ABSTRACT

COONLEY, BRETT WILLIAM. Sequential Programming for PDE Constrained Optimizations. (Under the direction of Kazufumi Ito.)

Sequential Progamming (SP) is a method for solving PDE constrained optimizations. These include many applications of practical importance, especially optimal control problems, such as the optimal control of fluid flow governed by the incompressible Navier-Stokes equations. The Sequential Programming (SP) method is guaranteed to converge to an optimizer of a PDE constraint problem under proper conditions. A hallmark of the method is that there is no line search needed for the damped update involved in sequential iterations.

In this thesis we discuss the theory for the Sequential Programming (SP) method. We introduce the (basic) SP method for constrained optimization and analyze its convergence and convergence rate using the Lagrange multiplier theory. The Lagrange multiplier theory is the basis for developing the necessary optimality condition for a given PDE constraint optimization and developing solvers for the SP method. The necessary optimality condition is of the form of a saddle point problem for the primal and dual variables. Existence of Lagrange multipliers for the necessary optimality system is based on the constraint qualifications, i.e., the regular point condition. Non-smoothness in the cost and constraints is treated as a variational inequality for the optimality condition. We make comparisons with the projected gradient method and SQP method, and show that SP is the middle ground between the two methods. It is a very effective method for a large scale optimal control problem. Specifically, we introduce and analyze the iterative method for the inequality constraint case and non-smooth cost functional based on the semi-smooth Newton method.

Sequential Programming for PDE Constrained Optimizations

by
Brett William Coonley

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

Mathematics

Raleigh, North Carolina

2015

APPROVED BY:

Zhilin Li	Xiao-Biao Lin
Hien Tran	Kazufumi Ito Chair of Advisory Committee

DEDICATION

To my parents Doug and Suzanne Coonley.

BIOGRAPHY

The author was born in New Hampshire where he attended the Montessori School. He moved to Connecticut and attended Union Elementary School, Lake Garda Elementary School, Har-Bur Middle School, and graduated valedictorian of Lewis S. Mills High School in 1995. He was a member of the high school Math Team during 8th grade through senior year when he was President and MVP of the team. He was voted DAR Good Citizen by his high school class, performed classical piano in the high school talent show, qualified for the State Wrestling Championships, and was a Boy Scout in Troop 37 of Unionville.

The author attended Bates College in Lewiston, Maine, where he received his Bachelor of Arts in Mathematics in 2001. He did his undergraduate thesis on "Applications of Hilbert Space Theory to Quantum Mechanics." After college, he moved to Boston, Massachusetts where he worked at the MIT Department of Mathematics as technical publishing assistant until 2008. He moved to Durham, North Carolina and began graduate studies at North Carolina State University in 2009. In his spare time the author likes to do woodworking and play piano and guitar.

ACKNOWLEDGEMENTS

My greatest thanks of all go to my advisor Dr. Kazufumi Ito for his guidance and dedication. Dr. Ito showed me patience, kindness, and persistence, and instilled in me a drive to reach for more than I need.

I would also like to thank my committee members Dr. Zhilin Li, Dr. Xiao-Biao Lin, and Dr. Hien Tran for being available and encouraging. Each of you has unique skill and talents that have shaped and improved my work.

Special thanks go to my colleagues Amanda Landi and Melissa Ngamini, and my good friend Ryan Patridge.

Lastly I thank my family and friends and neighbors for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In this thesis we consider constraint optimization problems in Hilbert spaces. Optimization is one of the key components for mathematical modeling of real world problems and the solution method provides an accurate and essential description and validation of the mathematical model. Optimization problems are encountered frequently in engineering and sciences and have widespread practical applications. An appropriately chosen cost functional is minimized subject to constraints. For example, the constraints are in the form of equality constraints and inequality constraints. We also discuss a class of non-smooth optimizations. A non-smooth optimization method is a very basic modeling tool and enlarges and enhances the applications of the optimization method in general. For example in imaging/signal analysis the sparsity optimization is used by means of L^1 and TV norm regularization.

A sequential algorithm based on Sequential Programming (SP) is developed and analyzed for the constraint optimization. Especially, we consider the optimal control problem for ODEs and PDEs and inverse problems and structural design problems. Sequential Programming (SP) is a method for solving constrained optimization problems and uses a linearization of the equality constraints. Sequential Programming is shown to be a middle ground between other solution methods. Specifically, we compare the SP method to the gradient method and SQP (Sequential Quadratic Programming) and show that SP remedies drawbacks of each method.

In particular, the gradient method can be very slowly convergent for systems that are stiff and highly nonlinear. SP remedies this by linearizing the equality constraint and solving the resulting saddle point problem [29, 7, 25, 53, 46]. The SQP (Sequential Quadratic Programming) method [29, 42, 32, 33, 62] solves a sequence of optimization subproblems, each of which optimizes a quadratic model for the Lagrangian functional, e.g., [29], subject to a linearization of the constraints. If the problem has only equality constraints, then the method is equivalent to applying Newton's method to the first order optimality conditions [48, 67, 29] or the Karush-Kuhn-Tucker conditions if we use the exact Hessian. Like for the Newton method the

objective function and the constraints are assumed to be twice continuously differentiable for the SQP method. SQP is quadratically convergent but requires a consistent quadratic model to the Hessian of the Lagrangian and globalization methods for a stable implementation.

It is a distinctive feature that SP does not rely directly on the Hessian information of the Lagrange functional and thus it can be used for non-smooth problems and in many problems of practical interest. It can avoid instabilities due to possible indefiniteness of the Hessian of the Lagrange functional during the iteration.

The SP method is guaranteed to converge to an optimal solution with (very improved) linear rate under appropriate conditions. In fact, we show that the necessary optimality for the original constraint optimization is a perturbation of the one for the SP constraint problem. In general, the convergence is dependent on the error between the equality constraint and its linearization and the curvature error of the Lagrange functional.

Each SP method requires a (nonlinear) saddle point solver for a linear equality constraint. We develop the fixed point iterate method for the saddle point problem based on the (linear) primal solver and the adjoint equation solver and the optimality condition. That is, it involves exactly the same as in a step in the gradient method for the constraint optimization. But we use it as an inner loop for the saddle point problem. As a consequence SP requires much fewer gradient-like steps in total.

The plan of the presentation is as follows. In Chapter 2 we present the basic optimization theory for constrained and unconstrained optimizations. We discuss the existence of optimizers (Section 2.2.1) and formulate the necessary optimality as a variational inequality (Section 2.2.2).

The Lagrange multiplier theory (Section 2.3) is the basis for developing the necessary optimality condition for a given PDE constraint optimization [8, 29] and developing solvers for the SP method. The necessary optimality condition is of the form of a saddle point problem for the primal and dual variables. Existence of Lagrange multipliers for the necessary optimality system is based on the constraint qualifications, i.e., the regular point condition [48, 67, 29]. Non-smoothness in the cost and constraints is treated as a variational inequality for the optimality condition.

In Chapter 3 we introduce the (basic) SP method for constrained optimization and analyze its convergence and convergence rate using the Lagrange multiplier theory (Section 3.4.1). Comparison with the projected gradient method and SQP is detailed and we discuss pros and cons. Also, we develop and analyze a second order SP method (Section 3.4.2). The second order information of the Lagrange functional is incorporated by a sequential difference of derivatives of the Lagrange functional instead of the quadratic model for the Hessian of the Lagrange functional.

In Chapter 4 we discuss solvers of the saddle point problems for each SP step. The fixed point problems are formulated for SP as well as for the second order variant and for non-

smooth problems. We use the damped fixed point iterate for the resulting fixed point problem and thus develop a sequential method for a general class of constrained optimizations using the SP framework. As a specific example the fixed point problem for the control variables is developed for the optimal control problem as a reduced order method and we use the conjugate gradient method for a well-conditioned fixed point problem (Section 3.6.1). It is a very effective method for a large scale optimal control problem. Specifically, we introduce and analyze the iterative method for the inequality constraint case and non-smooth cost functional based on the semi-smooth Newton method (Section 4.4).

We are most interested in applying the SP method to solve the optimal control problems and PDE constrained optimizations. For PDE constraint optimization, we consider three specific problems and develop implementations for them based on SP. First, we develop the inverse medium (coefficient control) for the elliptic equations (Section 2.2.4). We also consider the moving damping actuator problem (Section 4.3.1). This is an optimization problem for a wave equation with moving damping, i.e., we formulate the optimal control problem for determining the optimal path of dampers with prescribed damping distribution.

Finally, we consider the control of fluid flow [22, 19, 29] governed by the incompressible Navier-Stokes equations. We present the Navier-Stokes theory in Chapter 5, and discuss the weak form of the equality constraint and define the weak solution to the constraint (Section 5.1.3). Then, we formulate the weak form of the necessary optimality condition (Section 5.2) and develop a discretization in time and in space (Section 5.4) for the solution of the controlled Navier-Stokes problem.

In Chapter 6 we develop the time integration method for the dynamical system. It is essential to have an efficient (few number of unknowns) but accurate (high order approximation) method. We use the Runge-Kutta-Gauss time integration [10] (Section 6.1.2) and develop an effective saddle point solver for SP applied to the resulting finite dimensional constraint optimization.

In Chapter 7 we present specific examples to demonstrate the applicability of the SP method for the constrained optimization. First, we show how to apply SP for the finite dimensional optimal control problem. We use the Lorenz attractor system as a test example. Second, we consider a non-smooth coefficient optimal control problem in elliptic equations. The third problem is the moving damping actuator control problem. They represent the constrained minimization with technical difficulties (challenge) and show the feasibility of the SP method. Also, we present an open loop simulation for our proposed numerical method for Navier-Stokes control systems.

Chapter 2

Optimization Theory

In this chapter we introduce a class of constrained optimization problems and develop the corresponding Lagrange multiplier theory. Our motivation is PDE constrained optimization problems such as the incompressible Navier-Stokes control problem for fluid flow and the moving damping actuator problem. We develop the necessary tools to solve constrained optimizations which involves the generalized Lagrange multiplier theory for constrained optimizations including equality and inequality constraints. Algorithmically, we cast the necessary optimality system as a saddle point problem with primal and dual variables. Also, there is the necessity to develop the necessary optimality for non-smooth optimization problems. We develop the necessary optimality as a variational inequality. It contains PDE constrained optimization and the inequality constraint problem and non-smooth optimization. Concrete examples and applications of the multiplier theory are presented.

2.1 Constrained Optimization

In general most problems of interest can be cast in the following form. We discuss the constrained optimization of the form

$$\min\ F(y) \quad \text{subject to} \quad E(y) = 0,\ G(y) \leq 0 \text{ and } y \in \mathcal{C}, \tag{2.1}$$

where $\mathcal{C}$ is a closed convex subset of a Banach space X, and $F : X \to R$ is a functional over $\mathcal{C}$. $E : X \to Y$ is an equality constraint taking values in a Banach space Y, and $G : X \to Z$ is an inequality constraint with ordered ($\leq$) Banach space Z.

The cost functional F may not be continuously differentiable in general, but we assume E and G are continuously Fréchet differentiable. An important class of constrained optimizations

we discuss is for the pair (x, u) on the product space $X \times U$ of the form

$$\min\ F(x) + H(u) \quad \text{subject to} \quad E(x, u) = 0,\ u \in \mathcal{C}, \tag{2.2}$$

where $\mathcal{C}$ is a closed convex subset of U. This is the "control form" of equality constrained optimization. Problems of the form (2.2) are important for applications such as the control problem and the inverse medium problem as discussed in Section 2.2.4. Without any difficulty $(y = (x, u), G = 0, \mathcal{C} := X \times \mathcal{C})$ one can relate (2.2) with a specific case of (2.1).

For (2.2) we have $y = (x, u)$ in the product space $X \times \mathcal{C}$ and thus $E(y) = E(x, u) = 0$ and $u \in \mathcal{C}$, i.e., we have a constraint set for u only. In applications x denotes the state $x \in X$ with state space X and u denotes the design, medium, or control variable $u \in U$ with parameter space U. The equality constraint $E(x, u) = 0$ defines an equation between the state $x \in X$ and the control variable $u \in \mathcal{C}$. We often assume that given $u \in \mathcal{C}$ the equation $E(x, u) = 0$ has a unique solution $x = x(u) \in X$. That is, the state x is implicitly a function of u by the equality constraint. Thus, we define the composite cost functional $J(u) = F(x(u), u)$, and minimize J over $u \in \mathcal{C}$ only. The cost functional is in general not necessarily separable, i.e., $F(y) = F(x, u)$. In addition, F and H are not necessarily continuously differentiable. For example

$$F(y) = \int_\Omega (f(x(\omega)) + h(u(\omega)))\, d\omega,$$

where $f : R^n \to R$ and $h : R^m \to R$ are lower semicontinuous functions.

It is essential to analyze solutions to the equality constraint $E(x, u) = 0$. For example, if $E(x, u) = 0$ is the incompressible Navier-Stokes equation (5.3), we must be concerned with the well-posedness (feasibility) of the PDE constraint. That is, a solution to the PDE constraint must exist in order for the minimization of the cost functional to be performed. Uniqueness of solutions to the constraint must also be addressed.

2.2 Unconstrained Non-smooth Minimization

In this section we discuss unconstrained minimization of the form

$$\min\ F(x) \quad \text{over}\ \ x \in \mathcal{C}, \tag{2.3}$$

which is the specific case of (2.1) without equality or inequality constraint. First, we discuss the existence of minimizers to (2.3). We then derive the necessary optimality condition in the form of a variational inequality for all $x \in \mathcal{C}$.

2.2.1 Existence of Minimizers

Existence of minimizers to F uses the Weierstrass theorem in Banach spaces [66]. Suppose $\mathcal{C}$ is compact and F is lower semicontinuous. Then (2.3) has a minimizer $x^* \in \mathcal{C}$. In general, we assume F is weakly sequentially lower semicontinuous, i.e.,

$$F(x) \leq \liminf F(x_n)$$

for all weakly convergent sequences (x_n) to x in X or for all weakly-star convergent sequences (x_n) to x in $X = Y^*$. Assume either F is coercive on $\mathcal{C}$, i.e.,

$$F(x) \to \infty \quad \text{as} \quad |x| \to \infty,\ x \in \mathcal{C},$$

or $\mathcal{C}$ is bounded. In fact, let $\eta = \inf_{x\in\mathcal{C}} F(x)$ and $x_n \in \mathcal{C}$ be a minimizing sequence, i.e., $F(x_n)$ is decreasing and $\lim_{n\to\infty} F(x_n) = \eta$. If X is reflexive there exists a weakly convergent subsequence (x_{n_k}) to $\bar{x} \in \mathcal{C}$ and since is F is weakly sequentially lower semicontinuous

$$\eta = \lim_{n_k\to\infty} F(x_{n_k}) \geq F(\bar{x}) \geq \eta,$$

which implies $F(\bar{x}) = \eta$, and $\bar{x}$ is a minimizer. If $X = Y^*$ we use the weak-star topology.

In the case of (2.1) we assume that F, E, G are weakly continuous, i.e., for $y_n \in \mathcal{C}$ converging weakly to $y \in \mathcal{C}$

$$F(y_n) \to F(y), \quad E(y_n) \text{ weakly converges to } E(y), \quad G(y_n) \text{ weakly converges to } G(y).$$

Then the above arguments show the existence of minimizers to (2.1).

2.2.2 Necessary Optimality

The necessary optimality for unconstrained minimization (2.3) is of the form of a variational inequality. There are two types of variational inequality to cover the case where F is differentiable and the case where F can be decomposed into a differentiable part and convex part.

Theorem 2.1. *(Variational inequality of first kind) If F is Gateaux differentiable at a minimizer $x^* \in \mathcal{C}$, i.e.,*

$$F'(x^*)(d) = \lim_{t\to 0} \frac{F(x^* + td) - F(x^*)}{t}$$

exists for all directions $d \in X$, then the necessary optimality for (2.3) *is*

$$F'(x^*)(x - x^*) \geq 0 \quad \textit{for all } x \in \mathcal{C}. \tag{2.4}$$

Proof. If $x^* \in \mathcal{C}$ is a minimizer of (2.3), then

$$0 \le F(x^* + t\,(x - x^*)) - F(x^*) \tag{2.5}$$

for all $x \in \mathcal{C}$ and $t \in [0,1]$ where we use the convexity of $\mathcal{C}$. The Gateaux derivative of F at x^* in the direction of $x - x^*$ is

$$F'(x^*)(x - x^*) = \lim_{t \to 0} \frac{F(x^* + t\,(x - x^*)) - F(x^*)}{t} = \lim_{t \downarrow 0} \frac{F(x^* + t\,(x - x^*)) - F(x^*)}{t} \ge 0$$

where we used the minimality condition (2.5) to obtain the inequality. □

Example Consider the minimization in R^2 of a differentiable function $F : R^2 \to R$:

$$\min F(x_1, x_2) \quad \text{over} \quad \mathcal{C} = [0,1] \times [0,1].$$

If a minimizer (x_1^*, x_2^*) is on the right boundary, $x_1^* = 1$ and $0 < x_2^* < 1$, the necessary optimality is

$$\frac{\partial F}{\partial x_1} \le 0, \quad \frac{\partial F}{\partial x_2} = 0.$$

This follows since we have direction $x_1 - x_1^* \le 0$ for all $x_1 \in [0,1]$, so the derivative in x_1 must be negative to satisfy (2.4). Similarly, if a minimizer is on the top boundary, $x_2^* = 1$ and $0 < x_1^* < 1$, the necessary optimality is

$$\frac{\partial F}{\partial x_1} = 0, \quad \frac{\partial F}{\partial x_2} \le 0.$$

Next we consider the case

$$F = F_0 + F_1, \tag{2.6}$$

where F_0 is differentiable and F_1 is convex on $\mathcal{C}$, i.e.,

$$F_1(tx_1 + (1-t)x_2) \le tF_1(x_1) + (1-t)F_1(x_2)$$

for all $x_1, x_2 \in \mathcal{C}$ and $t \in [0,1]$.

Theorem 2.2. *(Variational inequality of second kind) Let $F = F_0 + F_1$ where F_0 is differentiable and F_1 is convex on $\mathcal{C}$. Assume $x^* \in \mathcal{C}$ is a minimizer of* (2.3)*. The necessary optimality for* (2.3) *is given by*

$$F_0'(x^*)(x - x^*) + F_1(x) - F_1(x^*) \ge 0 \quad \textit{for all } x \in \mathcal{C}. \tag{2.7}$$

Proof. Since F_1 is convex,

$$F_1(x^* + t\,(x - x^*)) - F_1(x^*) \le t\,(F_1(x) - F_1(x^*))$$

for all $x \in \mathcal{C}$ and $t \in [0, 1]$. Thus for $t \in (0, 1]$ we have

$$\begin{aligned} 0 &\le \frac{F(x^* + t(x - x^*)) - F(x^*)}{t} \\ &= \frac{F_0(x^* + t(x - x^*)) - F_0(x^*) + F_1(x^* + t(x - x^*)) - F_1(x^*)}{t} \\ &\le \frac{F_0(x^* + t(x - x^*)) - F_0(x^*)}{t} + F_1(x) - F_1(x^*). \end{aligned}$$

Letting $t \to 0$ we obtain (2.7). □

This encompasses a great deal of examples and applications (e.g., see Section 2.2.4). In the next two sections we introduce two problems for which the theorems work.

2.2.3 Deconvolution Problem

In this section we discuss the case of non-smooth cost functional $F(y)$ in (2.1). For example, consider the *deconvolution problem* for the convolution K defined by

$$y(x) = Ku(x) = \int_\Omega k(x, \omega) u(\omega)\, d\omega$$

where Ω is a bounded open subset of R^d and $u(\omega)$ represents an image and $k \ge 0$ is a convolution kernel, i.e., K is a convolution operator on $L^2(\Omega)$. The deconvolution problem is to determine u from observation y, where y may contain an additive noise.

It is a very ill-posed problem since K is compact and thus the variational formulation is used to construct a robust and accurate reconstruction u from y. The variational formulation is

$$\min\ F(u) = \frac{1}{2}|Ku - y|^2 + \int_\Omega \Big(\frac{\alpha}{2}|(\nabla u)(\omega)|^2 + \beta\,|u(\omega)|\Big)\, d\omega \tag{2.8}$$

subject to pointwise constraint $u(\omega) \in \widetilde{U}$, a closed convex set in R^m. The first term in (2.8) is the least squares fitting (a fidelity) and the last two terms are for regularizations on u. The first regularization term is to regularize variation of u in terms of the gradient ∇u and the second regularization term is for L^1 sparsity measure of u. Parameters $\alpha > 0$ and $\beta > 0$ are Tikhonov regularization parameters [29]. The second regularization term is not smooth on $H^1(\Omega)$.

2.2.4 Inverse Medium Problem

Also, consider an *inverse medium problem*. Consider the equation for $(y, u) \in H_0^1(\Omega) \times L^2(\Omega)$

$$-\Delta y + u\, y = 0 \ \text{ in } \Omega, \quad \frac{\partial y}{\partial \nu} = g \ \text{ at boundary } \partial\Omega \tag{2.9}$$

for equality constraint $E(y, u) = 0$, where u is the absorption coefficient and g is an applied current at the boundary $\partial\Omega$. We observe the voltage $z = y$ at the boundary $\partial\Omega$. The inverse medium problem is to determine u from the observation z. The problem is very ill-posed and we use a variational formulation

$$\min \ F(y) + H(u) = \frac{1}{2}\int_{\partial\Omega} |y - z|^2\, ds + \int_{\Omega} (\frac{\alpha}{2}|(\nabla u)(\omega)|^2 + \beta\, |u(\omega)|)\, d\omega \tag{2.10}$$

subject to the equality constraint (2.9) and the inequality constraint $u \geq 0$ pointwise. Thus we consider the constrained minimization of the form (2.2):

$$\min \ F(y) + H(u) \quad \text{subject to} \quad E(y, u) = 0, \ u \in \mathcal{C}.$$

The necessary condition (system for (y^*, u^*, λ)) is given by

$$\begin{aligned} &F'(y^*) + E_y(y^*, u^*)^*\lambda = 0 \\ &\langle E_u(y^*, u^*)^*\lambda, u - u^*\rangle + H(u) - H(u^*) \geq 0 \text{ for all } u \in \mathcal{C}. \end{aligned} \tag{2.11}$$

Similarly, the SP step (e.g., see in Chapter 3) has

$$\begin{aligned} &F'(y^+) + E_y(y, u)^*\lambda = 0 \\ &\langle E_u(y, u)^*\lambda, u^+ - u\rangle + H(u^+) - H(u) \geq 0 \text{ for all } u \in \mathcal{C}, \end{aligned} \tag{2.12}$$

which is a variational inequality for (y^+, u^+, λ).

2.3 Lagrange Multiplier Theory

In this section we discuss the Lagrange multiplier theory for the constrained optimization (2.1). The multiplier theory is essential for analyzing, developing algorithms, and solving (2.1). The Lagrange multiplier theory is the basis of the Sequential Programming method developed in Chapter 3.

We describe the Lagrange multiplier theory for the optimization problems (2.1). Define the

Lagrange functional

$$L(y, \lambda, \mu) = F(y) + \langle \lambda, E(y) \rangle + \langle \mu, G(y) \rangle, \tag{2.13}$$

for $y \in X$, $\lambda \in Y^*$, and $\mu \in Z^*$, where Y^* and Z^* are the (strong) dual spaces of Y and Z, respectively. The idea of the multiplier theory is that one can state the necessary optimality condition for a minimizer $y^* \in X$ by stating there exists a multiplier $\lambda \in Y^*$ (for simplicity we assume equality constraint only and $\mathcal{C} = X$) such that

$$L_y(y^*, \lambda) = 0, \quad E(y^*) = 0,$$

i.e.,

$$\begin{pmatrix} F'(y^*) + E'(y^*)^* \lambda \\ \\ E(y^*) \end{pmatrix} = 0, \tag{2.14}$$

which is a system of equations for (y^*, λ). Most algorithms for determining a minimizer y^* use the saddle point system (2.14) for (y^*, λ) as in Section 3.6.

For the general case (2.1), we introduce the constraint qualifications in terms of the regular point condition at a minimizer $y^* \in \mathcal{C}$:

$$0 \in \text{int} \left\{ \begin{pmatrix} E'(y^*)(y - y^*) \\ G'(y^*)(y - y^*) - K + G(y^*) \end{pmatrix} : y \in \mathcal{C} \right\}, \tag{2.15}$$

where K is the closed convex (negative) cone defined by

$$K = \{ z \in Z : z \leq 0 \},$$

and we assume F, E, G are C^1.

It follows from [48, 29] that the following necessary optimality holds at $y^* \in \mathcal{C}$. Assuming the regular point condition (2.15) at a minimizer y^*, there exist Lagrange multipliers $\lambda \in Y^*$, $\mu \in Z^*$ such that the necessary optimality holds

$$\begin{aligned} (F'(y^*) + E'(y^*)^* \lambda + G'(y^*)^* \mu, y - y^*) \quad &\geq 0 \quad \text{for all } y \in \mathcal{C} \\ E(y^*) \quad &= 0 \\ G(y^*) \leq 0, \quad \mu \geq 0, \quad \langle \mu, G(y^*) \rangle \quad &= 0. \end{aligned} \tag{2.16}$$

2.3.1 Equation Form of Complementarity Condition

In this section we write the complementarity condition, the condition in (2.16)

$$G(y^*) \le 0, \quad \mu \ge 0, \quad \langle \mu, G(y^*) \rangle = 0, \tag{2.17}$$

as an equation. Let $Z = L^2(\Omega)$. That is, we claim that (2.17) is equivalent to

$$\mu = \max(0, \mu + c\, G(y)) \tag{2.18}$$

where $c > 0$ is arbitrary, and the max operation is pointwise.

Note that if $\mu + c\, G(y) > 0$ then from (2.18) we have

$$\mu = \mu + c\, G(y) \quad \Longrightarrow \quad G(y) = 0 \text{ and } \mu > 0.$$

If $\mu + c\, G(y) \le 0$, then

$$\mu = 0 \text{ and } G(y) \le 0.$$

This means (2.18) is equivalent to (2.17).

Also (2.17) is the basis for the primal-dual active set method for the inequality constraint optimization (2.1). That is, given a current iterate (y, μ) define the active set

$$\mathcal{A} = \{\omega \in \Omega : (\mu + cG(y))(\omega) > 0\},$$

and the inactive set

$$\mathcal{I} = \{\omega \in \Omega : (\mu + cG(y))(\omega) \le 0\}.$$

According to the above discussion, the next iterate (y^+, μ^+) satisfies

$$\mu^+ = 0 \text{ on } \mathcal{I} \quad \text{ and } \quad G(y^+) = 0 \text{ on } \mathcal{A}.$$

We state the primal-dual active set method for the quadratic programming

$$\min \quad \frac{1}{2}(Ay, y) - (b, y) \quad \text{ subject to } \quad Gy \le c$$

where A is a positive self-adjoint operator on a Hilbert space X.

Primal-Dual Active Set Method

1. Initialize $\mu_1 = 0$, and y_1 by $Ay_1 = b$.

2. Define the active and inactive sets at the kth iterate

$$\mathcal{A}_k = \{\omega \in \Omega : (\mu_k + G(y_k))(\omega) > 0\},$$

$$\mathcal{I}_k = \{\omega \in \Omega : (\mu_k + G(y_k))(\omega) \leq 0\}.$$

3. Solve for (y_{k+1}, μ_{k+1}):

$$\mu_{k+1} = 0 \text{ on } \mathcal{I}_k \quad \text{and} \quad G(y_{k+1}) = 0 \text{ on } \mathcal{A}_k,$$

and

$$Ay_{k+1} + G^*\mu_{k+1} = b.$$

4. Check convergence. Otherwise, set $k = k + 1$ and return to Step 2.

The convergence of the primal-dual active set method has been investigated [27].

2.4 Optimal Control Problem

In this section we discuss the optimal control problem as an example of contrained optimization (2.2). We derive the necessary condition for an optimal control using the Lagrange multiplier theory. The necessary optimality condition is the form of a two point boundary value problem (i.e., a special case of the saddle point problem (2.14) in Section 2.3). The optimal control problem (Bolza form) is to find an admissible control in U_{ad} that minimizes the cost functional (performance criterion). The optimal control problem is an important example of constrained optimization with control.

The optimal control problem (Bolza form) is

$$\min \quad \int_0^T f^0(t, x, u)\, dt + g(x(T)) \tag{2.19}$$

subject to the dynamical constraint

$$\frac{d}{dt}x(t) = f(t, x(t), u(t)), \quad x(0) = x_0 \in R^n \tag{2.20}$$

and the control constraint

$$u(t) \in U, \text{ a closed convex set in } R^m, \text{ a.e. in } [0, T].$$

The variable $x \in X = H^1(0, T; R^n)$ is the *state* and $u \in L^2(0, T; R^m)$ is the *control*. U is the

pointwise constraint set for u. The functional $f^0(t,x,u)$ is for a running cost, and g is for a terminal cost. The function $f : R^+ \times R^n \times U \to R^n$ is the dynamical model. $U_{ad} = \{u \in L^2(0,T;R^m) : u(t) \in U\}$ is the set of admissible controls. Thus, the first term of (2.19) is a running cost and the second term is a terminal cost at $t = T$.

First, we have a well-posedness assumption on (2.20), i.e., for existence and uniqueness of solutions and the continuity of the solution map $(x_0, u) \in R^n \times L^1(0,T;U) \to x = x(t,x_0,u) \in C(0,T;R^m)$. In order to have well-posedness of the optimal control problem we introduce the the following sufficient conditions (see also Section 6.1.1).

1. The set of admissible controls is integrable, i.e.,

$$U_{ad} = \left\{u \in L^1(0,T;R^m) : u(t) \in U\right\}.$$

2. There exists a solution to the control dynamics (2.20), and there exists a solution to the optimal control problem (2.19) subject to (2.20) and $u \in U_{ad}$.

3. Assume the following for ω-dissipativeness in x (2.21) and for control growth (2.22):

$$(f(t,x,u) - f(t,y,u), x - y) \le \omega |x-y|^2 \quad \text{for all } \ u \in U \tag{2.21}$$

and moreover that either $U \subset R^m$ is bounded or that $f^0(t,0,u) \ge c_1 |u|^2$ and

$$(f(t,x,u), x) \le \omega |x|^2 + c_2 |u|^2 \tag{2.22}$$

for constants $\omega,\ c_1,\ c_2 > 0$ independent of $x,\ y \in R^n$ and $u \in U$.

4. Define the Hamiltonian $\mathcal{H} : R^+ \times R^n \times R^m \times R^n \to R$

$$\mathcal{H}(t,x,u,\lambda) = f^0(t,x,u) + \langle \lambda, f(t,x,u) \rangle.$$

For each fixed t, x, λ assume $\mathcal{H}$ admits a unique minimizer over $u \in U$ denoted by $\Psi(x,\lambda)$.

5. Assume that f^0, g, and f are C^1 with f^0 and g bounded from below.

Now let

$$E(x,u) = f(t,x,u) - \frac{d}{dt}x(t), \quad u \in \mathcal{C} = U.$$

Then (2.19)–(2.20) is a special case of the control form of constrained optimization problem (2.2) with $X = H^1(0,T;R^n) \times L^2(0,T;R^m)$.

The optimal control problem (2.19)–(2.20) is widely used in the general area of sciences and engineering and it is a cornerstone of mathematical modeling sciences in the mid to last century.

It contains the Lagrange problem (running cost only) and the Mayer problem (terminal cost only).

2.4.1 Necessary Optimality

In this section we derive the necessary optimality by the Lagrange multiplier theory for the optimal control problem (2.19)–(2.20).

First, we have the Lagrange functional

$$L(x,u,\lambda) = \int_0^T f^0(t,x,u)\,dt + g(x(T)) + \int_0^T (f(t,x,u) - \frac{d}{dt}x(t), \lambda(t))\,dt. \tag{2.23}$$

Thus,

$$L_x(x,u,\lambda)(h) = \int_0^T (f_x^0(t,x,u), h(t))\,dt + (g'(x(T)), h(T)) + \int_0^T (f_x(t,x,u)h(t) - \tfrac{d}{dt}h(t), \lambda(t))\,dt$$

$$L_u(x,u,\lambda)(v) = \int_0^T (f_u^0(t,x,u), v(t))\,dt + \int_0^T (f_u(t,x,u)v(t), \lambda(t))\,dt.$$

By integration by parts

$$L_x(x,u,\lambda)(h) = \int_0^T \left(\int_t^T (f_x(s,x,u), \lambda(s)) + f_x^0(s,x,u)\,ds - \lambda(t) + g'(x(T)), \frac{d}{dt}h(t) \right) dt.$$

Thus $L_x(x,u,\lambda)(h) = 0$ for all $h \in H_0^1(0,T;R^n)$ implies

$$\lambda(t) = g'(x(T)) + \int_t^T (f_x(s,x,u), \lambda(s)) + f_x^0(s,x,u)\,ds,$$

which implies $\lambda \in H^1(0,T;R^n)$ is absolutely continuous and the Lagrange multiplier $\lambda(t)$ for the equality constraint $E(x,u) = 0$ satisfies the adjoint equation

$$-\frac{d}{dt}\lambda(t) = f_x(t,x,u)^t\lambda + f_x^0(t,x,u), \quad \lambda(T) = g'(x(T)).$$

From $L_u(x,u,\lambda)(v-u) \geq 0$ for all $v \in U_{ad}$ it follows that for an optimal pair (x^*,u^*)

$$\mathcal{H}(t,x^*,u^*) \leq \mathcal{H}(t,x^*,v) \text{ for all } v \in U_{ad},$$

where $\mathcal{H}$ is the Hamiltonian defined by

$$\mathcal{H}(t,x,u) = f_u^0(t,x,u) + (\lambda, f_u(t,x,u)).$$

In summary we have the necessary optimality as a system of equations for (x^*, u^*, λ):

$$\begin{aligned}
&\frac{d}{dt}x^*(t) = f(t, x^*, u^*), \quad x(0) = x_0 \\
&-\frac{d}{dt}\lambda(t) = f_x(t, x^*, u^*)^t\lambda + f_x^0(t, x^*, u^*), \quad \lambda(T) = g'(x^*(T)) \\
&\mathcal{H}(t, x^*, u^*) \le \mathcal{H}(t, x^*, v) \text{ for all } v \in U.
\end{aligned} \tag{2.24}$$

If $U = R^m$ and f^0 is C^1 then the optimality condition (the last equation in (2.24)) for u is equivalent to

$$f_u^0(t, x^*, u^*) + f_u(t, x^*, u^*)^t\lambda = 0, \tag{2.25}$$

which is equivalent to $(\mathcal{H}(t, x^*, u^*), \lambda) =$ a constant. But, in general the optimality condition reduces to

$$u^*(t) = \Psi(t, x^*, \lambda).$$

Thus the necessary optimality system (2.24) becomes the so-called two point boundary value problem for (x^*, λ):

$$\begin{aligned}
&\frac{d}{dt}x^*(t) = f(t, x^*, u^*), \quad u^*(t) = \Psi(t, x^*(t), \lambda(t)) \text{ and } x(0) = x_0 \\
&-\frac{d}{dt}\lambda(t) = f_x(t, x^*, u^*)^t\lambda + f_x^0(t, x^*, u^*), \quad \lambda(T) = g'(x^*(T)).
\end{aligned} \tag{2.26}$$

Chapter 3

Sequential Programming Method and Theory

In this chapter we introduce the Sequential Programming method for solving constrained optimization problems (2.1) and (2.2). There are many popular methods for the nonlinear programming which has equality and inequality constraints. For example, we have the projected gradient method for constrained minimization and the SQP (Sequential Quadratic Programming) method. All of the methods use the Lagrange calculus and result in iterative methods that converge to a minimizer. The gradient method involves the two steps, i.e., the primal equality constraint solver and the adjoint equation solver for the multiplier λ, and then determines a gradient of the composite cost functional by evaluating a derivative of the Lagrange functional (2.13). We compute the gradient and use it for the projected gradient step, which requires a line search algorithm. SQP uses a quadratic model for the Lagrange functional and solves the necessary optimality system (2.14) for (y, λ) by Newton-like method. Thus, it involves the curvature information of the Lagrange functional and a system solver. For convergence the gradient method in general requires a line search algorithm [48, 29].

Our proposed method, i.e., the Sequential Programming (SP) method, is a middle ground between the gradient method and SQP. The SP method involves a sequence of linearized equality constraint optimizations, i.e., we linearize the equality constraint at the current iterate and then solve the optimization problem (no change in cost) subject to the linearized constraint. We must have a good solver for the resulting saddle point problem for the linearized equality constraint problem. We develop specific methods for the saddle point problems in various cases, inequality constraint and non-smooth cost. We also describe its implementation in detail. The SP method is very flexible for adapting to cases with non-smoothness and inequality constraints.

We analyze the convergence and convergence rate of the SP method. The basic SP method is of the first order with a good convergence rate. SP is guaranteed to converge under certain

conditions that we describe in Section 3.4.1. We also introduce the second order variant of SP in Section 3.4.2. In summary the SP method opens up the possibility for the large scale and non-smooth optimization problems. First we introduce the Lagrange calculus method.

3.1 Lagrange Calculus

The Lagrange multiplier theory described in Section 2.3 is the basis for most of the constrained optimization algorithm. In this section we discuss the so-called Lagrange calculus for the equality constrained optimization

$$\min \quad F(x) + H(u) \quad \text{subject to} \quad E(x,u) = 0, \;\; u \in \mathcal{C} \tag{3.1}$$

over $(x,u) \in X \times U$. Define the Lagrangian

$$L(x,u,\lambda) = F(x) + H(u) + \langle \lambda, E(x,u) \rangle.$$

We assume the equation $E(x,u) = 0$ has a continous solution branch $x = \Phi(u)$ in a neighborhood of $(\bar{x},\bar{u})$. Then we define the composite cost functional

$$J(u) = F(\Phi(u)) + H(u) \quad \text{on } u \in U. \tag{3.2}$$

The Lagrange calculus evaluates the implicit derivative of $F(x) + H(u)$ by the Lagrange multiplier theory.

Theorem 3.1. *(Lagrange) Let $(\bar{x},\bar{u})$ be a solution to $E(x,u) = 0$. Assume $E_x(\bar{x},\bar{u}) : X \to Y$ has a bounded inverse. Then there exists a C^1 solution branch $x = \Phi(u)$ to $E(x,u) = 0$ in a neighborhood of $(\bar{x},\bar{u})$. Assume multiplier $\lambda \in Y^*$ satisfies the adjoint equation*

$$E_x(\bar{x},\bar{u})^*\lambda + F'(\bar{x}) = 0. \tag{3.3}$$

Then

$$\frac{\partial}{\partial u} J(\bar{u}) = L_u(\bar{x},\bar{u},\lambda) = H'(\bar{u}) + E_u(\bar{x},\bar{u})^*\lambda. \tag{3.4}$$

Proof. By the implicit function theorem [66] there exists a C^1 solution $x = \Phi(u)$ in a neighborhood of $(\bar{x},\bar{u})$. By the chain rule (dot denotes differentiation with respect to u)

$$\frac{\partial}{\partial u} J(\bar{u})(\dot{u}) = F'(\bar{x})(\dot{x}) + H'(\bar{u})(\dot{u})$$

where

$$E_x(\bar{x},\bar{u})(\dot{x}) + E_u(\bar{x},\bar{u})(\dot{u}) = 0. \tag{3.5}$$

From the adjoint equation (3.3) and (3.5)

$$F'(\bar{x})(\dot{x}) = -\langle \lambda, E_x(\bar{x},\bar{u})(\dot{x})\rangle = \langle \lambda, E_u(\bar{x},\bar{u})(\dot{u})\rangle,$$

which implies the implicit derivative (3.4). □

Remark If (x^*, u^*) is a minimizing pair for (3.1), it follows from the Theorem 3.1 that (x^*, u^*, λ) satisfy the necessary optimality system:

$$E(x^*, u^*) = 0$$

$$E_x(x^*, u^*)^*\lambda + F'(x^*) = 0$$

$$H'(u^*) + E_u(x^*, u^*)^*\lambda = 0.$$

In general, we have

Corollary 3.1. *Assume $u \to H(u)$ is convex on $\mathcal{C}$. If $u^* \in \mathcal{C}$ is a minimizer of (3.1), then*

$$H(u) - H(u^*) + \langle \lambda, E_u(x^*, u^*)(u - u^*)\rangle \geq 0 \quad \textit{for all } u \in \mathcal{C}. \tag{3.6}$$

Proof. For $u \in \mathcal{C}$ and $t \in (0, 1)$, let $u_t = \bar{u} + t\,v$ with $v = u - \bar{u}$ and $x_t = \Phi(u_t)$. Then, we have

$$0 = \langle \lambda, E(x_t, u_t) - E(\bar{x},\bar{u})\rangle = \langle \lambda, E_x(\bar{x},\bar{u})(x_t - \bar{x}) + E_u(\bar{x},\bar{u})(u_t - \bar{u})\rangle + o(|t|), \tag{3.7}$$

and from the adjoint equation

$$F'(\bar{x})(x - \bar{x}) = -\langle \lambda, E_x(\bar{x},\bar{u})(x - \bar{x})\rangle. \tag{3.8}$$

Thus, from (3.7)–(3.8)

$$\begin{aligned} J(u_t) - J(\bar{u}) &= F(x_t) - F(\bar{x}) + H(u_t) - H(\bar{u}) = F'(\bar{x})(x_t - \bar{x}) + H(u_t) - H(\bar{u}) + o(|t|) \\ &= \langle \lambda, E_u(\bar{x},\bar{u})(u_t - \bar{u})\rangle + H(u_t) - H(\bar{u}) + o(|t|), \end{aligned}$$

and

$$\lim_{t\to 0} \frac{J(u_t) - J(\bar{u})}{t} \leq \langle \lambda, E_u(\bar{x},\bar{u})(u - \bar{u})\rangle + H(u) - H(\bar{u})$$

for all $u \in \mathcal{C}$. If we let $\bar{u} = u^*$, then we obtain the optimality condition (3.6). □

Remark In the proof of the Corollary 3.1 it suffices to assume $x_t = \Phi(u_t)$ exists for $E(x_t, u_t) = 0$

and the adjoint equation (3.8) has a solution λ and the following estimates hold:

$$F(x_t) - F(\bar{x}) - F'(\bar{x})(x_t - \bar{x}) = o_1(|t|)$$

$$\langle \lambda, E(x_t, u_t) - E(\bar{x}, \bar{u}) - E_x(\bar{x}, \bar{u})(x_t - \bar{x}) - E_u(\bar{x}, \bar{u})(u_t - \bar{u}) \rangle = o_2(|t|).$$

Of course, if F is twice differentiable at $\bar{x}$ and E is twice differentiable at $(\bar{x}, \bar{u})$, then the Remark follows under the assumption of the implicit function theory.

3.2 Gradient Method

One can apply the implicit (Lagrange) calculus described in Section 3.1 for the gradient method for the constrained minimization (3.1).

Algorithm (Projected Gradient Method)

1. Pick $u_1 \in \mathcal{C}$ and let $k = 1$.
2. Solve $E(x_k, u_k) = 0$ for x_k, given u_k.
3. Solve the adjoint equation $E_x(x_k, u_k)^* \lambda_k + F'(x_k) = 0$ for λ_k.
4. Evaluate the gradient $g_k = H'(u_k) + E_u(x_k, u_k)^* \lambda_k$.
5. Perform a line search for stepsize $\alpha > 0$

$$u_{k+1} = \mathrm{Proj}_{\mathcal{C}}(u_k - \alpha\, g_k).$$

6. Check convergence. Otherwise set $k = k + 1$ and return to Step 2.

The gradient method consists of forward solution for x_k given $u_k \in \mathcal{C}$ and adjoint solution for λ_k (given u_k and x_k) and line search in $\alpha > 0$ for update u_{k+1}. It is convergent under conditions of sufficient descent [3, 65] but the convergence may be very slow. For example, we consider the unconstrained minimization of $F(x) = \frac{1}{2}|Ax - b|^2_{R^n}$ for $x \in R^n$. The convergence rate is given by $1 - \frac{1}{\rho}$ where $\rho > 1$ is by the square of the condition number of A (i.e., the ratio of the largest and smallest singular values of A). If the condition number is large we have no convergence practically. Thus, we may apply the preconditioned gradient Pg_k so that the condition number is improved [13]. For example the preconditioning can be done by an incomplete Newton's method and we can utilize the the inner loop of SP as a preconditioner.

3.3 Sequential Quadratic Programming (SQP)

The Sequential Quadratic Programming involves a sequence of quadratic constrained problems. Let L be the Lagrangian for (2.1) with $y = (x, u)$ and cost $F(x) + H(u)$:

$$L(x, u, \lambda, \mu) = F(x) + H(u) + \langle \lambda, E(x, u) \rangle + \langle \mu, G(x, u) \rangle.$$

We make a quadratic model for $L(x, u, \lambda, \mu)$ in (x, u). For example, we use the second order Taylor approximation at $(x_c, u_c, \lambda_c, \mu_c)$ as

$$\mathcal{L} = L(x_c, u_c, \lambda_c, \mu_c) + (\nabla L(x_c, u_c, \lambda_c, \mu_c), \delta y) + \frac{1}{2}(L''(x_c, u_c, \lambda_c, \mu_c)\delta y, \delta y), \tag{3.9}$$

where $\delta y = (x - x_c, u - u_c)$, and we linearize the constraints

$$\begin{aligned} \mathcal{E}_c(x, u) &= E(x_c, u_c) + E_x(x_c, u_c)(x - x_c) + E_u(x_c, u_c)(u - u_c) = 0 \\ \mathcal{G}_c(x, u) &= G(x_c, u_c) + G_x(x_c, u_c)(x - x_c) + G_u(x_c, u_c)(u - u_c) \leq 0. \end{aligned} \tag{3.10}$$

Each SQP step solves the quadratic programming:

$$\min \mathcal{L}(x, u, \lambda_c, \mu_c)$$

subject to the linearized constraints (3.10) and the quadratic model (3.9) over (x, u), with $u \in \mathcal{C}$. By the Lagrange multiplier theory (2.16)–(2.18) we obtain the system for $(x^+, u^+, \lambda^+, \mu^+)$

$$\begin{cases} E(x_c, u_c) + E_x(x_c, u_c)(x - x_c) + E_u(x_c, u_c)(u - u_c) = 0 \\ \\ F''(x_c)(x - x_c) + F'(x_c) + E_x(x_c, u_c)^*\lambda + G_x(x_c, u_c)^*\mu = 0 \\ \\ H''(u_c)(u - u_c) + E_u(x_c, u_c)^*\lambda + G_u(x_c, u_c)^*\mu = 0 \\ \\ \mu = \max(0, \mu + \tilde{c}\, \mathcal{G}_c(x, u)). \end{cases} \tag{3.11}$$

The SQP update for each component is

$$u = \mathrm{Proj}_{\mathcal{C}}(u_c + \alpha\,(u^+ - u_c))$$

$$x = x_c + \alpha\,(x^+ - x_c)$$

$$\lambda = \lambda_c + \alpha\,(\lambda^+ - \lambda_c)$$

with a stepsize $\alpha > 0$ (determined by a line search). In the case of equality constraint alone, (3.11) reduces to

$$\begin{aligned} &E(x_c, u_c) + E_x(x_c, u_c)(x - x_c) + E_u(x_c, u_c)(u - u_c) = 0 \\ &F''(x_c)(x - x_c) + F'(x_c) + E_x(x_c, u_c)^*\lambda = 0 \\ &H''(u_c)(u - u_c) + E_u(x_c, u_c)^*\lambda = 0, \end{aligned} \tag{3.12}$$

which is equivalent to Newton's method applied to the necessary optimality conditions (2.14) on (x, u, λ):

$$\nabla L(x, u, \lambda) = 0,$$

where the ∇ is the derivative with respect to every component (x, u, λ). Moreover $(x^+ - x_c, u - u_c, \lambda - \lambda_c)$ is the preconditioned gradient (Newton direction) since

$$L''(x_c, u_c, \lambda)(x^+ - x_c, u - u_c, \lambda - \lambda_c) + \nabla L(x^+ - x_c, u - u_c, \lambda - \lambda_c) = 0.$$

Pros: (3.12) is a linear saddle point problem for (x, u, λ). If SQP converges, it converges quadratically under proper conditions. But also we can check the type of constrained problems in terms of $L''(x^*, u^*, \lambda_c)$.

Cons: We must solve the linear system for (x, u, λ). The system is symmetric, but can be indefinite for $L''(x_c, u_c, \lambda_c)$. We have to compute the second derivative $L''(x_c, u_c, \lambda_c)$. It is not globally convergent in general (we must do a line search $\alpha > 0$).

Remedy of Cons: We use BFGS for sequential approximations of $L''(x_c, u_c, \lambda_c)$. If Deniss-More condition is satisfed it is super-linear convergence. One can use incomplete Newton step solver or the Newton step is performed by regularized least squares

$$\min \quad |L''(x_c, u_c, \lambda_c)(x - x_c, u - u_c, \lambda - \lambda_c) + \nabla L(x_c, u_c, \lambda_c)| + \frac{\beta}{2} |(x - x_c, u - u_c, \lambda - \lambda_c)|^2$$

for a proper choice of $\beta > 0$.

We now introduce the Sequential Programming (SP) method and will show that it may remedy many aspects of the drawbacks of SQP and the gradient method.

3.4 Sequential Programming

Sequential Programming (SP) is a method to solve the constrained minimization

$$\min \quad F(y) \quad \text{subject to} \quad E(y) = 0, \quad y \in \mathcal{C}, \tag{3.13}$$

which can include an inequality constraint by letting $\mathcal{C} = \{y \in X : G(y) \leq 0\}$. We linearize the equality constraint at the current iterate $y_c \in \mathcal{C}$, i.e., the tangent equation

$$E'(y_c)(y - y_c) + E(y_c) = 0.$$

We may use the Broyden update for the Jacobian E' [42, 9]. Then, we solve the sequence of linearized equality constraint problems:

$$\min \quad F(y) \quad \text{subject to} \quad E'(y_n)(y - y_n) + E(y_n) = 0, \quad y \in \mathcal{C}. \tag{3.14}$$

The necessary optimality condition for (3.14) is the form of the saddle point problem:

$$\begin{aligned} &(F'(y^+) + E'(y_n)^*\lambda, y - y^+) \geq 0, \quad y \in \mathcal{C} \\ &E'(y_n)(y^+ - y_n) + E(y_n) = 0. \end{aligned} \tag{3.15}$$

Let y^+ be a solution to (3.14) at the nth iterate. We update the current iterate y_n with a damped update

$$y_{n+1} = y_n + \alpha\,(y^+ - y_n), \tag{3.16}$$

where $\alpha \in (0, 1]$ is selected. In the initial steps of SP update we relax α to adjust the quadratic errors of the linearization; see our analysis of convergence in Section 3.4.1. Note that under-damping also keeps iterates in the convex set $\mathcal{C}$.

3.4.1 Convergence

In this section we present the convergence analysis of the SP method. Suppose $y^* \in \mathcal{C}$ is a solution to the original problem (3.13). Then y^* is a solution to the perturbed problem of (3.14):

$$\min \quad F(y) \quad \text{subject to} \quad E'(y_n)(y - y_n) + E(y_n) = \Delta_2, \quad y \in \mathcal{C} \tag{3.17}$$

where

$$\Delta_2 = E(y_n) - E(y^*) + E'(y_n)(y^* - y_n) \sim O(|y_n - y^*|^2).$$

In fact, the necessary optimality system for (3.13)

$$(F'(y^*) + E'(y^*)^*\lambda^*, y - y^*) \geq 0, \quad y \in \mathcal{C}$$

$$E(y^*) = 0$$

can be written alternatively as

$$(F'(y^*) + E'(y_n)^*\lambda^* - (E'(y_n)^* - E'(y^*)^*)\lambda^*, y - y^*) \geq 0, \quad y \in \mathcal{C}$$

$$E'(y_n)(y^* - y_n) + E(y_n) = E'(y_n)(y^* - y_n) + E(y_n) - E(y^*),$$

which is a perturbation of the necessary optimality system for (3.14), i.e.,

$$\begin{aligned} &(F'(y^*) + E'(y_n)^*\lambda^* - \Delta_1, y - y^*) \geq 0, \quad y \in \mathcal{C} \\ &E'(y_n)(y^* - y_n) + E(y_n) = \Delta_2 \end{aligned} \tag{3.18}$$

with perturbation (Δ_1, Δ_2) given by

$$\Delta_1 = (E'(y_n)^* - E'(y^*)^*)\lambda^*$$

$$\Delta_2 = E(y^*) - E(y_n) - E'(y_n)(y^* - y_n).$$

That is, the necessary optimality system for the original problem (3.13) is a perturbation of the necessary optimality system for the linearized equality constraint problem (3.14). Recall the SP step for (3.14)

$$\begin{aligned} &(F'(y^+) + E'(y_n)^*\lambda^+, y - y^+) \geq 0, \quad y \in \mathcal{C} \\ &E'(y_n)(y^+ - y_n) + E(y_n) = 0. \end{aligned} \tag{3.19}$$

Combining (3.19) with (3.18) we obtain

$$(F'(y^+) - F'(y^*) + E'(y_n)^*(\lambda^+ - \lambda^*) + \Delta_1, y^+ - y^*) \geq 0, \quad y \in \mathcal{C}$$

$$E'(y_n)(y^+ - y^*) = \Delta_2.$$

Assume the Lipschitz continuity of solution $y^+ - y^*$ with respect to perturbations Δ_1 and Δ_2:

$$|y^+ - y^*| \leq M_1 |\Delta_1| + M_2 |\Delta_2|. \tag{3.20}$$

That is, the solution y^* (and y^+) is a function of the perturbation (Δ_1, Δ_2). Then, for the damped update

$$y_{n+1} = y_n + \alpha(y^+ - y_n), \quad \alpha \in (0, 1],$$

we have by (3.20)

$$|y_{n+1} - y^*| \le (1-\alpha)|y_n - y^*| + \alpha|y^+ - y^*| \le (1-\alpha)|y_n - y^*| + \alpha(M_1 |\Delta_1| + M_2 |\Delta_2|).$$

That is,

$$|y_{n+1} - y^*| \le (1 - \alpha + \alpha\widetilde{M_1})|y_n - y^*| + \alpha\widetilde{M_2}|y_n - y^*|^2. \tag{3.21}$$

Thus, rate $(1 - \alpha + \alpha\widetilde{M_1}) < 1$ if $\widetilde{M_1} < 1$. The best convergence rate is achieved with $\alpha = 1$, but we need to bound the quadratic term by adjusting $\alpha \in (0, 1)$, i.e., we may need to choose $\alpha < 1$ so that $\alpha\widetilde{M_2}|y_n - y^*|^2$ is dominated by the first order error. Thus, for small perturbation (Δ_1, Δ_2), the SP iterates y_{n+1} converge to the optimal solution y^* with linear rate.

For example, we estimate $\widetilde{M_1}$ for the case where $\mathcal{C} = X$ and $F(y) = \frac{1}{2}(Qy, y)$. Then the necessary optimality condition for SP is

$$\begin{cases} Qy + E'(y_n)^*\lambda = 0 \\ \\ E'(y_n)(y - y_n) + E(y_n) = 0. \end{cases}$$

Define

$$G_n = \begin{pmatrix} Q & E'(y_n)^* \\ \\ E'(y_n) & 0 \end{pmatrix}.$$

Then

$$G_n \begin{pmatrix} y^+ - y^* \\ \\ \lambda^+ - \lambda^* \end{pmatrix} = \begin{pmatrix} -\Delta_1 \\ \\ \Delta_2 \end{pmatrix}. \tag{3.22}$$

Let $E_n = E'(y_n)$, $\delta y = y^+ - y^*$, and $\delta\lambda = \lambda^+ - \lambda^*$. Then (3.22) is written

$$\begin{cases} Q(\delta y) + E_n^*(\delta\lambda) & = -\Delta_1 \\ \\ E_n(\delta y) & = \Delta_2. \end{cases} \tag{3.23}$$

We solve the first equation of (3.23) for increment δy:

$$\delta y = Q^{-1}(-\Delta_1 - E_n^*(\delta\lambda)), \tag{3.24}$$

substitute into the second equation of (3.23):

$$-E_nQ^{-1}\Delta_1 - \Delta_2 - E_nQ^{-1}E_n^*(\delta\lambda) = 0,$$

and solve for increment $\delta\lambda$:

$$\delta\lambda = -(E_nQ^{-1}E_n^*)^{-1}(E_nQ^{-1}\Delta_1 + \Delta_2). \tag{3.25}$$

Then combining (3.24) and (3.25) we obtain

$$y^+ - y^* = \delta y = -Q^{-1}\Delta_1 + Q^{-1}E_n^*(E_nQ^{-1}E_n^*)^{-1}(E_nQ^{-1}\Delta_1 + \Delta_2).$$

Thus,

$$|y^+ - y^*| \le |Q^{-1}\Delta_1 - Q^{-1}E_n^*(E_nQ^{-1}E_n^*)^{-1}E_nQ^{-1}\Delta_1| + |Q^{-1}E_n^*(E_nQ^{-1}E_n^*)^{-1}\Delta_2|.$$

Observe that

$$P = Q^{-1} - Q^{-1}E_n^*(E_nQ^{-1}E_n^*)^{-1}E_nQ^{-1}$$

is the projection operator onto $\ker(E_n)$ (one can easily verify $E_nP = 0$). Thus,

$$y^+ - y^* = \mathrm{Proj}_{\ker(E_n)}\,\Delta_1 + Q^{-1}E_n^*(E_nQ^{-1}E_n^*)^{-1}\Delta_2.$$

We assume

$$|\,\mathrm{Proj}_{\ker(E_n)}\,\Delta_1| \le \gamma\,|y_n - y^*|.$$

Then we have

$$|y^{n+1} - y^*| \le \gamma\,|y_n - y^*| + c\,|y_n - y^*|^2,$$

which shows (3.21).

In many applications one can show that $\gamma < 1$. Moreover, the rate of convergence of the SP method is on the order of

$$1 - \alpha + \alpha\gamma = 1 + \alpha(\gamma - 1),$$

which can be less than 1 (contractive) if $\gamma < 1$.

Remarks (1) Suppose $F'(y^*)$ is small, e.g., the small residual case for the target optimization $F(y) = \frac{1}{2}|y - \bar{y}|^2$ where $\bar{y}$ is the desired state and thus $\gamma < 1$.
(2) If the constraint E is nearly linear in the direction of λ^*, then $|(E'(y^*)^* - E'(y_n))\lambda^*|$ is sufficiently small and $\gamma < 1$.
(3) Consider the case when $E(y) = Ax + f(x) + Bu$. Suppose A is self-adjoint, positive and f

is monotone Lipschitz, then one can estimate γ using PDE analysis.

3.4.2 Second Order Sequential Programming Method

In this section we introduce the second order version of SP (Sequential Programming) to achieve the quadratic convergence. In order to obtain a second order method it is necessary to incorporate the term $(E'(y_n)^* - E'(y^*)^*)\lambda^*$ to the update [31]. Thus we consider

$$\begin{aligned} \min \quad & F(y) + \langle\, \lambda_n, E(y) - (E'(y_n)(y - y_n) + E(y_n))\,\rangle \\ & \text{subject to } E'(y_n)(y - y_n) + E(y_n) = 0 \text{ and } y \in \mathcal{C}. \end{aligned} \tag{3.26}$$

Observe that the term $\langle \lambda_n, E(y) - (E'(y_n)(y-y_n) + E(y_n))\rangle$ can be understood as approximation to $\frac{1}{2}\langle \lambda_n, E''(y_n)(y - y_n, y - y_n)\rangle$.

The necessary optimality condition for (3.26) is given by

$$\begin{cases} (F'(y^+) + (E'(y^+)^* - E'(y_n)^*)\lambda_n + E'(y_n)^*\lambda^*, y - y^+) \geq 0 \text{ for all } y \in \mathcal{C} \\ \\ E'(y_n)(y^* - y_n) + E(y_n) = 0. \end{cases} \tag{3.27}$$

When compared to (3.15) this is the saddle point problem involving the linearized equation where the term $(E'(y)^* - E'(y_n)^*)\lambda_n$ is added. The necessary optimality condition (3.18) to (3.17) can be expressed to follow the structure of (3.27) as

$$\begin{cases} (F'(y^*) + (E'(y^*)^* - E'(y_n)^*)\lambda_n + E'(y_n)^*\lambda^* - \Delta_1, y - y^*) \geq 0 \text{ for all } y \in \mathcal{C} \\ \\ E'(y_n)(y^* - y_n) + E(y_n) = \Delta_2, \end{cases} \tag{3.28}$$

where

$$\Delta_1 = (E'(y^*)^* - E'(y_n)^*)(\lambda_n - \lambda^*) \text{ and } \Delta_2 = E'(y_n)(y^* - y_n) + E(y_n) - E(y^*). \tag{3.29}$$

Observe that there is now a $(\lambda_n - \lambda^*)$ term in the perturbation Δ_1. This gives

$$\Delta_1 \sim O(|y_n - y^*|\,|\lambda_n - \lambda^*|) \quad \text{and} \quad \Delta_2 \sim O(|y_n - y^*|^2).$$

The update according to (3.26) results in the following algorithm, which is locally quadratically convergent.

Algorithm (Second Order Sequential Programming)

1. Choose (y_0, λ_0). Set $n = 0$.

2. Given (y_n, λ_n), solve the saddle point problem (3.27) for (y^+, λ^+).

3. Update $(y_{n+1}, \lambda_{n+1}) = (y_n, \lambda_n) + \alpha\,((y^+, \lambda^+) - (y_n, \lambda_n))$. Iterate until convergence criterion is satisfied.

3.4.3 Fixed Point Formulation of Saddle Point Problem

Consider the case

$$E(x, u) = E_0(x) + Bu.$$

Then the saddle point problem (2.14) for SP step is written in stepwise:

$$\begin{aligned} x^+ &= x_n - (E_0'(x_n))^{-1}(Bu^+ + E_0(x_n)) \\ \lambda &= -(E_0'(x_n))^{-*}F'(x^+) \\ u^+ &= \Psi(p) = \mathrm{argmin}_{u\in\mathcal{C}}\{H(u) + (u, p)\}, \quad p = B^*\lambda \end{aligned} \tag{3.30}$$

where $u^+ = \Psi(B^*\lambda)$ solves the optimality condition:

$$H(\tilde{u}) - H(u^+) + (B^*\lambda, \tilde{u} - u^+) \geq 0 \text{ for all } \tilde{u} \in \mathcal{C}.$$

Thus, we obtain the fixed point formulation for $u^+ \in \mathcal{C}$

$$u^+ = \Psi(B^*\lambda), \tag{3.31}$$

where given u^+, $\lambda = \lambda(u^+)$ is determined by the first two equations of (3.30). If $H(u) = \frac{1}{2}(u, Ru)$ we have

$$u^+ = \mathrm{Proj}_{\mathcal{C}}(-R^{-1}B^*\lambda).$$

We use the nonlinear CG or CR method, Section 3.6.2, for (3.31). The specific case is developed as the reduced order CG method in Section 3.6.1.

3.4.4 Second Order Version

Suppose $F(y) = \frac{1}{2}(y, Qy)$ and $\mathcal{C} = X$. Then the saddle point problem (2.14) for (y^+, λ) is written in stepwise as

$$\begin{aligned} &y^+ = Q^{-1}((E'(y^+)^* - E(y_n)^*)\lambda_n - E'(y_n)^*\lambda = 0 \\ &\lambda = -(E'(y_n)Q^{-1}E'(y_n)^*)^{-1}(E(y_n) + E'(y_n)^*(y_n + Q^{-1}(E'(y^+)^* - E(y_n)^*)\lambda_n). \end{aligned} \tag{3.32}$$

That is, y^+ is a fixed point of

$$y^+ = -Q^{-1}((E'(y^+)^* - E(y_n)^*)\lambda_n + E'(y_n)^*\lambda) \tag{3.33}$$

where $\lambda = \lambda(y^+)$ is determined by the second equation of (3.32). We use the damped fixed point update

$$y_{n+1} = \alpha\, y^+ + (1 - \alpha)y_n, \tag{3.34}$$

where $\alpha > 0$ is selected such that it becomes a contraction mapping for (3.33).

3.4.5 Non-smooth Cost Case

Consider cost functional

$$F(y) = \frac{1}{2}(y, Qy) + F_1(y)$$

where F_1 is differentiable but not necessarily C^1. The saddle point problem (2.14) for (y^+, λ) is

$$\begin{aligned} &y^+ = Q^{-1}(-F_1'(y^+) - E'(y_n)^*\lambda) = 0 \\ &\lambda = -(E'(y_n)Q^{-1}E'(y_n)^*)^{-1}(E(y_n) + E'(y_n)^*(y_n) + Q^{-1}E'(y_n)F_1'(y^+)). \end{aligned} \tag{3.35}$$

That is, y^+ is a fixed point of

$$y^+ = -Q^{-1}(F_1'(y^+) + E'(y_n)^*\lambda) \tag{3.36}$$

where $\lambda = \lambda(y^+)$ is determined by the second equation of (3.35). We use the damped update

$$y_{n+1} = \alpha\, y^+ + (1 - \alpha)\, y_n.$$

Again, $\alpha > 0$ is selected such that it becomes a contraction mapping for (3.36).

3.4.6 Properties

Recall the necessary optimality system at a minimizer $y^* \in \mathcal{C}$:

$$\langle F'(y^*) + E'(y^*)^*\lambda, y - y^* \rangle \geq 0, \quad y \in \mathcal{C}, \quad E(y^*) = 0.$$

If we linearize the equality constraint at the current iterate y_n, we obtain

$$\langle F'(y^+) + E'(y_n)^*\lambda, y - y^+ \rangle \geq 0 \quad y \in \mathcal{C}, \quad E'(y_n)(y^+ - y_n) + E(y_n) = 0.$$

This is exactly the necesary optimality system for the SP method (3.19):

$$\min \quad F(y^+) \;\text{ subject to }\; E'(y_n)(y^+ - y_n) + E(y_n) = 0, \;\; y^+ \in \mathcal{C}.$$

Thus, the SP method is a partial linearization method of the necessary optimality system compared to the SQP method which fully linearizes the system (3.12). Also, we can use the quadratic model for $F'(y^+) \sim F'(y_n) + Q(y^+ - y_n)$. Then it becomes SQP method with no quadratic model for $E(y)$. Moreover, we describe the second order SP method to treat the $E(y)$ term sequentially. Thus, SP is indeed the middle ground.

3.5 Bilinear Control Problem and Partial SQP

Consider the specific case of (2.2), i.e., the coefficient control of the form

$$\min F(y) + H(u) \quad \text{ subject to } \quad Ay + B(u)\, y = b, \quad u \geq 0$$

where u is control, design variable and medium coefficient. For example, $u \geq 0$ is the absorption coefficient in the diffusive medium Ω for the dynamical constraint

$$-\Delta y + u(x)y = f, \quad x \in \Omega. \tag{3.37}$$

Consider the following medium identification problem. Find $u \geq 0$ from the boundary observation of $\bar{y}$ at $\partial\Omega$ with Neumann boundary condition $\frac{\partial}{\partial \nu} u = 0$. The problem is formulated as

$$\min \quad |y - \bar{y}|^2_{\partial\Omega} + \int_\Omega (\frac{\alpha}{2}|\nabla u|^2 + \beta\,|u|)\, d\omega$$

subject to (3.37). Similary, the coefficient control problem is to minimize

$$\min \quad |y - y_d|^2_{\Omega} + \int_\Omega (\frac{\alpha}{2}|\nabla u|^2 + \beta\,|u|)\, d\omega$$

subject to (3.37), where y_d is the desired diffusive field. For material design problem with respect to u we select a proper design performance as a cost functional. Thus, inverse medium, coefficient control, and material design problems are formulated the same but with different cost functional $F(y)$. A feature of the bilinear control problem is that the second derivative of a bilinear term such as $(Bu)y$ is a constant B. Thus computing and using the second derivative information (as is done for SQP) is affordable and can enhance the solution method. This results in the partial SQP method.

Consider the bilinear control problem with $E(x,u) = Ex + f(x) + (Bu)x = 0$, where $f(x)$ may be nonlinear. The necessary optimality for the SP step is

$$Ex + f'(x_c)(x - x_c) + f(x_c) + (Bu_c)(x - x_c) + (Bu)x_c = 0$$

$$Qx + (E + f'(x_c))^*\lambda + (Bu_c)\lambda = 0$$

$$u = -\tfrac{1}{\alpha}\,(B^*\lambda)x_c.$$

Partial SQP uses the second derivative of the bilinear term $(Bu)x$ and results in the following saddle point problem:

$$\begin{cases} Ex + f'(x_c)(x - x_c) + f(x_c) + (Bu_c)(x - x_c) + (Bu)x_c = 0 \\ \\ Qx + (E + f'(x_c))^*\lambda + (Bu_c)\lambda + (Bu)\lambda_c = 0 \\ \\ \alpha u + B^*(\lambda\, x_c + \lambda_c\, x) = 0. \end{cases}$$

Note that SQP would also include the second derivative term for $f(x)$, which can be expensive. Partial SQP remedies this by only doing SQP for the bilinear term.

3.6 Saddle Point Solver

In this section we discuss solvers for each SP step for determining $y^+ \in \mathcal{C}$ that minimizes

$$F(y) \quad \text{subject to} \quad E'(y_n)(y - y_n) + E(y_n) = 0, \quad y \in \mathcal{C}$$

at the nth iterate $y_n \in \mathcal{C}$. We use the necessary optimality condition (3.15)

$$\begin{cases} F'(y^+) + E'(y_n)^*\lambda = 0 \\ \\ E'(y_n)(y^+ - y_n) + E(y_n) = 0, \end{cases} \tag{3.38}$$

which is a saddle point problem for (y^+, λ). Thus, we describe solution methods to solve the saddle point problem (3.38) for (y^+, λ) in various cases.

3.6.1 Reduced Order CG

In order to solve the saddle point problem (3.38) for SP we have an efficient algorithm, especially for large scale problems. In this section we introduce the reduced order method based on the conjugate gradient (CG) method. Consider the specific case for (2.2)

$$E(x,u) = E(x) + Bu, \qquad F(x) + H(u) = \frac{1}{2}((Qx,x) + (Hu,u))$$

where $B \in \mathcal{L}(U,X)$ and thus E is linear in u. Q and H are positive self-adjoint operators on X and U, respectively. The saddle point problem (3.38) for SP at the current iterate (x_n, u_n) is

$$\begin{cases} E'(x_n)(x^+ - x_n) + E(x_n) + Bu^+ = 0 \\ \\ Qx^+ + E'(x_n)^*\lambda = 0 \\ \\ Hu^+ + B^*\lambda = 0, \end{cases} \tag{3.39}$$

where the triple (x^+, u^+, λ) is unknown. Now we describe how to reduce this system to a system just for u. From the first equation of (3.39)

$$x^+ = E'(x_n)^{-1}(-Bu^+ + E'(x_n)x_n - E(x_n)), \tag{3.40}$$

and from the second equation

$$\lambda = -E(x_n)^{-*}Qx^+. \tag{3.41}$$

Thus, from the third equation we obtain the reduced order equation for u^+

$$(H + B^*E'(x_n)^{-*}QE'(x_n)^{-1}B)u^+ = B^*E'(x_n)^{-*}QE'(x_n)^{-1}(E'(x_n)x_n - E(x_n)). \tag{3.42}$$

Notice that

$$L = (H + B^*E'(x_n)^{-*}QE'(x_n)^{-1}B)$$

is a positive self-adjoint operator on U, and moreover H is coercive and $B^*E'(x_n)^{-*}QE'(x_n)^{-1}B$ is compact. Thus, we use the conjugate gradient (CG) method to solve (3.42) for u^+. That is, we have reduced the size of the system to solve from a system for (x^+, u^+, λ) to a single equation in u^+. This can be advantageous especially for dealing with large scale problems.

Remarks (1) System (3.42) for u^+ is linear and well-posed, i.e., the condition number of L is not large.
(2) We do not need to use a preconditioner for solving system (3.42) since it is well-conditioned.
(3) The CG step requires we evaluate $B^*E'(x_n)^{-*}QE'(x_n)^{-1}Bu$ given $u \in U$, which involves the forward solution for x^+ by (3.40) followed by the adjoint solution for λ by (3.41).
(4) Thus, each CG step is identical to the gradient step described in Section 3.2, except that the forward equation is linear for SP. Comparatively, the gradient method solves the nonlinear equation $E(x^+, u_n) = 0$ for x^+ given u_n.
(5) We apply the "hot" start CG (i.e., we initialize the CG step by the previous iterate u_n). A few iterates of CG works practically since an SP solution sequence (x_n) is assumed convergent and L depends on x Lipschitz continuously.
(6) In summary, SP is very efficient because it is equivalent to CG, we do not need to precondition L, and we can use the "hot" start to proceed immediately from the previous iterate. Thus, the total number of CG steps needed is possibly much less than the number of steps needed for the gradient method to converge.

3.6.2 Preconditioned Conjugate Residual (CR) Method

In this section we introduce the preconditioned conjugate residual method for solving equations of the form $F(x) = Ax - b = 0$. We present a method for the case where F is nonlinear. To solve the nonlinear equation $F(x) = 0$ by the conjugate residual method, we make the identification of $F(x)$ with $Ax - b$ so that we can approximate the quantity Ar_k that is need to compute the Cauchy stepsize. That is, since the derivative of $Ax - b$ is A, we deduce that $F'(x) \approx A$ and thus Ar_k can be approximated by $F'(x)(r_k)$, the Gateaux derivative of F at x in the direction r_k. Since F is truly nonlinear, we approximate the derivative by the secant. Recall the saddle point problem for minimizing $F(y)$ subject to $E(y) = 0$, with $x = (y, \lambda)$

$$F(x) = \begin{pmatrix} F'(y) + E'(y)^*\lambda \\ E(y) \end{pmatrix} = Ax = \begin{pmatrix} Q & E^* \\ E & 0 \end{pmatrix} x = RHS$$

where A is self-adjoint but indefinite. The (preconditioned) CR method is the iterate method (like CG) with conjugate directions $\{p_k\}$ satisfying $(Ap_k, Ap_j) = 0,\ j < k$. It is based on the descent algorithm for

$$\frac{1}{2}|Ax - b|^2 = \frac{1}{2}(AAx, x) - (Ab, x) + const.$$

Algorithm (Nonlinear Conjugate Residual Method)

1. Calculate s_k for approximating Ar_k by

$$s_k = F(x_k + r_k) - F(x_k).$$

2. Update the direction by

$$p_k = r_k - \beta_k p_{k-1}, \qquad \beta_k = \frac{(s_k, q_{k-1})}{(q_{k-1}, q_{k-1})}.$$

3. Calculate

$$q_k = r_k - \beta_k q_{k-1}.$$

4. Update the solution by

$$x_{k+1} = x_k + \alpha_k p_k, \qquad \alpha_k = \frac{(r_k, s_k)}{(q_k, q_k)}.$$

5. Calculate the residual

$$r_{k+1} = F(x_{k+1}).$$

Remarks (1) The operator A may be indefinite for the CR method. Thus, it is a good iterative solver (like CG) for problems with system matrix that is not necessarily positive definite. For example, the saddle point problem for the incompressible Navier-Stokes control problem described in Section 5.3 has this property.
(2) The CR method has computational complexity on the order of the CR method [57]. CR is not the same as CG applied to the normal equations. Thus we do not lose performance due to the condition number of A being squared.

3.6.3 Inequality Case

We consider the constrained optimization problem with inequality constraint, i.e.,

$$\min J(x) \text{ subject to } Gx \leq c,\ y \in \mathcal{C} \tag{3.43}$$

with $\mathcal{C} = \{x \leq 0\}$, or $\mathcal{C}$ = box, e.g., $a_i \leq x_i \leq b_i$ component-wise, or $\mathcal{C} = \{Gx \leq c\}$ (affine constraint) with $G : X \to Z$. The necessary optimality condition for (3.43) is

$$\begin{aligned} &J'(x) + G^*\mu = 0 \\ &\mu \geq 0,\ \ Gx \leq c,\ \ (\mu, Gx - c) = 0. \end{aligned} \tag{3.44}$$

That is, at least one of μ and $Gx - c$ is zero coordinate-wise. Or equivalently

$$\mu = \max(0, \mu + \beta\,(Gx - c)).$$

Thus, we have an equation form of the necessary optimality (3.44) for (x, μ)

$$\begin{aligned} J'(x^*) + G^*\mu &= 0 \\ \mu &= \max(0, \mu + \beta\,(Gx - c)). \end{aligned} \tag{3.45}$$

First, the primal-dual equation is the necessary condition for (x, μ). From the complementarity condition: if $\mu+\beta\,(Gx-c) \geq 0$, then $\mu = \mu+\beta\,(Gx-c)$ and $Gx = c$, otherwise if $\mu+\beta\,(Gx-c) < 0$, then $\mu = 0$ (and $Gx < c$). Thus, given (x, μ), define active and inactive index sets:

$$\mathcal{A} = \{k : \mu + \beta\,(Gx - c) \geq 0\}$$

$$\mathcal{I} = \{i : \mu + \beta\,(Gx - c) < 0\}.$$

Now, we define the primal-dual active set method for SP.

Algorithm (Primal-Dual Active Set Method for SP)

1. Select x_0 and μ_0.

2. Define the active and inactive index sets:

$$\mathcal{A}_n = \{k : \mu_n + \beta\,(Gx_n - c) \geq 0\}$$

$$\mathcal{I}_n = \{i : \mu_n + \beta\,(Gx_n - c) < 0\}.$$

3. Solve system for (x^{n+1}, μ^{n+1})

$$J'(x^{n+1}) + G^*\mu^{n+1} = 0$$

$$Gx = c \text{ on } \mathcal{A}_k, \;\; \mu^{n+1} = 0 \text{ on } \mathcal{I}_k.$$

Remarks (1) If $J'(x) = Ax - b$ and $G = I$ and $c = 0$, then one can solve the system (3.45) by the partition of the coordinate by the active and inactive indices:

$$A_{\mathcal{I}\mathcal{I}} x_{\mathcal{I}} = b_{\mathcal{I}}$$

$$x_{\mathcal{A}} = 0, \quad \mu_{\mathcal{A}} = b_{\mathcal{A}} - A_{\mathcal{A}\mathcal{I}} x_{\mathcal{I}}.$$

This is the reduced system for $x_{\mathcal{I}}$ we need to solve.
(2) The primal-dual active method is trying to find the active indicies for $Gx \leq c$ sequentially based the complementarity condition (2.18) for (x, μ).
(3) In general we need to solve

$$\begin{pmatrix} A & G'_{\mathcal{A}} \\ \\ G_{\mathcal{A}} & 0 \end{pmatrix} \begin{pmatrix} x^{n+1} \\ \\ \mu_{\mathcal{A}}^{n+1} \end{pmatrix} = \begin{pmatrix} b \\ \\ c_A \end{pmatrix}$$

where $G_{\mathcal{A}} = G$ on $\mathcal{A}$.
(4) If J' is nonlinear we may use the linearization

$$J'(x^{n+1}) \sim H(x^n)(x^{n+1} - x^n) + J(x^n).$$

(5) It is equivalent to solving the linear equality constrained problem

$$J(x) \text{ subject to } G_{\mathcal{A}} x = c_{\mathcal{A}}.$$

3.7 Comparison to Gradient Method and SQP

Recall the projected gradient method in Section 3.2 for the (control form) equality constrained minimization (2.2):

$$\min \quad F(x) + H(u) \quad \text{subject to} \quad E(x, u) = 0, \;\; u \in \mathcal{C}.$$

The method consists of forward solve $E(x_n, u_n) = 0$ for x_n given $u_n \in \mathcal{C}$. Next, solve the adjoint equation $F'(x_n) + E_x(x_n, u_n)^* \lambda = 0$ for λ given u_n and new x_n from the forward step. Then, compute the gradient by the implicit Lagrange calculus $g_n = E_u(x_n, u_n)^* \lambda + H'(u_n)$. The projected gradient step is

$$u_{n+1} = \text{Proj}_{\mathcal{C}}(u_n - \alpha\, g_n),$$

for selected stepsize $\alpha > 0$. The corresponding SP step involves solving the saddle point problem for (x^+, u^+, λ)

$$E_x(x_n, u_n)(x^+ - x_n) + E_u(x_n, u_n)(u^+ - u_n) + E(x_n, u_n) = 0$$

$$F'(x^+) + E_x(x_n, u_n)^* \lambda = 0 \tag{3.46}$$

$$u^+ = \mathrm{Proj}_{\mathcal{C}}(\, u_n - \alpha \, (\, E_u(x_n, u_n)^* \lambda + H'(u^+) \,) \,).$$

Recall in Section 3.6.1 we consider the case $E(x, u) = E(x) + Bu$ and $F(x) = \frac{1}{2}(Qx, x)$ and the reduced order conjugate gradient method. Each step of the CG method is exactly the same as for the gradient method, except the forward solution step is linear and well-posed CG application, guaranteed to converge, and "hot start" is employed to reduce the number of inner loop iterates and overall number of iterates for the SP method. In summary, the total number of gradient iterates for the gradient method should be compared to the number of SP steps (outer loop) times the number of CG steps (inner loop).

In general, we use the Newton-like method for (3.46) or the alternating direction method in Section 4.4. Especially, the alternating direction method is very similar to the gradient step. The difference is we solve the saddle point problem for the linearized equality constraint problem and we iterate the inner loop and then update the solution by the SP update.

In summary, we use an inexact solution for each SP saddle point problem and "hot start." Now, we compare SP to SQP. In section 3.4.6 we state that SP is a partial SQP method that involves the linearization of the equality constraint. SP does not use the second derivatives of the Lagrangian and can avoid the ill-posed system for the SQP step. But, we need to solve the saddle point problem for each SP step. We introduce the Newton-like method for the saddle point solver and it is closely related to the semi-smooth Newton method (Section 4.4).

Thus we have stated that SP is a middle ground between the gradient method and SQP. SP combines both methods, utilizing the pros of the gradient and Newton's method. SP is regarded as a preconditioner for the gradient step, i.e., we use an inexact solution for each sequence of saddle point problem with gradient-like iterates.

Chapter 4

Applications of SP

In this chapter we describe concrete applications and provide a more detailed account of the SP method case by case. That is, each case needs a specific discussion on how to apply the generic SP method (3.14) and how to solve each SP step. Thus, each section in this chapter is very independent and demonstrates different aspects of the SP method.

4.1 Optimal Control Problem

The optimal control problem is introduced in Section 2.4 and recall the necessary optimality condition is a saddle point problem (2.26). The SP step is given as

$$\begin{aligned} \frac{d}{dt}x^+ &= f_x(t,x_n,u_n)(x^+ - x_n) + f_u(t,x_n,u_n)(u^+ - u_n) + f(t,x_n,u_n) \\ -\frac{d}{dt}\lambda &= f_x(t,x_n,u_n)^*\lambda + f_x^0(t,x^+,u^+) \\ u^+ &= \Psi(t,x^+,\lambda), \end{aligned} \tag{4.1}$$

where $u^+ = \Psi(t,x^+,\lambda)$ is implicitly a function of x^+ and λ by the optimality condition $f_u(t,x^+,u^+)^*\lambda + f_u^0(t,x^+,u^+) = 0$.

If $f^0(t,x,u) = \frac{1}{2}(Qx,x) + h(u)$, then $f_x^0(t,x^+,u^+) = Qx^+$. Moreover, if $h(u) = \frac{1}{2}(Ru,u)$ and $f(t,x,u) = f(x) + Bu$, then

$$Ru^+ = -B^*\lambda.$$

In this case (4.1) becomes the linear saddle point problem

$$\frac{d}{dt}x^+ = f_x(x_n)(x^+ - x_n) + f(x_n) + Bu^+$$

$$-\frac{d}{dt}\lambda = f_x(x_n)^*\lambda + Qx^+$$

$$Ru^+ = -B^*\lambda,$$

and the reduced order CG method discussed in Section 3.6.1 can be used as a saddle point solver.

In general, we have

$$h(u) - h(u^+(t)) + f_u(t, x_n, u_n)(u - u^+(t)) \geq 0 \quad \text{for } u \in U,$$

where $u^+ \in \mathcal{C} = \{u^+(t) \in U, \ \ a.e. \ \ t \in (0,T)\}$. Thus, we must use the method presented for non-smooth problems as in Section 2.2. For the fully nonlinear case (especially with f^0) we try to use the preconditioned CR method as described in Section 3.6.2 to solve the saddle point problem for the SP step (4.1).

4.2 Ill-posed Inverse Problem

Many inverse problems are formulated as follows:

$$\min \quad \frac{1}{2}|K(u) - y|_Y^2 + \int_\Omega (h(u) + \frac{\alpha}{2}|\nabla u|^2)\, d\omega \tag{4.2}$$

where $K : X \to Y$ is a compact nonlinear map and $y \in Y$ is given data. Since the Jacobian K' is compact, the inverse of K' is unbounded. Thus, we need to use the regularization method, for example those in (2.8) or (2.10), e.g., with $h(u) = \frac{\beta}{2}(|u| + |u|^2)$.

Often, $K(u)$ is determined by the equality constraint $E(x,u) = 0$ and $K(u) = x$. In this case, we use the formulation (2.2), i.e.,

$$\min \quad F(x) + H(u) \ \text{ subject to } \ E(x,u) = 0, \ \ u \in \mathcal{C}.$$

Or, we use the implicit Lagrange calculus (3.2) to compute $F(K(u))$ with respect to u. The SP method in terms of Gauss-Newton minimizes

$$\frac{1}{2}|K'(u_n)(u^+ - u) + K(u_n) - f|^2 + \int_\Omega (h(u^+) + \frac{\alpha}{2}|\nabla u^+|^2)\, d\omega,$$

and the SP step solves

$$K'(u_n)^*(K'(u_n)(u^+ - u) + K(u_n) - f) + h(v) - h(u^+) + \alpha\,(v - u^+) \geq 0, \quad v \in \mathcal{C}. \tag{4.3}$$

For non-smooth case of h as in Section 2.2 we use a sequential linearization method for solving (4.3). The method uses a linear equation solver and the linearization $K'(u_n)$ and may be costly, but is a very effective method as a full Newton method. Otherwise, we use the equality constraint formulation discussed above.

4.3 Semilinear Control Problem

In this section we discuss a semilinear control problem of the form

$$\frac{d}{dt}x(t) = A_0x(t) + f(x(t)) + B_0u(t) + (B_1u)x(t), \quad x(0) = x_0, \tag{4.4}$$

where A_0 is a nonnegative self-adjoint operator on a Hilbert space X and $B_0,\ B_1 \in \mathcal{L}(U, X)$. Thus, we have the linear control $B_0\,u$ and the bilinear control $(B_1u)\,x$ in this control model. Assume that $A_0 \in \mathcal{L}(V, V^*)$ where V is a closed subspace of X and define a symmetric bounded quadratic form on $V \times V$ by

$$-\sigma(x, y) = \langle Ax, y\rangle_{V^*\times V}.$$

That is,

$$\mathrm{dom}(A) = \{x \in V : |\sigma(x, y)| \leq c\,|y| \text{ for all } y \in V\}.$$

Assume $f : V \to V^*$ is weakly continuous and

$$\langle f(x_1) - f(x_2), x_1 - x_2\rangle \leq \omega\,|x_1 - x_2|_X^2.$$

It can be proven [63, 64] that (4.4) has a weak solution

$$x(t) \in W(0, T) = \{x \in H^1(0, T; V^*) \cap L^2(0, T; V)\}$$

given $u \in \mathcal{C} = \{u \in L^2(0, T; U) : u \in \widehat{U}\}$ with a closed convex set $\widehat{U}$ in U and the initial condition $x(0) = x_0 \in X$.

Consider the optimal control problem on the finite time horizon $[0, T]$

$$\min \quad J(x, u) = \int_0^T (\ell(x(t)) + h(u(t)))\,dt + g(x(T)) \tag{4.5}$$

subject to (4.4) and $u \in \mathcal{C}$. The incompressible Navier-Stokes control problem in Section 5.1 is

a concrete example of such an optimal control problem. Under an appropriate condition (e.g., see Section 2.2.1) there exists an optimal pair $(x^*, u^*) \in W(0,T) \times \mathcal{C}$ to problem (4.5). Let $E : W(0,T) \to L^2(0,T;V^*)$ be defined by

$$\langle E(x,u), \phi \rangle = \int_0^T \langle \frac{d}{dt}x(t) - A_0 x(t) - f(x(t)) - B_0 u(t) - (B_1 u)x(t), \phi(t) \rangle \, dt.$$

Then, (4.4)–(4.5) is a specific case of (2.3). The necessary optimality is given by

$$\begin{aligned}
&\frac{d}{dt}x^*(t) = A_0 x^*(t) + f(x^*(t)) + B_0 u^*(t) + (B_1 u^*)x^*(t), \quad x^*(0) = x_0 \\
&-\frac{d}{dt}\lambda(t) = A_0{}^*\lambda(t) + f'(x^*(t))^*\lambda(t) + (B_1 u^*)^*\lambda(t) + \ell'(x^*(t)), \quad \lambda(T) = g'(x^*(T)) \\
&h'(u^*(t)) + B_0{}^*\lambda(t) + (B_1 x^*)^*\lambda(t) = 0.
\end{aligned} \tag{4.6}$$

In general we have the optimality condition $u^* = \Psi(x^*, \lambda)$, e.g., see Section 2.4.1. The SP method solves

$$\min \quad J(x^+, u^+), \quad u^+ \in \mathcal{C} \tag{4.7}$$

subject to the linearized equality constraint

$$\frac{d}{dt}x^+ = A_0 x^+ + f'(x_n)(x^+ - x_n) + f(x_n) + B_0 u^+ + (B_1(u^+ - u_n))x_n + (B_1 u_n)x^+. \tag{4.8}$$

Thus, for each SP step, given (x_n, u_n) we solve the saddle point problem for update (x^+, u^+, λ)

$$\begin{aligned}
&\frac{d}{dt}x^+ = A_0 x^+ + f'(x_n)(x^+ - x_n) + f(x_n) + B_0 u^+ + (B_1(u^+ - u_n))x_n + (B_1 u_n)x^+, \; x^+(0) = x_0 \\
&-\frac{d}{dt}\lambda = A_0{}^*\lambda + f'(x_n)^*\lambda + (B_1 u_n)^*\lambda + \ell'(x^+), \quad \lambda(T) = g'(x^+(T)) \\
&h'(u^+) + B_0{}^*\lambda + (B_1 x_n)^*\lambda = 0,
\end{aligned}$$

where $u^+ = \Psi(x_n, \lambda)$.

Remark In general we also consider the semilinear dynamics of the form

$$\frac{d}{d}x(t) = Ax + F(x) + B_0 u(t) + B_1 u(t)\, x(t) \tag{4.9}$$

where A is the infinitesimal generator [29] of semigroup $S(t)$ on X and $F : X \to X$ is locally Lipschitz. We will discuss the moving actuator control for the damped wave equation, which is

of the form (4.9), in Section 4.3.1. The mild (weak) solution $x(t) \in X$ [29] to (4.9) is given by

$$x(t) = S(t)x(0) + \int_0^t S(t-s)(\, F(x(s)) + B_0 u(t) + (B_1 u(s))x(s)\,)\, ds.$$

One can derive the corresponding necessary optimality (4.6) and SP method (4.7)–(4.8) in exactly the same manner.

We have the SP step for (x^+, u^+) is written as

$$\frac{d}{dt}x^+(t) = Ax^+ + F'(x)(x^+ - x) + F(x) + B_0 u^+ + (B_1(u^+ - u))x + (B_1 u)x^+, \quad x^+(0) = x_0$$

$$-\frac{d}{dt}\lambda(t) = A^*\lambda + F'(x)^*\lambda + (B_1 u)^*\lambda + \ell'(x), \quad \lambda(T) = G'(x^+(T))$$

$$u^+(t) = \Psi(x^+(t), \lambda(t)).$$

Here, the adjoint equation for λ is understood in the mild solution form

$$\lambda(t) = G'(x^+(T)) + \int_t^T S^*(s-t)(\, F'(x^+)^*\lambda(s) + (B_1 u)^*\lambda(s) + \ell'(x^+(s))\,)\, ds.$$

For example we consider an optimal control problem

$$\min \quad \int_0^T \int_\Omega \left(|\nabla y|^2 + |y_t|^2\right) dx + |u(t)|^2\, dt \tag{4.10}$$

subject to

$$y_{tt} = \Delta y + f(y) + B(u(t))\, y_t \tag{4.11}$$

where f is a Lipschitz function on R (e.g., $f(y) = \sin(y)$ for the sine-Gordon equation, in the study of crystal dislocations [18]).

For the moving actuator control via location $u(t)$ of an actuator (see Section 4.3.1) we have

$$(B(u)\phi)(x) = \exp(-\frac{|x-u|^2}{2\sigma^2})\phi(x).$$

Define the velocity $v = \frac{\partial}{\partial t}y$. The first order form for $x = (y, v) \in X = H^1(\Omega) \times L^2(\Omega)$ of (4.11) is given as (4.9) with

$$A = \begin{pmatrix} 0 & I \\ -\Delta & 0 \end{pmatrix}, \quad F(y,v) = \begin{pmatrix} 0 \\ f(y) \end{pmatrix}, \quad B(u) = \begin{pmatrix} 0 \\ b(u)v \end{pmatrix}.$$

It is known [29] that A generates the group $S(t)$ on X.

Next, we consider the stationary case, i.e.,

$$\min \quad \ell(x) + h(u), \quad u \in \mathcal{C}$$

subject to

$$E(x,u) = A_0 x + f(x) + B_0 u + (B_1 u)\, x = 0.$$

Recall the coefficient control problem and inverse medium problem in Section 2.2.4 in which

$$(B_1 u)x = u(\omega)\, x(\omega), \quad \omega \in \Omega.$$

The SP step for the stationary case solves the system

$$A_0 x^+ + f'(x_n)(x^+ - x_n) + f(x_n) + B_0 u^+ + (B_1(u^+ - u_n))x_n + (B_1 u_n)x^+ = 0$$

$$A_0{}^*\lambda + f'(x_n)^*\lambda + (B_1 u_n)^*\lambda + \ell'(x^+) = 0$$

$$h'(u^+) + B_0{}^*\lambda + (B_1 x_n)^*\lambda = 0$$

for the update (x^+, u^+, λ).

For example we have $X = L^2(\Omega)$ and $u \in L^2(\widetilde{\Omega})$ where $\widetilde{\Omega}$ is a subset of domain $\Omega \subset R^d$ (represents a locally supported control region) and

$$A_0 = \Delta, \quad f(x) = |x|, \quad \text{and} \quad B_0 u = \chi_{\widetilde{\Omega}}.$$

In such problems we discretize the equality constraint in space $\omega \in \Omega$ and apply the SP algorithm for the discretized problem (see Chapter 6) with

$$(A_0)_h, \quad f_h(x_h), \quad (B_0)_h, \quad (B_1)_h$$

the corresponding finite rank operators.

4.3.1 Moving Damping Actuator Control Problem

Now we revisit specifically the moving damping actuator control problem introduced in Section 4.3 to discuss more details. That is, consider the constrained optimization with PDE constraint

given by the damped wave equation

$$y_{tt} + \sum_{i=1}^{s} \gamma_i(w_i) y_t = \Delta y, \quad \frac{d}{dt} w(t) = u(t) \tag{4.12}$$

where w_i is the location of the ith actuator and $\gamma_i \in L^1(\Omega)$ is its damping distribution, e.g., Gaussian $\gamma_i(w) = \exp(-\frac{|x-w|^2}{2\sigma^2})$. Thus $\gamma(w(t))\, y_t = \sum_{i=1}^{s} \gamma_i(w_i) y_t$ represents damping by the actuators.

The cost to minimize is the standard wave energy plus a penalty for movement of the actuators, i.e.,

$$\min \quad F(y) + H(u) = \int_0^T \left(\int_\Omega (|\nabla y|^2 + |y_t|^2)\, dx + \alpha\, |u(t)|^2 \right) dt \tag{4.13}$$

where $u(t) = \frac{\partial w}{\partial t}$ is the velocity control of the location of actuators.

The first order differential form for the system (4.12) is a system for (y, v, w)

$$\frac{d}{dt} \begin{pmatrix} y \\ \\ v \end{pmatrix} = \mathcal{A}(w) \begin{pmatrix} y \\ \\ v \end{pmatrix}, \quad \frac{d}{dt} w = u \tag{4.14}$$

where $(y, v) \in X$, $v = \frac{d}{dt} y$, and $\mathcal{A}(w)$ is a linear operator on $X = H^1(\Omega) \times L^2(\Omega)$ defined by

$$\mathcal{A}(w) \begin{pmatrix} y \\ \\ v \end{pmatrix} = \begin{pmatrix} v \\ \\ \Delta y - \gamma(w) v \end{pmatrix} = \begin{pmatrix} 0 & I \\ \\ \Delta & -\gamma(w) \end{pmatrix} \begin{pmatrix} y \\ \\ v \end{pmatrix},$$

given $w \in (\mathbb{R}^d)^m$, with domain

$$\operatorname{dom} \mathcal{A}(w) = \{(y, v) \in X : v \in H_0^1(\Omega) \text{ and } -\Delta y + \gamma(w) v \in L^2(\Omega)\}.$$

An optimal control problem for the system (4.12)–(4.13) is the optimal actuator problem

$$\min \quad \int_0^T \frac{1}{2} \left(|\nabla y|_{L^2}^2 + |y_t|_{L^2}^2 \right) dt + \int_0^T \frac{\alpha}{2} |u|^2\, dt + \frac{\beta}{2} |w(T) - \bar{w}|^2 \tag{4.15}$$

over admissible controls u, where $\bar{w}$ is a desired location for actuators (at terminal time T).

Note that (4.15) may be written for the first order differential form (4.14) as

$$\min \quad \int_0^T \frac{1}{2} \left(|\nabla y|^2 + |v|^2 \right) dt + \int_0^T \frac{\alpha}{2} |u|^2\, dt + \frac{\beta}{2} |w(T) - \bar{w}|^2,$$

where $v = y_t$.

To formulate the saddle point problem for SP we need to linearize the system (4.14). The only nonlinear term is $-\gamma(w)v$, which we linearize at w_c, v_c, i.e.,

$$-\gamma(w_c)v_c - \gamma'(w_c)v_c(w - w_c) - \gamma(w_c)(v - v_c) = -\gamma'(w_c)v_c(w - w_c) - \gamma(w_c)v.$$

Thus each SP step (3.38) solves the following saddle point system for $(y^+, v^+, w^+, \lambda, \mu)$:

$$\begin{cases} \dfrac{d}{dt}\begin{pmatrix} y^+ \\ v^+ \end{pmatrix} = \mathcal{A}(w_c)\begin{pmatrix} y^+ \\ v^+ \end{pmatrix} + \dot{\mathcal{A}}(w^+ - w_c)\begin{pmatrix} y_c \\ v_c \end{pmatrix}, \qquad \dfrac{d}{dt}w^+ = u^+ \\ \\ -\dfrac{d}{dt}\lambda = \mathcal{A}(w_c)^*\lambda + \begin{pmatrix} \nabla y \\ v \end{pmatrix}, \qquad -\dfrac{d}{dt}\mu = -\gamma'(w_c){v_c}^*\mu \\ \\ \alpha\, u^+ + \mu = 0, \end{cases}$$

where

$$\dot{\mathcal{A}} = \begin{pmatrix} 0 & 0 \\ & \\ 0 & -\gamma'(w_c) \end{pmatrix}.$$

Next we describe the solution method for the SP step (3.38). We use the conjugate gradient method for solving each step of SP and the gradient g^n is computed as follows. Given the current u^n, solve

$$\frac{dw^n}{dt} = u^n, \qquad w^n(0) = w_0$$

$$\frac{dy^n}{dt} = \mathcal{A}(w_c)y^n + \dot{\mathcal{A}}(w^n - w_c)y^n, \qquad y^n(0) = y_0$$

$$-\frac{d}{dt}\lambda^n = \mathcal{A}(w_c)^*\lambda^n + \begin{pmatrix} \nabla y^n \\ v^n \end{pmatrix}, \qquad \lambda^n(T) = 0.$$

$$\frac{d}{dt}\mu^n = -\gamma'(w_c){v_c}^*\mu^n, \qquad \mu^n(T) = \beta\,(w^n(T) - \bar{w}).$$

Then compute the gradient

$$g^n = \alpha\, u^n + \mu^n.$$

Then, we use the nonlinear gradient updates (3.46) for SP. In general it has the following

(moving actuator problem) form

$$\frac{dy}{dt} = A(w)y, \quad \frac{d}{dt}w = u$$

where $A(w)$ is a closed operator on a Hilbert space X and w is the location of actuators in the domain Ω. The problem is to optimize a class of cost functionals defined for (y, w, u) with respect to velocities u and initial locations $w(0)$.

4.4 Non-smoothness in SP

Consider the non-smooth constrained optimization of the form

$$\min \quad F(x) + H(u) \quad \text{subject to} \quad E(x, u) = 0$$

where $u \to H(u)$ is convex but not necessarily C^1. The convex constraint $u \in \mathcal{C}$ is a specific case by

$$H(u) = \begin{cases} 0 & u \in \mathcal{C} \\ \\ \infty & u \notin \mathcal{C}. \end{cases}$$

For the case of inequality constraint $Gu \leq c$ we let $\mathcal{C} = \{u \in U : Gu \leq c\}$. The saddle point problem for (x, u) is given by

$$\begin{aligned} E_x(x, u)^*\lambda + F'(x) &= 0 \\ -E_u(x, u)^*\lambda &\in \partial H(u) \end{aligned} \tag{4.16}$$

where the subdifferential [12] $\partial H(u)$ of H at u is defined by

$$\partial H(u) = \{p \in U^* : H(v) - H(u) \geq (p, v - u) \text{ for all } v \in U\}.$$

Thus, the second equation of (4.16) is a set valued equation. In order to obtain an equivalent equation form, we define the Yosida-Moreau approximation of H [30, 29], given $c > 0$ and $\mu \in U^*$

$$H_c(u, \mu) = \inf_{v \in U} \{H(v) + \frac{c}{2}|u - v|_U^2 + (\mu, u - v)_{U^* \times U}\}.$$

It is based on the equivalent formulation of (2.2)

$$\min \quad F(x) + H(v) + \frac{c}{2}|u - v|^2 + (\mu, u - v)$$

subject to

$$E(x,u)=0, \quad u=v.$$

It is shown in [29] that $u \to H_c(u,\mu)$ is differentiable with

$$\langle H_c'(u,\mu),h\rangle = \langle c\,(u-v)+\mu,h\rangle \quad \text{for all } h\in U$$

where

$$v=\text{argmin}_{v\in U}\{H(v)+\frac{c}{2}|u-v|_U^2+(\mu,u-v)_U\}$$

and $u \to H_c'(u,\mu)$ is Lipschitz continuous. Most importantly, we have the equation form

$$\begin{aligned} &E_u(x,u)^*\lambda+\mu=0 \\ &\mu=H_c'(u,\mu) \end{aligned} \tag{4.17}$$

is equivalent to the second condition in (4.16).

For example, consider the L^1 cost

$$H(u)=\int_\Omega |\Lambda u|_1\,dx$$

where Λ on U is a closed linear operator. Then, (4.17) is equivalent to

$$\begin{aligned} &E_u(x,u)^*\lambda+\Lambda^*\mu=0 \\ &\mu(s)=\frac{\mu+\beta\,\Lambda u}{\max(1,|\mu+\beta\,\Lambda u|)} \quad \text{almost all } s\in\Omega. \end{aligned}$$

The SP step solves the following system for (x^+,u^+,λ,μ):

$$\begin{aligned} &E_x(x_c,u_c)(x^+-x_c)+E_u(x_c,u_c)(u^+-u_c)+E(x_c,u_c)=0 \\ &E_x(x_c,u_c)^*\lambda+F'(x^+)=0 \\ &E_u(x_c,u_c)^*\lambda+\mu=0 \\ &\mu=H_c'(u^+,\mu). \end{aligned} \tag{4.18}$$

Now, we introduce methods for solving (4.18).

4.4.1 Alternating Direction Iterate Solver

We introduce the so-called alternating direction method for solving (4.18).

1. Given $u^+ \in \mathcal{C}$ solve for x^+

$$E_x(x_c, u_c)(x^+ - x_c) + E_u(x_c, u_c)(u^+ - u_c) + E(x_c, u_c) = 0.$$

2. Solve the adjoint equation for λ

$$E_x(x_c, u_c)^* \lambda + F'(x^+) = 0.$$

3. Given v and μ update u^+ by

$$E_u(x_c, u_c)^* \lambda + c\,(u^+ - v) + \mu = 0.$$

4. Update v^+ by

$$v^+ = \text{argmin}_v\{H(v) + \frac{c}{2}|u^+ - v|^2 + (\mu, u^+ - v)\}.$$

5. Update μ^+ by

$$\mu^+ = \mu + c\,(u^+ - v^+).$$

6. Repeat steps by resetting the variables by (u^+, v^+, μ^+) until convergence.

Remarks (1) If the method converges, $u^+ = v^+$ and $\mu^+ \in \partial H(u^+)$.
(2) The method works even if H is not necessarily convex.

4.4.2 Newton-like Solver

Now we introduce a Newton-like solver for (4.18). Consider the viscous version with $\alpha > 0$ for $H(u)$ given by

$$H(u) = \int_\Omega \Big(\frac{\alpha}{2}|u(s)|^2 + \beta\,|u(s)|\Big)\,ds,$$

where $U = L^2(\Omega)$. Consider the minimization over $u \in R$ of $u \to (p, u) + \frac{\alpha}{2}|u|^2 + \beta\,|u|$. We have the necessary optimality condition

$$\alpha\,u + p + \beta = 0 \quad \text{if } u > 0$$

$$\alpha\,u + p - \beta = 0 \quad \text{if } u < 0,$$

and

$$\begin{aligned} u = -\tfrac{p+\beta}{\alpha} > 0 \quad &\iff \quad p < -\beta \\ u = -\tfrac{p-\beta}{\alpha} < 0 \quad &\iff \quad p > \beta. \end{aligned} \tag{4.19}$$

Thus, u is given by (4.19) if $|p| > \beta$, and otherwise $u = 0$ if $|p| < \beta$. In summary,

$$-E_u(x_c, u_c)^* \lambda \in \partial H(u^+) \tag{4.20}$$

is equivalent to

$$\text{if } |E_u(x_c, u_c)^* \lambda| < \beta, \text{ then } u^+ = 0$$

$$\text{if } |E_u(x_c, u_c)^* \lambda| \geq \beta, \text{ then } u^+ = -\frac{p \pm \beta}{\alpha}, \quad p = E_u(x_c, u_c)^* \lambda$$

where all operations are pointwise.

Numerically, we solve (4.20) using the penalty method. Let P be the diagonal operator defined by

$$P_{ii} = 10^8 \quad \text{if } |E_u(x_c, u_c)^* \lambda_c| \leq \beta \text{ and otherwise zero.}$$

We define the linearized equation for the Newton update for (x^+, u^+, λ):

$$E_x(x_c, u_c)(x^+ - x_c) + E_u(x_c, u_c)(u^+ - u_c) + E(x_c, u_c) = 0$$

$$E_x(x_c, u_c)^* \lambda + F'(x_c)(x^+ - x_c) + F(x_c) = 0$$

$$E_u(x_c, u_c)^* \lambda + \alpha u^+ + P u^+ + \beta \, \text{sign}(u_c) = 0.$$

After solving this system, we take the projection step:

$$|E_x(x^+, u^+)^* \lambda| \leq \beta \quad \implies \quad u^+ = 0.$$

4.4.3 Inequality Constraint

Consider the inequality constraint problem

$$\min \quad F(x) + H(u) \quad \text{subject to} \quad E(x, u) = 0, \;\; u \leq 0$$

i.e., $\mathcal{C} = \{u \leq 0\}$. As we discussed in Section 4.4, we have the necessary optimality

$$\begin{aligned} &E_x(x_c, u_c)(x^+ - x_c) + E_u(x_c, u_c)(u^+ - u_c) + E(x_c, u_c) = 0 \\ &E_x(x_c, u_c)^* \lambda + F'(x^+) = 0 \\ &E_u(x_c, u_c)^* \lambda + \mu = 0 \\ &\mu = \max(0, \mu + \beta u^+). \end{aligned} \tag{4.21}$$

We can apply the alternating direction method in Section 4.4 for solving this saddle point problem. Now we introduce a Newton-like solver for (4.21).

The complementarity condition (last equation of (4.21)) is equivalent to

$$\mu + \beta\, u^+ > 0 \Rightarrow \mu^+ = \mu + \beta u^+$$

$$\mu + \beta\, u^+ \leq 0 \Rightarrow \mu^+ = 0,$$

where all operations are pointwise.

Numerically, we solve (4.21) using the penalty method. Let P be the diagonal operator defined by

$$P_{ii} = 10^8 \quad \text{if } \mu + \beta\, u^+ > 0 \text{ and otherwise zero.}$$

We define the linearized equation for the Newton update for (x^+, u^+, λ, μ):

$$E_x(x_c, u_c)(x^+ - x_c) + E_u(x_c, u_c)(u^+ - u_c) + E(x_c, u_c) = 0$$

$$E_x(x_c, u_c)^* \lambda + F'(x_c)(x^+ - x_c) + F(x_c) = 0$$

$$E_u(x_c, u_c)^* \lambda + \alpha u^+ + P u^+ + \max(0, \mu_c + \beta\, u_c) = 0.$$

After solving this system, we take the projection step:

$$\mu + \beta\, u^+ > 0 \quad \Longrightarrow \quad u^+ = 0.$$

Chapter 5

Optimal Control of Navier-Stokes System and SP

In this chapter we first introduce the controlled Navier-Stokes and the basic Navier-Stokes theory, i.e., existence of solutions, the Hodge decomposition, and the time integration by the second order implicit and explicit scheme. Then, we derive the necessary optimality condition based on the Lagrange multiplier theory and formulate the SP method for this problem. Also, we develop a specific space discretization method for the Navier-Stokes equations.

5.1 Navier-Stokes Control Problem

The Navier-Stokes equation describes fluid flow in a compressible medium. The equations arise from the two laws of conservation of mass and conservation of momentum. Conservation of mass is given by

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho u) = 0, \tag{5.1}$$

where $\rho = \rho(t, x)$ is the mass density, and $u = u(t, x)$ is the velocity of the fluid over domain Ω in R^d. Conservation of momentum is given by

$$\rho \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) + \nabla p = \mu \, \Delta u + f, \tag{5.2}$$

where p is the pressure, μ is the viscosity of the medium, and f is the applied force vector. The quantity in brackets is the Lagrangian derivative (or total derivative or material derivative) of the velocity, i.e.,

$$\frac{Du}{Dt} = \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right).$$

It is the rate change of the velocity along the flow. It comes from taking the time derivative of $u = u(t, x(t))$ (a function of two variables—time t and position $x(t)$, with $u(t) = \frac{d}{dt}x(t)$). Thus, we have

$$\frac{d}{dt}u(t, x(t)) = \frac{\partial u}{\partial t} + u \cdot \nabla u = \frac{Du}{Dt},$$

coordinate-wise, i.e.,

$$(u \cdot \nabla u)_j = u \cdot \nabla u_j.$$

For example, a two-dimensional flow $u = (u^1, u^2)$ in spacial coordinates x_1, x_2 has

$$u \cdot \nabla u = \begin{pmatrix} u^1\, u^1_{x_1} + u^2\, u^1_{x_2} \\ \\ u^1\, u^2_{x_1} + u^2\, u^2_{x_2} \end{pmatrix} = \begin{pmatrix} u \cdot \nabla u^1 \\ \\ u \cdot \nabla u^2 \end{pmatrix}.$$

In order to complete the compressible Navier-Stokes equation (5.1)–(5.2) we equip the equation of state

$$p = p(\rho),$$

e.g., $p(\rho) = \frac{1}{\gamma}\rho^\gamma$. If ρ is a constant, the mass conservation is equivalent to the divergence-free condition for u:

$$\operatorname{div} u = \nabla \cdot u = 0.$$

The divergence-free condition implies the volume under a divergence-free vector field is conserved. The incompressible Navier-Stokes system is written as

$$\rho \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) + \nabla p = \mu\, \Delta u + f, \quad \nabla \cdot u = 0, \tag{5.3}$$

where the gradient term ∇p balances the momentum equation for the divergence-free condition $\operatorname{div} u = 0$. We may say the Stokes equation is the "massless" version of the Navier-Stokes equation, i.e.,

$$\nabla p = \mu\, \Delta u + f, \quad \nabla \cdot u = 0. \tag{5.4}$$

The Stokes equation (5.4) is the time-independent Navier-Stokes equation, and thus finds stationary (time-independent) flows u.

Now, we consider the control problem for the incompressible Navier-Stokes system (5.3) in which the applied force f is induced by control inputs $v(t)$, (e.g., magnetic, thermally induced), i.e.,

$$f(t, x) = Bv(t) = \sum_{k=1}^{m} b_k(x)\, v_k(t)$$

for a finite number of control variables v_k, $1 \le k \le m$ with control distributions $b_k(x)$.

For injection or sucking control

$$n \cdot u(t,x) = \sum_{k=1}^{m} \tilde{b}_k(x)\, v_k(t)$$

where n is the outward normal vector at the boundary $\partial\Omega$ and $\tilde{b}_k(x)$ is the kth control distribution on $\partial\Omega$.

Define the stress tensor S by

$$S = 2\mu\,\epsilon + p\,I$$

where ϵ is the linear strain defined by

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^t), \quad \text{where} \quad \epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right).$$

Then, the incompressible Navier-Stokes equation (5.3) can be written as

$$\rho\,\frac{Du}{Dt} + \mathrm{Div}(S) = f.$$

The Navier boundary condition (proposed by Navier in 1823, see [49, 6, 50]) asserts that the velocity on the boundary should be proportional to the tangential component of the normal stress, i.e.,

$$(S \cdot n)_\tau + \kappa\, u = 0, \quad n \cdot u = 0, \tag{5.5}$$

where $(S \cdot n)_\tau$ is the tangential components of the normal stress and κ is a constant of proportionality. Thus, the Navier boundary (control) condition is given by

$$(S \cdot n)_\tau + \kappa\, u = \sum_{k=1}^{m} \tilde{b}_k(x)\, v_k(t), \quad n \cdot u = 0,$$

where $\tilde{b}_k(x)$ is the kth control distribution on the boundary $\partial\Omega$.

5.1.1 Stokes Theory

In this section we discuss the variational formulation of (5.4). Consider the constrained optimization

$$\min \quad J(u) = \frac{1}{2}\int_\Omega (\frac{1}{2}\,\nabla u : \nabla u - f \cdot u)\, dx$$

subject to the divergence-free condition

$$\nabla \cdot u = 0$$

over the vector field $u \in X = H_0^1(\Omega)^d$. Note that ∇u is in $R^{d\times d}$ and we use the Frobenius products of ∇u. Thus, $\int_\Omega \nabla u : \nabla u \, dx$ is the Dirichlet norm of u. Here, we assume the no-slip condition

$$u = 0 \quad \text{at boundary } \partial\Omega.$$

Using the Lagrange multiplier theory,

$$L(u, \lambda) = \frac{1}{2} \int_\Omega (\frac{1}{2} \nabla u : \nabla u - f \cdot u) \, dx + (\lambda, \nabla \cdot u),$$

and

$$L_u(u, \lambda)(h) = (\nabla u, \nabla h) + (\lambda, \nabla \cdot h) - (f, h) = 0 \quad \text{for all } h \in X, \quad \nabla \cdot u = 0 \text{ in } \Omega,$$

which is the weak form of the Stokes equation. By integration by parts (Green's formula),

$$(\nabla u, \nabla h) = \langle -\Delta u, h \rangle$$

$$(\lambda, \nabla \cdot h) = \langle -\nabla \lambda, h \rangle.$$

Thus, we obtain the strong form of the Stokes equation

$$-\Delta u - \nabla \lambda = f, \quad \nabla \cdot u = 0,$$

with Lagrange multiplier equal the negative pressure, i.e., $\lambda = -p$.

5.1.2 Hodge Decomposition

Every vector field can be uniquely decomposed into the sum of a gradient field and a divergence-free vector field, i.e., given a vector field f on a domain Ω there exists a scalar function p and a vector field w such that

$$f = \nabla p + w, \tag{5.6}$$

where $\operatorname{div} w = 0$, and $n \cdot w = 0$ on the boundary $\partial\Omega$. Moreover, the gradient field ∇p and the vector field w are orthogonal in the L^2 sense:

$$\int_\Omega \nabla p \cdot q \, dx = 0.$$

Since w is divergence-free, if we take the divergence of both sides of (5.6) we obtain

$$\operatorname{div} f = \Delta p, \quad \text{with} \quad \frac{\partial p}{\partial \nu} = n \cdot \nabla p = n \cdot f \quad \text{on } \partial\Omega. \tag{5.7}$$

Boundary value problem (5.7) has a unique solution p (up to an additive constant) provided $\int_\Omega \operatorname{div} f\, dx = \int_{\partial\Omega} n \cdot f\, ds$, which is guaranteed by the divergence theorem [11]. Then, we obtain w by (5.6), i.e.,

$$w = f - \nabla p.$$

Note that w is divergence-free and satisfies the boundary condition $n \cdot w = 0$ on $\partial\Omega$ follows since p satisfies (5.7).

The Hodge decomposition (5.6) is a very useful tool for solving the incompressible Navier-Stokes equation (5.3) since it allows us to eliminate the pressure term ∇p when we cast the weak form of the equality constraint:

$$(E(u,v),\lambda) = \int_0^T \left[(u_t,\lambda) + (u\cdot\nabla u,\lambda) + \nu(\nabla u,\nabla\lambda) - (Bv,\lambda)\right] dt = 0.$$

5.1.3 Weak Solution of Navier-Stokes

In this section we discuss weak solutions to the Navier-Stokes equation (5.1)–(5.2). Let V be the divergence-free subspace of $H_0^1(\Omega)^d$:

$$V = \{u \in H_0^1(\Omega)^d : \nabla \cdot u = 0\}.$$

Let H be the completion of V with respect to $L^2(\Omega)^d$, i.e.,

$$H = \{u \in L^2(\Omega)^d : \nabla \cdot u = 0, \quad n \cdot u = 0 \text{ at boundary } \partial\Omega\}.$$

We define the trilinear form b on $V \times V \times V$ by

$$b(u,v,\phi) = \int_\Omega (u\cdot\nabla v,\phi)\, dx, \quad u,v,\phi \in V,$$

the quadratic form on $V \times V$ by

$$a(u,\phi) = \int_\Omega \nu\,(\nabla u,\nabla\phi)\, dx,$$

and the control form $c(v,\lambda)$ is defined by

$$c(v,\phi) = \langle Bv,\phi\rangle, \quad \text{for } v \in U \text{ and } \phi \in V.$$

For example, for the distributed control

$$\langle Bv,\phi\rangle = \int_\Omega \sum_k b_k(x) v_k\, \phi(x)\, dx,$$

and for the Navier control at the boundary

$$\langle Bv, \phi \rangle = \int_{\partial\Omega} \sum_k \tilde{b}_k(x) v_k \, \phi(x) \, ds_x.$$

An H-valued function $u(t) \in L^2(0,T;V) \cap L^\infty(0,T;H)$ is a weak solution to (5.1)–(5.2) if for $\phi \in V$, $u = u(t)$ satisfies

$$\langle \frac{\partial u}{\partial t}, \phi \rangle_{V^* \times V} + b(u,u,\phi) + a(u,\phi) = c(v(t),\phi). \tag{5.8}$$

It is shown [63, 64] that for given $u(0) \in H$ and $f(t) \in L^2(0,T;V^*)$ there exists a unique weak solution $u(t) \in L^2(0,T;V) \cap C(0,T;H) \cap H^1(0,T;V^*)$ in the two dimensional case and $u(t) \in L^2(0,T;V) \cap L^\infty(0,T;H) \cap W^{1,4/3}(0,T;V^*)$ (may not be unique) in the three dimensional case. We refer to [63, 64] for more facts on the solutions to (5.8).

An important property used in the existence proof of weak solutions is the conservation property of the convective term:

$$b(u,v,w) + b(u,w,v) = 0, \quad b(u,v,v) = 0$$

for $u \in H$ and $w, v \in H^1(\Omega)^d$. In fact,

$$b(u,v,w) + b(u,w,v) = \int_\Omega u \cdot \nabla (w \cdot v) \, dx = 0$$

by the divergence theorem. In detail, we have

$$b(u,v,w) + b(u,w,v) = \int_\Omega (u \cdot \nabla v) \cdot w + (u \cdot \nabla w) \cdot v \, dx$$

and by the chain rule

$$u \cdot \nabla (w \cdot v) = u \cdot (\nabla w \cdot v + \nabla v \cdot w) = (u \cdot \nabla w) \cdot v + (u \cdot \nabla v) \cdot w.$$

By the divergence theorem,

$$\int_\Omega u \cdot \nabla (w \cdot v) dx = \int_{\partial\Omega} (n \cdot u) \, (w \cdot v) \, ds + \int_\Omega \operatorname{div}(u) \, (w \cdot v) \, dx = 0,$$

where we used the properties $n \cdot u = 0$ on $\partial\Omega$ and $\operatorname{div} u = 0$.

Letting $\phi = u$ in (5.8) we obtain energy identity

$$\frac{1}{2} \frac{d}{dt} |u|_H^2 + a(u(t),u(t)) - \langle f(t), u(t) \rangle_{V^* \times V} = 0 \quad \text{a.e. in } t \in (0,T)$$

if $u \in H^1(0,T;V^*)$ [63, 64] and

$$\frac{1}{2}|u(t)|_H^2 + \int_0^t a(u(s),u(s))\,ds = \frac{1}{2}|u(0)|_H^2 + \int \langle f(s),u(s)\rangle_{V^*\times V}\,ds. \tag{5.9}$$

Note that

$$\langle f(s),u(s)\rangle_{V^*\times V} \le \frac{1}{2}|f(s)|_{V^*}^2 + \frac{1}{2}|u(s)|_V^2.$$

(This follows from the Cauchy-Schwarz inequality $\langle f,u\rangle \le |f|\,|u|$ and the inequality identity $0 \le (|f|-|u|)^2 = |f|^2 - 2|f|\,|u| + |u|^2$.) Thus we have the a priori estimate

$$|u(t)|_H^2 + \int_0^t a(u(s),u(s))\,ds \le M\,(|u(0)|_H^2 + \int_0^t |f(s)|_{V^*}^2\,ds)$$

for some constant $M > 0$.

For the control problem we assume $B \in \mathcal{L}(U,V^*)$. Then, for all $u(0) \in H$ and $u \in L^2(0,T;U)$ the equation (5.8) has a weak solution.

5.1.4 Optimal Control Problem for Incompressible Navier-Stokes Flow

In this section we discuss the Navier-Stokes optimal control problem:

$$\min \quad \int_0^T \ell(u(t)) + h(v(t))\,dt + g(u(T)) \tag{5.10}$$

subject to

$$\frac{\partial u}{\partial t} + u\cdot\nabla u + \nabla p = \nu\,\Delta u + Bv, \quad \nabla\cdot u = 0, \tag{5.11}$$

over admissible controls

$$V_{ad} = \{v \in L^2(0,T;U),\ \ v(t) \in K \ a.e.\}$$

where K is a closed convex set in U. Here the performance cost ℓ must be chosen so that the desired property of flow u is met. For example,

$$\ell(u) = \int_{\widetilde{\Omega}} |\nabla \times u|^2\,dx$$

where $\omega(x) = (\nabla\times u(x))$ is the vorticity and $\widetilde{\Omega}$ is a subset of domain Ω, i.e., we try to minimize the circulation on subdomain $\widetilde{\Omega}$. Another example is tracking to a desired flow $\bar{u}$, i.e.,

$$\ell(u) = |u-\bar{u}|_V^2.$$

For the control cost $h(v)$, we assume $v \to h(v)$ is lower semicontinuous, but may not be differentiable.

We consider next the existence of optimal controls to (5.10)–(5.11). We need to show the weak continuity of weak solution u as a function of control v since the cost functional is weakly sequentially lower semicontinuous, i.e., if $v_n \in L^2(0,T;U)$ weakly converges to v, then (at least one) corresponding solution u_n to (5.11) converges weakly to u in $L^2(0,T;V)$. It can be shown that weak continuity holds if $\dim(U) = m$, or K is a compact set.

5.2 Optimality System

In this section we derive the necessary optimality for the optimal control problem (5.10)–(5.11). We assume we normalize the equations with respect to mass density, i.e., we set $\rho = 1$ and thus the pressure p is replaced by p/ρ, and the viscosity μ is replaced by $\nu = \mu/\rho$. We first show how to derive the adjoint equation for the adjoint variable λ in the the Navier-Stokes equation. Define the Lagrange functional for (5.10)–(5.11):

$$L(u,v,\lambda) = F(u) + H(v) + (E(u,v),\lambda)$$

where

$$(E(u,v),\lambda) = \int_0^T \left[\langle \frac{\partial u}{\partial t}, \lambda \rangle + b(u,u,\lambda) + a(u,\lambda) - c(v,\lambda) \right] dt.$$

The Fréchet derivative of L with respect to u is given by

$$L_u(u,v,\lambda)(h) = \langle F'(u), h \rangle + \int_0^T \langle \frac{\partial h}{\partial t}, \lambda \rangle + b(u,h,\lambda) + b(h,u,\lambda) + a(h,\lambda)\, dt \tag{5.12}$$

in the direction of $h \in W_0(0,T)$ where

$$W_0(0,T) = \{h \in H^1(0,T;V^*) \cap L^2(0,T;H) : h(0) = 0\}.$$

The Fréchet derivative with respect to control v of L is given by

$$L_v(u,v,\lambda)(w) = \langle H'(v), w \rangle + \int_0^T -c(w,\lambda)\, dt \tag{5.13}$$

in the direction of $w \in L^2(0,T;U)$.

We derive the strong form of the adjoint equation for λ as follows. By the divergence theorem

$$\begin{aligned} a(h,\lambda) &= \langle -\nu\Delta\lambda, h\rangle \\ \\ b(u,h,\lambda) &= (u\cdot\nabla h,\lambda) = (u^1 h^1_{x_1} + u^2 h^1_{x_2},\lambda^1) + (u^1 h^2_{x_1} + u^2 h^2_{x_2},\lambda^2) \\ &= -((u^1\lambda^1)_{x_1} + (u^2\lambda^1)_{x_2}, h^1) - ((u^1\lambda^2)_{x_1} + (u^2\lambda^2)_{x_2}, h^2) \\ \\ b(h,u,\lambda) &= (h\cdot\nabla u,\lambda) = (h^1 u^1_{x_1} + h^2 u^1_{x_2},\lambda^1) + (h^1 u^2_{x_1} + h^2 u^2_{x_2},\lambda^2) \\ &= (\lambda^1 u^1_{x_1} + \lambda^2 u^2_{x_1}, h^1) + (\lambda^1 u^1_{x_2} + \lambda^2 u^2_{x_2}, h^2). \end{aligned}$$

Thus,

$$(u\cdot\nabla h,\lambda) + (h\cdot\nabla u,\lambda) = (B(u)^*\lambda, h)$$

where

$$B(u)^*\lambda = \begin{pmatrix} -(u^1\lambda^1)_{x_1} - (u^2\lambda^1)_{x_2} + \lambda^1 u^1_{x_1} + \lambda^2 u^2_{x_1} \\ \\ -(u^1\lambda^2)_{x_1} - (u^2\lambda^2)_{x_2} + \lambda^1 u^1_{x_2} + \lambda^2 u^2_{x_2} \end{pmatrix}.$$

In conclusion the adjoint equation for λ is

$$-\frac{\partial\lambda}{\partial t} + B(u)^*\lambda + \nabla q + F'(u) = \nu\Delta\lambda, \quad \lambda(T) = g'(u(T)), \tag{5.14}$$

where $q \in L^2(\Omega)$ is the corresponding pressure for the adjoint equation. Assume (u^*, v^*) is an optimal pair and assume λ is the solution to (5.14) with $u = u^*$. It follows from (2.25) that the optimality condition for v^* is

$$H'(v^*) + B^*\lambda = 0$$

if H is differentiable. In general (if H is convex) the optimality condition is given by

$$H(v) - H(v^*(t)) + \langle B(v - v^*(t)), \lambda(t)\rangle \geq 0 \tag{5.15}$$

for all $v \in K$, a.e. in $(0,T)$. In summary, we have the necessary optimality system (5.11)–(5.14)–(5.15) for (u^*, v^*, λ).

5.3 SP for Navier-Stokes Control

The linearized equality constraint $E'(x)(x^+ - x) + E(x) = 0$ for $x = (u,v)$ is given by

$$\langle\frac{\partial u^+}{\partial t}, \phi\rangle + b(u, u^+ - u, \phi) + b(u^+ - u, u, \phi) + b(u,u,\phi) + a(u^+,\phi) = \langle Bv^+, \phi\rangle$$

for all $\phi \in V$ and a.e. in $(0,T)$. Thus the saddle point problem for the SP update (u^+, v^+) is written as

$$\frac{\partial}{\partial t}u^+ + u^+ \cdot \nabla(u^+ - u) + u^+ \cdot \nabla u + \nabla p = \nu\,\Delta u^+ + Bv^+$$

$$-\frac{\partial}{\partial t}\lambda + B(u)^*\lambda + F'(u^+) = 0, \qquad \lambda(T) = g'(u(T)) \tag{5.16}$$

$$H'(v^+) + B^*\lambda = 0$$

if $K = U$ and H is differentiable. In general the last equation is

$$H(v) - H(v^*(t)) + \langle B(v - v^*(t)), \lambda(t)\rangle \geq 0$$

for all $v \in K$. Thus we have the necessary optimality written for the SP step for the incompressible Navier-Stokes control problem.

5.4 Numerical Methods

We consider a 2-dimensional step flow for the incompressible Navier-Stokes system. Step flow includes channel flow and cavity flow, see Figure 5.1. We first partition domain $\Omega \subset R^2$ into square cells. Each cell is $h \times h$ where $h = \Delta x = \Delta y$ is the meshsize. We use a staggered grid with pressure p located at the center of each cell, and velocity $u = (u^1, u^2)$ located at the corners of each cell. Staggering the data is necessary for the method to work, which has been discovered experimentally [48, 29].

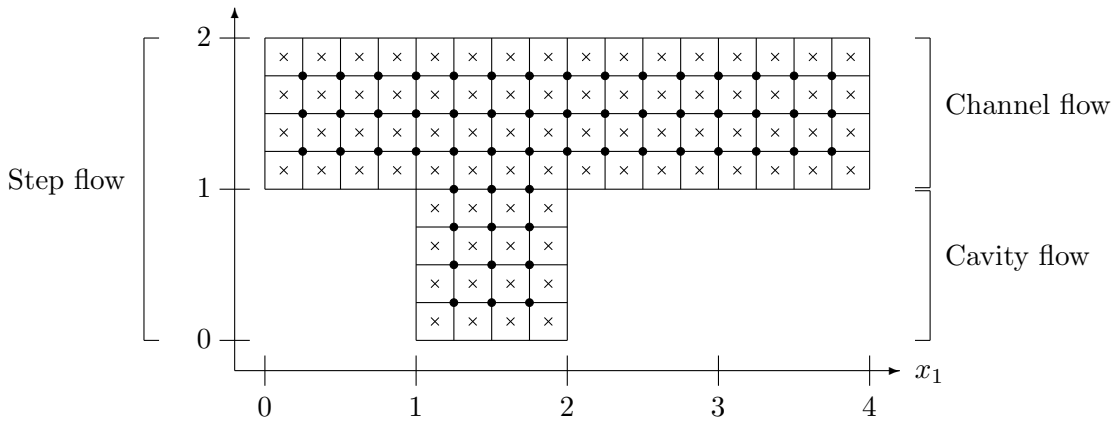

Figure 5.1: Two-dimensional staggered grid for step flow with $h \times h$ square cells, $h = 1/4$. Pressure nodes $\times$ at cell centers and velocity nodes $\bullet$ at cell corners.

In order to develop an SP algorthim for the Navier-Stokes control problem we introduce a numerical integration in space and for solutions to (5.10)–(5.11).

5.4.1 Gradient of Pressure

First we describe how to discretize ∇p on the grid (Figure 5.1) for the Navier-Stokes equation (5.11). We use the volume integral on $\widehat{\Omega}_{ij} = [x_i - \frac{h}{2}, x_i + \frac{h}{2}] \times [y_j - \frac{h}{2}, y_j + \frac{h}{2}]$, i.e.,

$$\int_{\widehat{\Omega}_{ij}} \nabla p \, d\omega = \begin{pmatrix} \int_{\widehat{\Omega}_{ij}} p_x \, dx \, dy \\ \\ \int_{\widehat{\Omega}_{ij}} p_y \, dy \, dx \end{pmatrix} = \begin{pmatrix} \int_{y_j - \frac{h}{2}}^{y_j + \frac{h}{2}} (p(x_i + \frac{h}{2}, y) - p(x_i - \frac{h}{2}, y)) \, dy \\ \\ \int_{x_i - \frac{h}{2}}^{x_i + \frac{h}{2}} (p(x, y_j + \frac{h}{2}) - p(x, y_j - \frac{h}{2})) \, dx \end{pmatrix}.$$

We use the trapezoidal rule [66]

$$\int_a^b f(x) \, dx = \frac{f(b) + f(a)}{2}(b - a) + \frac{(b - a)^3}{12} f''(c), \quad c \in [a, b]$$

to evaluate the line integrals and we obtain the second order approximation of ∇p:

$$\nabla_h p = \begin{pmatrix} \dfrac{p_{i+\frac{1}{2},j+\frac{1}{2}} + p_{i+\frac{1}{2},j-\frac{1}{2}} - p_{i-\frac{1}{2},j+\frac{1}{2}} + p_{i-\frac{1}{2},j-\frac{1}{2}}}{2h} \\ \\ \dfrac{p_{i+\frac{1}{2},j+\frac{1}{2}} + p_{i-\frac{1}{2},j+\frac{1}{2}} - p_{i+\frac{1}{2},j-\frac{1}{2}} + p_{i-\frac{1}{2},j-\frac{1}{2}}}{2h} \end{pmatrix},$$

where $p_{i+\frac{1}{2},j+\frac{1}{2}}$ is the pressure at the pressure node located at position $(x_i + \frac{h}{2}, y_j + \frac{h}{2})$ on the grid. Notice that we place the gradient of p at the velocity node, which is the center of the subdomain $\widehat{\Omega}_{i,j}$ (see Figure 5.2).

5.4.2 Divergence of Velocity

Second we describe how to discretize the incompressibility $\nabla \cdot u$ on the grid (Figure 5.1) for the Navier-Stokes equation (5.11). We use the volume integral on cell $\Omega_{ij} = [x_i, x_i + h] \times [y_j, y_j + h]$, i.e.,

$$\begin{aligned} \int_{\Omega_{ij}} \nabla \cdot u \, d\omega &= \int_{\partial\Omega_{ij}} n \cdot u \, ds \\ &= \int_{y_j}^{y_j + h} \big(u^1(x_i + h, y) - u^1(x_i, y)\big) \, dy + \int_{x_i}^{x_i + h} \big(u^2(x, y_j + h) - u^2(x, y_j)\big) \, dx. \end{aligned}$$

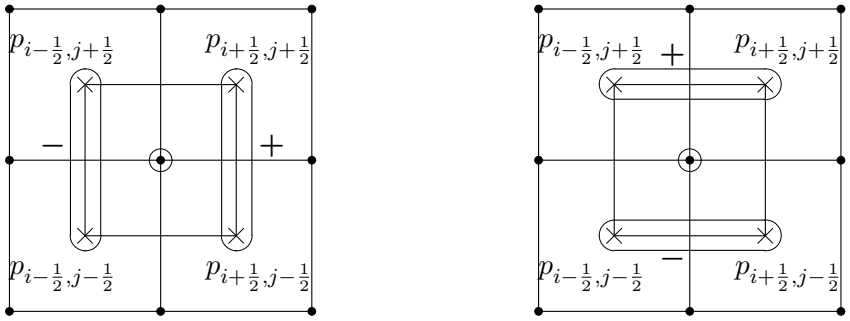

Figure 5.2: Gradient term $\nabla p = (p_x, p_y)$ located at i, j grid node.

We use the trapezoidal rule to evaluate the line integrals and we obtain the second order approximation of $\nabla \cdot u$:

$$\nabla_h \cdot u = \frac{u^1_{i+1,j+1} + u^1_{i+1,j} - u^1_{i,j+1} - u^1_{i,j}}{2h} + \frac{u^2_{i+1,j+1} + u^2_{i,j+1} - u^2_{i,j+1} - u^2_{i,j}}{2h},$$

where $u_{i,j} = (u^1_{i,j}, u^2_{i,j})$ is the velocity at the grid node located at position (x_i, y_j), see Figure 5.3. Thus, we place the divergence of u at the center of the cell (which is the pressure node).

Note that $\nabla_h \cdot \ = -(\nabla_h)^*$, i.e., the divergence approximation is the dual of the (negative) gradient approximation. This is not surprising since we have, in general,

$$\int_\Omega (\operatorname{div} u) f = \int_{\partial\Omega} (n \cdot u) f - \int_\Omega u \cdot \nabla f, \tag{5.17}$$

for all vector fields u and scalar functions f. If $n \cdot u = 0$ on $\partial\Omega$, then (5.17) reduces to

$$\int_\Omega (\operatorname{div} u) f = -\int_\Omega u \cdot \nabla f,$$

which is equivalent to

$$\langle \operatorname{div} u, f \rangle = \langle u, -\nabla f \rangle,$$

and implies that the divergence is the dual of the negative gradient.

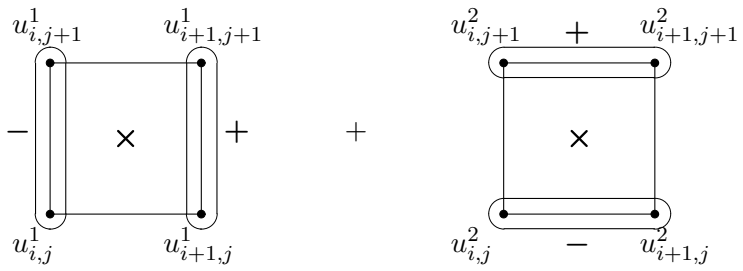

Figure 5.3: Divergence term $\operatorname{div} u = u^1_x + u^2_y$ located at i, j pressure node.

5.4.3 Convective Term

For the convective term note that for $u \in V$ and $\phi \in H^1(\Omega)$ we have

$$\begin{aligned} J_1 &= \int_{\hat{\Omega}_{ij}} u \cdot \nabla \phi \, dx = \int_{\partial \hat{\Omega}_{ij}} n \cdot (u\phi) \, ds \\ &= \int_{y_j - \frac{h}{2}}^{y_j + \frac{h}{2}} (u^1 \phi)(x_i + \frac{h}{2}, y) \, dy - \int_{y_j - \frac{h}{2}}^{y_j + \frac{h}{2}} (u^1 \phi)(x_i - \frac{h}{2}, y) \, dy \\ &+ \int_{x_i - \frac{h}{2}}^{x_i + \frac{h}{2}} (u^2 \phi)(x, y_j + \frac{h}{2}) \, dx - \int_{x_i - \frac{h}{2}}^{x_i + \frac{h}{2}} (u^2 \phi)(x, y_j - \frac{h}{2}) \, dx, \end{aligned}$$

where $\widehat{\Omega}_{ij} = [x_i - \frac{h}{2}, x_i + \frac{h}{2}] \times [y_j - \frac{h}{2}, y_j + \frac{h}{2}]$. By the trapezoidal rule we obtain

$$\begin{aligned} J_1 &\approx h \left(\Big(\frac{u^1_{i+1,j} + u^1_{i,j}}{2} \Big) \Big(\frac{\phi_{i+1,j} + \phi_{i,j}}{2} \Big) - \Big(\frac{u^1_{i-1,j} + u^1_{i,j}}{2} \Big) \Big(\frac{\phi_{i-1,j} + \phi_{i,j}}{2} \Big) \right) \\ &+ h \left(\Big(\frac{u^2_{i,j+1} + u^2_{i,j}}{2} \Big) \Big(\frac{\phi_{i,j+1} + \phi_{i,j}}{2} \Big) - \Big(\frac{u^2_{i,j-1} + u^2_{i,j}}{2} \Big) \Big(\frac{\phi_{i,j-1} + \phi_{i,j}}{2} \Big) \right). \end{aligned}$$

Thus we obtain the second order approximation of $u \cdot \nabla u$:

$$u \cdot \nabla_h u = \begin{pmatrix} u \cdot \nabla_h u^1 \\ \\ u \cdot \nabla_h u^2 \end{pmatrix},$$

where

$$
\begin{aligned}
u \cdot \nabla_h u^1 &= \frac{1}{4h}\left((u^1_{i+1,j} + u^1_{i,j})^2 - (u^1_{i-1,j} + u^1_{i,j})^2\right) \\
&\quad + \frac{1}{4h}\left((u^2_{i,j+1} + u^2_{i,j})(u^1_{i,j+1} + u^1_{i,j}) - (u^2_{i,j-1} + u^2_{i,j})(u^1_{i,j-1} + u^1_{i,j})\right) \\
u \cdot \nabla_h u^2 &= \frac{1}{4h}\left((u^1_{i+1,j} + u^1_{i,j})(u^2_{i+1,j} + u^2_{i,j}) - (u^1_{i-1,j} + u^1_{i,j})(u^2_{i-1,j} + u^2_{i,j})\right) \\
&\quad + \frac{1}{4h}\left((u^2_{i,j+1} + u^2_{i,j})^2 - (u^2_{i,j-1} + u^2_{i,j})^2\right).
\end{aligned}
$$

Figure 5.4 depicts the computation $u \cdot \nabla\phi$ on the grid, where $\phi = u^1$ or u^2.

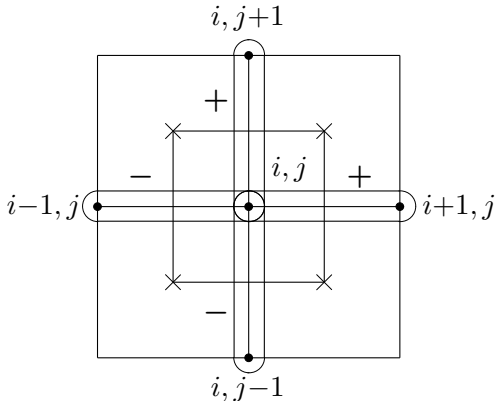

Figure 5.4: Convective term $u \cdot \nabla\phi$ (with $\phi = u^1$ or u^2) located at i, j grid node.

5.4.4 Diffusion Term

For the diffusion (Laplace) term $-\Delta u$ we have

$$
J_2 = -\int_{\hat{\Omega}_{ij}} \Delta u \, dx = -\int_{\partial\hat{\Omega}_{ij}} n \cdot \nabla u \, ds = \begin{pmatrix} -\int_{\partial\hat{\Omega}_{ij}} n \cdot \nabla u^1 \, ds \\ \\ -\int_{\partial\hat{\Omega}_{ij}} n \cdot \nabla u^2 \, ds \end{pmatrix}.
$$

For $\phi = u^1$ or u^2 we have

$$-\int_{\partial\hat{\Omega}_{ij}} n\cdot\nabla\phi\, ds = -\left(\int_{y_j-\frac{h}{2}}^{y_j+\frac{h}{2}} \frac{\partial\phi}{\partial x_1}(x_i+\frac{h}{2},y)\,dy - \int_{y_j-\frac{h}{2}}^{y_j+\frac{h}{2}} \frac{\partial\phi}{\partial x_1}(x_i-\frac{h}{2},y)\,dy \right.$$
$$\left. + \int_{x_i-\frac{h}{2}}^{x_i+\frac{h}{2}} \frac{\partial\phi}{\partial x_2}(x,y_j+\frac{h}{2})\,dx - \int_{y_j-\frac{h}{2}}^{y_j+\frac{h}{2}} \frac{\partial\phi}{\partial x_2}(x,y_j-\frac{h}{2},y)\,dx\right).$$

We use the midpoint rule

$$\int_a^b f(x)\,dx = f\left(\frac{a+b}{2}\right)(b-a) - \frac{(b-a)^3}{24}f''(c), \quad c\in[a,b]$$

and the second order central difference

$$\frac{\partial\phi}{\partial x_1}(x_i+\frac{h}{2},y) \approx \frac{\phi(x_i+h,y)-\phi(x_i,y)}{h}$$

to obtain

$$\frac{J_2}{h^2} \approx \begin{pmatrix} \dfrac{4u^1_{i,j}-u^1_{i+1,j}-u^1_{i-1,j}-u^1_{i,j+1}-u^1_{i,j-1}}{h^2} \\ \\ \dfrac{4u^2_{i,j}-u^2_{i+1,j}-u^2_{i-1,j}-u^2_{i,j+1}-u^2_{i,j-1}}{h^2} \end{pmatrix} = -(\Delta_h u).$$

Thus we place the Laplace term $-\Delta u$ at the i,j grid node (see Figure 5.5).

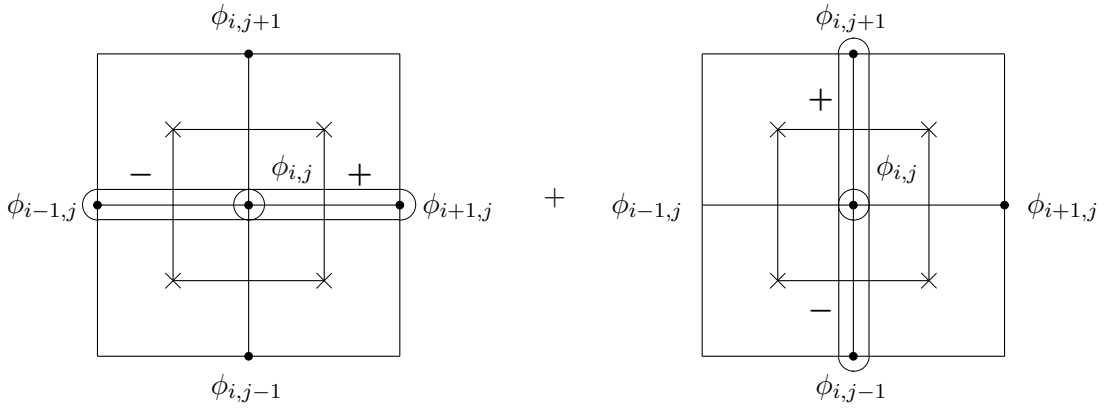

Figure 5.5: Diffusion (Laplace) term $-\Delta u = (-\Delta u^1, -\Delta u^2)$ located at i,j grid node. ($\phi = u^1$ for $-\Delta u^1$ and $\phi = u^2$ for $-\Delta u^2$.)

5.4.5 Second Order Implicit–Explicit Time Integration Method

In this section we introduce the second order implicit-explicit time integration method for the resulting approximation dynamics from Sections 5.4.1—5.4.4:

$$\frac{d}{dt}u_{i,j} + (u \cdot \nabla u)_h + \nabla_h p = \nu \Delta_h u + f_{i,j}, \quad \nabla_h \cdot u = 0. \tag{5.18}$$

For the control problem we have $f_{i,j} = Bv_{i,j}$ where v is the control and B is the input matrix (see Section 5.1). To integrate (5.18) in time we use an implicit scheme for the Stokes part and an explicit scheme with two-step method for the convective part. Given current velocities u^n and pressure p^n, as well as previous velocities u^{n-1} (for use in computing the convective term), we update u^{n+1} and p^{n+1} by solving

$$\begin{cases} \dfrac{u^{n+1} - u^n}{\Delta t} - \nu \Delta_h \left(\dfrac{u^{n+1} + u^n}{2} \right) + \dfrac{3}{2} \left(u^n \cdot \nabla_h u^n \right) - \dfrac{1}{2} \left(u^{n-1} \cdot \nabla_h u^{n-1} \right) + \nabla_h p^{n+1} = f \\ \\ \nabla_h \cdot u^{n+1} = 0. \end{cases} \tag{5.19}$$

Equivalently we solve

$$\begin{cases} \left(I - \dfrac{\nu}{2} \Delta t \Delta_h \right) u^{n+1} + \Delta t \nabla_h p^{n+1} = u^n + \dfrac{\nu}{2} \Delta t \Delta_h u^n \\ \qquad\qquad - \dfrac{3}{2} \Delta t \left(u^n \cdot \nabla_h u^n \right) + \dfrac{1}{2} \Delta t \left(u^{n-1} \cdot \nabla_h u^{n-1} \right) + \Delta t \, f \\ \\ \nabla_h \cdot u^{n+1} = 0. \end{cases}$$

This uses the Crank-Nicolson method for the Stokes operator and the two-step explicit scheme for the convective term. It is the second order Adams-Bashforth two-step method for approximating the convective term using the previous two terms.

In general, a linear multistep method is an algorithm to approximate solutions to ODEs

$$\frac{dx}{dt} = Ax + f(x) \tag{5.20}$$

where A is a matrix and f is nonlinear but quadratic. Assume A is very stiff. (5.19) corresponds to

$$\frac{x^{n+1} - x^n}{\Delta t} = A \left(\frac{x^{n+1} + x^n}{2} \right) + \frac{3}{2} f(x^n) - \frac{1}{2} f(x^{n-1}).$$

It is based on

$$\begin{aligned} x(t_{n+1}) - x(t_n) &= \int_{t_n}^{t_{n+1}} (Ax(t) + f(x(t)))dt \\ &= \frac{1}{2}A(x(t_{n+1}) + x(t_n))\Delta t + \left(\frac{3}{2}f(x^n) - \frac{1}{2}f(x^{n-1})\right)\Delta t + O(\Delta t^3). \end{aligned}$$

Thus, the resulting discretized (in time and space) optimal control problem is

$$\min \quad \sum_{n=1}^{N} \left(\ell(\frac{u^n + u^{n-1}}{2}) + h(v^{n-\frac{1}{2}})\right)\Delta t + g(u_N)$$

subject to (5.19). The SP method involves solving the saddle point problem:

$$\begin{cases} \dfrac{u^{n+1} - u^n}{\Delta t} - \nu\Delta_h\left(\dfrac{u^{n+1} + u^n}{2}\right) + \dfrac{3}{2}\left(u^n \cdot \nabla_h u^n\right) - \dfrac{1}{2}\left(u^{n-1} \cdot \nabla_h u^{n-1}\right) + \nabla_h p^{n+1} = Bv^n \\ \nabla_h \cdot u^{n+1} = 0. \\ -\dfrac{\lambda^n - \lambda^{n-1}}{\Delta t} - \nu\Delta_h\left(\dfrac{\lambda^n + \lambda^{n-1}}{2}\right) + \dfrac{3}{2}B(u^n)^*\lambda^n - \dfrac{1}{2}B(u^{n-1})^*\lambda^{n-1} + \nabla q^n + \ell'(u^n) = 0 \\ h'(v^n) = B(u^n)^*\lambda^n, \end{cases}$$

where u_0 is given, and $\lambda_N^n = g'(u_N)$.

Chapter 6

Discretization in Time and Space

In this chapter we discuss how to discretize PDE constraint problems in time and space. We then apply the SP (Sequential Programming) method described in Chapter 3 to the discretized problem to find an approximate solution to the original constrained problem.

Let $x_h \in \mathcal{C}$ be a solution to the discretized problem:

$$\min \quad F(x_h) + H(u) \quad \text{subject to} \quad E_h(x_h, u) = 0, \quad u_h \in \mathcal{C}. \tag{6.1}$$

$E_h : X \times \mathcal{C} \to Y$ represents a family of discretized problems for $h > 0$. E_h is stable and consistent in the following sense. Assume that $E_h(x_h, u) = 0$ given $u \in \mathcal{C}$ has a solution x_h, and there exists a solution x to $E(x, u) = 0$ given $u \in \mathcal{C}$, such that $x_h \to x$ in X. Let (x_h^*, u_h^*) be an optimal pair for (6.1). Then for any $\epsilon > 0$ there exists $h > 0$ such that

$$F(x_h^*) + H(u_h^*) \leq F(x) + H(u) + \epsilon \quad \text{for all } u \in \mathcal{C}.$$

Suppose there exists a weakly convergent subsequence of u_h^* to $u^* \in \mathcal{C}$ and F and H are weakly sequentially lower semicontinuous, then for all $\epsilon > 0$

$$F(x^*) + H(u^*) \leq F(x) + H(u) + \epsilon \quad \text{for all } u \in \mathcal{C},$$

and (x^*, u^*) is an optimal pair for the original contrained problem. Therefore, under the stability and consistency condition the solution pair (x_h^*, u_h^*) approximates the optimal pair (x^*, u^*) for the constrained optimization (2.3).

One can also argue the strong convergence and the convergence rate based on the well-

posedness and convergence analysis of the SP method in Section 3.4.1, i.e., we assume

$$G_h = \begin{pmatrix} Q & (E_h)_x{}^* \\ \\ (E_h)_x & 0 \end{pmatrix}$$

is uniformly invertible in $h > 0$ (sufficiently small).

Next we introduce the time integration method for controlled ODEs by the Runge-Kutta-Gauss method and the space discretization method for the incompressible Navier-Stokes control system.

6.1 Application of SP for Optimal Control Problem

In this section we introduce the Runge-Kutta-Gauss method for high order integration in time of the saddle point problem for the optimal control problem. We show how to implement the method for the discretized two point boundary value (TPBV) problem.

6.1.1 High Order Discretized Problem

In this section we discuss the optimal control problem (2.19)–(2.20) described in Section 2.4:

$$\min \quad \int_0^T (\ell(x(t)) + h(u(t))\, dt + g(x(T)) \tag{6.2}$$

subject to

$$\frac{d}{dt}x(t) = f(x(t), u(t)), \quad x(0) = x_0, \quad u(t) \in U, \tag{6.3}$$

where U is a constraint set in R^m for control $u(t)$.

We introduce the high order time integration based on the s-stage Runge-Kutta-Gauss method which is of $2s$ order, and the corresponding Gauss quadrature rule for the cost functional. The method enables us to develop the time decomposition method and we need much fewer control variables. We apply the Lagrange multiplier method for the resulting constrained optimization and derive the necessary optimality condition. It turns out it is exactly the same as applying the Runge-Kutta-Gauss method for the TPBV (two point boundary value) problem (2.24).

For the well-posedness of (6.2)–(6.3) we assume the following. The set of admissible controls is integrable, i.e.,

$$U_{ad} = \left\{ u \in L^1(0, T; R^m) : u(t) \in U \right\},$$

and (1) existence of solutions to control dynamics (6.3), and (2) existence of solution to the

optimal control problem (6.2) subject to (6.3) and $u \in U_{ad}$.

We also assume the following for ω-dissipativeness in x (6.4) and for control growth (6.5):

$$(f(x,u) - f(y,u), x - y) \leq \omega \, |x - y|^2 \quad \text{for all } u \in U, \tag{6.4}$$

and moreover that either $U \subset R^m$ is bounded or that $h(u) \geq c_1 \, |u|^2$ and

$$(f(x,u), x) \leq \omega \, |x|^2 + c_2 \, |u|^2 \tag{6.5}$$

for constants $\omega,\ c_1,\ c_2 > 0$ independent of $x,\ y \in R^n$ and $u \in U$. Also we assume that for each $(x,p) \in R^n \times R^n$ the Hamiltonian

$$u \to h(u) + p \cdot f(x,u)$$

admits a unique minimizer over U denoted by $\Psi(x,p)$. Finally we assume that $\ell,\ h,\ g$ and f are sufficiently smooth with ℓ and g bounded from below.

6.1.2 Runge-Kutta-Gauss Method

In this section we introduce the Runge-Kutta-Gauss method for the time integration of the two point boundary value problem (2.24).

We divide the horizon $[0,T]$ into N uniform intervals $[(k-1)\,\Delta t, k\,\Delta t]$, $k = 1, \ldots, N$, with $N\Delta t = T$. Let q_i, $1 \leq i \leq s$ be the s-point Gauss-Legendre quadrature points of $[-1,1]$ and define $q^{k,i} = (k-1)\Delta t + \frac{q_i+1}{\Delta t}$, the Gauss points on the subhorizon $[(k-1)\,\Delta t, k\,\Delta t]$. First, we use the Gauss quadrature rule for the cost functional:

$$\min \quad J^N(u^N) = \sum_{k=1}^{N} \sum_{i=1}^{s} w_i \, (\ell(x^{k,i}) + h(u^{k,i}))\, \Delta t + g(x^N), \tag{6.6}$$

where

$$u^N = \{\mathrm{col}(u^{k,1}, \ldots, u^{k,s})\}, \quad 1 \leq k \leq N,$$

with $u^{k,i} = u(t)$ at the time point $t = q^{k,i}$. Thus, we have control vector at the nth iterate u^n of size msN, where m is the number of controls, s is the number of stages on each subinterval, and N is the total number of subintervals on $[0,T]$. That is, the total number of unknowns is msN, which implies we have limited the number of unknowns. Number of stages s dictates accuracy on each subinterval and N dictates accuracy on the entire time horizon. Since the RKG method is $2s$ order we have accuracy in time based on $(\Delta t)^{2s} = (\frac{T}{N})^{2s}$. Thus we control the overall accuracy of the method by balancing the size of s and N together.

The implicit s-stage Runge-Kutta-Gauss (RKG) approximation uses on each interval the

piecewise polynomial approximation $x_{N,s}(t)$ of $x(t)$ which is continuous but not C^1:

$$x_{N,s}(t) = \sum_{k=0}^{s-1} \alpha_k L_k \left(2\frac{t - (k-\frac{1}{2})\Delta t}{\Delta t} \right), \quad t \in [(k-1)\Delta t, k\Delta t].$$

The method is given by

$$\begin{cases} \dfrac{x^k - x^{k-1}}{\Delta t} = \displaystyle\sum_{j=1}^{s} w_j \, f^{k,j} \\ \\ f^{k,j} = f(x^{k,j}, u^{k,j}) \\ \\ x^{k,i} = x^{k-1} + \Delta t \, \sum_{j=1}^{s} a_{i,j} f^{k,j} \end{cases} \tag{6.7}$$

where x^k are unknown states at nodes (terminal points of each subinterval) and $x^{k,i}$ are unknown states at the Gauss points. Here $(c_j, w_j),\ 1 \le j \le s$ is the Gauss-Legendre quadrature rule and the Butcher tableau is given by

$$a_{i,j} = \int_0^{c_i} L_j(t)\, dt, \quad L_j(t) = \textstyle\prod_{k \neq j} \dfrac{t - c_k}{c_j - c_k}. \tag{6.8}$$

Thus, $x^{k,j}$ approximates the value $x(t)$ at $t = q^{k,j},\ 1 \le k \le N,\ 1 \le j \le s$ and $\{x^{k,j}\}_{1 \le j \le s}$ satisfies the system of nonlinear equations, i.e., the second and third equations of (6.7). Given x^{k-1}, we solve the nonlinear system, i.e., the second and third equations of (6.7), for the intermediate values $x^{k,i}$, then update x^k by the first equation of (6.7).

For example, $s = 1$ coincides with the implicit midpoint rule, and for $s = 2$ we have

$$\begin{cases} \dfrac{x^k - x^{k-1}}{\Delta t} = w_1 \, f^{k,1} + w_2 \, f^{k,2} \\ \\ f^{k,1} = f(x^{k,1}, u^{k,1}), \quad f^{k,2} = f(x^{k,2}, u^{k,2}) \\ \\ x^{k,1} = x^{k-1} + \Delta t \, (a_{11} f^{k,1} + a_{12} f^{k,2}), \quad x^{k,2} = x^{k-1} + \Delta t \, (a_{21} f^{k,1} + a_{22} f^{k,2}) \end{cases}$$

where

$$A = \begin{pmatrix} \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \\ \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \end{pmatrix}, \quad w = \begin{pmatrix} \frac{1}{2} \\ \\ \frac{1}{2} \end{pmatrix}, \quad c = \begin{pmatrix} \frac{3-\sqrt{3}}{6} \\ \\ \frac{3+\sqrt{3}}{6} \end{pmatrix}.$$

In summary we have the discretized optimal control problem: minimize (6.4) subject to (6.5) over $u^N = \{\mathrm{col}(u^{k,1}, \ldots, u^{k,s})\}_{k=1}^N$ in U.

6.1.3 Necessary Optimality for Discretized Control Problem

In order to derive the necessary optimality condition we define the Lagrange functional for (6.4)–(6.5) by

$$\begin{aligned}\mathcal{L}(x^k, u^{k,\cdot}, f^{k,\cdot}; p^k, p^{k,\cdot}) &= \sum_{k=1}^{N}\sum_{i=1}^{s} w_i \left(\ell(x^{k,i}) + h(u^{k,i})\right)\Delta t + g(x^N) \\ &+ \sum_{k=1}^{N}\left[\sum_{i=1}^{s} w_i \left(f(x^{k,i}, u^{k,i}) - f^{k,i}\right)p^{k,i} + \left(-\frac{x^k - x^{k-1}}{\Delta t} + \sum_{i=1}^{s} w_i\, f^{k,i}\right)p^k\right]\Delta t.\end{aligned} \tag{6.9}$$

Applying the Lagrange multiplier theory with (6.9), it can be shown that the necessary optimality condition for (6.6)–(6.7) is given by

$$\begin{cases} \dfrac{x^k - x^{k-1}}{\Delta t} = \displaystyle\sum_{j=1}^{s} w_j\, f^{k,j} \\ f^{k,j} = f(x^{k,j}, u^{k,j}) \\ x^{k,i} = x^{k-1} + \Delta t \displaystyle\sum_{j=1}^{s} a_{i,j} f^{k,j} \end{cases}$$

$$\begin{cases} -\dfrac{p^k - p^{k-1}}{\Delta t} = \displaystyle\sum_{j=1}^{s} w_j\, g^{k,j} \\ g^{k,j} = f_x(x^{k,j}, u^{k,j})^t p^{k,j} + \ell_x(x^{k,j}) \\ p^{k,i} = p^k + \Delta t \displaystyle\sum_{j=1}^{s} a_{(s+1)-i,(s+1)-j} g^{k,j} \end{cases} \tag{6.10}$$

$$u^{k,j} = \Psi(x^{k,j}, p^{k,j}), \quad 1 \le k \le N, \;\; 1 \le j \le s$$

with $x^0 = x_0$ and $p^N = g_x(x^N)$. Note that (6.9) could be obtained by applying the RKG method to the two point boundary value problem for x and p, but we derived it by the Lagrange multiplier theory and variational formulation (2.16).

6.1.4 Discretized Saddle Point Problem for SP

In this section we discretize the saddle point problem (6.10) for applying the SP method. We discretize (6.10) for SP with increments in x, p, f, and g:

$$\Delta x^k = x^k_{n+1} - x^k_n, \quad \Delta x^{k,i} = x^{k,i}_{n+1} - x^{k,i}_n, \quad \Delta f^{k,j} = f^{k,j}_{n+1} - f^{k,j}_n,$$

$$\Delta p^k = p^k_{n+1} - p^k_n, \quad \Delta p^{k,i} = p^{k,i}_{n+1} - p^{k,i}_n, \quad \Delta g^{k,j} = g^{k,j}_{n+1} - g^{k,j}_n,$$

where the superscript denotes the time step and the subscript denotes the SP step.

$$\Delta x^k = x^k_{n+1} - x^k_n, \;\; \Delta x^{k-1} = x^{k-1}_{n+1} - x^{k-1}_n$$

implies

$$\frac{\Delta x^k - \Delta x^{k-1}}{\Delta t} = \sum_{j=1}^{s} w_j (f^{k,j}_{n+1} - f^{k,j}_n).$$

$$f^{k,j}_{n+1} = f(x^{k,j}_{n+1}, u^{k,j}_{n+1}) \approx f_x(x^{k,j}_n, u^{k,j}_n)(x^{k,j}_{n+1} - x^{k,j}_n) + f_u(x^{k,j}_n, u^{k,j}_n)(u^{k,j}_{n+1} - u^{k,j}_n) + f(x^{k,j}_n, u^{k,j}_n)$$

$$f^{k,j}_n = f(x^{k,j}_n, u^{k,j}_n)$$

implies

$$f^{k,j}_{n+1} - f^{k,j}_n = f_x(x^{k,j}_n, u^{k,j}_n)(x^{k,j}_{n+1} - x^{k,j}_n) + f_u(x^{k,j}_n, u^{k,j}_n)(u^{k,j}_{n+1} - u^{k,j}_n).$$

$$x^{k,i} = x^{k-1} + \Delta t \sum_{j=1}^{s} a_{i,j} f^{k,j}$$

implies

$$\Delta x^{k,i} = x^{k,i}_{n+1} - x^{k,i}_n = (x^{k-1}_{n+1} - x^{k,i}_n) + \Delta t \sum_{j=1}^{s} a_{i,j} (f^{k,j}_{n+1} - f^{k,j}_n).$$

Thus we have the discretized saddle point problem for (6.10):

$$\begin{cases} \dfrac{\Delta x^k - \Delta x^{k-1}}{\Delta t} = \sum_{j=1}^{s} w_j \Delta f^{k,j} \\ \Delta f^{k,j} = f_x(x_n^{k,j}, u_n^{k,j})\Delta x^{k,j} + f_u(x_n^{k,j}, u_n^{k,j})\Delta u^{k,j} \\ \Delta x^{k,i} = \Delta x^{k-1} + \Delta t \sum_{j=1}^{s} a_{i,j}\Delta f^{k,j} \end{cases} \tag{6.11}$$

$$\begin{cases} -\dfrac{\Delta p^k - \Delta p^{k-1}}{\Delta t} = \sum_{j=1}^{s} w_j \, \Delta g^{k,j} \\ \Delta g^{k,j} = f_x(x^{k,j}, u^{k,j})^t \Delta p^{k,j} \\ \Delta p^{k,i} = \Delta p^k + \Delta t \sum_{j=1}^{s} a_{(s+1)-i,(s+1)-j}\Delta g^{k,j}. \end{cases}$$

6.1.5 Saddle Point Solver and Preconditioner

Next we discuss our method for solving the saddle point problem (6.11). We define the unknown vector $y = \text{col}\{\text{col}((x^{k,1}, \cdots x^{k,s}, x^k), (p^{k-1}, p^{k,1}, \cdots, p^{k,s}))\}_{k=1}^{N} \in (R^{n(s+1)} \times R^{n(s+1)})^N$. Then (6.10) is a sparse system of nonlinear equations for y, i.e., the Jacobian J of equation (6.10) is a block tri-diagonal matrix with bandwidth $2n(s+1)$. The diagonal block of J has the form

$$J_{k,k} = \begin{bmatrix} A_k & S \\ Q & -A_k^t \end{bmatrix}$$

where $A_k,\ Q_k,\ S \in R^{n(s+1)\times n(s+1)}$ are defined by

$$A_k = \frac{I}{\Delta t} + \begin{bmatrix} A \\ w \end{bmatrix} diag([f_x(x^{k,1}), \cdots f_x(x^{k,s})])$$

$$Q_k = -\begin{bmatrix} A \\ w \end{bmatrix} diag(\ell_x(x^{k,i}))$$

$$S = \frac{1}{\beta}\begin{bmatrix} A \\ w \end{bmatrix} BB^t.$$

We solved the system (6.10) by Newton method in our calculations of the optimal control $\{u^{k,i}\}$. The preconditioner P for system (6.11) (matrix J) is defined by $P = diag(J_{k,k}^{-1})$ where $diag(J_{k,k})$ is the diagonal block matrix with entry $J_{k,k}$.

Remarks (1) If we apply the Newton method for nonlinear system (6.10) the Jacobian is similar to J except

$$Q_k = - \begin{bmatrix} A \\ w \end{bmatrix} diag(\{\ell_{xx}(x^{k,i}) + p^{k,i}\, f_{xx}(x^{k,i})\}_{k=1}^{s})$$

which adds the indefinite term (2nd term) for Q_k.
(2) In this sense P is the diagonal Newton preconditioner.
(3) It is an advantage of the SP method compared to Newton's method for (6.10).

Chapter 7

Concrete Examples and Numerics

In this chapter we present specific examples to demonstrate the applicability of the SP method for the constrained optimization. First, we show how to apply SP for the finite dimensional optimal control problem. We use the Lorenz attractor system as a test example. Second, we consider a non-smooth coefficient control problem in elliptic equations. The third problem is the moving damping actuator control problem. They represent the constrained minimization with technical difficulties (challenge) and show the feasibility of the SP method. Also, we present an open loop simulation for our proposed numerical method for Navier-Stokes control systems.

7.1 Lorenz Attractor With Control

In this section we discuss the optimal control of the Lorenz attractor. This is a benchmark problem for contrained optimization with equality constraint governed by an ODE. The control enters linearly into the system dynamics (we add it to the third component of state). Our purpose is to use this benchmark problem (Lorenz system) to illustrate our general purpose optimal control solver based on SP. Moreover, we show how the RKG method discussed in Section 6.1.2 is implemented for the two point boundary value problem.

The general Lorenz system (without control) takes the form

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z,$$

where $\sigma, \rho, \beta > 0$ are system parameters [45]. When $\rho > 1$ there are three equilibria

$$(0, 0, 0), (\pm\sqrt{\beta(\rho - 1)}, \pm\sqrt{\beta(\rho - 1)}, \rho - 1).$$

The two nonzero equilibria are stable [59] if and only if

$$\rho < \sigma \frac{\sigma + \beta + 3}{\sigma - \beta - 1}, \tag{7.1}$$

which can occur only when $\sigma > \beta + 1$. The stability is local stability (since the system is nonlinear) based on the eigenvalues of the linearization at an equilibrium.

In fact, the origin loses stability due to a pitchfork bifurcation [61] that occurs at parameter value $\rho = 1$. Thus, the origin is not a stable equilibrium. For the "interesting" cases (i.e., when there are three equilibria) there are two possibilities: (1) the origin is not stable, and the other two equilibria are stable, (2) all three equilibria are not stable. The two additional equilibria lose stability through a Hopf bifurcation [61] that occurs when $\rho = \sigma \frac{\sigma+\beta+3}{\sigma-\beta-1}$. Orbits are still attracted towards these equilibria, but spiral out from them. For this reason we see the well-known butterfly shape associated with the Lorenz attractor in this case.

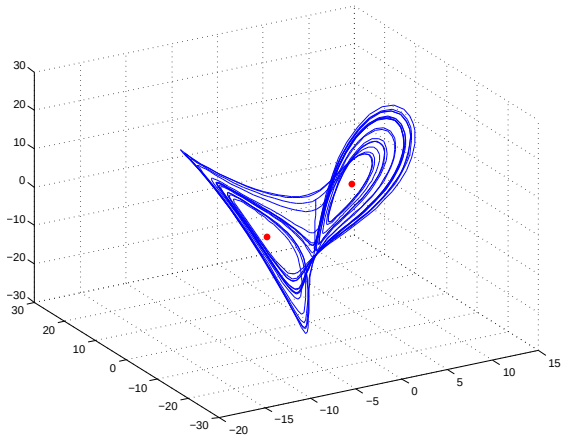

Figure 7.1: Lorenz attractor with parameters $\sigma = 4, \rho = 50, \beta = 1$ exhibits a butterfly shape.

We consider the specific problem with $\sigma = 4, \rho = 50, \beta = 1$ on a finite time horizon $[0, T]$, with change of coordinates $x = x_1$, $y = x_2$, and $z = x_3 + 50$ (the third coordinate is shifted down by -50.) Thus, our specific Lorenz system (with control) is

$$\frac{dx_1}{dt} = 4(-x_1 + x_2), \quad \frac{dx_2}{dt} = x_1 x_3 - x_2, \quad \frac{dx_3}{dt} = x_1 x_2 - x_3 - 50 + u, \tag{7.2}$$

where $u(t) \in U_{ad} = \{u \in L^2(0, T; R) : |u(t)| \leq M \text{ a.e. } t \in [0, T]\}$. The three equilibria (for the uncontrolled system) are $e_1 = (0, 0, -50)$, $e_2 = (7, 7, -1)$, and $e_3 = (-7, -7, -1)$. The first

equilibrium e_1 is not stable, and the two additional equilibria e_2 and e_3 are also not stable, which follows from (7.1) for our choice of parameters:

$$\rho = 50 > 16 = \sigma \, \frac{\sigma + \beta + 3}{\sigma - \beta - 1}.$$

The controlled Lorenz dynamics (7.2) may be written for $\vec{x} = (x_1, x_2, x_3)$ as

$$\frac{d\vec{x}}{dt} = A(\vec{x}) + Bu, \qquad \text{where} \qquad A(\vec{x}) = \begin{pmatrix} -4x_1 + 4x_2 \\ x_1 x_3 - x_2 \\ x_1 x_2 - x_3 - 50 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{7.3}$$

Linearization of the Lorenz system involves the derivative of $A(\vec{x})$, i.e.,

$$A'(\vec{x}) = \begin{pmatrix} -4 & 4 & 0 \\ -x_3 & -1 & -x_1 \\ x_2 & x_1 & -1 \end{pmatrix}. \tag{7.4}$$

We evaluate $A'(\vec{x})$ at each of the three equilibria e_1, e_2, e_3 and check the eigenvalues of the linearized system matrix. Explicitly we have

$$A'(e_1) = \begin{pmatrix} -4 & 4 & 0 \\ 50 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad A'(e_2) = \begin{pmatrix} -4 & 4 & 0 \\ 1 & -1 & -7 \\ 7 & 7 & -1 \end{pmatrix} \quad A'(e_3) = \begin{pmatrix} -4 & 4 & 0 \\ 1 & -1 & 7 \\ -7 & -7 & -1 \end{pmatrix}$$

and we have computed the eigenvalues of the $A'(e_i)$ at each equilibrium

Equilibrium	Eigenvalues of $A'(e_i)$
$e_1 = (0, 0, -50)$	$-16.7215, \;\; 11.7215, \;\; -1.0000$
$e_2 = (7, 7, -1)$	$-6.6887, \;\; 0.3443 + 7.6477i, \;\; 0.3443 - 7.6477i$
$e_3 = (-7, -7, -1)$	$-6.6887, \;\; 0.3443 + 7.6477i, \;\; 0.3443 - 7.6477i$

None of the equilibria is stable since not all real parts of eigenvalues are negative [23]. However, orbits are attracted to one of the equilibria e_2 or e_3 due to the negative real eigenvalue -6.6887, but repelled in a slow outward spiral due to the pair of imaginary eigenvalues with positive real part $0.3443 \pm 7.6477i$. This behavior is consistent with the butterfly shape observed in Figure 7.1.

Our goal is to steer the system to one of the equilibria e_2 or e_3 by control, and we choose to target the equilibrium $e_2 = (7, 7, -1)$. We design a cost functional for this purpose:

$$\int_0^T \frac{1}{2}|x|^2 + \frac{\beta}{2}|u|^2 dt + \frac{c}{2}|x(T) - x_{\text{target}}|^2, \tag{7.5}$$

with (Tikhonov) regularization parameters $\beta, c > 0$. The first term is the standard quadratic regularization cost of the state. The second term is for L^2 regularization (a fidelity) of the control. The last term is a targeting cost for the desired (target) state $x_{\text{target}} = (7, 7, -1)$. Thus, (7.2)–(7.5) is a specific case of the optimal control problem described in Section 2.4 with

$$f^0(t, x, u) = \frac{1}{2}|x|^2 + \frac{\beta}{2}|u|^2, \quad \text{and} \quad g(x(T)) = \frac{c}{2}|x(T) - x_{\text{target}}|^2.$$

The parameters $\beta > 0$ and $c > 0$ must be chosen for our purpose. We offer a parameter study in Section 7.1.3.

7.1.1 Implementation of Controlled Lorenz Problem

In this section we discuss our implementation for the optimal control problem (7.2)–(7.5). We have a very good solver based on the SP method for the two point boundary value (TPBV) problem (2.26) which is the necessary optimality system for this problem. Our implementation is in MATLAB and called Tpbdry.m. We explain the code Tpbdry.m in detail in Section 7.1.4, but start by showing how to use Tpbdry.m to solve (7.2)–(7.5).

To run Tpbdry.m we specify the number of stages s which we denote by `m`, the number of subintervals N denoted by `nt`, and the terminal time T:

```
m=5; nt=20; T=5; dt=T/nt;
```

We set flag `idx` to 0 to compute the Gauss point quadrature and weights, `imax` is for the number of inner SP loops, `beta` corresponds to $1/\beta$ where β is the coefficient for the control cost, c is the coefficient for the targeting cost, and `al` is α corresponding to the damped update in the outer SP loop.

```
idx=0; imax=1; beta=1; c=1; al=.7;
```

We set our initial condition $x(0) = x_0$ and run the two point boundary value problem solver:

```
x0=[1 1 1]'; Tpbdry;
```

The output is vector y containing the state vectors $x = (x_1, x_2, x_3)$ and adjoint vectors λ at the Gauss points and endpoints of each subinterval. Thus y has size $2n(s + 1) \times$ `nt` where $n = 3$ is the system dimension (i.e., the number of components of state).

We extract the state at each time point (Gauss points and endpoints of each subinterval) using the indices for state ix, adjoin the initial condition x_0, and reshape so that each state is a column:

```
xxx=[x0;y(ix)]; xxx=reshape(xxx,3,mp1*nt+1);
```

Thus, we can plot the orbit of x_0 (by control) over $[0, T]$:

```
plot3(xxx(1,:),xxx(2,:),xxx(3,:)); grid on
```

Since we set imax to 1 we did one SP step. To repeat the process we set flag idx to 1 so that the code Tpbdry does not recompute the Gauss points and weights (we can reuse them). Now, running Tpbdry again uses the update y from the first iteration and outputs the next damped update:

```
idx=1; Tpbdry;
```

Thus we obtain a sequence of damped updates $y_1, y_2, y_3, \ldots$. Figure 7.2 shows the orbit of x_0 for 10 iterations. We can see the iterations converging to a limit (bold curve).

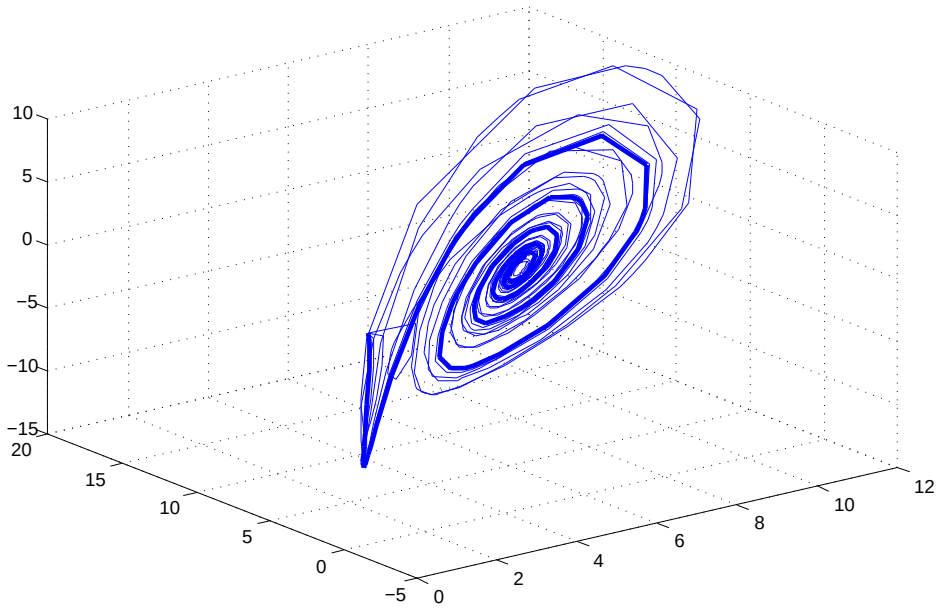

Figure 7.2: Iterates $x_1, \ldots, x_{10}$ by Tpbdry.m for the controlled Lorenz system. Parameters $\beta = 1, c = 1$ over time horizon $[0, 5]$ with 5 stages and 20 subintervals, targeting equilibrium $(7, 7, -1)$ with initial condition $x_0 = (1, 1, 1)$ and damping parameter $\alpha = .7$.

We can implement the TPBV solver for other choices of damping parameter α, control cost parameter β, and targeting cost parameter c with the following MATLAB script:

```
m=5; nt=20; T=5; dt=T/nt; idx=0; imax=1; beta=1; c=1; al=.7;
x0=[1 1 1]'; Tpbdry; xxx=[x0;y(ix)]; xxx=reshape(xxx,3,mp1*nt+1);
plot3(xxx(1,:),xxx(2,:),xxx(3,:)); grid on;
idx=1; Tpbdry; xxx=[x0;y(ix)]; xxx=reshape(xxx,3,mp1*nt+1);
hold on; plot3(xxx(1,:),xxx(2,:),xxx(3,:))
for i=1:8, Tpbdry; xxx=[x0;y(ix)]; xxx=reshape(xxx,3,mp1*nt+1);
plot3(xxx(1,:),xxx(2,:),xxx(3,:)) end
```

Parameters α, β, c may be changed in the first line of code.

The output vector y of Tpbdry also contains the adjoint vector λ which we use to determine the control u by the optimality condition:

$$u = -\frac{1}{\beta} B^t \lambda.$$

We extract u from λ with

```
ppp=y(ip); pT=xxx(:,end)-[7 7 -1]'; pT=c*pT;
ppp=[ppp;pT]; ppp=reshape(ppp,3,mp1*nt+1);
uuu=-beta*ppp(3,:);
```

Here ip is the index for the adjoint components of y at the Gauss points and initial (left) endpoints of each subinterval. We compute the terminal condition $\lambda(T) = g'(x(T))$ and adjoin to λ in the second line. Since our input matrix is $B = (0, 0, 1)^t$ we can form the product $B^t\lambda$ by just picking off the third component of the adjoint at each time point, which we do in the third line. In Figure 7.3 we plot the sequence of controls $u_1, \ldots, u_{10}$ corresponding to states $x_1, \ldots, x_{10}$ from Figure 7.2. We can see the iterations converging to a limit (bold curve).

Remark Notice in Figure 7.3 that the optimal control u^* (i.e., the limit of the SP updates u_i) does not appear to be very smooth. This is because we have found a finite dimensional (vector) approximation of what should be a function $u(t)$. We can achieve a better result by increasing the number of stages s or total number of subintervals N. In Figure 7.4 we exhibit the effect when N is increased from 20 to 40 and 80 (i.e., the N is doubled each time). Alternatively, we can take the vector approximation and fit with a polynomial interpolate (e.g., a Legendre interpolating polynomial).

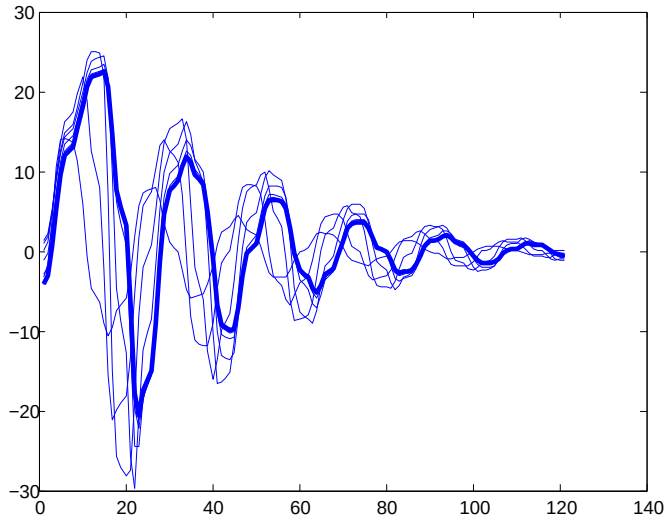

Figure 7.3: Controls $u_1, \ldots, u_{10}$ corresponding to states $x_1, \ldots, x_{10}$ in Figure 7.2.

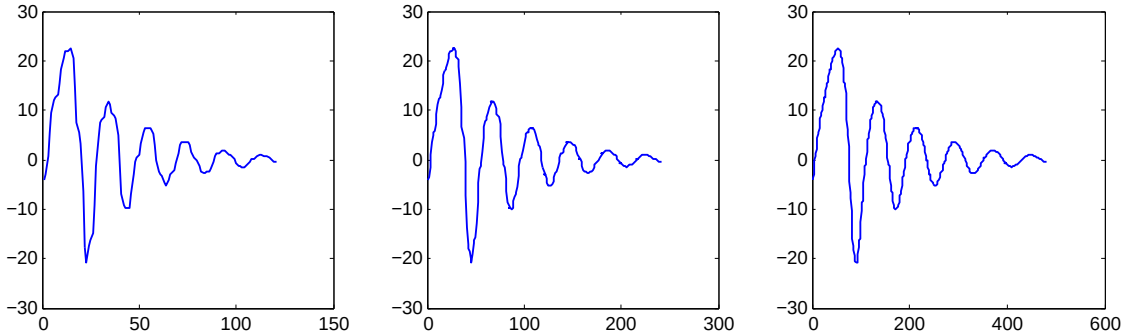

Figure 7.4: Control u after 10 iterations for varying number of subintervals $N = 20, 40, 80$.

7.1.2 Cost and Convergence Study

In this section we discuss how to compute the cost (7.5) and analyze the convergence rate of SP iterates. We study the effect on cost and convergence of varying the number of stages s and subintervals N.

Recall the cost functional (6.6) for the RKG time integration method, i.e.,

$$J^N(u^N) = \sum_{k=1}^{N} \sum_{i=1}^{s} w_i \left(\ell(x^{k,i}) + h(u^{k,i})\right) \Delta t + g(x^N), \tag{7.6}$$

where $u^N = \{\mathrm{col}(u^{k,1}, \ldots, u^{k,s})\}_{k=1}^{N}$. We obtain state $x = (x^{k,i})$ and control $u = (u^{k,i})$ as described in Section 7.1.1, and then compute the cost (7.6) as follows.

First, we compute the state cost, i.e., the term involving $\ell(x^{k,i})$:

```
x=y(ii); x=reshape(x,3,m*nt); x=x.*x; x=x(1,:)+x(2,:)+x(3,:);
x=reshape(x,m,nt)*ones(nt,1); l=.5*w'*x;
```

Then, we compute the control cost, i.e., the term involving $h(u^{k,i})$:

```
p=y(jj); p=reshape(p,3,m*nt); u=p(3,:).*p(3,:);
u=reshape(u,m,nt)*ones(nt,1); h=.5*beta*w'*u;
```

Lastly, we compute the terminal cost $g(x^N)$:

```
g=y(i0); g=g(end-2:end); g=g-[7 7 -1]'; g=.5*c*g'*g;
```

Then the total cost $J^N(u^N)$ is given by

```
J=dt*(l+h)+gg;
```

We examine the effect on the cost of varying the number of stages s and number of subintervals N. To make a fair comparison we require that $sN = 100$, i.e., the total number of unknowns is the same for each choice of s and N. Also we use the same parameter values $\beta = 1, c = 1, \alpha = .7, T = 5$ for each. In Table 7.1 we compute the cost for various choices of s and N and observe the cost over 10 iterations.

We can also judge how well we targeted the desired state $x_{\text{target}} = (7, 7, -1)$ by computing $|x(T) - x_{\text{target}}|^2$:

```
xT=gg/(.5*c);
```

In Table 7.2 we compute $|x(T) - x_{\text{target}}|^2$ for various choices of s and N with $sN = 100$ for 10 iterations.

Table 7.1: Cost $J^N(u^N)$ for varying numbers of stages s and subintervals N over 10 iterations.

$s =$	1	2	4	5	10	20
$N =$	100	50	25	20	10	5
1	872.4280	377.9610	303.6741	295.0073	282.0225	282.1531
2	714.6322	419.9762	362.5071	353.6355	344.6961	364.6363
3	2675.7616	456.1193	435.5630	439.4228	447.1956	428.8027
4	2055.9682	460.3020	460.3399	464.9749	465.6503	445.9826
5	6203.1047	454.8385	455.9088	457.4785	457.6720	450.6510
6	3600.2891	452.7456	453.1277	453.6627	453.7255	451.7035
7	3156.8460	452.1051	452.2196	452.4364	452.3974	451.9780
8	2640.8438	451.9125	451.9428	452.0631	451.9856	452.0562
9	2036.8921	451.8547	451.8595	451.9507	451.8608	452.0794
10	2035.2123	451.8373	451.8344	451.9170	451.8232	452.0863

Table 7.2: Computation of $|x(T) - x_{\text{target}}|^2$ for varying numbers of stages s and subintervals N over 10 iterations.

$s =$	1	2	4	5	10	20
$N =$	100	50	25	20	10	5
1	132.6040	13.3909	3.2982	2.9440	3.0459	8.2824
2	7.9952	1.2650	0.3652	0.4968	0.1814	1.3819
3	7.7708	0.4418	0.5654	0.6684	0.7336	1.2955
4	27.4051	0.6731	0.7348	0.8996	0.7954	0.9113
5	333.4218	0.5682	0.6653	0.7764	0.6497	0.8652
6	156.4473	0.5168	0.6062	0.6964	0.5909	0.8517
7	127.7013	0.5003	0.5851	0.6692	0.5720	0.8477
8	294.6828	0.4954	0.5785	0.6607	0.5662	0.8464
9	162.5837	0.4938	0.5765	0.6582	0.5645	0.8461
10	68.1464	0.4934	0.5758	0.6574	0.5640	0.8460

Now, we examine the convergence rate of iterations by varying the (damping) stepsize $\alpha > 0$ for the damped update $y_{n+1} = y_n + \alpha(y^+ - y_n)$. We compute the relative error between state iterates, i.e., $err_{n+1} = |x_{n+1} - x_n|/|x_n|$ where x_n is the state component of iterate $y_n = (x_n, \lambda_n)$. We fix $s = 5$ and $N = 20$ and compute the relative errors for the choices $\alpha = .1$, .3, .5, .7, and .9 over 10 iterations (see Table 7.3).

Table 7.3: Relative error $err_{n+1} = |x_{n+1} - x_n|/|x_n|$ between state iterates for varying (damping) stepsizes $\alpha = .1, .3, .5, .7, .9$ over 10 iterations, with $s = 5$ stages and $N = 20$ subintervals.

$\alpha =$	.1	.3	.5	.7	.9
err_1	0.0346	0.1019	0.1968	0.3344	0.4948
err_2	0.0329	0.1021	0.2078	0.3586	0.5831
err_3	0.0316	0.1028	0.1845	0.2996	1.0196
err_4	0.0308	0.0984	0.1450	0.1589	0.4842
err_5	0.0303	0.0896	0.0979	0.0541	0.5036
err_6	0.0300	0.0787	0.0568	0.0168	0.4391
err_7	0.0299	0.0668	0.0301	0.0051	0.3755
err_8	0.0298	0.0546	0.0154	0.0015	0.1654
err_9	0.0295	0.0431	0.0077	0.0005	0.0443

We compute the rate of decrease in (relative) error between iterations by taking the ratio of the current error and the previous error, i.e., $r_n = err_{n+1}/err_n$. This is the rate of convergence of the SP implementation. According to the data in Table 7.4, the convergence rate of our SP implementation Tpbdry.m is on the order of $1 - \alpha$, which verifies the convergence analysis of the SP method discussed in Section 3.4.1.

7.1.3 Parameter Study for Controlled Lorenz Problem

In this section we discuss the effects of varying parameters β and c for the control cost and targeting cost (7.5). We also examine changing the terminal time T. We fix parameters $s = 5$ stages, $N = 20$ subintervals, and $\alpha = .7$ for all tests in this section.

First we examine the effect of varying the control authority parameter β, which penalizes the control cost. Figure 7.5 exhibits the orbit of x_0 (by control) and corresponding control u for three choices $\beta = 1, \frac{1}{10}, \frac{1}{100}$ after 10 iterations. We see that a small β (e.g., $\beta = \frac{1}{100}$) penalizes the cost (7.5) less for having a control. Thus, the control u can be larger (in norm) and have more influence over the state dynamics. What we see is the corresponding orbit of initial condition x_0 being steered to the target equilibrium more quickly in a more direct path. Conversely, a large

Table 7.4: Convergence rates $r_n = err_{n+1}/err_n$ corresponding to iterates in Table 7.3 for varying (damping) stepsizes $\alpha = .1, .3, .5, .7, .9$ over 10 iterations.

$\alpha =$	.1	.3	.5	.7	.9
r_1	0.9513	1.0022	1.0562	1.0724	1.1784
r_2	0.9622	1.0067	0.8875	0.8353	1.7486
r_3	0.9738	0.9571	0.7862	0.5303	0.4748
r_4	0.9839	0.9111	0.6754	0.3404	1.0402
r_5	0.9911	0.8784	0.5795	0.3113	0.8719
r_6	0.9948	0.8486	0.5301	0.3032	0.8552
r_7	0.9952	0.8177	0.5109	0.3009	0.4405
r_8	0.9932	0.7883	0.5041	0.3002	0.2681

β (e.g., $\beta = 1$) penalizes the cost more for having a control, and control u must be smaller, and has less influence on the state dynamics. Thus, we see the orbit of x_0 being steered to the target more slowly in a less direct path.

For the case $\beta = \frac{1}{100}$ (rightmost Figure 7.5) notice that the control u is effectively zero after time $t \approx 1.25$ (i.e., after about one fourth of the terminal time $T = 5$). We deduce that we should be able to effectively steer x_0 to the target equilibrium in a shorter time, i.e., over a shorter time horizon, since most of the controlling is done in the first quarter of the given time horizon. We test this by varying the terminal time $T = 5$ to $T = 2$ and to $T = 1$ in Figure 7.6. Note that we focus on the controls u only since the corresponding trajectories were found to be the same practically. That is, we were able to effectively steer initial state x_0 to the target state x_{target} by each of the controls in Figure 7.6.

Finally, we explore the effect of changing the target authority parameter c. We use $\beta = 1$ over time horizon $[0, 5]$ with 10 iterations, and vary parameter $c = \frac{1}{10}$ to $c = 1$ and $c = 10$ (see Figure 7.7). Observe that a small c (e.g., $c = \frac{1}{10}$) penalizes the cost (7.5) less for the state at terminal time $x(T)$ being far from the target x_{target}, i.e., $|x(T) - x_{\text{target}}|^2$ may be large. Indeed, we see in Figure 7.7 that the trajectory of x_0 is still spiraling around the target equilibrium by terminal time T. On the other hand, a large c (e.g., $c = 10$) puts a strong penalty on missing the target, and thus the trajectory of x_0 has almost spiraled to the target equilibrum by terminal time T. In fact, we have computed $|x(T) - x_{\text{target}}|^2$ for each $c = \frac{1}{10}, 1, 10$ as 1.9256, 0.5741, and 0.0190, respectively, which confirms our analysis.

7.1.4 The Code Tpbdry.m

In this section we discuss our MATLAB implementation Tpbdry.m for the Lorenz attractor system with control (7.2)–(7.5). The code is a direct implementation of the saddle point solver

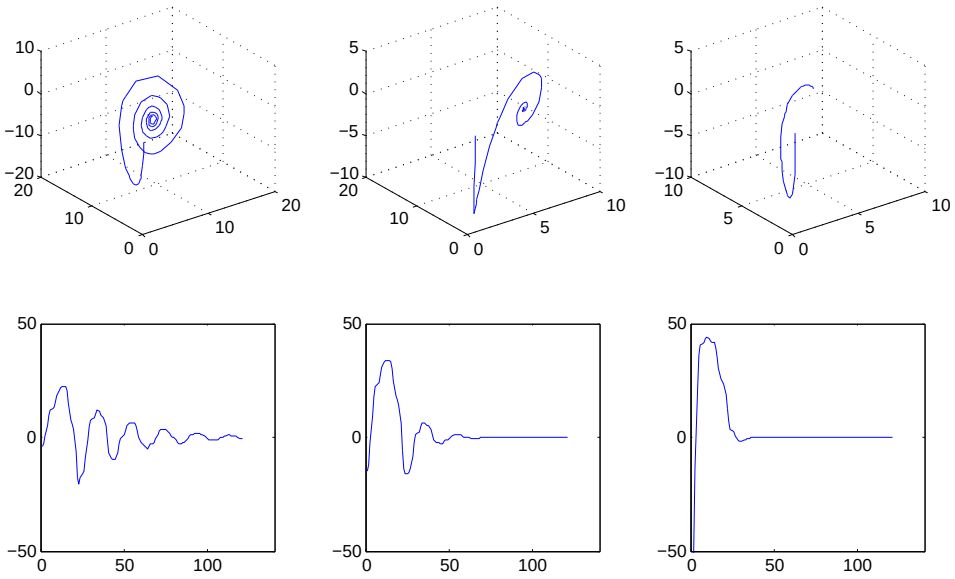

Figure 7.5: Orbit of $x_0 = (1, 1, 1)$ and corresponding control u for varied control authority parameter $\beta = 1, \frac{1}{10}, \frac{1}{100}$. (Other parameters $\alpha = .7$ and $c = 1$, over time horizon $[0, 5]$.)

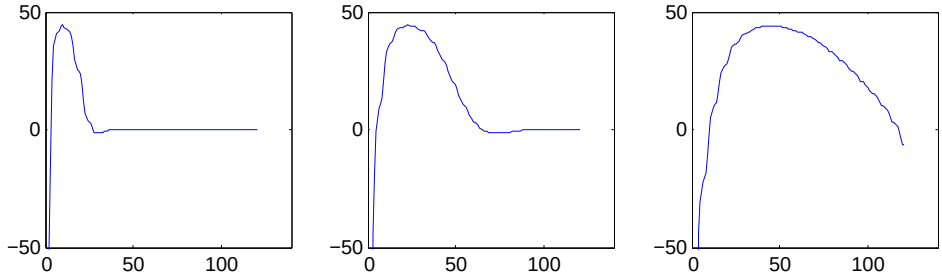

Figure 7.6: Effect on control u of varying terminal time $T = 5, 2, 1$.

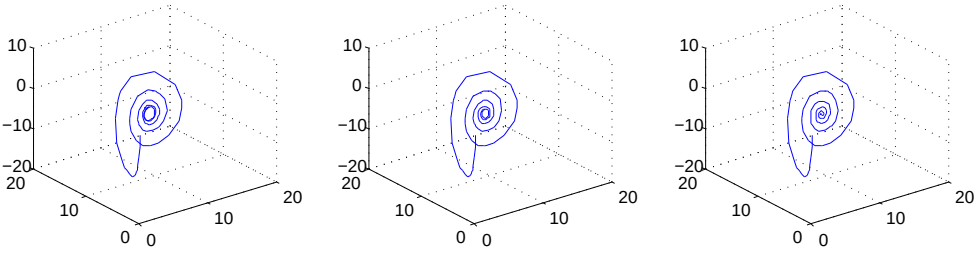

Figure 7.7: Effect on orbit of x_0 of varying target authority parameter $c = \frac{1}{10}, 1, 10$.

described in Section 6.1.5. We first present the code Tpbdry.m in full.

```
% Tpbdry.m
if idx==0; n=3; mp1=m+1; n0=mp1*n*nt; n1=m*n*nt; n2=2*n0;
mn=m*n; mn1=mp1*n; mn2=2*mn1;

x=1:1:m-1; x=x./sqrt((2*x+1).*(2*x-1));
j=diag(x,1)+diag(x,-1); [u x]=eig(j); x=diag(x); [x i]=sort(x);
u=u(:,i); w=u(1,:).^2; w=w'; x=.5*(x+1);

for i=1:m; yy=x(i)-x; yy(i)=x(i); a(i,i)=sum(1./yy);
for j=1:i-1; z=x(j)-x; z(j)=x(j); s=-real(exp(sum(log(yy./z))));
a(i,j)=s/z(i); a(j,i)=1/s/yy(j); end; end; a=inv(a);

y=[7 7 -1]'*ones(1,mp1); y=y(:); y=[y;0*y]*ones(1,nt); y=y(:);
a1=[a;w']; a1=kron(a1,speye(n)); a1=kron(speye(nt),a1);
a2=[w';a(m:-1:1,m:-1:1)]; a2=kron(a2,speye(n)); a2=kron(speye(nt),a2);
j=reshape(1:1:n2,mn2,nt); ix=j(1:mn1,:); ip=j((mn1+1):mn2,:);
ii=j(1:mn,:);jj=j((n*(m+2)+1):mn2,:); i0=j(mn+1:mn1,:);j0=j(mn1+1:mn1+n,:);
j=1:1:n1; i1=j(1:n:n1); i2=j(2:n:n1); i3=j(3:n:n1);
d=zeros(n1,1); d1=d; d2=d; d3=d; d4=d; d5=d; d6=d; d7=d; d8=d;
d(i1)=7*ones(m*nt,1); d(i2)=d(i1); d(i3)=-ones(m*nt,1);
q0=speye(n1); d3(i3)=ones(m*nt,1); bb=beta*spdiags(d3,0,n1,n1);
J0=speye(n2)/dt; J0(ix,jj)=a1*bb; e=kron(ones(mp1,1),speye(n));
a0=kron(speye(nt),e); e=e/dt;
for i=2:nt;  J0(ix(:,i),i0(:,i-1))=-e; J0(ip(:,i-1),j0(:,i))=-e; end;
i00=i0(:,nt); i0=i0(:,1:(nt-1)); j0=j0(:,2:nt); j00=ip(:,nt);
J0(j00,i00)=-c*e; ix=ix(:); ip=ip(:); ii=ii(:); jj=jj(:); i0=i0(:); j0=j0(:);
w1=ones(n,m)*spdiags(w,0,m,m); w1=w1(:)*ones(1,nt); w1=w1(:);
w2=w*ones(1,nt); w2=w2(:); end

for iout=1:imax; J=J0;
yy=y(ii); x1=yy(i1); x2=yy(i2); x3=yy(i3);
g(i1)=4*(-x1+x2); g(i2)=-x2-x1.*x3; g(i3)=x1.*x2-x3-50;
f(ix)=(y(ix)-a0*[x0;y(i0)])/dt-a1*(g'-bb*y(jj));
d1(i1)=x2; d2(i1)=-x3; d2(i2)=x1;
d3(i1)=-4*ones(nt*m,1); d3(i2)=-ones(nt*m,1); d3(i3)=d3(i2);
```

```
d4(i2)=-d3(i1); d4(i3)=-x1;
tmp=spdiags([d1 d2 d3 d4],-2:1,n1,n1);
J(ix,ii)=J(ix,ii)-a1*tmp; J(ip,jj)=J(ip,jj)-a2*tmp';
f(ip)=(y(ip)-a0*[y(j0);c*(y(i00)-[7 7 -1]')])/dt-a2*(tmp'*y(jj)+y(ii)-d)

z=y(jj);
d1(i1)=-z(i2); d6(i1)=z(i3); d7(i2)=z(i3); d8(i3)=-z(i2);
qq=q0+spdiags([d1 d6 d7 d8],[-2 -1 1 2],n1,n1);
J(ip,ii)=J(ip,ii)-a2*qq; gg=-J\f';
y=y+al*gg; end
```

Now we explain aspects of the code in detail. Recall from Section 7.1.1 we must first specify the number of stages s denoted by m, and number of subintervals N denoted by nt, and terminal time T. If flag idx $= 0$ we compute dimensions for the problem:

```
if idx==0; n=3; mp1=m+1; n0=mp1*n*nt; n1=m*n*nt; n2=2*n0;
mn=m*n; mn1=mp1*n; mn2=2*mn1;
```

The system dimension is n $= 3$, i.e., the number of components of state. mp1 $= m + 1$ is the number of stages plus one, which is for the number of time points on each subinterval, plus one for the endpoint of the subinterval. n0 is the total number of unknowns for state for all stages and subintervals, including endpoints. n1 is the total number of unknowns for all stages and subintervals, excluding endpoints. n2 is the total number of unknowns for both state and adjoint. mn is the number of components of state (or adjoint) at all stages of a subinterval. mn1 is the number of components of state (or adjoint) at all stages and the endpoint of a subinterval. mn2 is the total number of unknown components for both state and adjoint at all stages, including endpoints, for all subintervals.

We also need to compute the Gauss points and weights for the RKG method:

```
x=1:1:m-1; x=x./sqrt((2*x+1).*(2*x-1));
j=diag(x,1)+diag(x,-1); [u x]=eig(j); x=diag(x); [x i]=sort(x);
u=u(:,i); w=u(1,:).^2; w=w'; x=.5*(x+1);

for i=1:m; yy=x(i)-x; yy(i)=x(i); a(i,i)=sum(1./yy);
for j=1:i-1; z=x(j)-x; z(j)=x(j); s=-real(exp(sum(log(yy./z))));
a(i,j)=s/z(i); a(j,i)=1/s/yy(j); end; end; a=inv(a);
```

The matrix a is the Runge-Kutta matrix and w are the weights for the Gauss quadrature. We use a and w to form a1 for the forward solution of state and a2 for the backward solution of adjoint:

```
y=[7 7 -1]'*ones(1,mp1); y=y(:); y=[y;0*y]*ones(1,nt); y=y(:);
a1=[a;w']; a1=kron(a1,speye(n)); a1=kron(speye(nt),a1);
a2=[w';a(m:-1:1,m:-1:1)]; a2=kron(a2,speye(n)); a2=kron(speye(nt),a2);
```

y sets an initial iterate y_0 for SP. The form of matrices a1 and a2 is important to relate to the RKG method, i.e., a1 includes the Runge-Kutta matrix a and weights w in order to set up the discretized system (6.11) for the state variable:

$$\frac{\Delta x^k - \Delta x^{k-1}}{\Delta t} = \sum_{j=1}^{s} w_j \, \Delta f^{k,j}$$

$$\Delta f^{k,j} = f_x(x^{k,j}, u^{k,j})\Delta x^{k,j} + f_u(x^{k,j}, u^{k,j})\Delta u^{k,j}$$

$$\Delta x^{k,i} = \Delta x^{k-1} + \Delta t \sum_{j=1}^{s} a_{i,j} \Delta f^{k,j}.$$

Note that for SP we use the linearized state dynamics, i.e., given current state iterate $x_n = (x_n^{k,i})$, $k = 1, \ldots, s+1$, where $x_n^{k,s+1} = x_n^{k+1}$

$$f^{k,i} = f(x^{k,i}, u^{k,i}) = A'(x_n^{k,i})(x^{k,i} - x_n^{k,i}) + A(x_n^{k,i}) - \frac{1}{\beta} B B^t p^{k,i}$$

where we used the optimality condition $u = -\frac{1}{\beta} B^t \lambda$ to eliminate u as a function of $\lambda = p$, and $A(x)$ and $A'(x)$ are the matrices defined in (7.3)–(7.4). Similarly, a2 is set up to solve the discretized system (6.11) for the adjoint.

We set up index counters for components of state, denoted ix, and components of adjoint, denoted ip.

```
j=reshape(1:1:n2,mn2,nt); ix=j(1:mn1,:); ip=j((mn1+1):mn2,:);
ii=j(1:mn,:);jj=j((n*(m+2)+1):mn2,:); i0=j(mn+1:mn1,:);j0=j(mn1+1:mn1+n,:);
j=1:1:n1; i1=j(1:n:n1); i2=j(2:n:n1); i3=j(3:n:n1);
```

i1, i2, i3 are for extracting the first, second, or third components of state or adjoint. Next, we form d for the targeting cost, and bb for the input matrix $B = (0, 0, 1)^t$:

```
d=zeros(n1,1); d1=d; d2=d; d3=d; d4=d; d5=d; d6=d; d7=d; d8=d;
d(i1)=7*ones(m*nt,1); d(i2)=d(i1); d(i3)=-ones(m*nt,1);
q0=speye(n1); d3(i3)=ones(m*nt,1); bb=beta*spdiags(d3,0,n1,n1);
```

Finally, we form the system matrix of (6.10):

```
J0=speye(n2)/dt; J0(ix,jj)=a1*bb; e=kron(ones(mp1,1),speye(n));
a0=kron(speye(nt),e); e=e/dt;
for i=2:nt;  J0(ix(:,i),i0(:,i-1))=-e; J0(ip(:,i-1),j0(:,i))=-e; end;
i00=i0(:,nt); i0=i0(:,1:(nt-1)); j0=j0(:,2:nt); j00=ip(:,nt);
J0(j00,i00)=-c*e; ix=ix(:); ip=ip(:); ii=ii(:); jj=jj(:); i0=i0(:); j0=j0(:);
w1=ones(n,m)*spdiags(w,0,m,m); w1=w1(:)*ones(1,nt); w1=w1(:);
w2=w*ones(1,nt); w2=w2(:); end
```

This completes the overhead for flag idx $= 0$.

Next we have the solve step, which includes the forward solve for state and backward solve for adjoint. The forward solve involves

```
for iout=1:imax; J=J0;
yy=y(ii); x1=yy(i1); x2=yy(i2); x3=yy(i3);
g(i1)=4*(-x1+x2); g(i2)=-x2-x1.*x3; g(i3)=x1.*x2-x3-50;
f(ix)=(y(ix)-a0*[x0;y(i0)])/dt-a1*(g'-bb*y(jj));
```

The controlled Lorenz dynamics (7.2) are given by g, where x1, x2, x3 are components of state.

Next we compute and use the derivative of the state dynamics:

```
d1(i1)=x2; d2(i1)=-x3; d2(i2)=x1;
d3(i1)=-4*ones(nt*m,1); d3(i2)=-ones(nt*m,1); d3(i3)=d3(i2);
d4(i2)=-d3(i1); d4(i3)=-x1;
tmp=spdiags([d1 d2 d3 d4],-2:1,n1,n1);
J(ix,ii)=J(ix,ii)-a1*tmp; J(ip,jj)=J(ip,jj)-a2*tmp';
f(ip)=(y(ip)-a0*[y(j0);c*(y(i00)-[7 7 -1]')])/dt-a2*(tmp'*y(jj)+y(ii)-d)
```

The matrix tmp corresponds to (7.4).

Lastly, we insert the Q matrix which appears in the quadratic state cost $\frac{1}{2}(Qx, x)$:

```
z=y(jj);
d1(i1)=-z(i2); d6(i1)=z(i3); d7(i2)=z(i3); d8(i3)=-z(i2);
qq=q0+spdiags([d1 d6 d7 d8],[-2 -1 1 2],n1,n1);
J(ip,ii)=J(ip,ii)-a2*qq; gg=-J\f';
y=y+al*gg; end
```

Thus the system matrix J is formed, and the right hand side f. We solve for increments $(\Delta x, \Delta\lambda)$, the result is gg. Finally we compute the damped update y, which includes both the state x and the adjoint λ, i.e., $y = (x, \lambda)$.

7.2 Coefficient Optimal Control in Elliptic Equations

We consider the non-smooth constraint optimization as in Section 4.3:

$$\min \quad \int_\Omega \frac{1}{2}|y - y_d|^2 \, d\omega + \int \frac{\alpha}{2}|u|^2 + \beta|u|$$

subject to

$$-\epsilon \, \Delta y + f(y) + uy = f_0, \quad y = 0 \text{ at } \partial\Omega,$$

where $u(x) \in L^2(\Omega)$ is the coefficient control (bilinear term) and y_d is the desired state in $L^2(\Omega)$. We let $\alpha = 1 \times 10^{-5}$, $\beta = 5 \times 10^{-5}$, $\epsilon = .1$ and

$$y_d = (-\Delta + diag(u_0))^{-1} f_0, \quad f_0 = \sin(\pi(x_1 + x_2)),$$

where $u_0 = 1 + \sin(\pi(x_1 + x_2))$. We initialized with $u = ones$ and $y = (-\Delta + diag(u))^{-1} f_0$. We tested for the different nonlinear functions

$$f(y) = |y|, \;\; -y^3, \;\; \text{and} \;\; -\exp(y)$$

which represent several challenges, i.e., we must adjust a damping weight α in (3.16) according to the difficulty of the original optimization problem. For example, we set $\alpha = .3$ for the last example $f(y) = -\exp(y)$ for good convergence of the method. The first example $f(y) = |y|$ is not C^1. For the second example $f(y) = -y^3$, the solution may blow up without control. The last problem with $f(y) = -\exp(y)$ is harder because, first we made $\epsilon = .01$ very small, the exponential function is highly nonlinear, and the solution to the equality constraint may blow up without control. This is a model for explosion.

The following MATLAB code performs SP ($s = 0$) and partial SQP ($s = 1$) for $f(y) = |y|$.

```
n=100; m=n-1; dx=1/n;
h=ones(m,1); h=n^2*spdiags([-h 2*h -h],-1:1,m,m);
h=kron(speye(m),h)+kron(h,speye(m));
[x y]=meshgrid([dx:dx:1-dx]); f0=sin(pi*(x(:)+y(:))); u0=1+sin(pi*(x+y));
f=(h+spdiags(u0(:),0,m^2,m^2))\f0;
u=ones(m^2,1); y=(.1*h+spdiags(u,0,m^2,m^2))\f0; p=0*u; uu=0*u;
s=0; al=.7; bt=1.e-6;
H=.1*h+spdiags(sign(y)+u(:),0,m^2,m^2);
b=bt*ones(m^2,1); k=find(abs(y.*p)<5*bt); b(k)=1.e8;
A=[speye(m^2) s*spdiags(p,0,m^2,m^2) H;
   s*spdiags(p,0,m^2,m^2) spdiags(b,0,m^2,m^2)  spdiags(y,0,m^2,m^2)];
```

```
A=[A; H  spdiags(y,0,m^2,m^2) zeros(m^2)];
tmp=A\[f-y-H*p;-b.*u-5*bt*sign(u)-y.*p;f0-.1*h*y-abs(y)-y.*u];
y=y+al*tmp(1:m^2); u=u+al*tmp(m^2+1:2*m^2); p=p+al*tmp(2*m^2+1:3*m^2);
norm(tmp(m^2+[1:m^2]))/norm(u)
k=find(abs(y.*p)<5*bt); u(k)=0;
```

This uses exactly the algorithm in Section 4.4. For example, the last line in the code does the projection.

The following figures summarize our findings. We use the monitoring index

$$\frac{|u_{n+1} - u_n|}{|u_n|}$$

which is used for determining the convergence of the SP algorithm and the convergence rate along the norm of equation error, i.e., the error in the necessary optimality. For example, for the case of *fidelity* $= 1$ (on the left columns) the rate is less than .3.

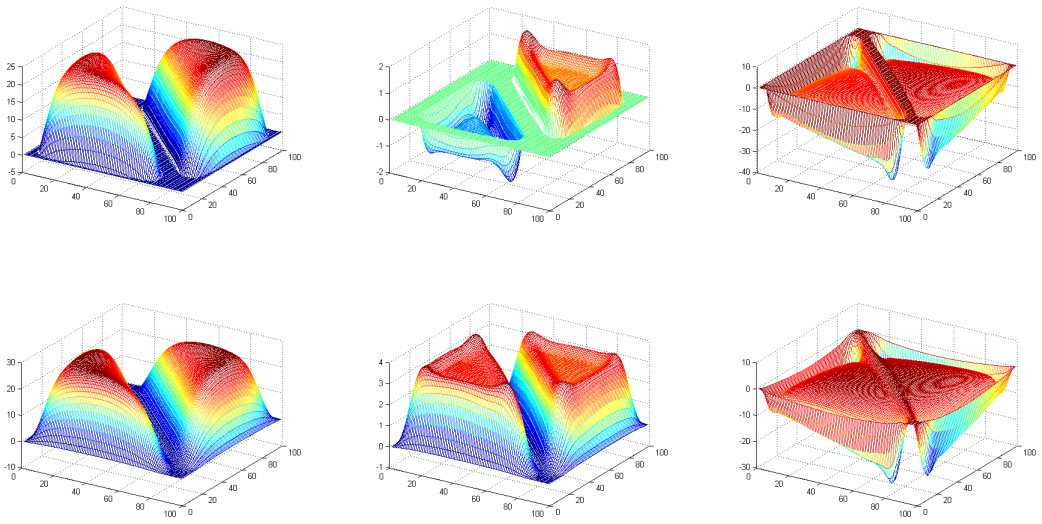

Figure 7.8: Optimal control u satisfying $\frac{|u_{n+1}-u_n|}{|u_n|} < 10^{-7}$ for $f(y) = |y|$ (top row) and $f(y) = -y^3$ (bottom row) for increasing fidelity $1, 10, 100$ (left to right).

As the fidelity weight increases, the number of iterates required for convergence increases, and we have to adjust α accordingly. The number of iterates required for convergence was 15, 25, 40, respectively. The sparsity of the control u is significantly observed in the figure.

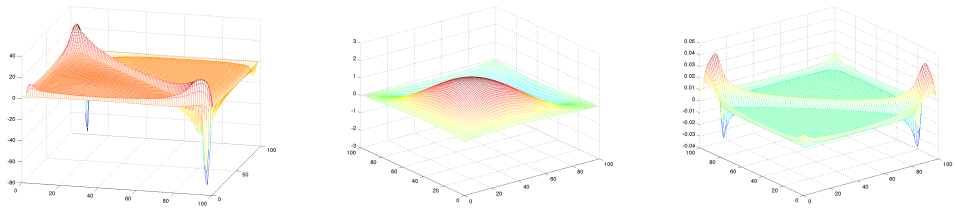

Figure 7.9: Optimal control u, target state y_d, and fidelity $y - y_d$, for $f(y) = -\exp(y)$.

For the case of $f(y) = -\exp(y)$ we terminated our iterates after 100 iterations, with monitoring index = .005. Note that the scale on the third plot in Figure 7.9 is very small. That is, the magnitude of the fidelity $y - y_d$ is less than .05 pointwise.

7.3 Moving Damping Actuator Control Problem

In this section we revisit the moving damping actuator problem introduced in Section 4.3.1. The following MATLAB file performs the SP method.

```
m=2*50; dx=1/m; x=0:dx:1; n=99; xx(n,1:100)=.3;
x0=[[1 .5 .25 zeros(1,46)]';[1 .25 zeros(1,47)]'];
xx=[[x0;.3] xx]; xx=.5*(xx(:,1:100)+xx(:,2:101));
h=pi^2*[1:49].^2; h=spdiags(h',0,49,49);
N=100; T=2; dt=T/N; J=zeros(99);
J(1:98,1:98)=.5*dt*[zeros(49) eye(49); -h zeros(49)];
e=ones(N,1); d=spdiags([-e e],-1:0,N,N); c=1; bt=1; %c=100; al=.5; bt=10;
JJ=kron(d,speye(n)); JA=0*JJ; u=c*ones(N,1); f=zeros(2*n*N,1);
b=[zeros(98,1);1]; BB=-c*dt*kron(speye(N),b*b');
e=ones(N,1); d=spdiags([e/4 e/2 e/4],-1:1,N,N); q=zeros(n);
q(1:98,1:98)=[h 0*h; 0*h bt*speye(49)]; QQ=-dt*kron(d,q);
for k=1:N; w=xx(n,k); b0=exp(-500*(x-w).^2)/5; bp=1000*(x-w).*b0;
ff=real(fft(b0,200));gg=real(fft(bp,200));
for i=1:49;for j=i:49;t=.5*(gg(j-i+1)-gg(j+i+1));
s=.5*(ff(j-i+1)-ff(j+i+1)); D(i,j)=s; D(j,i)=s; Dp(i,j)=t; Dp(j,i)=t;end;end,
x2=xx(50:98,k); J(50:98,50:98)=-.5*dt*D; J(50:98,99)=-.5*cc*dt*Dp*x2;
k1=[(k-1)*n+1:k*n]; JA(k1,k1)=J; if k>1; JA(k1,k1-n)=J; end;
f(k1)=cc*dt*[zeros(49,1);w*Dp*x2;0]; end; JJ=JJ-JA; H=[JJ -BB; -QQ -JJ'];
f(1:98)=f(1:98)+x0+.5*dt*[zeros(49) eye(49); -h -D]*x0; f(n)=.3;
```

```
y=H\f; xx=reshape(y(1:n*N),n,N); pp=reshape(y(n*N+[1:n*N]),n,N);
xx=[[x0;.3] xx]; xx=.5*(xx(:,1:100)+xx(:,2:101));
```

b0 is the damping distribution of actuators, and y contains the state and adjoint, i.e., $y = (x, \lambda)$.

7.4 Tests for Numerical Method for Controlled Navier-Stokes System

In this section we present our tests for the numerical method, Section 5.4, for the controlled Navier-Stokes system for the cavity flow. The following MATLAB file performs the implicit-explicit method (5.18)–(5.19) in Section 5.4.5.

```
%------------ Implicit-Explicit for Navier-Stkes systems
n=100; m=n-1; e=ones(m,1); h=spdiags([-e 2*e -e],-1:1,m,m);
h=kron(speye(m),h)+kron(h,speye(m)); h=n^2*h;
d1=spdiags([e e],-1:0,n,m)'; d2=spdiags([-e e],-1:0,n,m)';
b1=kron(d2,d1); b2=kron(d1,d2); b1=n*b1; b2=n*b2;
dx=1/n; [x y]=meshgrid([dx:dx:1-dx]);
f1=y.*sin(pi*x); f2=x.*sin(pi*y);
H=[h 0*h b1; 0*h h b2; b1' b2' -1.e-10*speye(n^2)];
H(1:m^2,1:m^2)=speye(m^2)+.005*dt*h;   %mu=.01; .005=mu/2
H(m^2+[1:m^2],m^2+[1:m^2])=speye(m^2)+.005*dt*h;
u=zeros(m); v=u; bu=u; bv=v; o1=u; o2=v; f=u; g=v;
uu=zeros(n+1); vv=uu;
xx=H\[u(:)-.005*dt*h*u(:)-dt*(1.5*bu(:)-.5*o1(:))+dt*(f1(:)-f(:));
v(:)-.005*dt*h*v(:)-dt*(1.5*bv(:)-.5*o2(:))+dt*(f2(:)-g(:));zeros(n^2,1)];
o1=bu; o2=bv;
u=reshape(xx(1:m^2),m,m); v=reshape(xx(m^2+[1:m^2]),m,m);
t=2:n;uu(t,t)=u; vv(t,t)=v;
u1=(uu(t,2:end)+uu(t,1:n))/2;v1=(vv(t,2:end)+vv(t,1:n))/2;
u2=(uu(2:end,t)+uu(1:n,t))/2;v2=(vv(2:end,t)+vv(1:n,t))/2;
t1=u1.^2;t2=v2.*u2;bu=t1(:,2:n)-t1(:,1:m)+t2(2:n,:)-t2(1:m,:);
t3=u1.*v1;t4=v2.^2;bv=t3(:,2:n)-t3(:,1:m)+t4(2:n,:)-t4(1:m,:);
bu=n*bu; bv=n*bv; mesh(v);

%----- 4th order convection
s=spdiags([-e e],[-1 1],m,m);
```

```
s1=kron(s,speye(m)); s2=kron(speye(m),s);
e=ones(m,1); ss=spdiags([e 4*e e]/6,-1:1,m,m);
s1=kron(s,ss); s2=kron(ss,s);
s1=n*s1/2; s2=n*s2/2;
u=reshape(xx(1:m^2),m,m); v=reshape(xx(m^2+[1:m^2]),m,m);
bu=s1*(u(:).^2)+s2*(u(:).*v(:));
bv=s1*(u(:).*v(:))+s2*(v(:).^2);
```

The first part is to set up the Stokes operator H for the square domain. The implicit-explicit method (Section 5.4.5) corresponds to the line

```
xx=H\[u(:)-.005*dt*h*u(:)-dt*(1.5*bu(:)-.5*o1(:))+dt*(f1(:)-f(:));
v(:)-.005*dt*h*v(:)-dt*(1.5*bv(:)-.5*o2(:))+dt*(f2(:)-g(:));zeros(n^2,1)];
```

$(f1, f2)$ is a body force and (f, g) is a potential force. We use $\Delta x = 1/100$, $\Delta t = .01$, $Re = 100$.

Chapter 8

Summary and Conclusion

The contribution of this dissertation is primarily the exposition of the theory behind the Sequential Programming (SP) method for solving constrained optimizations. The method itself is straightforward: to solve a constrained optimization problem, solve a sequence of linearized equality constraint optimizations and do a damped update for convergence. The convergence of the sequence of damped updates to an optimizer of the original problem is shown using the perturbation analysis of Alt and Robinson [2, 55], i.e., the necessary optimality condition for the original constrained optimization is a perturbation of the necessary optimality condition for the SP step. Thus, for small perturbations (often happens in applications) the SP method is guaranteed to converge. In fact, the relaxation parameter α for the damped update is chosen in $(0, 1]$, and a line search is not needed. Specific algorithms for SP are presented for various types of constraint optimizations. In general, the saddle point problem for the SP step lends itself to a fixed point iterate formulation. Thus, for convergence we require only that the relaxation parameter α be chosen such that the fixed point iteration is a contraction. A second order variant of Sequential Programming is described to reveal to the reader how the method can be extended and improved. Throughout, the Lagrange multiplier theory is a fundamental tool for the development and analysis of the SP method.

In Chapter 2 we introduced the constrained optimization problem in its general form (2.1) and in the control form (2.2). The control form is particulary important since many applications take this form. We derived the necessary optimality for constrained optimization problems by the Lagrange multiplier theory (Section 2.3). In general the necessary optimality system is a saddle point problem, and in the case of the optimal control problem it is the form of the two point boundary value problem (2.26).

In Chapter 3 we discussed methods for solving constrained optimizations and the saddle point problem. The focus was on the Sequential Programming (SP) method. We showed SP to be a middle ground between the gradient method and SQP (Sequential Quadratic Programming).

Most importantly, we established the convergence analysis for the SP method (Section 3.4.1) which strongly supports the use of SP over other methods like the gradient method and SQP.

In Chapter 4 we examined specific cases for SP and discussed how to treat different aspects of the SP method. We showed how to formulate an ill-posed problem as a variational problem and solve the resulting minimization by SP. We discussed the semilinear control problem (Section 4.3) which involves the linearization of a bilinear term and relates to the moving actuator control problem. We discussed non-smoothness that can appear in the cost or constraint and how to handle a subdifferential inclusion by the Yosida-Moreau approximation (Section 4.4).

In Chapter 5 we introduced the Navier-Stokes theory and outlined an optimal control problem for the controlled flow of an incompressible fluid on a step. We developed numerical methods for the evaluation of the gradient, divergence, convective, and diffusion terms that appear in the incompressible Navier-Stokes equations. We introduced the second order implicit-explicit time integration (Section 5.4.5) for the solution of the discretized problem for the controlled Navier-Stokes problem.

In Chapter 6 we discussed the time and space discretization of PDE constraint problems. We developed the high order discretized problem and introduced the higher order Runge-Kutta-Gauss time integration method (Section 6.1.2).

In Chapter 7 we discussed concrete examples for the implementation of the SP method. We introduced the Lorenz attractor system and provided an analysis of the Lorenz dynamics. We exhibited our implementation for the solution of the controlled Lorenz system which included the use of the Runge-Kutta-Gauss time integration introduced in Chapter 6. We performed a parameter study (Section 7.1.3) to show how different parameters affect the resulting solution by the SP method.

In conclusion, the Sequential Programming (SP) method is a very effective method for solving constrained optimizations. The convergence of the SP method is a hallmark of the method in that there is no line search in relaxation parameter $\alpha > 0$ for the damped update of iterations, we just pick $\alpha \in (0, 1]$. The method is guaranteed to converge to an optimal solution of the original constrained optimization (Section 3.4.1). It is a distinctive feature that SP does not rely directly on the Hessian information of the Lagrange functional and thus it can be used for non-smooth problems (Section 4.4) and in many problems of practical interest. It can avoid instabilities due to possible indefiniteness of the Hessian of the Lagrange functional during the iteration. As a specific example the fixed point problem for the control variables is developed for the optimal control problem as a reduced order method (Section 3.6.1) and we use the conjugate gradient method for a well-conditioned fixed point problem. The resulting SP method, with inner CG loop, is a very effective method for a large scale optimal control problem.

REFERENCES

[1] Alt, W. and K. Malanowski. "The Lagrange-Newton method for nonlinear optimal control problems." Comput. Optim. Appl., Vol. 2, 1993, pp. 77–100.

[2] Alt, W. "Lipschitzian perturbations in infinite optimization problems." In Mathematical Programming with Data Perturbations II, A.V. Fiacco, ed., Lecture Notes in Pure and Appl. Math., Vol. 85, Marcel Dekker, New York, 1983, pp. 7–21.

[3] Armijo, Larry. "Minimization of functions having Lipschitz continuous first partial derivatives." Pacific J. Math., Vol. 16, No. 1, 1966, pp. 1–3.

[4] Bartle, Robert G. The Elements of Integration and Lebesgue Measure. John Wiley & Sons, 2014.

[5] Barzilai, Jonathan and Jonathan M. Borwein. "Two-point step size gradient methods." IMA Journal of Numerical Analysis, Vol. 8, 1988, pp. 141–148.

[6] Bellout, H., J. Neustupa, and P. Penel. "On the Navier-Stokes equation with boundary conditions based on vorticity." Math. Nachr. 269-270, 2004, pp. 59–72.

[7] Bertsekas, D.P. Constrained Optimization and Lagrange Multiplier Methods. Academic Press, Paris, 1982.

[8] Biegler, L.T., O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, eds. Real-Time PDE-Constrained Optimization. SIAM, Philadelphia, 2007.

[9] Broyden, C.G. "A class of methods for solving nonlinear simultaneous equations." Mathematics of Computation, Vol. 19, No. 92, October 1965, pp. 577–593.

[10] Butcher, John C. Numerical Methods for Ordinary Differential Equations. John Wiley, 2003.

[11] Chorin, Alexandre J. and Jerrold E. Marsden. A Mathematical Introduction to Fluid Mechanics. Springer-Verlag, New York, 1993.

[12] Clarke, F.H. et al. Nonsmooth Analysis and Control Theory. Springer, New York, 1998.

[13] Concus, P., G.H. Golub, and G. Meurant. "Block preconditioning for the conjugate gradient method." SIAM J. Scl. Stat. Comput., Vol. 6, No. 1, January 1985.

[14] Dai, Yu-Hong and Roger Fletcher. "Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming." Numerical Analysis Report NA/215, September 2013.

[15] Ekeland, I. and R. Temam. Convex Analysis and Variational Problems. North-Holland, Amsterdam, 1976.

[16] Ekeland, I. and T. Turnbull. Infinite Dimensional Optimization and Convexity. The University of Chicago Press, Chicago, 1983.

[17] Fortin, M. and R. Glowinski. Augmented Lagrangian Methods: Applications to Numerical Solutions of Boundary Value Problems. North-Holland, Amsterdam, 1983.

[18] Frenkel, J. and T. Kontorova. "On the theory of plastic deformation and twinning." Izvestiya Akademii Nauk SSSR, Seriya Fizicheskaya 1, 1939, pp. 137–149.

[19] Fursikov, A.V., M.D. Gunzburger, and L.S. Hou. "Boundary value problems and optimal boundary control for the Navier-Stokes systems: The two-dimensional case." SIAM J. Control Optim., Vol. 36, 1998, pp. 852–894.

[20] Glowinski, R. Numerical Methods for Nonlinear Variational Problems. Springer-Verlag, Berlin, 1984.

[21] Glowinski, R., J.L. Lions, and R. Tremoliers. Numerical Analysis of Variational Inequalities. North-Holland, Amsterdam, 1981.

[22] Gunzburger, M.D. Perspectives of Flow Control and Optimization. SIAM, Philadelphia, 2003.

[23] Hale, Jack. Ordinary Differential Equations. Wiley, 1969.

[24] Hestenes, Magnus R. and Eduard Stiefel. "Methods of conjugate gradients for solving linear systems." Journal of Research of the National Bureau of Standards, Vol. 49, No. 6, December 1952.

[25] Hestenes, M.R. Optimization Theory: The Finite Dimensional Case. John Wiley and Sons, New York, 1975.

[26] Hestenes, M.R. "Multiplier and gradient methods." J. Optim. Theory Appl., Vol. 4, 1968, pp. 303–320.

[27] Hintermuller, M., K. Ito, and K. Kunisch. "The primal-dual active set strategy as a semismooth Newton method." SIAM J. Optim., Vol. 13, 2002, pp. 865–888.

[28] Ito, K. and F. Kappel. Evolution Equations and Approximations. World Scientific, River Edge, NJ, 2002.

[29] Ito, K. and K. Kunisch. Lagrange Multiplier Approach to Variational Problems and Applications. Advances in Design and Control, SIAM, Philadelphia, 2008.

[30] Ito, K. and K. Kunisch. "The augmented Lagrangian method for equality and inequality constraints in Hilbert space." Math. Programming, Vol. 46, 1990, pp. 341–360.

[31] Ito, K. and K. Kunisch. "An augmented Lagrangian technique for variational inequalities." Appl. Math. Optim., Vol. 21, 1990, pp. 223–241.

[32] Ito, K. and K. Kunisch. "Augmented Lagrangian-SQP-methods in Hilbert spaces and application to control in the coefficients problems." SIAM J. Optim., Vol. 6, 1996, pp. 96–125.

[33] Ito, K. and K. Kunisch. "Augmented Lagrangian-SQP methods for nonlinear optimal control problems of tracking type." SIAM J. Control Optim., Vol. 34, 1996, pp. 874–891.

[34] Ito, K. and K. Kunisch. "Augmented Lagrangian methods for nonsmooth convex optimization in Hilbert spaces." Nonlinear Anal., Vol. 41, 2000, pp. 573–589.

[35] Ito, K. and K. Kunisch. "An active set strategy based on the augmented Lagrangian formulation for image restoration." MZAN Math. Model Numer. Anal., Vol. 33, 1999, pp. 1–21.

[36] Ito, K. and K. Kunisch. "Optimal control of elliptic variational inequalities." Appl. Math. Optim., Vol. 41, 2000, pp. 343–364.

[37] Ito, K. and K. Kunisch. "Optimal control." In Encyclopedia of Electrical and Electronic Engineering, J. G. Webster, ed., 15, John Wiley and Sons, New York, 1999, pp. 364–379.

[38] Ito, K. and K. Kunisch. "Semi-smooth Newton methods for variational inequalities of the first kind." MZAN Math. Model. Numer. Anal., Vol. 37, 2003, pp. 41–62.

[39] Ito, K. and K. Kunisch. "The primal-dual active set method for nonlinear optimal control problems with bilateral constraints." SIAM J. Control Optim., Vol. 43, 2004, pp. 357–376.

[40] Ito, K. and K. Kunisch. "Semi-smooth Newton methods for state-constrained optimal control problems." Systems Control Lett., Vol. 50, 2003, pp. 221–228.

[41] Kanzow, C. "Inexact semismooth Newton methods for large-scale complementarity problems." Optim. Methods Softw., Vol. 19, 2004, pp. 309–325.

[42] Kelley, C.T. and E.W. Sachs. "Solution of optimal control problems by a pointwise projected Newton method." SIAM J. Control Optim., Vol. 33, 1995, pp. 1731–1757.

[43] Kreyszig, Erwin. Introductory Functional Analysis with Applications. Wiley, New York, 1978.

[44] Lions, J.L. Optimal Control of Systems Governed by Partial Differential Equations. Springer-Verlag, New York, 1971.

[45] Lorenz, E.N. "Deterministic nonperiodic flow." J. Atmos. Sci., Vol. 20, 1963, pp. 130–141.

[46] Luenberger, David G. Optimization by Vector Space Methods. John Wiley and Sons, 1969.

[47] Luenberger, David G. "The conjugate residual method for constrained minimization problems," SIAM J. Numer. Anal., Vol. 7, No. 3, September 1970.

[48] Maurer, H. and J. Zowe. "First and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems." Math. Programming, Vol. 16, 1979, pp. 98–110.

[49] Navier, C.L. "Mémoire sur les lois du mouvement des fluides." Mémoires Acad. Roy. Sci., Vol. 6, 1823, pp. 389–440.

[50] Neustupa, J. and P. Penel. "Incompressible viscous fluid flows and the generalized impermeability boundary conditions." IASME Transactions 7, Vol. 2, 2005, pp. 1254–1261.

[51] Nocedal, Jorge, and Stephen J. Wright. Numerical Optimization, Vol. 2. Springer, New York, 1999.

[52] Polak, E. and A.L. Tits. "A globally convergent implementable multiplier method with automatic limitation." Appl. Math. Optim., Vol. 6, 1980, pp. 335–360.

[53] Powell, M.J.D. "A method for nonlinear constraints in minimization problems." In Optimization, R. Fletcher, ed., Academic Press, New York, 1968, pp. 283–298.

[54] Robinson, S.M. "Regularity and stability for convex multivalued functions." Math. Oper. Res., Vol. 1, 1976, pp. 130–143.

[55] Robinson, S.M. "Strongly regular generalized equations." Math. of Oper. Res., Vol. 5, 1980, pp. 43–62.

[56] Rockafeller, R.T. "The multiplier method of Hestenes and Powell applied to convex programming," J. Optim. Theory Appl., Vol. 12, 1973, pp. 34–46.

[57] Saad, Yousef. Iterative Methods for Sparse Linear Systems. Siam, 2003.

[58] Sachs, E. "Broydens method in Hilbert space." Math. Programming, Vol. 35, 1986, pp. 71–82.

[59] Smale, S. "Mathematical Problems for the Next Century." Mathematics: Frontiers and Perspectives 2000, ed. by V. Arnold, M. Atiyah, P. Lax, and B. Mazur, Amer. Math. Soc., Providence, RI, 2000.

[60] Sontag, E.D. Mathematical Control Theory: Deterministic Finite Dimensional Systems. Springer-Verlag, New York, 1990.

[61] Sparrow, C. The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors. Springer-Verlag, New York, 1982.

[62] Stoer, J. and R. A. Tapia. The Local Convergence of Sequential Quadratic Programming Methods. Technical Report 87-04, Rice University, 1987.

[63] Temam, Roger. Navier-Stokes Equations: Theory and Numerical Analysis. Elsevier Science, New York, 1984.

[64] Temam, R. Infinite-Dimensional Dynamical Systems in Mechanics and Physics. Springer-Verlag, Berlin, 1988.

[65] Wolfe, P. "Convergence Conditions for Ascent Methods." SIAM Review, Vol. 11, No. 2, 1969, pp. 226–235.

[66] Yosida, K. Functional Analysis. Springer-Verlag, 1971.

[67] Zowe, J. and S. Kurcyusz. "Regularity and stability for the mathematical programming problem in Banach spaces." Appl. Math. Optim., Vol. 5, 1979, pp. 49–62.

APPENDIX

Appendix A

Analysis of the RKG Method

In this appendix we introduce the fundamental properties of the RKG method and conclude with a stability analysis of the method.

The first property of the RKG method is the collocation property. It follows from [23] that the Runge-Kutta-Gauss method satisfies the collocation property:

$$x^k - x^{k-1} = P(t_{k-1} + \Delta t) - P(t_{k-1}) = \Delta t \int_0^1 \sum_{i=1}^{s} L_i(s) f^{k,i}\, ds$$

$$x^{k,i} = P(t_{k-1} + c_i\, \Delta t),$$

i.e., $P(t_{k-1} + \sigma \Delta t)$ is a polynomial of degree s and satisfy the collocation property:

$$\frac{d}{dt} P(t_{k-1} + c_i\, \Delta t) = f^{k,i} = f(x^{k,i}, u^{k,i}) \tag{A.1}$$

for $1 \leq i \leq s$.

It is also known [23] that the Runge-Kutta-Gauss method is of $2s$ order. This fact follows from the collocation property and the fact that the Gauss-Legendre quadrature rule is exact for polynomials of degree up to $2s - 1$. In fact, from (A.1) there exists a polynomial p of degree s such that

$$\frac{d}{dt} P(t_{k-1} + t) = f(t_{k-1} + t, P(t_{k-1} + t)) + p(t)\, \zeta(t) + O(|t|^{2s+1}), \quad 0 \leq t \leq \Delta t,$$

where $\zeta(t) = \Pi_{i=1}^{s} (t - c_i\, \Delta t)$. It thus follows that

$$\frac{d}{dt}(P(t_{k-1} + t) - x(t_{k-1} + t)) = J(t)(P(t_{k-1} + t) - x(t_{k-1} + t)) + p(t)\, \zeta(t) + O(|t|^{2s+1})$$

where

$$J(t) = \int_0^1 f'(x(t_{k-1}+t) + \sigma\,(P(t_{k-1}+t) - x(t_{k-1}+t)))\,d\sigma.$$

Let $U(t,\sigma)$ be the fundamental matrix solution to $\dfrac{d}{dt}x(t) = J(t)x(t)$ and define $q(t) = U(t_k - t, t_{k-1})p(t),\ 0 \le t \le \Delta t$. If

$$q(t) = \sum_{i=0}^{s} q_i\, t^i + O(|t|^{s+1}), \quad 0 \le t \le \Delta t,$$

then

$$|P(t_k) - x(t_k)| = |U(t_k, t_{k-1})(P(t_{k-1}) - x(t_{k-1}))| + |\int_{t_{k-1}}^{t_k} U(t_k - s, t)p(s)\zeta(s)\,ds| + O(|\Delta t|^{2s+2})$$

$$= e^{\omega_1 \Delta t}\,|P(t_{k-1}) - x(t_{k-1})| + |\int_0^1 q_s\,\sigma^s \Pi_{i=1}^s(\sigma - c_i)\,d\sigma|\,|\Delta t|^{2s+1} + O(|\Delta t|^{2s+2}).$$

Here we used the fact that the Gauss-Legendre quadrature rule is exact for polynomials of degree up to $2s-1$. Therefore the Runge-Kutta-Gauss method is of $2s$ order.

Next, we present the stability analysis. The stability analysis of Runge-Kutta methods can be found in [29, 66] for the dissipative system, i.e., $\omega = 0$. We give a self-contained stability analysis for the control dynamics (6.3) and we relate our result to those in [29, 66]. Let $\bar{x} = \frac{x^k + x^{k-1}}{2}$. We start with the case of $s = 2$, i.e.,

$$x^k - x^{k-1} = \Delta t\,(f(\bar{x} - \alpha\, f^{k,2}, u_1) + f(\tilde{x} + \alpha\, f^{k,1}, u_2)),$$

where $\alpha = \frac{\sqrt{3}}{6}\,\Delta t$, and thus by taking the inner product of this with $\bar{x}$

$$\frac{1}{2}\,(|x^k|^2 - |x^{k-1}|^2) = \frac{\Delta t}{2}\,[(f(x^{k,1}, u^{k,1}), x^{k,1}) + \alpha\,(f^{k,1}, f^{k,2}) + (f(x^{k,2}, u^{k,2}), x^{k,2}) - \alpha\,(f^{k,1}, f^{k,2})]$$

$$= \frac{\Delta t}{2}\,[(f(x^{k,1}, u^{k,1}), x^{k,1}) + (f(x^{k,2}, u^{k,2}), x^{k,2})], \tag{A.2}$$

where we used the fact that

$$\frac{x^k + x^{k-1}}{2} = \frac{x^{k,1} + x^{k,2}}{2} + \frac{\alpha}{2}(f^{k,2} - f^{k,1}).$$

Now we are interested in generalizing (A.2) for the general case of s.

There are several steps required to analyze the stability of the RKG method. We have a number of lemmas to develop the results.

Lemma A.1 *For $s \geq 1$*

$$\sum_{i=1}^{s} w_i \, (f(x^{k,i}, u^{k,i}), \frac{x^k + x^{k-1}}{2}) = \sum_{i=1}^{s} w_i \, (f(x^{k,i}, u^{k,i}), x^{k,i}), \tag{A.3}$$

and

$$\frac{1}{2} \, (|x^k|^2 - |x^{k-1}|^2) = \Delta t \sum_{i=1}^{s} w_i \, (f(x^{k,i}, u^{k,i}), x^{k,i}).$$

Proof: Let $\bar{x} = \frac{x^k + x^{k-1}}{2}$. It follows from (A.1) and the fact that the Gauss-Legendre quadrature rule is exact for polynomials of degree up to $2s-1$ that

$$\Delta t \sum_{i=1}^{s} w_i \, (f^{k,i}, x^{k,i} - \bar{x}) = \Delta t \int_0^1 (P'(t_{k-1} + \sigma \Delta t), P(t_{k-1} + \sigma \Delta t) - \bar{x}) \, d\sigma$$

$$= \frac{1}{2} \, (|P(t_{k-1} + \Delta t)|^2 - |P(t_{k-1})|^2) - (P(t_{k-1} + \Delta t) - P(t_{k-1}), \bar{x}) = 0,$$

since $x_k = P(t_{k-1} + \Delta t)$ and $x_{k-1} = P(t_{k-1})$. Hence

$$\sum_{i=1}^{s} w_i \, (f(x^{k,i}), \bar{x}) = \sum_{i=1}^{s} w_i \, [(f^{k,i}, x^{k,i}) - (f^{k,i}, x^{k,i} - \bar{x})]$$

$$= \sum_{i=1}^{s} w_i \, (f^{k,i}, x^{k,i}). \square$$

From (A.3) we have the following lemma.

Lemma A.2 *Let $A = (a_{i,j}) \in R^{s \times s}$ be defined by* (6.8)*. Then*

$$C = W(A - \frac{1}{2} \, e w^t) \;\; \textit{is skew-symmetric} \tag{A.4}$$

where W is the diagonal matrix with diagonal $(w_1, w_2, \ldots, w_s)$ and $e = col \, (1, 1, \ldots, 1)$.

Proof: We consider the scalar equation $\frac{d}{dt} x(t) = f(t)$ with $\Delta t = 1$ and let $f_i = f(c_i), \; 1 \leq i \leq s$. Then (6.7) is equivalent to

$$y = x \, e + A f \quad \text{and} \quad \hat{x} = x + w \cdot f.$$

It thus follows from (A.3) that for $f \in R^s$ and $x \in R$

$$(W(Af + x \, e), f)_{R^s} = (Wf, \bar{x})_{R^s} = (w \cdot f) \, (\frac{1}{2} \, (w \cdot f) + x).$$

Thus $(WAf, f) = \frac{1}{2}\,|w \cdot f|^2$ for all $f \in R^s$. $\square$

Lemma A.2 implies the algebraic stability [23], i.e., the matrix B defined by

$$B = WA + A^tW - ww^t$$

is nonnegative. In fact from Lemma A.2 we have $B = C + C^t = 0$. It is shown that the algebraic stability implies that the method is BN-stable, i.e., for any stepsize $\Delta t > 0$ the method generates contractive numerical solutions, provided that all $c_i,\ 1 \le i \le s$ are distinct (nonconfluent).

From Lemma A.1 and (6.3) we have

$$\frac{1}{2}\,(|x^k|^2 - |x^{k-1}|^2) = \Delta t \sum_{i=1}^{s} w_i\,(\omega\,|x^{k,i}|^2 + c_2\,|u^{k,i}|^2).$$

So, we establish the estimate for $\sum_{i=1}^{s} w_i\,|x^{k,i}|^2$ in what follows. The collocation property (A.1) implies that A^{-1} exists and

$$(A^{-1}x)_i = \frac{dQ}{dt}(c_i) \tag{A.5}$$

where Q is the polynomial of degree s satisfying $Q(0) = 0$ and $Q(c_i) = x_i,\ 1 \le i \le s$. In fact we have

$$\begin{aligned} (A^{-1})_{i,j} &= \frac{c_i}{c_j}\frac{1}{c_j - c_i}\Pi_{\ell \ne i,j}\,\frac{c_i - c_\ell}{c_j - c_\ell} \quad \text{for } \ i \ne j \\ (A^{-1})_{i,i} &= \frac{1}{c_i} + \sum_{\ell \ne i}\frac{1}{c_i - c_\ell} \quad \text{for } \ 1 \le i \le s. \end{aligned} \tag{A.6}$$

From (6.7) $y = \mathrm{col}(x^{k,1}, \ldots, x^{k,s})$ satisfies

$$A^{-1}(y - \mathrm{col}(x^{k-1}, \ldots, x^{k-1})) = \Delta t\,\mathrm{col}(f(x^{k,1}, u^{k,1}), \ldots, f(x^{k,s}, u^{k,s})). \tag{A.7}$$

We show that (A.7) has a unique solution. To this end we need the following lemma on the property of matrix A.

Lemma A.3 *Let P be the diagonal matrix with diagonal $P_{i,i} = \sqrt{(1 - c_i)c_i^{-1}},\ 1 \le i \le s$. Then we have*

$$PA^{-1}W^{-1}P^{-1} = \Lambda + S$$

where Λ is the positive diagonal matrix with diagonal $\Lambda_{i,i} = \dfrac{1}{2}\,\dfrac{1}{c_i(1 - c_i)w_i},\ 1 \le i \le s$, and S is skew-symmetric.

Proof: From (A.5)

$$(PA^{-1}W^{-1}P^{-1})_{i,j} = (\frac{d}{dt}Q_j)(c_i)(p_jw_j)^{-1}Q_i(c_i)p_i$$
$$= \sum_{\ell=1}^{s} (\frac{d}{dt}\tilde{Q}_j)(c_\ell)(p_\ell)^2 w_\ell \tilde{Q}_i(c_\ell) \tag{A.8}$$

where $\tilde{Q}_i = (p_iw_i)^{-1}\, Q_i$ and Q_i is the polynomial of degree s satisfying $Q(0) = 0$ and $Q_i(c_j) = \delta_{i,j}$. We set

$$p_i = \sqrt{(1-c_i)c_i^{-1}}, \quad 1 \le i \le s.$$

Since

$$(1-t)\frac{d}{dt}Q_j(t)\frac{Q_i(t)}{t} \text{ is a polynomial of degree } 2s-1$$

and

$$\frac{d}{dt}(Q_j(t)\frac{Q_i(t)}{t}) = \frac{d}{dt}Q_j(t)\frac{Q_i(t)}{t} + \frac{d}{dt}Q_i(t)\frac{Q_j(t)}{t} - \frac{Q_i(t)}{t}\frac{Q_j(t)}{t},$$

thus it follows from (A.8)

$$(PA^{-1}W^{-1}P^{-1})_{i,j} + (PA^{-1}W^{-1}P^{-1})_{j,i} = \int_0^1 (1-t)\,(\frac{d}{dt}\tilde{Q}_j(t)\frac{\tilde{Q}_i(t)}{t} + \frac{d}{dt}\tilde{Q}_i(t)\frac{\tilde{Q}_j(t)}{t})\,dt$$

$$= \int_0^1 (\tilde{Q}_i(t)\frac{\tilde{Q}_j(t)}{t} + (1-t)\frac{\tilde{Q}_i(t)}{t}\frac{\tilde{Q}_j(t)}{t}\,dt = 0$$

if $i \neq j$ and

$$(PA^{-1}W^{-1}P^{-1})_{i,i} = \frac{1}{2}\int_0^1 (\tilde{Q}_i(t)\frac{\tilde{Q}_i(t)}{t} + (1-t)\frac{\tilde{Q}_i(t)}{t}\frac{\tilde{Q}_i(t)}{t}\,dt = \frac{1}{2}\,\frac{1}{c_i(1-c_i)w_i}.\square$$

Now we state the stability theorem.

Theorem A.1 *Assume that* (6.4)–(6.5) *hold. Then system* (A.7) *has a unique solution* $y = col\,(x^{k,1}, \ldots, x^{k,s}s) \in R^{sn}$ *and we have*

$$\sum_{i=1}^{s} w_i\,|x^{k,i}|^2 \le c_3\,|x^{k-1}|^2 + c_4\Delta t \sum_{i=1}^{s} w_i\,|u^{k,i}|^2 \tag{A.9}$$

for some $c_3,\ c_4 \ge 0$ *and* $\Delta t > 0$ *sufficiently small.*

Proof: Define $\psi = col\,(\psi_1, \ldots, \psi_s) \in R^{sn}$ by $\psi_i = p_iw_i\,x^{k,i},\ 1 \le i \le s$. Then (A.7) is equiva-

lently written as

$$\begin{aligned} &PA^{-1}(W^{-1}P^{-1}\psi - \mathrm{col}(x^{k-1},\ldots,x^{k-1})) \\ &\quad = \Delta t\,\mathrm{col}(p_1\,f((p_1w_1)^{-1}\,\psi_1,u^{k,1}),\ldots,p_s\,f((p_sw_s)^{-1}\,\psi_s,u^{k,s})) \end{aligned} \tag{A.10}$$

where P is the diagonal matrix as defined in Lemma A.3. Define the function $\Phi : R^{ns} \to R^{ns}$ by

$$\Phi(\psi) = PA^{-1}W^{-1}P^{-1}\psi - \Delta t\,\mathrm{col}(p_1\,f((p_1w_1)^{-1}\,\psi_1,u^{k,1}),\ldots,p_s\,f((p_sw_s)^{-1}\,\psi_s,u^{k,s})).$$

Then by (6.5) and Lemma A.3 we have

$$(\Phi(\psi^{(1)}) - \Phi(\psi^{(2)}),\psi^{(1)} - \psi^{(2)}) \geq \sum_{i=1}^{s}(\frac{1}{2c_i(1-c_i)} - \omega\,\Delta t)\frac{1}{w_i}\,|\psi_i^{(1)} - \psi_i^{(2)}|^2$$

provided that $(W^{-1}P^{-1}\Psi^{(1)})_i,\ (W^{-1}P^{-1}\Psi^{(2)})_i \in B_R$ for $1 \leq i \leq s$. For $\Delta t\,\omega_R < \min_i \frac{1}{2c_i(1-c_i)}$, Φ is dissipative and thus the solution to (A.7) is unique and continuously depends on $\mathrm{col}(u^{k,1},\ldots,u^{k,s})$. Moreover by taking the inner product of (A.10) with ψ it follows from (6.5) that

$$\sum_{i=1}^{s}(\frac{1}{2c_i(1-c_i)}\,|x^{s,i}|^2 - \alpha_i\,(x^{s,i},x^{k-1}))\,w_ip_i^2 \leq \Delta t\sum_{i=1}^{s}(\omega\,|x^{s,i}|^2 + |u^{s,i}|^2)\,w_ip_i^2 \tag{A.11}$$

where $\alpha = A^{-1}e$. Thus for $\Delta t > 0$ sufficiently small there exists $c_3,\ c_4$ such that (A.9) holds. □

We state the following convergence theorem.

Theorem A.2 *Assume that* (6.4)–(6.5) *hold,* Ψ *is locally Lipschitz continuous, and* $\omega\Delta t < \min_i \frac{1}{2c_i(1-c_i)}$. *Then for* $u_N = \{col(u^{k,1},\ldots,u^{k,s})\}_{k=1}^N$ *with* $\sum_{k=1}^N\sum_{i=1}^s w_i\,h(u^{k,i})\,\Delta t$ *being uniformly bounded in* N *there exists a unique* $x_N = \{col(x^{k,1},\ldots,x^{k,s},x^k)\}_{k=1}^N$ *in* B_R *for some* $R > 0$, *satisfying the constraint* (6.7). *Also,* x_N *continuously depends on* u_N. *For each* N *there exists an optimal solution* u_N *to* (6.6)–(6.7) *and associated primal and adjoint states* $x_N = \{col(x^{k,1},\ldots,x^{k,s},x^k)\}_{k=1}^N$, $p_N = \{col(p^{k,1},\ldots,p^{k,s},p^{k-1})\}_{k=1}^N$ *such that* (6.10) *holds. Let* $\tilde{u}_N$ *denote the piecewise polynomial defined by*

$$\tilde{u}_N(t_{k-1} + \sigma\,\Delta t) = \sum_{i=1}^{s} u^{k,i}\,L_i(\sigma) \tag{A.12}$$

on (t_{k-1}, t_k), $1 \le k \le N$ *and* $\tilde{x}_N$ *and* $\tilde{p}_N$ *be the piecewise polynomial functions defined by*

$$\begin{aligned} \tilde{x}_N(t_{k-1} + \sigma\,\Delta t) &= x^{k-1} + \Delta t \int_0^\sigma \sum_{i=1}^s f^{k,i}\, L_i(s)\, ds \\ \tilde{p}_N(t_k - \sigma\,\Delta t) &= p^k + \Delta t \int_0^\sigma \sum_{i=1}^s g^{k,s+1-i}\, L_i(s)\, ds. \end{aligned} \tag{A.13}$$

Then the sequence $(\tilde{x}_N, \tilde{u}_N, \tilde{p}_N)$ *in* $H^1(0,T;R^n) \times L^2(0,T;R^m) \times H^1(0,T;R^n)$ *has a convergent subsequence as* $\Delta t \to 0$ *and for every cluster point* (u,x,p), $u \in U$ *is an optimal control of* (6.2)–(6.3) *and* (u,x,p) *satisfies the necessary optimality condition* (6.10)*.*

Proof: It follows from (6.5), (A.3) and Lemma A.4 that there exist constants $c_5,\ c_6 \ge 0$ (independent of $0 < \Delta t \le \bar{h}$) such that

$$|x^k|^2 \le (1 + c_5 \Delta t)|x^{k-1}|^2 + c_6 \Delta t \sum_{i=1}^s w_i\, |u^{k,i}|^2. \tag{A.14}$$

If U is bounded, then by assumption $c_2 = 0$ and thus $c_6 = 0$. If U is unbounded by assumption (6.5), $\sum_{k=1}^N \sum_{i=1}^s w_i\, |u^{k,i}|^2\, \Delta t$ is bounded uniformly in N if the corresponding cost satisfies $J^N(u_N) \le J^N(0) \le \bar{J}$. Hence $|x^k|,\ |x^{k,i}| \le R,\ 1 \le i \le s$ for $1 \le k \le N$ and some $R > 0$. Note that the adjoint equation for $(p^{k,i}, p^{k-1})$ is equivalently written as

$$p^{k-1} = p^k + \Delta t \sum_{i=1}^s w_i\, [f_x(x^{k,i}, u^{k,i})^t p^{k,i} + \ell_x(x^{k,i})]$$

$$PA^{-1}(W^{-1}P^{-1}\xi - \mathrm{col}(p^k, \ldots, p^k))$$

$$= \Delta t\, P\, \mathrm{col}(f_x(x^{k,s}, u^{k,s})^t \xi_1 + \ell_x(x^{k,s}), \ldots, f_x(x^{k,1}, u^{k,1})^t \xi_s + \ell_x(x^{k,1}))$$

where $\xi_i = (p_i w_i)^{-1}\, p^{k,(s+1)-i},\ 1 \le i \le s$. Since (6.4) implies that $(f_x(x,u)p, p) \le \omega\, |p|^2$ for $x \in B_R$ and $u \in U$, it follows from the above arguments leading to (A.3) and Lemma A.4 that

$$|p^{k-1}|^2 \le (1 + c_5 \Delta t)|p^k|^2 + c_6 \Delta t \sum_{i=1}^s w_i\, |\ell_x(x^{k,i})|^2$$

and thus $|p^{k-1}|,\ |p^{k,(s+1)-i}|,\ 1 \le i \le s$ are uniformly bounded in $1 \le k \le N$. Since Ψ is locally Lipschitz continuous, it follows that $|u^{k,i}|,\ 1 \le i \le s$ are uniformly bounded in $1 \le k \le N$.

Since

$$\frac{d}{dt}\tilde{x}_N(t_{k-1}+c_i\,\Delta t) = f^{k,i} = f(x^{k,i},u^{k,i})$$

$$-\frac{d}{dt}\tilde{p}_N(t_k - c_i\,\Delta t) = g^{k,i} = f_x(x^{k,i},u^{k,i})^t p^{k,i} + \ell_x(x^{k,i},u^{k,i}),$$

$$|\frac{d\tilde{x}_N}{dt}(t)| + |\frac{d\tilde{p}_N}{dt}(t)| \le M, \quad \text{a.e. in } (0,T).$$

Using Lipschitz continuity of Ψ again, we find that $|\frac{d\tilde{u}_N}{dt}(t)|_\infty$ is uniformly bounded as well. By the compactness of Lipschitz continuous sequences in $L^2(0,T)$ there exists a subsequence $\hat{N}$ such that $(\tilde{x}_{\hat{N}},\tilde{u}_{\hat{N}},\tilde{p}_{\hat{N}})$ converges to (u^*,x^*,p^*) in $L^2(0,T;R^n\times R^m\times R^n)$ and pointwise a.e. in $(0,T)$. From (6.10) and (A.1)

$$\tilde{x}^N(t) = x_0 + \int_0^t f^N(s)\,ds$$

where

$$f^N(t_{k-1}+\sigma\,\Delta t) = \Delta t \int_0^\sigma \sum_{i=1}^s f^{k,i}\,L_i(s)\,ds$$

on (t_{k-1},t_k). By Lebesgue's dominated convergence theorem we find that x^* coincides with the solution $x(t;u^*)$ to (6.3) associated with u^*. For $v\in L^2(0,T;R^m)$ let v_N be the piecewise constant approximation of v defined by $v^{k,i} = \frac{1}{N}\int_{t_{k-1}}^{t_k} v(t)\,dt$, $1\le i\le s$, $1\le k\le N$. Then

$$J^N(u_N) \le J^N(v_N) \quad \text{for all admissible controls } v$$

and $x(t;v_N)\to x(t;v)$ as $N\to\infty$. Thus by the Lebesgue dominated convergence theorem $J^{0,x_0}(u^*)\le J^{0,x_0}(v)$ for all admissible controls, i.e., (u^*,x^*) is an optimal pair. It is not difficult to argue that the triple (u^*,x^*,p^*) satisfies the necessary optimality (6.10). □

The following theorem establishes the convergence rate.

Theorem A.3 Suppose that the sequence $(\tilde{u}^N(t),\tilde{x}^N(t),\tilde{p}^N(t))$ defined by (A.12)–(A.13) converges to the triple $(u(t),x(t),p(t))$ that satisfies TPBV problem (2.26) as $N\to\infty$. Assume that $(x(t),p(t))\in C^{2s+1}(0,T:R^{2n})$. Then there exists a constant M (independent of N) such that

$$|(\tilde{x}^N(t)-x(t),\tilde{p}^N(t)-p(t))| \le M\,|\Delta t|^{2s}, \quad t\in[0,T].$$

Proof: It follows from the collocation property (A.1) and (6.10) that

$$\frac{d}{dt}\tilde{x}^N(t_{k-1}+c_i\,\Delta t) = f^{k,i} = f(x^{k,i},u^{k,i})$$

$$-\frac{d}{dt}\tilde{p}^N(t_{k-1}+c_i\,\Delta t) = g^{k,i} = f_x(x^{k,i},u^{k,i})^t p^{k,i} + \ell_x(x^{k,i})$$

$$\tilde{u}^N(t_{k-1}+c_i\,\Delta t) = u^{k,i} = \Psi(x^{k,i},p^{k,i}), \quad 1\le k\le N,\ \ 1\le i\le s$$

with $\tilde{x}^N(0) = x_0$ and $\tilde{p}^N(T) = g_x(\tilde{x}^N(T))$. Thus if $\zeta(t) = \Pi_{i=1}^s\,(t-c_i\,\Delta t)$, then there exist polynomials $p = \{\mathrm{col}(p_k^{(1)},p_k^{(2)})\}_{k=1}^N$ of degree s such that

$$\frac{d}{dt}\tilde{x}_N(t_{k-1}+t) = f(\tilde{x}_N(t_{k-1}+t),\tilde{u}_N(t_{k-1}+t)) + p_k^{(1)}(t)\,\zeta(t) + O(|t|^{2s+1}),$$

$$-\frac{d}{dt}\tilde{p}^N(t_{k-1}+t) = f_x(\tilde{x}_N(t_{k-1}+t),\tilde{u}_N(t_{k-1}+t))^t\tilde{p}_N(t_{k-1}+t))$$

$$+\ell_x(\tilde{x}_N(t_{k-1}+t)) + p_k^{(2)}(t)\,\zeta(t) + O(|t|^{2s+1})$$

$$\tilde{u}_N(t_{k-1}+t) = \Psi(\tilde{x}_N(t_{k-1}+t),\tilde{p}_N(t_{k-1}+t)),$$

for $0\le t\le\Delta t$. For the simplicity of our discussion we assume that $f(x,u) = f(x)+Bu$ and $h(u) = \frac{\beta}{2}|u|^2$. Thus if $e_1(t) = \tilde{x}_N(t_{k-1}+t) - x(t_{k-1}+t)$ and $e_2(t) = \tilde{p}_N(t_{k-1}+t) - p(t_{k-1}+t)$, then

$$\frac{d}{dt}\begin{pmatrix} e_1(t)\\ e_2(t)\end{pmatrix} = H_k(t)\begin{pmatrix} e_1(t)\\ e_2(t)\end{pmatrix} + \begin{pmatrix} p_k^{(1)}(t)\\ p_k^{(2)}(t)\end{pmatrix}\zeta(t) + O(|t|^{2s+1}) \tag{A.15}$$

for $1\le k\le N$ and $0<t<\Delta t$, and $e_2(T) = Ge_1(T)$. Here

$$G = \int_0^1 g'(x(T)+\sigma\,(\tilde{x}_N(T)-x(T))\,d\sigma$$

and

$$H_k(t) = \begin{pmatrix} A_k(t) & -\frac{1}{\beta}BB^t\\ -Q_k(t) & -A_k(t)^t\end{pmatrix}$$

with

$$A_k(t) = \int_0^1 f'(x(t_{k-1}+t)+\sigma\,(\tilde{x}_N(t_{k-1}+t)-x(t_{k-1}+t)))\,d\sigma$$

and

$$Q_k(t) = \int_0^1 \ell'(x(t_{k-1}+t) + \sigma\left(\tilde{x}_N(t_{k-1}+t) - x(t_{k-1}+t)\right))\, d\sigma.$$

If we define the matrix functions $A(t)$, $Q(t)$ by $A(t) = A_k(t)$ and $Q(t) = Q_k(t)$ on $(t_{k-1}, t_k]$, by the Riccati transformation we have $e_2(t) = P(t)e_1(t) + r(t)$ where the symmetric matrix $P(t) \in R^{n\times n}$ satisfies

$$\frac{d}{dt}P(t) + A^t(t)P(t) + P(t)A(t) - \frac{1}{\beta}\,P(t)BB^tP(t) + Q(t) = 0, \quad P(T) = G,$$

and $r(t) \in R^n$ satisfies $r(T) = 0$ and

$$\frac{d}{dt}r(t_{k-1}+t) + (A_k(t) - \frac{1}{\beta}\,BB^tP(t))^t r(t) + (P(t_{k-1}+t)p_k^{(1)}(t) + p_k^{(2)}(t))\,\zeta(t) + O(|\Delta t|^{2s+1})$$

for $1 \le k \le N$ and $0 < t < \Delta t$. As we argued in Section 2.2, using the fact that the Gauss-Legendre quadrature rule is exact for polynomials of degree up to $2s-1$, there exist constants ω_1 and M_1 such that

$$|r(t_{k-1})| \le (1 + \omega_1\Delta t)|r(t_k)| + M_1\,|\Delta t|^{2s+1}.$$

Hence $|r(0)| \le \tilde{M}\,|\Delta t|^{2s}$ for some constant $\tilde{M}$ (independent of N). Since $e_1(0) = 0$ we have $|e_2(0)| = |r(0)|$. Since (A.15) is a system of ODEs it thus follows from the same arguments applied to (A.15) that $|e_1(t), e_2(t))| \le M\,|\Delta t|^{2s}$ for some constant M (independent of N and Δt). $\square$