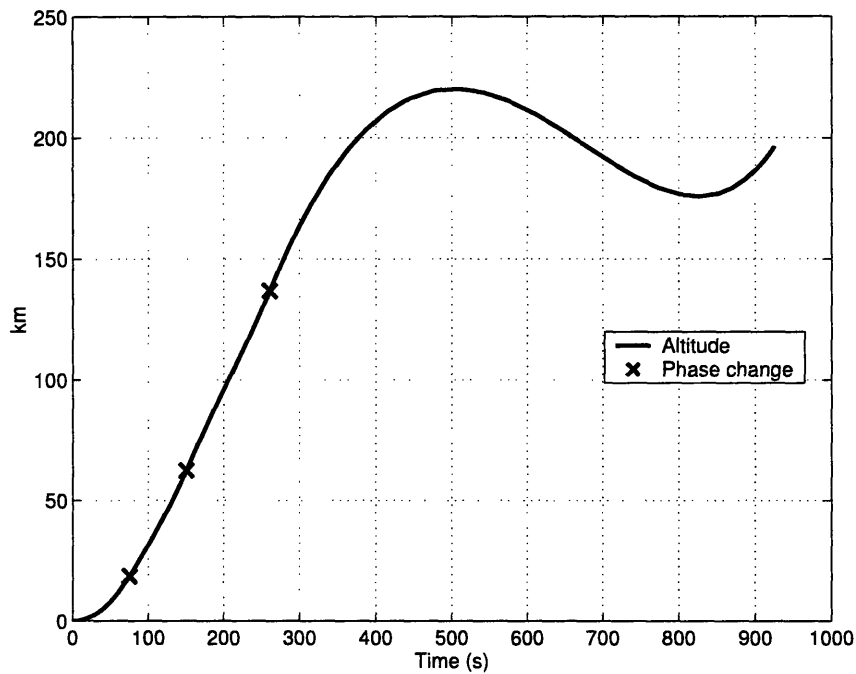


Figure 8-2: Delta III Vehicle Altitude Profile

8.2.2 Results

The launch vehicle optimization problem was solved in MATLAB using the TOMLAB [39] toolbox. The NLP was solved using the TOMLAB version of the sparse NLP solver SNOPT [30].

The optimal solution was found using 30 collocation nodes per phase. An initial guess was generated by integrating the system dynamics numerically using an arbitrary control direction, in this case the direction of the inertial velocity. The states and control generated in this way satisfies the initial conditions and the differential dynamic constraints, but does not satisfy the terminal constraints, nor are they optimal. Using this initial guess, SNOPT was able to solve the NLP in approximately two minutes.

The burn time of the second stage was found to be 665.3 seconds out of a possible 700 seconds. This leaves 668.2 kg of fuel remaining in the tank, or 4.0% of the total fuel available.

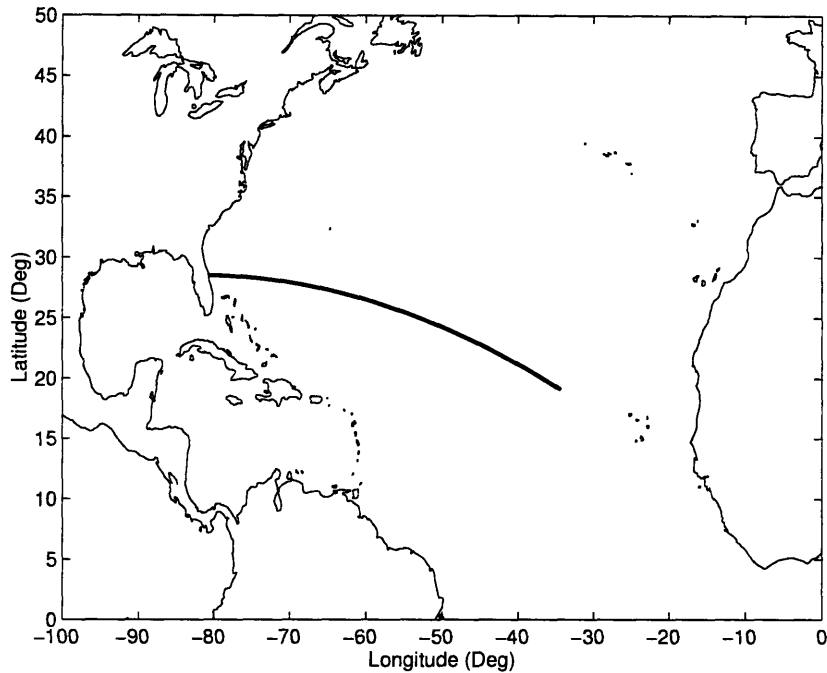


Figure 8-3: Delta III Vehicle Ground Track

The control direction is shown in Fig. 8-1, where U_x , U_y , and U_z are the x , y , z , ECI components of the unit thrust direction vector. The phase boundaries are also shown in the figure. The vehicle altitude profile is shown in Fig. 8-2, along with the phase boundaries. Finally, the ground track (Fig. 8-3) shows the vehicle path from lift-off at Cape Canaveral and its path over the Atlantic ocean.

This example demonstrates that the Gauss pseudospectral method can be used for the optimal trajectory design of the Delta III launch vehicle. In the next section the real time control approach (see Chapter 7) is used to control the launch vehicle in flight to correct for disturbances.

8.3 Real Time Control

The launch vehicle control problem is solved in real time by finding the estimate of the initial costates and integrating the state/costate adjoint equations forward in time. The optimal control is found by applying Pontryagin's maximum principle with

the optimal state and costates. At a future time, the new initial costate is estimated and the process is repeated until the final conditions are met. The two variables that effect the performance of the solution are the number of nodes used to solve the optimal control problem and how often the initial costate is reestimated. There is a trade off in these two variables. Using a larger number of nodes will give a better estimate of the initial costate, but will require longer computation times. Solving for the initial costate more often will correct any errors in the system faster, but it may not be possible to solve the optimal control problem fast enough.

The state dynamics for each phase of the problem is given in (8.1). The costate dynamics are found by the first-order necessary conditions (2.59), so that

$$\frac{d\boldsymbol{\lambda}}{dt} = -\frac{\partial \mathcal{H}^T}{\partial \mathbf{x}}, \quad (8.20)$$

where \mathcal{H} is the Hamiltonian and \mathbf{x} is the vector of states. The Hamiltonian for the launch vehicle optimization problem is

$$\begin{aligned} \mathcal{H} = & \boldsymbol{\lambda}_r^T \cdot \mathbf{v} + \boldsymbol{\lambda}_v^T \cdot \left(-\frac{\mu}{|r|^3} \cdot \mathbf{r} + \frac{T}{m} \cdot \mathbf{u} + \frac{\mathbf{D}}{m} \right) \\ & + \lambda_m \cdot \left(-\frac{T}{g_o \cdot Isp} \right) + c \cdot (|\mathbf{u}| - 1), \end{aligned} \quad (8.21)$$

where $\boldsymbol{\lambda}_r(t) \in \mathbb{R}^3$, $\boldsymbol{\lambda}_v(t) \in \mathbb{R}^3$, and $\lambda_m(t) \in \mathbb{R}$ are the costates and $c(t) \in \mathbb{R}$ is the Lagrange function associated with the control path constraint. The costate dynamics are derived as

$$\begin{aligned} \frac{d\boldsymbol{\lambda}_r}{dt} &= -\frac{\partial H}{\partial \mathbf{r}} = \frac{\mu}{|r|^3} \boldsymbol{\lambda}_v - 3 \frac{\mu}{|r|^5} (\boldsymbol{\lambda}_v^T \cdot \mathbf{r}) \cdot \mathbf{r} - \frac{\partial \mathbf{D}^T}{\partial \mathbf{r}} \cdot \frac{\boldsymbol{\lambda}_v}{m}, \\ \frac{d\boldsymbol{\lambda}_v}{dt} &= -\frac{\partial H}{\partial \mathbf{v}} = -\boldsymbol{\lambda}_r - \frac{\partial \mathbf{D}^T}{\partial \mathbf{v}} \cdot \frac{\boldsymbol{\lambda}_v}{m}. \end{aligned} \quad (8.22)$$

Details of derivation appear in Appendix D.

The optimal control for the launch vehicle problem is determined by applying

Pontryagin's maximum principle, so that

$$\frac{\partial \mathcal{H}^T}{\partial \mathbf{u}} = \frac{T}{m} \boldsymbol{\lambda}_v + c \frac{\mathbf{u}}{|\mathbf{u}|} = \mathbf{0}. \quad (8.23)$$

Solving for the control, and using the fact that the magnitude is constrained to be one, results in

$$\mathbf{u} = -\frac{T}{c \cdot m} \boldsymbol{\lambda}_v. \quad (8.24)$$

Again the fact that the control is unit length is used to determine the value of the Lagrange multiplier c . The multiplier is then

$$\begin{aligned} |\mathbf{u}| &= \frac{T}{c \cdot m} |\boldsymbol{\lambda}_v| = 1, \\ c &= \frac{T}{m} |\boldsymbol{\lambda}_v|. \end{aligned} \quad (8.25)$$

Using the value of the multiplier in the constraint (8.24) results in the optimal control. The control is therefore determined to be in the opposite direction of the velocity costate, so that

$$\mathbf{u} = -\frac{\boldsymbol{\lambda}_v}{|\boldsymbol{\lambda}_v|}. \quad (8.26)$$

In summary, the dynamics of the system are defined by (8.1) for the states, (8.22) for the costates, and (8.26) for the control. Note that the costate λ_m is not needed, because it does not appear in any of the relations for the state, costates, or control. Given the initial states and costates, the solution to the trajectory problem can be generated by integrating the states and costates forward in time, while applying the optimal control. The initial costate is estimated using the Gauss pseudospectral method on the integral form of the problem. The states and costates are integrated forward for a small time interval, and then the problem is resolved using the current states as the initial conditions, and the current costate is updated. This approach will correct for errors generated by not using the exact costate, as well as any disturbances or numerical integration error.

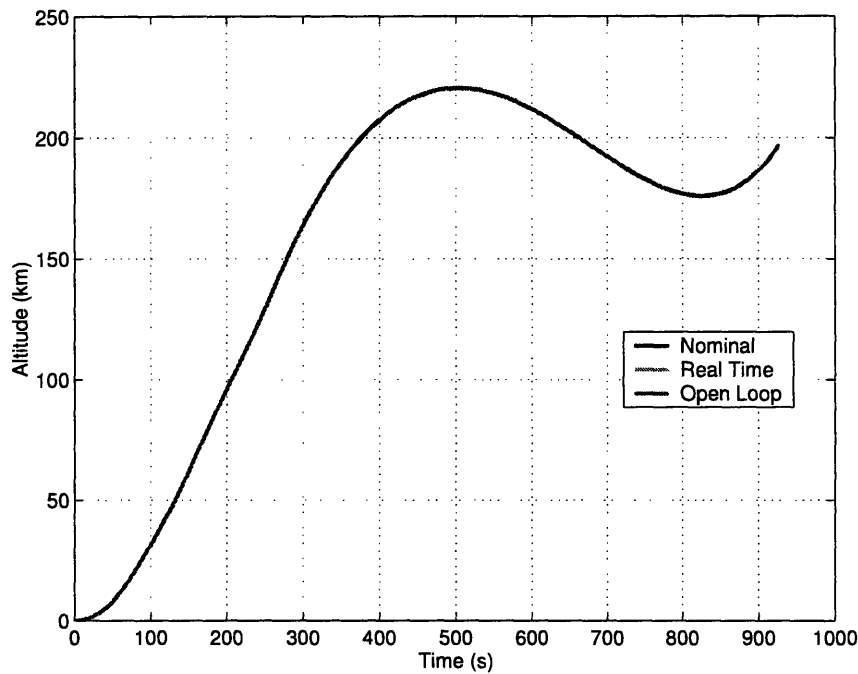


Figure 8-4: Delta III Real Time Altitude Profile

8.3.1 Results

The Delta III launch vehicle trajectory optimization problem was solved using the real time approach first without any disturbances. The solution was found using 5 nodes per phase and updating the solution approximately every 40.3 seconds of simulation time. The average time required to solve the NLP was approximately 0.5 seconds. This solution will be referred to as the real time solution. The system equations were propagated forward in time using the MATLAB ode45 numerical integrator.

These results are then compared to the trajectory found by integrating the vehicle dynamics (8.1) using the control found from interpolating the pseudospectral solution in Section 8.2.2. This solution will be referred to as the open loop solution. The altitude profiles for both methods are shown in Fig. 8-4, and compared to the nominal profile computed in Fig. 8-2. A summary of the terminal condition errors in the semi-major axis and eccentricity are shown in Table 8.2.

As expected, both the real time and open loop solutions are nearly identical, and

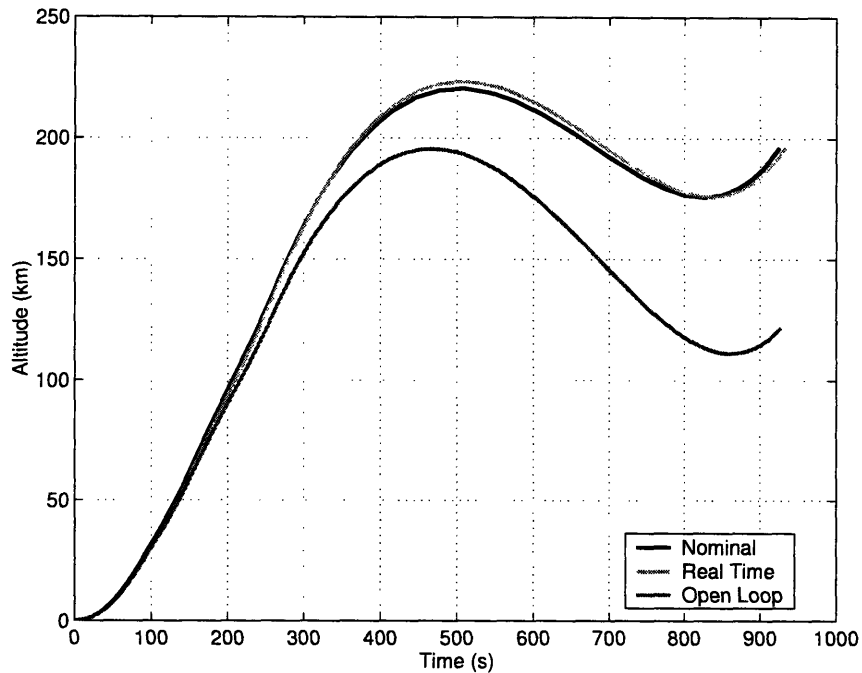


Figure 8-5: Delta III Real Time Altitude Profile with Disturbance

Table 8.2: Delta III Trajectory Errors

	Error	Relative Error
Real Time Semi-Major Axis	0.259 km	$1.07 \cdot 10^{-3}\%$
Real Time Eccentricity	$3.15 \cdot 10^{-6}$	$4.31 \cdot 10^{-4}\%$
Open Loop Semi-Major Axis	0.006 km	$2.45 \cdot 10^{-5}\%$
Open Loop Eccentricity	$7.59 \cdot 10^{-8}$	$1.04 \cdot 10^{-5}\%$

meet the terminal orbit conditions. This result is due to the fact that there were no disturbances considered in the problem.

There were significantly different results when a small disturbance was added. The thrust on the solid rocket boosters was changed to 98% of its nominal value, and the real time approach was compared to the open loop result. The altitude profiles are shown in Fig. 8-5, and the terminal condition errors are summarized in Table 8.3.

The results indicate that the real time approach was able to correct for the disturbance and still meet the terminal conditions. The new real time solution required the second stage to burn for an additional 7 seconds, leaving 504.9 kg of fuel, or 3.0%

Table 8.3: Delta III Trajectory Errors with Disturbance

	Error	Relative Error
Real Time Semi-Major Axis	0.307 km	$1.26 \cdot 10^{-3}\%$
Real Time Eccentricity	$3.76 \cdot 10^{-6}$	$5.14 \cdot 10^{-4}\%$
Open Loop Semi-Major Axis	1650 km	6.77%
Open Loop Eccentricity	$1.67 \cdot 10^{-2}$	2.27 %

(compared to 4.0% for the original solution). The open loop solution did not correct for the disturbance and resulted in significant errors in the terminal conditions.

8.4 Summary

In this chapter, the trajectory optimization of a Delta III launch vehicle was computed using a simplified dynamic model. The optimal control problem was defined to find the thrust direction of the launch vehicle to place a payload into a desired geosynchronous transfer orbit (GTO), while minimizing the amount of fuel used. The optimization problem was solved using the Gauss pseudospectral method on the integral form of the problem. It was shown that a trajectory could be found to achieve the trajectory parameters that used approximately 96.0% of the fuel in the second stage.

The real time control approach using the initial costate was used to correct the trajectory in flight. It was shown that the real time control algorithm was able to find the optimal control that corrected for accumulating error, as well as an applied disturbance. For the case with the applied disturbance (an error in the solid booster thrust) the nominal control was unable to achieve the trajectory parameters. The real time control algorithm, however, was able to find the optimal control to meet the orbit parameters. The optimal trajectory in this case, used approximately 97.0% of the total fuel in the second stage. The results demonstrate that the initial costate can be used for real time control of the simplified launch vehicle model.

Chapter 9

Conclusion

There are many numerical methods to approximate the solution of an optimal control problem. These methods can generally be put into one of two categories, indirect and direct. Indirect methods attempt to find a solution to the optimal control problem by approximating the first order necessary conditions derived from the calculus of variations and Pontryagin's maximum principle. Direct methods convert the continuous control problem into a discrete nonlinear programming problem (NLP). The resulting NLP can then be solved by well-developed NLP algorithms. Indirect methods generally are more accurate, while direct methods have simpler, more convenient formulations and are more robust. Pseudospectral methods for solving optimal control problems are a class of direct transcription methods that are based on spectral methods, which were developed for solving partial differential equations. The Legendre pseudospectral method with nodes at Lagrange-Gauss-Lobatto points is a formulation where costate estimates have been derived directly from the Karush-Kuhn-Tucker (KKT) multipliers from the resulting NLP. However, the method suffers from a defect in the costate estimates at the boundary points. At these points the costate estimates do not satisfy the costate boundary conditions or the discretized costate dynamics. This deficiency results in a relatively poor estimate of the costate especially at the boundary points.

In this thesis, a direct pseudospectral transcription method has been developed based on collocation at Gauss points, which do not include the boundary points. The

Gauss pseudospectral method has been derived from both the integral and differential forms of the optimal control problem. It has been shown that the continuous optimal control problem in integral form is exactly equivalent to the continuous problem in differential form. In the integral form, the optimal control problem is discretized using pseudospectral approximations of the integral of the differential dynamic constraints. The problem was originally solved in this way to allow for the discretization at Gauss points while still enforcing the boundary conditions. It was later found that the discretization of the problem could be made directly from the differential form of the optimal control problem. It has been shown that the differential and integral forms of the Gauss pseudospectral method are mathematically equivalent and result in the same solution. The differential form, however, results in a more sparse NLP that can be solved faster.

The primary property that distinguishes the Gauss pseudospectral method from other pseudospectral methods is the fact that the dynamic equations are not collocated at the boundary points. As a result the KKT conditions of the resulting NLP are exactly equivalent to the discretized form of the first-order necessary conditions. This result indicates that the Gauss method does not suffer from a defect in the costate estimates. The fact that the optimality conditions of the Gauss method are consistent with the continuous first-order necessary conditions allows the method to take advantage of the convenient formulations and robustness of direct methods, while preserving the accuracy of indirect methods.

Empirical evidence has suggested that the Gauss pseudospectral method converges rapidly (exponentially) for a large class of problems and gives a better costate estimate than the Legendre pseudospectral method. These advantages have been shown on several linear and nonlinear example problems. However, it has been shown that the Gauss pseudospectral method is not well suited for solving problems that have discontinuities in the solution (or discontinuities in the derivatives of the solution), problems with singularities on or near the solution interval, and problems that contain singular arcs. For some problems with discontinuities (such as bang-bang control), good solutions can be found by dividing the optimal control problem into multiple

phases once the switching structure is known.

Additionally, the Gauss pseudospectral method is well-suited for real time optimal control. The method has the advantage of rapid convergence, which allows for accurate solutions that can be found quickly, as well as providing a very good estimate for the initial costate. This initial costate has been used in the development of a real time optimal control algorithm for nonlinear systems. In this approach, the state and costate dynamic equations are integrated forward in time using the state and estimate for the initial costate. The optimal control is then defined from Pontryagin's maximum principle. The costate is updated at a later time by re-solving the optimal control problem. By continually updating the costate, the algorithm is able to correct for the accumulation of error, changes in optimization parameters, modeling errors, and applied disturbances. The real time control approach has been demonstrated on several examples include the trajectory optimization of a Delta III launch vehicle.

[This page intentionally left blank.]

Appendix A

MATLAB Code

Included are the MATLAB scripts that were used to generate the Gauss points, weights, and Gauss matrices for both the integral and differential formulations.

A.1 Gauss Points

```
function r = legroots(N);  
% The function r = legroots(N) computes the roots of the  
% Legendre polynomial of degree N.  
  
% J.A.C. Weideman, S.C. Reddy 1998.  
  
n = [1:N-1];           % Indices  
d = n./sqrt(4*n.^2-1); % Create subdiagonals  
J = diag(d,1)+diag(d,-1); % Create Jacobi matrix  
r = sort(eig(sparse(J))); % Compute eigenvalues
```

A.2 Gauss Weights

```
function [x, w] = gauss_points(n)  
% function returns weights and points for gauss quadrature
```

```

x = legroots(n);

Pnp1 = legendre(n+1,x);
Pnp1 = Pnp1(1,:)' ;
Pndot = -(n+1)./(1-x.^2).*Pnp1;
w = 1./(Pndot).^2.*(2./(1-x.^2));

```

A.3 Integration Approximation Matrix

```

function [A, x, w] = legint(n)
% Function generates the Gauss collocation points and weights (x, w) and the
% integration approximation matrix A

[x, w] = gauss_points(n);

if n == 1
    A = w/2;
    return
end

g = 1;

Pn = zeros(n+1,n);
for v = 0:n
    P = legendre(v,x);
    if v == 0
        P = P';
    end
    Pn(v+1,:) = P(1,:);

```

```

end

% loop through i,k
for i = 1:n;
    for k = 1:n;
        SUMP = 0;
        for v = 1:n-2
            SUMP = SUMP + Pn(v+1,k)*(Pn(v+2,i) - Pn(v,i));
        end
        A(i,k) = w(k)/2*(1+x(i) + SUMP + g*Pn(n,k)*(Pn(n+1,i) - Pn(n-1,i)));
    end
end
end

```

A.4 Differential Gauss Approximation Matrix

```

function [Dg, Dbar, x, w] = legdiff_G(n)
% Function generates the Gauss collocation points and weights (x, w) and the
% Differential approximation matrix D, bar D

% Gauss Pts
[x, w] = gauss_points(n);
% Add initial point -1
x = [x; -1];
x = sort(x);
n = n+1;

% Eval derivative of Lagrange polynomials
for j = 1:n
    for i = 1:n;
        prod = 1;

```

```

sum = 0;
if j == i
    for k = 1:n
        if k~=i
            sum = sum+1/(x(i)-x(k));
        end
    end
    D(i,j) = sum;
else
    for k = 1:n
        if (k~=i)&(k~=j)
            prod = prod * (x(i)-x(k));
        end
    end
    for k = 1:n
        if k~=j
            prod = prod/(x(j)-x(k));
        end
    end
    D(i,j) = prod;
end
end
end
Dg = D(2:end,2:end);
Dbar = D(2:end,1);
x = x(2:end);

```

Appendix B

LQR Pseudospectral Solution

The solution to the LQR problem can be found by solving the KKT conditions of the NLP, which is the direct solution, or by discretizing the continuous necessary conditions, which is the indirect solution. Both methods involve solving a set of linear equations. These equations only differ by the use of the integral costate mapping principle (4.76).

B.1 Indirect Solution

The indirect pseudospectral solution to the LQR problem can be found easily from the first order necessary conditions because they are all linear. The continuous necessary conditions for the one dimensional LQR problem, described in Section 6.1, in integral form are

$$\begin{aligned}x(t) &= x_o + \int_0^t (x(\tau) + u(\tau)) d\tau , \\p(t) &= x(t) + \int_t^5 p(\tau) d\tau - \nu_f , \\0 &= u(t) + \int_t^5 p(\tau) d\tau - \nu_f ,\end{aligned}\tag{B.1}$$

with boundary conditions,

$$\begin{aligned}
x_o &= 1, \\
x_f &= 0, \\
\int_0^5 p(t) dt - \nu_f &= \nu_o, \\
x_f &= x_o + \int_0^5 (x(t) + u(t)) dt.
\end{aligned} \tag{B.2}$$

The discretized form of the first order necessary conditions using the integration approximation matrix, A , and its adjoint, A^\dagger , along with the Gauss weights as a vector w , are

$$\begin{aligned}
\mathbf{X}_N &= X_o + \frac{5}{2} \cdot A \cdot (\mathbf{X}_N + \mathbf{U}_N), \\
\mathbf{P}_N &= \mathbf{X}_N + \frac{5}{2} A^\dagger \cdot \mathbf{P}_N - \nu_f, \\
\mathbf{0} &= \mathbf{U}_N + \frac{5}{2} \cdot A^\dagger \cdot \mathbf{P}_N - \nu_f, \\
X_o &= 1, \\
X_f &= 0, \\
\frac{5}{2} \cdot w^T \cdot \mathbf{P}_N - \nu_f &= \nu_o, \\
X_f &= X_o + \frac{5}{2} \cdot w^T \cdot (\mathbf{X}_N + \mathbf{U}_N),
\end{aligned} \tag{B.3}$$

where $\mathbf{X}_N, \mathbf{U}_N, \mathbf{P}_N \in \mathbb{R}^N$ are the approximations to the state, control, and integral costate, respectively at the N Gauss points. The variables, X_o and X_f , are the boundaries of the state, and ν_o and ν_f are the Lagrange multipliers of the boundary constraints. Note that the fraction $\frac{5}{2}$ was used in the equations to transfer the time interval from $[0, 5]$ to $[-1, 1]$.

The linear system of equations (B.3), involves $3 \cdot N + 4$ equations and $3 \cdot N + 4$ unknowns, which can be solved by the inversion of a matrix. Expressing in matrix

form results in

$$\begin{bmatrix} \frac{5}{2} \cdot A - I & \frac{5}{2} \cdot A & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ I & \mathbf{0} & \frac{5}{2} \cdot A^\dagger - I & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{1} \\ \mathbf{0} & I & \frac{5}{2} \cdot A^\dagger & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \frac{5}{2} \cdot w^T & \frac{5}{2} \cdot w^T & \mathbf{0} & \mathbf{1} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{5}{2} \cdot w^T & \mathbf{0} & \mathbf{0} & -\mathbf{1} & -\mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_N \\ \mathbf{U}_N \\ \mathbf{P}_N \\ X_o \\ X_f \\ \nu_o \\ \nu_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} . \quad (\text{B.4})$$

The solution to the problem can be found by inverting the matrix on the left hand side.

The estimates for the differential costates $\lambda_N \in \mathbb{R}^N$ can be found from the integral costates as

$$\lambda_N = \frac{5}{2} \cdot A^\dagger \cdot \mathbf{P}_N - \nu_f , \quad (\text{B.5})$$

and the estimates for the initial and final costates are

$$\begin{aligned} \lambda_o &= \nu_o , \\ \lambda_f &= -\nu_f . \end{aligned} \quad (\text{B.6})$$

B.2 KKT Conditions

The NLP found from the LQR problem in Section 6.1 is to minimize

$$J = \frac{1}{2} \cdot \frac{5}{2} \cdot w^T \cdot (\mathbf{X}_N^2 + \mathbf{U}_N^2) , \quad (\text{B.7})$$

subject to the constraints,

$$\begin{aligned}
\mathbf{X}_N &= X_o + \frac{5}{2} \cdot A \cdot (\mathbf{X}_N + \mathbf{U}_N) , \\
X_o &= 1 , \\
X_f &= 0 , \\
X_f &= X_o + \frac{5}{2} \cdot w^T \cdot (\mathbf{X}_N + \mathbf{U}_N) .
\end{aligned} \tag{B.8}$$

The KKT conditions are found by adjoining the cost with the constraints and KKT multipliers $\tilde{\mathbf{P}}_N \in \mathbb{R}^N$ and Lagrange multipliers ν_o and ν_f , and setting the gradient to zero. The resulting conditions are

$$\begin{aligned}
\mathbf{X}_N &= X_o + \frac{5}{2} \cdot A \cdot (\mathbf{X}_N + \mathbf{U}_N) , \\
\tilde{\mathbf{P}}_N &= \frac{5}{2} \cdot W \cdot \mathbf{X}_N + \frac{5}{2} \cdot A^T \cdot \tilde{\mathbf{P}}_N - \frac{5}{2} \cdot \nu_f \cdot w , \\
\mathbf{0} &= \frac{5}{2} \cdot W \cdot \mathbf{U}_N + \frac{5}{2} \cdot A^T \cdot \tilde{\mathbf{P}}_N - \frac{5}{2} \cdot \nu_f \cdot w , \\
X_o &= 1 , \\
X_f &= 0 , \\
X_f &= X_o + \frac{5}{2} \cdot w^T \cdot (\mathbf{X}_N + \mathbf{U}_N) , \\
0 &= \mathbf{1}^T \cdot \tilde{\mathbf{P}}_N - \nu_o - \nu_f ,
\end{aligned} \tag{B.9}$$

where W is a matrix with the Gauss weights on the diagonals. The set of equations (B.9) can be shown to be exactly the same as the set of equations (B.3) by using the integral costate mapping principle (4.76) and the definition of the integration

operator adjoint (Lemma 4.2.1). The set of equations in matrix form is

$$\begin{bmatrix} \frac{5}{2} \cdot A - I & \frac{5}{2} \cdot A & \mathbf{0} & 1 & 0 & 0 & 0 \\ \frac{5}{2} \cdot W & \mathbf{0} & \frac{5}{2} \cdot A^T - I & 0 & 0 & 0 & -\frac{5}{2} \cdot w \\ \mathbf{0} & \frac{5}{2} \cdot W & \frac{5}{2} \cdot A^T & 0 & 0 & 0 & -\frac{5}{2} \cdot w \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 0 & 0 \\ \frac{5}{2} \cdot w^T & \frac{5}{2} \cdot w^T & \mathbf{0} & 1 & -1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1}^T & 0 & 0 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_N \\ \mathbf{U}_N \\ \tilde{\mathbf{P}}_N \\ X_o \\ X_f \\ \nu_o \\ \nu_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} . \quad (\text{B.10})$$

The solution to this set of linear equations is equivalent to the solution of the NLP (B.7 - B.8).

The estimate for the integral costate, \mathbf{P}_N , is found from the KKT multipliers, $\tilde{\mathbf{P}}_N$, by use of the integral costate mapping principle, so that

$$\mathbf{P}_N = \frac{2}{5} \cdot W^{-1} \cdot \tilde{\mathbf{P}}_N . \quad (\text{B.11})$$

The estimate for the differential costates, λ_N , can also be found directly from the KKT multipliers, so that

$$\lambda_N = W^{-1} \cdot A^T \cdot \tilde{\mathbf{P}}_N - \nu_f , \quad (\text{B.12})$$

with initial and final costates as

$$\begin{aligned} \lambda_o &= \nu_o , \\ \lambda_f &= -\nu_f . \end{aligned} \quad (\text{B.13})$$

This derivation demonstrates that the NLP resulting from the continuous optimal control problem can be solved by forming the KKT conditions and solving the resulting set of equations. This approach is particularly easy for the LQR problem because the resulting KKT conditions are all linear.

[This page intentionally left blank.]

Appendix C

Orbital Elements

The orbital elements can be determined from the inertial position \mathbf{r} and velocity \mathbf{v} [71]. The first step is to find the angular momentum \mathbf{h} ,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} . \tag{C.1}$$

Next is the vector pointing to the node,

$$\mathbf{n} = \vec{\mathbf{k}} \times \mathbf{h} , \tag{C.2}$$

where $\vec{\mathbf{k}}$ is the unit vector in the z direction. The eccentricity vector is defined as

$$\mathbf{e} = \frac{\left(|v|^2 - \frac{\mu}{|r|} \right) \cdot \mathbf{r} - (\mathbf{r} \bullet \mathbf{v}) \cdot \mathbf{v}}{\mu} , \tag{C.3}$$

and the eccentricity is the magnitude of the vector, so that

$$e = |\mathbf{e}| . \tag{C.4}$$

The semi-major axis is found from the energy of the orbit, so that

$$E = \frac{|v|^2}{2} - \frac{\mu}{|r|},$$

$$a = \frac{-\mu}{2E}.$$
(C.5)

The inclination is the angle between the angular momentum vector and the z direction, so that

$$\cos i = \frac{h_k}{|h|}.$$
(C.6)

The right ascension of the ascending node is the angle between the node vector and the x direction, so that

$$\cos \Omega = \frac{n_x}{|n|}.$$
(C.7)

If the y component of the node vector is negative, $n_y < 0$, then a quadrant correction must be made, $\Omega = 2\pi - \Omega$. Finally the argument of perigee is the angle between the node vector and the eccentricity vector, so that

$$\cos \omega = \frac{\mathbf{n} \cdot \mathbf{e}}{|n| \cdot |e|}.$$
(C.8)

A quadrant correction is also needed if $e_x < 0$, then $\omega = 2\pi - \omega$.

Appendix D

Launch Vehicle Costate Dynamics

The costate dynamics are derived from the Hamiltonian,

$$\begin{aligned}
 H = & \lambda_r^T \cdot \mathbf{v} + \lambda_v^T \cdot \left(-\frac{\mu}{|r|^3} \cdot \mathbf{r} + \frac{T}{m} \cdot \mathbf{u} + \frac{\mathbf{D}}{m} \right) \\
 & + \lambda_m \cdot \left(-\frac{T}{g_o \cdot Isp} \right) + c \cdot (|\mathbf{u}| - 1) ,
 \end{aligned} \tag{D.1}$$

where the drag is defined as

$$\mathbf{D} = -\frac{1}{2} C_d \cdot A_{ref} \cdot \rho \cdot |v_{rel}| \cdot \mathbf{v}_{rel} , \tag{D.2}$$

with the relative velocity and density as

$$\begin{aligned}
 \mathbf{v}_{rel} &= \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r} , \\
 \rho &= \rho_o \exp[-(|r| - r_{earth})/h_o] .
 \end{aligned} \tag{D.3}$$

The rotation rate of the earth is $\boldsymbol{\omega} = [0, 0, w_e]$. The costate dynamics are found by the partial derivatives of the Hamiltonian (2.59). Writing out the scalar components of the states,

$$\begin{aligned}
 \mathbf{r} &= [r_x, \quad r_y, \quad r_z]^T , \\
 \mathbf{v} &= [v_x, \quad v_y, \quad v_z]^T ,
 \end{aligned} \tag{D.4}$$

and the costates,

$$\begin{aligned}\boldsymbol{\lambda}_r &= [\lambda_{rx}, \lambda_{ry}, \lambda_{rz}]^T, \\ \boldsymbol{\lambda}_v &= [\lambda_{vx}, \lambda_{vy}, \lambda_{vz}]^T,\end{aligned}\tag{D.5}$$

and the relative velocity,

$$\mathbf{v}_{rel} = [v_x - w_e \cdot r_y, v_y + w_e \cdot r_x, v_z]^T,\tag{D.6}$$

the costate dynamics become

$$\begin{aligned}\frac{d\boldsymbol{\lambda}_r}{dt} &= -\frac{\partial H}{\partial \mathbf{r}}, \\ \frac{d\boldsymbol{\lambda}_v}{dt} &= -\frac{\partial H}{\partial \mathbf{v}}.\end{aligned}\tag{D.7}$$

Expressed term by term for the position costates,

$$\begin{aligned}\dot{\lambda}_{rx} &= \lambda_{vx} \cdot \frac{\mu}{|r|^3} - 3 \cdot r_x \frac{\mu}{|r|^5} \cdot \boldsymbol{\lambda}_v^T \mathbf{r} \\ &\quad - \frac{1}{2 \cdot m} C_d \cdot A_{ref} \cdot \rho \left[\frac{r_x}{h_o \cdot |r|} |v_{rel}| \cdot \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} + w_e \frac{(v_y - w_e \cdot r_x)}{|v_{rel}|} \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} - |v_{rel}| w_e \lambda_{vy} \right], \\ \dot{\lambda}_{ry} &= \lambda_{vy} \cdot \frac{\mu}{|r|^3} - 3 \cdot r_y \frac{\mu}{|r|^5} \cdot \boldsymbol{\lambda}_v^T \mathbf{r} \\ &\quad - \frac{1}{2 \cdot m} C_d \cdot A_{ref} \cdot \rho \left[\frac{r_y}{h_o \cdot |r|} |v_{rel}| \cdot \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} - w_e \frac{(v_x + w_e \cdot r_y)}{|v_{rel}|} \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} - |v_{rel}| w_e \lambda_{vx} \right], \\ \dot{\lambda}_{rz} &= \lambda_{vz} \cdot \frac{\mu}{|r|^3} - 3 \cdot r_z \frac{\mu}{|r|^5} \cdot \boldsymbol{\lambda}_v^T \mathbf{r} - \frac{1}{2 \cdot m} C_d \cdot A_{ref} \cdot \rho \cdot \frac{r_z}{h_o \cdot |r|} |v_{rel}| \cdot \boldsymbol{\lambda}_v^T \mathbf{v}_{rel},\end{aligned}\tag{D.8}$$

and term by term for the velocity costates,

$$\begin{aligned}\dot{\lambda}_{vx} &= -\lambda_{rx} + \frac{1}{2 \cdot m} C_d \cdot A_{ref} \cdot \rho \left[\frac{(v_x + w_e \cdot r_y)}{|v_{rel}|} \cdot \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} + \lambda_{vx} \cdot |v_{rel}| \right], \\ \dot{\lambda}_{vy} &= -\lambda_{ry} + \frac{1}{2 \cdot m} C_d \cdot A_{ref} \cdot \rho \left[\frac{(v_y - w_e \cdot r_x)}{|v_{rel}|} \cdot \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} + \lambda_{vy} \cdot |v_{rel}| \right], \\ \dot{\lambda}_{vz} &= -\lambda_{rz} + \frac{1}{2 \cdot m} C_d \cdot A_{ref} \cdot \rho \left[\frac{(v_z)}{|v_{rel}|} \cdot \boldsymbol{\lambda}_v^T \mathbf{v}_{rel} + \lambda_{vz} \cdot |v_{rel}| \right].\end{aligned}\tag{D.9}$$

Bibliography

- [1] Andrews Space & Technology, <http://www.spaceandtech.com/spacedata/elvs>.
- [2] Axelsson, O., "Global Integration of Differential Equations Through Lobatto Quadrature," *BIT* 4, 1964, pp. 69-86.
- [3] Bellman, R.E., Kalaba, R.E., *Quasilinearization and Nonlinear Boundary Value Problems*, American Elsevier, New York, 1965.
- [4] Bertsekas, D., *Nonlinear Programming*, Athena Scientific, 1995.
- [5] Betts, John T., "A Direct Approach to Solving Optimal Control Problems," *Computing in Science and Engineering*, May-June 1999, pp. 73-75.
- [6] Betts, John T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, March-April 1998.
- [7] Betts, John T., *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM, 2001.
- [8] Bock, H.G. Plitt, K.J., "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *IFAC 9th World Congress*, Budapest, Hungary, 1984.
- [9] Braeunig, Robert A., <http://users.commkey.net/Braeunig/space/index.htm>.
- [10] Bryson, A., Ho, Y., *Applied Optimal Control*, Blaisdell Publishing Co., 1969.
- [11] Bryson, A., *Dynamic Optimization*, Addison Wesley Longman, 1999.

- [12] Burden, R., Faires, J., Reynolds, A., *Numerical Analysis*, Prindle, Weber & Schmidt, 1978.
- [13] Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.
- [14] Clenshaw, C.W., "The Numerical Solution of Linear Differential Equations in Chebyshev Series," *Proc. Cambridge Philos. Soc.*, 53, pp. 134-149, 1957.
- [15] Clenshaw, C.W., Norton, H.J., "The Solution of Nonlinear Differential Equations in Chebyshev Series," *Comput. J.*, 6, pp. 88-92, 1963.
- [16] Davis, P., Rabinowitz, P., *Methods of Numerical Integration*, Academic Press, 1984.
- [17] Davis, P., *Interpolation & Approximation*, Dover Publications, 1975.
- [18] Dickmanns, E.D., Well, K.H., *Approximate Solution of Optimal Control Problems Using Third Order Hermite Polynomial Functions*, Lecture Notes in Computational Science, Vol. 27, Springer, Heidelberg, 1975, pp. 158 - 166.
- [19] 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.
- [20] Elnagar, G., Kazemi, M., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, 1998, pp. 195-217.
- [21] Enright, P.J., 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.
- [22] Fahroo, F., Ross, M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270-277.

- [23] Fahroo, F., Ross, M., “Direct Trajectory Optimization by a Chebyshev Pseudospectral Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160-166.
- [24] Finlayson, B.A., Scriven, L.E., “The Method of Weighted Residuals - a review,” *Appl. Mech.*, Rev. 19, 1966, pp. 735-748.
- [25] Floudas, C.A., *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic, 1999.
- [26] Fornberg, B., *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, 1998.
- [27] Fourier, J.B.J., *Théorie analytique de la chaleur*, Didot, Paris, 1822.
- [28] Froberg, C., *Introduction to Numerical Analysis*, Addison-Wesley, 1965.
- [29] Funaro, D., *Polynomial Approximation of Differential Equations*, Springer-Verlag, 1992.
- [30] 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.
- [31] Gottlieb, D., Orszag, S.A., *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, 1977.
- [32] Gottlieb, D., Hussaini, M.Y., Orszag, S.A., *Introduction: Theory and Applications of Spectral Methods*, R.G. Voigt, D. Gottlieb, and M.Y. Hussaini, eds., *Spectral Methods for Partial Differential Equations*, SIAM, 1984.
- [33] Hager, W.W., “Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System,” *Numer. Math.*, Vol. 87, 2000, pp. 247-282.
- [34] Hager, W.W., “Rate of Convergence for Discrete Approximations to Unconstrained Control Problems,” *SIAM J. Numer. Anal.*, Vol. 13, 1976, pp. 449-471.

- [35] Hargraves, C.R., 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.
- [36] *Handbook of Mathematical Functions*, ed. M. Abramowitz and I. Stegun, National Bureau of Standards, 1964.
- [37] Herman, A.L., 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.
- [38] Ho, Yu-Chi, "Neuro-fuzzy And Soft Computing - A Computational Approach To Learning And Machine Intelligence [Book Reviews]," *IEEE Trans. on Automatic Control*, Vol. 86, No. 3, March 1998.
- [39] Holmstrom, K. and Goran, A., "User's Guide for TOMLAB v.3.2.1," September 2002.
- [40] Horn, M.K., "Solution of the Optimal Control Problem Using the Software Package STOPM," *8th IFAC Workshop on Control Applications of Nonlinear Programming and Optimization*, Paris, 1989.
- [41] Isaacson, E., Keller, H.B., *Analysis of Numerical Methods*, Dover, 1994.
- [42] Kantorovic, L.V., "On a New Method of Approximate Solution of Partial Differential Equations," *Dokl. Akad. Nauk, SSSR* 4, pp. 532-536, 1934.
- [43] Keller, H.B., *Numerical Methods for Two-Point Boundary Value Problems*, Blainsdell, New York, 1968.
- [44] Kelly, L., *Handbook of Numerical Methods and Applications*, Addison-Wesley, 1967.
- [45] Kirk, D.E., *Optimal Control Theory: An Introduction*, Prentice-Hall, 1970.

- [46] Kraft, D., "On Converting Optimal Control Problems into Nonlinear Programming Codes," NATO ASI Series, Vol. F15, *Computational Mathematical Programming*, ed. K. Schittkowski, Springer, 1985, pp. 261 - 280.
- [47] Lanczos, C., "Trigonometric Interpolation of Empirical and Analytical Functions," *J. Math. Phys.*, 103, pp. 123 - 199, 1938.
- [48] Lewis, F., Syrmos, V., *Optimal Control*, John Wiley & Sons, INC., 1995.
- [49] Matrell, C.A., 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.
- [50] Moyer, H.G., Pinkham, G., "Several Trajectory Optimization Techniques, Part II: Applications," *Computing Methods in Optimization Problems*, A.V. Balakrishnan, L.W. Neustadt, eds., Academic Press, 1964, pp. 91 - 105.
- [51] Oberle, H.J., Grimm, "BOUNDSCO - A Program for Numerical Solution of Optimal Control Problems," English translation of DFVLR - mitt. 85-05, ICAM - Virginia Polytechnic Inst. and State Univ., Blacksberg, VA, May 1989.
- [52] Orszag, S.A., "Comparison of Pseudospectral and Spectral Approximations," *Stud. Appl. Math.*, 51, 1972, pp. 689-703.
- [53] Pontryagin, L.S., Boltyanskii, V., Gamkrelidze, R., Mischenko, E., *The Mathematical Theory of Optimal Processes*. New York: Interscience, 1962.
- [54] Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., "Two Point Boundary Value Problems," *Numerical Recipes*, Cambridge Univ. Press, New York, 1986.
- [55] Ross, I.M., 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.

- [56] Ross, I.M., Fahroo, F., "A Perspective on Methods for Trajectory Optimization," *Proceedings of the AIAA/AAS Astrodynamics Conference*, Monterey, CA., August 2002.
- [57] Ross, I.M., Fahroo, F., "A Direct Method for Solving Nonsmooth Optimal Control Problems," *Proceedings of the 2002 IFAC World Conference*, Barcelona, Spain, July 2002.
- [58] Russell, R.D., Shampine, L.F., "A Collocation Method for Boundary Value Problems," *Numerical Mathematics* , Vol. 19, No. 1, 1972, pp. 1-28.
- [59] Schwartz, A., "Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems," PhD Thesis, U.C. Berkeley, 1996.
- [60] Seywald, H., Kumar, R.R., "Method for Automatic Costate Calculation," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1252-1261.
- [61] Shevell, Richard S., *Fundamentals of Flight* , Prentice Hall, 1989.
- [62] Slater, J.C., "Electronic Energy Bands in Metal," *Phys., Rev.* 45, pp. 794 - 801, 1934.
- [63] Stoer, J., Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag, 1980.
- [64] Stryk, O., Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, 37 pp. 357-373 1992.
- [65] 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.
- [66] Teo, K.L., Goh C.J., Wong, K.H., *A Unified Computational Approach to Optimal Control Problems* , Longman Scientific & Technical, 1991.

- [67] Thorvaldsen, T., Proulx, R., Ross, I.M., "A Fast Accurate Method for Low-Thrust Trajectory Optimization," *Proceedings of the 16th Space Flight Dynamics Conference*, Pasadena, CA, December 2001.
- [68] Trefethen, L.N., *Spectral Methods in MATLAB*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [69] Williamson, W.E., "Use of Polynomial Approximations to Calculate Suboptimal Controls," *AIAA J.* 9, 1971, 2271 - 2273.
- [70] Wright, K., "Chebyshev Collocation Methods for Ordinary Differential Equations," *Computer J.*, 6, pp. 358-365, 1964.
- [71] Vallado, David A., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, 1997.
- [72] Vlassenbroeck, J., Van Doreen, R., "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE Trans. Automat. Cont.*, Vol. 33, No. 4, 1988, pp. 333-340.
- [73] Vlassenbroeck, J., "A Chebyshev Polynomial Method for Optimal Control with State Constraints," *Automatica*, Vol. 24, 1988, pp. 499-506.

