ABSTRACT

ENGELSONE, ANNA. Direct Transcription Methods in Optimal Control: Theory and Practice. (Under the direction of Stephen Campbell)

In optimal control, as in many other disciplines, individuals developing the theory and those applying it to real life problems do not always see eye to eye. Some results developed by theoreticians have very limited practical value, while other useful results may be unknown to practitioners or incorrectly interpreted. This work aims to bridge the gap between these two groups by presenting theoretical results in a way that will be useful to practitioners. We concentrate specifically on convergence results relating to a class of methods known as direct transcription, where the entire optimal control problem is discretized, in our case using a Runge-Kutta method, to form a nonlinear program.

For unconstrained problems, we present several convergence results, then give an original result that demonstrates that practically designed optimal control software will be unable to attain theoretically possible convergence order in most cases. We present a practical solution to this problem that is currently being implemented in an industrial software package.

In the next chapter, we also prove that many equality constrained problems, including problems unsolvable by other methods, are, for a direct transcription method, equivalent to unconstrained problems, so that convergence results from the previous chapter apply. We provide practical guidelines for regularizing a constrained problem to ensure accurate solution by a direct transcription method.

For inequality constrained problems, we give a detailed overview of different sets of necessary conditions and existing convergence results. We also present a phenomenon we call ”virtual boundary arcs”, demonstrating the advantage of direct transcription for another class of problems, in this case problems for which a boundary arc is theoretically impossible but the cost structure forces the solution very close to the constraint boundary.

DIRECT TRANSCRIPTION METHODS IN OPTIMAL CONTROL: THEORY AND PRACTICE

BY

ANNA ENGELSONE

A DISSERTATION SUBMITTED TO THE GRADUATE FACULTY OF
NORTH CAROLINA STATE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

OPERATIONS RESEARCH

RALEIGH,NORTH CAROLINA
MAY 2006

APPROVED BY:

DR. STEPHEN L. CAMPBELL
CHAIR OF ADVISORY COMMITTEE

DR. RALPH SMITH

DR. CARL T. KELLEY

DR. PIERRE A. GREMAUD

Biography

Anna Engelsone is a graduate of Riga School No. 40, in Riga, Latvia, the alma mater of many fine individuals scattered throughout the globe. In 2001, she graduated from Goshen College in Goshen, Indiana with an honors degree in Computer Science and Applied Mathematics. The title of Anna Engelsone's BA thesis is "Central Extensions of Partially Defined Games." She studied cooperative games with Dr. David Housman during the Maple Scholars research program in the summer of 1999 and graph coloring with Dr. Arie Bialostocki during the Research Experiences for Undergraduates program at the University of Idaho in the summer of 2000. Anna Engelsone entered North Carolina State University in 2001 and studied control theory with Dr. Joseph Dunn and Dr. Stephen Campbell. Under the guidance of Dr. Campbell, she has co-authored more than half a dozen papers and matured as a scientist. Anna Engelsone has presented her research at a number of national and international conferences. She is currently working for Mergeglobal, Inc. in Arlington, VA.

Acknowledgements

The research presented here was performed between 2002 and 2006 under the expert guidance of Dr. Stephen Campbell of NCSU and with frequent input from Dr. John Betts of the Boeing Company. Financial support was provided in part by NSF Grants DMS-0101802, DMS-020695, DMS-0404842, and ECS-0114095.

I would like to thank the committee members, Dr. Campbell, Dr. Gremaud, Dr. Kelley and Dr. Smith for taking time out of their busy schedules to read and evaluate this work.

A big thank you to all participants of the Numerical Analysis seminar, student and faculty alike, for their many valuable comments on my research and presentations over the past three years.

Many thanks to all the helpful staff at the Mathematics and Operations Research departments. Your contributions to the well-being and sanity of faculty, students and anyone else who comes to you for help cannot be overestimated.

And, last but not least, thank you Liz, my inspiration, motivation, and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Definitions

1.1 Example of a Control Problem

Optimal control is a discipline that studies the control of dynamic systems, i.e. systems described by differential or difference equations, with the goal of optimizing a certain objective function.

For a small academic example, consider a trolley that travels along a track. Tied to the bottom of the trolley is a rope with a weight at the bottom. The rope can be rolled up and down. We assume a one-dimensional track and a two-dimensional world in which the rope swings back and forth, subject to forces of tension and gravity. (Figure 1.1.) Let us assume that we have direct control over the acceleration of the trolley and the acceleration of the pulley rolling up the rope as the trolley travels from left end of the track to the right end.

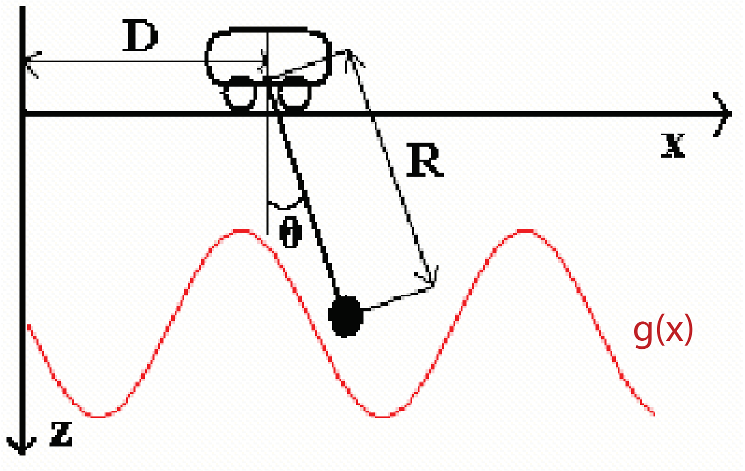


Figure 1.1: The Trolley Control System.

Mathematically, we can describe this dynamic system by

$$\begin{aligned} x''(t) &= -T(t)\sin\theta(t) \\ z''(t) &= -T(t)\cos\theta(t) \\ D''(t) &= a_1(t) \\ R''(t) &= a_2(t) \\ x(t) &= R(t)\sin\theta(t) + D(t) \\ z(t) &= R(t)\cos\theta(t) \end{aligned}$$

where (x, z) are the coordinates of the load, the mass of the load is assumed to be 1, the mass of the rope is assumed to be negligible and θ is the angle between the rope and the vertical axis. T is the tension of the rope. D and R are the position of the trolley and the length of the rope, respectively, and a_1, a_2 are the respective accelerations of the trolley and the rope.

Suppose that the trolley driver had direct control over the acceleration of the trolley and acceleration of the rope, through the gas pedal and the crank, for example. Then a_1 and a_2 are the **controls** of the system.

Suppose that in finite time τ we wanted to get the trolley as close as possible to the end of the track (point $(1,0)$) while keeping the swinging load above the terrain described by function $g(x)$ but no less than 2 meters away from the trolley. Mathematically, this means the addition of two constraints

$$\begin{aligned} z(t) &\leq g(x(t)) \\ R(t) &\geq 2 \end{aligned}$$

and the objective: minimize $(1 - D(\tau))^2$.

Common sense tells us that even if we are unaware of strict bounds on the maximum allowable acceleration, we should "regularize" the system to disallow infinite acceleration. One way to do this is to modify the objective function, putting small weights on the accelerations of trolley and rope along the entire time interval. Thus the new objective is

$$\min(1 - D(\tau))^2 + \delta \int_0^\tau a_1(t)^2 + a_2(t)^2 dt$$

where δ is small and positive.

Theoretical control theory results support this common sense logic, requiring the objective function to be positive definite with respect to all controls (see, for example, Table 3.3-1 in [28].)

1.2 Standard Form and Alternative Formulations

Once additional differential equations have been added to replace the 2nd derivatives on the left with 1st derivatives, the trolley problem takes on the form shared by many control problems which appear in applications, namely the form

$$
\begin{aligned}
\min \quad & \phi(x(t_f)) + \int_{t_0}^{t_f} L(x,y,t)dt && \text{(1.1a)}\\
x' &= f(x,y,t) && \text{(1.1b)}\\
0 &= g_1(x,y,t) && \text{(1.1c)}\\
0 &\leq g_2(x,y,t) && \text{(1.1d)}\\
x(t_0) &= \zeta && \text{(1.1e)}\\
0 &= \psi(x(t_f)) && \text{(1.1f)}
\end{aligned}
$$

where $x(t) : \mathbb{R} \rightarrow \mathbb{R}^{m_1}$ are **states** or **differential variables**, and $y(t) : \mathbb{R} \rightarrow \mathbb{R}^{m_2}$ are **algebraic variables**. The functions $\phi : \mathbb{R}^{m_1} \rightarrow \mathbb{R}$ and $L : \mathbb{R}^{m_1+m_2+1} \rightarrow \mathbb{R}$ determine what is alternately called the **cost function**, the objective function or the performance index. The differential constraints are determined by $f : \mathbb{R}^{m_1+m_2+1} \longrightarrow \mathbb{R}^{m_1}$, the algebraic **equality constraints** by $g_1 : \mathbb{R}^{m_1+m_2+1} \longrightarrow \mathbb{R}^{c_1}$ and the algebraic **inequality constraints** by $g_2 : \mathbb{R}^{m_1+m_2+1} \longrightarrow \mathbb{R}^{c_2}$, and boundary conditions determined by $\zeta \in \mathbb{R}^{m_1}, \psi : \mathbb{R}^{m_1} \longrightarrow c_3$.

For a problem designed to model a real-life process, the algebraic variables will include the **controls**, such as the acceleration of the trolley and the crank that winds the rope. They will also include algebraic variables such as the tension in the rope that are not controlled by the trolley driver. But from a mathematical standpoint, a control is any subset of y that determines the solution completely for a particular set of initial conditions.

Notice that a problem with a cost function of the form (1.1a) can be converted into a problem with the cost function $C(x(t_f))$, i.e. a problem in the so-called Mayer form, by letting $C = \phi + x_2$ where x_2 is another state defined by

$$
\begin{aligned}
x_2' &= L(x,u,t)\\
x_2(0) &= 0
\end{aligned}
$$

Similarly, a problem in which L, f, g_1, g_2 depend explicitly on t can be converted into a problem whose functions depend only on x and y by letting t be another state, x_3, defined by

$$\begin{aligned} x_3' &= 1 \\ x_3(0) &= 0 \end{aligned}$$

So the problem (1.1a) is equivalent to a problem of the form

$$\min \quad C(x(t_f)) \tag{1.2a}$$

$$x' = f(x, y) \tag{1.2b}$$

$$0 = g_1(x, y) \tag{1.2c}$$

$$0 \leq g_2(x, y) \tag{1.2d}$$

$$x(t_0) = \zeta \tag{1.2e}$$

$$0 = \psi(x(t_f)) \tag{1.2f}$$

Finally, notice that a problem of the form (1.1a) or (1.2) is equivalent to the same problem but with $t_0 = 0$ and $t_f = 1$, simply by letting $\bar{t} = t/(t_f - t_0)$. The results presented in the main chapters of this work were proved for problems in different forms. To make them easier to read and compare, we have rewritten some of them in a different form so that most results now refer to problems of the form 1.2.

1.3 Optimality Conditions

How does one find functions x and y that minimize the cost (1.2a) and also satisfy the differential and algebraic equations (1.2b) – (1.2f)? As with discrete minimization/maximization problems, there are additional equations and inequalities that the optimal solution must satisfy. Combined with (1.2b) – (1.2f), they form a system of equations that, under certain additional conditions, will uniquely determine a local minimum.

When minimizing a function of one variable with no constraints, a critical point occurs where the derivative of the function is zero. In the presence of constraints, the constraints are adjoined to the cost function with the aid of an **adjoint variable** to form what is sometimes called the **augmented performance index** which is then differentiated with respect to all variables. For continuous control problems, the first derivative is replaced by the first **variation** and the resulting equations are called the **Euler-Lagrange Equations** also known as **first-order optimality conditions** or first-order necessary conditions (for optimality).

For constrained problems, the derivation of the first-order optimality conditions is complicated (see Section 4.1) but for unconstrained problems of the form (1.2) the augmented performance index has the form

$$J = C(x(t_f)) + \nu^T \psi(x(t_f)) + \int_{t_0}^{t_f} \lambda^T (f(x,y) - x')dt$$

and the first-order optimality conditions take the form

$$\begin{aligned} x' &= \nabla_\lambda H && (1.3a) \\ \lambda' &= -\nabla_x H && (1.3b) \\ 0 &= \nabla_u H && (1.3c) \\ x(t_0) &= \zeta && (1.3d) \\ \lambda(t_f) &= \nabla_x C(x(t_f)) + \nu^T \nabla_x \psi(x(t_f)) && (1.3e) \end{aligned}$$

where $H(x,u,\lambda) = \lambda^T f(x,u)$ is the **Hamiltonian**. See [28], Section 3.2 for a detailed derivation.

Just as it is when we minimize a function, a critical point is not necessarily a minimum, and in addition to first order optimality conditions there are 2nd order conditions to consider. When minimizing a function of several variables, we are interested in whether its Jacobian is positive-definite. For optimal control problems, this is replaced with conditions such as coercivity (Definition 2.2) written in terms of the solution to a linearized problem or a Ricatti equation. For more information on necessary and sufficient conditions for optimality for different types of problems, see [11].

Suppose that the problem in (1.3) is actually a problem of the form

$$\begin{aligned} \min \quad & \phi(x(t_f)) + \int_{t_0}^{t_f} L(x, y, t)dt && \text{(1.4a)} \\ x' &= f(x, y, t) && \text{(1.4b)} \\ x(t_0) &= \zeta && \text{(1.4c)} \\ 0 &= \psi(x(t_f)) && \text{(1.4d)} \end{aligned}$$

"in disguise". Then the Hamiltonian takes the form $\lambda_2 L + \lambda^T f$ and we calculate $\lambda_2(t_f) = \nabla_{x_2}\phi(x) + x_2 = 1$ and $\lambda_2' = -\nabla_{x_2} H = 0$ giving us $\lambda_2(t) = 1$ for all t. So for problems of the form (1.4), the first order optimality conditions can be applied without loss of generality to the Hamiltonian defined by $H = L + \lambda^T f$.

1.4 Methods of Solution

Methods for solving optimal control problems can be divided into two basic categories: **direct** and **indirect** methods.

Indirect methods involve solving the boundary value problem (BVP) formed by the first-order optimality conditions. For most problems, this cannot be done analytically, so the equations are discretized in some fashion and solved numerically. For this reason, indirect methods are also sometimes referred to as **"optimize then discretize."** Currently the most popular methods for solving the BVP formed by the optimality conditions are so-called gradient algorithms. An overview of these and some of the other methods discussed here can be found in [31].

Direct methods approximate the original problem by a discrete optimization problem, an approach that is sometimes referred to as **"discretize then optimize."**

Some direct methods rely on techniques such as **shooting** or multiple shooting, where the equations (1.2b) – (1.2f) are solved for a particular control, usually assumed piecewise constant on a grid. Then the control is adjusted with the goal of making the cost smaller, and the whole process is repeated until a tolerance is met. The advantage of these methods is the relatively small size of the discretized problem, their major drawback is their stability. For a quick overview of shooting and multiple shooting methods see [3], Chapter 3.

In contrast, the class of methods we will call direct transcription methods discretize the entire problem on a grid, normally by using a collocation method based on a numerical integrator. For a problem in Mayer form (1.2) this amounts to discretizing the differential equations to obtain algebraic equations in the variables x_i, y_i, which represent the values of the states and the algebraic variables at the grid points t_i and, for higher order discretization methods, also variables χ_{ij}, y_{ij}, which represent the values of states and algebraic variables at the intermediate points. The algebraic constraints and boundary conditions, evaluated at the gridpoints, provide additional constraints for the discretized problem. The resulting problem is a large, sparse nonlinear program (NLP). For an overview of methods for solving large, sparse nonlinear programs that arise from optimal control problems, see [3], Chapters 1 and 2.

Direct transcription methods are well suited to problems where the functions f, g_i are "black boxes", since in these cases the formulation of optimality conditions may be difficult or impossible. However, for some inequality constrained problems they are less accurate than indirect methods ([7], Chapter 4). In the main chapters of this thesis we will present two classes of problems for which direct transcription outperforms other methods, inequality constrained problems that exhibit complicated behavior near the constraint boundary and certain equality constrained problems with high index constraints. But there is one thing that all numerical methods for solving optimal control problems have in common - the need to numerically integrate the differential equation (1.2b).

1.5 Convergence and Discretizations

The focus of this thesis is on the convergence properties of direct transcription methods. Most results given here evaluate convergence of a direct transcription method by the maximum difference between the optimal solution for the original problem and the optimal solution to the nonlinear program that is the discretization of the original problem for a particular discretization method and a particular grid. The grid can be uniform, with $N = 1/h$ evenly spaced nodes h units apart. It can be non-uniform, with the distances between nodes given by $h_i, i = 0, ..., N-1$. Because finer grids produce larger problems that take longer to solve, we are interested in the relationship between h, the maximum distance between gridpoints, and the error. We say that the error in variable z is order b if there exist $\bar{h}, c > 0$ such that

$$\max_k \|z_k - z^*(t_k)\| \leq ch^b$$

for all $h < \bar{h}$.

A variety of discretizations exist for approximating differential equations. (For an overview, see [22] and [23]). Many of these are **collocation methods**, i.e. methods that approximate a function by a piecewise-continuous function or spline. One type of collocation methods, using Radau points, was recently studied in connection with direct transcription by Kameswaran and Biegler (see [26] and [27]), and shown to be very useful on certain problems. For an overview of different types of discretizations used with direct transcription, see [31], Chapter 6.

Many of the discretizations used by practitioners for solving optimal control problems, both directly and indirectly, belong to the class of so-called classical Runge-Kutta methods. Some Runge-Kutta methods, like the trapezoid method, which approximates a function by linear splines, are also collocation methods. For our purposes, a Runge-Kutta method is any method characterized by its **Butcher array**, consisting of parameters $a \in \mathbb{R}^{s \times s}$ and $b, \sigma \in \mathbb{R}^s$. A Runge-Kutta method discretizes the differential equations (1.2b) as

$$\begin{aligned} x_{i+1} &= x_i + h_i \sum_{j=1}^{s} b_j f(\chi_{ij}, y_{ij}), \qquad i = 0, ..., N-1 \\ \chi_{ij} &= x_i + h_i \sum_{k=1}^{s} a_{jk} f(\chi_{ik}, y_{ik}), \qquad i = 0, ..., N-1, j = 1, ..., s. \end{aligned}$$

So, for the purpose of direct transcription methods, a nonlinear program based on the problem (1.2) obtained with a Runge-Kutta method characterized by (a, b, σ) has the form

$$\begin{aligned} \min \quad & C(x_N) && \text{(1.5a)} \\ x_{i+1} &= x_i + h_i \sum_{j=1}^{s} b_j f(\chi_{ij}, y_{ij}), \qquad i = 0, ..., N-1 && \text{(1.5b)} \\ \chi_{ij} &= x_i + h_i \sum_{k=1}^{s} a_{jk} f(\chi_{ik}, y_{ik}), \qquad i = 0, ..., N-1, j = 1, ..., s && \text{(1.5c)} \\ 0 &= g_1(x_i, y_i) && \text{(1.5d)} \\ 0 &\le g_2(x_i, y_i) && \text{(1.5e)} \\ x_0 &= \zeta && \text{(1.5f)} \\ 0 &= \psi(x_N) && \text{(1.5g)} \end{aligned}$$

where $t_{ij} = t_i + \sigma_j h_i$, $t_i = \sum_{l=0}^{i-1} h_l$, and $0 \le \sigma_1 \le ... \le \sigma_s \le 1$.

Many popular integrators, such as Euler's Method, Trapezoid Method, Hermite-Simpson or RK4, are Runge-Kutta methods. However, in practice, these methods are often implemented in a form different from (1.5), out of considerations ranging from time and storage to robustness.

Table 1.1: Order of Runge-Kutta discretization as an integrator.

Order	**Conditions** $c_i = \sum_{j=1}^{s} a_{ij}, \qquad d_j = \sum_{i=1}^{s} b_i a_{ij}$
1	$\sum_{i=1}^{s} b_i = 1$
2	$\sum_{i=1}^{s} d_i = 1/2$
3	$\sum_{i=1}^{s} c_i d_i = 1/6, \quad \sum_{i=1}^{s} c_i^2 b_i = 1/3,$
4	$\sum_{i=1}^{s} b_i c_i^3 = 1/4, \quad \sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i a_{ij} c_j = 1/8,$ $\sum_{i=1}^{s} d_i c_i^2 = 1/12, \quad \sum_{i=1}^{s} \sum_{j=1}^{s} d_i a_{ij} c_j = 1/24,$

Runge-Kutta methods are classified as explicit ($a_{ij} = 0$ whenever $j \geq i$) and implicit. They are also classified by their order as an integrator (see Table 1.1 , which is a copy of [21], Table 2). The order of a method as an integrator is the order of the error between the solution to the differential equation $x' = f(t, x, y)$ for a particular smooth function y and the solution to the discretization. As we will see in the following chapters, the order of the error between the optimal solution to the original problem and the optimal solution to the discretized problem is related to the order of the discretization as an integrator, but additional conditions and considerations are necessary.

1.6 Computation

The computational studies presented in this monograph were done using the sparse optimal control code SOCS (see `http://www.boeing.com/phantom/socs/`) developed at the Boeing company. SOCS is a collection of FORTRAN77 subroutines, suited for solving any optimal control problems with dynamics given by ordinary differential equations, including multiple-phase problems, and problems with right-hand sides of dynamics and constraints described by user-defined functions. For more information on SOCS , see [5].

SOCS allows the user to choose the discretization methods to be used as well as the initial grid. Unless told otherwise, SOCS will formulate the discretization on a coarse grid using a lower order method, find the optimal solution and then use various heuristics to refine the grid and/or switch to a higher order discretization method, reformulate the discretization, and repeat. To demonstrate convergence properties, specifically convergence order, we often ask SOCS instead to solve the problem on a particular grid using a particular discretization method and stop. For more information on mesh refinement and stopping criteria for same, see [3], Section 4.7 For a detailed description of the mesh refinement algorithm currently used in SOCS, see [8] and [7].

In the course of our work, we have produced a large number of FORTRAN77 codes that define particular problems and initialize SOCS with specific execution parameters. We have also written a number of Matlab (see `http://www.mathworks.com`) routines for manipulating and graphing SOCS output. Finally, we used Maple (http://www.maplesoft.com) for algebraic manipulations in a number of our theoretical results as well as to solve optimal control problems using a number of discretizations not implemented in SOCS. All these codes are available in electronic form at (http://www4.ncsu.edu/∼aengels/research/codes).

Chapter 2

Unconstrained Problems

2.1 Overview

In this chapter we will cover problems of the form

$$\min C(x(t_f)) \tag{2.1a}$$

$$\begin{aligned} x' &= f(x,y) && (2.1b) \\ x(t_0) &= \zeta. && (2.1c) \end{aligned}$$

Many results relating to convergence of direct transcription methods for optimal control problems belong to Hager, Dontchev and Veliov [12, 13, 14, 15, 16, 19, 20, 21]. We follow their lead in making the following two assumptions about the problem (2.1).

Definition 2.1. *The problem (2.1) is said to satisfy the* ***smoothness condition*** *if it has a local solution (x^*, u^*) which lies in $W^{2,\infty} \times W^{1,\infty}$, where $W^{k,p}$ is the Sobolev space consisting of vector-valued measurable functions $y : [t_0, t_f] \to \mathbb{R}^{m_1}$ whose jth derivative $y^{(j)}$ lies in L^p for all $j = 0, ..., k$ with the norm*

$$||y||_{W^{k,p}} = \sum_{j=0}^{k} ||x^{(j)}||_{L^p}. \tag{2.2}$$

Moreover, there exists an open set $\Omega \subset \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ and $\rho > 0$ such that

$$B_\rho(x^*(t), u^*(t)) \subset \Omega$$

for every $t \in [t_0, t_f]$, the first two derivatives of f are Lipschitz continuous in Ω, and the first two derivatives of C are Lipschitz continuous in $B_\rho(x^*(t_f))$.

Under the smoothness condition (Definition 2.1), we know that there exists a λ^* such that x^*, u^*, λ^* satisfy the optimality conditions (1.3) with $\psi = 0$.

Let

$$A(t) = \nabla_x f(x^*(t), y^*(t)), B(t) = \nabla_y f(x^*(t), y^*(t)), \tag{2.3a}$$

$$V = \nabla^2_{xx} C(x^*(t_f)), Q_1(t) = \nabla^2_{xx} H(x^*(t), y^*(t), \lambda^*(t)), \tag{2.3b}$$

$$Q_2(t) = \nabla^2_{xx} H(x^*(t), y^*(t), \lambda^*(t)), Q_3(t) = \nabla^2_{xx} H(x^*(t), y^*(t), \lambda^*(t)). \tag{2.3c}$$

Definition 2.2. *We say that the problem (2.1) satisfies the **coercivity condition** if for any (x, y) satisfying*

$$\begin{aligned} x' &= A(t)x + B(t)y \\ x(t_0) &= 0 \end{aligned}$$

there exists $\alpha > 0$ such that

$$x(t_f)^T V x(t_f) + \int_{t_0}^{t_f} x(t)^T Q_1 x(t) + 2x(t)^T Q_3 y(t) + y(t)^T Q_2 y(t) dt \geq \alpha \int_{t_0}^{t_f} y(t)^2 dt.$$

Coercivity is related to positivity of the Hessian of H which makes it a type of 2nd order optimality condition.

In [19], W. Hager proves second order convergence in y for certain explicit Runge-Kutta methods. Translated into standard Runge-Kutta form, his assumptions take the form:

D1 $a_{jk} = 0$ for $k \geq j$

D2 $\rho_j = \sum_{k=1}^{s} a_{jk}$ for $j = 1, ..., s$

D3 $b_j > 0$ for $j = 1, ..., s$

D4 $b_j = b_{s-j+1}$ for $j = 1, ..., s$

D5 $\frac{a_{pj}}{a_{s-j+1,s-p+1}} = \frac{b_j}{b_p}$ for $j = 1, ..., s-1; p = j+1, ..., s.$

The assumption (D1) means that the RK is explicit by definition and (D5) imposes no additional restrictions on a and b if $s < 3$. If $s = 3$ or 4, (D5) imposes only the restriction $\frac{a_{j1}}{a_{s,s-j+1}} = \frac{b_1}{b_j}$ for $j = 2, ..., s-1$.

Hager's result can be interpreted as follows:

Theorem 2.3. *If the optimal control problem (2.1) satisfies smoothness (Definition 2.1) and coercivity (Definition 2.2) and its RK discretization is at least 2nd order as an integrator (Table 1.1) and satisfies conditions D.1-D.5 and has a local optimal solution* (x, y)*, then*

$$\max_k \|x_k - x^*(t_k)\| + \max_k \|y_{k1} - y^*(t_k)\| \leq ch^2.$$

See Section 2.4.1 for a proof of how Theorem 2.3 follows from the result in [19]. Note that this result proves only second order convergence, even for methods of higher order.

In [14], Dontchev, Hager and Veliov prove second order convergence for 2nd order

Runge-Kutta methods that satisfy the conditions

$$\sum_{i\in N_l} b_i c_i = \sum_{i\in N_l} b_i\sigma_i \tag{2.4a}$$

$$\sum_{i=1}^{s}\sum_{j\in N_l} b_i a_{ij} = \sum_{i\in N_l} b_i(1-\sigma_i) \tag{2.4b}$$

$$\sum_{i\in N_l} b_i > 0 \tag{2.4c}$$

for all $l \in [1,s]$ where $N_i = \{j \in [1,s] : \sigma_j = \sigma_i\}$.

The result, reproduced here as Theorem 2.4, is also applicable to problems with generalized control constraints of the form $y \in U$.

Theorem 2.4. *(Adapted from [14]) If the optimal control problem (2.1) satisfies the smoothness and coercivity conditions (Definitions 2.1 and 2.2), and the Runge-Kutta scheme is 2nd order (see Table 1.1) and satisfies the conditions (2.4), then for all sufficiently small $h = \max h_k$, the discretization of (2.1) obtained according to this Runge-Kutta scheme has a strict local minimizer (x,y) and an associated adjoint variable λ such that, if $\frac{dy^*}{dt}$ has bounded variation,*

$$\max_{i=1,\dots,N,j=1,\dots,s} ||x_i - x^*(t_i)|| + ||\lambda_i - \lambda^*(t_i)|| + ||y_{i-1,j} - y^*(t_{i-1,j})|| \le ch^2$$

Of special importance is the result in [21], where Hager derives conditions on the parameters of a Runge-Kutta scheme (see Table 2.1, which is a copy of [21], Table 1) that guarantee high order convergence for the multipliers as well as the states. These conditions are much more general than either (D1)–(D5) or (2.4) and include many of the discretizations used by practitioners. They also guarantee convergence order higher than 2 for some higher order methods. Hager classifies a method that satisfies the additional conditions in Table 2.1 as "order κ for optimal control". The Table, which is copied directly from [21], contains conditions for orders up to 4. Later,

Table 2.1: Order of Runge-Kutta discretization for optimal control.

Order	**Conditions** $c_i = \sum_{j=1}^{s} a_{ij}, \quad d_j = \sum_{i=1}^{s} b_i a_{ij}$	
1	$\sum_{i=1}^{s} b_i = 1$	
2	$\sum_{i=1}^{s} d_i = 1/2$	
3	$\sum_{i=1}^{s} c_i d_i = 1/6, \quad \sum_{i=1}^{s} c_i^2 b_i = 1/3, \quad \sum_{i=1}^{s} d_i^2/b_i = 1/3$	
4	$\sum_{i=1}^{s} b_i c_i^3 = 1/4,$	$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i a_{ij} c_j = 1/8,$
	$\sum_{i=1}^{s} d_i c_i^2 = 1/12,$	$\sum_{i=1}^{s} \sum_{j=1}^{s} d_i a_{ij} c_j = 1/24,$
	$\sum_{i=1}^{s} c_i d_i^2/b_i = 1/12,$	$\sum_{i=1}^{s} d_i^3/b_i^2 = 1/4,$
	$\sum_{i=1}^{s} \sum_{j=1}^{s} b_i c_i a_{ij} d_j/b_j = 5/24,$	$\sum_{i=1}^{s} \sum_{j=1}^{s} d_i a_{ij} d_j/b_j = 1/8$

in [9], Bonnans and Laurent-Varin developed an algorithm that allowed them to derive "order for optimal control" conditions for orders up to 7. Hager's theorem can be stated in the following way:

Theorem 2.5. *(Adapted from [21], Theorem 2.1) If the optimal control problem (2.1) satisfies the smoothness and coercivity conditions (Definitions 2.1 and 2.2), and the Runge-Kutta scheme is of order κ for optimal control (Table 2.1) with $b_i > 0$ for each i, then for all sufficiently small $h = \max h_k$, the discretization of (2.1) obtained according to this Runge-Kutta scheme has a strict local minimizer (x, y) and* *an associated adjoint variable λ such that, if $\frac{dy^*}{dt}$ has bounded variation,*

$$\begin{aligned}
\max_{k=0,\ldots,N} ||x_k - x^*(t_k)|| &\leq ch^\kappa \\
\max_{k=1,\ldots,N} ||y(x_k, \lambda_k) - y^*(t_k)|| &\leq ch^\kappa \\
\max_{k=1,\ldots,N} ||\lambda_k - \lambda^*(t_k)|| &\leq ch^\kappa
\end{aligned}$$

where $y(x_k, \lambda_k)$ is a local minimizer of H corresponding to $\lambda = \lambda_k, x = x_k$.

Theorem 2.5 is the only theoretical result we have come across that proves higher order convergence for a large class of Runge-Kutta methods which includes the methods implemented in many software packages including SOCS, such as the Trapezoid Method, which has the Butcher array representation

$$a = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \end{pmatrix}, b = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \sigma = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and the Hermite-Simpson Method, which can be represented by

$$a = \begin{pmatrix} 0 & 0 & 0 \\ 5/24 & 1/3 & -1/24 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}, b = \begin{pmatrix} 1/6 \\ 2/3 \\ 1/6 \end{pmatrix}, \sigma = \begin{pmatrix} 0 \\ 1/2 \\ 1 \end{pmatrix}.$$

According to Table 2.1, this makes Trapezoid Method order 2 for optimal control and Hermite-Simpson method order 4.

However, notice that Theorem 2.5 only proves high order convergence of states and multipliers. Control convergence is notoriously harder to prove.

Theorems 2.3 and 2.4 establish second order convergence for the control under restrictive conditions that neither Trapezoid nor Hermite-Simpson satisfy. And both [21] and [14] give numerical results demonstrating that for many methods and even for very simple linear problems, the controls are often found to a much lower accuracy than the states and multipliers.

Theorem 2.5 provides a way of post-calculating the controls using the states and multipliers and guarantees that the resulting values are accurate to the same order. This post-calculation procedure would not be hard to implement in SOCS. All one would have to do is use an existing code that implements an unconstrained minimization algorithm such as some variation of Newton's method to find y that minimizes the function $H = \lambda_k^T f(x_k, y)$ and repeat for every k. For problems of reasonable size that satisfy Hager's smoothness and coercivity assumptions this should be both simple and fast.

However, if we tried to do it, we would encounter a big problem: the multiplier estimates produced by SOCS are not as accurate as the discrete multipliers in Theorem 2.5. In the next section, we will demonstrate that this is due to the fact that the theorem assumes a Butcher array implementation whereas SOCS implements more compact variations of popular Runge-Kutta methods such as Trapezoid and Hermite-Simpson. These implementations are mathematically equivalent, so the states and algebraic variables that solve the discretization are the same regardless of implementation, but the multipliers are in fact different.

In particular, for the Trapezoid method (TR) , we will show that, for uniform grids, the compressed implementation used in SOCS produces multipliers that are 2nd order accurate at midpoints, not gridpoints. We will also show that simple interpolation is sufficient to obtain 2nd order estimates of the multipliers at gridpoints. Finally, we will show that the control produced by the TR discretization regardless of implementation is 2nd order accurate at the inside gridpoints, that is, all gridpoints except t_0 and $t_N = t_f$. This is the first result to show 2nd order convergence in the control for TR, which had heretofore been believed to only produce 1st order convergence. In Section 2.3, we will present some preliminary results for the Hermite-Simpson method (HS).

2.2 Multiplier Convergence

2.2.1 New Theoretical Result

As noted above, TR is a Runge-Kutta method characterized by

$$\alpha = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \end{pmatrix}, b = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \sigma = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

This means that the TR discretization of problem (2.1) has the form

$$\begin{aligned} & \min C(x_N) && (2.5a) \\ x_{k+1} &= x_k + \frac{h}{2}(f(x_k, y_{k1}, t_{k1}) + f(\chi_k, y_{k2}, t_{k2})), k = 0, ..., N-1 && (2.5b) \\ \chi_{k1} &= x_k + \frac{h}{2}(f(x_k, y_{k1}, t_{k1}) + f(\chi_{k1}, y_{k2}, t_{k2})), k = 0, ..., N-1 && (2.5c) \\ x_0 &= \zeta, && (2.5d) \end{aligned}$$

and if the original problem satisfies the assumptions of Theorem 2.5 then (2.5) has a solution $w = (x, u, \lambda)$ which satisfies

$$\begin{aligned} \max_{k=0,\ldots,N} \|x_k - x^*(t_k)\| &\leq ch^2 \\ \max_{k=1,\ldots,N} \|\lambda_k - \lambda^*(t_k)\| &\leq ch^2. \end{aligned}$$

However, by subtracting equation (2.5c) from (2.5b) we get that $\chi_k = x_{k+1}$. Also, since $\sigma = [0, 1]$, $t_{k2} = t_{k+1}$ in (2.5b) and therefore $y_k^2 = y_{k+1}^1$, so that in practice TR is often simplified and implemented in the compressed form as

$$\min C(x_N) \quad (2.6a)$$

$$-x_{k+1} + x_k + \frac{h}{2}(f(x_k, y_k, t_k) + f(x_{k+1}, y_{k+1}, t_{k+1})) = \eta_k, k = 0, ..., N-1 \quad (2.6b)$$

$$x_0 = \zeta \tag{2.6c}$$

where η_k is a small tolerance. The η_k is there because the discretization is imposed as a constraint and thereby holds only up to a certain tolerance which is above machine precision.

It is clear that when $\eta_k = 0$ the two formulations are mathematically equivalent and therefore they must produce the same optimal values of x_k, y_k. However, if we formulate the optimality conditions for (2.5) and (2.6), we can see that the optimal multiplier variables are not related in any obvious ways and, in fact, take on different numerical values. To simplify the notation we set $t_0 = 0$, $t_f = 1$, in what follows.

THEOREM 2.6. *If the smoothness and coercivity conditions (Definitions 2.1 and 2.2) are satisfied and* $\frac{dy^*}{dt}$ *is of bounded variation and the problem (2.1) is discretized on a uniform grid* $h = h_i = 1/N$*, then for all sufficiently small* h *its compressed TR discretization (2.6) has a local optimal solution* (x, y, λ) *that satisfies*

$$\max_{k=0,\dots,N} \|x_k - x^*(t_k)\| \leq ch^2 \tag{2.7a}$$

$$\max_{k=1,\dots,N} \|\lambda_k - \lambda^*(t_k)\| \leq ch \tag{2.7b}$$

$$\max_{k=1,\dots,N} \|\lambda_k - \lambda^*\left(t_k - \frac{h}{2}\right)\| \leq ch^2 \tag{2.7c}$$

$$\max_{k=1,\dots,N-1} \left\|\frac{\lambda_{k+1} + \lambda_k}{2} - \lambda^*(t_k)\right\| \leq ch^2 \tag{2.7d}$$

$$\max_{k=1,\dots,N-1} \|y_k - y^*(t_k)\| \leq ch^2. \tag{2.7e}$$

This result, which is proved in Section 2.4.2, demonstrates that whereas the multipliers from the Butcher array formulation of TR are 2nd order accurate at gridpoints, the multipliers from the popular compressed implementation are 2nd order accurate at midpoints and only first order accurate at gridpoints. We have also proved that one can get second order estimates on the grid using averaging, or equivalently, interpolation. The new on the grid estimate for the adjoint is now being implemented in SOCS.

We have also shown that TR (regardless of implementation) gives second order estimates of the control on the inside gridpoints. Until now, second or higher order convergence in the control was only known to occur with certain restrictive classes of methods (Theorems 2.3 and 2.4), of which neither TR nor any other of the commonly used discretizations implemented in SOCS is a member.

The numerical results in the next section illustrate Theorem 2.6 on a particular example.

2.2.2 Numerical Example

EXAMPLE 2.1. *Consider the example problem from [21]:*

$$\min \int_0^1 y(t)^2 + x(t)y(t) + \frac{5}{4}x(t)^2 dt \tag{2.8a}$$

$$\begin{aligned} x'(t) &= 0.5x(t) + y(t) && (2.8b) \\ x(0) &= 1. && (2.8c) \end{aligned}$$

Note that this problem satisfies the coercivity condition (Definition 2.2) since the quadratic form inside the integral in (2.8a) is positive definite and the problem has one optimal solution given by

$$\begin{aligned} x^*(t) &= \frac{\cosh(1-t)}{\cosh(1)} && (2.9a) \\ y^*(t) &= -\frac{(\tanh(1-t)+0.5)\cosh(1-t)}{\cosh(1)} && (2.9b) \\ \lambda^*(t) &= \frac{2\cosh(1-t)\tanh(1-t)}{\cosh(1)}. && (2.9c) \end{aligned}$$

In order to examine the uncompressed formulation we set up a Maple worksheet that solves the necessary conditions to the uncompressed TR discretization of this problem analytically. In Table 2.2, we give the logarithm (base 2) of the max norm of the error in x, y, λ for $N = 10, 20, 40, 80$. As the number of gridpoints doubles, the logarithm of the error decreases by 2, so the error itself is a quarter of the previous grid error. Thus both x and λ errors are order 2 as proved by Hager.

However, when we solve the same problem using SOCS on the same uniform grids (by overriding the grid refinement algorithm and telling SOCS to only use TR), we see in Table 2.3 that even though the resulting x values are the same, the λ values are only first degree accurate just as equation (2.7b) of Theorem 2.6 states. In Table 2.4, we show that, as per equation (2.7c), these values are a second order approximation to the adjoint variables at the midpoints. We also show that, in accordance with equation (2.7d), simple linear interpolation is sufficient to obtain second order approximations to the adjoints on the inside gridpoints. Finally, in Table 2.5, we demonstrate that the controls produced by TR are second order accurate on the inside gridpoints, as stated in equation (2.7e). Figure 2.1 further illustrates this point by graphing $y^* - y$ for $N = 10, 20, 40, 80$.

In contrast, using SOCS with the discretization specified to be the classical 4th order Runge-Kutta method, given by

$$\alpha = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 1/6 \\ 1/3 \\ 1/3 \\ 1/6 \end{pmatrix}, \sigma = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \\ 1 \end{pmatrix}$$

we obtain results that conform exactly to Theorem 2.5. The state and the multiplier λ^1 are fourth order approximations to the states and adjoints at the gridpoint, u is third order, but u post-calculated from x and λ^1 in the way given in Theorem 2.5 is 4th order or nearly so (Table 2.6). Figure 2.2 shows the plot of the error in u (not post-calculated u). Note that it is similar to Figure 2.1 except for a vertical offset.

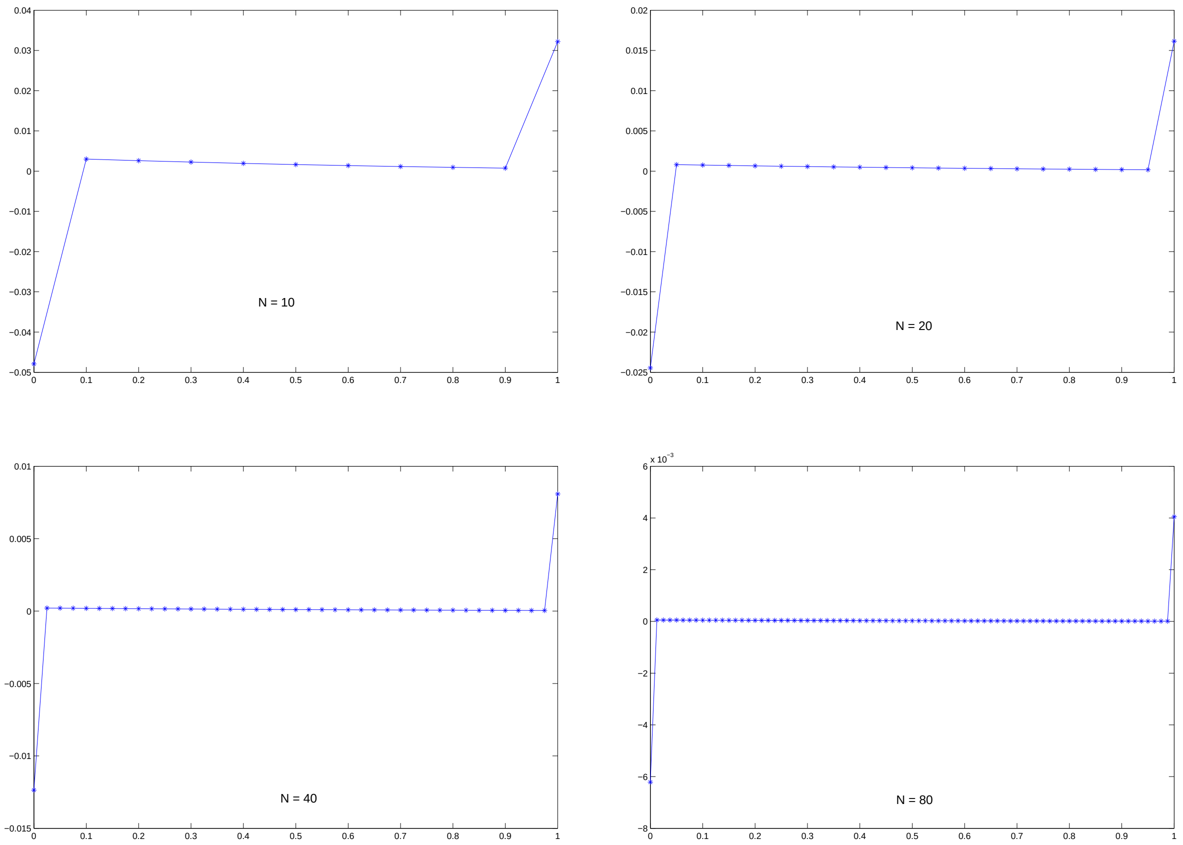


Figure 2.1: Example problem solved with TR. Graph of $y^* - y$ for $N = 10, 20, 40, 80$.

Table 2.2: $-\log_2$ of uncompressed TR error to gridpoint values of x, y, λ.

N	$\max_{i=0\ldots N}\|x_i - x^*(t_i)\|$	$\max_{i=1\ldots N}\|\lambda_i^1 - \lambda^*(t_i)\|$
10	8.8010	9.4278
20	10.7249	11.1800
40	12.6850	13.0671
80	14.6646	15.0129
Order	2	2
N	$\max_{i=0\ldots N}\|y_{i1} - y^*(t_i)\|$	$\max_{i=1\ldots N}\|y(\lambda_i^1, x_i) - y^*(t_i)\|$
10	4.3833	9.0806
20	5.3535	10.9346
40	6.3380	13.8634
80	7.3300	14.8282
Order	1	2

Table 2.3: $-\log_2$ of compressed TR error to gridpoint values of x, y, λ.

N	$\max_{i=0\ldots N}\|x_i - x^*(t_i)\|$	$\max_{i=0\ldots N}\|y_i - y^*(t_i)\|$	$\max_{i=1\ldots N}\|\lambda_i - \lambda^*(t_i)\|$
10	8.8010	4.3833	3.3677
20	10.7249	5.3535	4.3446
40	12.6850	6.3380	5.3332
80	14.6646	7.3300	6.3275
Order	2	1	1

Table 2.4: $-\log_2$ of compressed TR error in λ: at gridpoint, at midpoint, interpolated at gridpoint.

N	$\max_{i=1\ldots N}\|\lambda_i - \lambda^*(t_i)\|$	$\max_{i=1\ldots N}\|\lambda_i - \lambda^*\left(t_i - \frac{h}{2}\right)\|$	$\max_{i=1\ldots N-1}\|\frac{\lambda_i+\lambda_{i+1}}{2} - \lambda^*(t_i)\|$
10	3.3677	8.7633	8.0477
20	4.3446	10.6731	9.9185
40	5.3332	12.6283	11.8554
80	6.3275	14.6053	13.8243
Order	1	2	2

Table 2.5: $-\log_2$ of compressed TR error in y: inside gridpoints and endpoints.

N	$\max\limits_{i=1\ldots N-1} \|y_i - y^*(t_i)\|$	$\|y_0 - y^*(t_0)\|$	$\|y_N - y^*(t_f)\|$
10	8.3757	4.3833	4.9563
20	10.2660	5.3535	5.9522
40	12.2114	6.3380	6.9500
80	14.1841	7.3300	7.9489
Order	2	1	1

Table 2.6: $-\log_2$ of RK4 error to gridpoint values of x, y, λ.

N	$\max\limits_{i=0\ldots N} \|x_i - x^*(t_i)\|$	$\max\limits_{i=1\ldots N} \|\lambda_i^1 - \lambda^*(t_i)\|$
5	15.6440	15.8353
10	19.4282	19.7289
20	23.3203	23.6685
Order	4	4
N	$\max\limits_{i=0\ldots N} \|y_{i1} - y^*(t_i)\|$	$\max\limits_{i=1\ldots N} \|y(\lambda_i^1, x_i) - y^*(t_i)\|$
5	9.7356	17.5819
10	12.7113	21.2261
20	15.7039	24.4543
Order	3	3-4

The reasons for this offset are not clear.

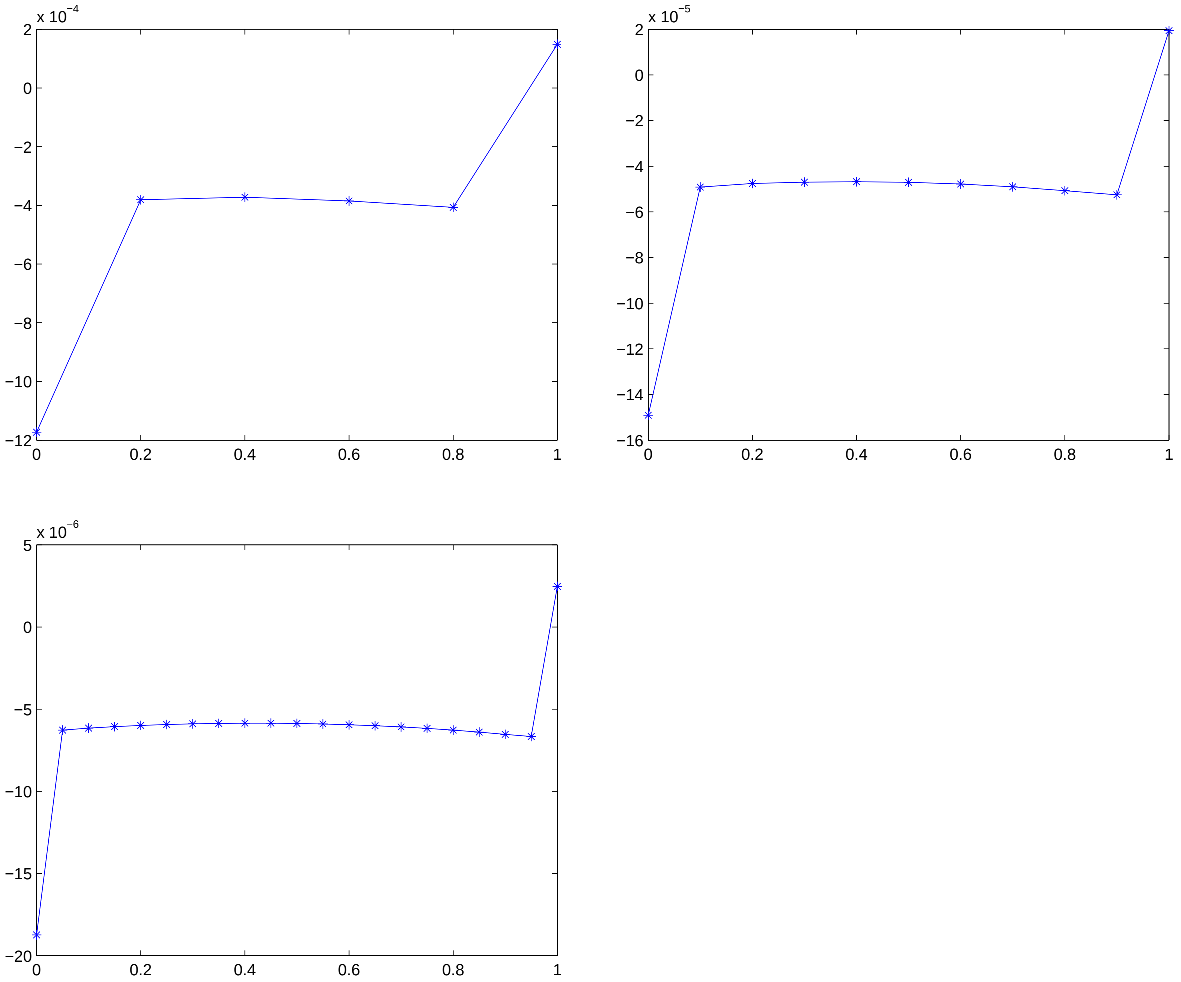


Figure 2.2: Example problem solved with RK4. Graph of $y^* - y$ for $N = 5, 10, 20$.

2.3 Open Questions

We have proved some practical results regarding multiplier and control convergence for Trapezoid Method. However, commercial direct transcription codes also use compressed versions of other discretizations that may be preferable to TR because they give higher order convergence in the states. The default option in SOCS , for example, is to discretize the problem on a coarse grid using TR, then after two mesh refinement iterations switch to a discretization called Hermite-Simpson method (HS).

Both TR and HS are collocation methods, but whereas the TR approximation to the right-hand side of a differential equation is piecewise linear, HS approximates a function with cubic splines. TR is 2nd order as integrator as well as 2nd order for optimal control, which means that, in its uncompressed form, it gives 2nd order convergence in states, multipliers and post-calculated controls. HS is 4th order as an integrator as well as 4th order for optimal control.

The Butcher-array formulation of HS is given by

$$\begin{aligned} x_{k+1} &= x_k + h(\frac{1}{6}f(x_k, y_{k1}, t_k) + \frac{2}{3}f(\chi_{k1}, y_{k2}, t_{k+\frac{1}{2}}) + \frac{1}{6}f(\chi_{k2}, y_{k3}, t_{k+1})) && (2.10) \\ \chi_{k1} &= x_k + h(\frac{5}{24}f(x_k, y_{k1}, t_k) + \frac{1}{3}f(\chi_{k1}, y_{k2}, t_{k+\frac{1}{2}}) - \frac{1}{24}f(\chi_{k2}, y_{k3}, t_{k+1})) && (2.11) \\ \chi_{k2} &= x_k + h(\frac{1}{6}f(x_k, y_{k1}, t_k) + \frac{2}{3}f(\chi_{k1}, y_{k2}, t_{k+\frac{1}{2}}) + \frac{1}{6}f(\chi_{k2}, y_{k3}, t_{k+1})). && (2.12) \end{aligned}$$

Two implementations of HS are available in SOCS. The default is HS-Compressed. The other one is called HS-Separated. HS-Separated is obtained from the Butcher array formulation by subtracting (2.12) from (2.10) to obtain $\chi_{k2} = x_{k+1}$, then solving (2.10) for $f(\chi_{k1}, y_k^2, t_{k+\frac{1}{2}})$ and substituting into (2.11). HS-Separated is implemented

Table 2.7: $-\log_2$ of HS-RK (Butcher array) error to gridpoint values of x, y, λ.

N	$\max\limits_{i=0\ldots N}\|x_i - x^*(t_i)\|$	$\max\limits_{i=0\ldots N}\|y_i^1 - y^*(t_i)\|$	$\max\limits_{i=0\ldots N-1}\|\lambda_i^1 - \lambda^*(t_i)\|$
5	17.6312	8.7187	16.8779
10	21.5477	10.6696	20.7249
20	25.5050	12.6455	24.6494
Order	2	1	1

in SOCS as

$$
\begin{aligned}
x_{k+1} - x_k - \frac{h}{6}(f(x_k, y_{k1}, t_k) + 4f(\chi_{k1}, y_{k2}, t_{k+\frac{1}{2}}) + f(x_{k+1}, y_{k+1,1}, t_{k+1})) &= \eta_1 \quad (2.13)\\
\chi_{k1} - x_k - \frac{1}{2}(x_k + x_{k+1}) - \frac{h}{8}(f(x_k, y_{k1}, t_k) + f(x_{k+1}, y_{k+1,1}, t_{k+1})) &= \eta_2 \quad (2.14)
\end{aligned}
$$

where η_1, η_2 are preset tolerances.

The most compact formulation, HS-Compressed, is obtained by solving (2.14) ($\eta_2 = 0$) for χ_{k1} and consists of only one equation

$$
x_{k+1} - x_k - \frac{h}{6}(f(x_k, y_{k1}, t_k) + 4f(\chi, y_{k2}, t_{k+\frac{1}{2}}) + f(x_{k+1}, y_{k+1,1}, t_{k+1})) = \eta \quad (2.15)
$$

where χ is defined exactly by

$$
\chi = x_k + \frac{1}{2}(x_k + x_{k+1}) + \frac{h}{8}(f(x_k, y_{k1}, t_k) + f(x_{k+1}, y_{k+1,1}, t_{k+1})).
$$

Note that HS-Compressed has only $N(m_1 + 2m_2)$ variables and Nm_1 constraints compared to $N(m_1 + m_2)$ and Nm_1 constraints for TR-Compressed but it offers 4th degree approximation of the state instead of only 2nd degree. It does not, however, offer a 4th order approximation of adjoints at the gridpoint in the same way that the Butcher array formulation ($N(3m_1 + 2m_2)$ variables, $3Nm_1$ constraints) does according to Theorem 2.5.

We implemented the Butcher array formulation of HS in Maple and applied it to Example 2.1. The states, controls and multipliers we obtained were 4th, 2nd and 4th

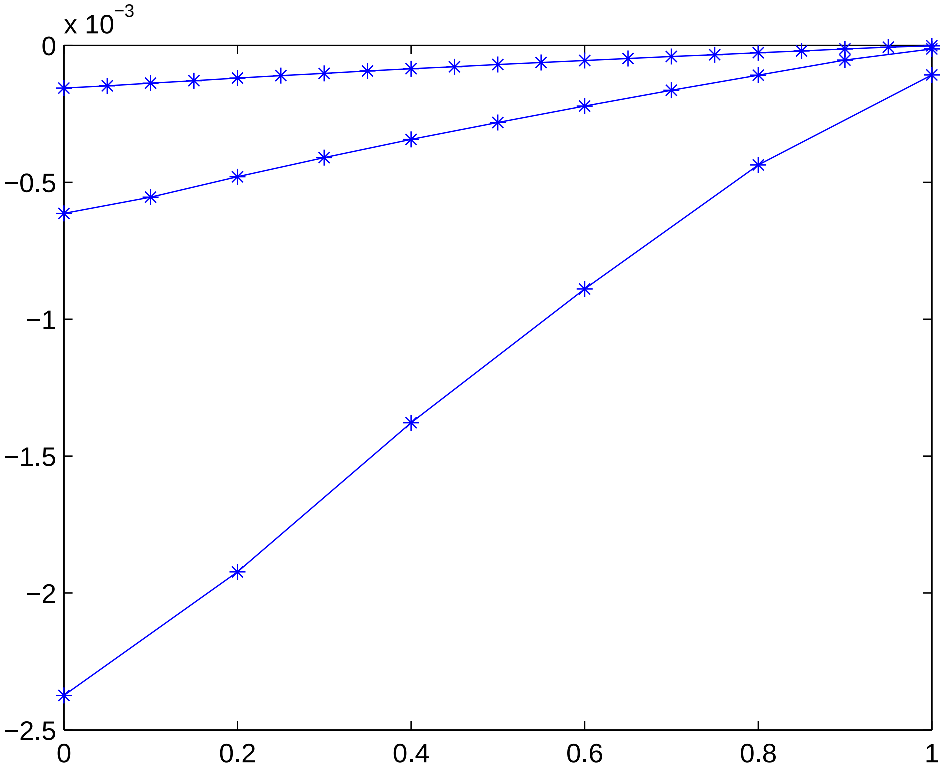


Figure 2.3: Example problem solved with HS. Graph of $y^* - y$ for $N = 5, 10, 20$.

Table 2.8: $-\log_2$ of HS-Compressed error to gridpoint values of x, y, λ.

N	$\max\limits_{i=0\ldots N} \|x_i - x^*(t_i)\|$	$\max\limits_{i=0\ldots N} \|y_i^1 - y^*(t_i)\|$	$\max\limits_{i=1\ldots N} \|\lambda_i - \lambda^*(t_i)\|$
5	17.6312	8.7187	2.4610
10	21.5459	10.6696	3.3934
20	25.4973	12.6455	4.3581
Order	4	2	1

Table 2.9: $-\log_2$ of HS-Compressed error in λ: at gridpoint, at midpoint, interpolated at gridpoint.

N	$\max_{i=1...N}\|\lambda_i - \lambda^*(t_i)\|$	$\max_{i=1...N}\|\lambda_i - \lambda^*\left(t_i - \frac{h}{2}\right)\|$	$\max_{i=1...N-1}\|\frac{\lambda_i+\lambda_{i+1}}{2} - \lambda^*(t_i)\|$
5	2.4610	8.8156	7.0228
10	3.3934	10.7175	8.8161
20	4.3581	12.6697	10.7177
Order	1	2	2

Table 2.10: $-\log_2$ of HS-Compressed error in y: inside gridpoints and endpoints.

N	$\max_{i=1...N-1}\|y_i^1 - y^*(t_i)\|$	$\|y_0^1 - y^*(t_0)\|$	$\|y_N^1 - y^*(t_f)\|$
5	9.0229	8.7187	13.1719
10	10.8162	10.6696	16.1703
20	12.7176	12.6455	19.0974
Order	2	2	3

order respectively (see Table 2.7). However, solving the same example with SOCS demonstrated (see Tables 2.8 – 2.10) that HS-Compressed can only offer a 2nd degree approximation to the adjoints at the midpoint, same as TR.

Unlike with TR, the control error on the inside gridpoints is not significantly smaller than at the ends (Table 2.10, Figure 2.3). However, we did notice the linear nature of the gridpoint errors in Figure 2.3. Notice in Table 2.10 that the endpoint error appears to be 3rd order. Taken together with Figure 2.3, this may indicate that HS-Compressed in fact gives a third order approximation to the control on some rescaled grid. As expected, the uncompressed formulation, which we implemented with Maple, gives multipliers that are 4th order accurate.

The slight discrepancy between the state values obtained with SOCS and Maple can be explained by the difference in the solution algorithms (SQP in SOCS, exact algebraic solution in Maple). The tables only give the logarithm of the errors, but the state errors themselves are on the order of $10^{-6}, 10^{-7}, 10^{-8}$ for $N = 5, 10, 20$ respectively, whereas the control errors are only on the order of 10^{-3} to 10^{-4}.

In summary, future work would include

- A theorem similar to Theorem 2.6 for HS-Compressed. We would like to establish 2nd order convergence at the midpoint, and the validity of simple interpolation to obtain 2nd order approximations at the gridpoint. Can adjoint approximations of order greater than 2 be obtained on the original grid or on some other grid, either directly from the discrete multipliers or though some kind of manipulation?
- Some insight into control error with HS. Can we prove 2nd order convergence overall for y_{k1} and perhaps higher order convergence on a subgrid or a shifted grid as we did with TR? Does y_{k2} converge to the either the gridpoint or the midpoint values of the control and to which order?
- Studying the differences and similarities between HS-Compressed and HS-Separated. Are the multipliers produced by HS-Separated and HS-Compressed always the same? What are the advantages of one formulation versus the other?

We have produced Maple worksheets and Matlab routines as well as organized data from various SOCS implementations of Example 2.1. This setup allows for easy data analysis and implementation of numerical experiments to test out any further conjectures relating to the above three questions. All that remains is theoretical work.

2.4 Proofs

2.4.1 Proof of Theorem 2.3

The paper [19], page 458 makes the following assumptions about a problem of the form (2.1):

A1 There exist an optimal control y^* and a corresponding trajectory x^*

A2 The differential equation $x' = f(x, y)$ can be integrated for all y in some neighborhood of y^*

A3 There exists an optimal solution χ^h, y^h to the discretization

A4 Both the discrete and the continuous optimality conditions hold

A5 The discretization is order b as an integrator, where $b \geq 2$.

A6 If x^1, x^2 are solutions to $x' = f(x, y)$ satisfying $x^1(s) = p^1, x^2(s) = p^2$ for some $s \in [0, 1]$ then for all $t \in [0, 1]$, $\|x^1(t) - x^2(t)\| = O(\|p^1 - p^2\|)$.

Now (A1), (A2), (A4) follow from smoothness and (A6) follows from smoothness and coercivity (see [16]), whereas (A3) and (A5) are part of the theorem statement.

The RK discretization of (2.1) has the form

$$\min \quad C(x_N) \tag{2.16a}$$

$$x_{i+1} = x_i + h_i \sum_{j=1}^{s} b_j f(\chi_{ij}, y_{ij}), \qquad i = 0, ..., N-1 \tag{2.16b}$$

$$\chi_{ij} = x_i + h_i \sum_{k=1}^{s} a_{jk} f(\chi_{ik}, y_{ik}), \qquad i = 0, ..., N-1, j = 1, ..., s \tag{2.16c}$$

$$x_0 = \zeta. \tag{2.16d}$$

Let $\alpha_{jp} = a_{j+1,p+1}$ for $j = 1, ..., s-1; p = 0, ..., s-2$ and $\alpha_{sp} = b_{p+1}$ for $p = 0, ..., s-1$. Using (D1), we can rewrite (2.16) as

$$\begin{aligned} \min \quad & C(\bar{x}_{N0}) \\ \bar{x}_{ij} &= \bar{x}_{i0} + h_i \sum_{p=0}^{j-1} \alpha_{jp} f(\bar{x}_{ik}, \bar{y}_{ik}), \qquad i = 0, ..., N-1, j = 1, ..., s \\ \bar{x}_{i+1,0} &= \bar{x}_{i,s}, \qquad i = 0, ..., N-1 \\ \bar{x}_{00} &= (0, 0, \zeta) \end{aligned}$$

where $\bar{x}_{i0} = x_i$ for $i = 0, ..., N$, $\bar{x}_{ij} = \chi_{i,j+1}$ for $j = 1, ..., s-1$, $\bar{y}_{ij} = y_{i,j+1}$ for $j = 0, ..., s-1$.

Moreover, we can establish that the following conditions are satisfied

P1 ([19],p.458) $\bar{x}_{ij}, \bar{y}_{ij}$ approximate the state and the control at time $t_i + \rho_j h_i$ where $0 = \rho_0 \leq ... \leq \rho_s = 1$. This follows from $0 \leq \sigma_1 \leq ... \leq \sigma_s \leq 1$, (D2), (D4) and the relationship between $(\bar{x}_{ij}, \bar{y}_{ij})$ and x_i, χ_{ij}, y_{ij}.

P2 ([19],p.460) $\alpha_{sj} \neq 0$ for $j = 0, ..., s-1$, which follows from (D3).

P3 ([19], Eq.39) $\alpha_{sj} = \alpha_{s,s-j-1}$ for $j = 0, ..., s-1$, which follows from (D5).

P4 ([19], Eq.38) $\alpha_{pj}\alpha_{sp} = \alpha_{sj}\alpha_{s-j-1,s-p-1}$ for $j = 0, ..., s-2; p = j+1, ..., s-1$, which follows from D.6.

The assumption in [19], Theorem 3.1 that $h\left(\nabla^2_{yy} H\right)^{-1}$ is bounded in y and h follows directly from coercivity [21]. So the bound on the error in y follows from Lemma 3.1 and Theorem 3.1 in [19] and the bound on the error in x follows from the discretization being 2nd order as an integrator.

2.4.2 Proof of Theorem 2.6

We will utilize an abstract result from [21], reproduced here as Lemma 2.1.

LEMMA 2.1. *([21], Proposition 5.1) Let X be a Banach space and let Y be a linear normed space with the norms in both spaces denoted $\|\cdot\|$. Let $F : X \to 2^Y$ be a set-valued map, let $L : X \to Y$ be a bounded linear operator, and let $T : X \to Y$ be continuously Frechét differentiable in $B_r(w^*)$ for some $w^* \in X$ and $r > 0$. Suppose that the following conditions hold for some $\delta \in Y$ and scalars ϵ, γ, and $\tau > 0$:*

Q1 *$T(w^*) + \delta \in F(w^*)$.*

Q2 *$\| \bigtriangledown T(w) - L\| \leq \epsilon$ for all $w \in B_r(w^*)$.*

Q3 *The map $(F - L)^{-1}$ is single-valued and Lipschitz continuous in $B_\tau(\pi), \pi = (T - L)(w^*)$, with Lipschitz constant γ.*

If $\epsilon\gamma < 1, \epsilon r \leq \tau, \|\delta\| \leq (1 - \gamma\epsilon)r/\gamma$, then there exists a unique $w \in B_r(w^*)$ such that $T(w) \in F(w)$. Moreover, we have the estimate

$$\|w - w^*\| \leq \frac{\gamma}{1 - \gamma\epsilon}\|\delta\|.$$

Formulate the Hamiltonian for (2.6)

$$\bar{H}(x, y, \lambda) \quad = \quad C(x_N) + \sum_{k=0}^{N-1} \lambda_{k+1} \left(-x_{k+1} + x_k + \frac{h}{2} \left(f(x_k, y_k) + f(x_{k+1}, y_{k+1}) \right) \right),$$

and let

$$T(w) = \begin{pmatrix} T^1(w) \\ T^2(w) \\ T^3(w) \end{pmatrix} = \begin{pmatrix} \bar{H}_{\lambda_k}, & k = 1, ..., N \\ \bar{H}_{x_k}, & k = 1, ..., N \\ \bar{H}_{y_k}, & k = 0, ..., N \end{pmatrix}$$

where $w = (x, y, \lambda) = (x_0, ..., x_N, y_0, ..., y_N, \lambda_1, ..., \lambda_N)$.

We apply function space norms to x, y and λ by considering them as piecewise constant functions on $[t_0,\ t_f]$ with respect to the gridpoints t_k with value x_k (y_k, λ_k) on the kth subinterval.

Define a norm on the domain of T by

$$\|w\| = \|x\|_{L^\infty} + \|y\|_{L^2} + \|\lambda\|_{L^\infty} \tag{2.17}$$

and a norm on the range of T by

$$\|(p,q,r)\| = \|p\|_{L^1} + \|q\|_{L^1} + \|r\|_{L^2}. \tag{2.18}$$

Let A_k, B_k, Q_{i_k} denote A, B, Q_i evaluated at $t = t_k$ and let $L(w)$ be given by

$$-x_{k+1} + x_k + \frac{h}{2}\left(A_k x_k + B_k y_k + A_{k+1} x_{k+1} + B_{k+1} y_{k+1}\right), \quad k = 0, ..., N-1 \tag{2.19a}$$

$$h(Q_{1_k} x_k + Q_{3_k} y_k) + \left(I + \frac{h}{2}A_k^T\right)\lambda_{k+1} - \left(I - \frac{h}{2}A_k^T\right)\lambda_k, \quad k = 1, ..., N-1 \tag{2.19b}$$

$$\frac{h}{2}(2Vx_N + Q_{1_N} x_N + Q_{3_N} y_N) - \left(I - \frac{h}{2}A_N^T\right)\lambda_N \tag{2.19c}$$

$$\frac{h}{2}\left(Q_{2_0} y_0 + Q_{3_0}^T x_0 + B_0^T \lambda_1\right) \tag{2.19d}$$

$$h\left(Q_{2_k} y_k + Q_{3_k}^T x_k + \frac{1}{2}B_k^T(\lambda_{k+1} + \lambda_k)\right), \quad k = 1, ..., N-1 \tag{2.19e}$$

$$\frac{h}{2}\left(Q_{2_N} y_N + Q_{3_N}^T x_N + B_N^T \lambda_N\right). \tag{2.19f}$$

We will apply Lemma 2.1 (case $F = 0$) to two different values of $w^*, \hat{w}$ and $\check{w}$, where

$$\begin{aligned}
\hat{x} \;=\; \check{x} &= (x(t_0)^*, ..., x(t_N)^*), \\
\hat{\lambda} &= (\lambda^*(t_1), ..., \lambda^*(t_f)), \\
\check{\lambda} &= \left(\lambda^*\left(t_0 + \frac{h}{2}\right), ..., \lambda^*\left(t_{N-1} + \frac{h}{2}\right)\right).
\end{aligned}$$

and $\hat{y}, \check{y}$ are defined in Lemma 2.5.

Next, we have several small lemmas that will help us prove two useful theorems. Then, Theorem 2.7 will establish that (Q3) of Lemma 2.1 holds for L defined above for $F = 0$ near both $\hat{w}$ and $\check{w}$. In Theorem 2.8, we will calculate $T(\hat{w})$ and $T(\check{w})$.

Finally, we tie it all together in the proof of Theorem 2.6. The next Lemma is known but we include its proof to illustrate the role of h in the norms used.

LEMMA 2.2. *Given $N \geq 1, h = \frac{1}{N}$, then for any vector $z \in R^{m(N+1)}$,*

$$\|z\|_{L^1} \leq c_1 \|z\|_{L^2} \leq c_2 \|z\|_{L^\infty}.$$

Proof. Let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product and let $\bar{1}$ be a vector of $N+1$ ones. Using the Schwartz inequality,

$$\begin{aligned} \|z\|_{L^1} &= \sum_{i=1}^{N+1} h\|z_i\| = \langle \sqrt{h}\bar{1}, \sqrt{h}\|z_i\| \rangle \leq \sqrt{h}\|\bar{1}\|_2 \|\sqrt{h}z\|_2 \\ &= \sqrt{\frac{N+1}{N}} \|z\|_{L^2} \leq \sqrt{\frac{2N}{N}} \|z\|_{L^2} = \sqrt{2}\|z\|_{L^2}. \end{aligned}$$

For the second inequality,

$$\|z\|_{L^2} = \sqrt{\sum_{i=1}^{N+1} h\|z_i\|^2} \leq \sqrt{\sum_{i=1}^{N+1} h(\max_i \|z_i\|)^2} = \sqrt{\frac{N+1}{N}(\max_i \|z_i\|)^2} \leq \sqrt{2}\|z\|_{L^\infty}.$$

□

LEMMA 2.3. *For all sufficiently small h, the solution x to the system of equations*

$$x_{k+1} = x_k + \frac{h}{2}\left(A_k x_k + A_{k+1} x_{k+1}\right) + z_k, k = 0, ..., N-1 \tag{2.20a}$$

$$x_0 = \zeta, \tag{2.20b}$$

where z, ζ are given, can be described by $x = M_1(z) + M_2(\zeta)$ where M_1, M_2 are linear operators.

Proof. We rewrite (2.20) as $\bar{A}x = z + e_1\zeta$ where the block bidiagonal matrix

$$\bar{A} = \begin{pmatrix} I & 0 & \cdots & 0 \\ -I - \frac{h}{2}A_0 & I - \frac{h}{2}A_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ & & & 0 \\ 0 & \cdots & -I - \frac{h}{2}A_{N-1} & I - \frac{h}{2}A_N \end{pmatrix}$$

is invertible and $e_1 = (I, 0, ..., 0)^T$. Thus $M_1 = \bar{A}^{-1}, M_2 = \bar{A}^{-1}e_1$.

□

Lemma 2.4. *For all $(x, \lambda) \in B_\tau(x^*(t_k), \lambda^*(t_k))$, where τ is independent of h, there exists a unique y satisfying $H_y(x, y, \lambda) = 0$ and $\|y - y^*(t_k)\| \leq c(\|\lambda - \lambda^*(t_k)\| + \|x - x^*(t_k)\|)$.*

Proof. Coercivity implies that $H_{yy}(x^*(t), y^*(t), \lambda^*(t)) = Q_2(t)$ is invertible for all t and hence uniformly positive definite since we are working on a closed finite interval [21]. So the result follows from the Implicit Function Theorem.

□

Lemma 2.5. *There exist $\hat{y}, \check{y}$ satisfying*

$$\begin{aligned} \|y^*(t_k) - \hat{y}_k\| &\leq ch, \ k = 0, ..., N \\ \|y^*(t_k) - \check{y}_k\| &\leq ch^2, \ k = 1, ..., N-1 \\ \|y^*(t_k) - \check{y}_k\| &\leq ch, \ k = 0, N \end{aligned}$$

such that $\hat{y}, \check{y}$ solve $\bar{H}_{y_k}(x^*(t_k), y, \lambda)$ for $\lambda = \hat{\lambda}$ and $\lambda = \check{\lambda}$ respectively.

Proof. First, note that

$$\bar{H}_{y_k} = \begin{cases} hH_y(x_k, y_k, \frac{\lambda_{k+1}+\lambda_k}{2}), & k = 1, ..., N-1 \\ \frac{h}{2}H_y(x_0, y_0, \lambda_1), & k = 0 \\ \frac{h}{2}H_y(x_N, y_N, \lambda_N), & k = N. \end{cases}$$

Applying Lemma 2.4 with $x = x^*(t_k)$ and $\lambda = \frac{\hat{\lambda}_{k+1}+\hat{\lambda}_k}{2}$ we obtain, for $k = 1, ..., N-1$,

$$\begin{aligned} \|\hat{y}_k - y^*(t_k)\| &\leq \|\frac{\lambda^*(t_{k+1}) + \lambda^*(t_k)}{2} - \lambda^*(t_k)\| \\ &= \frac{\|\lambda^*(t_{k+1}) - \lambda^*(t_k)\|}{2} \leq ch. \end{aligned}$$

Similarly, if $k = N$ and $\lambda = \hat{\lambda}_N$ then

$$\|\hat{y}_k - y^*(t_k)\| \leq \|\lambda^*(t_N) - \lambda^*(t_N)\| = 0$$

and if $k = 0$,

$$\|\hat{y}_k - y^*(t_k)\| \leq \|\lambda^*(t_1) - \lambda^*(t_0)\| = ch.$$

This establishes Lemma 2.5 for $\hat{y}$.

For $\check{y}$, we proceed in the same way, noting that, for $k = 1, ..., N-1$,

$$\left\| \frac{\lambda^*\left(t_k + \frac{h}{2}\right) + \lambda^*\left(t_k - \frac{h}{2}\right)}{2} - \lambda^*(t_k) \right\| \leq ch^2,$$

but

$$\left\| \lambda^*\left(t_1 - \frac{h}{2}\right) - \lambda^*(t_1) \right\| \leq ch$$

and

$$\left\| \lambda^*\left(t_N - \frac{h}{2}\right) - \lambda^*(t_N) \right\| \leq ch.$$

□

Theorem 2.7. *In the norms given by equations (2.17) and (2.18), the function L^{-1} is Lipschitz continuous everywhere with Lipschitz constant $\gamma = \frac{c}{h}$.*

Proof. Consider the equation $L(w) - \pi = 0$ where $\pi = -(p, q, r) \in R^{3N+1}$. It can be easily shown that this equation forms a set of first order necessary conditions to the quadratic programming problem

$$\begin{aligned}\min_{x_k,y_k} \quad & \left(\sum_{k=1}^{N-1} L_k(x_k,y_k) + q_k^T x_k + r_k^T y_k\right)\\ & +\frac{1}{2}(hL_0(x_0,y_0)+2r_0y_0)\\ & +\frac{1}{2}(hL_N(x_N,y_N)+2hx_N^TVx_N+2q_N^Tx_N+2r_N^Ty_N) \qquad (2.21a)\end{aligned}$$

$$\begin{aligned} x_{k+1} &= x_k + \frac{h}{2}\left(A_kx_k+B_ky_k+A_{k+1}x_{k+1}+B_{k+1}y_{k+1}\right)+p_k \qquad (2.21b)\\ x_0 &= \zeta \qquad (2.21c)\end{aligned}$$

where $L_k(x_k,y_k)=\frac{1}{2}\left(x_k^TQ_{1_k}x_k+2x_k^TQ_{3_k}y_k+y_k^TQ_{2_k}y_k\right)$.

By Lemma 2.3, (2.21) can be written as [1]

$$\begin{aligned}\min_{y,x} \quad & \bar{L}(x,y)+\bar{q}^Tx+\bar{r}^Ty \qquad (2.22)\\ x &= M_1(P\bar{B}y+p)+M_2(\zeta) \qquad (2.23)\end{aligned}$$

where

$$P=\frac{h}{2}\begin{pmatrix} I & I & \ddots & \vdots\\ 0 & \ddots & \ddots & 0\\ \vdots & \cdots & I & I\end{pmatrix},$$

$$\begin{aligned}\bar{L} &= x^T\bar{Q}_1x+y^T\bar{Q}_2y+2x^T\bar{Q}_3y\\ &= \sum_{k=1}^{N-1}L_k(x_k,y_k)+\frac{1}{2}(L_0(x_0,y_0)+L_N(x_N,y_N)+x_N^TVx_N),\\ \bar{q} &= \frac{1}{h}(0,q_1,...,q_{N-1},2q_N),\\ \bar{r} &= \frac{1}{h}(2r_0,r_1,...,r_{N-1},2r_N),\\ \bar{B} &= (B_0,...,B_N).\end{aligned}$$

[1] The idea behind this transformation was borrowed from [20], page 1066.

Substituting (2.23) into (2.22) we obtain the unconstrained problem $\min_y C(y)$ where

$$\begin{aligned} C(y) &= ((y^T\bar{B}^TP^T + p^T)M_1^T + \zeta^T M_2^T)\bar{Q}_1(M_1(P\bar{B}y + p) + M_2(\zeta)) + y^T\bar{Q}_2 y \\ &\quad +2((y^T\bar{B}^TP^T + p^T)M_1^T + \zeta^T M_2^T)\bar{Q}_3 y + \bar{q}^T(M_1(P\bar{B}y + p) + M_2(\zeta)) + \bar{r}^T y \\ &= y^T\bar{Q}y + (\phi_1 + \phi_2\zeta)^T y + \phi_3(\zeta) + \phi_4 \end{aligned}$$

where

$$\begin{aligned} \bar{Q} &= \bar{B}^TP^TM_1^T\bar{Q}_1M_1P\bar{B} + \bar{Q}_2 + 2\bar{B}^TP^TM_1^T\bar{Q}_3 \\ \phi_1 &= 2(\bar{B}^TP^TM_1^T\bar{Q}_1 + \bar{Q}_3{}^T)M_1p + \bar{B}^TP^TM_1^T\bar{q} + \bar{r} \\ \phi_2 &= 2\bar{B}^TP^TM_1^T\bar{Q}_1M_2 + 2\bar{Q}_3{}^TM_2. \end{aligned}$$

Since $C(y) = y^T\bar{Q}y$ corresponds to the problem (2.21) with $p = q = r = \zeta = 0$, the coercivity condition on the original problem implies $y^T\bar{Q}y \geq \beta\|y\|_{L^2}$. So, by [15], Lemma 4, given y^1, y^2 corresponding to two different values $\bar{\phi}^1, \bar{\phi}^2$ of $\bar{\phi} = \phi_1 + \phi_2\zeta$, we have $\|y^1 - y^2\|_{L^2} \leq c\|\bar{\phi}^1 - \bar{\phi}^2\|_{L^2}$. If $\bar{\phi}^i = \bar{\phi}(p^i, q^i, r^i, \zeta)$, then we have

$$\|y^1 - y^2\|_{L^2} \leq c\|(\bar{B}^TP^TM_1^T\bar{Q}_1 + \bar{Q}_3{}^T)M_1(p^1 - p^2) + \bar{B}^TP^TM_1^T(\bar{q}^1 - \bar{q}^2) + (\bar{r}^1 - \bar{r}^2)\|_{L^2}.$$

To make the following discussion more readable, let $\ddot{z} \doteq z^1 - z^2$ where z can be p, q, r, x, y or λ. Note that $x = M_1\ddot{p}$ is the solution to

$$x_{k+1} = x_k + \frac{h}{2}\left(A_kx_k + A_{k+1}x_{k+1}\right) + \ddot{p}_k \tag{2.24a}$$

$$x_0 = 0. \tag{2.24b}$$

Thus we have

$$\begin{aligned}
x_{k+1} &= \left(I-\frac{h}{2}A_{k+1}\right)^{-1}\left(I+\frac{h}{2}A_k\right)x_k+\left(I-\frac{h}{2}A_{k+1}\right)^{-1}\ddot{p}_k\\
&= \left(\Pi_{i=k-1}^{k}(I-\frac{h}{2}A_{i+1})^{-1}(I+\frac{h}{2}A_i)\right)x_{k-1}\\
&\quad +(I-\frac{h}{2}A_{k+1})^{-1}(I+\frac{h}{2}A_k)(I-\frac{h}{2}A_k)^{-1}\ddot{p}_{k-1}+(I-\frac{h}{2}A_{k+1})^{-1}\ddot{p}_k\\
&= \left(\Pi_{i=0}^{k}(I-\frac{h}{2}A_{i+1}^T)^{-1}(I+\frac{h}{2}A_i^T)\right)x_0\\
&\quad +\sum_{i=0}^{k}\left(\Pi_{j=i}^{k}(I-\frac{h}{2}A_{j+1}^T)^{-1}(I+\frac{h}{2}A_j^T)\right)(I-\frac{h}{2}A_{i+1})^{-1}\ddot{p}_i\\
&= \sum_{i=0}^{k}\left(\Pi_{j=i}^{k}(I-\frac{h}{2}A_{j+1}^T)^{-1}(I+\frac{h}{2}A_j^T)\right)(I-\frac{h}{2}A_{i+1})^{-1}\ddot{p}_i.
\end{aligned}$$

Let $a=\max_{k=0,\dots,N}\|A_k\|/2$ and assume $h<1/a$, so that, for $j=0,...,N$,

$$\begin{aligned}
\|I+\frac{h}{2}A_j\| &\leq 1+ha\\
\|\left(I-\frac{h}{2}A_j\right)^{-1}\| &\leq (1-ha)^{-1}
\end{aligned}$$

We also use the fact that $(1+ha)^{1/h},(1-ha)^{-1/h}$ are both bounded from above by e^a which in turn is bounded by $\max_{t_0<t<t_f}e^{\|A(t)\|/2}$. And so we have, for $k=0,...,N-1$,

$$\begin{aligned}
\|x_{k+1}\| &\leq \sum_{i=0}^{k}\left(\Pi_{j=i}^{k}\|(I-\frac{h}{2}A_{j+1}^T)^{-1}\|\|I+\frac{h}{2}A_j^T\|\right)\|(I-\frac{h}{2}A_{i+1})^{-1}\|\|\ddot{p}_i\|\\
&\leq \sum_{i=0}^{N-1}\frac{(1+ha)^{N-i}}{(1-ha)^{N-i+1}}\|\ddot{p}_i\|\leq e^{2a}\sum_{i=0}^{N-1}\|\ddot{p}_i\|\\
&\leq \frac{c}{h}\|\ddot{p}\|_{L^1}.
\end{aligned}$$

Hence

$$\|x\|_{L^\infty}=\max_k\|x_k\|\leq\frac{c}{h}\|\ddot{p}\|_{L^1}$$

and therefore by Lemma 2.2,

$$\|M_1\ddot{p}\|_{L^2}=\|x\|_{L^2}\leq\frac{c}{h}\|\ddot{p}\|_{L^1}.$$

The second term can be evaluated as follows:

$$\|P^T M_1^T(\bar{q}^1-\bar{q}^2)\|_{L^2} \leq \|\frac{2}{h}P^T M_1^T\tbinom{0}{\ddot{q}}\|_{L^2} \leq \|\frac{2}{h}P^T\|_{L^2}\|M_1^T\tbinom{0}{\ddot{q}}\|_{L^2}.$$

Now

$$\begin{aligned}
\|\frac{2}{h}P^T\|_{L^2} &= \max_{\|z\|_{L^2}=1}\sqrt{h(z_1^2+(z_1+z_2)^2+...(z_{N-1}+z_N)^2+z_N^2)} \\
&= \max_{\|z\|_{L^2}=1}\|(0,z_1,...,z_N)+(z_1,...,z_N,0)\|_{L^2} \\
&\leq \max_{\|z\|_{L^2}=1} 2\|(z_0,z_1,...,z_N)\|_{L^2} \\
&= 2.
\end{aligned}$$

and $\mu = M_1^T\tbinom{0}{\ddot{q}}$ is the solution to

$$\mu_0 = \mu_1 + \frac{h}{2}A_0^T\mu_1 \tag{2.25a}$$

$$\mu_k = \mu_{k+1} + \frac{h}{2}A_k^T\left(\mu_k+\mu_{k+1}\right) + \dot{q}_k, k=1,...,N-1 \tag{2.25b}$$

$$\mu_N = \frac{h}{2}A_N^T\mu_N + 2\dot{q}_N. \tag{2.25c}$$

where $\dot{q}_k = \ddot{q}_k$ for $k = 1, ..., N-1$, $\dot{q}_N = 2\ddot{q}_N$. So we have, for $k = 1, ..., N-1$,

$$\begin{aligned}
\mu_k &= (I - \frac{h}{2}A_k^T)^{-1}(I + \frac{h}{2}A_k^T)\mu_{k+1} + (I - \frac{h}{2}A_k^T)^{-1}\dot{q}_k \\
&= \left(\Pi_{i=k}^{k+1}(I - \frac{h}{2}A_i^T)^{-1}(I + \frac{h}{2}A_i^T)\right)\mu_{k+2} \\
&\quad + (I - \frac{h}{2}A_k^T)^{-1}(I + \frac{h}{2}A_k^T)(I - \frac{h}{2}A_{k+1}^T)^{-1}\dot{q}_{k+1} + (I - \frac{h}{2}A_k^T)^{-1}\dot{q}_k \\
&= \left(\Pi_{i=k}^{N-1}(I - \frac{h}{2}A_i^T)^{-1}(I + \frac{h}{2}A_i^T)\right)\mu_N \\
&\quad + \sum_{i=k}^{N-1}\left(\Pi_{j=k}^{i-1}(I - \frac{h}{2}A_j^T)^{-1}(I + \frac{h}{2}A_j^T)\right)(I - \frac{h}{2}A_i^T)^{-1}\dot{q}_i \\
&= \left(\Pi_{i=k}^{N-1}(I - \frac{h}{2}A_i^T)^{-1}(I + \frac{h}{2}A_i^T)\right)(I - \frac{h}{2}A_N^T)^{-1}\dot{q}_N \\
&\quad + \sum_{i=k}^{N-1}\left(\Pi_{j=k}^{i-1}(I - \frac{h}{2}A_j^T)^{-1}(I + \frac{h}{2}A_j^T)\right)(I - \frac{h}{2}A_i^T)^{-1}\dot{q}_i \\
&= \sum_{i=k}^{N}\left(\Pi_{j=k}^{i-1}(I - \frac{h}{2}A_j^T)^{-1}(I + \frac{h}{2}A_j^T)\right)(I - \frac{h}{2}A_i^T)^{-1}\dot{q}_i.
\end{aligned}$$

Note that the last expression also describes μ_N.

We calculate an upper bound on $\|\mu_k\|$ similarly to $\|x_{k+1}\|$ in (2.24), by letting $a = \max_{k=1,...,N} \|A_k^T\|/2$ and assuming $h < 1/a$, so that, for $k = 1, ..., N$,

$$\begin{aligned}
\|\mu_k\| &\leq \sum_{i=k}^{N}\left(\Pi_{j=k}^{i-1}\|(I - \frac{h}{2}A_j^T)^{-1}\|\|I + \frac{h}{2}A_j^T\|\right)\|(I - \frac{h}{2}A_i^T)^{-1}\|\|\dot{q}_i\| \\
&\leq \sum_{i=1}^{N}\frac{(1+ha)^{i-1}}{(1-ha)^i}\|\dot{q}_i\| \leq \frac{(1+ha)^N}{(1-ha)^N}\sum_{i=1}^{N}\|\dot{q}_i\| \leq 2e^{2a}\sum_{i=1}^{N}\|\ddot{q}_i\| \\
&= \frac{c}{h}\|\ddot{q}\|_{L^1}.
\end{aligned}$$

Also,

$$\|\mu_0\| \leq \|(I + \frac{h}{2}A_0^T)\|\|z_0\| = c\|z_0\| \leq \frac{c}{h}\|\ddot{q}\|_{L^1}$$

so that

$$\|\mu\|_{L^\infty} = \max_k \|\mu_k\| \leq \frac{c}{h}\|\ddot{q}\|_{L^1}.$$

and therefore by Lemma 2.2

$$\|M_1^T \ddot{q}\|_{L^2} = \|\mu\|_{L^2} \leq \frac{c}{h}\|\ddot{q}\|_{L^1}.$$

Thus we have

$$\|\ddot{y}\|_{L^2} \leq \frac{c}{h}(\|\ddot{p}\|_{L^1} + \|\ddot{q}\|_{L^1} + \|\ddot{r}\|_{L^2}.) \tag{2.26}$$

Now subtract $L(w^1) - \pi^2 = 0$ from $L(w^2) - \pi^1 = 0$ to obtain

$$\begin{aligned} \ddot{x}_{k+1} &= \ddot{x}_k + h\left(A_k\ddot{x}_k + B_k\ddot{y}_k + A_{k+1}\ddot{x}_{k+1} + B_{k+1}\ddot{y}_{k+1}\right) + \ddot{p}_k, && (2.27\text{a}) \\ & \quad k = 1, ..., N-1 \\ \ddot{x}_0 &= 0 && (2.27\text{b}) \end{aligned}$$

$$\begin{aligned} \ddot{\lambda}_k &= \ddot{\lambda}_{k+1} + \frac{h}{2}A_k^T\left(\ddot{\lambda}_k + \ddot{\lambda}_{k+1}\right) + h(Q_{1_k}\ddot{x}_k + Q_{3_k}\ddot{y}_k) + \ddot{q}_k, && (2.28\text{a}) \\ & \quad k = 1, ..., N-1 \\ \ddot{\lambda}_N &= \frac{h}{2}A_N^T\ddot{\lambda}_N + \frac{h}{2}(2Vx_N + Q_{1_N}\ddot{x}_N + Q_{3_N}\ddot{y}_N) + \ddot{q}_N && (2.28\text{b}) \end{aligned}$$

Compare (2.24) to (2.27) to conclude

$$\|\ddot{x}\|_{L^\infty} \leq \frac{c}{h}\|h\bar{B}\ddot{y} + \ddot{p}\|_{L^1} \leq c\|\ddot{y}\|_{L^2} + \frac{c}{h}\|\ddot{p}\|_{L^1}.$$

Next, compare (2.25) to (2.28) to conclude

$$\begin{aligned} \|\ddot{\lambda}\|_{L^\infty} &\leq c\sum_{k=1}^{N-1}(h\|Q_{1_k}\ddot{x}_k + Q_{3_k}\ddot{y}_k\| + \|\ddot{q}_k\|) + \frac{ch}{2}\|2V\ddot{x}_N + Q_{1_N}\ddot{x}_N + Q_{3_N}\ddot{y}_N\| + \|\ddot{q}_N\| \\ &\leq c\|\ddot{y}\|_{L^1} + c\|\ddot{x}\|_{L^1} + \frac{c}{h}\|\ddot{q}\|_{L^1} \\ &\leq c\|\ddot{y}\|_{L^2} + \frac{c}{h}(\|\ddot{p}\|_{L^1} + \|\ddot{q}\|_{L^1}). \end{aligned}$$

Combining these results with (2.26), we have

$$\|x^1 - x^2\|_{L^\infty} + \|y^1 - y^2\|_{L^2} + \|\lambda^1 - \lambda^2\|_{L^\infty} \leq \frac{c}{h}(\|p^1 - p^2\|_{L^1} + \|q^1 - q^2\|_{L^1} + \|r^1 - r^2\|_{L^2}).$$

This completes the proof of Theorem 2.7.

□

THEOREM 2.8. *Given the previous definitions and assumptions it follows in the norm given by (2.18) that,*

$$\begin{aligned} \|T(\hat{w}^*)\| = \|(\hat{p}, \hat{q}, \hat{r})\| &= ch^2 \\ \|T(\check{w}^*)\| = \|(\check{p}, \check{q}, \check{r})\| &= ch^3. \end{aligned}$$

Proof. By definition of $\hat{y}, \check{y}$, we have $\|\hat{r}\|_{L^2} = \|\check{r}\|_{L^2} = 0$. Using Lemma 2.5 and the fact that f is Lipschitz continuous we have, for $k = 0, ..., N-1$,

$$\begin{aligned} \|\hat{p}_k\| &= \| - x^*(t_{k+1}) + x^*(t_k) + \frac{h}{2}(f(x^*(t_k), \hat{y}_k) + f(x^*(t_{k+1}), \hat{y}_{k+1}))\| \\ &= \| - x^*(t_{k+1}) + x^*(t_k) + \frac{h}{2}(f(x^*(t_k), y^*(t_k)) + f(x^*(t_{k+1}), y^*(t_{k+1})) \\ &\quad + \frac{h}{2}(f(x^*(t_k), \hat{y}_k) - f(x^*(t_k), y^*(t_k))) + \frac{h}{2}(f(x^*(t_{k+1}), \hat{y}_{k+1}) - f(x^*(t_k), y^*_{k+1}))\| \\ &\leq \| - x^*(t_{k+1}) + x^*(t_k) + \frac{h}{2}(f(x^*(t_k), y^*(t_k)) + f(x^*(t_{k+1}), y^*(t_{k+1})))\| \\ &\quad + ch\|\hat{y}_k - y^*(t_k) + \hat{y}_{k+1} - y^*(t_{k+1})\| \\ &\leq c_1 h^3 + c_2 h^2 = ch^2. \end{aligned}$$

giving $\|\hat{p}\|_{L^1} = ch^2$.

For $\check{p}$ we have, for $k = 1, ..., N-2$,

$$\|\check{p}_k\| \leq c_1 h^3 + ch\|\check{y}_k - y^*(t_k) + \check{y}_{k+1} - y^*(t_{k+1})\| = ch^3.$$

But

$$\begin{aligned} \|\check{p}_0\| &\leq c_1 h^3 + ch\|\check{y}_0 - y^*(t_0) + \check{y}_1 - y^*(t_1)\| \\ &\leq c_1 h^3 + h(c_2 h + c_3 h^2) = ch^2 \end{aligned}$$

and

$$\begin{aligned}
\|\check{p}_{N-1}\| &\leq c_1h^3 + ch\|\check{y}_{N-1} - y^*(t_{N-1}) + \check{y}_N - y^*(t_N)\| \\
&\leq c_1h^3 + h(c_2h^2 + c_3h) = ch^2.
\end{aligned}$$

So

$$\|\check{p}\|_{L^1} \leq \left(\sum_{k=1}^{N-2} h(c_1h^3)\right) + 2h(c_2h^2) \leq ch^3.$$

To evaluate $\hat{q}$ and $\check{q}$, we first rewrite them in terms of H_x. Thus we have, for $k = 1, ..., N-1$,

$$\begin{aligned}
\hat{q}_k &= \lambda^*(t_{k+1}) - \lambda^*(t_k) + \frac{h}{2} f_x(x^*(t_k), \hat{y}_k)^T(\lambda^*(t_k) + \lambda^*(t_{k+1})) \\
&= \lambda^*(t_{k+1}) - \lambda^*(t_k) + hH_x\left(x^*(t_k), \hat{y}_k, \frac{\lambda^*(t_k) + \lambda^*(t_{k+1})}{2}\right)
\end{aligned}$$

and

$$\begin{aligned}
\hat{q}_N &= -\lambda^*(t_N) + \frac{h}{2} f_x(x^*(t_N), \hat{y}_N)^T\lambda^*(t_N) + C_x(x^*(t_N)) \\
&= \frac{h}{2} H_x(x^*(t_N), \hat{y}_N, \lambda^*(t_N))
\end{aligned}$$

and for $\check{q}$ we have

$$\begin{aligned}
\check{q}_k &= \lambda^*\left(t_k + \frac{h}{2}\right) - \lambda^*\left(t_{k-1} + \frac{h}{2}\right) \\
&\quad + hH_x\left(x^*(t_k), \check{y}_k, \frac{\lambda^*\left(t_k + \frac{h}{2}\right) + \lambda^*\left(t_{k-1} + \frac{h}{2}\right)}{2}\right), k = 1, ..., N-1 \\
\check{q}_N &= -\lambda^*\left(t_N - \frac{h}{2}\right) + \frac{h}{2} H\left(x^*(t_N), \check{y}_N, \lambda^*\left(t_N - \frac{h}{2}\right)\right) + \lambda^*(t_N).
\end{aligned}$$

Next, let $\tilde{H}(t, \lambda) = H_x(x^*(t), y^*(t), \lambda)$. Consider the RK method given by $\alpha = [1/2], b = [1], \sigma = [0]$. Checking Table 1 in [21], we determine that it is a 2nd order method. So, integrating $\lambda' = -\tilde{H}(t, \lambda)$ from t_k to t_{k+1} with step size h, we obtain

$$\tilde{q}_k^1 = -\lambda^*(t_{k+1}) + \lambda^*(t_k) - hH_x\left(x^*(t_k), y^*(t_k), \frac{\lambda^*(t_k) + \lambda^*(t_{k+1})}{2}\right) = O(h^3).$$

Applying the 2nd order RK method given by $\alpha = [1/2], b = [1], \sigma = [1/2]$ with step size h to the same equation from $t_{k-1} + \frac{h}{2}$ to $t_k + \frac{h}{2}$, we obtain

$$\begin{aligned}\tilde{q}_k^2 &= -\lambda^* \left(t_k + \frac{h}{2} \right) + \lambda^* \left(t_{k-1} + \frac{h}{2} \right) \\ &\quad -hH_x \left(x^*(t_k), y^*(t_k), \frac{\lambda^* \left(t_k + \frac{h}{2} \right) + \lambda^* \left(t_{k-1} + \frac{h}{2} \right)}{2} \right) = O(h^3).\end{aligned}$$

We also define

$$\tilde{q}^3 = \lambda^* \left(t_N - \frac{h}{2} \right) - \frac{h}{2} H \left(x^*(t_N), y^*(t_N), \lambda^* \left(t_N - \frac{h}{2} \right) \right) + \lambda^*(t_N) = O(h^2),$$

which is obtained by applying the 1st order RK method given by $\alpha = [0], b = [1], \sigma = [1]$, with step size $h/2$ to the equation $\lambda' = -\tilde{H}(t, \lambda)$ from $t_N - \frac{h}{2}$ to t_N.

Finally, we use Lipschitz continuity of H_x and λ^* and Lemma 2.5 to show

$$\begin{aligned}\|\hat{q}_k\| &= \|\hat{q}_k + \tilde{q}_k^1\| \leq ch^3 + ch\|\hat{y}_k - y^*(t_k)\| \leq ch^2, \\ \|\hat{q}_N\| &\leq ch\end{aligned}$$

and

$$\begin{aligned}\|\check{q}_k\| &= \|\check{q}_k + \tilde{q}_k^2\| \leq ch^3 + ch\|\check{y}_k - y^*(t_k)\| \leq ch^3, \\ \|\check{q}_N\| &= \|\check{q}_N + \tilde{q}^3\| \leq h^3c + ch\|\check{y}_N - y^*(t_N)\| \leq ch^2.\end{aligned}$$

Thus

$$\|\hat{q}\|_{L^1} \leq \left(\sum_{k=1}^{N-1} h(c_1h^2) \right) + h(c_2h) \leq ch^2$$

and

$$\|\check{q}\|_{L^1} \leq \left(\sum_{k=1}^{N-1} h(c_1h^3) \right) + h(c_2h^2) \leq ch^3.$$

□

With these lemmas and theorems in place we can now prove our main theorem.

Proof. [Proof of Theorem 2.6] Let $\hat{\delta} = -T(\hat{w})$ and $\check{\delta} = -T(\check{w})$. By Theorem 2.8 we have, in the norm defined in (2.18),

$$\|\hat{\delta}\| = \|\hat{p}\|_{L^1} + \|\hat{q}\|_{L^1} + \|\hat{r}\|_{L^2} \leq c(h^2 + h^2 + 0) = ch^2$$

and similarly

$$\|\check{\delta}\| \leq c(h^3 + h^3 + 0) = ch^3.$$

By Theorem 2.7, (Q3) of Lemma 2.1 holds for both $\hat{w}$ and $\check{w}$ with $\gamma = c/h$ and $\tau = \infty$, and (Q2) follows from [14], Lemma 5.1, so that we have

$$\| \bigtriangledown T(w) - L\| \leq ch(\|w - w^*\| + h)$$

For $\epsilon = ch(\|w - \hat{w}\| + h)$ (where c is the same constant as in the previous equation), choose r independent of h but small enough to satisfy $\epsilon\gamma < 1$. Since $\tau = \infty$, we have $\epsilon r \leq \tau$ and $(1 - \gamma\epsilon)r/\gamma \geq r/\gamma = ch \geq ch^2 \geq \hat{\delta}$. Then by Lemma 2.1 there exists a locally unique w such that $T(w) = 0$, that is, w satisfies the first order necessary conditions for the discretized problem. Moreover,

$$\|w - \hat{w}\| \leq \frac{\gamma}{1 - \gamma\epsilon}\|\hat{\delta}\| = c\gamma h^2 = ch.$$

By the same reasoning, we have

$$\|w - \check{w}\| \leq c\gamma\|\check{\delta}\| \leq ch^2.$$

By our definition of the norm on w (see (2.17)), this implies

$$\|x - x^*\|_{L^\infty} + \|\lambda - \hat{\lambda}\|_{L^\infty} \leq ch \tag{2.29}$$

$$\|x - x^*\|_{L^\infty} + \|\lambda - \check{\lambda}\|_{L^\infty} \leq ch^2 \tag{2.30}$$

and the first three identities of Theorem 2.6 (equations (2.7a) – (2.7c)) follow immediately. The fourth identity, (2.7d) follows from (2.7c) by smoothness of λ, indeed,

$$\left\|\frac{\lambda_k + \lambda_{k+1}}{2} - \lambda^*(t_k)\right\| \leq \left\|\frac{\lambda^*\left(t_k - \frac{h}{2}\right) + \lambda^*\left(t_k + \frac{h}{2}\right)}{2} - \lambda^*(t_k)\right\| + ch^2 \leq ch^2.$$

Now y_k satisfies $\bar{H}_{y_k}(x, y, \lambda) = 0$ and for $k = 1, ..., N-1$,

$$\bar{H}_{y_k}(x, y, \lambda) = hH_y(x_k, y_k, \frac{\lambda_{k+1} + \lambda_k}{2}),$$

so by Lemma 2.4 and (2.7a),(2.7c) we have

$$\|y_k - y_k^*\| \leq c \left(\|\frac{\lambda_{k+1} + \lambda_k}{2} - \lambda^*(t_k)\| + \|x_k - x^*(t_k)\| \right) \leq ch^2$$

which gives us (2.7e).

□

Chapter 3

Equality Constrained Problems

3.1 Overview

In this chapter we will consider equality constrained optimal control problems of the form

$$\min \quad C(x(t_f)) \tag{3.1a}$$

$$x' = f(t,x,y) \tag{3.1b}$$

$$0 = g(t,x,y) \tag{3.1c}$$

$$x(t_0) = \zeta. \tag{3.1d}$$

The dynamics of an equality constrained problem form a Differential Algebraic Equation (DAE). There is extensive theory dealing with DAEs and methods for solving them (see [10] to start). The following definition is central to that theory

Definition 3.1. *The index of a DAE, $F(x,x',t)=0$, is defined as the minimum number of times all or part of the DAE needs to be differentiated with respect to t to determine x' as a continuous function of x and t.*

Remember from Chapter 1 that indirect and control parameterization methods require that a subset of y, called the control, be designated. Let u be the control and let x_2 be defined by $y = (x_2, u)$. Control parameterization methods require the dynamics to be solved for x, x_2 and indirect methods require that the optimality conditions be solved for x, x_2 and the adjoint variables. All of the classical discretizations of DAEs only converge for index three or less and also require that the DAE have special structure if the index is greater than one. Often the restriction is to index one or two [10].

Direct transcription methods approach the problem differently and do not require the selection of u. Does that mean that the index of the dynamics no longer plays a role in their convergence properties? In the next section, we will present a new convergence result that is based on the property of the dynamics we call the "virtual index" . In our review of existing literature, we found no other results that relate to equality constrained problems with constraints involving x as well as y, i.e. constraints that can be anything but index 1.

Note that some results in the previous chapter are also applicable to problems with general control constraints of the form $y \in U$.

The result in [29] follows a line of thinking most similar to ours. Our assumptions are more restrictive, but we are also able to establish higher order convergence for most Runge-Kutta methods, whereas [29] only proves convergence for Euler's method.

3.2 The Virtual Index

3.2.1 Theoretical Results

Consider a problem of the form

$$\begin{aligned} \min \quad & C(x(t_f)) && (3.2a) \\ x' &= f(t,x,y) && (3.2b) \\ 0 &= \phi(t,x) + F(t,x)y && (3.2c) \\ x(t_0) &= \zeta && (3.2d) \end{aligned}$$

where $\phi : \mathbb{R}^{m_1+1} \rightarrow \mathbb{R}^c$,$F : \mathbb{R}^{m_1+1} \rightarrow \mathbb{R}^{c\times m_2}$.

By definition (see Section 1.5), a RK discretization of (3.2) is given by

$$\begin{aligned} \min \quad & C(x_N) && (3.3a) \\ x_{i+1} &= x_i + h_i \sum_{j=1}^{s} b_j f(t_{ij}, \chi_{ij}, y_{ij}), \quad i = 0, ..., N-1 && (3.3b) \\ \chi_{ij} &= x_i + h_i \sum_{k=1}^{s} a_{jk} f(t_{ij}, \chi_{ik}, y_{ik}), \quad i = 0, ..., N-1, j = 1, ..., s && (3.3c) \\ 0 &= \phi(t_{ij}, \chi_{ij}) + F(t_{ij})y_{ij}, \quad i = 0, ..., N, j = 1, ..., s && (3.3d) \\ x_0 &= \zeta. && (3.3e) \end{aligned}$$

Our main result takes the form

Theorem 3.2. *Let $F(t,x)$ be sufficiently smooth and full row rank on a sufficiently large open neighborhood Ω of Γ, where $\Gamma \in R \times R^n$ is the graph of x^*. Then there exists U which is as smooth as F such that $F\begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} I & 0 \end{pmatrix}$ and for all (x,u) the problem (3.2) is equivalent to the unconstrained problem given by*

$$\begin{aligned} \min \quad & C(x(t_f)) && (3.4a) \\ x' &= \theta(t,x,u) && (3.4b) \\ x(t_0) &= \zeta && (3.4c) \end{aligned}$$

where y is connected to u by the relationship

$$y = -U_1(t,x)\phi(t,x) + U_2(t,x)u(t).$$

Moreover, applying any RK to (3.2) is equivalent to applying the same discretization to (3.4).

Proof. First, we must establish the existence and smoothness of U. For $F = F(t)$ this was established by Evard in [18]. For the more general case, where F is a function of x as well as t, the proof is due to Campbell (see [17]). This result gives us U that is defined and smooth for all t, x and not just locally. Now we can make the substitution $y = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix}$ transforming (3.2) into

$$\begin{aligned} \min \quad & C(x(t_0)) && \text{(3.5a)} \\ x' &= f(t,x,U_1(t,x)v + U_2(t,x)u) && \text{(3.5b)} \\ 0 &= \phi(t,x) + v && \text{(3.5c)} \\ x(t_0) &= \zeta. && \text{(3.5d)} \end{aligned}$$

Then we can eliminate the constraint (3.5c) and express v as a function of x, obtaining the problem

$$\begin{aligned} \min \quad & C(x(t_f)) \\ x' &= f(t,x,-U_1(t,x)\phi(t,x) + U_2(t,x)u) \doteq \theta(t,x,u) \\ \zeta &= x(0), \end{aligned}$$

which is exactly the problem (3.4).

Now, apply the transformation

$$y_{ij} = -U_1(t_{ij},\chi_{ij})\phi(t_{ij},\chi_{ij}) + U_2(t_{ij},\chi_{ij})u_{ij}$$

for $i = 0, ..., N, j = 1, ..., s$ to (3.3) to obtain

$$\begin{aligned} \min \quad & C(x(t_f)) && \text{(3.6a)} \\ x^{i+1} - x^i &= h_i \sum_{j=1}^{s} b_j \theta(t_{ij}, \chi_{ij}, u_{ij}), \quad i = 0, ..., N-1 && \text{(3.6b)} \\ \chi_{ij} &= x_i + h_i \sum_{k=1}^{s} a_{jk} \theta(t_{ik}, \chi_{ik}, u_{ik}), \quad i = 0, ..., N-1, j = 1, ..., s && \text{(3.6c)} \\ x_0 &= \zeta && \text{(3.6d)} \end{aligned}$$

where $\theta(t, x, u) = f(t, x, -U_1(t, x)\phi(t, x) + U_2(t, x)u)$. Notice that (3.6) is the Runge-Kutta discretization of (3.4) with the same Butcher array parameters b_j, a_{ij} and σ_j. □

The implication of Theorem 3.2 is that the convergence results presented in Chapter 2 for unconstrained problems will apply to problems of the form (3.2) as long as $F(t, x)$ is full row rank and as smooth as the order of the RK discretization being used. We will now present corollaries for all of the theorems in Chapter 2.

Corollary 3.3. *If*

- *$F(t, x)$ is full row rank in a sufficiently large open neighborhood of Γ*
- *The optimal control problem (3.1) satisfies smoothness (Definition 2.1) with the conditions applied to F and ϕ as well as C and f*
- *The operators U_1, U_2 are bounded everywhere in the neighborhood of Γ and ϕ is Lipschitz continuous in x*
- *The equivalent unconstrained problem satisfies coercivity (Definition 2.2)*
- *The RK used to discretize the problem is at least 2nd order as an integrator (Table 1.1)*
- *The RK used to discretize the problem satisfies conditions (D1)-(D5) in Section 2.1*
- *The discretized problem has a local optimal solution (x, y)*

then

$$\max_k \|x_k - x^*(t_k)\| + \max_k \|y_k^1 - y^*(t_k)\| \leq ch^2.$$

Proof. Smoothness of the original problem implies smoothness in the equivalent unconstrained problem and so we have, by Theorem 2.3,

$$\max_k \|x_k - x^*(t_k)\| + \max_k \|u_k^1 - u^*(t_k)\| \leq ch^2$$

for the equivalent unconstrained problem. We then have

$$\begin{aligned} \|y_k^1 - y^*(t_k)\| &= \|U_1(t_k)(\phi(t_k, x_k) - \phi(t_k, x^*(t_k))) + U_2(t)(u_k^1 - u^*(t_k))\| \\ &\leq c_1\|x_k - x^*(t_k)\| + c_2\|u_k^1 - u^*(t_k)\| \\ &= O(h^2). \end{aligned}$$

□

Corollary 3.4. *If*

- *$F(t,x)$ is full row rank in a sufficiently large open neighborhood of Γ*
- *The optimal control problem (3.1) satisfies smoothness (Definition 2.1) with the conditions applied to F and ϕ as well as C and f*
- *The operators U_1, U_2 are bounded in a neighborhood of Γ and ϕ is Lipschitz continuous in x*
- *The equivalent unconstrained problem satisfies coercivity (Definition 2.2)*
- *The RK used to discretize the problem is 2nd order as an integrator (Table 1.1)*
- *The RK used to discretize the problem satisfies the conditions (2.4)*

then for all sufficiently small $h = \max h_k$, the discretization of (3.1) obtained according to this Runge-Kutta scheme has a strict local minimizer (x, y) and an associated adjoint variable λ such that, if $\frac{dy^*}{dt}$ has bounded variation,

$$\max_{k=0,\ldots,N_{i=1,\ldots,s}} \|x_k - x^*(t_k)\| + \|y_{ki} - y^*(t_{ki})\| \leq ch^2$$

This corollary follows from Theorem 2.4 in the same way that the previous one follows from Theorem 2.3. We have left out the adjoint error because, as we have seen in the previous section, multipliers are not necessarily equivalent for two different formulations of the same problem and, moreover, multiplier convergence is implementation-dependent, limiting the practical usefulness of most existing multiplier- convergence results.

Finally, we have a corollary of Theorem 2.5. Since we are not interested in control convergence, we can leave out the conditions on U_1, U_2 and ϕ for this one:

COROLLARY 3.5. *If*

- $F(t, x)$ *is full row rank in a sufficiently large open neighborhood of* Γ
- *The optimal control problem (3.1) satisfies smoothness (Definition 2.1) with the conditions applied to* F *and* ϕ *as well as* C *and* f
- *The equivalent unconstrained problem satisfies coercivity (Definition 2.2)*
- *The RK used to discretize the problem is order* κ *for optimal control (Table 2.1) and has* $b_i > 0$ *for each* i

then for all sufficiently small $h = \max h_k$*, the discretization of (3.1) obtained according to this Runge-Kutta scheme has a strict local minimizer* (x, y) *such that, if* $\frac{dy^*}{dt}$ *has bounded variation,*

$$\max_{k=0,\ldots,N} ||x_k - x^*(t_k)|| \leq ch^{\kappa}.$$

All of these corollaries require coercivity, which is a difficult condition to test for in practice. However, note that given matrices $Q_i(t), V(t)$ defined for the equivalent unconstrained problem, if $Q_3(t) = 0$ for all x, y, λ then $x^T V x \geq 0$ and $x^T Q_1 x \geq 0$ for all x and $u^T Q_2(t) u > \alpha \|u\|^2$ for all u, t forms a set of sufficient conditions for coercivity. Then note that if H is the Hamiltonian to the original problem and $\bar{H}$ is the Hamiltonian to the equivalent unconstrained problem then

$$Q_2(t) = \bar{H}_{uu}(x^*, u^*, \lambda^*) = U_2^T H_{yy}(x^*, y^*, \nu^*, \eta^*)U_2.$$

In the following sections, we will illustrate our result with several examples. The first example is fairly straightforward, the 2nd presents a more complicated time-dependent U matrix, the third example models a physical problem and presents F and U that are dependent on both t and x. However, all have linear constraints and simple dynamics and cost structure that allow for easy verification of coercivity. For each of these examples, we will see how performance of the method is altered when the cost parameters are changed so that coercivity is no longer satisfied.

As the examples will demonstrate, ensuring coercivity is often simply a matter of regularizing the problem with respect to all algebraic variables. Thus if the user sees the problem as high index due to his choice of control, but he suspects that another choice of control might make it an index one problem, he does not need to calculate what that "virtual control" is or perform the transformation. All he has to do to make sure that a direct transcription code will solve the problem correctly is regularize it with respect to all algebraic variables, instead of just the one he thinks of as the control.

3.2.2 Example 1: Time-Invariant Transformation

In the following example problem, the choice of control that makes the problem index 1 is fairly obvious. It is also designed is such a way as to make it easy to find the exact optimal solution and compare it to the result given by SOCS. We use it as a simple illustration of the concepts in the previous section.

Let h_i, f_i be continuous functions. Consider the following problem.

$$\min \frac{1}{2} \int_0^1 \sum_{i=1}^{5} q_i(z_i - h_i)^2 dt \tag{3.7a}$$

$$\begin{aligned}
z_1' &= z_4 + f_1(t) \quad &(3.7\text{b})\\
z_2' &= z_1 + f_2(t) \quad &(3.7\text{c})\\
z_3' &= z_2 + f_3(t) \quad &(3.7\text{d})\\
0 &= z_3 + z_5 + f_4(t) \quad &(3.7\text{e})\\
z_1(0) &= z_1(1) = 1 \quad &(3.7\text{f})\\
z_2(0) &= z_2(1) = -1 \quad &(3.7\text{g})\\
z_3(0) &= z_3(1) = 1. \quad &(3.7\text{h})
\end{aligned}$$

Here, z_1, z_2 and z_3 are differential variables, and z_4, z_5 are algebraic variables. Suppose that the user designates the algebraic variable z_5 as the control. Treating z_5 as a known input function and differentiating the constraint (3.7e) four times, we obtain

$$z_4' = -f_1' - f_2'' - f_3''' - f_4'''' - z_5'''',$$

an equation for z_4' in terms of other variables.

This makes the DAE (3.7b)–(3.7e) index 4 in the remaining variables $\{z_1, z_2, z_3, z_4\}$. As expected, given this choice of control, an indirect method (implemented through SOCS) fails on this problem.

However, if we let z_4 be the control, then differentiating (3.7e) just once, we have

$$z_5' = -f_4' - z_2 - f_3,$$

so the DAE (3.7b)–(3.7e) is index 1 for this choice of control.

We can also confirm this using the theory we developed in the previous chapter. For this problem, $F = (0\ 1)$, so

$$\begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad U_2(t)^T \nabla_{yy} H U_2(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} q_4 & 0 \\ 0 & q_5 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = q_4$$

(see note on the Hamiltonian for problems in equivalent forms at the end of Section 1.3.)

We also have $V = 0$ and $Q_3 = 0$ for all t.

$$x^T Q_1 x = q_1 x_1^2 + q_2 x_2^2 + q_3 x_3^2 \geq 0$$

for all x, t as long as $q_1, q_2, q_3 \geq 0$ and $u^T Q_2 u = q_4 u^2 > \alpha u^2$ as long as $q_4 > \alpha$.

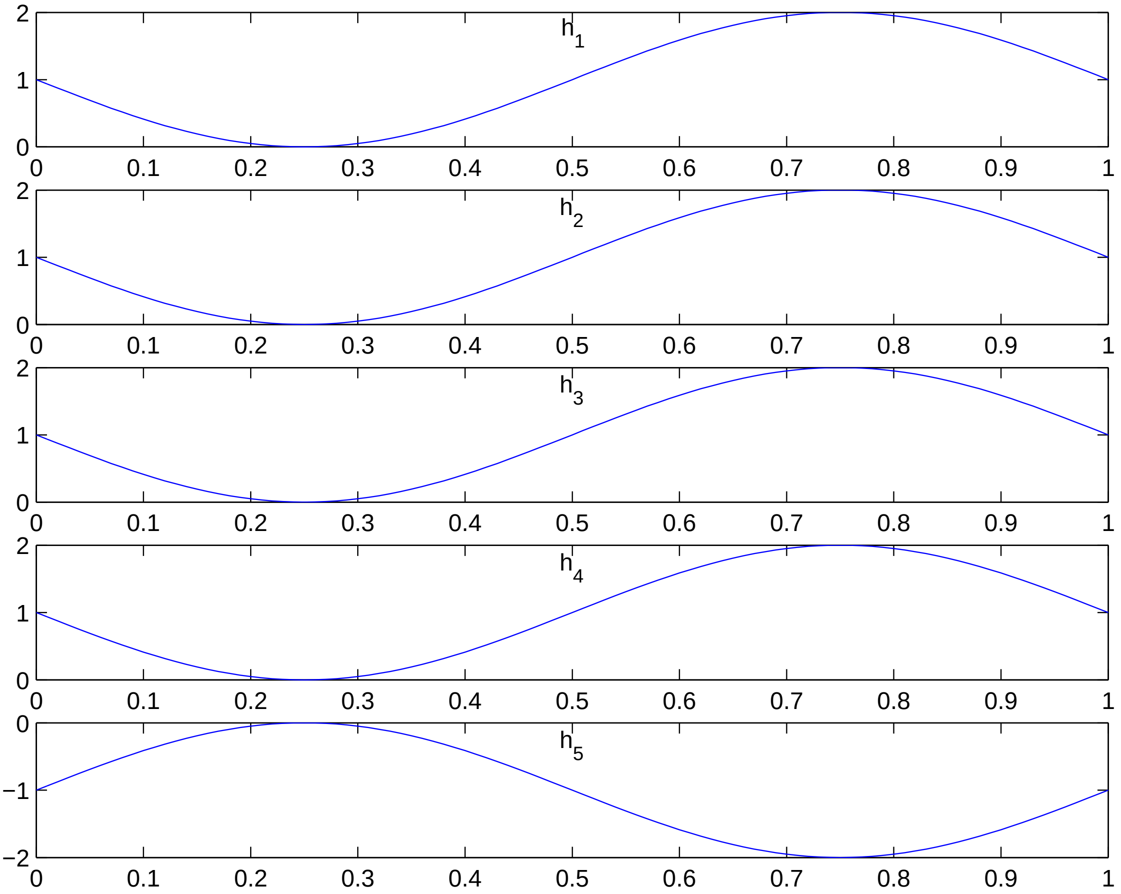


Figure 3.1: Functions $h_i = \pm(\sin(2\pi t) - 1)$ for Example 1. ($i = 1$ is top graph.)

To demonstrate numerically how the accuracy of the solution is affected by the value of q_4, we pick smooth functions $h_i(t)$ with amplitude 1 (Figure 3.1) and choose $f_i(t)$ so that $z_i = h_i, i = 1, ..., 5$ satisfies the dynamics and the constraints. We then

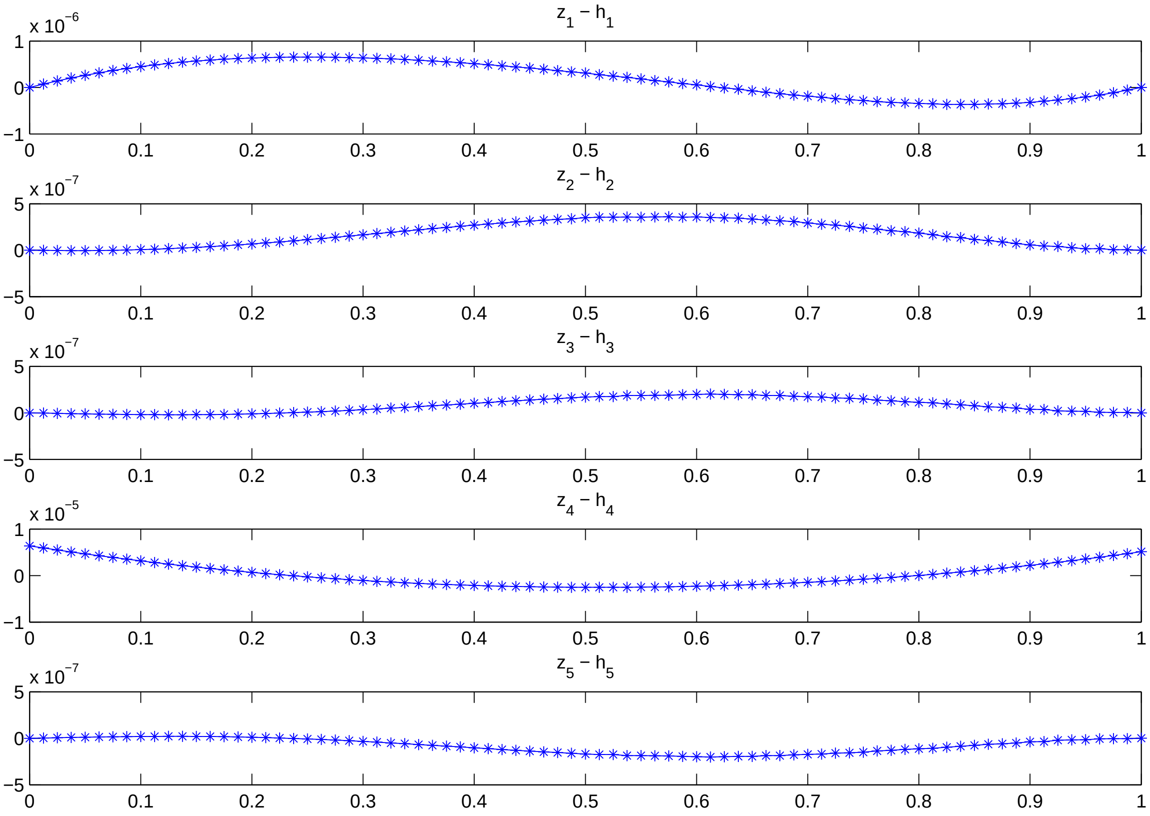


Figure 3.2: Error graphs $z_i - h_i$ for Example 1 using $h_i = \pm(\sin(2\pi t) - 1), q_i = 1$. ($i = 1$ is the top graph)

solve the problem (3.7) using standard options in SOCS. SOCS uses HS to produce its final answer. It is a 4th order RK that is also 4th order for optimal control and the problem satisfies all other conditions in the corollaries. Figure 3.2 shows the difference between h_i and the optimal solution calculated by SOCS when all q_i are set to be 1. We also see that setting q_5 and q_3 to zero does not affect the accuracy. However, when we set $q_4 = 0$, the resulting solution does not satisfy the dynamics as shown by $z_4 - h_4$ in Figure 3.4.

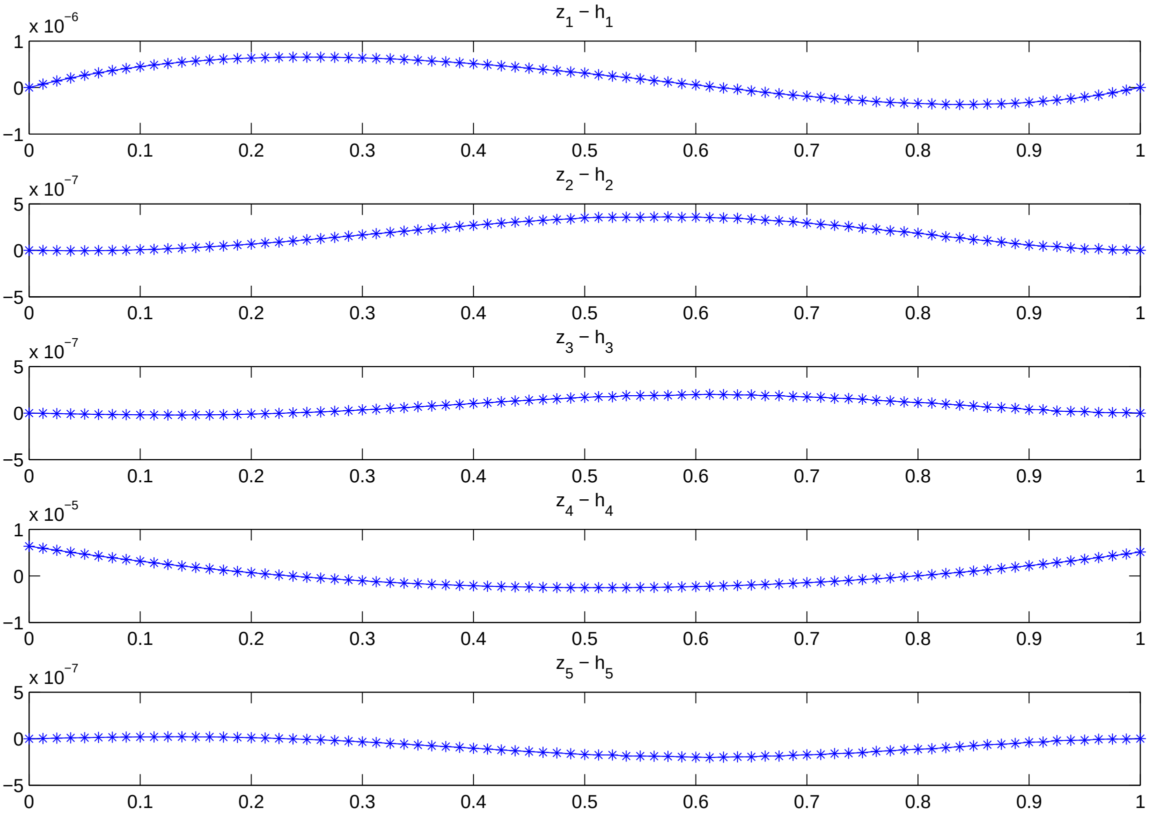


Figure 3.3: Error graphs $z_i - h_i$ for Example 1 using $h_i = \pm(\sin(2\pi t) - 1)$, with $q_1 = q_2 = q_4 = 1$, $q_3 = q_5 = 0$. ($i = 1$ is the top graph.)

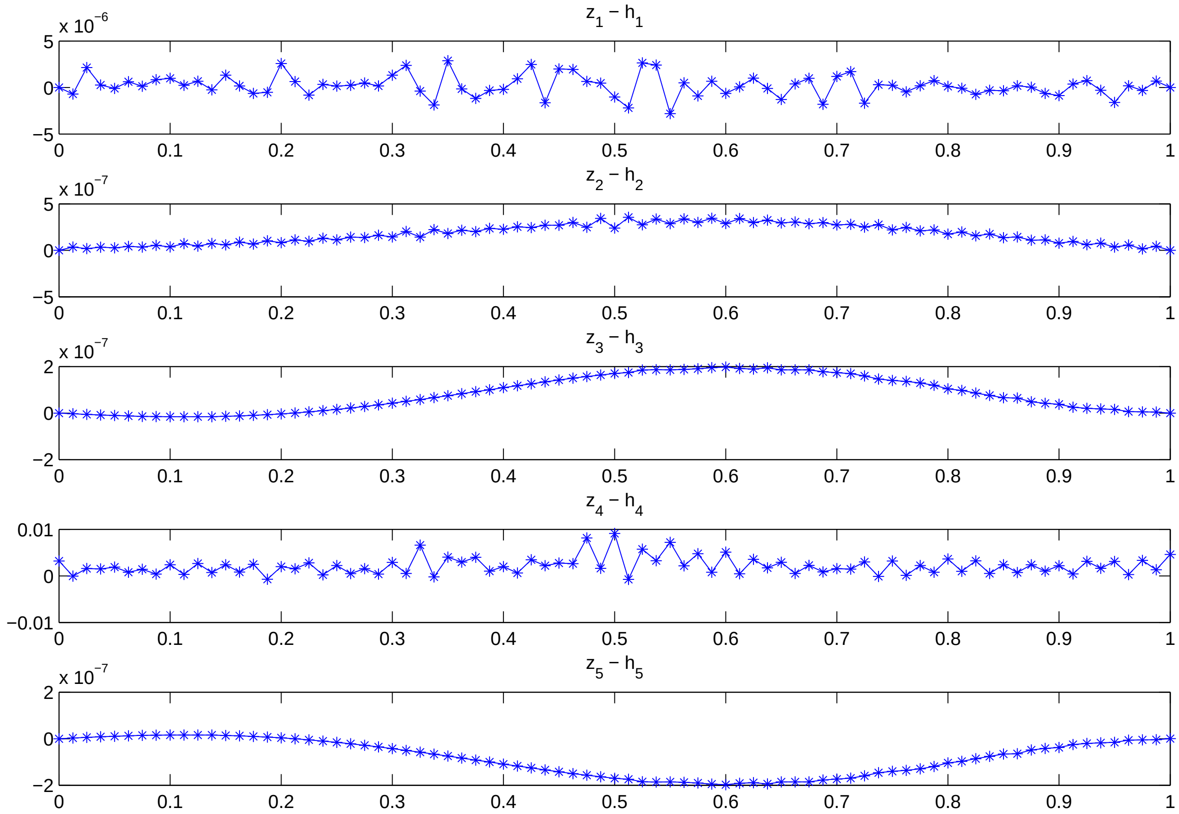


Figure 3.4: Error graphs $z_i - h_i$ for Example 1 using $h_i = \pm(\sin(2\pi t) - 1)$, with $q_1 = q_2 = q_3 = 1$, $q_4 = 0$. ($i = 1$ is the top graph.)

3.2.3 Example 2: Time-Dependent Transformation

Our second example is the following problem with a time varying constraint.

$$\begin{aligned}
\min \quad & \int_0^1 q_1x^2 + q_2y_1^2 + q_3y_2^2 dt && (3.8a)\\
x' \quad = \quad & -x + 2y_1 + 3y_2 && (3.8b)\\
0 \quad = \quad & x + \sin(2\pi t)y_1 + \cos(2\pi t)y_2 && (3.8c)\\
& x(0) = x(1) = 1. && (3.8d)
\end{aligned}$$

Notice that if the user takes either y_1 or y_2 or any fixed linear combination of the two to be the control, then there will always exist point of singularity at which the problem is no longer index 1. Indeed, if $u = ay_1 + by_2, v = y_2$, then

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

and

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1/a & -b/a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

so that

$$\begin{aligned}
\sin(2\pi t)y_1 + \cos(2\pi t)y_2 &= \begin{pmatrix} \sin(2\pi t) & \cos(2\pi t) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\\
&= \frac{\sin(2\pi t)}{a}u + \left(\cos(2\pi t) - \frac{b}{a}\sin(2\pi t)\right) v.
\end{aligned}$$

This means the problem is high index in x and v whenever $\cos(2\pi t) = \frac{b}{a}\sin(2\pi t)$.

However, we can calculate

$$\begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} \sin(2\pi t) & \cos(2\pi t) \\ \cos(2\pi t) & -\sin(2\pi t) \end{pmatrix}$$

that transforms the constraint into $0 = x + v$. We also have $V = Q_3 = 0$, $Q_1 = q_1$,

$$Q_2 = U_2^T \nabla_{yy}^2 H U_2 = \cos^2(2\pi t)q_2 + \sin^2(2\pi t)q_3,$$

so q_1 must be non-negative and both q_2 and q_3 must be nonzero on the entire interval for coercivity to be satisfied. When either q_2 or q_3 are 0, SOCS fails to come up with a solution. But when $q_2 = q_3 = 1$, the solution is easily obtained (Figure 3.5).

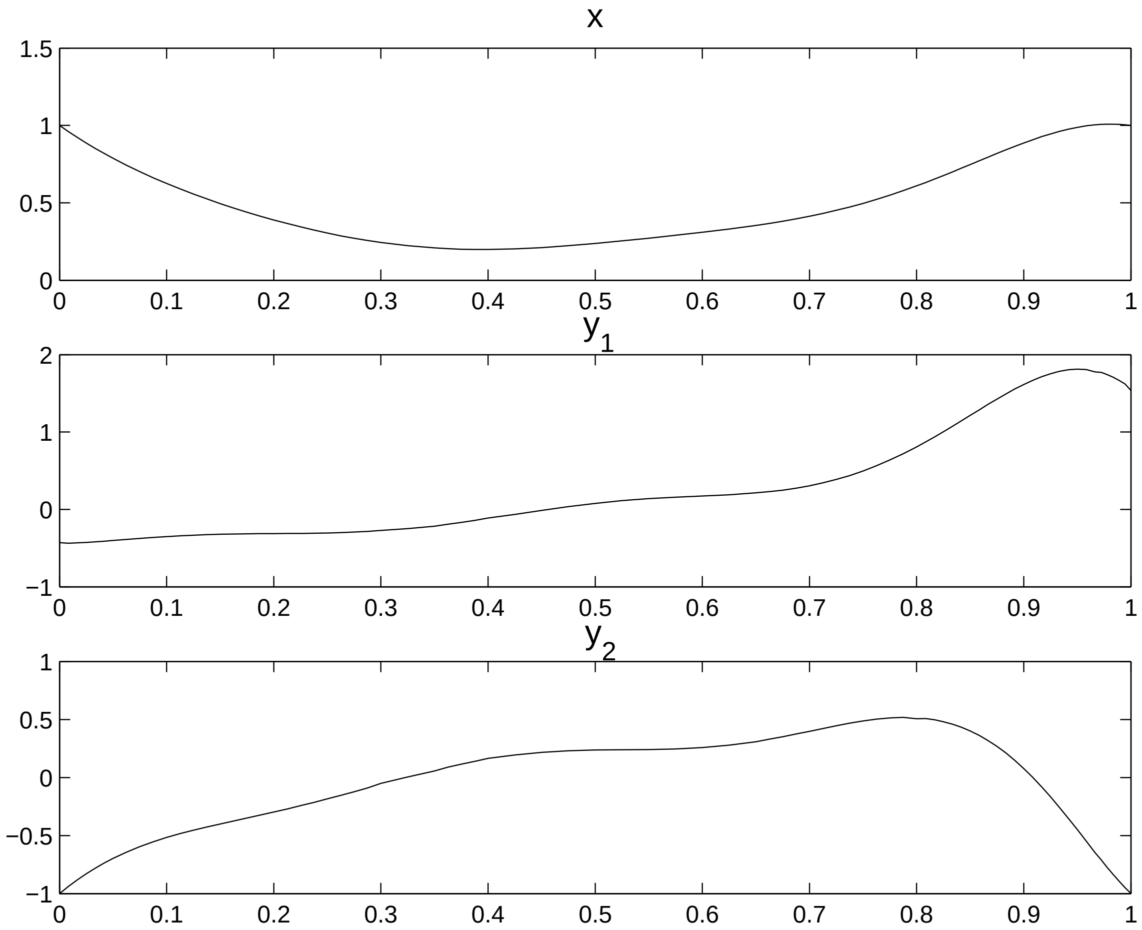


Figure 3.5: Graph of optimal solution for Example 2, $q_i = 1$.

3.2.4 Example 3: Physical Problem, t and x Dependent Transformation

A variety of state constrained mechanical systems can be modeled by

$$\begin{aligned} x' &= v && \text{(3.9a)} \\ v' &= F(x,v,t) + B(x,t)\lambda && \text{(3.9b)} \\ 0 &= G(x,t) && \text{(3.9c)} \end{aligned}$$

where $B = G_x^T$ and λ is a generalized force that arises because of the physical constraint (3.9c).

Now suppose that the constraint (3.9c) depends on a control u and we want to minimize a cost which we will assume in integral form. Then we have a problem in the form

$$\min_u \; P(x(T),T) + \int_0^T L(x,v,u,\lambda,t)dt \qquad \text{(3.10a)}$$

$$\begin{aligned} x' &= v && \text{(3.10b)} \\ v' &= F(x,v,t) + B(x,u,t)\lambda && \text{(3.10c)} \\ 0 &= G(x,u,t). && \text{(3.10d)} \end{aligned}$$

One example of the control acting through the constraint in this manner would be an object, such as a load or vehicle, being pushed by a surface which could be part of another vehicle or machine. The control in this formulation determines the position and orientation of the pushing surface and the constraint (3.10d) models the contact between the object being pushed, which is modeled by (3.10b), (3.10c), and the pushing surface.

It is a well-known and easily verifiable fact that (3.10b)–(3.10d) is an index three DAE in x, v, λ. That is, the optimization problem (3.10) is index three for a given control u. However, if we consider λ to be the control and G_u is invertible, then (3.10b)–(3.10d) is an index one DAE in x, v, u. So that while the designer is not likely to consider the constraint force λ as the control because of implementation considerations, the fact that such a possibility exists in theory makes (3.10) an index one problem for direct transcription software.

A simple instance of the problem (3.10) can be stated as follows:

$$\min \quad x_1(1)^2 + x_2(1)^2 + \int_0^1 q_1 L^2 + q_2 b^2 + q_3 c^2 \, dt \tag{3.11a}$$

$$\begin{aligned}
x_1' &= v_1 && (3.11b)\\
x_2' &= v_2 && (3.11c)\\
v_1' &= -v_1 + L && (3.11d)\\
v_2' &= -v_2 - bL && (3.11e)\\
0 &= x_1 - bx_2 - c && (3.11f)\\
1 &= -x_1(0) = x_2(0) && (3.11g)\\
2 &= -v_1(0) = v_2(0). && (3.11h)
\end{aligned}$$

The dynamics and constraint (3.11f) can be thought of as a flat surface pushing an object at (x_1, x_2) across a smooth flat surface which has the usual type of friction model proportional to the velocity. The friction coefficient is -1 in (3.11d), (3.11e). The controls b, c determine the location and slope of the pushing surface. The initial conditions specify that the object starts at point $(-1, 1)$, moving away from the origin with initial velocity $(-2, 2)$. The objective is to push the object close to the origin while keeping the algebraic variables bounded (or small, depending on the weights).

In this case, $F(t, x) = (0, -x_2, -1)$, and we can take

$$\begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & x_2 \end{pmatrix}.$$

We also have $Q_1 = Q_3 = 0$, $V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, which is nonnegative and

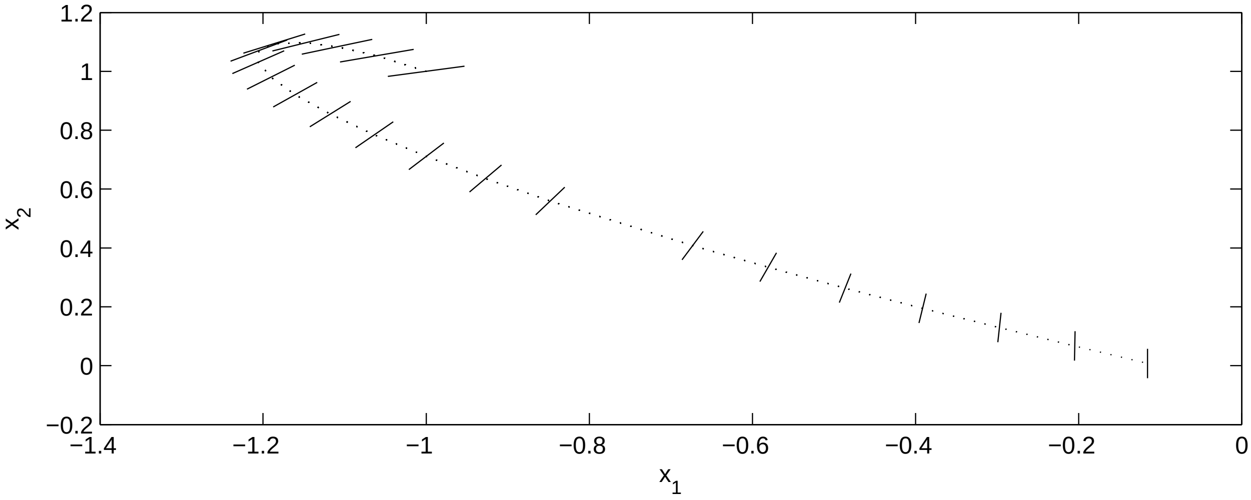
$$Q_2 = U_2^T \nabla^2_{L,b,c} H U_2 = \begin{pmatrix} q_1 & 0 \\ 0 & q_2 + x_2(t)^2 q_3 \end{pmatrix} \tag{3.12}$$

which is positive definite for all x and t as long as q_1 and q_2 are positive.

The calculated trajectory of the object in x_1-x_2 space and the values of L, b, c versus time for $q_1 = q_2 = q_3 = 0.01$ are shown in Figure 3.6. This trajectory achieves $x_1 = -0.1155, x_2 = 0.0074$ at time $t = 1$, so that $\|x(1)\| = 0.0134$. (All values are rounded to four decimal places). The trajectories look the same when q_3 is changed to zero since (3.12) is still nonsingular. As expected, SOCS fails to come up with a solution when just q_1 is changed to zero and (3.12) is always singular. When both q_2 and q_3 are set to zero, the resulting b and c trajectories oscillate wildly (see Figure 3.7). This solution gives $x(1) = (-0.1271, 0), \|x(1)\| = 0.0162$. If we consider getting the object close to the origin to be our main objective, and the second part of the cost function as merely there for regularization, than this is worse by 21 percent than the first solution.

When $q_2 = 0$ and q_1, q_3 are positive, the computed trajectories (Figure 3.8) look similar to Figure 3.6 except at the last node, where b takes a sharp dive and x_2 is pushed a little closer to 0. This is due to the $x_2^2 q_3$ term being nearly singular when x_2 is close to 0 resulting in a near singularity of (3.12). In this case $x(1) = (-0.1183, 0.0042)$ and $\|x(1)\| = 0.014$, 4 percent worse than the first solution. The fact that the main objective is not satisfied as well does not necessarily mean that the solutions we obtained are suboptimal for their respective problems, but our theory allows us to conclude that the first solution is definitely optimal in the case $q_1 = q_2 = q_3 = 0.01$.

The numerical examples confirm the theoretical result in the previous section. As long as there is some choice of control that will make the dynamics of the problem an index one DAE and this ”virtual control” is positively weighted in the cost, the code (in this case, SOCS) can find an optimal solution. Thus what is important numerically is this ”virtual index.” One consequence of this is that a user seeking to regularize a constrained control problem that appears to have high index dynamics should add a small regularizing cost on all the algebraic variables rather than just the control variables as is common practice.



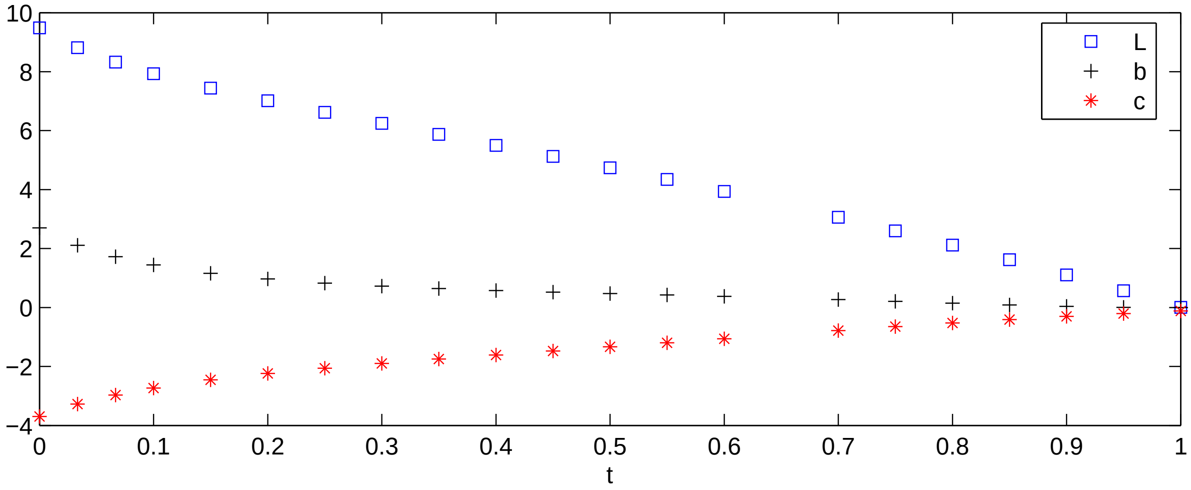


Figure 3.6: Graph of calculated trajectory of the object in x_1-x_2 space with time-lapsed view of the pushing surface (upper graph) and graph of L, b, c vs time (lower graph) for Example 3, $q_i = 0.01$.

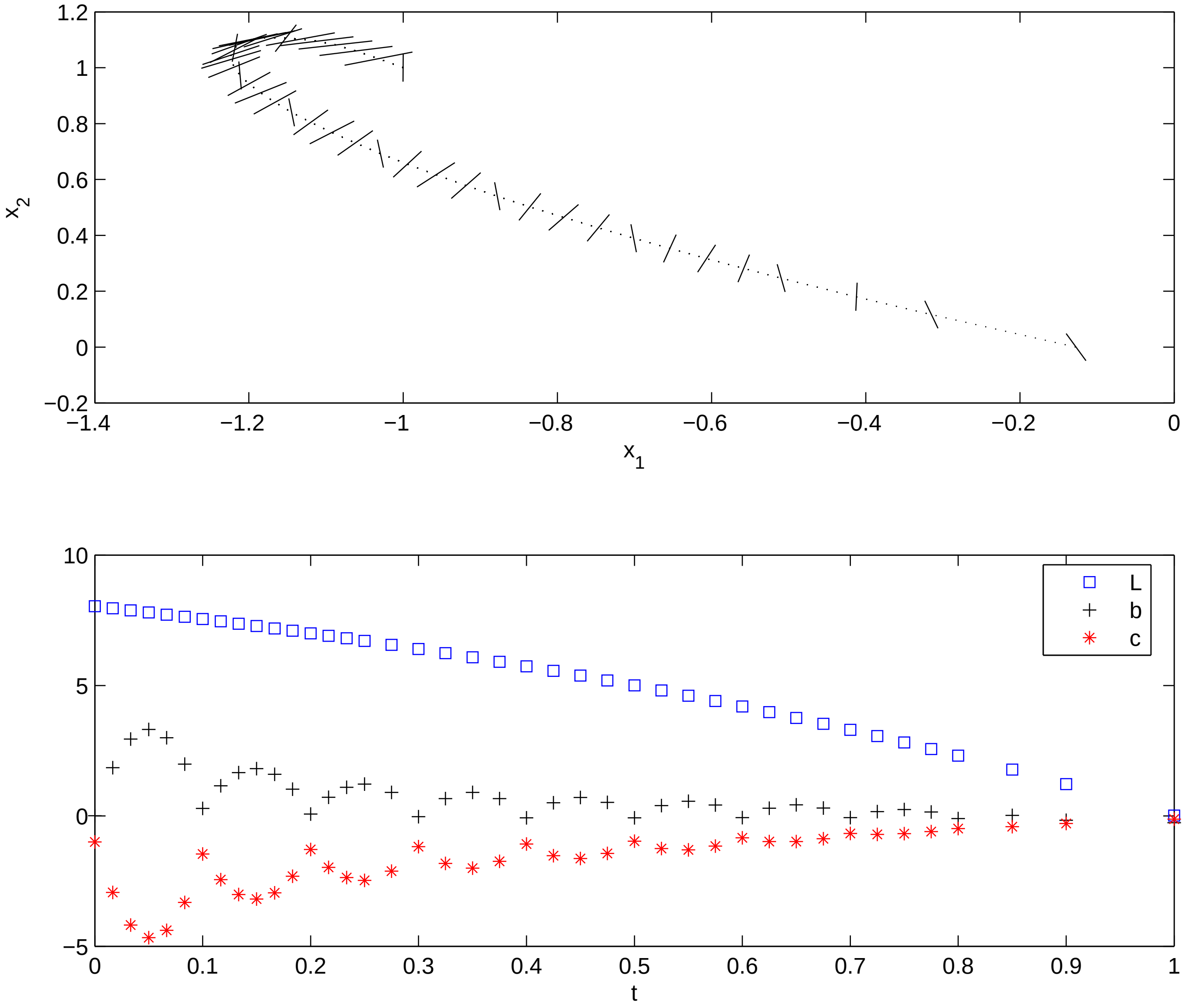


Figure 3.7: Graph of calculated trajectory of the object in x_1-x_2 space and graph of L, b, c vs time for Example 3, $q_1 = 0.01, q_2 = q_3 = 0$.

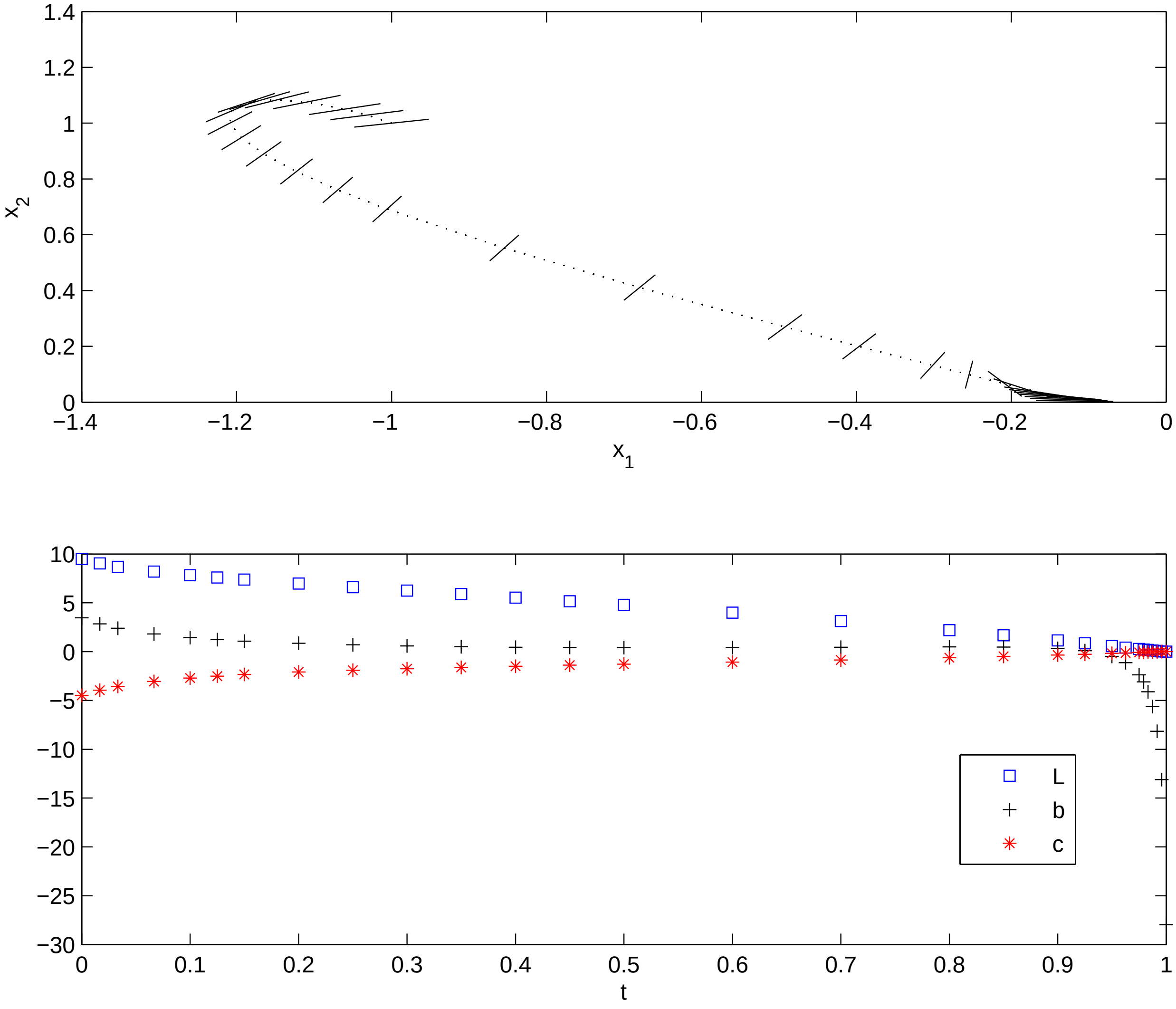


Figure 3.8: Graph of calculated trajectory of the object in x_1-x_2 space and graph of L, b, c vs time for Example 3, $q_1 = q_3 = 0.01, q_2 = 0$.

Chapter 4

Inequality Constrained Problems

4.1 Overview

In this chapter we will consider problems of the form

$$\min C(x(t_f)) \tag{4.1a}$$

$$\begin{aligned} x' &= f(x,y) && (4.1\text{b}) \\ 0 &\geq g(x,y) && (4.1\text{c}) \\ x(t_0) &= \zeta && (4.1\text{d}) \\ \psi(x(t_f)) &= 0. && (4.1\text{e}) \end{aligned}$$

Remember that in order to solve an optimal control problem with an indirect method one must solve the boundary value problem made up of its first-order optimality conditions. So one must have a set of first-order necessary conditions that, under 2nd order assumptions such as coercivity (Definition 2.2) will have a unique solution that is also a local minimum for the problem. Direct transcription does not require defining or solving the optimality conditions, but, even though there are other ways to prove convergence for some discretization methods (see [30, 32]), most authors whose work is discussed in this chapter use some form of necessary conditions in their proofs.

However, finding such a set of optimality conditions is a very nontrivial problem for an inequality constrained problem. The authors of the two convergence results described in this section (Theorems 4.6 and 4.7) use two distinctly different Hamiltonians and derive two different sets of optimality conditions, (4.6) and (4.11).

The problem is that in order to derive a good set of optimality conditions even under very restrictive assumptions on problem structure one must have some knowledge of the behavior of the optimal solution. At any given time τ and for any given constraint $g_i(x(t), y(t), t) \leq 0$, one of three things is possible with regards to the optimal solution $(x^*(t), y^*(t))$:

1. the constraint is inactive, $g_i(x^*(\tau), y^*(\tau), \tau) < 0$;

2. the solution lies on a boundary arc, $g_i(x^*(t), y^*(t)) = 0$ at $t = \tau$ and, for some $\bar{\delta} > 0$, $g_i(x^*(t), y^*(t)) = 0$ for either $t = \tau + \delta$ or $t = \tau - \delta$ for all $\delta < \bar{\delta}$;

3. there is a touchpoint $(g_i(x^*(t), y^*(t)) = 0$ at $t = \tau$ but $g_i(x^*(t), y^*(t)) < 0$ for $t \in (\tau - \delta) \cup (\tau + \delta)$ for some $\delta > 0$.

Thus, to formulate a set of optimality conditions that has a unique solution, one must first guess at the number of boundary arcs and touchpoints in the solution, and the order in which they occur. One set of optimality conditions that takes into account problem structure was developed by Jacobson, Lele and Speyer in their 1977 paper [24]. The same paper also contains a result concerning the existence of boundary arcs that will be very important in the next section. A concept central to Jacobson, Lele and Speyer's paper is that of **order**, which can be applied to both equality and inequality constraints.

Definition 4.1. *The* ***order*** *of a constraint is the minimum number of times it* *needs to be differentiated with respect to t to obtain an expression that can be solved for y.*

The paper [24] makes the following assumptions about a problem of the form (4.1) for all $t \in [t_0, t_f]$:

A1 $m_2 = c = 1$ (g and y are scalars)

A2 g is a function of x only and is of order p

A3 y is piecewise continuous, $\sup_t ||y(t)|| < \infty$

A4 f is continuously differentiable up to $(p+1)$ times in both x and y

A5 The problem has an optimal solution with finite cost

A6 $g^{(2p-1)}$ exists

A7 Along a boundary arc, the y that maintains $g^{(p)}(x, y) = 0$ is p times continuously differentiable w.r.t. t.

A8 Along the optimal solution, $\bigtriangledown_y g^{(p)}(x, y) \neq 0$

Jacobson, Lele and Speyer define

$$H(w) = \lambda^T f(x, y) + \nu^T g(x, y) \tag{4.2}$$

where $w = (x, y, \lambda, \nu)$ and prove

Theorem 4.2. *([24], Theorem 5) If a problem of the form (4.1) satisfies the assumptions (A1)-(A9) then a set of necessary conditions for optimality is formed by*

$$x' = f(x,y) \tag{4.3a}$$

$$\lambda' = -\nabla_x H(w) \tag{4.3b}$$

$$0 = \nabla_y H(w) \tag{4.3c}$$

$$x(t_0) = \zeta \tag{4.3d}$$

$$\lambda(t_f) = -\nabla C(x(t_f)) - \nabla_x(\pi^T \phi(x(t_f))) \tag{4.3e}$$

where $\nu(t) \geq 0$ when $g(x(t)) = 0$ and $\nu(t) = 0$ when $g(x(t)) < 0$. Moreover, at the junction points t_i of boundary and interior arcs (i.e. at the touchpoints or beginning and end of boundary arcs),

$$\lambda(t_i^+) = \lambda(t_i^-) - \eta(t_i)\nabla g(x(t_i)) \tag{4.4a}$$

$$H(w(t_i^+)) = H(w(t_i^-)) \tag{4.4b}$$

$$\eta(t_i) \geq 0 \tag{4.4c}$$

This theorem has an important corollary :

Corollary 4.3. *([24], page 274) If the problem satisfies the assumptions (A1)–(A9) and the order of the constraint p is odd and greater than 1, then the optimal solution will, at most, only touch the boundary.*

This result restricts the appearance of boundary arcs in optimal solutions to high-order inequality constrained problems. A good complement to Corollary 4.3 is Corollary 2 in [33], which imposes very restrictive conditions on the existence of touchpoints in problems of order 1.

As for convergence results relating to direct transcription methods, all the ones we were able to find are limited to problems discretized using Euler's method. Euler's method is a first order integrator that discretizes $x' = f(x,y)$ by

$$x_{i+1} = x_i + h_i f(x_i, y_i).$$

In [13], Dontchev and Hager prove a convergence result for problems with inequality constraints on the state only under assumptions similar to the ones they made for unconstrained problems in results presented in Chapter 2. The optimality conditions they use do not require a guess at the problem structure and are very different from the conditions in Theorem 4.2. We rewrite their result to fit a problem of the form (4.1) with $g = g(x)$ and $\phi = 0$.

For the Hamiltonian given by

$$H(x, y, \lambda, \nu) = \lambda^T f(x, y) - \nu^T \nabla g(x) f(x, y) \tag{4.5}$$

define the matrices $A(t), B(t), Q_1(t), Q_2(t), Q_3(t)$ as in equation (2.3) and define $K(t)$ by $K = \nabla g(x^*(t))$. In addition to coercivity (see Definition 2.2), impose the following form of the **smoothness condition** (see Definition 2.1):

Definition 4.4. *The problem (4.1) is said to satisfy the* ***smoothness condition*** *if it has a local solution (x^*, y^*) which lies in $W^{2,\infty} \times W^{1,\infty}$ and there exists an open set $\Omega \subset \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ and $\rho > 0$ such that $B_\rho(x^*(t), y^*(t)) \subset \Omega$ for every $t \in [t_0, t_f]$ and the first two derivatives of f as well as the first three derivatives of g are Lipschitz continuous in Ω and the first two derivatives of C are Lipschitz continuous in $B_\rho(x^*(t_f))$. Finally, there exist $\lambda^* \in W^{2,\infty}$ and $\nu^* \in W^{1,\infty}$ such that x^*, y^*, λ^* and*

ν^* satisfy the first order optimality conditions of the form

$$
\begin{aligned}
x' &= f(x,y) && \text{(4.6a)}\\
\lambda' &= -\nabla_x H(x,y,\lambda,\nu) && \text{(4.6b)}\\
0 &= \nabla_y H(x,y,\lambda,\nu) && \text{(4.6c)}\\
g(x,y) &\in \mathcal{N}(\nu) && \text{(4.6d)}\\
x(t_0) &= \zeta && \text{(4.6e)}\\
\lambda(t_f) &= 0 && \text{(4.6f)}\\
\nu &\leq 0 && \text{(4.6g)}\\
\nu' &\geq 0 && \text{(4.6h)}
\end{aligned}
$$

where $\mathcal{N}(\nu)$ is a set of functions such that $y(t) \leq 0$ and $\nu'(t)^T y(t) = 0$ almost everywhere in $[t_0, t_f]$ and $\nu(t_f)^T y(t_f) = 0$.

Dontchev and Hager also impose the condition of **independence at $\mathcal{A}$**: $\mathcal{A}(t)$

Definition 4.5. *Let*

$$\mathcal{A}(t) = \{j \in \{1, ..., c\} : g_j(x^*(t)) = 0\} \tag{4.7}$$

*be the set of constraints active at time t. We say that a problem of the form (4.1) satisfies **independence at $\mathcal{A}$** if $\mathcal{A}(0)$ is empty and there exists a scalar $\beta > 0$ such that*

$$\left| \sum_{j \in \mathcal{A}(t)} v_j K_j(t) B(t) \right| \geq \beta \left| v_{\mathcal{A}(t)} \right|$$

for any $v \in \mathbb{R}^k$ and for every $t \in [t_0, t_f]$ where $\mathcal{A}(t)$ is non-empty.

The result is

Theorem 4.6. *If Smoothness, Independence at $\mathcal{A}$ and Coercivity hold, then for all sufficiently small h, there exists a local solution (x, y) of the Euler discretization of the problem (4.1) with g a function of x only and no final state constraints and* *associated Lagrange multipliers* (λ, ν) *such that*

$$||x - x^*||_{H^1} + ||y - y^*||_{L^2} + ||\lambda - \lambda^*||_{H^1} + ||\nu - \nu^*||_{L^2} \leq ch$$

and

$$||x - x^*||_{W^{1,\infty}} + ||y - y^*||_{L^\infty} + ||\lambda - \lambda^*||_{w^{1,\infty}} + ||\nu - \nu^*||_{L^\infty} \leq ch^{2/3}.$$

where $(x^*, y^*, \lambda^*, \nu^*)$ *are evaluated at the gridpoints,* $W^{k,\infty}$ *is the norm defined by equation 2.2 and*

$$||z||_{H^1} = \sqrt{||z||^2_{L^2} + ||z||^2_{L^2}}. \tag{4.8}$$

Moreover, $\bar{x}, \bar{\lambda}, y, \nu$ *are Lipschitz continuous in time with a Lipschitz constant independent of* h *where*

$$\bar{x}_k = \frac{x_{k+1} - x_k}{h}, \quad \bar{\lambda}_k = \frac{\lambda_{k+1} - \lambda_k}{h}.$$

Malanowski, Buskens and Maurer prove the most general result in [29], showing first order convergence of differential, algebraic and adjoint variables (states, controls and multipliers) for problems with combined control and state inequality constraints. We rewrite it here to fit problems of the form (4.1). Assume

B1 The functions C, f, ϕ and g are twice Frechet differentiable in all arguments, and the respective derivatives are locally Lipschitz continuous in x, y.

B2 There exists a local solution x^*, y^*.

and define

$$A(t) = \nabla_x f(x^*(t), y^*(t)), \quad B(t) = \nabla_y f(x^*(t), y^*(t)), \tag{4.9}$$

$$K^x(t) = \nabla_x g(x^*(t), y^*(t)), \quad K^y(t) = \nabla_y g(x^*(t), y^*(t)), \quad E(t) = \nabla \phi(x^*(t_f)). \tag{4.10}$$

Malanowski et al define $\mathcal{A}$ as in Equation (4.7) and let $\Theta^x(t), \Theta^y(t)$ be the matrices whose rows are made up of the rows of $K^x(t), K^y(t)$ respectively that are in $\mathcal{A}(t)$. They then make two additional assumptions:

B3 There exists $\beta > 0$ such that for almost every $t \in [t_0, t_f]$, $\|\Theta^y(t)^T z\| \geq \beta \|z\|$ for all $z \in \mathbb{R}^{card(\mathcal{A}(t))}$.

B4 For any $e \in \mathbb{R}^d$ the following BVP has a solution (is completely output controllable):

$$\begin{aligned} x' &= \tilde{A}x + \tilde{B}y \\ x(t_0) &= 0 \\ Ex(t_f) &= e \end{aligned}$$

where

$$\begin{aligned} \tilde{A} &= A(t) - B(t)\Theta^y(t)^T(\Theta^y(t)\Theta^y(t)^T)^{-1}\Theta^x(t), \\ \tilde{B} &= B(t)(I - \Theta^y(t)^T(\Theta^y(t)\Theta^y(t)^T)^{-1}\Theta^x(t)). \end{aligned}$$

The Hamiltonian defined by Malanowski et al is different from the one used by Dontchev and Hager (4.5). In fact it is equivalent to the one used by Jacobson et al (4.2):

$$\mathcal{H} = \lambda^T f(x, y) + \nu^T g(x, y)$$

and the optimality conditions they propose are

$$
\begin{aligned}
x' &= f(x,y) && (4.11a)\\
\lambda' &= -\nabla_x \mathcal{H}(x,y,\lambda,\nu) && (4.11b)\\
0 &= \nabla_y \mathcal{H}(x,y,\lambda,\nu) && (4.11c)\\
\lambda(t_f) &= \nabla C(x^*(t_f)) - \nabla_x(\pi^T \phi(x^*(t_f))) && (4.11d)\\
\lambda(t_0) &= -\mu. && (4.11e)
\end{aligned}
$$

Next, define

$$\tilde{\mathcal{A}}(t) = \{j \in \mathcal{A}(t) : \nu^*(t) > 0\}$$

and let $\tilde{\Theta}^x(t), \tilde{\Theta}^y(t)$ be the matrices whose rows are made up of the rows of $K^x(t), K^y(t)$ respectively that are in $\tilde{\mathcal{A}}(t)$. Letting $w^* = (x^*, y^*, \lambda^*, \nu^*)$, the authors impose the following two 2nd order conditions on the problem

B5 There exists $\gamma > 0$ such that for all $t \in [t_0, t_f]$

$$v^T \nabla^2_{yy} \mathcal{H}(w^*(t)) v \geq \gamma \|v\|$$

for all $v \in \mathbb{R}^{m_2}$ satisfying $\tilde{\Theta}^y(t)v = 0$.

B6 The Ricatti equation

$$
\begin{aligned}
Q' &= -QA - A^TQ - \nabla^2_{xx}\mathcal{H}(w^*) + \left(\begin{bmatrix} \nabla^2_{yx}\mathcal{H}(w^*) \\ \tilde{\Theta}^x \end{bmatrix}^T + Q \begin{bmatrix} B^T \\ 0 \end{bmatrix}^T \right) \times \\
&\times \begin{bmatrix} \nabla^2_{yy}\mathcal{H}(w^*) & \left(\tilde{\Theta}^y\right)^T \\ \tilde{\Theta}^y & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} \nabla^2_{yx}\mathcal{H}(w^*) \\ \tilde{\Theta}^x \end{bmatrix} + \begin{bmatrix} B^T \\ 0 \end{bmatrix} Q \right)
\end{aligned}
$$

has a solution Q that satisfies

$$x^T \left(\nabla^2 C(x^*(t_f)) + \nabla^2_{xx}(\pi^T \phi(x^*(t_f))) - Q(t_f) \right) x \geq 0$$

for all $x \in \mathbb{R}^{m_1}$ that satisfies $Ex = 0$

The final result takes the form

Theorem 4.7. *(Adapted from [29], Theorem 5.7) If assumptions (B1)–(B6) are satisfied then for all sufficiently small* h *there exist locally unique* $x^h, y^h, \lambda^h, \nu^h, \pi^h$ *and* μ^h *which satisfy the optimality conditions for the Euler discretization of problem (4.1) given by* $\nabla H^h = 0$ *where*

$$H^h = C(x_N) + \pi^T\phi(x_N) + \mu^T(x_0 - \zeta) + \sum_{i=0}^{N-1}\left(\lambda_{i+1}^T(-x_{i+1} + x_i + hf(x_i, y_i)) + h\nu_i^T g(x_i, y_i)\right)$$

such that

$$\begin{aligned} \|x^* - x^h\|_1 + \|\lambda^* - \lambda^h\|_1 &\leq ch \\ \|x^* - x^h\|_\infty + \|y^* - y^h\|_\infty + \|\lambda^* - \lambda^h\|_\infty + \|\nu^* - \nu^h\|_\infty &\leq ch \\ |\pi^* - \pi^h| + |\mu^* - \mu^h| &\leq ch. \end{aligned}$$

Notice the similarities between "independence at $\mathcal{A}$" (Definition 4.5) and (B3), coercivity as defined by Hager (Definition 2.2) and (B5), the 2nd order conditions for unconstrained problems written in terms of Ricatti equations in [28] and [11] and (B6).

We will not be presenting any new convergence results in the following sections. However, we will demonstrate how the fact that they do not require the formulation of optimality conditions makes direct transcription methods very well suited for certain inequality constrained problems [25]. For the problems in the next section, a boundary arc in the solution is theoretically impossible due to the result in [24], but the cost structure forces the solution very close to the constraint boundary. This results in a large number of closely spaced touchpoints with very small deviations from the constraint in between. We will show that the formulation of optimality conditions for these problems may be impossible, and even where it is possible, the resulting boundary value problem is extremely ill-conditioned. Direct transcription methods, on the other hand, can solve these problems to a high degree of accuracy as our numerical results will demonstrate.

In Section 4.3, we briefly discuss open questions and partial results related to another important issue, initialization. Finding an initial feasible solution is necessary to initialize any numerical method for solving optimal control problems, but this becomes a non-trivial task when inequality constraints are involved.

4.2 Virtual Boundary Arcs

4.2.1 The Heat Equation Problem

Consider an insulated metal rod of length π meters that is heated at both ends for a period of 5 hours. Suppose that we are able to apply heat in precisely measured quantities at each end and suppose that we want to minimize the total heat output along the rod while keeping the temperature at a point x meters from the left end of the rod at time t hours higher than $g(x,t)$.

Let $u(x,t)$ be the temperature at point x at time t and let v_0, v_π be the heat applied at each end. Then the problem can be represented mathematically as

$$\min J(u, v_0, v_\pi) = \int_0^\pi \int_0^5 u(x,t)^2 dtdx + \int_0^5 q_1 v_0^2(t) + q_2 v_\pi^2(t)\ dt \tag{4.12a}$$

$$\begin{aligned} u_t &= u_{xx} && (4.12\text{b}) \\ u(0,x) &= u_0(x) && (4.12\text{c}) \\ u(t,0) &= v_0(t) && (4.12\text{d}) \\ u(t,\pi) &= v_\pi(t) && (4.12\text{e}) \\ u(x,t) &\geq g(x,t). && (4.12\text{f}) \end{aligned}$$

We will approximate the PDE control problem (4.12a)–(4.14) by an ODE problem by partitioning the rod into N equal intervals. The values of N we will discuss here would be much too small if our goal were to solve the PDE problem as accurately as possible. However, our primary interest in this problem is that this approximation gives rise to a family of similarly structures ODE control problems with constraints whose order depends on N and can be arbitrarily high. We will show that SOCS is able to solve these problem for values of N that result in a high order. As for the original PDE problem, it is interesting to note that the solution changes very little for $N > 10$ (see Figure 4.2.)

We let $x_i = i\frac{\pi}{N}$ and let $u_i(t)$ be an approximation to $u(t, x_i)$ for $i = 1, \ldots N-1$. We let $\delta = \frac{\pi}{N}$, assume $u_0 = v_0$ and $u_N = v_\pi$ with u_0, u_N the new controls and approximate u_{xx} using centered differences. Finally, we approximate the x integration in the original cost by the Trapezoid method, obtaining the following problem

$$\min \frac{\delta}{2}\left(\int_0^5 u_0^2(t) + u_N^2(t) + \sum_{i=1}^{N-1} 2u_i(t)^2 \; dt\right) + \int_0^5 q_1 u_0^2(t) + q_2 u_N^2(t) dt \tag{4.13a}$$

$$\begin{aligned}
u_1' &= \frac{1}{\delta^2}(u_2 - 2u_1 + u_0) && \text{(4.13b)}\\
u_i' &= \frac{1}{\delta^2}(u_{i+1} - 2u_i + u_{i-1}), \quad u_i(0) = u(0, x_i), \quad i = 2, ..., N-2 && \text{(4.13c)}\\
u_{N-1}' &= \frac{1}{\delta^2}(u_N - 2u_{N-1} + u_{N-2}) && \text{(4.13d)}\\
0 &\geq g_i = g(x_i, t) - u_i(t), \quad i = 0, ..., N. && \text{(4.13e)}
\end{aligned}$$

We take g to be

$$g(x, t, a, c) = c \sin x \sin\left(\frac{\pi t}{5}\right) - a. \tag{4.14}$$

If the values of a, c are clear from the discussion, we shall omit them from the notation.

We also take $q_1 = q_2 = 10^{-3}$ which is typical of the situation where the control weight is really for numerical regularization. The symmetry in the cost and dynamics when $q_1 = q_2$ means that the optimal controls u_0 and u_N are equal. We will speak of the optimal control and only plot one of them in the discussion that follows. We also take the initial temperature profile to be zero, $u(0, x) = 0$.

4.2.2 Numerical Results

Fixing $c = 1, a = 0.7$, we solve this problem numerically using SOCS for different values of N. Figure 4.1 displays the profile for $N = 10$ and Figure 4.2 shows the control for $N = 10, 31$. The pictures vary very little once $N > 5$.

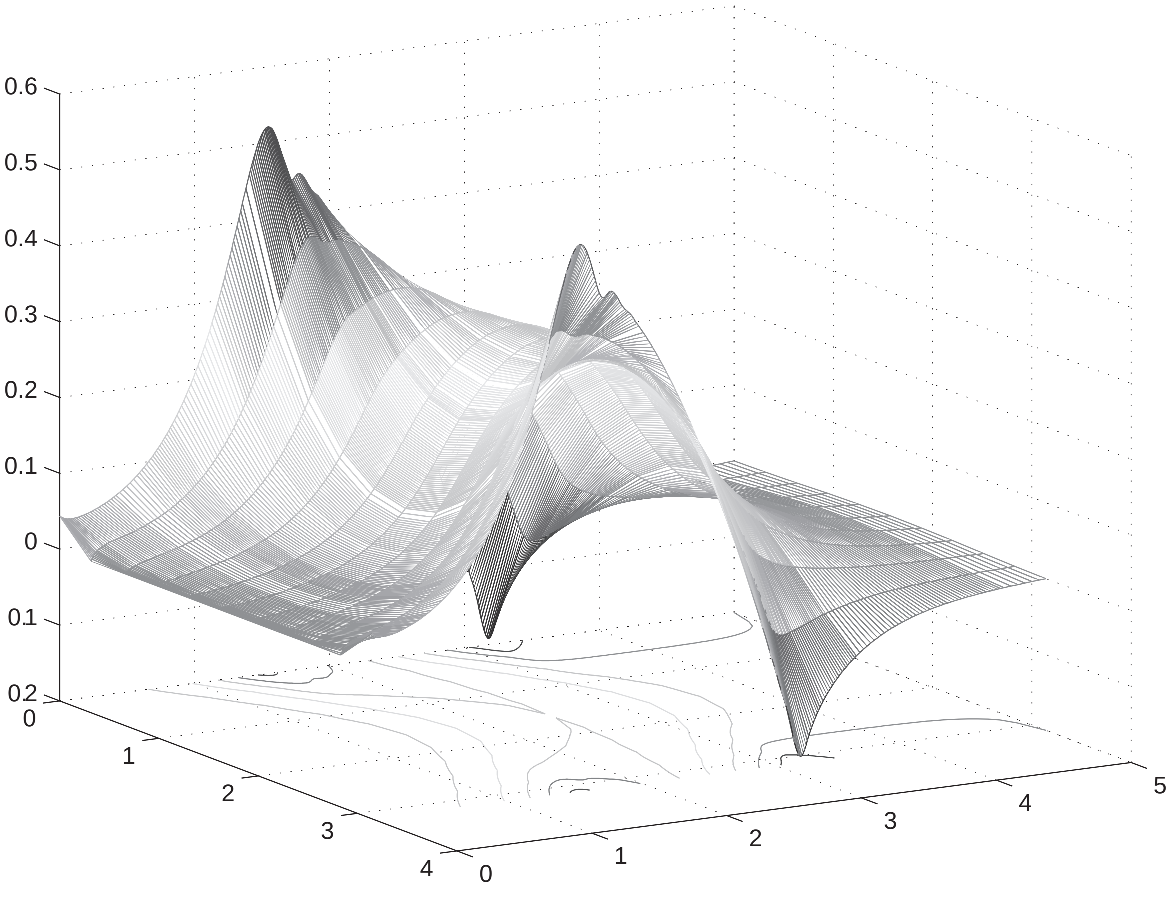


Figure 4.1: Optimal state $u(x, t)$ for problem (4.13) with $N = 10$.

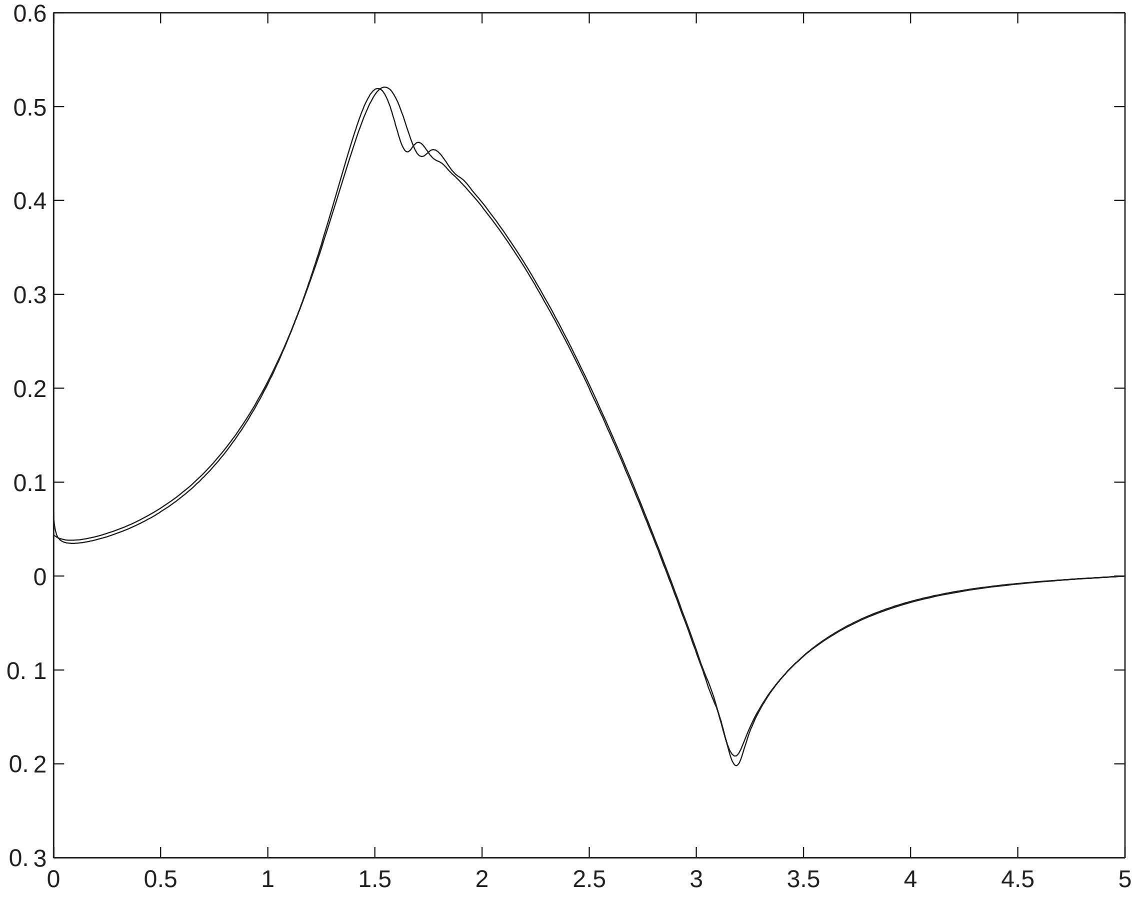


Figure 4.2: Optimal control for problem (4.13) with $N = 10$ and $N = 31$.

We are interested in the relationship between the solution profile and the constraints. Our numerical experiments show that, when N is even, the only constraint active anywhere on the time interval is $g_{N/2}$. Similarly, when N is odd, the only active constraints are $g_{(N-1)/2}$ and $g_{(N+1)/2}$. So the problem is equivalent to a problem with a single constraint of order $N/2$ (or two constraints of order $(N-1)/2$). Figure 4.3 shows $g(x_{N/2}, t)$ and $u_{N/2}(t)$ for the case $N = 10$ where $N/2 = 5$ is odd and greater than 1. However, the figure appears to show a smooth transition onto the constraint surface, riding the surface (a boundary arc), and then a departure. Does this contradict the result in [24]?

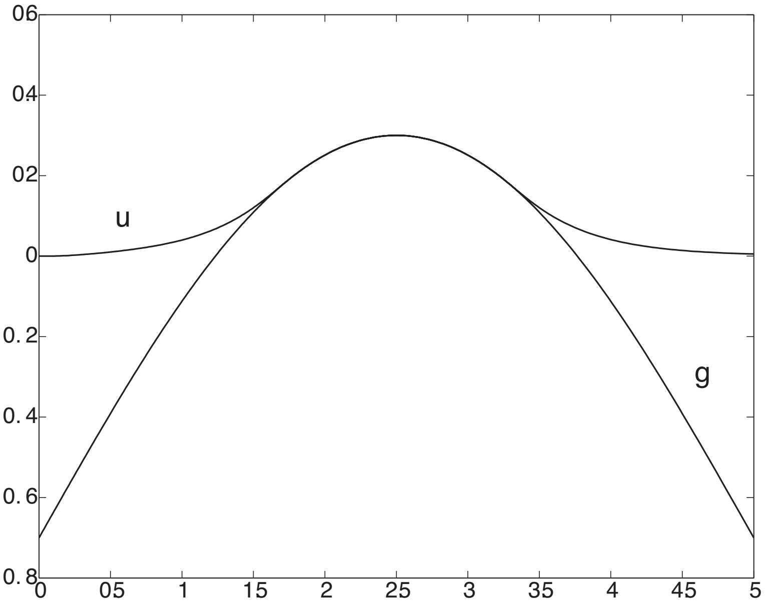


Figure 4.3: $u_{N/2}(t)$ and $g(x_{N/2}, t)$ for problem (4.13) for $N = 10$.

If we look more closely, it turns out that the results are not that simple. Figure 4.4 shows the central section of $g_{N/2}$ for $N = 10, 31$. The graph for $N = 31$ has been displaced upward to make comparison easier. Both graphs show spikes at either end. They are rather small, of height only about 10^{-6}, but this value is well above floating point error, which is 10^{-8}. In the middle, we see many smaller spikes and notice that the two curves look like noisy copies of the same curve. To get an even better idea of the size of the constraint deviations, we rescale the central portion of $g_{N/2}$. Let $w(v) = sign(v)|v|^{1/6}$. The curve in Figure 4.5 is $w(g_{N/2}$ for $N = 10$ and the horizontal lines are $w(10^{-k})$ where $k = 5, 6, 7, 8, 9$.

To understand this phenomenon even better, let us modify the constraint. We keep $N = 10, c = 1$ but allow a to increase from its default value of 0.7. That is, we lower the constraint surface. When a reaches 1, the solution is just $u = 0$ and the cost is zero. Figures 4.6 and 4.7 show $g_{N/2}$ for $a = 0.7, 0.9, 0.9$ and $a = 0.9, 0.925, 0.95$ respectively. We notice that the interval where $g_{N/2}$ is very small is shrinking as expected, but the pattern of oscillations remains the same. As the constraint is lowered even further, the smaller oscillations disappear, and only three, two, then one touchpoint remain.

Interestingly enough, others have tried to solve this problem (with the original parameter values) using an indirect method, and failed. In [4], index reduction was used to reformulate a set of optimality conditions for the problem (4.13) as a low-index boundary value problem. The authors assumed a boundary arc in the solution, and the boundary value problem, formulated for different values of N, all even, was passed to SOCS. Solutions were obtained for $N = 4$, but not for any $N > 4$. Note that when $N = 4$, the order of the problem appears to be 2 which is even, but when $N = 6$, the order is 3 which is odd and greater than 1.

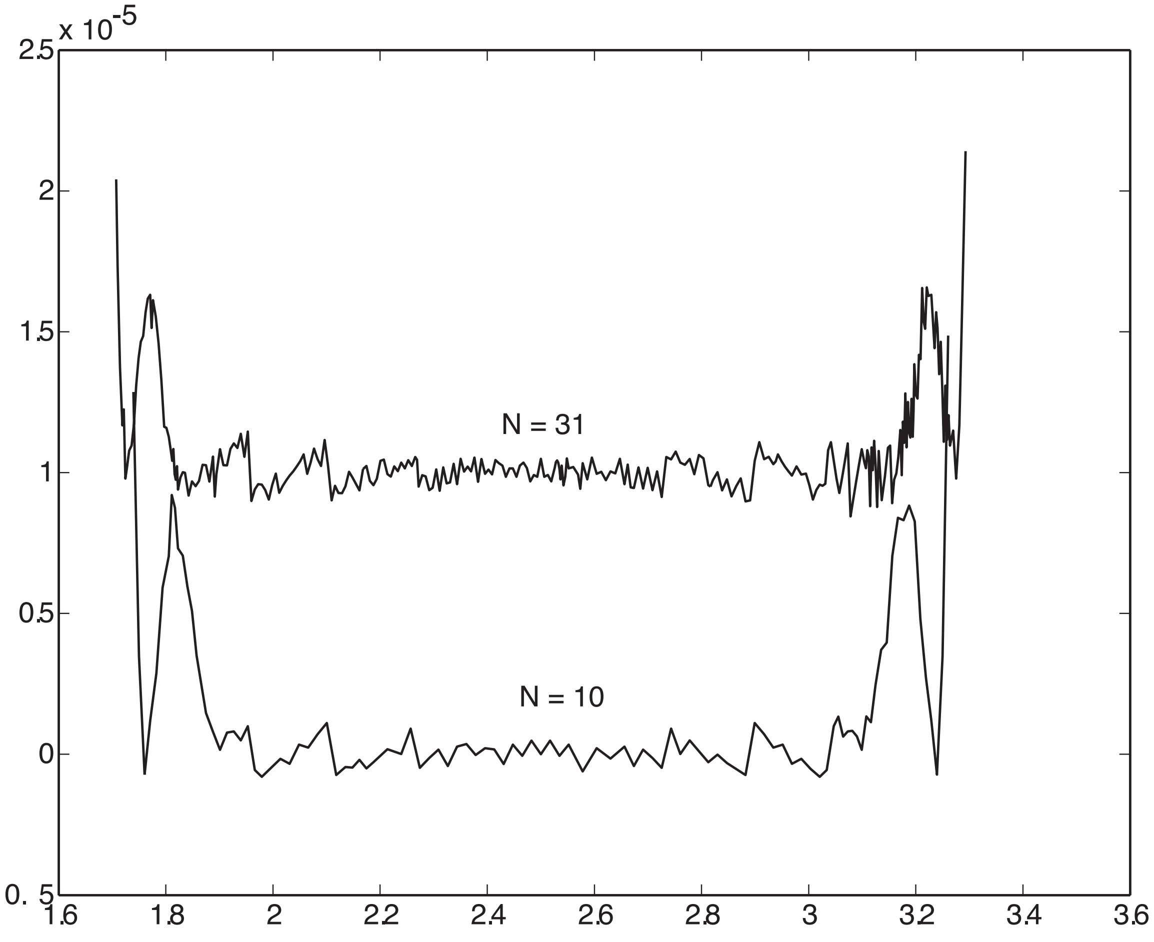


Figure 4.4: Constraint deviation for problem (4.13) for $N = 10, 31$.

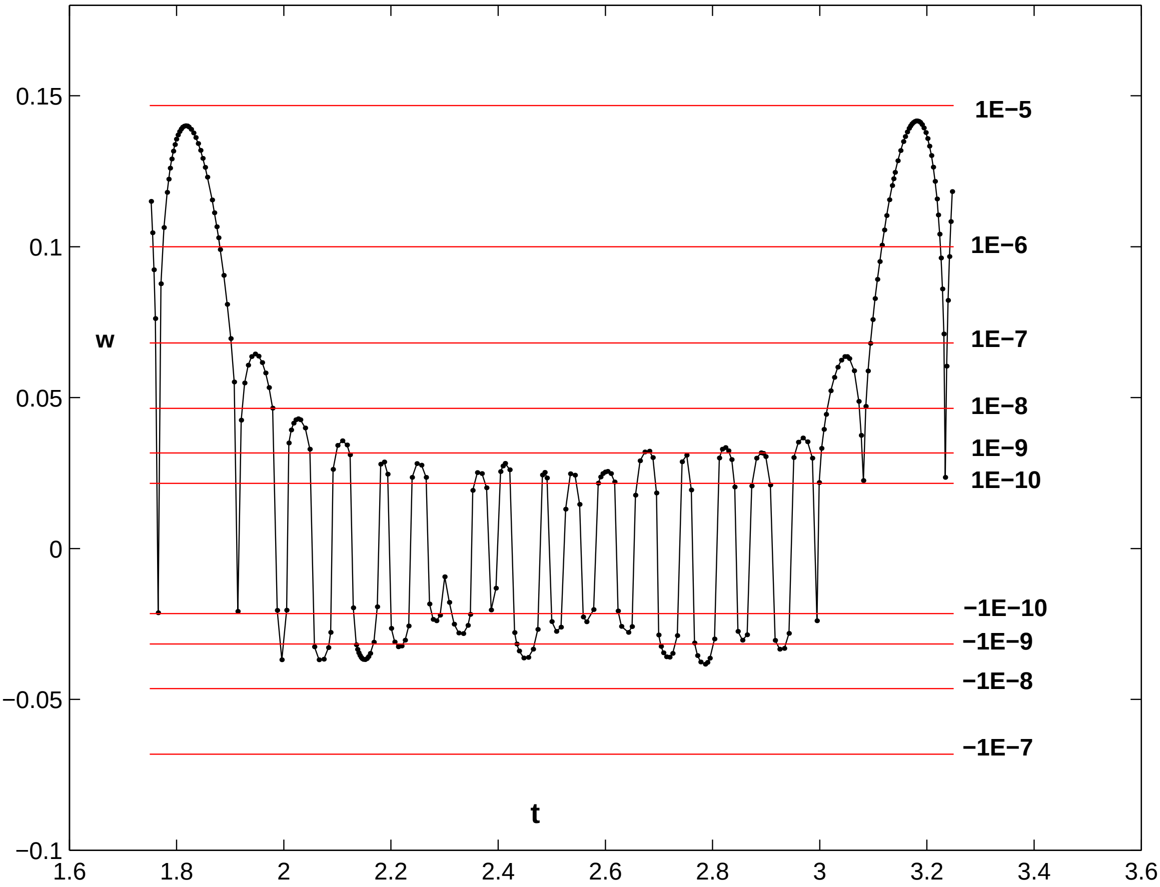


Figure 4.5: Rescaled constraint deviation for problem (4.13), for $N = 10$.

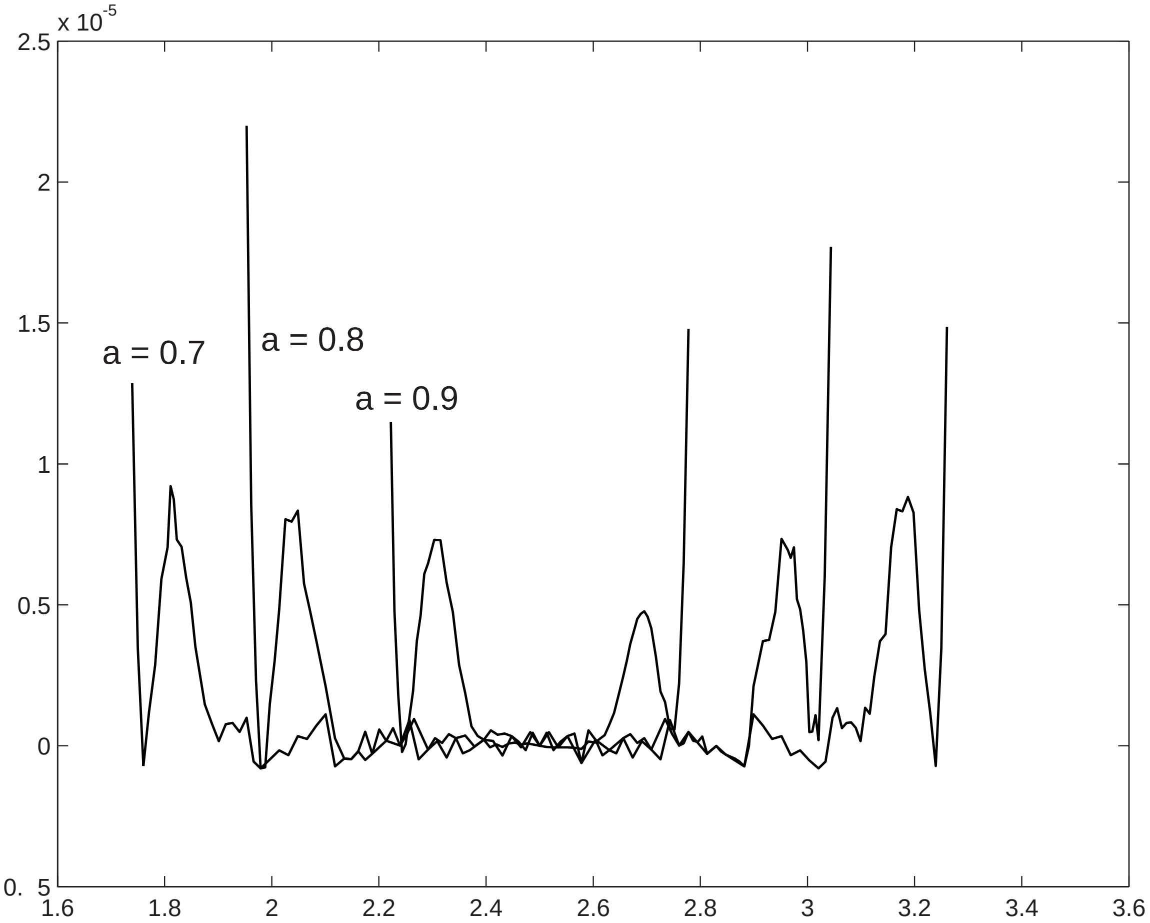
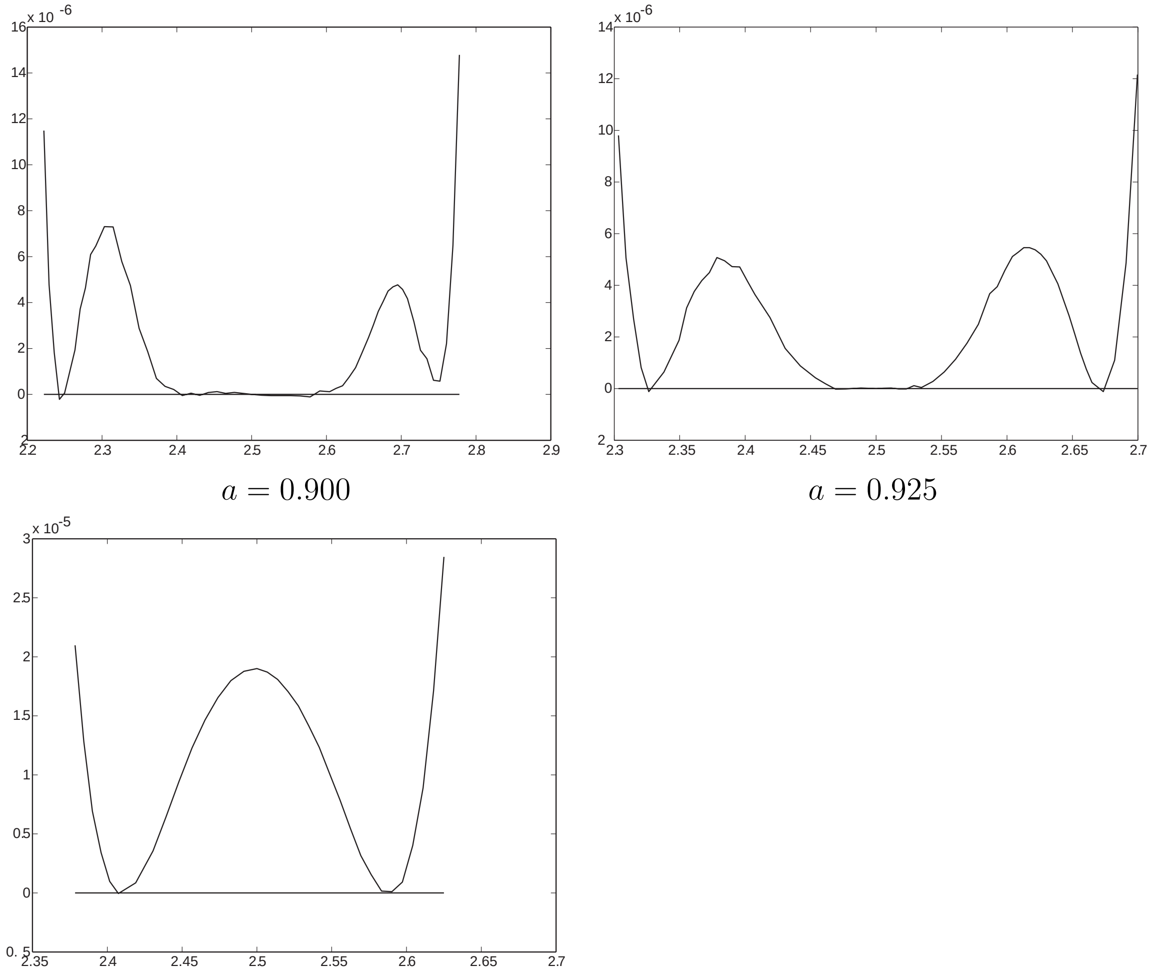


Figure 4.6: Constraint deviation for problem (4.13) for $N = 10, a = 0.7, 0.8, 0.9$.



$a = 0.900$

$a = 0.925$

$a = 0.950$

Figure 4.7: Constraint deviation for problem (4.13), for $N = 10, a = 0.9, 0.925, 0.95$.

4.2.3 Discussion

What is the significance of these numerical results, and how do they match up with existing theoretical and numerical results?

First of all, looking at Figures 4.1, 4.2 and 4.3, we see that a direct transcription algorithm (SOCS) appears to be successful in solving the problem (4.13a)–(4.13), arriving, within a very reasonable amount of time, at a solution that "makes sense" as an approximation to the optimal solution to the original PDE problem (4.12a)–(4.12f), even though we don't know the optimal solution to this PDE problem.

We also notice that the problem is equivalent to a problem with a single constraint of order $N/2$ or two constraints of order $(N-1)/2$ depending on whether N is odd or even. Using this observation and the symmetry in the problem when $q_1 = q_2 = q$, we can rewrite it in the form given in [24]. For N even, it is

$$\min u_{N/2+1} \tag{4.15a}$$

$$\begin{aligned}
u_1' &= \frac{1}{\delta^2}(u_2 - 2u_1 + u_0) && (4.15b)\\
u_i' &= \frac{1}{\delta^2}(u_{i+1} - 2u_i + u_{i-1}), \quad i = 2, ...N/2-1 && (4.15c)\\
u_{N/2}' &= \frac{2}{\delta^2}(u_{N/2-1} + u_{N/2}) && (4.15d)\\
u_{N/2+1} &= (1 + \frac{2q}{\delta})u_0^2(t) + 2u_{N/2}^2 + 4\sum_{i=1}^{N/2-1} u_i(t)^2 && (4.15e)\\
0 &\geq g(\pi/2, t) - u_{N/2}(t). && (4.15f)
\end{aligned}$$

This problem is in Mayer form as required by (A1), has a scalar control u_0 and a scalar constraint (4.15f) as required by (A2), which is a function of $u_{N/2}$ only and is order $N/2$ (A3). The right hand side of the differential equations (4.15b)–(4.15d) is a simple linear function, continuously differentiable $N/2+1$ times in u_0 as well as $u_1, u_2, ...u_{N/2}$ as required by (A5).

For g given by (4.14), the constraint (4.15f) takes the form

$$\bar{g}(u,t) = \sin\left(\frac{\pi t}{5}\right) - 0.7 - u_{N/2}(t).$$

Since the sine function is infinitely differentiable, we know that $\bar{g}(N-1)$ exists as required by (A7). On the other hand $\bar{g}^{(N/2)} = \sin^{(N/2)}\left(\frac{\pi t}{5}\right) - u_2 + 2u_1 - u_0$ and $\nabla_{u_0}\bar{g}^{(N/2)} = -1 \neq 0$ as required by (A9). Given the convexity of f, (A6) is safe to assume by coercivity (see Definition 2.2 and discussion in Section 2.4.1). (A4) and (A8) are the hardest to verify but notice that the control is Figure 4.2 appears smooth.

So assuming that this problem satisfies all the assumptions in Theorem 4.3 we know that the solution cannot have a boundary arc when $N > 4$ and $N/2$ is odd. Thus the result in [4] makes sense. SOCS could not find a solution to the optimality conditions assuming a boundary arc because a solution did not exist.

Finally, notice that the solution near the constraint boundary becomes better behaved as the constraint is lowered (Figures 4.6 and 4.7). By lowering the constraint, the cost of staying above the boundary is reduced. Intuitively, when the cost structure forces the solution close to a boundary arc that is theoretically impossible, the BVP formed by the optimality conditions becomes very ill-conditioned, making the problem unsolvable by indirect methods. However, a direct transcription method is able to resolve the problem to a high degree of accuracy by introducing small oscillations which grow smaller in size and greater in number as the cost of staying away from the boundary is increased. In essence, direct transcription is able to regularize the problem by ignoring ill-conditioned parts of the optimality conditions.

Conditioning of the heat equation problem was studied in some detail in [25]. However, with a problem as large as this, analysis can only be carried so far. We will now present a smaller example which exhibits behavior very similar to that of the heat equation problem. Because this is a smaller problem with fixed order, we are able to do some theoretical calculations and demonstrate the effects of the ill-conditioning on our ability to solve a problem with a "virtual boundary arc". We are also able to produce some interesting graphs that, even more clearly than Figures 4.6 and 4.7, demonstrate what happens as the solution is forced closer and closer to the constraint.

4.2.4 Example Problem

Consider the problem

$$\overline{J} = \min_v \ \frac{1}{2}\int_0^1 \rho(x_1-1)^2 + u^2\,dt \tag{4.16a}$$

$$\begin{aligned} x_1' &= x_2, \quad x_1(0) = 0 = x_1(1) && (4.16b)\\ x_2' &= x_3, \quad x_2(0) = 1 = -x_2(1) && (4.16c)\\ x_3' &= u, \quad x_3(0) = 2 = x_3(1) && (4.16d)\\ x_1(t) &\le L && (4.16e) \end{aligned}$$

where $\rho \geq 0$ and $1 \geq L > 0$.

For the case $\rho = 0$, the problem (4.16) is studied in [24], where it is demonstrated that the constraint is 3rd order and exact optimal solutions are found for ranges of L values.

We solve the problem with SOCS for $L = 0.134$ and different values of ρ. For $\rho = 0$ and 10^4 the solution appears to have two touchpoints (Figure 4.8.)

As ρ approaches 5×10^4, the solution starts to exhibit an apparent boundary arc. To examine the solution behavior near the constraint more closely, we use expansion functions

$$\mathcal{F}_K(x_1) = 0.134\left(\frac{x_1}{0.134}\right)^K \tag{4.17}$$

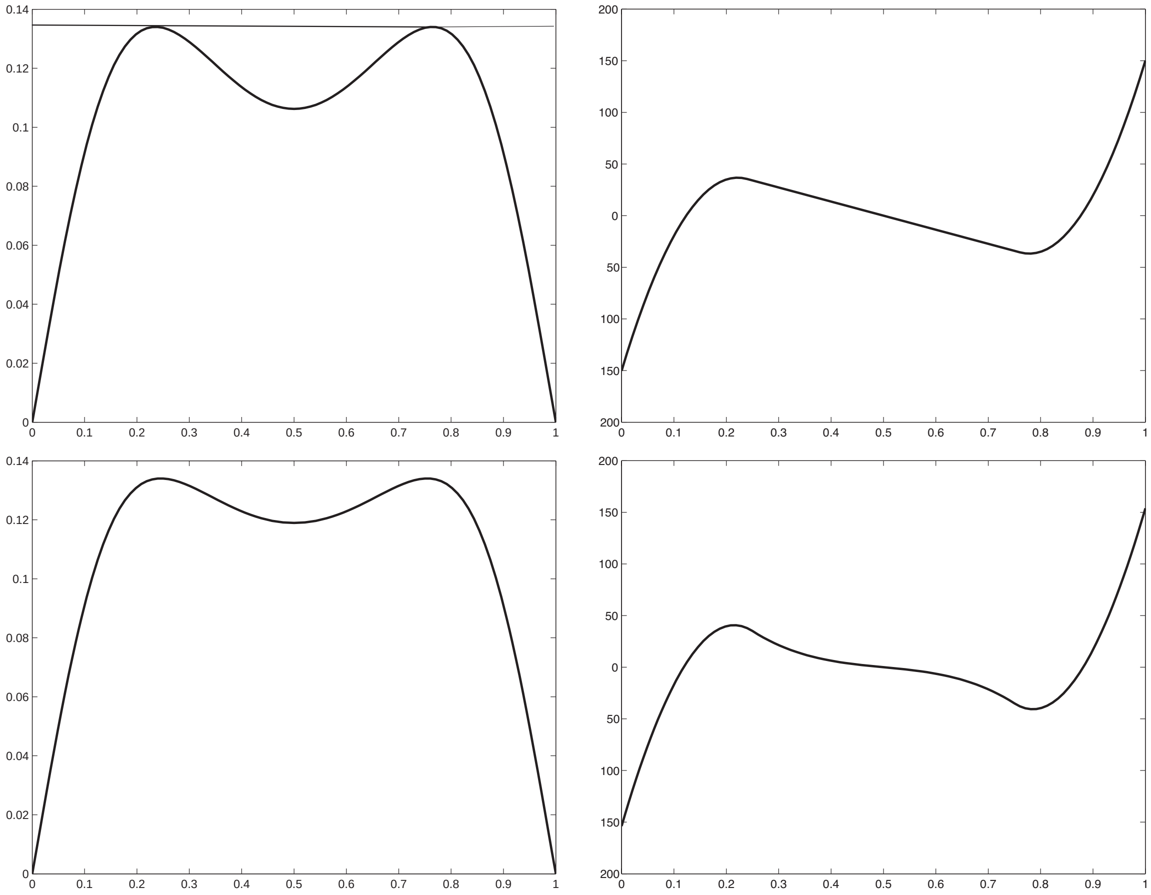


Figure 4.8: x_1 (left) and control u (right) for $\rho = 0$ (top) and $\rho = 10000$ (bottom.)

which compress the graph away from 0.134 and greatly expand it near 0.134. The amount of compression and expansion increases with K. Applying $\mathcal{F}$ with $K = 5 \times 10^2, 5 \times 10^4, 5 \times 10^4, 2 \times 10^7$ to the problem with $\rho = 5 \times 10^4, 10^5, 1.5 \times 10^5, 3 \times 10^5$ respectively we see the number of touchpoints increase from two to three to four to five to at least seven (Figures 4.9–4.12). It is interesting to note that while $\overline{J}$ increases with H, $\int_0^1 (x_1(t) - 1)^2 dt$ stays almost constant around 0.798.

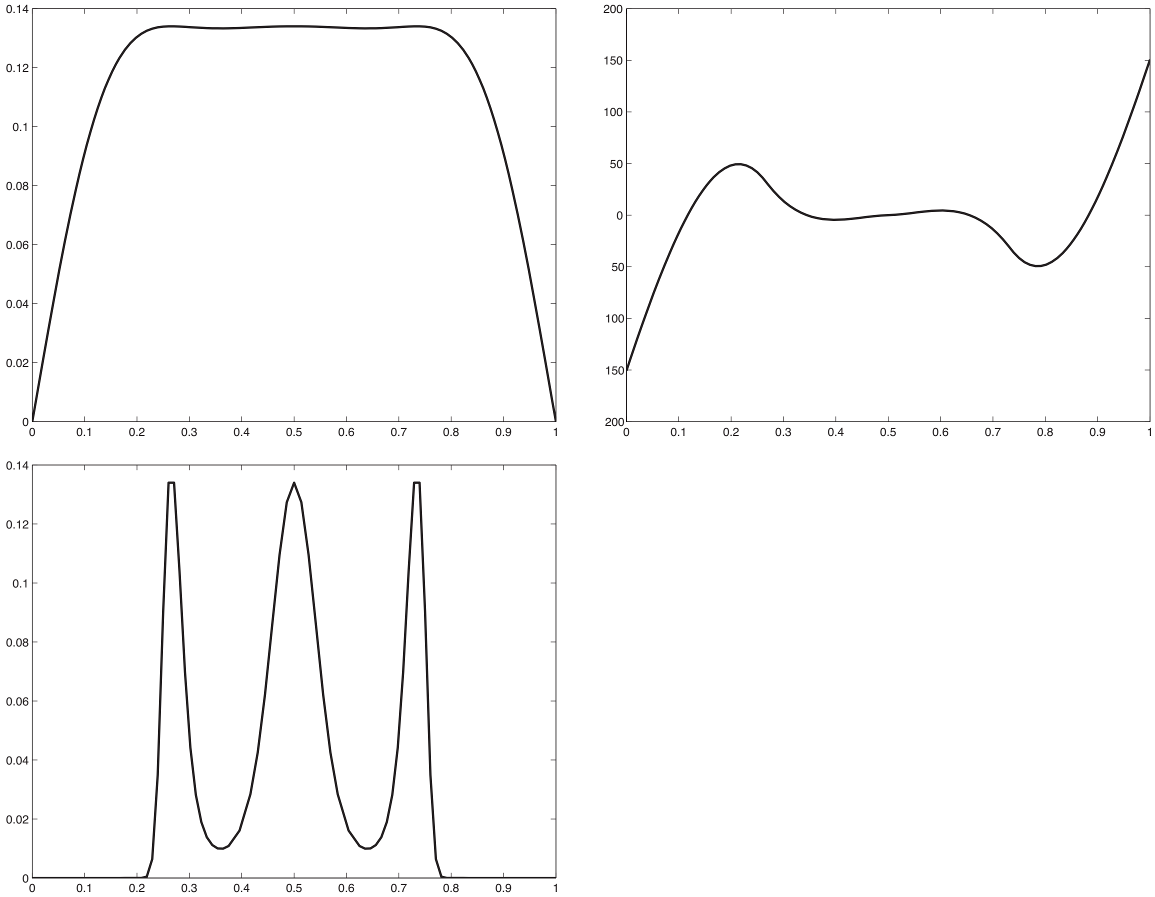


Figure 4.9: Example problem with $\rho = 5 \times 10^4$, plots of x_1 (top left), control u (top right), and $\mathcal{F}_{500}(x_1)$ (bottom.) Number of touchpoints: 3.

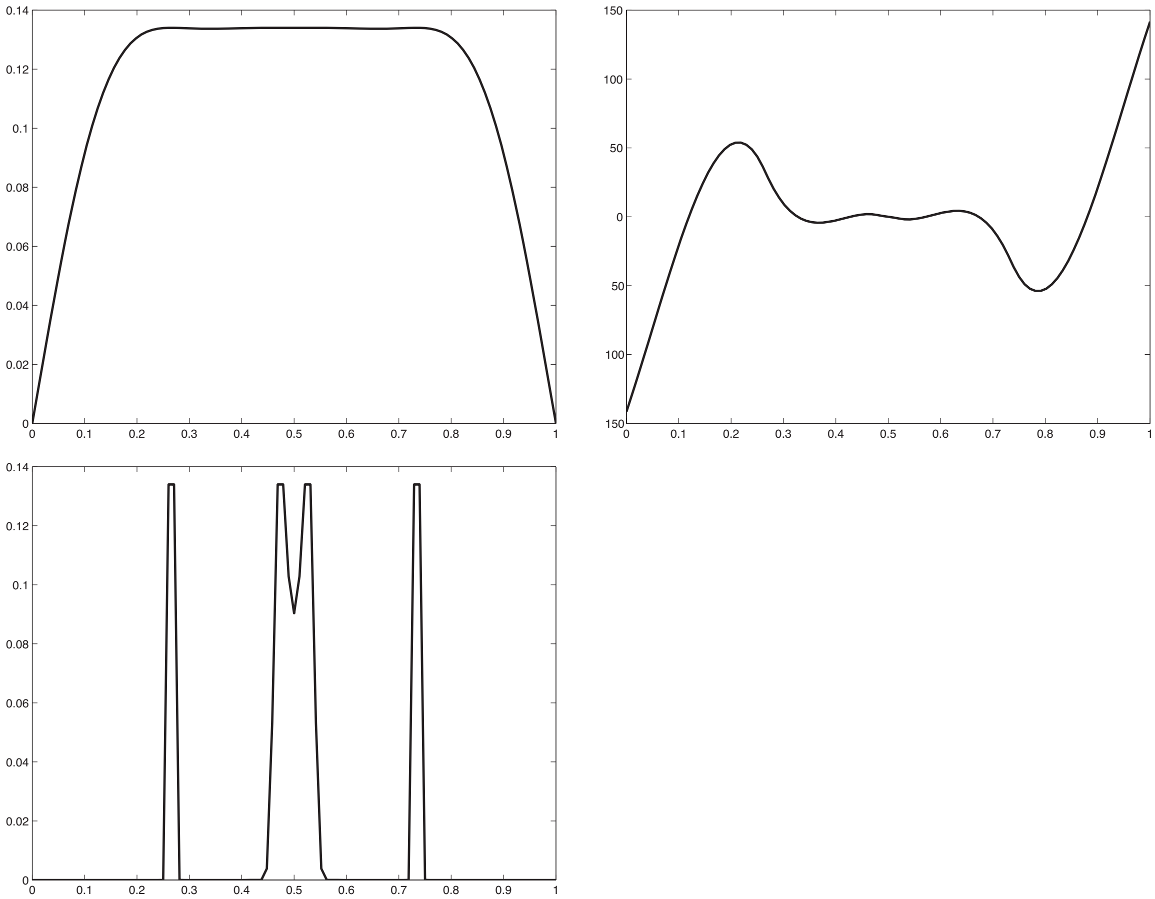


Figure 4.10: Example problem with $\rho = 10^5$, plots of x_1 (top left), control u (top right), and $\mathcal{F}_{50000}(x_1)$ (bottom.) Number of touchpoints: 4.

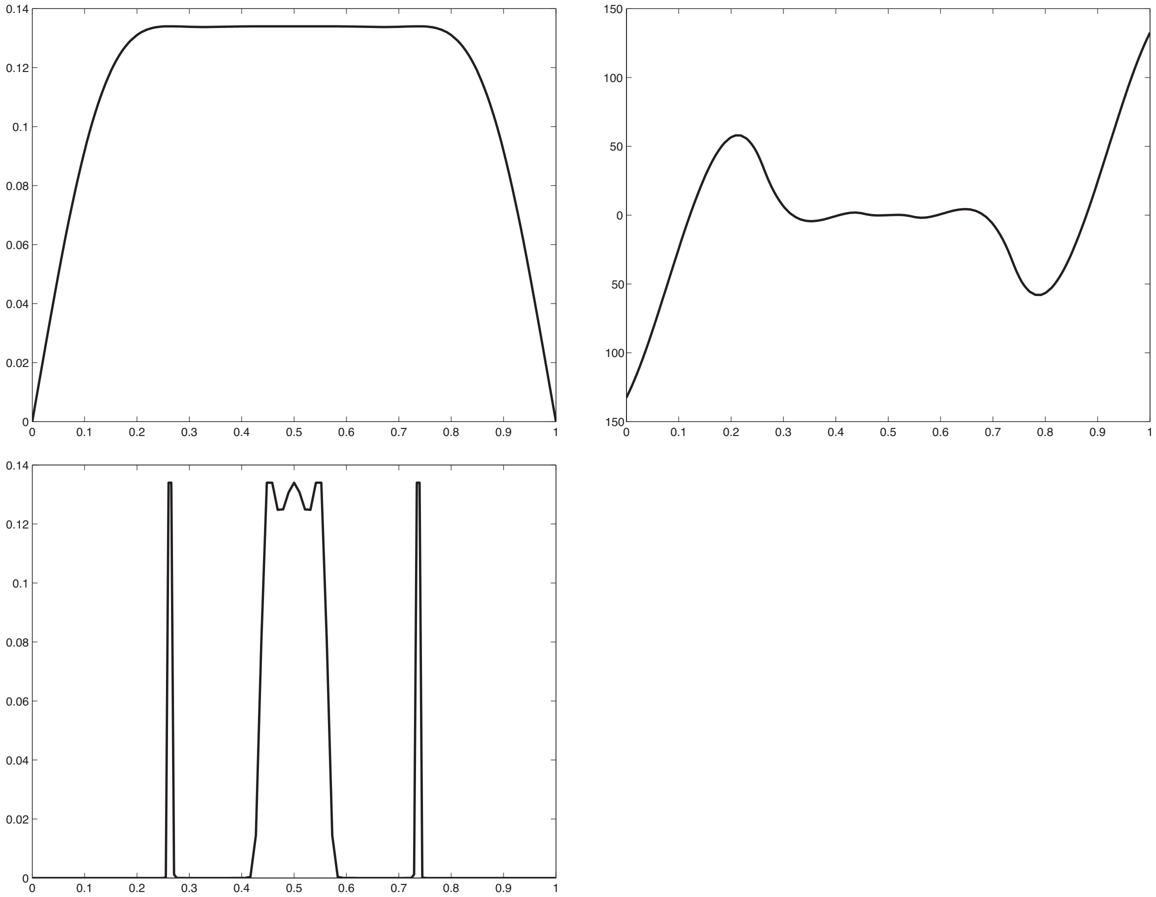


Figure 4.11: Example problem with $H = 1.5 \times 10^5$, plots of x_1 (top left), control u (top right), and $\mathcal{F}_{50000}(x_1)$ (bottom.) Number of touchpoints: 5.

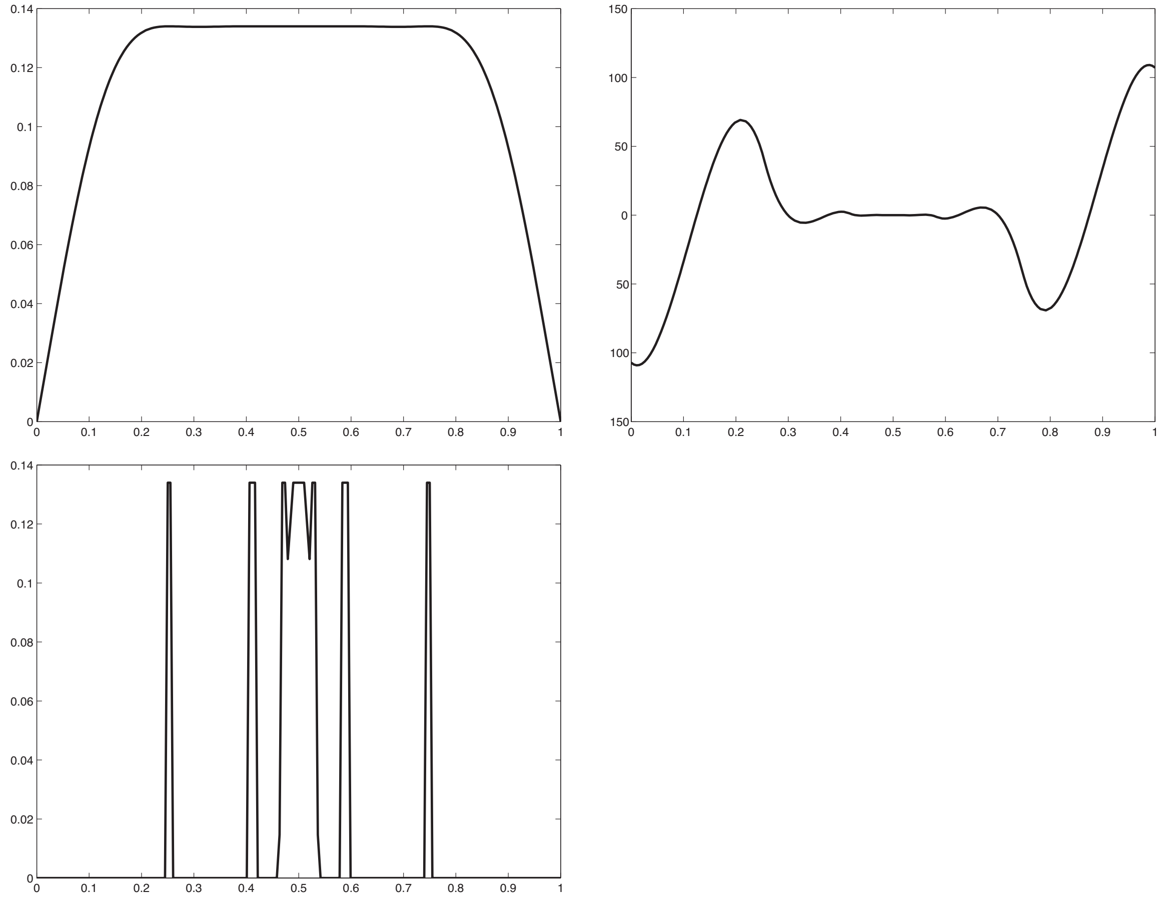


Figure 4.12: Example problem with $H = 3 \times 10^5$, plots of x_1 (top left), control u (top right), and $\mathcal{F}_{20000000}(x_1)$ (bottom.) Number of touchpoints: > 7.

4.2.5 Theoretical Result

We have demonstrated that the example problem, like the heat equation problem, exhibits ”virtual boundary arcs”. Next, we will further examine the issue of using indirect methods to solve virtual boundary arc problems. In order to even consider the use of an indirect method on a problem like this one, one must have at least an upper bound on the number of touchpoints. As we have seen that can be difficult to determine even on a simple problem like (4.16), as the oscillations get smaller and closer together. We will now demonstrate that, even if the number of touchpoints could be determined, conditioning problems would prevent this type of problem from being successfully solved by an indirect method.

Let t_k for $k = 1, \ldots, n-1$, be the kth point of contact between the solution and the constraint. Let $t_0 = 0$ and $t_n = 1$. Writing down the optimality conditions of the form given in [24] for the example problem we have

$$\begin{aligned} x_1' &= x_2 && \text{(4.18a)} \\ x_2' &= x_3 && \text{(4.18b)} \\ x_3' &= u && \text{(4.18c)} \\ \lambda_1' &= -\rho(x_1 - 1) - \eta && \text{(4.18d)} \\ \lambda_2' &= -\lambda_1 && \text{(4.18e)} \\ \lambda_3' &= -\lambda_2 && \text{(4.18f)} \\ 0 &= \lambda_3 + u && \text{(4.18g)} \end{aligned}$$

where $\eta = 0$ if $x_1 - L < 0$ and $\eta \geq 0$ if $x_1 = L$. In addition, we have the boundary conditions on x and the junction conditions

$$\lambda(t_k^-) = \lambda(t_k^+) + \zeta(t_k) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(4.18h)}$$

at every point t_k, where the constraint is touched, where $\zeta(t_k)$ is non-positive. From (4.18h) we conclude that λ_2, λ_3 are continuous at t_k but λ_1 can have a jump. Then from (4.18g) we have that u is continuous and hence x_1, x_2, x_3 are continuously differentiable on the whole interval.

Let $y_1 = x_1 - L$. Let $y_j = x_j$ for $j = 2, 3$ and $y_j = \lambda_{j-3}$ for $j = 4, 5, 6$. The optimality conditions on each subinterval $[t_k, t_{k+1}]$ (or more technically $[t_k^+, t_{k+1}^-]$), take the form $y' = Ay$ where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -\rho & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

and some boundary conditions at t_k^+, t_{k+1}^-.

Thus the optimal solution on each subinterval $[t_k, t_{k+1}]$ can be written as $e^{A(t-t_k)}z[k]$, $k = 0, \ldots, n-1$, where the $z[k]$ are constant vectors. For every touchpoint t_k, we have, by definition, $x_1(t_k) = L$ and, since the x_1 curve has a horizontal tangent at t_k and is convex down, $x_2(t_k) = 0$ and $x_3(t_k) \leq 0$. We can then define the remaining necessary conditions as

$$z[k]_1 = z[k]_2 = 0 \quad \text{for} \quad k = 1, \ldots, n-1 \tag{4.19}$$

$$z[k]_i = \left(e^{A\delta_k} z[k-1]\right)_i \quad \text{for} \quad k = 1, \ldots, n-1; \; i = 1, 2, 3, 5, 6 \tag{4.20}$$

$$z[k]_4 \geq \left(e^{A\delta_k} z[k-1]\right)_4 \quad \text{for} \quad k = 1, \ldots, n-1. \tag{4.21}$$

To find the exact solution we will need to guess n and solve for $z[k]$, and doing so will necessarily entail inverting $B = e^{A\delta_k}$. The condition number of B is $\kappa(B) = \|B\|\|B^{-1}\|$. Numerical experiments done with Matlab (calculating B for a fixed ρ

and different values of δ) show that the first element in the fourth row is dominant. So, a lower bound on $||B||$ is given by $|B_{4,1}| = |e^{A\delta_k}_{4,1}|$. A calculation in Maple shows that

$$\begin{aligned} e^{A\delta_k}_{4,1} &= \frac{1}{6}h^5 \left[e^{\frac{1}{2}h\delta_k} \left(\cos\left(\frac{\sqrt{3}}{2}h\delta_k \right) + \sqrt{3}\sin\left(\frac{\sqrt{3}}{2}h\delta_k \right) \right) \right. \\ &\quad \left. + e^{-\frac{1}{2}h\delta_k} \left(\cos\left(\frac{\sqrt{3}}{2}h\delta_k \right) - \sqrt{3}\sin\left(\frac{\sqrt{3}}{2}h\delta_k \right) \right) + e^{-h\delta_k} - e^{h\delta_k} \right] \\ &= \frac{1}{6}h^5 e^{h\delta_k} Q(h\delta_k) \end{aligned}$$

where $h = \rho^{\frac{1}{6}}$. Looking at the graph of $Q(z)$ we see that it goes positive near 1.24 and is above 0.4 by 1.82. Therefore

$$\|B\| \geq \frac{1}{15}\rho^{\frac{5}{6}} e^{h\delta_k} \text{ if } h\delta_k \geq 1.82. \tag{4.22}$$

For any matrix S we have $\|S\| \geq |\lambda|$ where λ is any eigenvalue of S. The matrix A has six distinct eigenvalues, including h and $-h$, so the eigenvalues of B include $e^{-h\delta_k}$ and $e^{h\delta_k}$. Thus the same holds for B^{-1} and we have

$$\|B^{-1}\| \geq e^{h\delta_k}, \quad \|B\| \geq e^{h\delta_k}. \tag{4.23}$$

Combining (4.22) and (4.23), we have

$$\kappa(B) \geq e^{2h\delta_k} \tag{4.24a}$$

and

$$\kappa(B) \geq e^{2h\delta_k} \max\{1, \frac{1}{15}\rho^{\frac{5}{6}}\} \text{ if } h\delta_k \geq 1.82. \tag{4.24b}$$

Conditions (4.24a) and (4.24b) show that as ρ increases, the problem becomes increasingly ill conditioned unless $h\delta_k$ stays bounded. But if $h\delta_k$ stays bounded with increasing h, then δ_k becomes small and the problem becomes ill-conditioned either because of the nearby touchpoints or for $k = 0$ because of the very short interval and nonzero boundary conditions on the left end. Conditions (4.24a) and (4.24b) describe the conditioning on a subinterval. Thus we have the optimality conditions for this problem become a very ill-conditioned boundary value problem for ρ in the range considered here independent of the actual number of touchpoints. The conditioning of the overall problem is much worse if the number of subintervals also increases as δ_k decreases which seems to be what we observe computationally.

Thus we have demonstrated a definite advantage of using direct transcription methods on inequality constrained problems. Since direct transcription methods only resolve behavior that is numerically significant, it is able to produce a meaningful solution to problems with "virtual boundary arcs" in spite of the inherent conditioning problems.

4.3 Open Questions: Initialization

4.3.1 Monitor Functions

In this section we will consider using monitor functions, a method that is commonly applied to solving boundary value problems [1], to initialize a code like SOCS. Even in the absence of constraints, stiffness in the function f or the lack of smoothness in the optimal control, as in, for example, with a bang-bang control, can result in a big difference between the properties of the original problem and the properties of a discretization of this problem on a coarse uniform grid. SOCS, which measures the accuracy of the solution by how close the behavior of the discretization approximates the behavior of the original problem, is then forced to refine the grid multiple times, resulting in finer and finer grids. Good grid refinement algorithms take care of this problem somewhat [7], but it still means more iterations to the grid refinement procedure and larger NLPs to work with at each iteration. One solution is a coarse but nonuniform initial grid that takes into account the structure of the problem and the initial guess.

The advantage of monitor functions for this task is that they can be used when f is a black box. The idea behind the algorithm, given in [6] is as follows:

1. Using a variable step integrator of fixed order P with tolerance 10^{-R}, integrate the system

$$\begin{aligned} \frac{dx}{dt} &= f(t,x,y(t,x)) \quad x(t_0) = \zeta \\ \frac{ds}{dt} &= \phi(t,x) \quad s(t_0) = 0 \end{aligned}$$

   where $y(t,x)$ is the initial guess for y, supplied by the user. Suppose that the integration terminates at t_f in Msteps and the value of s at the final gridpoint is s_f.

2. Let MAXGRID,MINGRID be given and let S be the tolerance desired for the initial guess and Q be the order of the discretization used by SOCS (2 for Trapezoid and 4 for Hermite-Simpson). Set the number of nodes in the initial grid according to the formula (explained in [6]):

$$N = \min\{MAXGRID, \max\{MINGRID, M10^{S/Q-R/P}\}\}.$$

   and let

$$s_k = s_f \frac{k-1}{N-1}, \quad k = 1, ..., N$$

3. Using a numerical integrator, integrate

$$\begin{aligned} \frac{dx}{ds} &= \frac{f(t,x,y(t,x))}{\phi(t,x)} \quad x(0) = \zeta \\ \frac{dt}{ds} &= \frac{1}{\phi(t,x)} \quad t(0) = t_0 \end{aligned}$$

   from 0 to s_f and output $(x(s), t(s))$ at $s = s_k$ for $k = 1, ..., N$.

Two types of monitor functions were considered by the authors of [6]:

$$\phi(t,x) = \sqrt{\alpha + \sum_{i=0}^{m_1} \beta_i f_i(t,x)^2}$$

and

$$\phi(t,x) = \gamma + \left(\sum_{i=0}^{m_1} \beta_i f_i(t,x)^2 \right)^{\omega}$$

where

$$\gamma = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \|f\|^{2\omega} dt$$

has to be determined by a separate integration.

The first monitor function with $\alpha = \beta_i = 1$ is the arclength function. It equidistributes the nodes along the graph of x. The second monitor function, first suggested in [2] to minimize the error in linear interpolation, is proved in [6] to provide a lower residual error than the first one. Numerical experiments in [6] have demonstrated the usefulness of monitor functions on some example problems, but several issues still remain to be investigated before initialization routines based on monitor functions can be integrated into a package like SOCS.

4.3.2 Order Reduction

The method discussed in this section applies specifically to inequality constrained problems that are hard to initialize because the initial solution must satisfy all constraints.

One technique for dealing with problems that have high order constraints is **order reduction**. The constraint (equality or inequality) of order p is differentiated p times. Each differentiation produces additional boundary conditions and differential equations. The resulting system is a first-order DAE, easily solved for the states given any initial control values. There are several disadvantages to this approach:

- It requires differentiating the constraint function, i.e. additional work by the user or by a numerical integrator, often very difficult or even impossible if the constraint function is a "black box".
- Repeated integrations affect the accuracy of the result, so the optimal solution to the problem thus transformed may be very different from the optimal solution to the original problem
- The transformed problem is much larger than the original problem, so solving it takes a lot more time and computational resources.

However, what if we used order reduction for initialization purposes only? If all we are interested in is a feasible solution, accuracy is no longer as important. A numerical integrator can be used to transform the constraints and to solve the transformed problem for some value of the control. The result will then be transmitted to a direct transcription solver as an initial feasible solution to the original problem.

For a problem of the form (4.1) without final state constraints ($\phi = 0$) and a single scalar p^{th} order inequality constraint the algorithm looks like this: rewrite $g(x) \leq 0$ as $g(x) + \frac{1}{2}\alpha_0^2 = 0$. From this, we get $\alpha_0(t_0) = \pm\sqrt{-2g(\zeta)}$. Differentiating once, we get $g'(x) + \alpha_1\alpha_0 = 0$, where $\alpha_1 = \alpha_0'$. From this, we get the boundary condition $\alpha_1(t_0) = -\frac{g'(\zeta)}{\alpha_0(t_0)}$. We do this p times, until the scalar control u appears in the equation, so that in the end the problem (4.1) is transformed into

$$\min C(x(t_f))dt \tag{4.25a}$$

$$\begin{aligned} x' &= f(x,y) && (4.25b)\\ \alpha_i' &= \alpha_{i+1}, i = 0, ..., p-1 && (4.25c)\\ x(t_0) &= \zeta && (4.25d)\\ \alpha(t_0) &= \bar{\zeta} && (4.25e)\\ 0 &= g^{(p)}(x,u) + S_p(\alpha). && (4.25f) \end{aligned}$$

The constraint (4.25f) can then be used to solve for u in terms of α, so that what we have as a result is an unconstrained problem with a single control α_p. Setting α_p to whatever we want, we can then simply integrate the equations (4.25b)–(4.25c) starting from $(x, \alpha)(t_0) = (\zeta, \bar{\zeta})$.

However, in practice it is not so simple. Some numerical experiments with using this approach done on a variation of the trolley problem (see Section 1.1) have demonstrated another problem that can arise when applying order reduction to difficult-to-solve inequality constrained problems - the resulting system of differential equations can be very stiff if we tried to solve it for a fixed α_p.

But there are still other things that can be tried. For example, the initialization strategies built into SOCS or another software package can be used to find an initial solution (x, u, α) to the transformed problem (4.25). If there is any success with using this method on problems with smooth constraints described by inifintely differentiable functions like the trolley problem, there still remain many interesting questions regarding numerical differentiation and integration of "black box" constraints, the goal being of course to eliminate additional work on the part of the user.

Another side question would be finding a non-recursive formula for determinitng the coefficients of the polynomial $S_p(\alpha)$. For p from 0 to 6, the polynomial takes the form

$$\begin{aligned}
S_0\,(\alpha) &= 1/2\,{\alpha_0}^2 \\
S_1\,(\alpha) &= \alpha_0\alpha_1 \\
S_2\,(\alpha) &= {\alpha_1}^2 + \alpha_0\alpha_2 \\
S_3\,(\alpha) &= 3\,\alpha_1\alpha_2 + \alpha_0\alpha_3 \\
S_4\,(\alpha) &= 3\,{\alpha_2}^2 + 4\,\alpha_1\alpha_3 + \alpha_0\alpha_4 \\
S_5\,(\alpha) &= 10\,\alpha_2\alpha_3 + 5\,\alpha_1\alpha_4 + \alpha_0\alpha_5 \\
S_6\,(\alpha) &= 10\,{\alpha_3}^2 + 15\,\alpha_2\alpha_4 + 6\,\alpha_1\alpha_5 + \alpha_0\alpha_6
\end{aligned}$$

and S_p is obtained from S_{p-1} by a simple application of the chain rule. Whereas numerical differentiation should be used on g, applying it to S would result in unnecessary loss of accuracy. It would be much better to append S_p to $g^{(p)}$, which can only be done if we have a non-recursive formula for its coefficients.

In summary, using order reduction for initialization of high-order optimal control problems may not be feasible, and the results so far are not very encouraging. However, there are many things one could still try, and there are also some interesting questions arising from this problem. Some of them have to do with numerical integration and differentiation. Others, like the determination of the coefficients of $S_p(\alpha)$ belong to the realm of "pure math", and may very well have been pursued by someone in a completely unrelated field.

Chapter 5

Summary of Contributions

Inequality Constrained Problems (Chapter 4)

Section 4.1 presents two results selected from among several papers studied by the author in connection with the project discussed in greater detail in Chapter 3. The results are restated to maintain uniformity of notation throughout the monograph and observations are made regarding the different assumptions and practical considerations.

The original material in Section 4.2 is based on work done by Dr. Betts, Dr. Campbell and the author in 2003. The heat equation problem was suggested by Dr. Campbell who also coined the term "virtual boundary arcs". The author's personal contributions include

1. Designing and implementing numerical experiments in SOCS for the heat equation problem (4.13a)–(4.13) as well as the example problem (4.16).

2. Designing the transformation functions $w(v)$ and $\mathcal{F}_{\mathcal{K}}$ which provide for a visualization of small oscillations exhibited by the solutions to both problems.

3. For the example problem which demonstrates the conditioning problem inherent

in problems with virtual boundary arcs, calculating the upper bounds on the condition number, (4.24a) and (4.24b).

Two papers were published based on the results of this research:

- *J. T. Betts, S. L. Campbell, and A. Engelsone, Direct transcription solution of optimal control problems with higher order state constraints: theory vs. practice, Optimization and Engineering, to appear.*
- *J. T. Betts, S. L Campbell, and A. Engelsone, Direct transcription solution of inequality constrained optimal control problems, Proc. 2004 American Control Conf., Boston, 1622-1626.*

Section 4.3 discusses the problem of initializing numerical algorithms for solving optimal control problems. The idea of using order reduction for initialization was suggested by the author, who worked on it independently in early 2004. Experiments in Matlab and SOCS showed the shortcomings of the approach originally adopted by the author. The section discusses possible solutions. Using monitor functions to initialize SOCS was a joint project of Dr. Campbell and a former student Mr. Kalla. In the spring of 2004, the author debugged, organized and rewrote some of the Matlab code written by Mr. Kalla and noted some numerical phenomena that had not been observed before.

Equality Constrained Problems (Chapter 3)

Section 3.1 includes the authors observations about the lack of convergence results for equality constrained problems and notes which of the results in other chapters may be applicable to this case.

The results in Section 3.2, developed by Dr. Campbell, Dr. Betts and the author in 2004, challenge common thinking about control problems and convergence and extend strong direct transcription convergence results to a class of problems that most other methods are unable to solve.

The proof of Theorem 3.2, which establishes equivalence between unconstrained problems and certain high-index equality constrained problems, is the author's as are the corollaries which extend prior convergence results to problems with "virtual index 1". However, the conjecture belongs to Dr. Campbell, who has also has provided a lot of guidance with the proofs.

Two papers were published on the subject of virtual index:

- *A. Engelsone, S. L. Campbell, and J. T. Betts, Direct transcription solution of higher-index optimal control problems, Proc. IMACS World Congress on Scientific Computation, Paris, 2005.*
- *A. Engelsone, S. L. Campbell, and J. T. Betts, Direct transcription solution of higher-index optimal control problems and the virtual index, Applied Numerical Mathematics, to appear.*

In addition, the author gave two presentations of these results,

- *Direct Transcription Solution of Higher-Index Optimal Control Problems McMaster Optimization Conference: Theory and Applications, Hamilton, Canada, July 28-30, 2004*
- *Direct Transcription Solution of Higher-Index Optimal Control Problems IMACS World Congress on Scientific Computation, Paris, France, July 12-16, 2005*

Unconstrained Problems (Chapter 2)

Section 2.1 contains a detailed overview of existing convergence results for unconstrained problems. The author has restated and reformulated some of the results to maintain consistency of notation and make them more easily accessible to practitioners. The section also introduces the issue of multiplier convergence which is the subject of the next section.

In spring of 2005, while running numerical experiments for example problems in Chapter 3, the author noticed the discrepancy between the theoretical result (reproduced here as Theorem 2.5) and the multiplier estimates provided by SOCS for the Trapezoid and Hermite-Simpson discretization methods. The author then designed numerical experiments using SOCS and Maple and talked to Dr. Betts, our contact at SOCS, at length about the SOCS procedures for estimating multipliers. The author eventually discovered the reason for the discrepancy and demonstrated it through numerical experiments.

Finally, the author proved Theorem 2.6, which shows that multiplier convergence for the Trapezoid method depends on the particular implementation of the method. The theorem states that the multipliers obtained from the compressed version of the method implemented in most codes are only 1st order accurate at the gridpoints, however, a simple interpolation (currently being implemented in SOCS) is sufficient to produce multiplier values that are 2nd order accurate. The theorem also states that the controls obtained from any implementation of the Trapezoid method are 2nd order accurate on the inside gridpoints. This is the only result known to the author that establishes higher order control convergence for a commonly used discretization method. All these results are presented and discussed in Section 2.2.

The one currently published paper

- *A. Engelsone, S. L. Campbell, and J. T. Betts, Order of convergence in the direct transcription solution of optimal control problems, Proc. IEEE Conf. Decision Control - European Control Conference, Seville, Spain, 2005.*

discusses the literature and notes the discrepancy between earlier theory and numerical results for both Trapezoid and Hermite-Simpson but does not include Theorem 2.6. The paper

- *A. Engelsone and S. L. Campbell, Adjoint Estimation using Direct Transcription Multipliers: Trapezoidal Method,*

which has been submitted for publication, includes more numerical results on Trapezoid method and Theorem 2.6. The author also hopes to have a theoretical result for the Hermite-Simpson method, discussed here in detail in Section 2.3, in the near future.

Index

List of References

[1] U.M. Ascher, R.M.M. Mattheij, and R.D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations.* SIAM, Philadelphia, 1995.

[2] G. Beckett, J.A. Mackenzie, A. Ramage, and D. M. Sloan. On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution. *J. Comp. Physics*, 167:372–392, 2001.

[3] J. T. Betts. *Practical Methods for Optimal Control using Nonlinear Programming.* SIAM, Philadelphia, 2001.

[4] J. T. Betts and S. L. Campbell. *Mathematics in Industry: Challenges and Frontiers. A Process View: Practice and Theory (PR121)*, chapter Discretize then Optimize. SIAM, 2004.

[5] J. T. Betts and W. P. Huffman. Sparse Optimal Control Software SOCS. Technical report, Mathematics and Engineering Analysis Technical Document MEA-LR-085, Boeing Information and Support Services, July 1997.

[6] J.T. Betts, S.L. Campbell, and N.N. Kalla. Initialization of direct transcription optimal control software. *Proc. IEEE Conf. Decision Control*, pages 3802–3807, 2003.

[7] N. Biehn. *Implicit Runge-Kutta Methods for Stiff and State Constrained Optimal Control Problems.* PhD thesis, North Carolina State University, 2001.

[8] N. Biehn, J. T. Betts, S. L. Campbell, and W.F. Huffman. Compensating for order variation in mesh refinement for direct transcription methods. *J. Comp. Appl. Math*, 125:147–158, 2000.

[9] J.F. Bonnans and J. Laurent-Varin. Computation of order conditions for symplectic partitioned runge-kutta schemes with application to optimal control. Technical report, INRIA Rocquentcourt, December 2004.

[10] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, Philadelphia, 1996.

[11] A.E. Bryson and Yu-Chi Ho. *Applied Optimal Control.* Hemisphere, Washington, 1975.

[12] A. L. Dontchev. Error estimates for a discrete approximation to constrained control problems. *SIAM J Numer Anal*, 18:500–514, 1981.

[13] A. L. Dontchev and W.W. Hager. The Euler approximation in state constrained optimal control. *Mathematics of Computation*, 70:173–203, 2001.

[14] A. L. Dontchev, W.W. Hager, and V.M. Veliov. Second-order runge-kutta approximations in control constrained optimal control. *SIAM J Numer Anal.*, 38:202–226, 2000.

[15] A.L. Dontchev and W.W. Hager. Lipschitzian stability in nonlinear control and optimization. *SIAM J. Control and Optimization*, 3:569–603, 1993.

[16] A.L. Dontchev and W.W. Hager. Lipschitzian stability for state constrained nonlinear optimal control. *SIAM J. on Control and Optimization*, 36:696–718, 1998.

[17] A. Engelsone, S. L. Campbell, and J. T. Betts. Direct transcription solution of higher-index optimal control problems and the virtual index. *Applied Numerical Mathematics*, 2006. to appear.

[18] Jean-Claude Evard. On the existence of bases of class C^p of the kernel and the image of a matrix function. *Linear Algebra and Its Applications*, 135:33–67, 1990.

[19] W.W. Hager. Rates of convergence for discrete approximations to unconstrained control problems. *SIAM J Numer Anal*, 13:321–338, 1976.

[20] W.W. Hager. Multiplier methods for nonlinear optimal control. *SIAM J. Numer. Anal.*, 27:1061–1080, 1990.

[21] W.W. Hager. Runge-kutta methods in optimal control and the transformed adjoint system. *Numer. Math.*, 87:247–282, 2000.

[22] E. Hairer, S. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer-Verlag, New York, 1993.

[23] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, 1991.

[24] D. H. Jacobsen, M. M. Lele, and J. L. Speyer. New necessary conditions of optimality for control problems with state variable inequality constraints. *J. Math. Anal. Appl.*, 35:255–284, 1971.

[25] Shivakumar Kameswaran and L.T. Biegler. Advantages of nonlinear programming based methodologies for the solution of inequality path-constrained optimal control problems - an analysis of the Betts and Campbell heat conduction problem. preprint, 2005.

[26] Shivakumar Kameswaran and L.T. Biegler. Convergence rates for direct transcription of optimal control problems using collocation at Radau points. preprint, 2005.

[27] Shivakumar Kameswaran and L.T. Biegler. Convergence rates for direct transcription of optimal control problems with final-time equality constraints using collocation at Radau points. preprint, 2005.

[28] F. L. Lewis and V. L. Syrmos. *Optimal Control.* Wiley and Sons, Inc., 1995.

[29] K. Malanowski, C. Buskens, and H. Maurer. *Mathematical Programming with Data Perturbations*, chapter Convergence of Approximations to Nonlinear Optimal Control Problems. Lecture Notes in Pure and Appl. Math. Dekker, New York, 1998.

[30] E. Polak. On the use of consistent approximations in the solution of semi-infinite optimization and optimal control problems. *Math. Programming*, 62:385–414, 1993.

[31] R. Pytlak. *Numerical Methods for Optimal Control Problems with State Constraints.* Springer-Verlag, New York, 1999.

[32] A. Schwartz and E. Polak. Consistent approximations for optimal control problems based on runge-kutta integration. *SIAM J. Control Optim.*, 34:1235–1269, 1996.

[33] H. Seywald and E. M. Cliff. Short communications on the existence of touch points for first-order state inequality constraints. *Optimal Control Applications Methods*, 17:357–366, 1996.