OPTIMAL GEOMETRY IN A SIMPLE MODEL OF TWO-DIMENSIONAL HEAT TRANSFER

OPTIMAL GEOMETRY IN A SIMPLE MODEL OF TWO-DIMENSIONAL HEAT TRANSFER

By

XIAOHUI PENG, B.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2011) McMaster University

(Mathematics) Hamilton, Ontario

TITLE:	OPTIMAL GEOMETRY IN A SIMPLE MODEL OF TWO DIMENSIONAL HEAT TRANSFER
AUTHOR:	Xiaohui Peng, B. Sc. (Beijing University of Posts and Telecommunications)
SUPERVISOR:	Dr. Bartosz Protas
NUMBER OF PAGES:	VIII, 101

To my dearest parents

Acknowledgements

Foremost, I would like to express my sincere and deep gratitude to my supervisor Prof. Bartosz Protas for his continuous support of my Master study. Without his patient guidance and enthusiasm I cannot accomplish this Thesis. His conscientiousness and working habit also have lifetime value to me. My sincere gratitude also goes to my master defence committee members: Prof. David Lozinski and Prof. Nicholas Kevlahan, for their valuable suggestion and comments on my work. In addition, I am grateful to all collaborators from the General Motors of Canada R&D Centre in Oshawa for constructive feedbacks and helpful discussions.

I would like to express my gratitude to AUTO21 who provides me with funding through the grant "Multidisciplinary Optimization of Hybrid and Electric Vehicle Batteries" (ED401-EHE). I am also indebted to Department of Mathematics and Statistics, McMaster University, for financial assistance throughout my stay as a master candidate.

My warm and sincere thanks also go to my colleagues: Diego Ayala, Vladislav Bukshtynov, Ramesh Yapalparvi, Christopher Cappadocia, Youzhou Zhou and Jason Haradyn, for all the assistance in research, MATLAB coding and modifying the thesis.

Last but not the least, I would like to thank my parents: Jun Peng and Xia Liu, for raising me up and supporting me both spiritually and economically with selfless heart.

Abstract

This investigation is motivated by the problem of optimal design of cooling elements in modern battery systems used in hybrid/electric vehicles. We consider a simple model of two-dimensional steady-state heat conduction generated by a prescribed distribution of heat sources and involving a one-dimensional cooling element represented by a closed contour. The problem consists in finding an optimal shape of the cooling element which will ensure that the temperature in a given region is close (in the least squares sense) to some prescribed distribution. We formulate this problem as PDE-constrained optimization and use methods of the shape-differential calculus to obtain the first-order optimality conditions characterizing the locally optimal shapes of the contour. These optimal shapes are then found numerically using the conjugate gradient method where the shape gradients are conveniently computed based on adjoint equations. A number of computational aspects of the proposed approach is discussed and optimization results obtained in several test problems are presented.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Mathematical Modelling of Heat Transfer 3

1.3 Earlier Works on Shape Optimization in Heat Transfer 5

1.4 Outline of the Thesis 6

2 Formulation of the Problem **7**

2.1 Mathematical Model 7

2.2 Optimization Problem 9

3 Shape Optimization Approach **12**

3.1 Sensitivity Analysis in Shape Optimization 12

3.1.1 Material and Shape Derivative 13

3.1.2 Some Useful Formulas 15

3.2 Optimality Conditions 16

3.2.1 Optimality Conditions without Length Constraint 16

3.2.2 Optimality Conditions in the Presence of Length Constraint . . 18
3.3 Gradient Descent Method 19
3.3.1 PDEs Defined on Variable Domains 19
3.3.2 Perturbation PDE System 21
3.3.3 Adjoint System 30
3.3.4 Preconditioning the Gradient 38

4 Numerical Implementation **39**

4.1 Spectral Methods for Periodic and Non-periodic Domains 40
4.1.1 Spectral Differentiation for Non-periodic Domain 41
4.1.2 Clenshaw-Curtis Quadrature 49
4.1.3 Spectral Differentiation for Periodic Domains 50
4.2 Boundary Integral Equation 53
4.3 Numerical Implementation 55
4.3.1 Discretizing Equation (4.48a) 58
4.3.2 Discretizing Equation (4.48b) 59
4.3.3 Discretization of Boundary Condition (4.48c) 61
4.3.4 Normal Derivatives on $\mathcal{C}$ and Curvature 63
4.3.5 Sobolev Gradients and Evaluation of Cost Functional 64
4.4 Optimization Algorithm 65
4.4.1 Optimal Step σ_m 65
4.4.2 Optimization Algorithm 66

5 Results **69**

5.1 Validation of Solvers for the Direct Problem ($\mathcal{D}$) and Adjoint Problem ($\mathcal{A}$) 69

5.2 Validation of the Cost Functional Gradient 73

5.3 Optimization Results 80

6 Conclusion and Future Work **97**

List of Figures

1.1 Battery pack (courtesy of GM) 2

1.2 Cooling system in automobile battery elements 3

2.1 Sketch of domain 7

4.1 Tensor product grid (N=8) [16] 44

4.2 Tensor product grid (N=2) 45

4.3 Flow-chart of optimization algorithm 68

5.1 Solution μ, u_p, u_h and u from PDE system ($\mathcal{D}$) obtained using different resolutions indicated on insets. 71

5.2 μ^*, u_p^*, u_h^* and u^* from PDE system ($\mathcal{A}$) obtained using different resolutions indicated on insets. 72

5.3 Five different initial contours applied in the κ-test. 75

5.4 κ-test 1, $\log_{10}|\kappa - 1|$ for $\boldsymbol{\nabla}\mathcal{J}$ with 4 different perturbations. 76

5.5 κ-test 2, $\log_{10}|\kappa - 1|$ for 4 different initial contours. 77

5.6 κ-test 3, $\log_{10}|\kappa - 1|$ for $\boldsymbol{\nabla}\mathcal{J}_L$ with 4 different perturbations. 78

5.7 κ-test 4, $\log_{10}|\kappa - 1|$ for $\boldsymbol{\nabla}\mathcal{J}_L$ with 4 different initial contours. 78

5.8 κ-test 5 for subdomain A; ζ_1 and $\mathcal{C}_0$ is applied 79

5.9 Contour of heat sources q_1, q_2 and q_3 81

5.10 Contour of desired temperatures $\overline{u}_1$, $\overline{u}_2$, $\overline{u}_3$ and $\overline{u}_4$ 82

5.11 Initial contours applied for different cases 82

5.12 Optimization results for CASE 1 when L^2 shape gradient is applied; Heat sources is q_1 and desired temperature is $\overline{u}_1$. 86

5.13 Optimization results for CASE 2 when q_2 and $\overline{u}_2$ are applied. 87

5.14 Optimization results for CASE 3 when q_1 and $\overline{u}_3$ are applied. 88

5.15 Optimization results for CASE 4 when q_1 and $\overline{u}_4$ are applied. 89

5.16 Optimization results for CASE 5 when q_3 [20] and $\overline{u}_4$ are applied. . . . 90

5.17 Optimization results for CASE 6 when subdomain of interest is $[-\frac{1}{2}, 1]\times$ $[-\frac{1}{2}, 1]$; q_1 and $\overline{u}_4$ are applied. 91

5.18 Optimization results for CASE 7 when different initial contours are applied; heat sources is q_1 and desired temperature is $\overline{u}_4$. 92

5.19 Initial and final temperatures for CASE 7 when $\mathcal{C}_3^{(0)}$, $\mathcal{C}_4^{(0)}$ and $\mathcal{C}_5^{(0)}$ are applied (the grid corresponds to cell pattern of $\overline{u}_4$). 93

5.20 Optimization results for CASE 8 when different α's are applied; heat sources is q_3 and desired temperature is $\overline{u}_4$. 94

5.21 Final temperatures for $\alpha = 1$, $\alpha = 10$, $\alpha = 10^2$, $\alpha = 10^2$ in CASE 8 (the grid corresponds to cell pattern of $\overline{u}_4$). 95

5.22 Optimization results for CASE 9 when subdomain of interest is $[-\frac{1}{2}, 1]\times$ $[-\frac{1}{2}, 1]$ and $\alpha = 100$; heat sources is q_1 and desired temperature is $\overline{u}_4$. . 96

List of Tables

5.1 Settings for different κ-tests 74

5.2 Settings for different optimization cases 80

Chapter 1

Introduction

1.1 Motivation

A hybrid electric vehicle (HEV) is an automobile propelled by both a combustion engine and an electric engine using the energy stored in battery packs. Compared with conventional combustion automobiles, HEVs can reduce green gas emissions, leading to a decrease in urban air pollution and global warming. Rising oil prices is another factor that makes HEVs more favourable over the conventional combustion vehicles which rely on oil consumption. Moreover, government subsidies and market demands also drive the development of HEVs.

The electric energy for HEVs is provided by vehicle batteries. Figure 1.1 shows the battery pack used in today's HEVs. Due to the low energy density, a certain number of battery cells are lined up in a space-efficient way to construct the battery pack. Between the cells there are cooling plates, usually consisting of coils with coolant liquid, to maintain the temperature of the batteries in a satisfactory range.

It is important to enhance the performance of vehicle batteries since it plays a vital role in the performance of HEV. There are several factors that will affect the

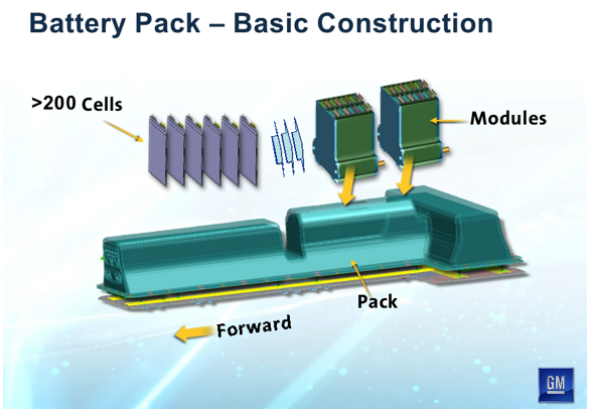


Figure 1.1: Battery pack (courtesy of GM)

battery's performance, such as state of charge, age and condition. One of the most important factors is the temperature, which will increase due to heat accumulation when the battery is in operation. High temperatures will cause unreliable performance, functional failure, decreasing reliability and corrosion which will reduce the battery's life. In addition, temperature variation will lead to inhomogeneous discharge and degradation, so that optimal use is prevented. Since the battery pack consists of batteries, coolant and cooling plate, it can result in high weight and large volume. Moreover, due to the complexity of the entire battery system, its efficiency can depend on several factors. Thus optimization is absolutely needed to finalize each component of battery pack in order to improve its efficiency, reduce wastes and decrease both volume and weight.

In this work we will focus on how to design the cooling element to maintain the battery pack at a suitable temperature with low variation. We start with the cooling element. Figure 1.2 shows a concept for a liquid coolant element design in today's HEV's battery system. The liquid enters one end with low temperature and exits the other end with high temperature; thus a certain amount of heat is absorbed. The question is: how can we design the circuit so that for some fixed length we can have a

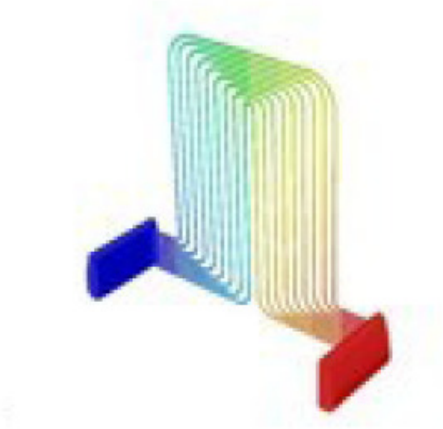


Figure 1.2: Cooling system in automobile battery elements

desirable heat extraction? As we know, the shape of the circuit can depend on many factors, like the distribution of heat source, the material of cooling pipe line,etc. In this dissertation we focus on the fundamental effect of the geometry of the cooling units on the efficiency of the heat transfer processes in the battery.This investigation is part of a boarder research initiative related to optimization of battery systems [1].

1.2 Mathematical Modelling of Heat Transfer

What we are dealing with now is a typical heat transfer problem, so it would be helpful for us to first get an idea of how we model the heat transfer processes mathematically. Heat is one form of energy transferred from a region with high temperature to that of low temperature. The driving force of this energy transfer is the temperature difference. Given constant material properties, the distribution of internal energy can be mapped to temperature.

There are 3 forms of heat transfer: conduction, convection and radiation. On

the microscopic scale, heat conduction is the transformation of kinetic energy from molecules to their neighbours, due to collisions in molecules. Heat convection is the heat transfer due to the movement of molecules. Radiation refers to the energy transformation as electromagnetic waves. Examples include sunlight.

Conduction can be modelled mathematically by the so-called heat equation. For example we have a two-dimensional (2D) domain $\Omega \in \mathbb{R}^2$ with a prescribed heat sources $q(\mathbf{x})$ defined on Ω. On the boundary Ω we have a heat flow $f(\mathbf{x})$ per unit time per unit surface area. The heat conduction process can be modelled as a parabolic PDE system with boundary conditions:

$$\frac{\partial u}{\partial t} - k\,\Delta\, u = q \qquad \text{in}\,\Omega, \tag{1.1a}$$

$$\frac{\partial u}{\partial n} = f \qquad \text{on}\,\partial\,\Omega, \tag{1.1b}$$

where u is a function of $\mathbf{x}$ and t and t is the time variable, k is the heat conduction coefficient and usually it is sufficient to assume $k = 1$. The derivation of (1.1a), which is called the heat equation, can be done using *Fourier law of heat conduction* with a detailed description in [2]. The term $\frac{\partial u}{\partial n}$ can be viewed as the heat flux defined as the amount of thermal energy flowing across some surface per unit time per unit surface area [2].

If $f = 0$, in the other words $\frac{\partial u}{\partial n} = 0$, then we have no heat flux across the boundary which means physically: Ω is isolated. As time goes we will arrive at some stationary state when u does not depend on time: $\frac{\partial u}{\partial t} = 0$. Then, we have for (1.1a):

$$-k\,\Delta\, u = q$$

which is called the steady heat equation and its solution u is the stationary solution.

1.3 Earlier Works on Shape Optimization in Heat Transfer

It is common to solve a heat transfer problem with some proscribed geometry of boundary and measurement in the domain. For example, in (1.1) the heat sources q, shape of boundary $\partial\,\Omega$ and Neumann data f are given and the temperature u is left to be determined. These kind of problems are usually called analysis problems [3] or direct problems. It would be of great interest to reverse the formulation of the problem: specify the desired performance and compute the configuration of the boundary or conditions on the boundary such that the desired performance can be achieved or the resulting performance is as close as possible to the desired one [3]. These are called "inverse problems".

If the undetermined property is the shape, we call these kind of inverse problems surface shape design (SSD) problems [3] which are also treated in this dissertation as shape optimization, since the inverse problem and the optimization problems share a lot in common. In [4] the basic concepts and techniques of inverse and optimization problems in heat transfer was discussed. Several works which dealt with the shape optimization problem are as follows: in [5] Lan, Cheng and Wu treat the shape design for heat conduction problems by combining curvilinear grid generation, conjugate gradient and redistribution method, in [3] an extension of element-based finite volume method was applied to surface shape design of thermofluids problems, Huang and Chao [6] compared two boundary element method (BEM)–based inverse algorithms: Levenberg-Marquardt method and conjugate gradient method (CGM) in solving an inverse geometry heat conduction problem to determine the irregular boundary configurations, also Huang and Hsiung [7] solved the inverse geometry problem using CGM and BEM-based inverse algorithm. The same strategy is applied in both [8] which deals with an inverse heat problem in three dimensions for two interfacial con-

figurations and [9] which dealt with a shape identification problem for two interfacial configurations in multiple domains.

1.4 Outline of the Thesis

In Chapter 2 we propose a simplified mathematical model of two dimensional steady heat conduction, its assumptions and statement of the optimization problem. Next a brief introduction to shape calculus is discussed, based on which we could set up our optimality conditions and the mathematical derivation of shape gradient in Chapter 3. Then, in Chapter 4 a detailed numerical implementation combining spectral methods and boundary integral techniques is discussed and a validation of our numerical scheme was also conducted. Optimization results are presented in Chapter 5 and, finally, conclusions are given in Chapter 6.

Chapter 2

Formulation of the Problem

2.1 Mathematical Model

In order to get a basic understanding of our problem we propose a simplified 2D model with only conduction involved at present, see Figure 2.1.

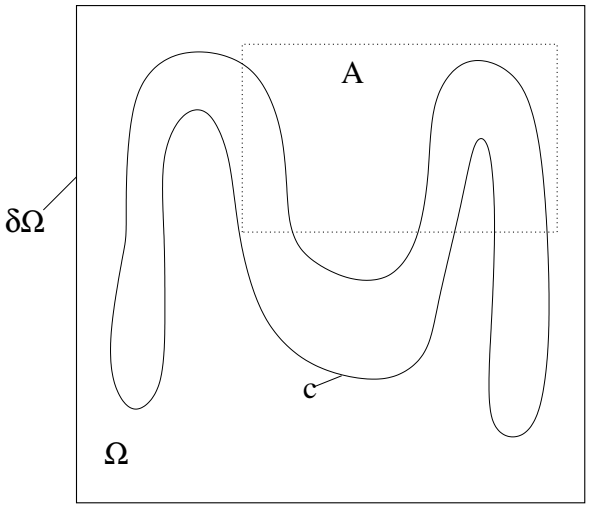


Figure 2.1: Sketch of domain

We make the following assumptions:

- We treat the battery pack as a 2D square region $\Omega \in \mathbb{R}^2$ with boundary $\partial\,\Omega$ and prescribed heat sources q inside Ω, and denote the temperature on Ω as a time-independent function $u : \Omega \to \mathbb{R}$ which means we are only interested in the steady state solution.

- The cooling element is simplified to a smooth closed curve $\mathcal{C}$ whose total length is L and has a constant temperature u_0. By "simplified" we mean $\mathcal{C}$ here only models the channel without coolant liquid, thus, no heat convection will be considered. Moreover, we denote our optimal shape as $\tilde{\mathcal{C}}$.

- The battery is isolated from surroundings, in other words, no heat flux across $\partial\Omega$.

- A is the domain where we want to maintain a prescribed temperature $\bar{u}$.

The question is how we can design the shape of $\mathcal{C}$ with a length L so that we can minimize the difference in the least square sense between u and $\bar{u}$ in A. Denoting Ω_1 as the domain inside $\mathcal{C}$ and Ω_2 as $\Omega \setminus \Omega_1$, we obtain the following mathematical model of the problem:

$$-k\,\Delta\,u_1 = q \qquad \text{in } \Omega_1, \tag{$\mathcal{D}\,1$}$$

$$-k\Delta\,u_2 = q \qquad \text{in } \Omega_2, \tag{$\mathcal{D}\,2$}$$

$$u_2 = u_1 \qquad \text{on } \mathcal{C}, \tag{$\mathcal{D}\,3$}$$

$$k\,(\frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n}) = \gamma\,(u_1 - u_0) \quad \text{on } \mathcal{C}, \tag{$\mathcal{D}\,4$}$$

$$k\,\frac{\partial u_2}{\partial n} = 0 \qquad \text{on } \partial\Omega \tag{$\mathcal{D}\,5$}$$

where

$$u_1 = u\ \chi_{\Omega_1}, \tag{2.2a}$$

$$u_2 = u\ \chi_{\Omega_2}. \tag{2.2b}$$

Here are some comments: $(\mathcal{D}\,1)$ and $(\mathcal{D}\,2)$ are heat equations in steady-state for u_1 and u_2. Relation $(\mathcal{D}\,3)$ represents the continuity of temperature u across $\mathcal{C}$. Relation $(\mathcal{D}\,4)$ is based on Newton's law of cooling which states that the net heat flux across the cooling channel is proportional to the temperature difference between the ambient temperature $u|_{\mathcal{C}}$ and the reference temperature u_0 which can be thought of as representing the temperature of cooling liquid. Relation $(\mathcal{D}\,5)$ states that the heat flux over $\partial\Omega$ is zero which means Ω is isolated. $\mathbf{n}$ is the unit outer normal, whereas k is the heat conduction coefficient and γ is some known constant.

2.2 Optimization Problem

Mathematically speaking, optimization is a process of maximizing or minimizing a function subject to some constraint on its variables [10]. The optimization model usually contains three parts: control variables, constraint on variables and objective function. Here our control variable is the shape of smooth closed curve $\mathcal{C}$. Mathematically let us denote our control set as:

$$\Gamma \triangleq \{\, \mathcal{C} \,|\, \mathcal{C} \textit{ is sufficiently smooth closed curve}\} \tag{2.3}$$

Our constraints are the PDE system $(\mathcal{D})$ together with length constraint:

$$\oint_{\mathcal{C}} ds = L \tag{2.4}$$

the objective function here can be formulated as a cost functional $\mathcal{J}$ depending on $\mathcal{C}$ which minimize the overall difference between u, the temperature corresponding to the given curve $\mathcal{C}$, and $\bar{u}$, the prescribed temperature in a least square sense in domain A:

$$\mathcal{J}(\mathcal{C}, u) \triangleq \frac{1}{2}\int_A (u - \overline{u})^2 \, d\,\mathbf{x}. \tag{2.5}$$

Notice that u in $\mathcal{D}$ depends on $\mathcal{C}$ and thus our $\mathcal{J}$ ultimately is a function of $\mathcal{C}$. Now we can state our optimization problem as:

$$\boxed{\begin{array}{ll} \min\limits_{\mathcal{C}} \mathcal{J}(\mathcal{C}) \triangleq & \dfrac{1}{2}\displaystyle\int_A (u - \overline{u})^2 \, d\,x \\ \text{subject to:} & \text{System } (\mathcal{D}) \\ & \displaystyle\oint_{\mathcal{C}} ds = L, \end{array}} \tag{2.6}$$

Here are the general steps to solve the optimization problem (2.6):

1. Formulate the optimality condition which characterizes the optimal shape $\tilde{\mathcal{C}}$ by differentiating (2.5) with respect to (w. r. t.) the control variable $\mathcal{C}$. This can be done using the tools of shape calculus.

2. Derive an expression for the shape gradient $\boldsymbol{\nabla}\mathcal{J}$ in a suitable control space by the help of adjoint PDE system $(\mathcal{A})$. The adjoint system is obtained by shape differentiation of $(\mathcal{D})$ w. r. t. $\mathcal{C}$ and some manipulations of the adjoint boundary conditions in order to express the shape derivative of (2.5) in Riesz form. Preconditioning for $\boldsymbol{\nabla}\mathcal{J}$ will be performed.

3. Setup a numerical approach to solve PDE system $(\mathcal{D})$ together with the adjoint PDE system $(\mathcal{A})$. In our paper the numerical approach to the two PDE system is accomplished by spectral method and Boundary Integral Equations techniques.

4. Implement a conjugate gradient approach to find the optimal shape $\tilde{\mathcal{C}}$. The

outline of algorithm is as follows:

Setup $\mathcal{C}^{(0)}$ *as the initial cooling coil;* $\quad m \leftarrow 0$

while the optimality condition on $\mathcal{C}^{(m)}$ *is not satisfied*

do

obtain $\boldsymbol{\nabla}\, \mathcal{J}^{(m)}$ *by solving direct PDE system* ($\mathcal{D}$) *and adjoint system* ($\mathcal{A}$)

find the conjugate gradient $\boldsymbol{\nabla}_c\, \mathcal{J}^{(m)}$ *based on* $\boldsymbol{\nabla}\, \mathcal{J}^{(m)}$

determine step length σ_m *via line minimization*

update the shape of the contour $\quad \mathcal{C}^{(m+1)} = \mathcal{C}^{(m)} - \sigma_m\, \boldsymbol{\nabla}_c \mathcal{J}^{(m)}$

$m \leftarrow m + 1$

As it is, an important task is to find an expression for $\boldsymbol{\nabla}\, \mathcal{J}$ since it give us the optimal direction for changing $\mathcal{C}$. We will address this topic with the idea of shape differentiation in the next chapter.

Chapter 3

Shape Optimization Approach

Our shape gradient $\boldsymbol{\nabla}\mathcal{J}$ can be obtained by the help of a suitably defined adjoint problem. The formulation of the adjoint PDE system ($\mathcal{A}$) corresponding to the direct one ($\mathcal{D}$) relies on the shape calculus. In this chapter, we will provide the basic information about shape differentiation including all the necessary mathematical background required in this thesis. Then the optimality condition is set up. Finally, a detailed derivation of the adjoint system is presented together with the expression of $\boldsymbol{\nabla}\mathcal{J}$. Let us first start with the idea of shape optimization.

3.1 Sensitivity Analysis in Shape Optimization

In order to perform optimization, the derivative of $\mathcal{J}(\mathcal{C})$ w. r. t. shape of $\mathcal{C}$ is clearly needed to be studied. Thus the theoretical results of sensitivity analysis are introduced in this section. For practical purpose, we assume that all the data below including functions, curves, etc, are smooth enough.

Suppose we have a simply connected domain $\Lambda \subset \mathbb{R}^d$, for some $d \in \mathbb{Z}^+$(we can take $d = 2$ in this work). The shape of Λ is changing over time. For instance, at time

0 we will have Λ_0 and at time τ we will have a new domain Λ_τ. There are lots of real world examples with the shape changing over time, such as the deformation of the clouds, water waves, falling droplets, etc. One common question might be: how can we describe these shape changes mathematically? We introduce a function F_τ mapping Λ to a new domain $\Lambda_\tau \in \mathbb{R}^d$.

$$F_\tau : \qquad \Lambda \longrightarrow \mathbb{R}.$$

More specifically, for some value of τ, $\forall\, \mathbf{x} \in \Lambda$, $\mathbf{x}_\tau = F_\tau(\mathbf{x})$ is the new position under the mapping F_τ in Λ_τ. Thus we can represent Λ_τ by $F_\tau(\Lambda)$. Can we find an explicit expression for F_τ? The answer is "yes". By introducing a velocity field $\boldsymbol{\mathcal{V}}$ defined as

$$\boldsymbol{\mathcal{V}} : \qquad \mathbb{R}^d \longrightarrow \mathbb{R}^d \tag{3.1}$$

the explicit expression of F_τ can be given as:

$$F_\tau(x) = I \cdot \mathbf{x} + \tau\, \boldsymbol{\mathcal{V}}; \qquad \forall\, \mathbf{x} \in \Lambda \tag{3.2}$$

where I is a $d \times d$ identity matrix.

Based on this setting we can introduce the idea of shape derivative.

3.1.1 Material and Shape Derivative

Consider the following model PDE problem:

$$\begin{cases} L(\psi) & = f_\tau \qquad \mathbf{x} \in \Lambda_\tau, \\ L_b(\psi) & = g_\tau \qquad \mathbf{x} \in \partial\Lambda_\tau \end{cases} \tag{$\mathcal{K}$}$$

where L and L_b are operators. For example, L can be a Laplacian operator Δ and L_b can be $\frac{\partial}{\partial n}$ representing the Neumann boundary condition. Model system ($\mathcal{K}$) can be also regarded as an abstract version of our governing system ($\mathcal{D}$). We have $\Lambda_\tau = F_\tau(\Lambda)$ in which F_τ is given by (3.2), $\partial\Lambda_\tau$ is the boundary of Λ (noticing that the boundary changes along with the deformation of Λ). f_τ and g_τ are applied here in order to indicate their dependence on the shape of Λ_τ. The consequence of the deformation of Λ_τ is that our solution ψ now is a function depending on Λ_τ,or on τ ultimately. We can denote this τ-dependent solution by $\psi : \mathbb{R}^+ \times \mathbb{R}^d \longrightarrow \mathbb{R}$.

$$\psi \triangleq \psi(\tau, \mathbf{x}_\tau); \quad \mathbf{x}_\tau = \mathbf{x} + \tau\, \boldsymbol{\mathcal{V}}. \tag{3.3}$$

The *material derivative* of ψ is defined as its total derivative $\frac{d\psi}{d\tau}\Big|_{\tau=0}$ given by:

$$\dot{\psi} = \frac{\partial \psi}{\partial \tau}\bigg|_{\tau=0} + \nabla_{\mathbf{x}}\psi \cdot \boldsymbol{\mathcal{V}} \tag{3.4}$$

which can be obtained via chain rule and the fact:

$$\mathbf{x}_\tau = \mathbf{x} + \tau\, \boldsymbol{\mathcal{V}}. \tag{3.5}$$

The term $\frac{\partial\psi}{\partial\tau}\Big|_{\tau=0}$ is called *shape derivative* of ψ which will be denoted by ψ' in the following and under some smoothness condition the shape derivative and spatial derivative commute with each other:

$$(\nabla_{\mathbf{x}}\, \psi)' = \nabla_{\mathbf{x}}\, \psi'. \tag{3.6}$$

3.1.2 Some Useful Formulas

In the following section we will be constantly dealing with the differentiation of shape-dependent functions in integral form, thus we provide here some fundamental results of shape calculus.

We want to find the material derivative of $\mathcal{F}$ and $\mathcal{G}$ defined as:

$$\begin{aligned} \mathcal{F} &= \int_{\Lambda_\tau} \psi(\tau, \mathbf{x}_\tau)\, d\mathbf{x}_\tau, \\ \mathcal{G} &= \int_{\partial\Lambda_\tau} \psi(\tau, \mathbf{x}_\tau)\, d\mathbf{s}_\tau, \end{aligned} \tag{3.7}$$

where Λ_τ, $\partial\Lambda_\tau, \psi$ and $\mathbf{x}_\tau$ are defined in Section 3.1.1.

Lemma 1. *The material derivative of $\mathcal{F}$ and $\mathcal{G}$ can be expressed as follows:*

$$\dot{\mathcal{F}} = \int_\Lambda \psi'\, d\mathbf{x} + \int_\Lambda \psi\, div\boldsymbol{\mathcal{V}}\, d\mathbf{x}, \tag{3.8a}$$

$$\dot{\mathcal{F}} = \int_\Lambda \psi'\, d\mathbf{x} + \int_{\partial\Lambda} \psi\boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, d\mathbf{s}, \tag{3.8b}$$

$$\dot{\mathcal{G}} = \int_{\partial\Lambda} \psi'\, d\mathbf{s} + \int_{\partial\Lambda} \left(\frac{\partial\psi}{\partial n} + \kappa\,\psi\right) \boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, d\mathbf{s}, \tag{3.8c}$$

where κ is the signed curvature of $\partial\Lambda$; $\dot{\psi} := \dot{\psi}(0, \mathbf{x})$ and $\psi' := \psi'(0, \mathbf{x})$ are the material and shape derivative respectively.

The detailed proof can be found in [11] and [12]. With the definition of f above we can differentiate a functional on a changing volume [11]:

Lemma 2. *Suppose a smooth function F:*

$$F : \mathbb{R}^+ \times \mathbb{R} \longrightarrow \mathbb{R}$$

is given and consider the functional:

$$\mathcal{E} := \int_{\Lambda_\tau} F(\tau, \psi(\tau, \mathbf{x}_\tau))\, d\mathbf{x}_\tau,$$

where Λ_τ, $\mathbf{x}_\tau$ and ψ are defined as above. Its material derivative can be given by

$$\begin{aligned} \dot{\mathcal{E}}(\Lambda, \boldsymbol{\mathcal{V}}) &= \left.\frac{d\mathcal{E}}{d\,\tau}\right|_{\tau=0} \\ &= \int_\Lambda \left[\frac{\partial F(0,\psi)}{\partial \tau} + \frac{\partial F(0,\psi)}{\partial \psi} \cdot \psi'\right] d\ \mathbf{x} + \int_{\partial\,\Lambda} F(0,\psi)\boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, d\mathbf{s} \end{aligned} \tag{3.9}$$

(ψ' is defined to be the same as the above Lemma 1).
If F does not depend on τ explicitly, we have:

$$\dot{\mathcal{E}}(\Lambda, \boldsymbol{\mathcal{V}}) = \int_\Lambda \frac{dF}{df} f'\ d\ \mathbf{x} + \int_{\partial\Lambda} F\boldsymbol{\mathcal{V}} \cdot\ \mathbf{n}\, d\mathbf{s}. \tag{3.10}$$

The proof can be obtained by applying chain rule and *Lemma* 1.

Till now the basic idea of shape optimization including all the Lemmas, properties and concepts is sufficient for us to proceed to the formulation of optimal conditions.

3.2 Optimality Conditions

3.2.1 Optimality Conditions without Length Constraint

Consider first the optimization problem without any length constraint on $\mathcal{C}$. For completeness purpose we restate our optimization goal: we want to find the shape of $\tilde{\mathcal{C}}$ such that $\mathcal{J}(\mathcal{C})$ is minimized:

$$\min_{\mathcal{C}} \mathcal{J}(\mathcal{C}) \triangleq \frac{1}{2}\int_A (u - \overline{u})^2\, d\mathbf{x}.$$

The optimality condition states:

$$\boxed{\forall\, \zeta,\ \dot{\mathcal{J}}(\tilde{\mathcal{C}}, \zeta) = 0} \tag{3.11}$$

ζ can be viewed as an arbitrary normal perturbation applied to the contour. The reason why we only focus on the perturbation in normal direction is that any tangential perturbation has no effect on the shape of $\mathcal{C}$, thus the functional $\mathcal{J}$ is only affected by the normal perturbation of $\mathcal{C}$. $\dot{\mathcal{J}}(\mathcal{C}; \zeta)$ is the material derivative at $\tau = 0$ (see section (3.1.1)). This ζ can also be treated as an arbitrary direction in the functional space Γ(section 2.2). In fact, by asserting our optimality condition in this way, we are looking for a local minimum. The global minimum is very hard to investigate due to the non-convexity of our functional $\mathcal{J}$ on $\mathcal{C}$. A equivalent statement of (3.11) is that the shape gradient $\boldsymbol{\nabla}\mathcal{J} = 0$ in some control space. The proof is straight forward. Applying (3.10) in *Lemma* 2 we have:

$$\dot{\mathcal{J}}(\mathcal{C}; \zeta) = \int_A (u - u_0) u' \, d\mathbf{x}$$

where u' is the shape derivative of u. Notice that we don't have the boundary term because the domain of interest is A with a fixed boundary. By Riesz representation theorem [13], we can represent the right-hand side term of above equation as:

$$\dot{\mathcal{J}}(\mathcal{C}; \zeta) = \langle \boldsymbol{\nabla}\mathcal{J}, \zeta \rangle_W \tag{3.12}$$

in which $\langle \centerdot, \centerdot \rangle_W$ is the inner product in some control space W. Considering that ζ is arbitrary we have $\dot{\mathcal{J}}(\mathcal{C}; \zeta) = 0$ if and only if $\boldsymbol{\nabla}\mathcal{J} = 0$. In the later section, we will work in L^2 control space and derive an expression for $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ by the help of adjoint analysis.

3.2.2 Optimality Conditions in the Presence of Length Constraint

Following the idea of multi-objective optimization, we can incorporate our length constraint (2.4) in our optimization problem with a scalar weight. Consider a functional:

$$\mathcal{J}_L(\mathcal{C}) := \mathcal{J}(\mathcal{C}) + \frac{\alpha}{2}\left(\int_{\mathcal{C}} d\mathbf{s} - L\right)^2, \tag{3.13}$$

where α is an arbitrary number controlling the weight of length constraint. By relation (3.8c) from *Lemma* 1 its shape derivative is given by:

$$\dot{\mathcal{J}}_L(\mathcal{C}) = \dot{\mathcal{J}}(\mathcal{C}) + \alpha\left(\int_{\mathcal{C}} d\,\mathbf{s} - L\right)\int_{\mathcal{C}} \kappa\,\zeta\, d\mathbf{s}.$$

We notice that the second term in the right-hand side is already in L^2 Riesz form. If our control space is L^2 then we have:

$$\dot{\mathcal{J}}_L(\mathcal{C}) = \left\langle \boldsymbol{\nabla}^{L^2}\mathcal{J}, \zeta\right\rangle_{L^2} + \left\langle \alpha\left(\int_{\mathcal{C}} d\mathbf{s} - L\right)\kappa, \zeta\right\rangle_{L^2}, \tag{3.14}$$

combining with (3.12) we can extract the L^2 shape gradient of $\mathcal{J}_L$ as:

$$\boldsymbol{\nabla}\mathcal{J}_L := \boldsymbol{\nabla}\mathcal{J} + \alpha\left(\int_{\mathcal{C}} d\mathbf{s} - L\right)\kappa \tag{3.15}$$

and the optimality condition after adding the length constraint thus becomes

$$\boxed{\forall\,\zeta,\ \dot{\mathcal{J}}_L(\tilde{\mathcal{C}}, \zeta) = 0} \tag{3.16}$$

3.3 Gradient Descent Method

In this section we will talk about how we conveniently compute the shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$. As we known, different cooling elements $\mathcal{C}$ will lead to different temperature distributions u, thus the solution of the governing PDE system $(\mathcal{D})$ depends on the shape of the cooling elements $\mathcal{C}$. In the framework of shape calculus, the change of $\mathcal{C}$ can be parametrized by a "velocity" field pushing $\mathcal{C}$ in some direction. Thus we can formulate the change of $\mathcal{C}$ and then the resulting change of our PDE system $(\mathcal{D})$ mathematically by the help of "velocity". Actually, this leads to the so called Perturbation PDE System $(\mathcal{P})$ for the shape derivative u'. With $(\mathcal{P})$ we will be able to find the adjoint system $(\mathcal{A})$ for the adjoint variable u^* which in turn will be used to obtain a convenient expression for the L^2 cost functional gradient.

3.3.1 PDEs Defined on Variable Domains

Suppose a "velocity" field $\boldsymbol{Z}$ on Ω defined as:

$$\boldsymbol{Z} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2;\ \boldsymbol{Z}(\partial\,\Omega) = 0, \tag{3.17}$$

then we can define our deformation function $T_\tau : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ as:

$$T_\tau = I + \tau\,\boldsymbol{Z}; \quad \tau \in \mathbb{R}^+, \tag{3.18}$$

where I is the 2 by 2 identity matrix and τ now serves as a small parameter: $1 \gg \tau \geqslant 0$. The inverse of T_τ is

$$T_\tau^{-1} = I - \tau\,\boldsymbol{Z}; \quad \tau \in \mathbb{R}^+. \tag{3.19}$$

We will use this property frequently in the following sections. The change of Ω_1 and Ω_2 can be expressed mathematically using this deformation function T_τ. Let $\Omega_{1\tau}$ and $\Omega_{2\tau}$ be the domain of Ω_1 and Ω_2 at time τ respectively:

$$\begin{aligned} \Omega_{1\tau} &:= T_\tau(\Omega_1), \\ \Omega_{2\tau} &:= T_\tau(\Omega_2). \end{aligned} \tag{3.20}$$

We also can apply this idea to the change of our contour $\mathcal{C}$:

$$\mathcal{C}_\tau := T_\tau(\mathcal{C}),$$

but considering the numerical implementation later on and the nature of our problem, we only take into account the normal perturbation of $\boldsymbol{Z}$. The reason is under the smoothness condition of $\mathcal{C}$, tangential perturbation will not change the shape of contour $\mathcal{C}$. Our $\mathcal{C}_\tau$ is defined as:

$$\mathbf{x}_{\mathcal{C}}(\tau) = \mathbf{x} + \tau\,\zeta, \quad \mathbf{x}_{\mathcal{C}}(\tau) \in \mathcal{C}_\tau \text{ and } \mathbf{x} \in \mathcal{C} \tag{3.21}$$

in which $\zeta := \boldsymbol{Z} \cdot \mathbf{n}$.

Given the change of our domain Ω parametrized by τ, the solution u of PDE system $(\mathcal{D})$ is a function of τ too. Thus we use u_τ describe the dependence of u on τ, and our

parameter-dependent PDE system is:

$$
\begin{aligned}
-k\Delta\, u_{1\tau} &= q\chi_{\Omega_{1\tau}} && \text{in } \Omega_{1\tau}, && \text{(3.22a)}\\
-k\Delta\, u_{2\tau} &= q\chi_{\Omega_{2\tau}} && \text{in } \Omega_{2\tau}, && \text{(3.22b)}\\
u_{1\tau} &= u_{2\tau} && \text{on } \mathcal{C}_\tau, && \text{(3.22c)}\\
k\left(\frac{\partial u_{2\tau}}{\partial n} - \frac{\partial u_{1\tau}}{\partial n}\right) &= \gamma\,(u_{1\tau} - u_0) && \text{on } \mathcal{C}_\tau, && \text{(3.22d)}\\
k\frac{\partial u_{2\tau}}{\partial n} &= 0 && \text{on } \partial\Omega. && \text{(3.22e)}
\end{aligned}
$$

We note that the same idea for $(\mathcal{D})$ applies here: namely we split our u_τ in two parts: One is $u_{1\tau} := u_\tau\chi_{\Omega_{1\tau}}$ and the other is $u_{2\tau} := u_\tau\chi_{\Omega_{1\tau}}$, representing solution on $\Omega_{1\tau}$ and $\Omega_{2\tau}$ respectively.

3.3.2 Perturbation PDE System

The shape derivative u' of u plays an important part in the adjoint analysis. u' satisfies a so called perturbation PDE system $(\mathcal{P})$which can be derived by apply the shape differentiation on governing PDE $(\mathcal{D})$. Details of the derivation of $(\mathcal{P})$ is provided in this section. All the proof is based on [12] and [11].

Theorem 1. *The shape derivative u' of the solution u satisfies the following perturbation PDE system:*

$$
\begin{aligned}
k\Delta\, u_1' &= 0 && \textit{in } \Omega_1\\
k\Delta\, u_2' &= 0 && \textit{in } \Omega_2\\
u_2' - u_1' &= \left(\frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n}\right)\zeta && \textit{on } \mathcal{C}_\tau\\
k\left(\frac{\partial u_2'}{\partial n} - \frac{\partial u_1'}{\partial n}\right) - \gamma\, u_1' &= \gamma\left[\frac{\partial u_1}{\partial n} + \kappa\,(u_1 - u_0)\right]\zeta && \textit{on } \mathcal{C}_\tau\\
k\frac{\partial u_2'}{\partial n} &= 0 && \textit{on } \partial\Omega
\end{aligned}
\qquad (\mathcal{P})
$$

where $\boldsymbol{Z}$ *is some certain perturbation on* $\mathcal{C}$, $\mathbf{n}$ *is the outer normal direction on* $\mathcal{C}$ *and* $\zeta = \boldsymbol{Z} \cdot \mathbf{n}$.

Proof:

Step 1: $k\Delta u_1' = 0 \quad \text{in}\,\Omega_1$

First we multiply $(\mathcal{D}\,1)$ with some testing function $v \in H_0^1(\mathbb{R} \times \Omega_1)$, integrate the resulting equation and then shape differentiate it. Finally the relation can be obtained from a weak form.

For any $v \in H_0^1(\mathbb{R} \times \Omega_1)$ define $v_\tau(\mathbf{x}) = v \circ T_\tau^{-1}(\mathbf{x})$, then $\dot{v} := \dot{v}(0, \mathbf{x}) = 0$, since:

$$\begin{aligned} \dot{v}(0,\mathbf{x}) &= \left.\frac{dv_\tau(\tau, \mathbf{x}_\tau)}{d\tau}\right|_{\tau=0} \\ &= \left.\frac{dv_\tau(\tau, \mathbf{x} + \tau\boldsymbol{Z})}{d\tau}\right|_{\tau=0} \\ &= \left.\frac{d}{d\tau}\left(v \circ T_\tau^{-1}(\mathbf{x} + \tau\boldsymbol{Z})\right)\right|_{\tau=0} \\ &= \left.\frac{dv(\mathbf{x})}{d\tau}\right|_{\tau=0} = 0 \end{aligned} \tag{3.23}$$

Thus we have

$$v' = -\nabla v \cdot \boldsymbol{Z} \tag{3.24}$$

in which $v' := \frac{\partial v(0,\mathbf{x})}{\partial \tau}$, since $\dot{v} = v' + \nabla \boldsymbol{Z}$. Multiply both sides of equation (3.22a) by v_τ and then do the integration over $\Omega_{1\tau}$, applying the Green's formula

we have:

$$LHS: \qquad -k\int_{\Omega_{1\tau}} \Delta\, u_{1\tau} v_\tau \, d\mathbf{x}_\tau = -k\int_{\mathcal{C}_\tau} \frac{\partial u_{1\tau}}{\partial n} v_\tau \, d\mathbf{x}_\tau + k\int_{\Omega_{1\tau}} \nabla\, u_{1\tau} \cdot \nabla\, v_\tau \, d\mathbf{x}_\tau$$
$$= k\int_{\Omega_{1\tau}} \nabla\, u_{1\tau} \cdot \nabla\, v_\tau \, d\mathbf{x}_\tau \quad \text{by } v_\tau \in H_0^1(\Omega_{1\tau}) \tag{3.25a}$$

$$RHS: \qquad \int_{\Omega_{1\tau}} q v_\tau \, d\mathbf{x}_\tau \tag{3.25b}$$

Take the material derivative of LHS and RHS w. r. t. τ at $\tau = 0$ and by Lemma 1 we have for the LHS:

$$\begin{aligned}
&-k\frac{d}{d\tau}\left(\int_{\Omega_{1\tau}} \Delta\, u_{1\tau} v_\tau \, d\mathbf{x}_\tau\right)\Big|_{\tau=0} \\
&= k\frac{d}{d\tau}\left(\int_{\Omega_{1\tau}} \nabla\, u_{1\tau} \cdot \nabla\, v_\tau \, d\mathbf{x}_\tau\right)\Big|_{\tau=0} \\
&= k\left[\int_{\Omega_1} \nabla\, u_1' \cdot \nabla\, v \, d\mathbf{x} + \int_{\Omega_1} \nabla\, \cdot\, u_1 \nabla\, v' \, d\mathbf{x} + \int_{\mathcal{C}} \nabla\, u_1 \cdot (\nabla\, v)\zeta \, d\mathbf{s}\right] \\
&= k\left[\int_{\Omega_1} \nabla\, u_1' \cdot \nabla\, v \, d\mathbf{x} - \int_{\Omega_1} \nabla\, u_1 \cdot \nabla\, (\nabla\, \cdot\, \zeta)\, d\mathbf{x} + \int_{\mathcal{C}} \nabla\, u_1 \cdot (\nabla\, v)\zeta \, d\mathbf{s}\right]
\end{aligned} \tag{3.26}$$

which is obtained by applying (3.8b), (3.24) and the fact $\zeta = \boldsymbol{Z} \cdot \mathbf{n}$. Now apply Green's formula and (3.24) to the second term of the RHS of the above equation:

$$\int_{\Omega_1} \nabla\, u_1 \cdot \nabla\, (\nabla\, v \cdot\, \boldsymbol{Z})\, d\mathbf{x} = -\int_{\Omega_1} \Delta\, u_1 (\nabla\, v \cdot\, \boldsymbol{Z})\, d\mathbf{x} + \int_{\mathcal{C}} \frac{\partial u_1}{\partial n} (\nabla\, v \cdot\, \boldsymbol{Z})\, d\mathbf{s}$$
$$= \int_{\Omega_1} \Delta\, u_1 v' \, d\mathbf{x} + \int_{\mathcal{C}} \frac{\partial u_1}{\partial n} (\nabla\, v \cdot\, \boldsymbol{Z})\, d\mathbf{s} \tag{3.27}$$

On the contour $\mathcal{C}$ we have:

$$\begin{aligned}\nabla u_1 \cdot (\nabla v)\zeta &= \left(\frac{\partial u_1}{\partial s}, \frac{\partial u_1}{\partial n}\right) \cdot \left(\frac{\partial v}{\partial s}, \frac{\partial v}{\partial n}\right)\zeta \\ &= \frac{\partial u_1}{\partial s}\frac{\partial v}{\partial s}\zeta + \frac{\partial v}{\partial n}\frac{\partial u_1}{\partial n}\zeta \\ &= \frac{\partial v}{\partial n}\frac{\partial u_1}{\partial n}\zeta\end{aligned} \tag{3.28}$$

and

$$\begin{aligned}\frac{\partial u_1}{\partial n}(\nabla \cdot \boldsymbol{Z}) &= \frac{\partial u_1}{\partial n}\left(\frac{\partial v}{\partial \mathbf{s}}, \frac{\partial v}{\partial n}\right) \cdot (\nu, \zeta) \\ &= \frac{\partial u_1}{\partial n}\left(\frac{\partial v}{\partial s}\nu + \frac{\partial v}{\partial n}\zeta\right) \\ &= \frac{\partial u_1}{\partial n}\frac{\partial v}{\partial s}\nu + \frac{\partial u_1}{\partial n}\frac{\partial v}{\partial n}\zeta \\ &= \frac{\partial u_1}{\partial n}\frac{\partial v}{\partial n}\zeta\end{aligned} \tag{3.29}$$

in which $\mathbf{s}$ is the unit tangential vector on $\mathcal{C}$, $\nu := \boldsymbol{Z} \cdot \mathbf{s}$ and $\zeta := \boldsymbol{Z} \cdot \mathbf{n}$. Also we notice that $v = 0$ on $\mathcal{C}$, so that $\frac{\partial v}{\partial \mathbf{s}} = 0$.

Combining (3.27), (3.28) and (3.29), we have:

$$k\frac{d}{d\tau}\left(\int_{\Omega_{1\tau}} \nabla u_{1\tau} \cdot \nabla v_\tau \, d\mathbf{x}_\tau\right)\Big|_{\tau=0} = k\left(\int_{\Omega_1} \nabla u_1' \cdot \nabla v \, d\mathbf{x} - \int_{\Omega_1} \Delta u_1 v' \, d\mathbf{x}\right). \tag{3.30}$$

For the material differentiation of RHS (3.25b) w. r. t. τ at $\tau = 0$, we apply (3.8b) and the fact that $g' = 0$ together with $v = 0$ on $\mathcal{C}$:

$$\begin{aligned}\frac{d}{d\tau}\left(\int_{\Omega_{1\tau}} q v_\tau \, d\mathbf{x}_\tau\right)\Big|_{\tau=0} &= \int_{\Omega_1} (qv)' \, d\mathbf{x} + \int_{\mathcal{C}} qv\zeta \, d\mathbf{s} \\ &= \int_{\Omega_1} qv' \, d\mathbf{x}.\end{aligned} \tag{3.31}$$

We notice the fact that

$$\begin{aligned} &-k\Delta\, u_1 = q \Longrightarrow \\ &-\int_{\Omega_1} \Delta\, u_1 v' d\mathbf{x} = \int_{\Omega_1} q v' d\mathbf{x} \end{aligned} \tag{3.32}$$

and also LHS (3.25a) equals RHS (3.25b), we have:

$$k\int_{\Omega_1} \nabla\, u_1' \cdot \nabla\, v\, d\mathbf{x} = 0$$

which means:

$$k\Delta\, u_1' = 0 \qquad \text{in}\Omega_1 \tag{3.33}$$

Step 2: $k\Delta\, u_1' = 0 \quad \text{in}\,\Omega_2$

The proof follows the same idea as what we did in Step 1. For all $v \in H_0^1(\Omega_2)$ we define $v_\tau(\mathbf{x}) := v \circ T_\tau^{-1}(\mathbf{x})$, then $v_\tau(\mathbf{x}) \in H_0^1(\Omega_{2\tau})$ and we have $v' = -\nabla\, v \cdot \boldsymbol{Z}$. First we multiply each side of (3.22b) by v_τ and then do the integration over $\Omega_{2\tau}$:

$$-k\int_{\Omega_{2\tau}} \Delta\, u_{2\tau} v_\tau\, d\mathbf{x}_\tau = \int_{\Omega_{2\tau}} q v_\tau\, d\mathbf{x}_\tau. \tag{3.34}$$

We apply Green's formula to the LHS of above equation:

$$\begin{aligned} &-k\int_{\Omega_{2\tau}} \Delta\, u_{2\tau} v_\tau\, d\mathbf{x}_\tau \\ &= k\left[\int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, v_\tau\, d\mathbf{x}_\tau + \int_{\mathcal{C}} \frac{\partial u_{2\tau}}{\partial n} v_\tau\, d\mathbf{s}_\tau - \int_{\partial\,\Omega} \frac{\partial u_{2\tau}}{\partial n} v_\tau\, d\mathbf{s}\right] \\ &= k\int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, v_\tau\, d\mathbf{x}_\tau \end{aligned} \tag{3.35}$$

by $v_\tau = 0$ on $\mathcal{C}_\tau$ and $\partial\,\Omega$. Next take the derivative of the LHS of (3.34) w. r. t.

τ at $\tau = 0$:

$$
\begin{aligned}
&- k\frac{d}{d\tau}\left(\int_{\Omega_{2\tau}} \Delta\, u_{2\tau} v_\tau \, d\mathbf{x}_\tau\right)\Big|_{\tau=0} \\
&= k\frac{d}{d\tau}\left(\int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, v_\tau \, d\mathbf{x}_\tau\right)\Big|_{\tau=0} \\
&= k\left[\int_{\Omega_2} (\nabla\, u_2 \cdot \nabla\, v)' d\mathbf{x} - \int_{\mathcal{C}} \nabla\, u_2 \cdot (\nabla\, v)\zeta \, d\mathbf{s} + \int_{\partial\Omega} \nabla\, u_2 \cdot \nabla\, v(\boldsymbol{Z} \cdot v)\, d\mathbf{s}\right] \\
&= k\left[\int_{\Omega_2} \nabla\, u_2' \cdot \nabla\, v d\mathbf{x} + \int_{\Omega_2} \nabla\, u_2 \cdot \nabla\, v' \, d\mathbf{x} - \int_{\mathcal{C}} \nabla\, u_2 \cdot \nabla\, v\zeta \, d\mathbf{s}\right]
\end{aligned}
\tag{3.36}
$$

when we also used $\zeta = 0$ on $\partial\Omega$. We will rearrange the second term of RHS in the above equation by Green's formula:

$$
\begin{aligned}
&\int_{\Omega_2} \nabla\, u_2 \cdot \nabla\, v' d\mathbf{x} \\
&= -\int_{\Omega_2} \Delta\, u_2 v' d\mathbf{x} - \int_{\mathcal{C}} \frac{\partial u_2}{\partial n} v' d\mathbf{s} + \int_{\partial\,\Omega} \frac{\partial u_2}{\partial n} v' d\mathbf{s} \\
&= -\int_{\Omega_2} \Delta\, u_2 v' d\mathbf{x} + \int_{\mathcal{C}} \frac{\partial u_2}{\partial n} (\nabla\, v \cdot \boldsymbol{Z}) d\mathbf{s}
\end{aligned}
\tag{3.37}
$$

using $v' = -\nabla \cdot \boldsymbol{Z}$ and $\frac{\partial u_2}{\partial n} = 0$ on $\partial\,\Omega$. Substituting (3.37) back in (3.36) and applying the fact:

$$
\nabla\, u_2 \cdot \nabla\, v\zeta = \frac{\partial u_2}{\partial n} (\nabla\, v \cdot \boldsymbol{Z}) \qquad \text{on}\, \mathcal{C} \tag{3.38}
$$

which was already proved in (3.28) and (3.29), we have:

$$
-k\frac{d}{d\tau}\left(\int_{\Omega_{2\tau}} \Delta\, u_{2\tau} v_\tau d\mathbf{x}_\tau\right)\Big|_{\tau=0} = k\left[\int_{\Omega_2} \nabla\, u_2' \cdot \nabla\, v d\mathbf{x} - \int_{\Omega_2} \Delta\, u_2 v' d\mathbf{x}\right]. \tag{3.39}
$$

Take the derivative of RHS of (3.34) w. r. t. τ at $\tau = 0$ and combine (3.8b):

$$\begin{aligned}
&\frac{d}{d\tau}\left(\int_{\Omega_{2\tau}} qv_\tau d\mathbf{x}_\tau\right)\Big|_{\tau=0} \\
&= \int_{\Omega_2} qv' d\mathbf{x} + \int_{\partial\,\Omega} qv(\boldsymbol{Z}\,\cdot\,\mathbf{n}) d\mathbf{s} - \int_{\mathcal{C}} qv\zeta\, d\mathbf{s} \\
&= \int_{\Omega_2} qv' d\mathbf{x}
\end{aligned} \tag{3.40}$$

using $q' = 0$, $v = 0$ on $\mathcal{C}$ and $\boldsymbol{Z} = 0$ on $\partial\,\Omega$. Remind that the derivative of RHS of (3.34) equals that of LHS, and $-k\Delta\, u_2 = q$. We have in weak form:

$$k\int_{\Omega_2} \nabla\, u_2' \cdot \nabla\, v d\mathbf{x} = 0 \tag{3.41}$$

which means: $k\Delta\, u_2' = 0$.

Step 3: $u_2' - u_1' = \left(\frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n}\right)\zeta \quad \text{on}\,\mathcal{C}$

This relation is proved by the material derivative of u in the local curvilinear coordinate. Recall our definition of material derivative $\dot{u} := u' + \nabla\, u \cdot \boldsymbol{Z}$. We can rearrange the term using the local curvilinear coordinate on $\mathcal{C}_\tau$ where $\mathbf{s}$ and $\mathbf{n}$ are unit tangential and normal vector respectively. Since we only take account of the normal perturbation of $\mathcal{C}_\tau$, we can set our $\boldsymbol{Z}$ as $\zeta\,\mathbf{n}$ and the $\nabla\, u$ in curvilinear coordinates is

$$\nabla\, u = \left(\frac{\partial}{\partial \mathbf{s}}, \frac{\partial}{\partial n}\right) u.$$

Also notice that $\zeta\,\mathbf{n} = (0, \zeta)$ in the local curvilinear coordinates, so that:

$$\begin{aligned}
\dot{u} &= u' + \left(\frac{\partial}{\partial \mathbf{s}}, \frac{\partial}{\partial n}\right) u \cdot \zeta\,\mathbf{n} \\
&= u' + \left(\frac{\partial}{\partial \mathbf{s}}, \frac{\partial}{\partial n}\right) u \cdot (0, \zeta) \\
&= u' + \frac{\partial u}{\partial n}\zeta.
\end{aligned} \tag{3.42}$$

This in fact reflects the material derivative of u in curvilinear coordinate. Take the material derivative of $u_{1\tau}$ and $u_{2\tau}$ of equation (3.22c) at $\tau = 0$ and by (3.42):

$$\begin{aligned} &\dot{u}_{1\tau}\big|_{\tau=0} = \dot{u}_{2\tau}\big|_{\tau=0} \Longrightarrow \\ &u_1' + \frac{\partial u_1}{\partial n}\zeta = u_2' + \frac{\partial u_2}{\partial n}\zeta \Longrightarrow \\ &u_2' - u_1' = \left(\frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n}\right)\zeta. \end{aligned} \tag{3.43}$$

Step 4: $k\,(\frac{\partial u_2'}{\partial n} - \frac{\partial u_1'}{\partial n}) - \gamma\, u_1' = \gamma\, \left[\frac{\partial u_1}{\partial n} + \kappa\,(u_1 - u_0)\right]\zeta \quad$ on $\mathcal{C}$

As in the Step 3, we can apply the material derivative u in local curvilinear coordinate and the Laplace-Beltrami operator to prove this relation. Taking the material derivative at $\tau = 0$ of both sides of (3.22d) gives:

$$\begin{aligned} &k\left[\frac{\partial u_2'}{\partial n} + \frac{\partial}{\partial n}\left(\frac{\partial u_2}{\partial n}\zeta\right) - \frac{\partial u_1'}{\partial n} - \frac{\partial}{\partial n}\left(\frac{\partial u_1}{\partial n}\zeta\right)\right] \\ &= k\left[\frac{\partial u_2'}{\partial n} - \frac{\partial u_1'}{\partial n} + \left(\frac{\partial^2 u_2}{\partial n^2} - \frac{\partial^2 u_1}{\partial n^2}\right)\zeta + \left(\frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n}\right)\frac{\partial \zeta}{\partial n}\right] \\ &= \gamma\,\left(u_1' + \frac{\partial u_1}{\partial n}\zeta\right). \end{aligned} \tag{3.44}$$

The Laplace-Beltrami operator is defined as:

$$\Delta_{\mathbf{n},\mathbf{s}} := \Delta - \kappa\,\frac{\partial}{\partial n},$$

where $\Delta_{\mathbf{n},\mathbf{s}} := \frac{\partial^2}{\partial n^2} + \frac{\partial^2}{\partial \mathbf{s}^2}$ and Δ is Laplacian in Cartesian coordinate system, so we have:

$$\frac{\partial^2}{\partial n^2} = \Delta\, - \frac{\partial^2}{\partial \mathbf{s}^2} - \kappa\,\frac{\partial}{\partial n}. \tag{3.45}$$

Thus:

$$\begin{aligned}&\left(\frac{\partial^2 u_2}{\partial n^2}-\frac{\partial^2 u_1}{\partial n^2}\right)=\\&\Delta\, u_2-\Delta\, u_1-\frac{\partial^2 u_2}{\partial \mathbf{s}^2}+\frac{\partial^2 u_1}{\partial \mathbf{s}^2}-\kappa\,\frac{\partial u_2}{\partial n}+\kappa\,\frac{\partial u_1}{\partial n}\\&=\kappa\,\left(\frac{\partial u_1}{\partial n}-\frac{\partial u_2}{\partial n}\right),\end{aligned}\tag{3.46}$$

noticing that since $u_2 = u_1$ on $\mathcal{C}$ we have $\frac{\partial^2 u_1}{\partial \mathbf{s}^2} = \frac{\partial^2 u_2}{\partial \mathbf{s}^2}$, and that $\Delta\, u_2 = \Delta\, u_1$ on $\mathcal{C}$ if we smoothly extend the Laplacian of u_1 and u_2 to $\mathcal{C}$.

We claim that $\frac{\partial \zeta}{\partial n} = 0$. The reason is that $\zeta := \boldsymbol{Z} \cdot \mathbf{n}$ and in local curvilinear coordinate system it can be viewed as a function of $\mathbf{s}$ only: $\zeta = \zeta(\mathbf{s})$, thus: $\frac{\partial \zeta(\mathbf{s})}{\partial n} = 0$. Also applying the fact:

$$\frac{\partial u_2}{\partial n}-\frac{\partial u_1}{\partial n}=\frac{\gamma}{k}(u_1-u_0),$$

then relation (3.44) can be simplified as:

$$k\left[\frac{\partial u_2'}{\partial n}-\frac{\partial u_1'}{\partial n}-\frac{\kappa\,\gamma}{k}(u_1-u_0)\zeta\right]=\gamma\,\left(u_1'+\frac{\partial u_1}{\partial n}\zeta\right).$$

which reads:

$$u_2'-u_1'=\left(\frac{\partial u_1}{\partial n}-\frac{\partial u_2}{\partial n}\right)\zeta$$

Step 5: $k\frac{\partial u_2'}{\partial n} = 0 \quad$ on $\partial\,\Omega$

Directly applying material derivative at $\tau = 0$ to (3.22d) gives:

$$k\frac{\partial u_2'}{\partial n}+k\frac{\partial}{\partial n}\left[u_2\boldsymbol{Z}\cdot\mathbf{n}\right]=0$$

and recalling that $\boldsymbol{Z} \cdot \mathbf{n} = 0$, we have

$$k\frac{\partial u_2'}{\partial n} = 0.$$

3.3.3 Adjoint System

We now need to find the adjoint system

$$\begin{cases} L^*(u^*) = f^* & \text{in } \Omega \\ B^*(u^*) = g^* & \text{on } \mathcal{C} \\ C^*(u^*) = l^* & \text{on } \partial\,\Omega \end{cases}$$

such that $\int_A (u - \overline{u})u' d\mathbf{x} = \int_{\mathcal{C}} \boldsymbol{\nabla}^{L^2}\mathcal{J}\zeta\, d\mathbf{s}$, for some $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ as a function of u^* and u.

First suppose two shape-independent functions

$$u_1^* : \Omega_1 \longrightarrow \mathbb{R},$$
$$u_2^* : \Omega_2 \longrightarrow \mathbb{R}.$$

By shape-independent we mean $(u_1^*)' = 0$ and $(u_2^*)' = 0$. We multiply (3.22a) by u_1^*, integrate over Ω, and then apply Green's formula:

$$\begin{aligned} & -k\int_{\Omega_{1\tau}} \Delta\, u_{1\tau} u_1^*\, d\mathbf{x} \\ &= -k\left[\int_{\mathcal{C}_\tau} \frac{\partial u_{1\tau}}{\partial n} u_1^*\, d\mathbf{x} - \int_{\Omega_{1\tau}} \nabla\, u_{1\tau} \cdot \nabla\, u_1^*\, d\mathbf{x}\right] \\ &= \int_{\Omega_{1\tau}} q u_1^*\, d\mathbf{x} \end{aligned} \tag{3.47}$$

Next we take the shape derivative of both sides at $\tau = 0$:

$$\begin{aligned}
&- k\frac{d}{d\tau}\left[\int_{\mathcal{C}_\tau} \frac{\partial u_{1\tau}}{\partial n} u_1^*\, d\mathbf{x} - \int_{\Omega_{1\tau}} \nabla\, u_{1\tau} \nabla\, u_1^*\, d\mathbf{x}\right]\Bigg|_{\tau=0} \\
&= -k\Bigg\{\int_{\mathcal{C}}\left(\frac{\partial u_1}{\partial n} u_1^*\right)' d\mathbf{s} + \int_{\mathcal{C}}\left[\frac{\partial}{\partial n}\left(\frac{\partial u_1}{\partial n} u_1^*\right) + \kappa\, \frac{\partial u_1}{\partial n} u_1^*\right]\zeta\, d\mathbf{s} \\
&- \int_{\Omega_1} \nabla\, u_1' \cdot \nabla\, u_1^*\, d\mathbf{x} - \int_{\mathcal{C}} \nabla\, u_1 \cdot (\nabla\, u_1^*)\zeta\, d\mathbf{s}\Bigg\} \\
&= \int_{\Omega_1} (q u_1^*)'\, d\mathbf{x} + \int_{\mathcal{C}} q u_1^* \zeta\, d\mathbf{s}
\end{aligned}$$

then we apply the Green's formula to the $\nabla\, u_1' \cdot \nabla\, u_1^*$ term:

$$\int_{\Omega_1} \nabla\, u_1' \cdot \nabla\, u_1^*\, d\mathbf{x} = -\int_{\Omega_1} \Delta\, u_1^* u_1'\, d\mathbf{x} + \int_{\mathcal{C}} \frac{\partial u_1^*}{\partial n} u_1'\, d\mathbf{s}$$

and consider the fact that $q' = 0$ and $(u_1^*)' = 0$:

$$\begin{aligned}
&- k\Bigg\{\int_{\Omega_1} \Delta\, u_1^* u_1'\, d\mathbf{x} + \int_{\mathcal{C}} \frac{\partial u_1'}{\partial n} u_1^* - \frac{\partial u_1^*}{\partial n} u_1'\, d\mathbf{s} \\
&+ \int_{\mathcal{C}}\left[\frac{\partial}{\partial n}\left(\frac{\partial u_1}{\partial n} u_1^*\right) + \kappa\, \frac{\partial u_1}{\partial n} u_1^* - \nabla\, u_1 \nabla\, u_1^*\right]\zeta\, d\mathbf{s}\Bigg\} = \int_{\mathcal{C}} q u_1^* \zeta\, d\mathbf{s}.
\end{aligned} \tag{3.48}$$

Then multiply (3.22b) by u_2^* and integrate over Ω_2, also apply Green's formula:

$$\begin{aligned}
&- k \int_{\Omega_{2\tau}} \Delta\, u_{2\tau} u_2^*\, d\mathbf{x}_\tau \\
&= -k\left[-\int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, u_2^*\, d\mathbf{x}_\tau + \int_{\partial\,\Omega} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, ds - \int_{\mathcal{C}_\tau} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, d\mathbf{s}_\tau\right] \\
&= k\left[\int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, u_2^*\, d\mathbf{x}_\tau + \int_{\mathcal{C}_\tau} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, d\mathbf{s}_\tau\right] \\
&= \int_{\Omega_{2\tau}} q u_2^*\, d\mathbf{x}_\tau
\end{aligned}$$

we notice that $\frac{\partial u_{2\tau}}{\partial n} = 0$. Apply shape differentiation to both sides of the above

equation:

$$
\begin{aligned}
&k\Big\{\int_{\Omega_2}\nabla u_2'\cdot\nabla u_2^*\,d\mathbf{x}+\int_{\partial\,\Omega}\nabla u_2\cdot\nabla u_2^*(\boldsymbol{Z}\cdot\mathbf{n})\,d\mathbf{s}-\int_{\mathcal{C}}\nabla u_2\cdot\nabla\,(u_2^*\zeta)\,d\mathbf{s}\\
&+\int_{\mathcal{C}}\frac{\partial u_2'}{\partial n}u_2^*\,d\mathbf{s}+\int_{\mathcal{C}}\left[\frac{\partial}{\partial n}\left(\frac{\partial u_2}{\partial n}u_2^*\right)+\kappa\,\frac{\partial u_2}{\partial n}u_2^*\right]\zeta\,d\mathbf{s}\Big\}\\
&=\int_{\Omega_2}(qu_2^*)'\,d\mathbf{x}+\int_{\partial\,\Omega}qu_2^*\boldsymbol{Z}\cdot\,\mathbf{n}-\int_{\mathcal{C}}qu_2^*\zeta\,d\mathbf{s}
\end{aligned}
$$

Combing with the $g'=0$, $(u_2^*)'=0$, $\zeta=0$ on $\partial\Omega$, and the Green's formula:

$$
\begin{aligned}
&\int_{\Omega_2}\nabla u_2'\cdot\nabla u_2^*\,d\mathbf{x}\\
&=-\int_{\Omega_2}\Delta\,u_2^*u_2'\,d\mathbf{x}+\int_{\partial\,\Omega}\frac{\partial u_2^*}{\partial n}u_2'\,d\mathbf{s}-\int_{\mathcal{C}}\frac{\partial u_2^*}{\partial n}u_2'\,d\mathbf{s},
\end{aligned}
$$

we have:

$$
\begin{aligned}
&k\Big\{-\int_{\Omega_2}\Delta\,u_2^*u_2'\,d\mathbf{x}+\int_{\partial\,\Omega}\frac{\partial u_2^*}{\partial n}u_2'\,d\mathbf{s}+\int_{\mathcal{C}}\frac{\partial u_2'}{\partial n}u_2^*-\frac{\partial u_2^*}{\partial n}u_2'\,d\mathbf{s}\\
&+\int_{\mathcal{C}}\Big[\frac{\partial}{\partial n}\left(\frac{\partial u_2}{\partial n}u_2^*\right)+\kappa\,\frac{\partial u_2}{\partial n}u_2^*-\nabla\,u_2\cdot\nabla\,u_2^*\Big]\zeta\,d\mathbf{s}\Big\}=-\int_{C}qu_2^*\zeta\,d\mathbf{s}.
\end{aligned}
\tag{3.49}
$$

Next we sum (3.48) and (3.49) then group the terms, we obtain:

$$
\begin{aligned}
&k\Big\{-\int_{\Omega_1}\Delta\,u_1^*u_1'\,d\mathbf{x}-\int_{\Omega_2}\Delta\,u_2^*u_2'\,d\mathbf{x}+\int_{\mathcal{C}}\frac{\partial u_2'}{\partial n}u_2^*-\frac{\partial u_1'}{\partial n}u_1^*+\frac{\partial u_1^*}{\partial n}u_1'-\frac{\partial u_2^*}{\partial n}u_2'\,d\mathbf{s}\\
&\int_{\mathcal{C}}\Big[\frac{\partial}{\partial n}\left(\frac{\partial u_1}{\partial n}u_1^*-\frac{\partial u_1}{\partial n}u_1^*\right)+\kappa\left(\frac{\partial u_2}{\partial n}u_2^*-\frac{\partial u_1}{\partial n}u_1^*\right)\\
&+(\nabla\,u_1\cdot\nabla\,u_1^*-\nabla\,u_2\cdot\nabla\,u_2^*)\Big]\zeta\,d\mathbf{s}+\int_{\partial\,\Omega}\frac{\partial u_2^*}{\partial n}u_2'\,d\mathbf{s}\Big\}=\int_{\mathcal{C}}(qu_1^*-qu_2^*)\zeta\,d\mathbf{s}.
\end{aligned}
\tag{3.50}
$$

Then next we choose

$$
\begin{aligned}
-k\Delta\,u_1^*&=(u-\overline{u})\chi_{A_1},\\
-k\Delta\,u_2^*&=(u-\overline{u})\chi_{A_2},
\end{aligned}
\tag{3.51}
$$

where $A_1 = A \bigcap \Omega_1$ and $A_2 = A \bigcap \Omega_2$, we can recast the first integration of (3.50) as:

$$\begin{aligned}
&- \int_{\Omega_1} \Delta\, u_1^* u_1'\, d\mathbf{x} - \int_{\Omega_2} \Delta\, u_2^* u_2'\, d\mathbf{x} \\
&= \int_{\Omega} (u - \overline{u}) \chi_{A_1} u_1'\, d\mathbf{x} + \int_{\Omega} (u - \overline{u}) \chi_{A_2} u_2'\, d\mathbf{x} \\
&= \int_{A} (u - \overline{u}) u'\, d\mathbf{x} \\
&= \mathcal{J}'(\mathcal{C}).
\end{aligned}$$

Also choose the boundary condition:

$$\frac{\partial u_2^*}{\partial n} = 0 \quad \text{on } \partial\, \Omega, \tag{3.52}$$

then (3.50) can be rewrote as:

$$\mathcal{J}'(\mathcal{C}) = B,$$

where

$$\begin{aligned}
B = &- k \int_{\mathcal{C}} \frac{\partial u_2'}{\partial n} u_2^* - \frac{\partial u_1'}{\partial n} u_1^* + \frac{\partial u_1^*}{\partial n} u_1' - \frac{\partial u_2^*}{\partial n} u_2'\, d\mathbf{s} + \int_{\mathcal{C}} (q u_1^* - q u_2^*) \zeta\, d\mathbf{s} \\
&- k \int_{\mathcal{C}} \Bigg[\frac{\partial}{\partial n} \left(\frac{\partial u_2}{\partial n} u_2^* - \frac{\partial u_1}{\partial n} u_1^* \right) + \kappa \left(\frac{\partial u_2}{\partial n} u_2^* - \frac{\partial u_1}{\partial n} u_1^* \right) \\
&+ (\nabla\, u_1 \cdot \nabla\, u_1^* - \nabla\, u_2 \cdot \nabla\, u_2^*) \Bigg] \zeta\, d\mathbf{s}.
\end{aligned} \tag{3.53}$$

First we can make some simplifications in B. Suppose its integrand is b, and let:

$$\begin{aligned}
e := &\nabla\, u_1 \cdot \nabla\, u_1^* - \nabla\, u_2 \cdot \nabla\, u_2^* + \frac{\partial}{\partial n} \left(\frac{\partial u_2}{\partial n} u_2^* - \frac{\partial u_1}{\partial n} u_1^* \right) \\
&+ \kappa \left(\frac{\partial u_2}{\partial n} u_2^* - \frac{\partial u_1}{\partial n} u_1^* \right)
\end{aligned} \tag{3.54}$$

then if expressed in local curvilinear coordinate system on the contour $\mathcal{C}$, e can be

recast as:

$$\begin{aligned} e &= \left(\frac{\partial u_1}{\partial \mathbf{s}}, \frac{\partial u_1}{\partial n}\right) \cdot \left(\frac{\partial u_1^*}{\partial \mathbf{s}}, \frac{\partial u_1^*}{\partial n}\right) - \left(\frac{\partial u_2}{\partial \mathbf{s}}, \frac{\partial u_2}{\partial n}\right) \cdot \left(\frac{\partial u_2^*}{\partial \mathbf{s}}, \frac{\partial u_2^*}{\partial n}\right) \\ &+ \frac{\partial^2 u_2}{\partial n^2} u_2^* + \frac{\partial u_2}{\partial n}\frac{\partial u_2^*}{\partial n} - \frac{\partial^2 u_1}{\partial n^2} u_1^* - \frac{\partial u_1}{\partial n}\frac{\partial u_1^*}{\partial n} + \kappa \frac{\partial u_1}{\partial n} u_2^* - \kappa \frac{\partial u_1}{\partial n} u_1^* \\ &= \frac{\partial u_1}{\partial \mathbf{s}}\frac{\partial u_1^*}{\partial \mathbf{s}} - \frac{\partial u_2}{\partial \mathbf{s}}\frac{\partial u_2^*}{\partial \mathbf{s}} + \frac{\partial^2 u_2}{\partial n^2} - \frac{\partial^2 u_1}{\partial n^2} u_1^* + \kappa \frac{\partial u_2}{\partial n} u_2^* - \kappa \frac{\partial u_1}{\partial n} u_1^*. \end{aligned}$$

Recalling the Laplacian-Beltrami operator which gives us:

$$\frac{\partial^2}{\partial n^2} + \kappa \frac{\partial}{\partial n} = \Delta - \frac{\partial^2}{\partial \mathbf{s}^2},$$

we have

$$e = \frac{\partial u_1}{\partial \mathbf{s}}\frac{\partial u_1^*}{\partial \mathbf{s}} - \frac{\partial u_2}{\partial \mathbf{s}}\frac{\partial u_2^*}{\partial \mathbf{s}} + \left(\Delta u_2 - \frac{\partial^2 u_2}{\partial \mathbf{s}^2}\right) u_2^* - \left(\Delta u_1 - \frac{\partial^2 u_1}{\partial \mathbf{s}^2}\right) u_1^*$$

which is obtained by applying the fact that $u_1 = u_2$ on $\mathcal{C}$ and by $\Delta u_2 = \Delta u_1$ on $\mathcal{C}$ which implies: $\frac{\partial^i u_2}{\partial \mathbf{s}^i} = \frac{\partial^i u_1}{\partial \mathbf{s}^i}$ for $i = 1, 2$ on $\mathcal{C}$, then we have:

$$e = \frac{\partial u_1}{\partial \mathbf{s}}\frac{\partial}{\partial \mathbf{s}}(u_1^* - u_2^*) + \left(\Delta u_1 - \frac{\partial^2 u_1}{\partial \mathbf{s}^2}\right)(u_2^* - u_1^*). \tag{3.55}$$

B is then simplified as:

$$\begin{aligned} B = &-k \int_{\mathcal{C}} \frac{\partial u_2'}{\partial n} u_2^* - \frac{\partial u_1'}{\partial n} u_1^* + \frac{\partial u_1^*}{\partial n} u_1' - \frac{\partial u_2^*}{\partial n} u_2' \, d\mathbf{s} \\ &- k \int_{\mathcal{C}} \left[\frac{\partial u_1}{\partial \mathbf{s}}\frac{\partial}{\partial \mathbf{s}}(u_1^* - u_2^*) + \left(\Delta u_1 - \frac{\partial^2 u_1}{\partial \mathbf{s}^2}\right)(u_2^* - u_1^*)\right] \zeta \, d\mathbf{s} \\ &+ \int_{\mathcal{C}} q(u_1^* - u_2^*)\zeta \, d\mathbf{s} \end{aligned} \tag{3.56}$$

Notice that the 2nd and 3rd term of B (3.56) have already been put into Riesz form. Our focus will be converging the 1st term to Riesz form.

The following procedure intend to determine the boundary conditions of $(\mathcal{A})$ with

idea from [14]. Define two symbolic vectors:

$$\begin{aligned} V' &:= \left[u_1', \frac{\partial u_1'}{\partial n}, u_2', \frac{\partial u_2'}{\partial n} \right]^T, \\ V^* &:= \left[u_1^*, \frac{\partial u_1^*}{\partial n}, u_2^*, \frac{\partial u_2^*}{\partial n} \right]^T, \end{aligned} \tag{3.57}$$

then the boundary condition on $\mathcal{C}$ of the perturbed system $(\mathcal{P})$ can be rewritten in matrix form:

$$M \cdot V' = \begin{bmatrix} \frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n} \\ r\left[\frac{\partial u_1}{\partial n} + \kappa \left(u_1 - u_0 \right) \right] \end{bmatrix} \zeta \tag{3.58}$$

where

$$M = \begin{bmatrix} -1 & 0 & 1 & 0 \\ -\gamma & -k & 0 & k \end{bmatrix} \tag{3.59}$$

we treat V' as an unknown solution for linear equations (3.58) and solve it for V', noticing that we have 2 degrees of freedom, so we introduce 2 free variables v_1 and v_2:

$$\begin{aligned} V' &= \zeta \begin{bmatrix} \frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n} \\ -\frac{\gamma}{k}\left[\frac{\partial u_2}{\partial n} + \kappa \left(u_1 - u_0 \right) \right] \\ 0 \\ 0 \end{bmatrix} + v_1 \begin{bmatrix} 1 \\ -\frac{\gamma}{k} \\ 1 \\ 0 \end{bmatrix} + v_2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \\ &= \zeta\, m_0 + v_1 m_1 + v_2 m_2. \end{aligned} \tag{3.60}$$

Next, let us use b_1 to denote the first integrand of B, and we can write b_1 in a matrix form:

$$\begin{aligned} b_1 &= \frac{\partial u_2'}{\partial n} u_2^* - \frac{\partial u_1'}{\partial n} u_1^* + \frac{\partial u_1^*}{\partial n} u_1' - \frac{\partial u_2^*}{\partial n} u_2' \\ &= V^{*T} A V' \end{aligned} \tag{3.61}$$

in which

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}. \tag{3.62}$$

Combining this with (3.60) we have:

$$\begin{aligned} b_1 &= V^{*T} A[\zeta\, m_0 + v_1 m_1 + v_2 m_2] \\ &= (V^{*T} A m_0)\zeta + (V^{*T} A m_1) v_1 + (V^{*T} A m_2) v_2. \end{aligned} \tag{3.63}$$

Clearly if we set $V^{*T} A m_1 = 0$ and $V^{*T} A m_2 = 0$ then our b_1 is a linear form on ζ thus B is in Riesz form. We have therefore:

$$\begin{aligned} V^{*T} A m_1 &= \left[u_1^*, \frac{\partial u_1^*}{\partial n}, u_2^*, \frac{\partial u_2^*}{\partial n} \right] \cdot \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -\frac{\gamma}{k} \\ 1 \\ 0 \end{bmatrix} \\ &= \frac{\partial u_1^*}{\partial n} + \frac{\gamma\, u_1^*}{k} - \frac{\partial u_2^*}{\partial n} = 0, \end{aligned} \tag{3.64}$$

$$\begin{aligned} V^{*T} A m_2 &= \left[u_1^*, \frac{\partial u_1^*}{\partial n}, u_2^*, \frac{\partial u_2^*}{\partial n} \right] \cdot \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \\ &= -u_1^* + u_2^* = 0. \end{aligned} \tag{3.65}$$

Noticing that the 2nd and 3rd integrand in B vanish because of $u_2^* = u_1^*$, we get:

$$\begin{aligned} B &= -k \int_{\mathcal{C}} (V^{*T} A m_0) \zeta \, d\mathbf{s} \\ &= -k \int_{\mathcal{C}} \left\{ \left(\frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n} \right) \frac{\partial u_1^*}{\partial n} + \frac{\gamma \, u_1^*}{k} \left[\frac{\partial u_2}{\partial n} + \kappa \, (u_1 - u_0) \right] \right\} \zeta \, d\mathbf{s} \\ &= \int_{\mathcal{C}} \left\{ -\gamma (u_1 - u_0) \frac{\partial u_1^*}{\partial n} - \gamma \, u_1^* \left[\frac{\partial u_2}{\partial n} + \kappa \, (u_1 - u_0) \right] \right\} \zeta \, d\mathbf{s} \\ &= \int_{\mathcal{C}} \left\{ -\gamma \, (u_1 - u_0) \left(\kappa \, u_1^* + \frac{\partial u_1^*}{\partial n} \right) - \gamma \, u_1^* \frac{\partial u_1}{\partial n} \right\} \zeta \, d\mathbf{s} \end{aligned} \tag{3.66}$$

where we also used the boundary condition from $(\mathcal{D})$: $\frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n} = \frac{\gamma}{k}(u_1 - u_0)$.

Then, our adjoint system is:

$$\begin{aligned} -k\Delta \, u_1^* &= (u - \overline{u}) \chi_{A_1} && \text{in } \Omega_1 \\ -k\Delta \, u_2^* &= (u - \overline{u}) \chi_{A_2} && \text{in } \Omega_2 \\ u_2^* - u_1^* &= 0 && \text{on } \mathcal{C} \\ k \, (\frac{\partial u_2^*}{\partial n} - \frac{\partial u_1^*}{\partial n}) &= \gamma \, u_1^* && \text{on } \mathcal{C} \\ \frac{\partial u_2^*}{\partial n} &= 0 && \text{on } \partial\Omega \end{aligned} \tag{$\mathcal{A}$}$$

with $u := u_1 \chi_{A_1} + u_2 \chi_{A_2}$. And our L^2 gradient without the length constraint is:

$$\boldsymbol{\nabla}^{L^2} \mathcal{J} = -\gamma \, (u_1 - u_0) \left(\kappa \, u_1^* + \frac{\partial u_1^*}{\partial n} \right) - \gamma \, u_1^* \frac{\partial u_1}{\partial n} \qquad \text{on } \mathcal{C}. \tag{3.67}$$

If adding the length constraint we have by (3.15):

$$\boldsymbol{\nabla}^{L^2} \mathcal{J}_L := -\gamma \, (u_1 - u_0) \left(\kappa \, u_1^* + \frac{\partial u_1^*}{\partial n} \right) - \gamma \, u_1^* \frac{\partial u_1}{\partial n} + \alpha \left(\int_{\mathcal{C}} d\mathbf{s} - L \right) \kappa \quad \text{on } \mathcal{C}. \tag{3.68}$$

3.3.4 Preconditioning the Gradient

The shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ is the space in L^2 which only assures its square integrability, without anything about the smoothness. If we update $\mathcal{C}$ along $\boldsymbol{\nabla}^{L^2}\mathcal{J}$, we might lose the smoothness of the contour. By preconditioning on $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ we can have a gradient in the space $H^1(\mathcal{C})$. According to Riesz representation theorem [13] in H^1 there exist $\boldsymbol{\nabla}^{H^1}\mathcal{J} \in H^1(\mathcal{C})$ such that [15]:

$$\begin{aligned}\mathcal{J}'(\mathcal{C};\zeta) &= \left\langle \boldsymbol{\nabla}^{H^1}\mathcal{J}, \zeta \right\rangle_{H^1} \\ &= \int_{\mathcal{C}} \boldsymbol{\nabla}^{H^1}\mathcal{J}\,\zeta + l^2\,\partial_s\,\boldsymbol{\nabla}^{H^1}\mathcal{J}\,\partial_s\,\zeta\,d\mathbf{s}\end{aligned}$$

where l is some constant called a Sobolev coefficient and s is the arc-length axis. Apply integration by parts and also consider (3.12):

$$\int_{\mathcal{C}}(1 - l^2\,\partial_s^2)\,\boldsymbol{\nabla}^{H^1}\mathcal{J}\zeta\,d\,t = \int_{\mathcal{C}}\boldsymbol{\nabla}^{L^2}\mathcal{J}\,\zeta\,d\,s$$

which implies

$$(1 - l^2\,\partial_s^2)\,\boldsymbol{\nabla}^{H^1}\mathcal{J} = \boldsymbol{\nabla}^{L^2}\mathcal{J} \tag{3.69}$$

We can solve (3.69) together with periodic boundary conditions for Sobolev shape gradient $\boldsymbol{\nabla}^{H^1}\mathcal{J}$. Mathematically we can write:

$$\boldsymbol{\nabla}^{H^1}\mathcal{J} = (1 - l^2\,\partial_s^2)^{-1}\,\boldsymbol{\nabla}^{L^2}\mathcal{J}$$

where operator $(1 - l^2\,\partial_s^2)^{-1}$ served as our preconditioner. The same idea can be applied to obtain $\boldsymbol{\nabla}^{H^1}\mathcal{J}_L$ after solving (3.69) with its right hand-side replaced by $\boldsymbol{\nabla}^{L^2}\mathcal{J}_L$. Thus, perturbing $\mathcal{C}$ along Sobolev gradient will ensure the smoothness of $\mathcal{C}$. Moreover, the smoothness of the gradient is controlled by l with high smoothness depending on large l.

Chapter 4

Numerical Implementation

In this chapter we will talk about how to put our mathematical model into numerical implementation and in particular how to compute our $\boldsymbol{\nabla}^{L^2}\mathcal{J}$. As given in the expression (3.67), our $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ is in fact a function of u and u^*, so the first problem is to solve $(\mathcal{D})$ and then $(\mathcal{A})$. Generally speaking, the numerical treatment involves the combination of the spectral method and Boundary Integral Equation (BIE) techniques. In order to make solving $(\mathcal{D})$ and $(\mathcal{A})$ numerically feasible, we divided our u in two parts: one is u_p which forms a Poisson equations in Ω, the other is u_h which is harmonic on $\Omega \setminus \mathcal{C}$. The first part only takes into account the heat generalization with given heat sources, while the harmonic part deals with the heat absorption performed by coolant $\mathcal{C}$. Then we apply BIE techniques to solve for u_p and u_h at the same time. In the boundary integral method we can represents our u_h as a convolution of some density function μ with the logarithmic potential on the contour and thus make it possible for us to first solve a problem of u_p and μ and then reconstruct our u_h in term of μ. Based on this approach, we also calculated the normal derivative terms on $\mathcal{C}$ both for the direct and adjoint problem for which we can obtain our shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ on $\mathcal{C}$. In representing the Laplacian of u_p and its derivatives, Chebyshev spectral method is applied on a non-uniform collocation grid. For the evaluation of our cost functional

Clenshaw-Curtis quadrature is used to maintain spectral accuracy. On the contour $\mathcal{C}$ we use the ordinary equal-spaced grid together with a spectral interpolation method for periodic grid to approximate the derivatives. In section 4.1 and 4.2 we will cover the idea of spectral differentiation method and BIE techniques, including how we couple these two methods together in the computational approach, and then the numerical implementation using the two methods will be discussed in section 4.3.

4.1 Spectral Methods for Periodic and Non-periodic Domains

Spectral method, more specifically, the spectral differentiation technique, is frequently applied in this paper. One important reason is that spectral accuracy can result in error of the order $O(c^N)$ when dealing with analytical functions, for some constant $c \in (0, 1)$ and the number of discretization points N, which is much more promising compared with $O(N^{-m})$ for some m using different finite difference method. Thus it makes it possible to maintain high accuracy. Generally speaking, in our optimization problem spectral methods treating two different scenarios are involved:

1. Spectral differentiation and integration for a bounded non-periodic domain.

2. Spectral differentiation for a periodic domain.

The first technique is applied in representing our Laplacian Δ numerically in Ω and approximating the cost functional on $\mathcal{A}$, while the 2nd technique is used to treat differentiation defined on our $\mathcal{C}$ with periodic structure. We will discuss the details in the following.

4.1.1 Spectral Differentiation for Non-periodic Domain

One common question in our optimization is: how to numerically represent our Laplacian in system $(\mathcal{D})$ and $(\mathcal{A})$. The governing equation $(\mathcal{D}\,1)$ calls for a solution of a Poisson problem:

$$-k\Delta\, u = q \quad \mathbf{x} \in \Omega. \tag{4.1}$$

In order to get an insight of the problem, let us assume we are in 1D case:

$$\begin{aligned} &\Delta := \frac{d^2}{dx^2}, \\ &u \; and \; q : \mathbb{R} \longrightarrow \mathbb{R}, \\ &\Omega^1 : [-1, 1]. \end{aligned} \tag{4.2}$$

There are several different numerical treatments for Δ. For instance, we can first discretize Ω^1 into N equispaced points: $x_1, x_2, ..., x_N$ thus vectors $\mathbf{u}$ and $\mathbf{q}$ defined on $x_i, i = 1, 2, ..., N$ are obtained. Then, we use central difference schemes to represent Δ as a matrix Q, after that we are ready to put (4.1) into vector form $Q\,\mathbf{u} = \mathbf{q}$ and solve it for $\mathbf{u}$ based on boundary information. These treatments, together with spectral method shares a common idea: Discretize the domain, represent Δ in matrix form, then solve the resulting linear system. The unique feature of spectral method is: instead of discretizing the domain with equispaced points, it captures the domain using non-equispaced points with a density approximately [16]:

$$\text{density of points} \sim \frac{N}{\pi\sqrt{1-x^2}}.$$

We can see the density of points is much higher around $x = \pm 1$ than in the interior domain. In other words, the points are clustered on the boundary. The reason why this particular density is chosen is beyond the scope of this paper and can be found at [16]. The advantage of non-equispaced points will be covered later. Let us come

back to the concept of spectral method. Notice that the Δu in one dimension (1D) is the second derivative of u. It is necessary to first look into how spectral method works for the derivative of a function. For simplicity, let us denote $u_i := u(x_i)$ representing function u evaluated at discrete point x_i. Basically, spectral method approximate $u'(x)$ in the following way [16]:

1. construct non-equispaced points on Ω^1: $x_i, i = 0, 2, 3, ..., N$,
2. interpolate $\mathbf{u} := \{u_i\}_{i=0}^{i=N}$ using a polynomial p of degree $\leqslant N$ such that $p_i = u_i$ for $0 \leqslant i \leqslant N$,
3. approximate u_i' by p_i' for $0 \leqslant i \leqslant N$.

In our practical application, we apply:

$$x_i := \cos\left(\frac{i\pi}{N}\right); \qquad i = 0, 1, 2, ..., N \tag{4.3}$$

which is also called *Chebyshev points*. While the idea of spectral method does not seem special at the first glance, in fact the magic is hidden in the so called *Chebyshev points*. Interpolating over this non-equispaced points can have much smaller error compared with doing so based on equispaced points. Because of Runge phenomenon the error of interpolating over equispaced grid using N degree polynomial can explode in the vicinity of boundary at a rate of 2^N. On the other hand, applying clustered points on the boundary can greatly counteract Runge phenomenon and thus attain a high accuracy.

Now it is the time to back to the question: how can we get the spectral differentiation matrix. Suppose we have the nodal values of a function $\{u_i\}_{i=0}^N$ defined on *Chebyshev grid* of size $N + 1$. We would like to find an approximation of the corresponding nodal value of its derivative $\{u_i'\}_{i=0}^N$. Let us denote it with $\{d_i\}_{i=0}^N \approx \{u_i\}_{i=0}^N$.

Following the idea of spectral differentiation we discussed before, we first interpolate $\{u_i\}_{i=0}^N$ by a polynomial function $p(x)$ of degree N and rewrite $p(x)$ in Lagrange form:

$$p(x) = \sum_{i=0}^{N} l_i(x) u_i, \tag{4.4}$$

then we set

$$d_j = p'(x_j) = \sum_{i=0}^{N} l_i'(x_j) u_i \quad \text{for} \, 0 \leqslant j \leqslant N, \tag{4.5}$$

written in matrix form it reads:

$$D_N \, \mathbf{u} = \mathbf{d} \tag{4.6}$$

where $\mathbf{u} = [u_0, u_1, ..., u_N]^T$ and $\mathbf{d} = [u_0, u_1, ..., u_N]^T$. D_N is the differentiation matrix with entries $(D_N)_{ij} := l_i'(x_j)$. In [16] it gives a convenient formula for the entries in D_N(indexed from 0 to N):

$$\begin{aligned}
[D_N]_{00} &= \frac{2N^2+1}{6}, \\
[D_N]_{NN} &= -\frac{2N^2+1}{6}, \\
[D_N]_{ii} &= \frac{-x_i}{2(1-x_i^2)} \quad 1 \leqslant i \leqslant N-1, \\
[D_N]_{ij} &= \frac{c_i(-1)^{i+j}}{c_j(x_i - x_j)} \quad i \neq j; 1 \leqslant i, j \leqslant N-1,
\end{aligned} \tag{4.7}$$

where

$$c_i = \begin{cases} 2, & i = 0 \text{ or } N, \\ 1, & \text{otherwise.} \end{cases}$$

Now we obtain our 1st order spectral differentiation matrix D_N in 1D such that $\mathbf{u}' \approx D_N \cdot \mathbf{u}$. Simply applying the same D_N twice we will have the spectral differentiation matrix D_N^2 for second order derivative of $\mathbf{u}$: $\mathbf{u}'' \approx D_N^2 \, \mathbf{u}$.

At this time, we are in the position to extend our spectral differentiation concept to the 2D case. Suppose for some N, we have a set of nodal values $\{u_{ij}\}, 0 \leqslant i, j \leqslant N$ defined on a 2D *Chebyshev grid*. By 2D *Chebyshev grid* we mean that in each direction independent 1D *Chebyshev grid* are applied, i.e, $[x_0, x_1, ..., x_N]$ in x direction and $[y_0, y_1, ..., y_N]$ in y direction. This 2D *Chebyshev gird* is also called *tensor product grid* (Tg) in [16] with a picture of which shown in Figure 4.1. Then what is the spectral partial derivative of $\{u_{ij}\}$ defined on the Tg in x or y direction? Without lost of generality, let us approximate the partial derivative of u_{ij} in x direction as $[u_x]_{ij} \approx [d_x]_{ij}$ and partial derivative of which in y direction as $[u_y]_{ij} \approx [d_y]_{ij}$. We will use an example to illustrate how can we find our spectral partial differentiation matrix $D_x^{(N)}$ and $D_y^{(N)}$ such that $\mathbf{d}_x = D_x^{(N)} \cdot \mathbf{u}$ and $\mathbf{d}_y = D_y^{(N)} \cdot \mathbf{u}$ where $\mathbf{d}_x$, $\mathbf{d}_y$ and $\mathbf{u}$ are vectors rearranged from $\{u_{ij}\}$, $\{[d_y]_{ij}\}$ and $\{[d_y]_{ij}\}$.

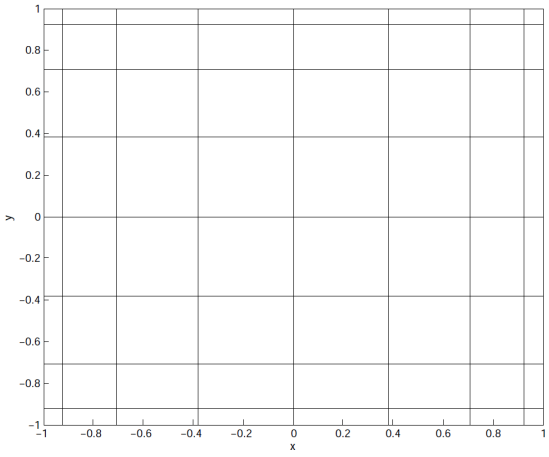


Figure 4.1: Tensor product grid (N=8) [16]

Example

Suppose $N = 2$, $\Omega = [-1, 1]^2$. Then our *Chebyshev points* in the x or y direction are $\mathbf{x} = \mathbf{y} = [1, 0, -1]$ by (4.3). The function u_{ij} is defined on the grid $(\mathbf{x}, \mathbf{y})$ and we rearrange it as a vector following the order as in Figure 4.2: Thus $\mathbf{u} = [u_1, u_2, ..., u_9]$.

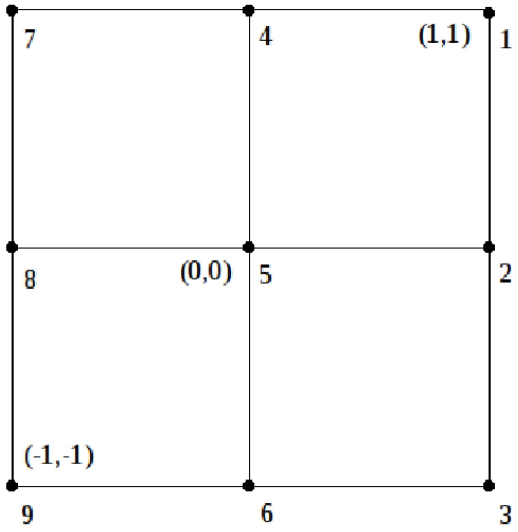


Figure 4.2: Tensor product grid (N=2)

Our $\mathbf{d}_x = [d_x^{(1)}, d_x^{(2)}, ..., d_x^{(9)}]$ and $\mathbf{d}_y = [d_y^{(1)}, d_x^{(2)}, ..., d_x^{(9)}]$ represent the partial derivatives on the grid points in x and y directions respectively. If we confine ourselves to position $1, 2, 3$, i.e, at level $x = 1$, we can find $[d_y^{(1)}, d_y^{(2)}, d_y^{(3)}]$ using 1D spectral differentiation matrix D_2:

$$\begin{bmatrix} d_y^{(1)} \\ d_y^{(2)} \\ d_y^{(3)} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \tag{4.8}$$

where

$$D_2 = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix}. \tag{4.9}$$

Similarly, if we are at level $x = 0$ or $x = -1$, we have similar relation as (4.8) between $d_y^{(4)}, d_y^{(5)}, ..., d_y^{(9)}$ and $u_4, u_5, ..., u_9$. If we put them together we will obtain:

$$\begin{bmatrix} d_y^{(1)} \\ d_y^{(2)} \\ d_y^{(3)} \\ d_y^{(4)} \\ d_y^{(5)} \\ d_y^{(6)} \\ d_y^{(7)} \\ d_y^{(8)} \\ d_y^{(9)} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 2 & -\frac{3}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{2} & -2 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 2 & -\frac{3}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{2} & -2 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix}. \tag{4.10}$$

The 9×9 matrix on the right-hand side applied to $\mathbf{u}$ can be constructed using *Kronecker product* of I_3 and D_2: $I_3 \otimes D_2^2$, where I_3 is 3 by 3 identity matrix and '$\otimes$' is the *Kronecker product* which can be defined as follows:

Definition 1. *A is a $j \times k$ matrix and B is a $r \times s$ matrix, the the Kronecker product of A and B is defined as:*

$$A \otimes B := \begin{pmatrix} a_{11}B & \dots & a_{1k}B \\ \vdots & \ddots & \vdots \\ a_{j1}B & \dots & a_{jk}B \end{pmatrix} \tag{4.11}$$

where

$$a_{lm}B := \begin{pmatrix} a_{lm}b_{11} & \dots & a_{lm}b_{1s} \\ \vdots & \ddots & \vdots \\ a_{lm}b_{r1} & \dots & a_{lm}b_{rs} \end{pmatrix} \qquad 1 \leqslant l \leqslant j; 1 \leqslant m \leqslant k. \tag{4.12}$$

Thus, in the case of arbitrary N we have for the nodal value of partial derivative of u in y direction:

$$\mathbf{d}_y = (I_{N+1} \otimes D_N)\,\mathbf{u} \tag{4.13}$$

noticing that D_N is by definition a $N+1$ by $N+1$ matrix. This formula can be easily extend to the nth partial derivatives case for y:

$$\mathbf{d}_y^{(n)} = (I_{N+1} \otimes D_N^n)\,\mathbf{u}. \tag{4.14}$$

The formula for the partial derivative in x direction is different from that in y direction but shares the same idea. By the same rearrangement of u as in Figure 4.2 the following formula for $\mathbf{d}_x$ holds:

$$\begin{bmatrix} \boxed{d_x^{(1)}} \\ d_x^{(2)} \\ d_x^{(3)} \\ \boxed{d_x^{(4)}} \\ d_x^{(5)} \\ d_x^{(6)} \\ \boxed{d_x^{(7)}} \\ d_x^{(8)} \\ d_x^{(9)} \end{bmatrix} = \begin{bmatrix} \boxed{\frac{3}{2}} & 0 & 0 & \boxed{-2} & 0 & 0 & \boxed{\frac{1}{2}} & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 0 & -2 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{3}{2} & 0 & 0 & -2 & 0 & 0 & \frac{1}{2} \\ \boxed{\frac{1}{2}} & 0 & 0 & \boxed{0} & 0 & 0 & \boxed{-\frac{1}{2}} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ \boxed{-\frac{1}{2}} & 0 & 0 & \boxed{2} & 0 & 0 & \boxed{-\frac{3}{2}} & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & 0 & 2 & 0 & 0 & -\frac{3}{2} & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 2 & 0 & 0 & -\frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} \boxed{u_1} \\ u_2 \\ u_3 \\ \boxed{u_4} \\ u_5 \\ u_6 \\ \boxed{u_7} \\ u_8 \\ u_9 \end{bmatrix}. \tag{4.15}$$

Here again the right-hand side 9×9 matrix can be obtained by $D_2 \otimes I_3$. In order to

get an insight of this 9 by 9 matrix, let us look into its details. If we take the boxed elements in (4.15) out and recapture their relation in linear system, we have:

$$\begin{bmatrix} d_x^{(1)} \\ d_x^{(4)} \\ d_x^{(7)} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_4 \\ u_7 \end{bmatrix} \tag{4.16}$$

and if looking back into Figure 4.2 we will notice that these are actually the nodal values of spectral differentiation in x direction at fixed $y = 1$. In fact, (4.15) is obtained in this way: first we find the nodal derivatives in x direction by 1D spectral differentiation method at different level $y = 1, y = 0$ and $y = -1$, then we rearrange the 3 linear system, all together 9 equations, in one big linear system based on the ordering in Figure 4.2. This is also the idea of (4.10). Now we have confidence to assert that for some N, the nth partial spectral derivative of u w. r. t. x is:

$$\mathbf{d}_x^{(n)} = (D_N^n \otimes I_{N+1})\, \mathbf{u} \tag{4.17}$$

based on the ordering in Figure 4.2. Thus we can write our discrete Laplacian discretized with spectral method as:

$$\Delta_N = I_{N+1} \otimes D_N^2 + D_N^2 \otimes I_{N+1}, \tag{4.18}$$

where the grids used here is the *tensor product grid* defined in $[-1, 1]^2$ based on *Chebyshev gird* in x and y direction respectively. While in our work we used a dense tensor grid of size $N \geqslant 50$ in each direction, following the same idea.

If we want to impose the boundary conditions, we only need to capture the boundary relation by spectral method and modify the corresponding row of the differentiation matrix. Suppose we are dealing with the Poisson problem with Neumann boundary

conditions:

$$-\Delta u = f \qquad \text{in } \Omega, \tag{4.19a}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on } \partial\Omega. \tag{4.19b}$$

By (4.18), we can write our relation (4.19b) as:

$$\Delta_N \cdot \mathbf{u} = \mathbf{f} \tag{4.20}$$

in which $\mathbf{f}$ is a vector with the nodal value on *tensor product grid* after a rearrangement by the ordering in Figure 4.2. We notice that the boundary conditions are in fact $\frac{\partial u}{\partial x}$ or $\frac{\partial u}{\partial y}$ with a sign difference since $\Omega = [-1, 1]^2$. We could approximate the conditions by spectral method: $I_{N+1} \otimes D_N$ and $D_N \otimes I_{N+1}$ respectively. Let us focus on $N = 2$ case, suppose we want to impose $\frac{\partial u}{\partial x} = 0$ on $x = 1$. First we find the position corresponding to $x = 1$ is position 1,2 and 3 by Figure 4.2. Then we replace the 1st, 2nd and 3rd row of Δ_2 with the 1st, 2nd and 3rd row of $I_{N+1} \otimes D_N$. At last, replace the 1st, 2nd and 3rd entries of column vector $\mathbf{f}$ as 0. The implement of boundary conditions on $x = -1$, $y = -1$ and $y = 1$ follows the same procedure.

4.1.2 Clenshaw-Curtis Quadrature

For the evaluation of our cost functional we applied a spectral integration method called Clenshaw-Curtis Quadrature. It shares the same idea of spectral integration. Suppose for some N we have a set of nodal value $\{f_i\}_{i=1}^N$ defined on the Chebyshev grid $[x_1, x_2, ..., x_N]^T$. The spectral integration approximate the integral of f on $[-1, 1]$ by the following philosophy [16]: First we approximate f using a polynomial p such that $p(x_i) = f_i,\ 0 \leqslant i \leqslant N$. Secondly, we approximate the integration of f on $[-1, 1]$

by I_N where

$$I_N = \int_{-1}^{1} p(x)\, d\, x. \tag{4.21}$$

For the Clenshaw-Curtis Quadrature, $p(x)$ is a linear combination of Chebyshev polynomials [16]:

$$p(x) = \sum_{n=0}^{N} a_n T_n(x), \tag{4.22}$$

where a_n can be obtained using the nodal value f_i with a Fast Fourier Transform (FFT) [16] and the Chebyshev polynomial $T_n(x)$ can be given in a recurrence way [16]:

$$\begin{aligned} T_{n+1}(x) &= 2xT_n x - T_{n-1}(x) \\ T_0(x) &= 1;\ T_1(x) = x. \end{aligned} \tag{4.23}$$

Based on this idea, L. N. Trefethen [16] gives a MATLAB code which will generate a weight w_i such that:

$$I_N \approx \sum_{i=1}^{N} w_i\, f_i \tag{4.24}$$

In this work we will develop our Clenshaw-Curtis Quadrature for the 2D integration based on the MATLAB code given in [16].

4.1.3 Spectral Differentiation for Periodic Domains

A periodic domain is a domain where a prescribed function repeats the same value on its boundary. For such functions spectral differentiation works in this way: first we approximate the function by a band-limited interpolant function p and then approximate its derivative by p'. Again, we will show the concept in the following setting:

The domain we are working on is $[0, 2\pi]$ with a N points equispaced discretization $t_i = \frac{2i\pi}{N}, 1 \leqslant i \leqslant N$. And there are a prescribed function u defined on $[0, 2\pi]$ for which only the nodal values $u_i = u(t_i)$ are known. We will demonstrate how we can find the

nodal value for u' by spectral method.

As it is stated in [16], we first perform a Discrete Fourier Transform (DFT) on $\{u_i\}_{i=1}^N$:

$$\hat{u}_k = \frac{2\pi}{N}\sum_{j=1}^{N} e^{-ikt_j}u_j,\ k = -\frac{N}{2}+1, ..., \frac{N}{2}, \tag{4.25}$$

where k is the wavenumber and the reason why we only focus on the $k \in (-\frac{N}{2}, \frac{N}{2})$ is because of the aliasing phenomenon [16]. Then, by the inverse DFT on $\hat{u}_k$ we recover u_j:

$$u_j = \frac{1}{2\pi}\sum_{k=-\frac{N}{2}+1}^{N/2} e^{ikt_j}\hat{u}_k, 1 \leqslant j \leqslant N \tag{4.26}$$

to counteract the effect of $e^{iNt/2}$ when $k = \frac{N}{2}$ we add an additional term $\hat{u}_{-\frac{N}{2}} = \hat{u}_{\frac{N}{2}}$ to (4.26) and replace our (4.26) by:

$$u_j = \frac{1}{2\pi}\left(\sum_{k=-\frac{N}{2}}^{N/2} e^{ikt_j}\hat{u}_k - \hat{u}_{\frac{N}{2}}\right) \qquad 1 \leqslant j \leqslant N. \tag{4.27}$$

Based on the above equation we have for the band-limited interpolant:

$$p(t) = \frac{1}{2\pi}\left(\sum_{k=-\frac{N}{2}}^{N/2} e^{ikt}\hat{u}_k - \hat{u}_{\frac{N}{2}}\right) \qquad t \in [0, 2\pi] \tag{4.28}$$

and the idea is that we interpolate u by this p and approximate $u'(t)$ by $p'(t)$. But this expression for p is not easy to be put into numerical computation so we would like to find a simple expression for our interpolant p. We now first interpolate a Delta function defined on $j = 0$:

$$\delta_j = \begin{cases} 1, & j = 0 \\ 0, & \text{otherwise} \end{cases}$$

and by [16] the final interpolant is

$$S_N(t) = \frac{\sin(\frac{Nt}{2})}{N\tan(\frac{t}{2})}, \tag{4.29}$$

noticing the fact that the discrete value $u_i = \sum_{j=1}^{N} u_j \delta_{\,i-j}$. So we can rewrite $p(t)$ in terms of $S_N(t)$:

$$p(t) = \sum_{j=1}^{N} u_j S_N(t - t_j). \tag{4.30}$$

This expression for p is what we applied in numerical framework. Then approximate $u'(t_i)$ and $u''(t_i$ by $p'(t_i)$ and $p''(t_i)$ respectively:

$$\begin{aligned} u'(t_i) &\approx \sum_{j=1}^{N} u_j S'_N(t_i - t_j) \qquad 1 \leqslant i \leqslant N, \\ u''(t_i) &\approx \sum_{j=1}^{N} u_j S''_N(t_i - t_j) \qquad 1 \leqslant i \leqslant N. \end{aligned} \tag{4.31}$$

If written in matrix form:

$$\begin{aligned} \mathbf{u}' &\approx P_N^{(1)} \cdot \mathbf{u}, \\ \mathbf{u}'' &\approx P_N^{(2)} \cdot \mathbf{u} \end{aligned} \tag{4.32}$$

where $\mathbf{u}' = \{u'(t_i)\}_{i=1}^{N}$ and $\mathbf{u}'' = \{u''(t_i)\}_{i=1}^{N}$ and $P_N^{(1)}$ and $P_N^{(2)}$ are N by N matrix with elements:

$$\begin{aligned} (P_N^{(1)})_{ij} &= S'_N(t_i - t_j), \\ (P_N^{(2)})_{ij} &= S''_N(t_i - t_j). \end{aligned} \tag{4.33}$$

These spectral differentiation matrices will be applied to obtaining the curvature of contour which will be required in subsequent developments.

4.2 Boundary Integral Equation

The Boundary Integral Equation (BIE) method transforms a d dimensional problem to a $(d-1)$ dimensional problem, representing the solution by a $(d-1)$ integration. There are many different techniques which rely on BIE. In our thesis an indirect BIE technique using *single-layer potential* is applied to transform the 2-D Laplacian problem to the integration of some *density function* of a *single-layer potential* over the contour $\mathcal{C}$. Thus, by solving for the *density function*, the solution to Laplacian problem is recovered. In the following section, we will cover the details of the BIE we used here and how we apply the BIE to the numerical solution of $(\mathcal{D})$ and $(\mathcal{A})$.

Let us start with the underlining idea of indirect BIE method by *single-layer potential*. Suppose we have simple smooth and closed curve $S \in \mathbb{R}^2$ Denote D_- and D_+ as the interior and exterior of S. If we have a harmonic function Φ defined in $\mathbb{R}^2 \setminus S$:

$$-\Delta \Phi = 0 \qquad \mathbb{R}^2 \setminus S, \tag{4.34}$$

then by [19] there exists a function ρ defined on S such that Φ can be expressed as:

$$\Phi(\mathbf{x}) = \oint_S \alpha(\mathbf{x}, \mathbf{y}) \rho(\mathbf{y})\, d\mathbf{y} \qquad \mathbf{x} \in \mathbb{R}^2 \setminus S \tag{4.35}$$

This is called the single-layer potential. The $\rho(\mathbf{y})$ is known as *density function* and $\alpha(\mathbf{x}, \mathbf{y}) := -\frac{1}{2\pi} \ln |\mathbf{x} - \mathbf{y}|$ is the fundamental solution for (4.34) $\mathbb{R}^2$ [19]. In fact, for $\alpha(\mathbf{x}, \mathbf{y})$ we also have another choice known as: $-\frac{1}{2\pi} \frac{\partial}{\partial n} \ln |\mathbf{x} - \mathbf{y}|$ with which (4.35) is called double-layer potential. But in our paper we applied the single-layer approach as it leads to a simpler formulation. Moreover, we can extend Φ from $\mathbb{R}^2/S$ to S. Despite the singularity of $\Phi(\mathbf{x})$ as $\mathbf{x}$ approaches $\mathbf{y}$ on S, since $\lim_{\mathbf{x} \to \mathbf{y}} \ln |\mathbf{x} - \mathbf{y}| = \infty$, if we define Φ as in (4.35) for $\mathbf{x} \in S$, $\Phi(\mathbf{x})$ exists in Cauchy Principle Value sense and can

be given by:

$$\Phi(\mathbf{x}) = \lim_{\varepsilon \to 0^+} \int_{S/B(\mathbf{x},\varepsilon)} \alpha(\mathbf{x},\mathbf{y})\rho(\mathbf{y})\, d\mathbf{y} \qquad \mathbf{x} \in S \tag{4.36}$$

The proof of existence of $\Phi(\mathbf{x})$ on S can be found in [19]. Here we are more interested in how we can apply (4.35) to our problem. We will demonstrate some nice properties of single-layer potential Φ which will be used in numerical implementation.

Lemma 3. *The single-layer potential* Φ *defined as* (4.35) *over* S *is continuous in* $\mathbb{R}^2$ *and we have*

$$\lim_{\mathbf{x}\to\infty} \Phi(\mathbf{x}) = 0, \tag{4.37}$$

if and only if:

$$\oint_S \rho\, ds = 0. \tag{4.38}$$

This Lemma basically tells us that Φ vanishes at infinity if the average density ρ is 0. The proof can be found in [19]. The derivative of Φ is given by the following:

Lemma 4. *The gradient of* Φ *is:*

$$\nabla \Phi = -\frac{1}{2\pi} \oint_S \frac{\mathbf{x}-\mathbf{y}}{|\mathbf{x}-\mathbf{y}|^2} \rho(\mathbf{y})\, d\mathbf{s}_y. \tag{4.39}$$

This result followed by interchanging differentiation w. r. t. x with integration and taking the gradient of $\alpha(\mathbf{x},\mathbf{y})$ w. r. t. $\mathbf{x}$. However, because of the singularity of Φ on S its normal derivative on S is not so straightforward to express. According to [19], we have some results for $\frac{\partial \Phi_\pm}{\partial n}$

Lemma 5. *Let* S *is Hölder continuously differentiable and the density function* ρ *of single-layer potential* Φ *satisfies* $\rho \in L^\infty(S) \cap C(S)$*. For any* $\mathbf{x} \in S$*, if we denote the unit outer normal vector at* $\mathbf{x}$ *as* $\mathbf{n}(\mathbf{x})$*, then*

1. the one-sided normal derivative for Φ_- and Φ_+ exists at $\mathbf{x}_s$ and is given by:

$$\frac{\partial \Phi_{\pm}}{\partial n}(\mathbf{x}) := \lim_{\varepsilon \to 0^+} \mathbf{n}(\mathbf{x}) \cdot \nabla \Phi \left(\mathbf{x} \pm \varepsilon \, \mathbf{n}(\mathbf{x})\right), \tag{4.40}$$

2. the jump relation holds as:

$$\frac{\partial \Phi_+(\mathbf{x})}{\partial n} - \frac{\partial \Phi_-(\mathbf{x})}{\partial n} = -\rho(\mathbf{x}), \tag{4.41}$$

3. mean value of $\frac{\partial \Phi_+(\mathbf{x})}{\partial n}$ and $\frac{\partial \Phi_-(\mathbf{x})}{\partial n}$ reads:

$$\frac{1}{2}\left[\frac{\partial \Phi_+(\mathbf{x})}{\partial n} + \frac{\partial \Phi_-(\mathbf{x})}{\partial n}\right] = -\frac{1}{2\pi}\int_S \frac{< \mathbf{n}(\mathbf{x}), \mathbf{x}-\mathbf{y} >}{|\mathbf{x}-\mathbf{y}|^2}\rho\left(\mathbf{y}\right) d\mathbf{y}. \tag{4.42}$$

Moreover, the right hand-side of (4.42) is in fact not an improper integration. We have for its integrand [19]:

$$\lim_{\mathbf{y}\to\mathbf{x}} \frac{< \mathbf{n}(\mathbf{x}), \mathbf{x}-\mathbf{y} >}{|\mathbf{x}-\mathbf{y}|^2} = \frac{\kappa}{2} \quad \mathbf{x}, \mathbf{y} \in S \tag{4.43}$$

where κ is the signed curvature of S. Lemma 5 together with (4.43) provide us with specific techniques necessary to apply the BIE method to problems $(\mathcal{D})$ and $(\mathcal{A})$.

4.3 Numerical Implementation

Having reviewed basic material about spectral methods and BIE techniques, it is the time for us to move forward with the numerical implementation: setting up numerical scheme for our PDE systems $(\mathcal{D})$ and $(\mathcal{A})$ to get u and u^* and then the shape gradient $\boldsymbol{\nabla}\mathcal{J}$ defined in (3.67). It will be easy for us to first solve the direct problem $(\mathcal{D})$ numerically and then move on with the same strategy to $(\mathcal{A})$, since $(\mathcal{A})$ only differs

from $(\mathcal{D})$ in some terms of boundary conditions. Let

$$u = u_p + u_h \tag{4.44}$$

in which u_p and u_h satisfy the following system:

$$\begin{aligned}
-k\,\Delta\, u_p &= q(\mathbf{x}) && \text{in } \Omega, && (\mathcal{D}'\,1)\\
\Delta\, u_h &= 0 && \text{in } \Omega \setminus \mathcal{C}, && (\mathcal{D}'\,2)\\
u_h\Big|_1 &= u_h\Big|_2 && \text{on }, \mathcal{C} && (\mathcal{D}'\,3)\\
k\left(\frac{\partial u_h}{\partial n}\Big|_2 - \frac{\partial u_h}{\partial n}\Big|_1\right) &= \gamma\,(u - u_0) && \text{on}\,\mathcal{C}, && (\mathcal{D}'\,4)\\
\frac{\partial u_p}{\partial n} &= -\frac{\partial u_h}{\partial n} && \text{on}\,\partial\Omega, && (\mathcal{D}'\,5)
\end{aligned}$$

where $u_h|_1 = u_h\,\chi_{\,\Omega_1}$ and $u_h|_2 = u_h\,\chi_{\,\Omega_2}$. One can verify that $(\mathcal{D})$ is satisfied if we set $u_1 = (u_p + u_h)\,\chi_{\Omega_1}$ and $u_2 = (u_p + u_h)\,\chi_{\Omega_2}$. Thus by solving $(\mathcal{D}')$ for u_p and u_h we can recover u according to (4.44).

From the discussion of the boundary integral techniques in section 4.2, we records that if u_h is harmonic in $\Omega/\mathcal{C}$, it can be given by *single-layer potential*:

$$u_h(\mathbf{x}) = -\frac{1}{2\pi}\oint_{\mathcal{C}} \ln\,|\mathbf{x} - \mathbf{x}_{\mathcal{C}}|\,\mu(\mathbf{x}_{\mathcal{C}})\,d\mathbf{x}_{\mathcal{C}}, \tag{4.46}$$

in which μ is the *density function*. Considering (4.41) in *Lamma* 5 and boundary condition $(\mathcal{D}'\,4)$ the density $\mu(\mathbf{x})$ satisfies:

$$-\mu(\mathbf{x}) = \frac{\gamma}{k}\,(u - u_0(\mathbf{x})) \qquad \forall\,\mathbf{x} \in \mathcal{C}.$$

Next we plug (4.46) and (4.44) in to the above equation we have for μ

$$\mu(\mathbf{x}) - \frac{\gamma}{2\pi\, k} \oint_{\mathcal{C}} \ln\, |\mathbf{x} - \mathbf{x}_{\mathcal{C}}|\, \mu(\mathbf{x}_{\mathcal{C}}) d\, \mathbf{x}_{\mathcal{C}} = \frac{\gamma}{k}(u_0 - u_1(\mathbf{x})) \quad \forall\, \mathbf{x} \in \mathcal{C}.$$

Plugging (4.46) to ($\mathcal{D}'$ 5) and combine with (4.39):

$$\frac{\partial u_h}{\partial n} = -\frac{1}{2\pi} \oint_{\mathcal{C}} \frac{< \mathbf{n}, \mathbf{x} - \mathbf{x}_{\mathcal{C}} >}{|\mathbf{x} - \mathbf{x}_{\mathcal{C}}|^2} \rho(\mathbf{x}_{\mathcal{C}})\, d\mathbf{x}_{\mathcal{C}}. \tag{4.47}$$

So we are now at a position to transform ($\mathcal{D}'$) to a system in which $u_p(\mathbf{x})$ and $\mu(\mathbf{x})$:

$$k\, \Delta\, u_p = q(\mathbf{x}) \qquad \mathbf{x} \in \Omega, \tag{4.48a}$$

$$\mu(\mathbf{x}) + \frac{\gamma}{2\pi\, k} \oint_{\mathcal{C}} \ln\, |\mathbf{x} - \mathbf{x}_{\mathcal{C}}|\, \mu(\mathbf{x}_{\mathcal{C}})\, d\mathbf{x}_{\mathcal{C}} = \frac{\gamma}{k}(u_1(\mathbf{x}) - u_0) \;\; \forall\, \mathbf{x} \in \mathcal{C}, \tag{4.48b}$$

$$\frac{\partial u_p}{\partial n} = -\frac{\partial u_h}{\partial n} \qquad \mathbf{x} \in \partial\Omega. \tag{4.48c}$$

We notice that by *Lemma* 3 continuity of u_h across $\mathcal{C}$ is automatically guaranteed. we notice that the solutions u_p and μ (equivalently, u_h) are coupled through the RHS in (4.48b) and the boundary condition (4.48c). Our next step will be to build our numerical scheme for (4.48) using spectral methods.

We assume $\Omega = [-1, 1] \times [-1, 1]$. For some N let us build up a *tensor product grid* Tg with *Chebyshev points* in both x and y directions (this concept has been already talked about in section 4.1.1). As regards to the contour $\mathcal{C}$, we can parametrize $\mathcal{C}$ in terms of t by a map $\mathbf{z} : \mathbb{R} \to \mathbb{R}^2$ such that:

$$\mathbf{x} = \mathbf{z}(t) := [z_1(t), z_2(t)] \qquad \mathbf{x} \in \mathcal{C};\; t \in [0, 2\pi] \tag{4.49}$$

We can define $\mathbf{z}$ such that $\|\mathbf{z}'(t)\| = \frac{L}{2\pi}$ where L is the total length of $\mathcal{C}$. This parametrization can be achieved in two steps. First define a natural (or arc Length)

parametrization such that: $\Theta : s \in [0, L] \to \mathbf{x} \in \mathcal{C}$. Notice that by definition we have:

$$\text{length}(C|_{[s_1,s_2]}) = |s_1 - s_2|$$

which implies $\|\Theta'(s)\| = 1$. Secondly, define a function $s(t) = \frac{Lt}{2\pi},\ t \in [0, 2\pi]$ mapping $[0, 2\pi]$ to $[0, L]$. Then we have $\|\Theta'(t)\| = \|\Theta'(s)\|s'(t) = \frac{L}{2\pi}$. Then define $\mathbf{z} := \Theta \circ s$. For some n, let us discretize t into $2n$ equispaced points as $\mathbf{t} = [t_1, t_2, ..., t_{2n}]^T$. Thus $\mathcal{C}$ can be discretized as

$$\hat{\mathbf{z}} = [\mathbf{z}_1, \mathbf{z}_2, ...\mathbf{z}_{2n}]^T \tag{4.50}$$

with $\mathbf{z}_i := \mathbf{z}(t_i)$ the complex notation of points on $\mathcal{C}$. Notice that $|\mathbf{z}_j - \mathbf{z}_{j+1}| = constant$ by construction. Once these grids on Ω and $\mathcal{C}$ are constructed we can move on to discretize (4.48) and develop MATLAB code to solve it numerically.

4.3.1 Discretizing Equation (4.48a)

Denote u_p evaluated on *tensor product grid* Tg as a matrix U with elements $U_{ij} = u(x_i, y_j), 1 \leqslant i, j \leqslant N+1$. Now we can reorder U from the top right all the way down to the bottom left as a column vector $\tilde{U}$. This reordering is the same as what we did in the example of section 4.1.1 (see Figure 4.2). Also evaluate q on Tg and do the same reordering , denoting the resulting column vector as $\tilde{Q}$. Now by (4.18) the Laplacian in (4.48a) can be numerically represented in terms of the spectral differentiation matrix 4.18:

$$-k\Delta_N \cdot \tilde{U} = \tilde{Q}. \tag{4.51}$$

We mention that by different ordering of U the discrete Laplacian Δ_N can be different. One has to be ensure that the same ordering is used consistently in all numerical implementations.

4.3.2 Discretizing Equation (4.48b)

In the discretization of (4.48b) we have to be careful about the singularity in the line integral as $\mathbf{x}_{\mathcal{C}} \to \mathbf{x}$. Let us denote $\mathbf{x}$ and $\mathbf{x}_{\mathcal{C}}$ as $\mathbf{z}(t)$ and $\mathbf{z}(t')$ so that we have:

$$\begin{aligned}\ln|\mathbf{z}(t)-\mathbf{z}(t')| &= \frac{1}{2}\left\{\ln[\mathbf{z}(t)-\mathbf{z}(t')]^2 + \ln\left(4\sin^2\frac{t-t'}{2}\right) - \ln\left(4\sin^2\frac{t-t'}{2}\right)\right\} \\ &= \frac{1}{2}\ln\left\{\frac{[\mathbf{z}(t)-\mathbf{z}(t')]^2}{4\sin^2\frac{t-t'}{2}}\right\} + \frac{1}{2}\ln\left(4\sin^2\frac{t-t'}{2}\right)\end{aligned}$$

Now we plug the above equation into the integral in (4.48b) and transform integral domain to $[0, 2\pi]$:

$$\begin{aligned}\mu(t) &+ \overbrace{\frac{\gamma}{2\pi\,k}\int_0^{2\pi}\mu(t')\ln\left|\frac{\mathbf{z}(t)-\mathbf{z}(t')}{2\sin(\frac{t-t'}{2})}\right|\cdot J\cdot\,dt'}^{(I)} \\ &+ \overbrace{\frac{\gamma}{4\pi\,k}\int_0^{2\pi}\mu(t')\ln\left[4\sin^2(\frac{t-t'}{2})\right]\cdot J\cdot\,dt'}^{(II)} \\ &= \frac{\gamma}{k}[u_p(\mathbf{z}(t))-u_0]; \quad t,t'\in\mathcal{C},\end{aligned} \tag{4.52}$$

where $J = |\mathbf{z}'(t)| = \frac{L}{2\pi}$ and notice that μ is also a function of t. Now (I) has no singularity and (II) can be evaluated exactly using a theorem from [18]. Then we can discretize (I) and (II). Trapezoidal rule and trigonometric interpolation method are employed.

By the underlining grid $\mathbf{t}$ on $\mathcal{C}$ together with trapezoidal rule:

$$(I) = \frac{\gamma\,L}{4\pi\,k\,n}\sum_{j=0}^{2n-1}\mu_j\ln\left|\frac{\mathbf{z}_i-\mathbf{z}_j}{2\sin(\frac{t_i-t_j}{2})}\right|, \text{for each} \quad i\in 0,1,...2n \tag{4.53}$$

where $\mu_i := \mu(t_i)$ and $\mathbf{z}_i := \mathbf{z}(t_i)$ for $i = 1, 2, ..., 2n$. Suppose $\mathbb{K}_1$ is a $2n \times 2n$ matrix

such that:

$$[\mathbb{K}_1]_{ij} = -\frac{L}{4\pi\, n} \ln\left|\frac{\mathbf{z}_i - \mathbf{z}_j}{2\sin(\frac{t_i - t_j}{2})}\right|, \tag{4.54}$$

we notice that when $i = j$, $\mathbb{K}_1^{(ii)} = -\frac{L}{4\pi\, n} \ln|\mathbf{z}_i'|$. We can derive the matrix expression for (4.53):

$$(I) = \frac{\gamma}{k}\mathbb{K}_1 \cdot \tilde{\mu} \tag{4.55}$$

in which $\tilde{\mu} := [\mu_1, \mu_2, ..., \mu_{2n}]^T$. For (II), approximate $\mu(t')$ by trigonometric interpolation method [17]:

$$\mu(t') = \sum_{j=1}^{2n} \mu_j\, L_j(t') \tag{4.56}$$

where $L_j(t')$ are interpolation polynomials which is on a periodic domain defined as [17]:

$$L_j(t) = \frac{1}{2n} \sin\, n(t - t_j) \cot\frac{t - t_j}{2}, \quad t \neq\, t_j.$$

Then if we define $R_j^n(t)$ as:

$$R_j^n(t) = -\frac{1}{n}\left\{\sum_{m=1}^{n-1}\frac{1}{m}\cos\, m(t - t_j) + \frac{1}{2n}\cos\, n(t - t_j)\right\}, \tag{4.57}$$

(II) can be given by [17]:

$$(II) = \frac{\gamma\, L}{4\pi\, k}\sum_{j=1}^{2n} \mu_j R_j^n(t).$$

We define a $2n \times\, 2n$ matrix $\mathbb{K}_2$ as follows:

$$[\mathbb{K}_2]_{ij} = \frac{L}{4\pi}\, R_j^n(t_i), \tag{4.58}$$

then we can discretize (II) as:

$$(II) = \frac{\gamma}{k}\mathbb{K}_2 \cdot \tilde{\mu}. \tag{4.59}$$

As for the right hand side term of (4.52) we use a bicubic interpolation method to approximate $u_p(\mathbf{x})$ on the contour $\mathcal{C}$: $\exists$ a $2n\times(N+1)^2$ matrix $\mathbb{P}$ such that $u_p(\mathbf{z}) = \mathbb{P}\cdot\tilde{U}$. Together with (4.55) and (4.59) we can obtain the discretization for (4.48b) in the following form:

$$(\mathbb{I} + \frac{\gamma}{k}\mathbb{K}_1 + \frac{\gamma}{k}\mathbb{K}_2)\cdot\tilde{\mu} - \frac{\gamma}{k}\mathbb{P}\cdot\tilde{U} = \frac{\gamma}{k}\tilde{U}_0 \tag{4.60}$$

where $\tilde{U}_0$ is a column vector with all entities given by: u_0. Now we can combine (4.51) and (4.60) to obtain the matrix form of equation (4.48a) and (4.48b):

$$\begin{pmatrix} -\Delta_N & \mathbf{0} \\ -\frac{\gamma}{k}\mathbb{P} & \mathbb{I} + \frac{\gamma}{k}\mathbb{K}_1 + \frac{\gamma}{k}\mathbb{K}_2 \end{pmatrix} \cdot \begin{pmatrix} \tilde{U} \\ \vec{\mu} \end{pmatrix} = \begin{pmatrix} \frac{1}{k}\tilde{Q} \\ \frac{\gamma}{k}\tilde{U}_0 \end{pmatrix} \tag{4.61}$$

where $\mathbf{0}$ is a zero matrix. The next thing will be to enclose the boundary conditions on the above linear equations.

4.3.3 Discretization of Boundary Condition (4.48c)

Condition (4.48c) can be imposed on the boundary point by point following the same idea as we did in treating the Poisson equation with Neumann boundary conditions in section 4.1.1. Suppose we are at the boundary position $\mathbf{x}_0^c = (-1, y_0)$ which corresponds to the cth element in the rearranged vector $\tilde{U}$. Since we are at $x = -1$ we have $\mathbf{n} = (-1, 0)$:

$$\frac{\partial u_p}{\partial n} = -\frac{\partial u_p}{\partial x}$$

and by (4.17) in section (4.1.1) we have for the $\frac{\partial u_p}{\partial x}$

$$\tilde{U}_x = D_N^n \otimes I_{N+1}\cdot\tilde{U}, \tag{4.62}$$

where $\tilde{U}_x$ is a column vector of size $(N+1)^2$ containing the nodal values of $\frac{\partial u_p}{\partial x}$ evaluated at the Tg by the same ordering as $\tilde{U}$. For the u_h part in (4.48c), combining with (4.39)

in section (4.2) and $\mathbf{n} = (-1, 0)$:

$$-\frac{\partial u_h}{\partial n}\bigg|_{x_0^c} = \frac{1}{2\pi} \oint_S \frac{1 + \mathbf{x}_{\mathcal{C}}^{(1)}}{|\mathbf{x}_0^c - \mathbf{x}_{\mathcal{C}}|^2} \mu(\mathbf{x}_{\mathcal{C}})\, d\mathbf{x}_{\mathcal{C}}$$

by $(\mathbf{x}_0^c - \mathbf{x}_{\mathcal{C}}) \cdot \mathbf{n} = 1 + \mathbf{x}_{\mathcal{C}}^{(1)}$ where $\mathbf{x}_{\mathcal{C}}^{(1)}$ is the first component of $\mathbf{x}_{\mathcal{C}}$. We transform the above integral to $[0, 2\pi]$ and apply trapezoidal rule on the grid $\mathbf{t}$:

$$\begin{aligned} -\frac{\partial u_h}{\partial n}\bigg|_{x_0^i} &\approx \frac{L}{4\pi\, n} \sum_{j=1}^{2n} \frac{1 + \mathbf{z}_j^{(1)}}{|\mathbf{x}_0 - \mathbf{z}_j|} \mu_j \\ &= \mathbf{b}^{(c)} \cdot \tilde{\mu}, \end{aligned} \tag{4.63}$$

where $\mathbf{z}_j^{(1)}$ is the first component of $\mathbf{z}_j$ and $\mathbf{b}^{(c)}$ is a row vector with entities given by

$$\mathbf{b}_j^{(c)} = \frac{L}{4\pi\, n} \frac{1 + \mathbf{z}_j^{(1)}}{|\mathbf{x}_0^c - \mathbf{z}_j|}, \qquad 1 \leqslant j \leqslant 2n$$

Now by (4.62) and (4.63) at the cth position we have:

$$-(D_N^n \otimes I_{N+1})_c \cdot \tilde{U} + \mathbf{b}^{(c)} \cdot \tilde{\mu} = 0 \tag{4.64}$$

where $-(D_N^n \otimes I_{N+1})_c$ is the cth row of $D_N^n \otimes I_{N+1}$. To impose the boundary condition at $\mathbf{x}_0^c$ we just have to replace the cth row of Δ_N in (4.61) with $-(D_N^n \otimes I_{N+1})_c$ and the cth row of zero matrix $\mathbf{0}$ as $\mathbf{b}^{(c)}$. For points other than $\mathbf{x}_0^c$ we can impose the corresponding boundary condition in the same way and finally our resulting numerical scheme for (4.48) is:

$$\begin{pmatrix} -\Delta_N' & \mathbb{B} \\ -\frac{\gamma}{k}\mathbb{P} & \mathbb{I} + \frac{\gamma}{k}\mathbb{K}_1 + \frac{\gamma}{k}\mathbb{K}_2 \end{pmatrix} \cdot \begin{pmatrix} \tilde{U} \\ \tilde{\mu} \end{pmatrix} = \begin{pmatrix} \frac{1}{k}\tilde{Q}' \\ \frac{\gamma}{k}\tilde{U}_0 \end{pmatrix} \tag{4.65}$$

in which Δ'_N, $\mathbb{B}$ and $\tilde{Q}'$ are Δ_N, $\mathbf{0}$ and $\tilde{Q}$ in ((4.61)) respectively after imposing the boundary conditions. Thus we can solve for $\tilde{U}$ and $\tilde{\mu}$ and then reconstruct our solution u_h by (4.46) using trapezoidal rule and u by $u = u_p + u_h$.

This is how we solve for $(\mathcal{D})$, and for the adjoint system $(\mathcal{A})$ we apply the same strategy but with different right-hand side:

$$\begin{pmatrix} -\Delta'_N & \mathbb{B} \\ -\frac{\gamma}{k}\mathbb{P} & \mathbb{I} + \frac{\gamma}{k}\mathbb{K}_1 + \frac{\gamma}{k}\mathbb{K}_2 \end{pmatrix} \cdot \begin{pmatrix} \tilde{U^*} \\ \tilde{\mu^*} \end{pmatrix} = \begin{pmatrix} \frac{1}{k}\tilde{Q}^* \\ 0 \end{pmatrix} \tag{4.66}$$

where $\tilde{U^*}$ and $\tilde{\mu^*}$ are u_p^* and μ^* for the adjoint PDE $(\mathcal{A})$ respectively, $\tilde{Q}^*$ is the reordering of $(u - \overline{u})\chi_{A_i}$ $i = 1, 2$ evaluated on the grid Tg after imposing boundary conditions of $(\mathcal{A})$, u is the solution of the PDE system $(\mathcal{D})$.

4.3.4 Normal Derivatives on $\mathcal{C}$ and Curvature

The outer normal derivatives $\frac{\partial u_1}{\partial n}$ and $\frac{\partial u_1^*}{\partial n}$ are applied in representing shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ in (3.67) together with $\boldsymbol{\nabla}^{L^2}\mathcal{J}_L$ in (3.68). Here we give a brief idea on their derivation. According to (4.41) and (4.42) for $\frac{\partial u_h^-}{\partial n}$ and $\frac{\partial u_h^+}{\partial n}$ on $\mathcal{C}$ we have:

$$\frac{\partial u_h^+(\mathbf{x})}{\partial n} - \frac{\partial u_h^-(\mathbf{x})}{\partial n} = -\mu(\mathbf{x}) \tag{4.67a}$$

$$\frac{1}{2}\left[\frac{\partial u_h^+(\mathbf{x})}{\partial n} + \frac{\partial u_h^-(\mathbf{x})}{\partial n}\right] = -\frac{1}{2\pi}\int_{\mathcal{C}} \frac{< \mathbf{n}(\mathbf{x}), \mathbf{x} - \mathbf{x}_{\mathcal{C}} >}{|\mathbf{x} - \mathbf{x}_{\mathcal{C}}|^2}\mu(\mathbf{x}_{\mathcal{C}})\, d\mathbf{x}_{\mathcal{C}} \tag{4.67b}$$

which can be solved for $\frac{\partial u_h^-}{\partial n}$ and $\frac{\partial u_h^+}{\partial n}$. Then $\frac{\partial u_p}{\partial n}$ on $\mathcal{C}$ can be evaluated through a spline interpolation of ∇u_p on *tensor product grid* (Tg). In detail, by section (4.1.1) we have $(\tilde{U}_x, \tilde{U}_y) \approx (I_{N+1} \otimes D_N \tilde{U}, D_N \otimes I_{N+1}\tilde{U})$. A spline interpolation will give us ∇u_p on

$\mathcal{C}$, then $\frac{\partial u_p}{\partial n} = \nabla u_p \cdot n$. Then $\frac{\partial u_1}{\partial n}$ and $\frac{\partial u_2}{\partial n}$ can be recovered as:

$$\begin{aligned} \frac{\partial u_1}{\partial n} &= \frac{\partial u_h^-}{\partial n} + \frac{\partial u_p}{\partial n}, \\ \frac{\partial u_2}{\partial n} &= \frac{\partial u_h^+}{\partial n} + \frac{\partial u_p}{\partial n}, \end{aligned} \tag{4.68}$$

noticing that $u_p \in C^2(\Omega)$ so that $\frac{\partial u_p^+}{\partial n} = \frac{\partial u_p^-}{\partial n}$. For the evaluation of $\frac{\partial u_1^*}{\partial n}$ and $\frac{\partial u_2^*}{\partial n}$ it follows the same idea.

The signed curvature κ is given by:

$$\kappa = \frac{z_1' z_2'' - z_2' z_1''}{(z_1'^2 + z_2'^2)^{\frac{3}{2}}} \tag{4.69}$$

if we parametrize $\mathcal{C}$ as $(z_1(t), z_2(t))$. All the derivatives on $\mathcal{C}$ are computed by spectral method on periodic grid (see section 4.1.3).

4.3.5 Sobolev Gradients and Evaluation of Cost Functional

After solving the PDE system $(\mathcal{D})$ and adjoint PDE system $(\mathcal{A})$, and then obtaining the normal derivative $\frac{\partial u_1}{\partial n}$ and $\frac{\partial u_1^*}{\partial n}$, we can now give $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ on $\mathcal{C}$ according to (3.67). Numerically we have a set of nodal values for $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ defined on the discrete grid of $\mathcal{C}$ (4.50). But this L^2 gradient does not guarantee any thing about smoothness. Through preconditioning, we can obtain an H^1 gradient (see section 3.3.4) $\boldsymbol{\nabla}^{H^1}\mathcal{J}$ in Sobolev space which satisfies the following PDE:

$$(1 - l^2\,\partial_s^2)\,\boldsymbol{\nabla}^{H^1}\mathcal{J} = \boldsymbol{\nabla}^{L^2}\mathcal{J} \qquad s \in [0, L], \tag{4.70a}$$

$$\text{Periodic Boundary Conditions}, \tag{4.70b}$$

where l is called a Sobolev coefficient here which controls the smooothness of $\boldsymbol{\nabla}^{H^1}\mathcal{J}$, s is the curvilinear coordinate and L is the total length of contour $\mathcal{C}$. For (4.70) with

periodic boundary conditions, its solution can be given with high accuracy by Fourier analysis. We first perform the Fourier transformation on both sides of (4.70a) then we have:

$$(1 - l^2\xi^2)\widehat{\boldsymbol{\nabla}^{H^1}\mathcal{J}} = \widehat{\boldsymbol{\nabla}^{L^2}\mathcal{J}} \tag{4.71}$$

in which ξ is the frequency or wavenumber, $\widehat{\boldsymbol{\nabla}^{H^1}\mathcal{J}}$ and $\widehat{\boldsymbol{\nabla}^{L^2}\mathcal{J}}$ are the Fourier transformation of $\boldsymbol{\nabla}^{H^1}\mathcal{J}$ and $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ respectively. Then

$$\widehat{\boldsymbol{\nabla}^{H^1}\mathcal{J}} = \widehat{\boldsymbol{\nabla}^{L^2}\mathcal{J}}/(1 - l^2\xi^2).$$

Here $(1 - l^2\xi^2)$ is the preconditioner. At last an inverse Fourier transformation is performed on $\widehat{\boldsymbol{\nabla}^{H^1}\mathcal{J}}$ to recover $\boldsymbol{\nabla}^{H^1}\mathcal{J}$. In the numerical implementation FFT is applied for its good balance between accuracy and efficiency. The Sobolev gradient for $\boldsymbol{\nabla}^{H^1}\mathcal{J}_L$ is obtained in the same way.

For the cost functional defined as (2.5) the Clenshaw-Curtis Quadrature is applied. In detail, since it is a 2D integration, we first do the integration in one direction and then integrate in the other direction. With each direction we applied the Clenshaw-Curtis Quadrature.

4.4 Optimization Algorithm

4.4.1 Optimal Step σ_m

The contour $\mathcal{C}^{(m+1)}$ at step $(m+1)$ is upgraded based on $\mathcal{C}^{(m)}$, shape gradient $\boldsymbol{\nabla}\mathcal{J}(\mathcal{C}^{(m)})$ and optimal step σ_m such that:

$$\mathcal{C}^{(m+1)} = \mathcal{C}^{(m)} - \sigma_m \boldsymbol{\nabla}\mathcal{J}(\mathcal{C}^{(m)}) \qquad m = 0, 1, ..., \tag{4.72}$$

where the optimal step σ_m is computed via line–minimization approach (Brent's method). Basically speaking σ_m is chosen so that we ensure a cost functional $\mathcal{J}(\mathcal{C}^{(m+1)})$ for the next step as small as possible:

$$\sigma_m = \text{argmin}_{\sigma>0}\{\mathcal{J}(\mathcal{C}^{(m)} - \sigma\, \boldsymbol{\nabla}\mathcal{J}(\mathcal{C}^{(m)})\}. \tag{4.73}$$

Note that $\boldsymbol{\nabla}\mathcal{J}(\mathcal{C}^{(m)})$ can be replaced by the conjugate gradient.

The Brent's method is a 1D minimization algorithm aiming at finding the local minima or maxima of a function. It combines the golden section search method and polynomial interpolation. If the function is parabolic near its minima, we evaluate the function at three points and then interpolate the function using a parabola. The next-stage bracket point is approximated by the minima of the parabolic interpolant. Otherwise, if the function is not parabolic, we will perform the golden section search. Thus Brent's method is efficient for parabolic type function. For our problem we can take a wild guess that the cost functional on stage $(m+1)$ which is given by:

$$\mathcal{J}(\mathcal{C}^{(m+1)}) := \mathcal{J}(\mathcal{C}^{(m)}, \sigma_m) \tag{4.74}$$

has a parabolic structure in σ_m if $\mathcal{C}^{(m)}$ is fixed, since by intuition too large or too small step σ_m will either result in little change in the functional $\mathcal{J}$ or an increase in $\mathcal{J}$. Thus it is reasonable to apply the Brent's method in the evaluation of σ_m.

4.4.2 Optimization Algorithm

Now we can propose our optimization algorithm where Polak-Ribière conjugate gradient [10] is applied. In order to ensure us in the descending direction we use the descent gradient instead of the conjugate gradient every tenth step. After each update on contour $\mathcal{C}$ we discretize it again to ensure the points are equispaced. Figure 4.3 is

the flow-chart where TOL is the tolerant number we applied for the stopping criteria.

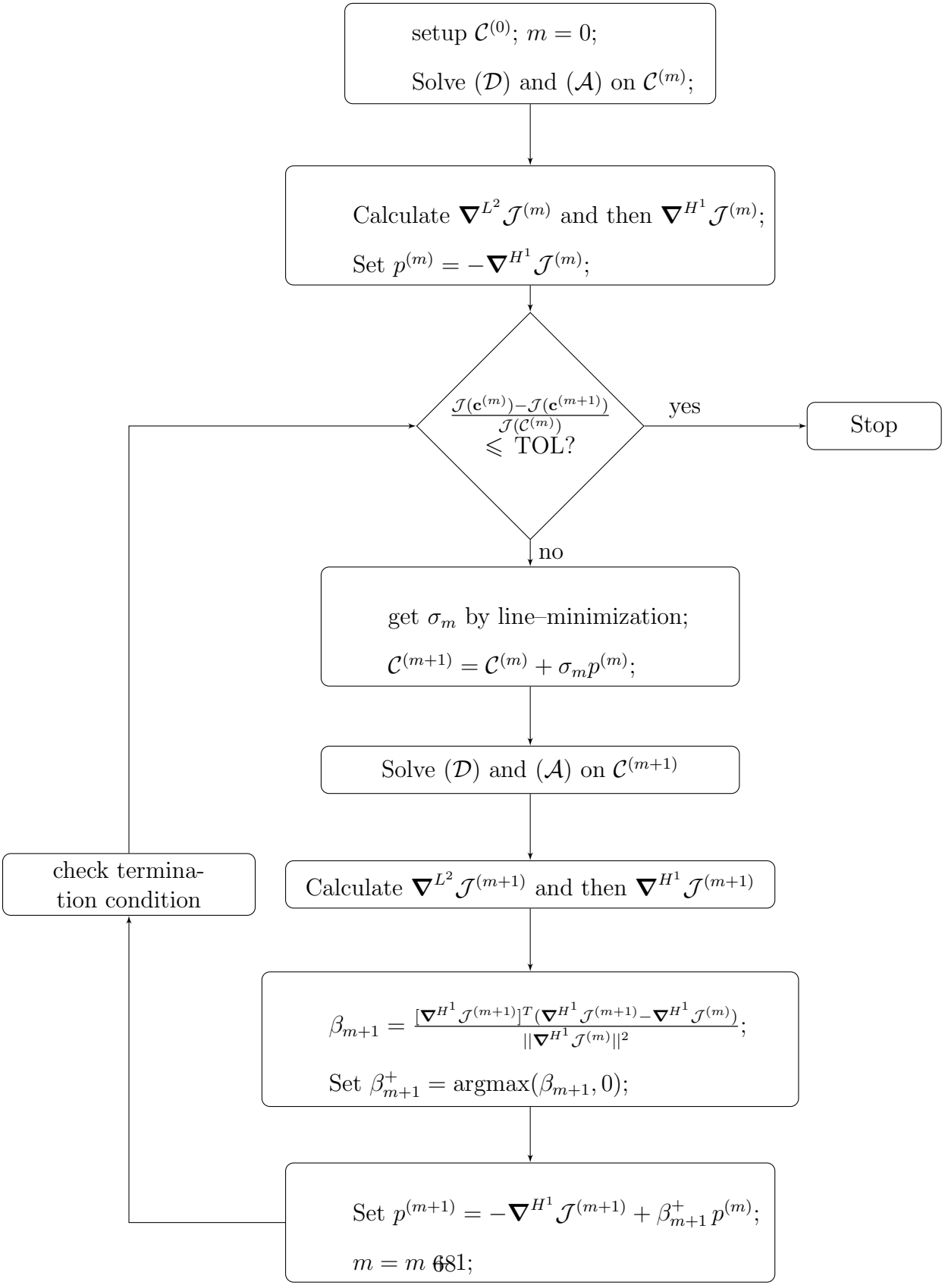


Figure 4.3: Flow-chart of optimization algorithm

Chapter 5

Results

In this chapter we will discuss the results from the numerical scheme. First we will talk about the validation of our numerical solver for PDE systems $(\mathcal{D})$ and $(\mathcal{A})$. Secondly, a κ-test aiming to validate our L^2 shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ is performed. Last, we will show some sample results of our optimization algorithm for different settings. Here for all the results, we set $\Omega = [-1,1] \times [-1,1]$, $k = 1$, $\gamma = 5$, $u_0 = 10$, $2n$ equispaced points on the contour and Chebyshev grid $(N+1)^2$.

5.1 Validation of Solvers for the Direct Problem $(\mathcal{D})$ and Adjoint Problem $(\mathcal{A})$

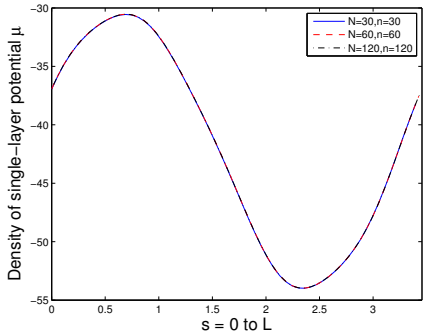
After setting up the numerical scheme for PDE $(\mathcal{D})$ and $(\mathcal{A})$, it is time for us to run a couple of tests to see if our numerical method is valid. We plotted intermediate results μ, u_p and u_h for different resolutions and fixed q, $\overline{u}$. Here are the settings: heat sources

and desired temperature distribution are:

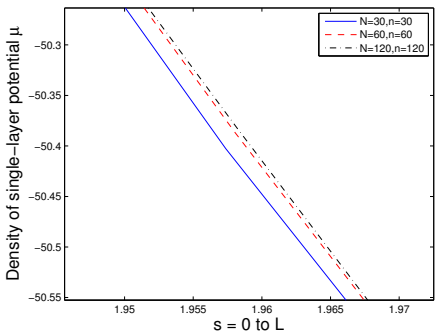
$$q = 50 - 15x^2 - 15(y - \frac{1}{2})^2 \quad (x, y) \in \Omega,$$
$$\overline{u} = 15 + \sin(4x - 1)\cos(4y - 1) \quad (x, y) \in \Omega,$$

and our contour is a circle centred at $(0.1, 0.1)$ with a radius 0.55 thus a total length of $L \approx 3.454$. We plot μ versus the arc length coordinate s, u_p for $y = 0$, u_h for $y = 0$ and u for $y = 0$. Different resolutions $(n = 30,\ N = 30)$, $(n = 60,\ N = 60)$ and $(n = 120,\ N = 120)$ are applied for each plot. Density function μ and u_p are solved based on equation (4.65), u_h is solved according to (4.46) and $u = u_p + u_h$. The results are shown in Figure 5.1. Figures (a), (c), (e) and (g) of 5.1 demonstrate the density function μ, u_p at $y = 0$, u_h at $y = 0$ and temperature u at $y = 0$ respectively, for different resolutions. In addition, to see the convergence property we also plot the zoomed-in data for μ, u_p at $y = 0$, u_h at $y = 0$ and u at $y = 0$ as in Figures (b), (d), (f) and (h). We can see as we increase the resolution, the result for μ, u_p, u_h and u converges, thus we have confidence to say that the numerical scheme we set up for $(\mathcal{D})$ is valid.

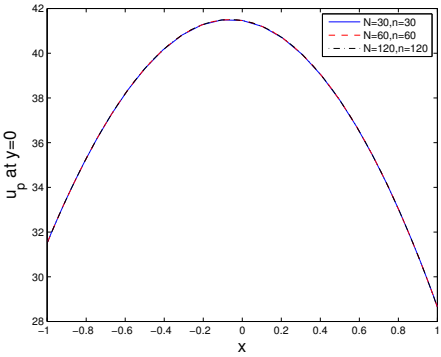
For the PDE $(\mathcal{A})$ we set $A = \Omega$ and use the solution u we got from PDE $(\mathcal{D})$, then μ^*, u_p^*, u_h^* and u^* for adjoint system $(\mathcal{A})$ are shown in Figure 5.2. Figures (a), (c), (e) and (g) in 5.2 are μ^*, u_p^* at $y = 0$, u_h^* at $y = 0$ and u^* at $y = 0$ respectively. While Figures (b), (d), (f) and (h) contains the closing view of μ^*, u_p^* at $y = 0$, u_h^* at $y = 0$ and u^* at $y = 0$. It also shows a convergence behaviour for the solutions of adjoint PDE $(\mathcal{A})$ as we increase the resolution.



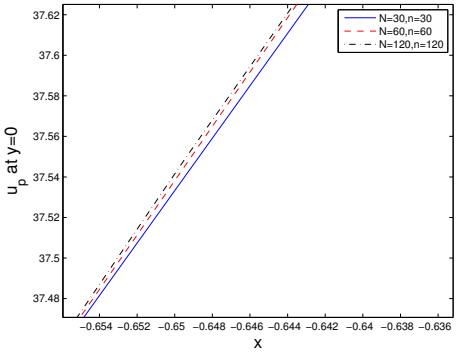
(a) Density of single-layer potential μ



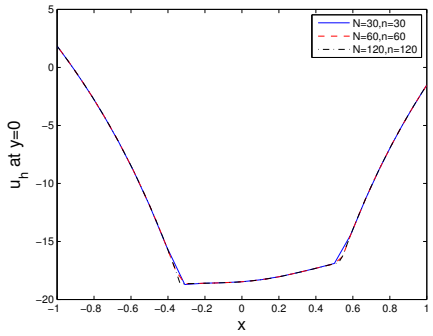
(b) Closing view of density function μ



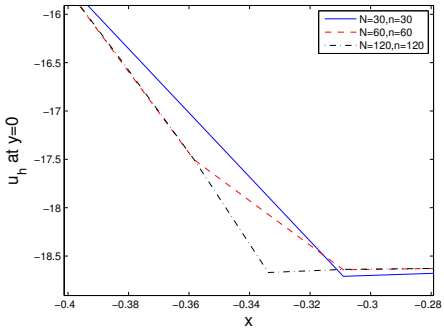
(c) Solution of u_p for $y = 0$



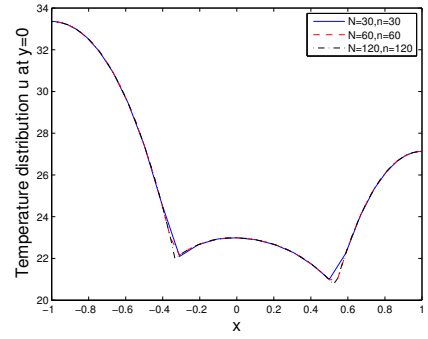
(d) Closing view of solution u_p for $y = 0$



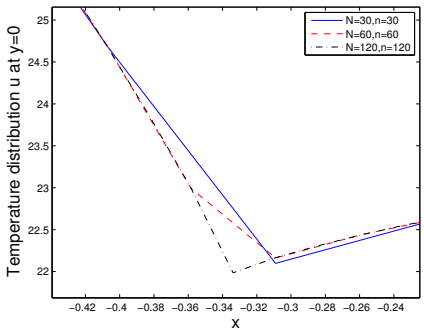
(e) Solution of u_h for $y = 0$



(f) Closing view of solution u_h for $y = 0$

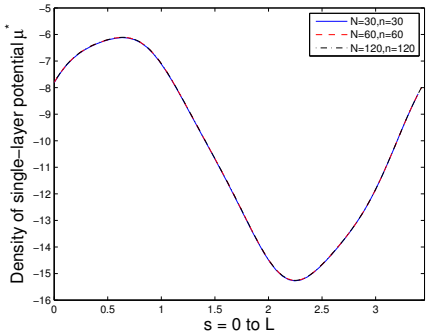


(g) Temperature distribution u for $y = 0$

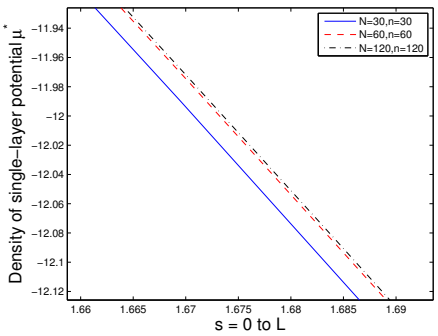


(h) Closing view of u for $y = 0$

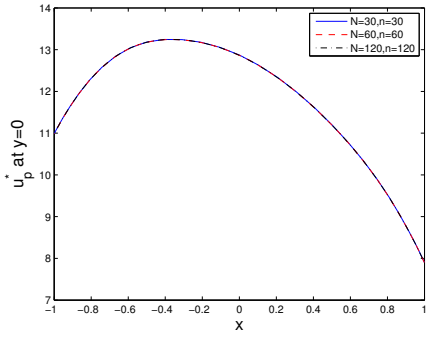
Figure 5.1: Solution μ, u_p, u_h and u from PDE system $(\mathcal{D})$ obtained using different resolutions indicated on insets.



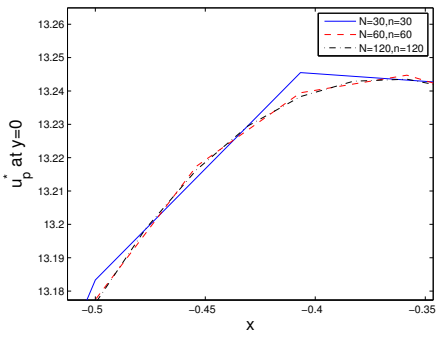
(a) Adjoint density of single-layer potential μ^*



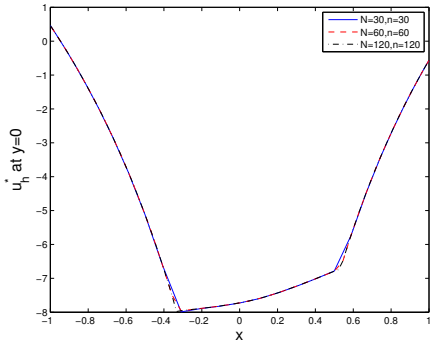
(b) Closing view of density function μ^*



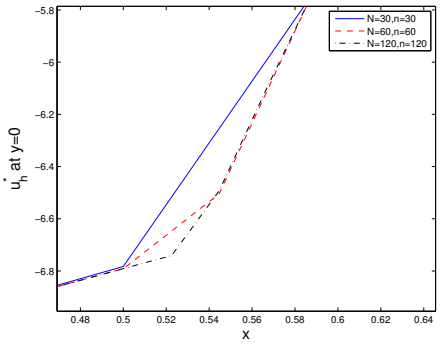
(c) Adjoint solution of u_p^* for $y = 0$



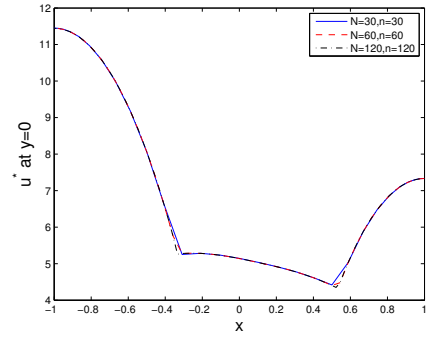
(d) Closing view of solution u_p^* for $y = 0$



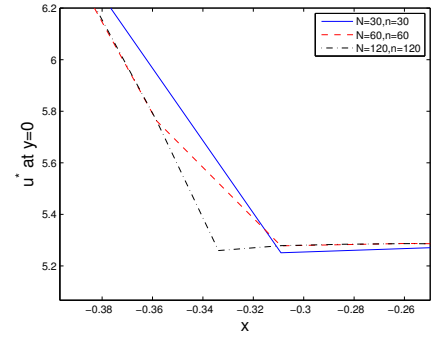
(e) Adjoint solution of u_h^* for $y = 0$



(f) Closing view of solution u_h^* for $y = 0$



(g) Adjoint solution of u^* for $y = 0$



(h) Closing view of u^* for $y = 0$

Figure 5.2: μ^*, u_p^*, u_h^* and u^* from PDE system ($\mathcal{A}$) obtained using different resolutions indicated on insets.

5.2 Validation of the Cost Functional Gradient

In this section we will validate our L^2 shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ and $\boldsymbol{\nabla}^{L^2}\mathcal{J}_L$ using the so-called κ-test. We have already derived formula (3.67) for $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ (see Chapter 3) as an expression in terms of u and u^* which can be obtained by solving the direct PDE ($\mathcal{D}$) and adjoint PDE ($\mathcal{A}$). By Riesz Representation theorem [13], for any perturbation ζ on the contour $\mathcal{C}$:

$$\mathcal{J}'(\mathcal{C};\zeta) = \int_{\mathcal{C}} \boldsymbol{\nabla}^{L^2}\mathcal{J}\,\zeta\,d\mathcal{C}, \tag{5.1}$$

where on the right-hand side is the directional derivative w. r. t. shape $\mathcal{C}$ in the direction ζ. In addition, by ε-definition

$$\mathcal{J}'(\mathcal{C};\zeta) = \lim_{\varepsilon\to 0}\frac{\mathcal{J}(\mathcal{C}+\varepsilon\,\zeta)-\mathcal{J}(\mathcal{C})}{\varepsilon}.$$

Define

$$\mathcal{J}'_\varepsilon(\mathcal{C};\zeta) := \frac{\mathcal{J}(\mathcal{C}+\varepsilon\,\zeta)-\mathcal{J}(\mathcal{C})}{\varepsilon}, \tag{5.2}$$

so that the κ-test can be constructed as follows:

$$\kappa(\varepsilon) := \frac{\mathcal{J}'_\varepsilon(\mathcal{C};\zeta)}{\int_{\mathcal{C}}\boldsymbol{\nabla}^{L^2}\mathcal{J}\,\zeta\,d\mathcal{C}}. \tag{5.3}$$

Then we build up a set of ε with an appropriate range and plot $\kappa(\varepsilon)$ for every ε. Since for extremely small ε the round-off error occurs and for large ε, $\mathcal{J}'_\varepsilon(\mathcal{C};\zeta)$ is a poor estimator for $\mathcal{J}'(\mathcal{C};\zeta)$, if our $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ is evaluated correctly, then for the plot we will have $\kappa \approx 1$ for intermediate values of ε and κ deviating from 1 for small and large ε. Thus, we can test if we have a valid result for shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$. In the following we will conduct five κ-tests for different cases shown in Table 5.1

	Cost gradient	Contour $\mathcal{C}$	Perturbations	Resolution $(N,\ n)$	Domain
κ-test 1	$\boldsymbol{\nabla}\mathcal{J}$	$\mathcal{C}_0$	ζ_1, ζ_2, ζ_3 and ζ_4	(50, 50), (100, 100), (80, 300)	Ω
κ-test 2	$\boldsymbol{\nabla}\mathcal{J}$	$\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$	ζ_1	(50, 50), (80, 100), (80, 200), (80, 300), (80, 400)	Ω
κ-test 3	$\boldsymbol{\nabla}^{L^2}\mathcal{J}$	$\mathcal{C}_0$	ζ_1, ζ_2, ζ_3 and ζ_4	(50, 50), (100, 100), (80, 300)	Ω
κ-test 4	$\boldsymbol{\nabla}^{L^2}\mathcal{J}$	$\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$	ζ_1	(50, 50), (80, 100), (80, 200), (80, 300), (80, 400)	Ω
κ-test 5	$\boldsymbol{\nabla}\mathcal{J}$	$\mathcal{C}_0$	ζ_1	(50, 50), (80, 100), (80, 200), (80, 300), (80, 400)	A

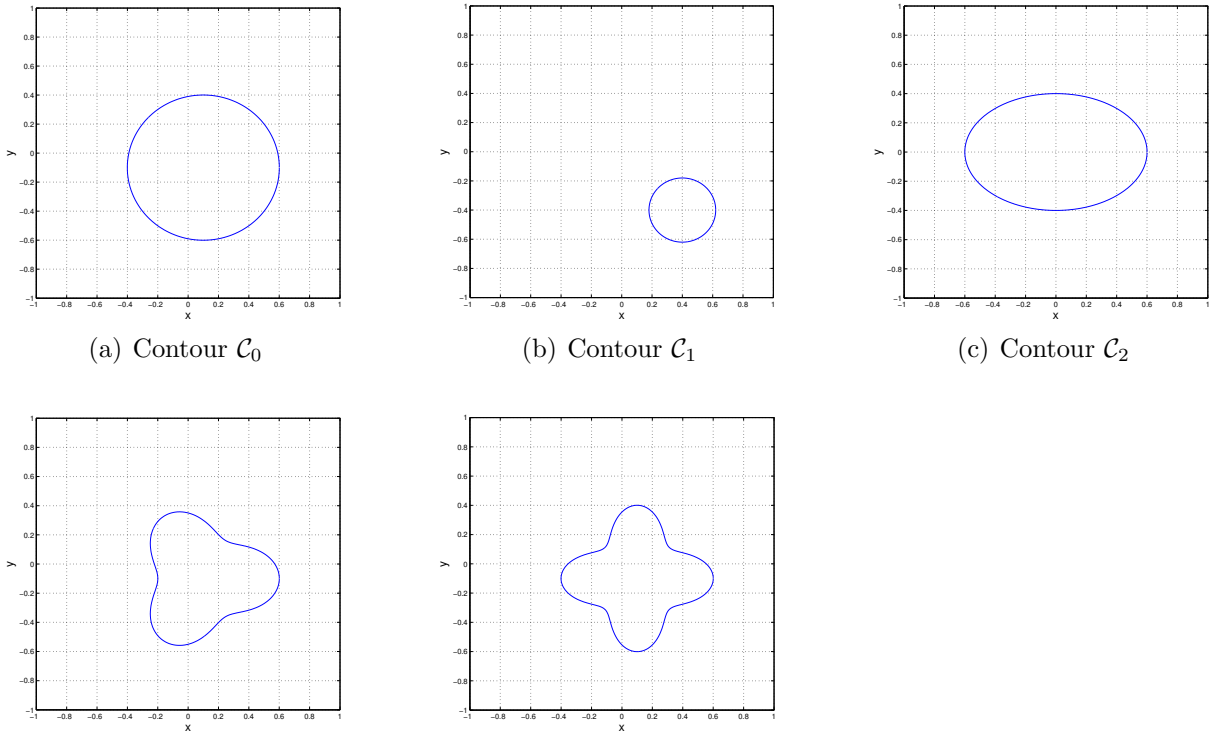
Table 5.1: Settings for different κ-tests

Here are a few comments about Table 5.1. The first column is the shape gradient we want to approximate. The second column is the cooling contour we applied with different contours $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ shown in Figure 5.3. where $\mathcal{C}_0$ is a circle of radius 0.4 centred at $(0.1, -0.1)$, $\mathcal{C}_1$ is a circle of radius 0.2 centred at $(0.4, -0.4)$, $\mathcal{C}_2$ is an ellipse with major diameter of 0.6 and a minor diameter of 0.4 centred at $(0, 0)$, $\mathcal{C}_3$ is a shape parametrized by:

$$x(t) = 0.4(1 + 0.1\cos(3t))\cos(t) + 0.1,$$
$$y(t) = 0.4(1 + 0.1\cos(3t))\sin(t) + 0.1,$$

and $\mathcal{C}_4$ is parametrized by:

$$x(t) = 0.4(1 + 0.1\cos(4t))\cos(t) + 0.1,$$
$$y(t) = 0.4(1 + 0.1\cos(4t))\sin(t) + 0.1.$$



(a) Contour $\mathcal{C}_0$ (b) Contour $\mathcal{C}_1$ (c) Contour $\mathcal{C}_2$

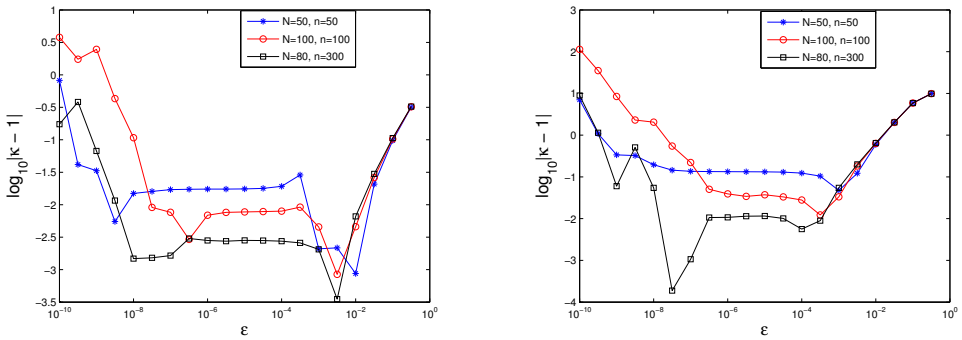
(d) Contour $\mathcal{C}_3$ (e) Contour $\mathcal{C}_4$

Figure 5.3: Five different initial contours applied in the κ-test.

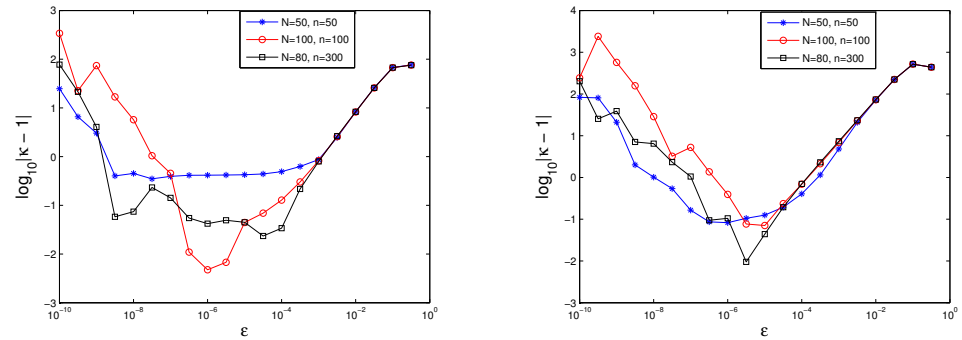
The third column is all the perturbations applied with

$$\zeta_j(t) = \sin(jt) \qquad j \in [1, 2, 3, 4]. \tag{5.4}$$

The forth column is the resolutions applied for different cases where we use a *tensor produc grid* of size $(N+1)^2$ for the domain and $2n$ equispaced points for the contour. The fifth column is the subdomain of our interest where $\Omega = [-1, 1] \times [-1, 1]$ and $A = [-0.5, 1] \times [-1, 0]$. Instead plotting $\kappa(\varepsilon)$ directly we use the quatity: $\log_{10} |\kappa - 1|$ from which accuracy of the κ-test can be determined in terms of significant digits. For



(a) Perturbation ζ_1 (b) Perturbation ζ_2



(c) Perturbation ζ_3 (d) Perturbation ζ_4

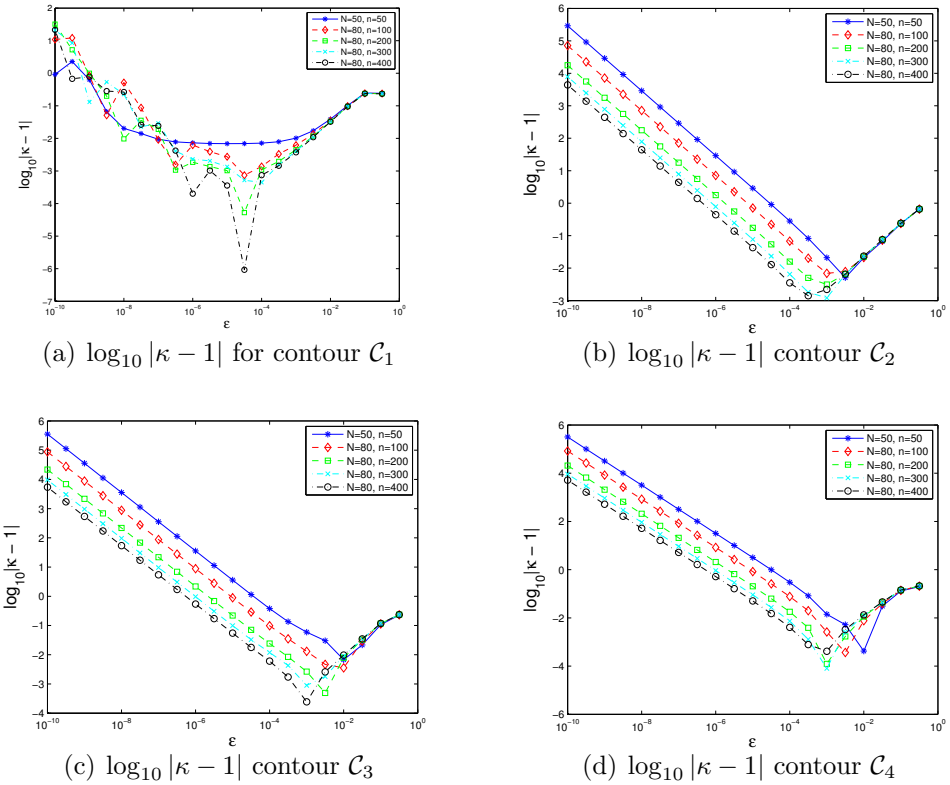
Figure 5.4: κ-test 1, $\log_{10}|\kappa - 1|$ for $\boldsymbol{\nabla}\mathcal{J}$ with 4 different perturbations.

all the tests the heat sources and desired temperature are given by:

$$q(x, y) = 50 - 15x^2 - 15 * (y - 0.5)^2$$
$$\overline{u}(x, y) = 15 + \sin(4x - 1)\cos(4y - 1)$$

Result of κ-test 1 are shown in Figure 5.4. We can see for different perturbation our κ stay very close to 1 and thus the L^2 shape gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ is evaluated accurately. Moreover, as we increase the resolution, we have a better estimate for $\boldsymbol{\nabla}^{L^2}\mathcal{J}$.

Results of κ-test 2 are shown in Figure 5.5. We can see that κ is closed to 1 in some region for all the cases and deviate from 1 for either too large or too small ε. In addition, for some cases we can have a very high accuracy (up to 4 digits) for the



(a) $\log_{10}|\kappa - 1|$ for contour $\mathcal{C}_1$

(b) $\log_{10}|\kappa - 1|$ contour $\mathcal{C}_2$

(c) $\log_{10}|\kappa - 1|$ contour $\mathcal{C}_3$

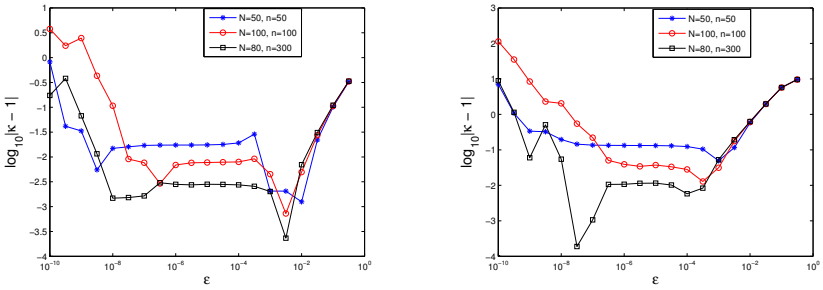
(d) $\log_{10}|\kappa - 1|$ contour $\mathcal{C}_4$

Figure 5.5: κ-test 2, $\log_{10}|\kappa - 1|$ for 4 different initial contours.

L^2 shape gradient as it is show in Figure 5.5(a) and the accuracy increases as we use more grid points in the discretization of coils.

After adding the length constraint, we perform the same κ-test for $\boldsymbol{\nabla}^{L^2}\mathcal{J}_L$ as we did for $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ in κ-test 1 and κ-test 2. The results of these tests also show the validation of $\boldsymbol{\nabla}^{L^2}\mathcal{J}_L$ (see the following κ-test 3 and 4). Results for κ-test 3 are in Figure 5.6 and results for κ-test 4 are in Figure 5.7.

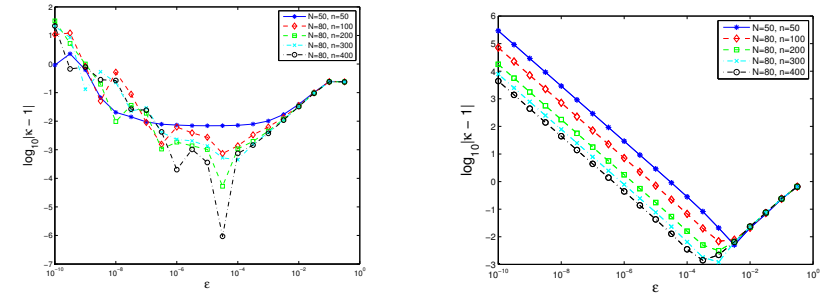
Now we set our subdomain $A = [-0.5, 1] \times [-1, 0]$, and the remaining settings are the same as κ-test 1. We can see that κ-test succeed for the subdomain A as it is showed in the Figure 5.8. Finally we conclude that the validation tests exhibit the anticipated results with refinement of resolution resulting in increased accuracy of the gradient.



(a) Perturbation ζ_1 (b) Perturbation ζ_2

(c) Perturbation ζ_3 (d) Perturbation ζ_4

Figure 5.6: κ-test 3, $\log_{10}|\kappa - 1|$ for $\boldsymbol{\nabla}\mathcal{J}_L$ with 4 different perturbations.



(a) Contour $\mathcal{C}_1$ (b) Contour ζ_2

(c) Contour ζ_3 (d) Contour ζ_4

Figure 5.7: κ-test 4, $\log_{10}|\kappa - 1|$ for $\boldsymbol{\nabla}\mathcal{J}_L$ with 4 different initial contours.

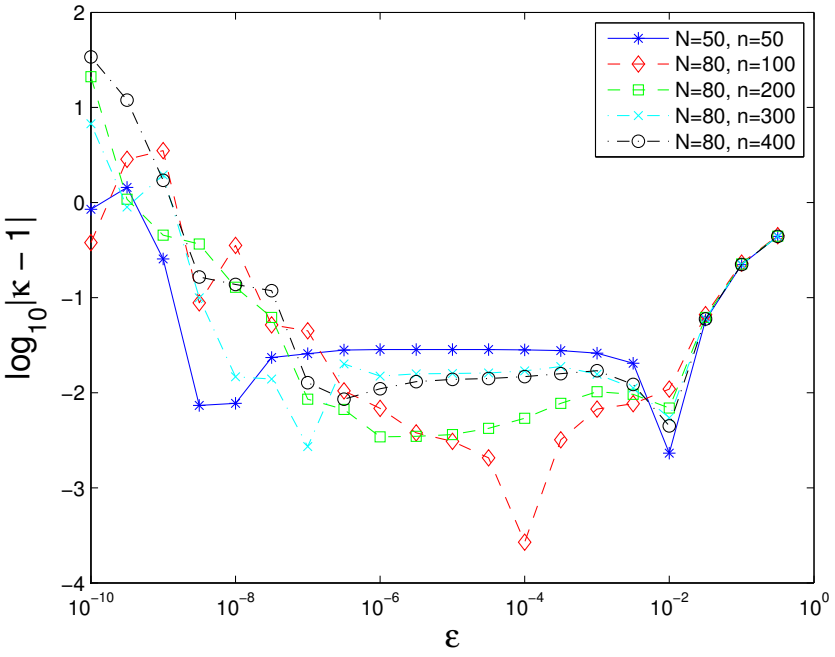


Figure 5.8: κ-test 5 for subdomain A; ζ_1 and $\mathcal{C}_0$ is applied

5.3 Optimization Results

The success of the κ-test provides us with the confidence in the gradients $\boldsymbol{\nabla}\mathcal{J}$, thus we could move forward to implement our optimization algorithm. In this section, we tested several cases with a variety of heat sources q and desired temperatures $\overline{u}$, both without and with the length constraint. Table 5.2 shows the settings for different cases.

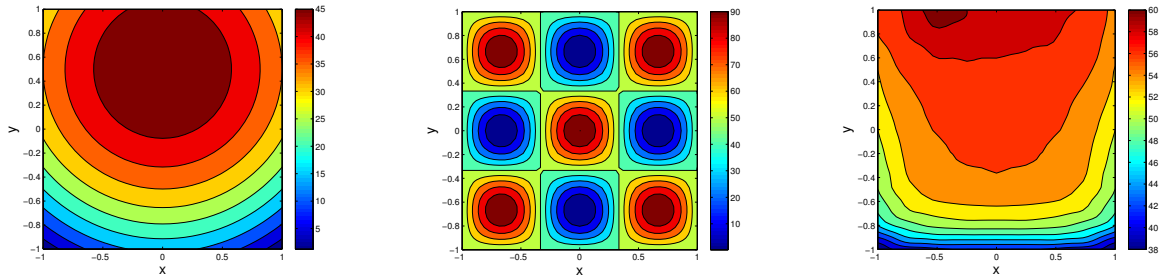
	q	$\overline{u}$	N	n	α	l	L	$\mathcal{C}^{(0)}$	A
CASE 1	q_1	$\overline{u}_1$	50	50	0	0	-	$\mathcal{C}_1^{(0)}$	Ω
CASE 2	q_2	$\overline{u}_2$	50	100	0	0.3	-	$\mathcal{C}_2^{(0)}$	Ω
CASE 3	q_1	$\overline{u}_3$	50	100	0	0.1	-	$\mathcal{C}_2^{(0)}$	Ω
CASE 4	q_1	$\overline{u}_4$	50	100	0	0.1	-	$\mathcal{C}_2^{(0)}$	Ω
CASE 5	q_3	$\overline{u}_4$	50	100	0	0.1	-	$\mathcal{C}_2^{(0)}$	Ω
CASE 6	q_1	$\overline{u}_4$	50	100	0	0.1	-	$\mathcal{C}_2^{(0)}$	$[-0.5, 1] \times [-0.5, 1]$
CASE 7	q_1	$\overline{u}_4$	50	100	0	0.3	-	$\mathcal{C}_3^{(0)}$, $\mathcal{C}_4^{(0)}$ and $\mathcal{C}_5^{(0)}$	Ω
CASE 8	q_3	$\overline{u}_4$	50	100	$1, 10$ 10^2, 10^3	0.1	6	$\mathcal{C}_6^{(0)}$	Ω
CASE 9	q_1	$\overline{u}_4$	50	100	10^2	0.1	3	$\mathcal{C}_2^{(0)}$	$[-0.5, 1] \times [-.05, 1]$

Table 5.2: Settings for different optimization cases

In Table 5.2, the second column is the heat sources q with:

$$\begin{aligned} q_1(x,y) &= 50 - 15x^2 - 15\left(y - \frac{1}{2}\right)^2 \qquad (x,y) \in \ \Omega \\ q_2(x,y) &= 50 + 50\sin\left(\frac{3\pi\, x}{2} + \frac{3\pi}{2}\right)\cos\left(\frac{3\pi\, y}{2} + \pi\right) \qquad (x,y) \in \ \Omega \end{aligned} \tag{5.5}$$

and $q_3(x, y)$ are obtained from the real world temperature data in [20], see Figure 5.9 for the contour of heat sources. The third column is desired temperature distribution



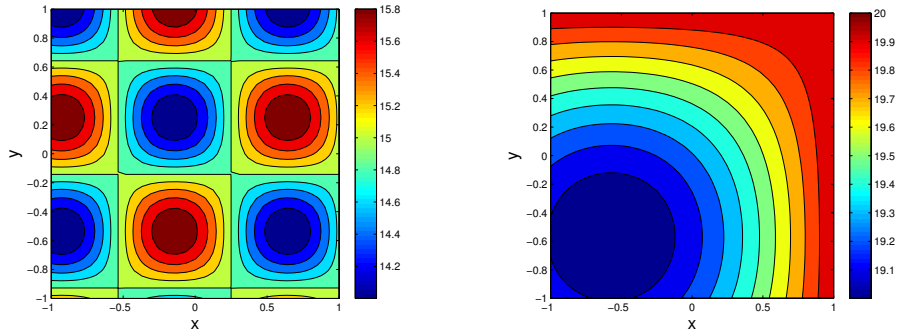
(a) Heat sources q_1 (b) Heat sources q_2 (c) Heat sources q_3

Figure 5.9: Contour of heat sources q_1, q_2 and q_3

$\overline{u}$ with:

$$\begin{aligned}
\overline{u}_1(x,y) &= 15 + \sin(4x-1)\cos(4y-1) \qquad (x,y) \in \ \Omega \\
\overline{u}_2(x,y) &= 20 - \sin(x-1)\sin(y-1) \qquad (x,y) \in \ \Omega \\
\overline{u}_4(x,y) &= 15 + \sin\left(\frac{3\pi\, x}{2} + \frac{3\pi}{2}\right)\cos\left(\frac{3\pi\, y}{2} + \pi\right) \qquad (x,y) \in \ \Omega \\
\overline{u}_4(x,y) &= 15 + \sin(2\pi\, x + \pi)\cos\left(2\pi\, y + \frac{\pi}{2}\right) \qquad (x,y) \in \ \Omega
\end{aligned} \tag{5.6}$$

and the plot for all desired temperature $\overline{u}$ is in Figure 5.10. The 4th column is resolution for the domain Ω which is understood as the size $(N+1)\times(N+1)$ of the Chebyshev grid and the 5th column is resolution for contour $\mathcal{C}$ which is understood as the size $2n$ of equispaced grid on the contour. The 6th column α is the weight for length constraint (see 3.13) and the 6th column is the Sobolev coefficient l (see 3.3.4). The 7th column is the total length L where for the case without length constraint we denote with '-'. The 8th column is the contour we used as the "initial guess" where $\mathcal{C}_1^{(0)}$, $\mathcal{C}_2^{(0)}$, $\mathcal{C}_3^{(0)}$, $\mathcal{C}_4^{(0)}$, $\mathcal{C}_5^{(0)}$ and $\mathcal{C}_6^{(0)}$ are shown as in Figure 5.11. The last column is the domain of interest A over which the cost functional is defined. In Figure 5.12(d) to 5.17(d) and 5.22(f) showing evolution of the contour during optimization iterations, we use blue dashed curve for the initial contour, black thin curve for the evolution and red wide curve for the final contour.



(a) Desired temperature $\overline{u}_1$ (b) Desired temperature $\overline{u}_2$

(c) Desired temperature $\overline{u}_3$ (d) Desired temperature $\overline{u}_4$

Figure 5.10: Contour of desired temperatures $\overline{u}_1$, $\overline{u}_2$, $\overline{u}_3$ and $\overline{u}_4$

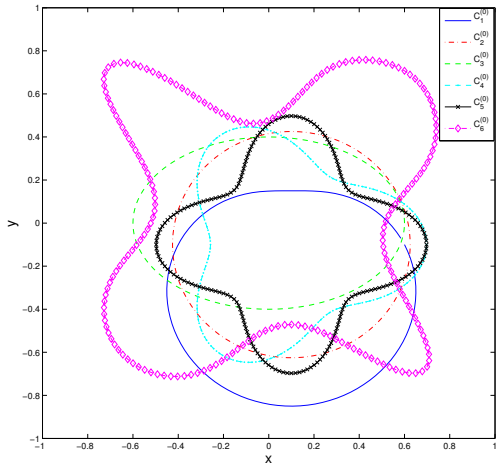


Figure 5.11: Initial contours applied for different cases

CASE 1 Here we applied the L^2 shape gradient and we can observe the non-smoothness occurs in the evolution of contour $\mathcal{C}$ in Figure 5.12(d). Figure 5.12(c) shows the evolution of cost functional at each iteration and we can observe a decrease from 10^3 to 1. Figure 5.12(e) shows the temperature distribution for the initial contour and Figure 5.12(f) for the final contour, both with black curve denoting the cooling contour. Comparing the final temperature with our target $\overline{u}$ in Figure 5.10(a) we could observe some similarity.

CASE 2 Here we applied the Sobolev shape gradient, compared with CASE 1 using L^2 shape gradient the evolution of contour is much more smooth (Figure 5.13(d)). The functional J also has a 10^2 scale decrease (Figure 5.13(c)).

CASE 3 In this case we applied a heat sources q_1 which has a low variation across the domain and a desired temperature $\overline{u}_3$ with a relatively high variation. And our method is also adaptive to this case as it is shown in Figure 5.14(d) with another final contour obtained. The cost functional decreases from around 10^2 to a scale around 1 in Figure 5.14(c). The final temperature is shown in Figure 5.14(f).

CASE 4 Here we applied a target $\overline{u}_4$ with a higher spatial variation compared with $\overline{u}_3$. We can see the final temperature distribution in Figure 5.15(f) after 30 iterations where the overall pattern matches $\overline{u}_4$.

CASE 5 In this case we applied a more or less coarse heat sources obtained from [20] and representing the actual heating in a battery cell. We can see that for $\overline{u}_4$ the final shape of the contour is more complicated in order to catch the pattern of target temperature (Figure 5.16(d)). For the final temperature (Figure 5.16(f)) we can see the high temperature along main diagonal direction and low temperature along the antidiagonal direction, which behaves the way $\overline{u}_3$ does. So far the above tests demonstrate the adaptivity of our method with respect to different heat sources q and $\overline{u}$.

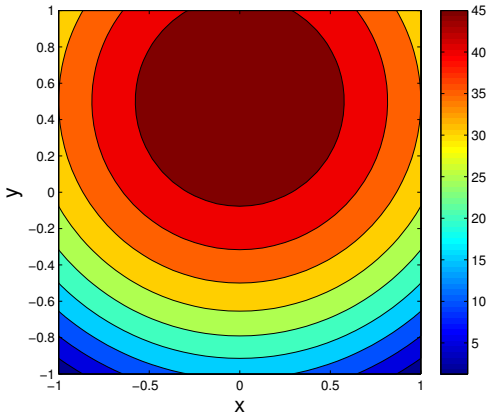
CASE 6 Here our subdomain of interest is $A = [-0.5, 1] \times [-0.5, 1]$ and we could see the evolution of contour $\mathcal{C}$ in Figure 5.17(d) showing that all the contours move upwards, ignoring the left and down part of Ω. Thus only the optimization in subdomain A is performed. We can also observe the matching pattern between the final temperature in Figure 5.17(f) and $\overline{u}_4$ only in the subdomain A. The functional is also minimized as in Figure 5.17(c).

CASE 7 In this test we applied different initial contours with $\mathcal{C}_3^{(0)}$ (dashed dot curve), $\mathcal{C}_4^{(0)}$ (dased curve) and $\mathcal{C}_5^{(0)}$ (solid curve) in Figure 5.18(c). Figure 5.18(d) shows the final curves for $\mathcal{C}_3^{(0)}$, $\mathcal{C}_4^{(0)}$ and $\mathcal{C}_5^{(0)}$. We can see for different initial contours our final contour can be different. This is possible since our optimization algorithm is designed to find the local minimum thus the optimal result may vary with different starting point. Also the decreasing rate for cost functional varies for different initial contour we applied as it is shown in Figure 5.18(e). In Figure 5.19 are the plots for temperature distributions on initial contour and final contour based on different starting contour $\mathcal{C}^{(0)}$.

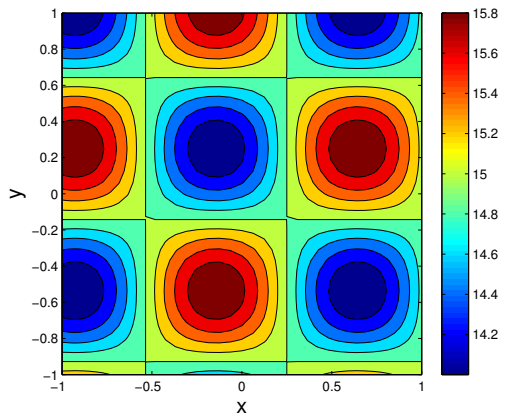
CASE 8 Here we obtain results with length constraint $L = 6$ for different α setting at 1, 10, 10^2 and 10^3. Figure 5.20(a) shows the cost functional with respect to different α. We can see that when α is large there is little improvement on our cost functional. That is because we have a high penalty on length for large α. Figure 5.20(b) shows the evolution of cost functional $\mathcal{J}$, the part without considering the length constraint. We can see at some points the functional goes up since it is not the functional we are minimizing. Figure 5.20(c) is the evolution for the length penalty and Figure 5.20(d) shows different final contours we got for variety of α. Because of high α we can not make too much improvement on the contour. Figure 5.20(e) shows the evolution of contour's length in which for high value of α the length of contour stay around the prescribed length 6. Figure 5.20(f) is the initial temperature distribution and in Figure 5.21 are the

final temperature distributions corresponding to different α's.

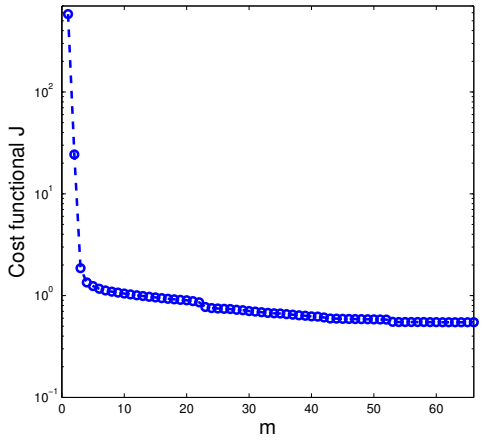
CASE 9 In this case we used the subdomain $A = [-0.5, 1] \times [-0.5, 1]$ in an optimization problem with a length constraint. The control variable for length constraint $\alpha = 100$. Figure 5.22(c), 5.22(d) and 5.22(e) are the evolution for cost functional $\mathcal{J}_L$, $\mathcal{J}$ and length penalty. Because of the α fact our cost functional have only a limited decrease. Also the improvement on our contour is limited as in Figure 5.22(f). Figure 5.22(h) shows the temperature distribution on the final contour.



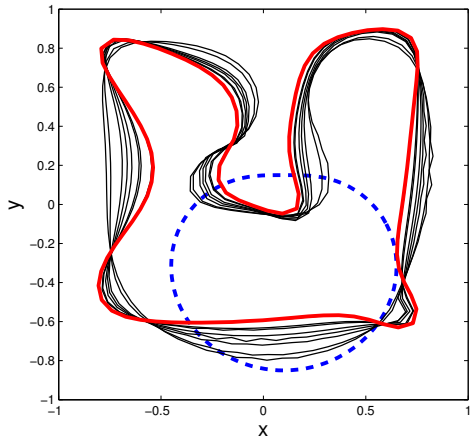
(a) Heat sources q_1



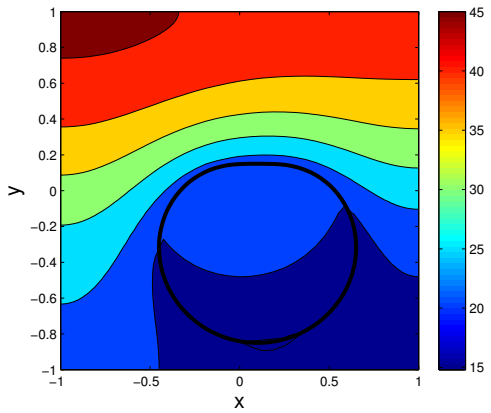
(b) Desired temperature $\overline{u}_1$



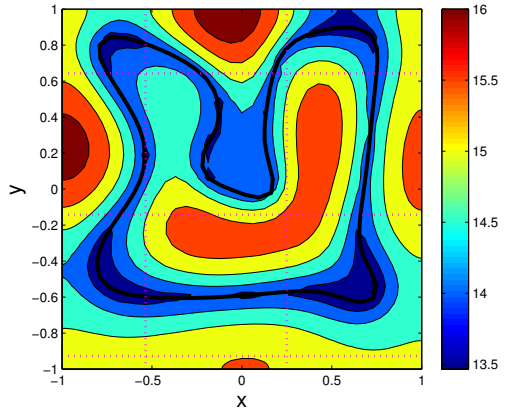
(c) Evolution of functional $\mathcal{J}^{(m)}$



(d) Evolution of contour $\mathcal{C}^{(m)}$

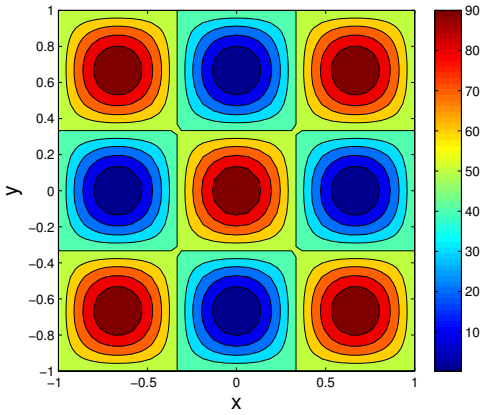


(e) Initial temperature

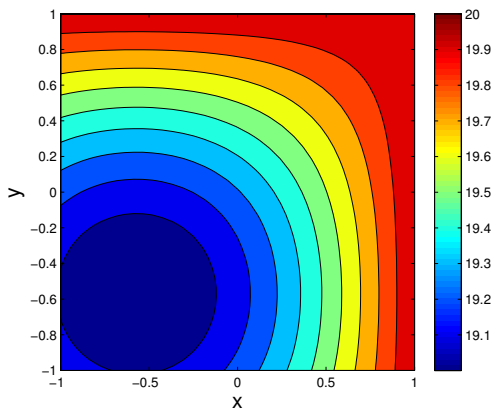


(f) Final temperature (the grid corresponds to cell pattern of $\overline{u}_1$)

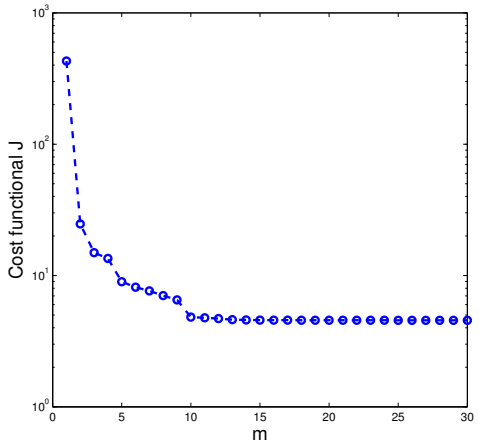
Figure 5.12: Optimization results for CASE 1 when L^2 shape gradient is applied; Heat sources is q_1 and desired temperature is $\overline{u}_1$



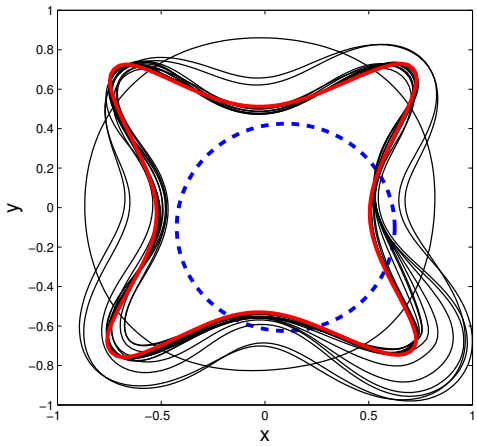
(a) Heat sources q_2



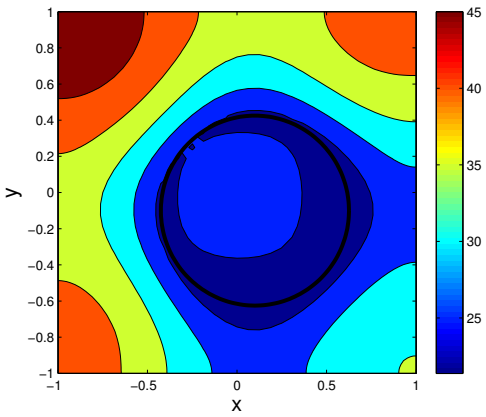
(b) Desired temperature $\overline{u}_2$



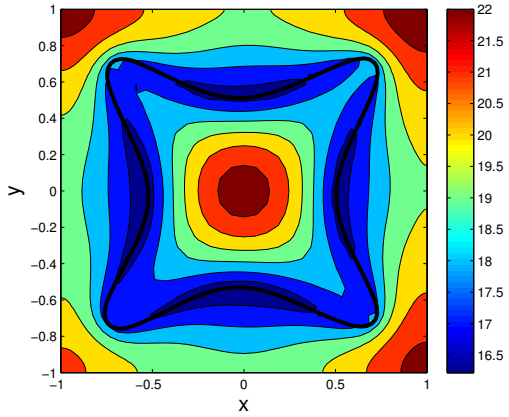
(c) Evolution of functional $\mathcal{J}^{(m)}$



(d) Evolution of contour $\mathcal{C}^{(m)}$

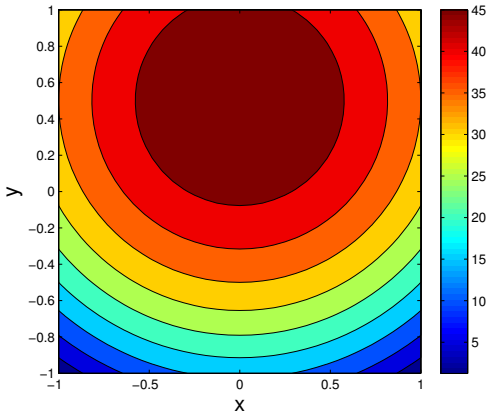


(e) Initial temperature

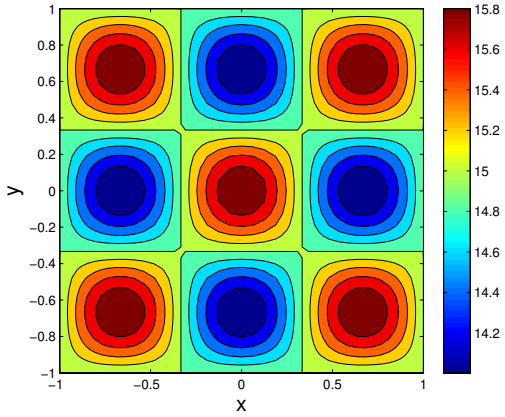


(f) Final temperature

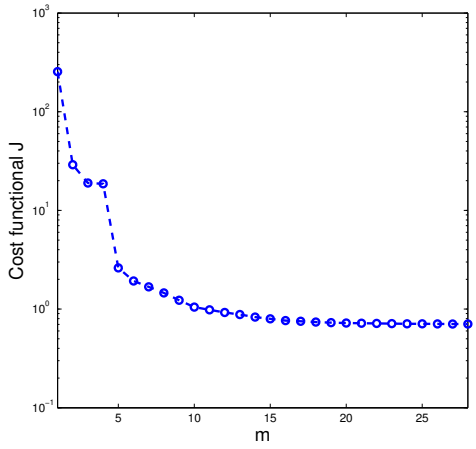
Figure 5.13: Optimization results for CASE 2 when q_2 and $\overline{u}_2$ are applied.



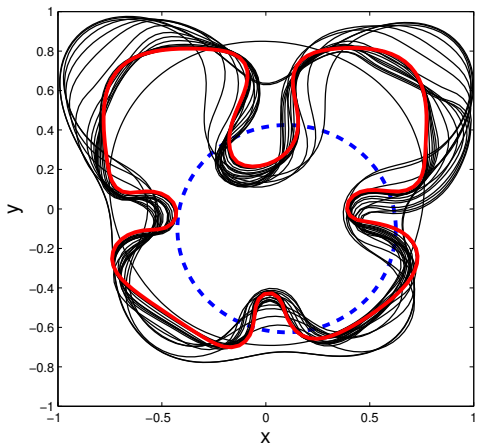
(a) Heat sources q_1



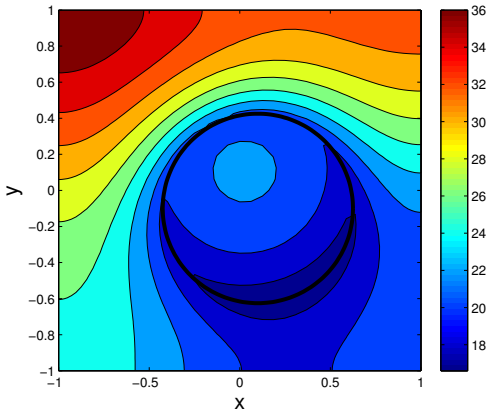
(b) Desired temperature $\overline{u}_3$



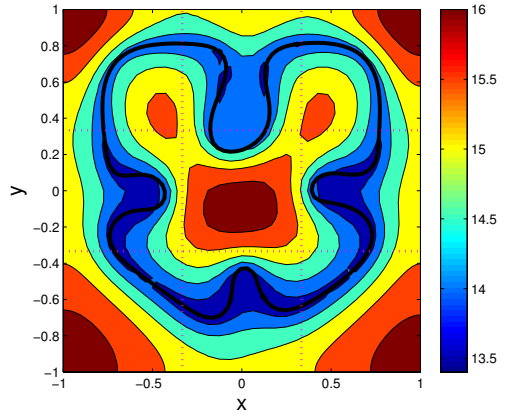
(c) Evolution of functional $\mathcal{J}^{(m)}$



(d) Evolution of contour $\mathcal{C}^{(m)}$

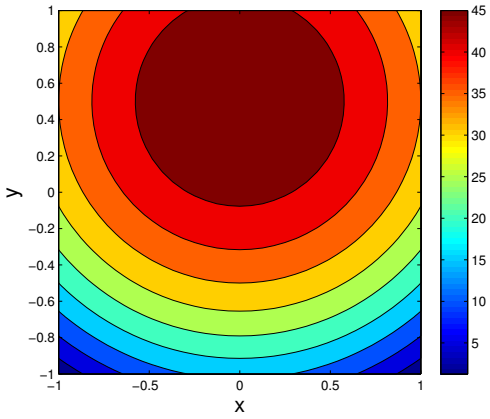


(e) Initial temperature

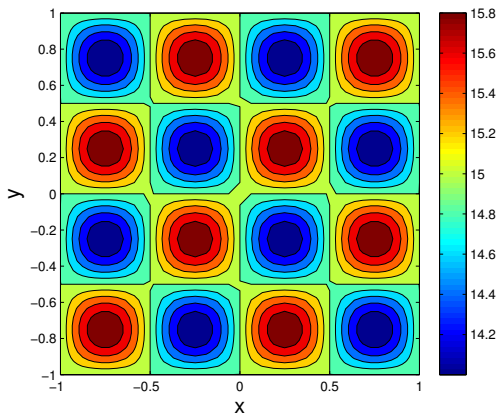


(f) Final temperature (the grid corresponds to cell pattern of $\overline{u}_3$)

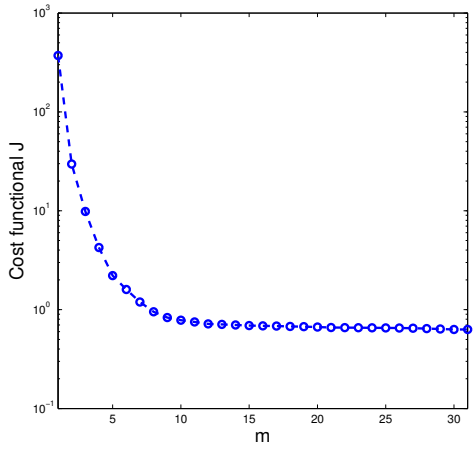
Figure 5.14: Optimization results for CASE 3 when q_1 and $\overline{u}_3$ are applied.



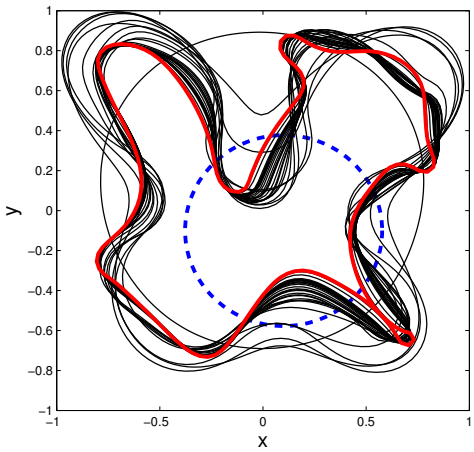
(a) Heat sources q_1



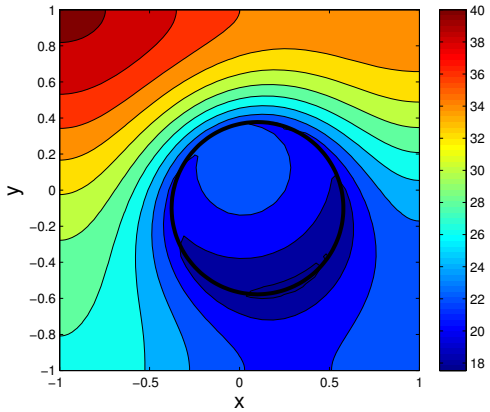
(b) Desired temperature $\overline{u}_4$



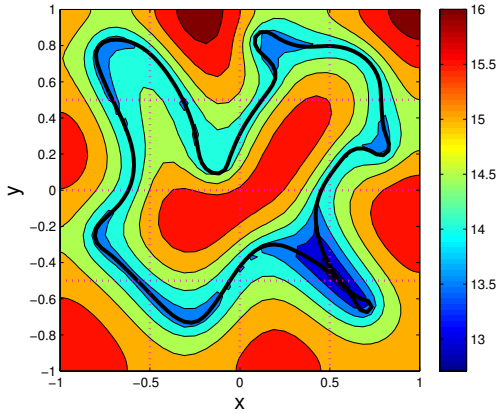
(c) Evolution of functional $\mathcal{J}^{(m)}$



(d) Evolution of contour $\mathcal{C}^{(m)}$

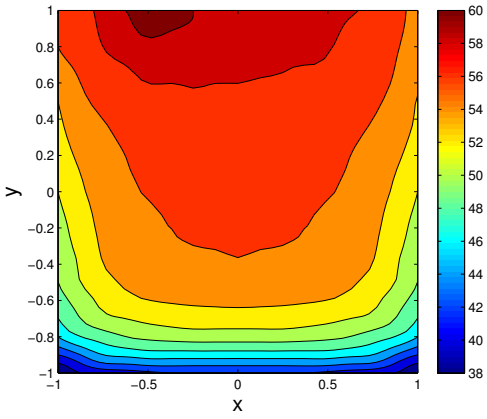


(e) Initial temperature

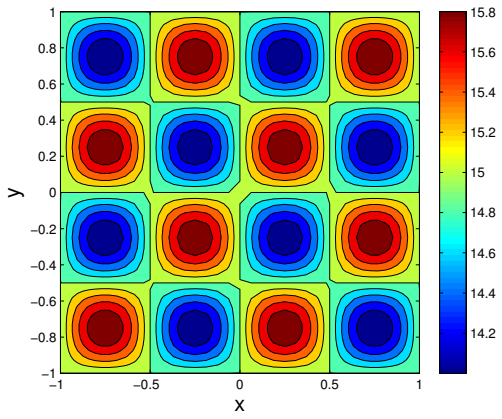


(f) Final temperature (the grid corresponds to cell pattern of $\overline{u}_4$)

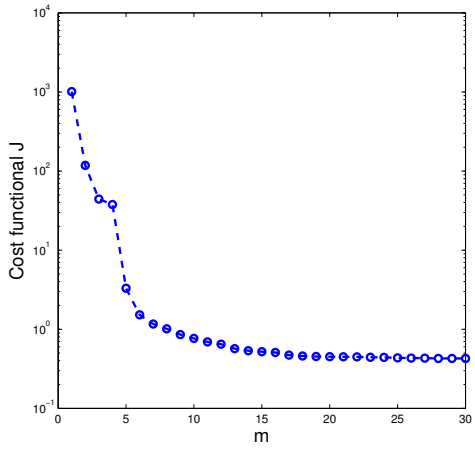
Figure 5.15: Optimization results for CASE 4 when q_1 and $\overline{u}_4$ are applied.



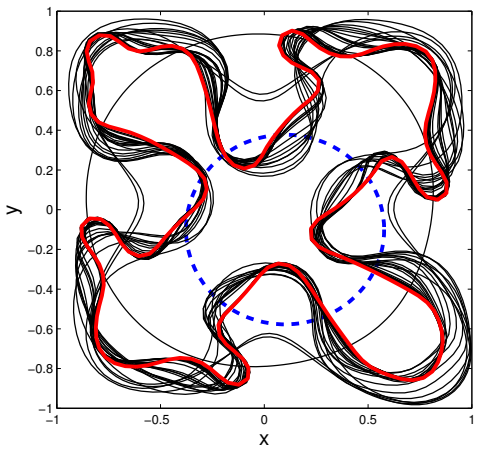
(a) Heat sources q_3



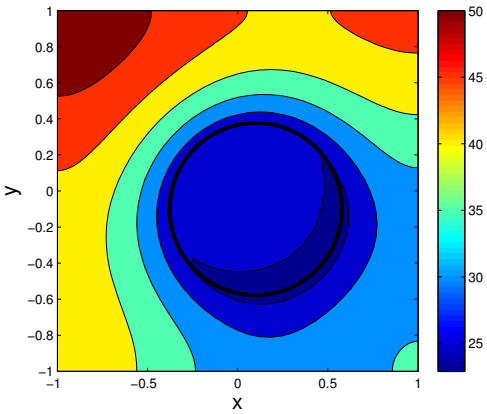
(b) Desired temperature $\overline{u}_4$



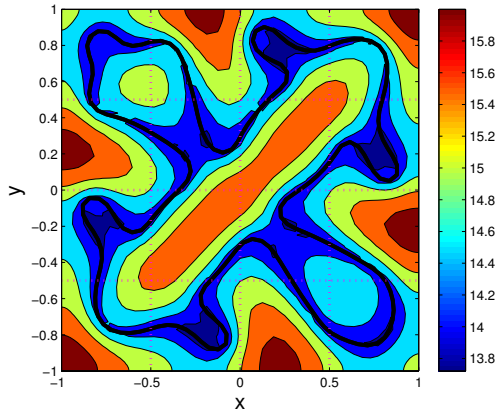
(c) Evolution of functional $\mathcal{J}^{(m)}$



(d) Evolution of contour $\mathcal{C}^{(m)}$

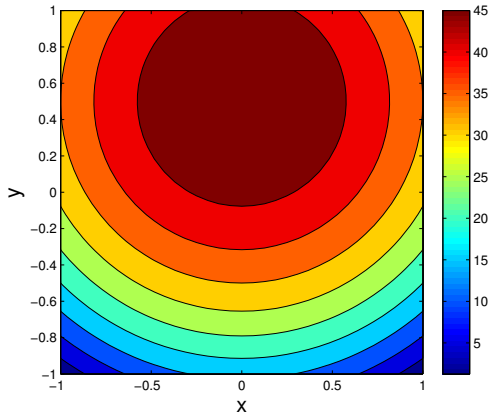


(e) Initial temperature

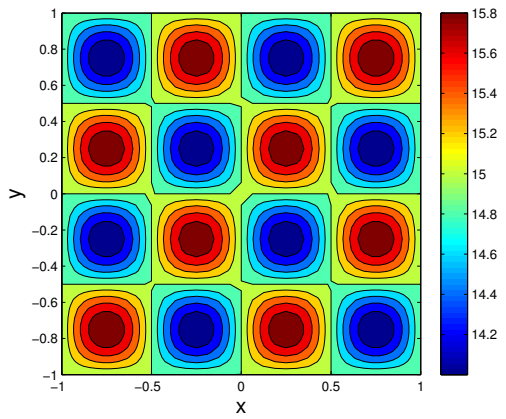


(f) Final temperature (the grid corresponds to cell pattern of $\overline{u}_4$)

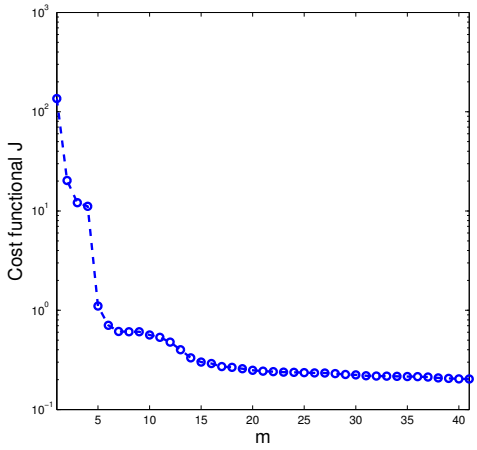
Figure 5.16: Optimization results for CASE 5 when q_3 [20] and $\overline{u}_4$ are applied.



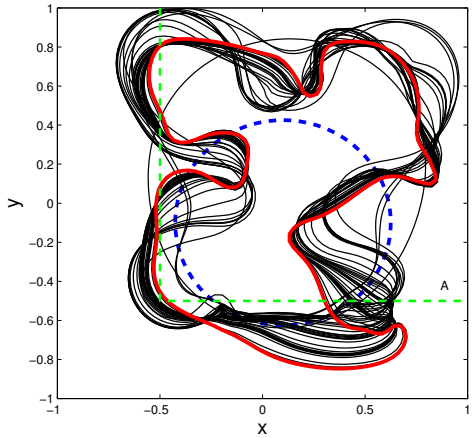
(a) Heat sources q_1



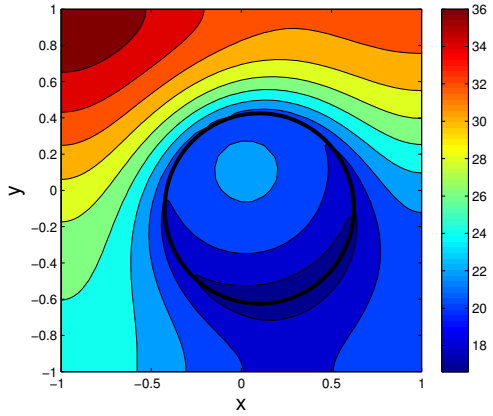
(b) Desired temperature $\overline{u}_4$



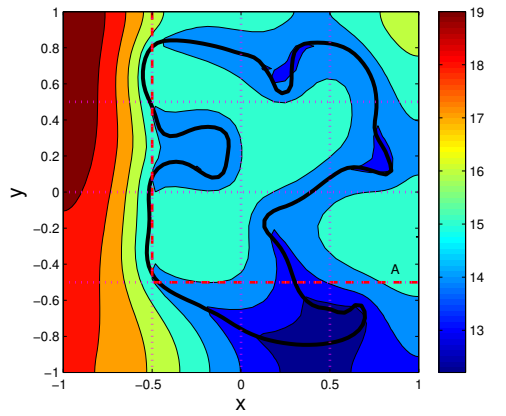
(c) Evolution of functional $\mathcal{J}^{(m)}$



(d) Evolution of contour $\mathcal{C}^{(m)}$

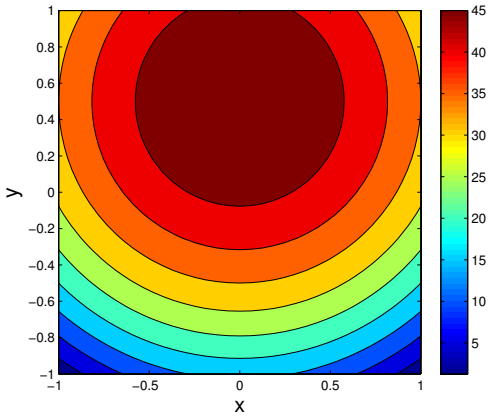


(e) Initial temperature

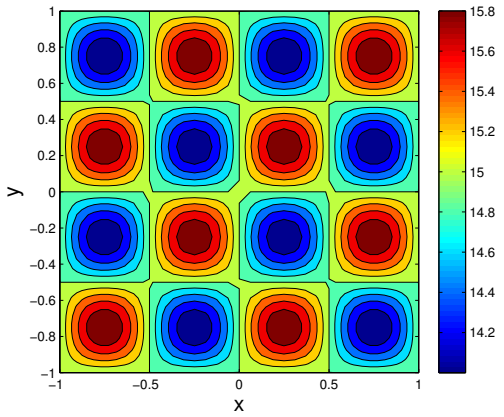


(f) Final temperature (the grid corresponds to cell pattern of $\overline{u}_4$)

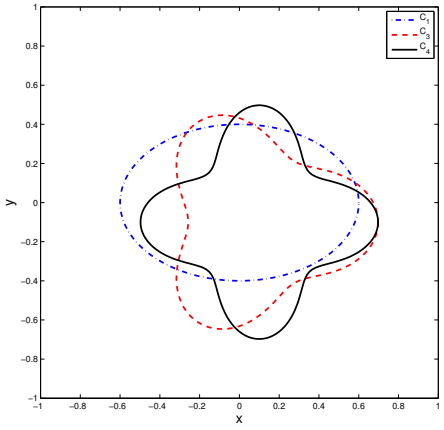
Figure 5.17: Optimization results for CASE 6 when subdomain of interest is $[-\frac{1}{2}, 1] \times [-\frac{1}{2}, 1]$; q_1 and $\overline{u}_4$ are applied.



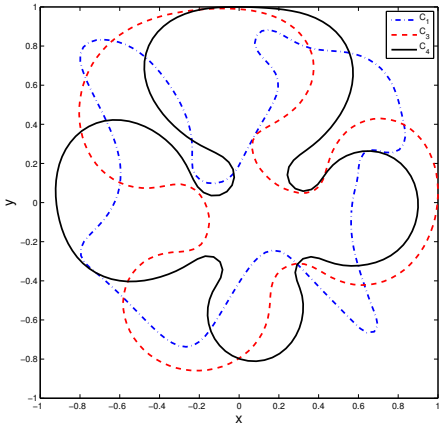
(a) Heat sources q_1



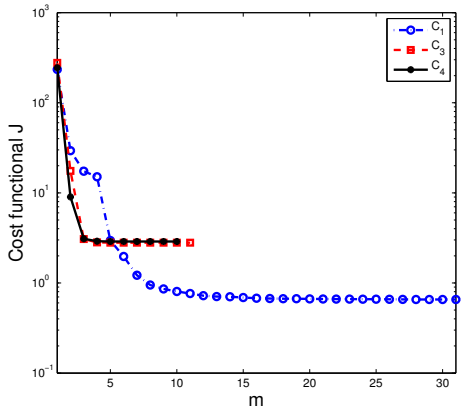
(b) Desired temperature $\overline{u}_4$



(c) Initial contours applied

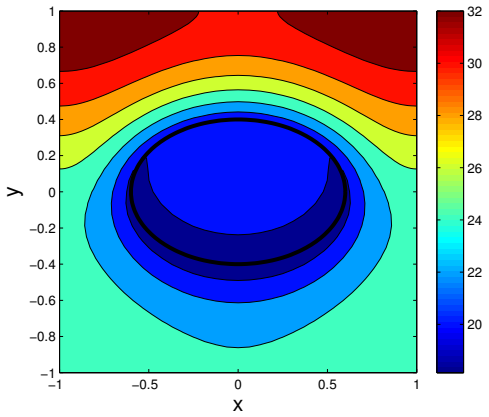


(d) Final contours

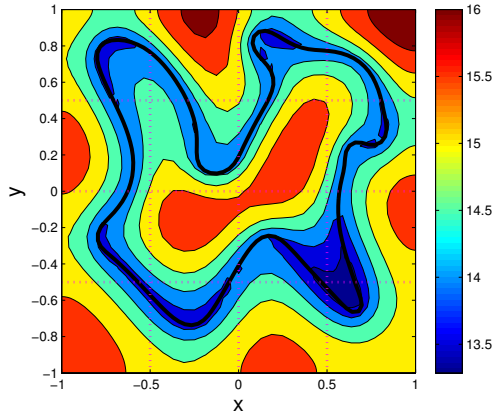


(e) Evolution of functional $\nabla \mathcal{J}^{(m)}$

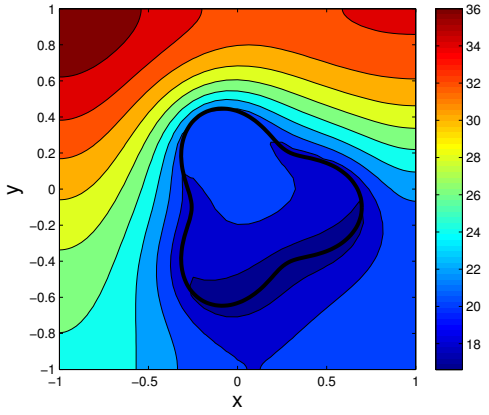
Figure 5.18: Optimization results for CASE 7 when different initial contours are applied; heat sources is q_1 and desired temperature is $\overline{u}_4$.



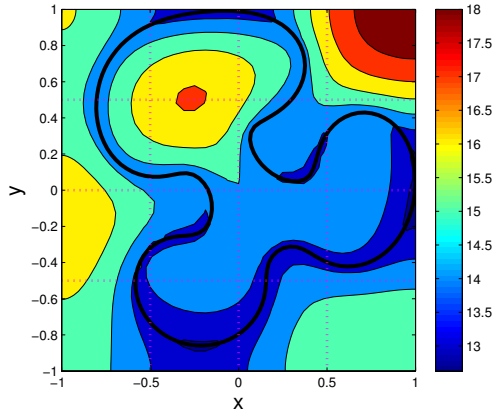
(a) Initial temperature for $\mathcal{C}_3^{(0)}$



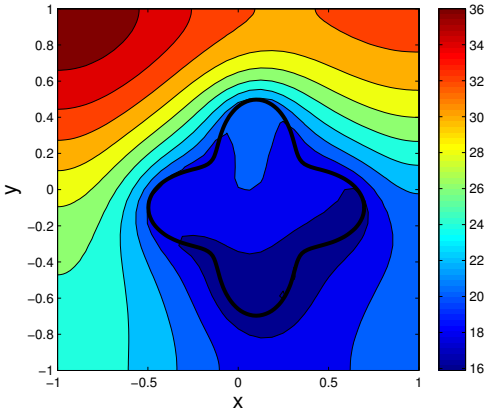
(b) Final temperature for $\mathcal{C}_3^{(0)}$



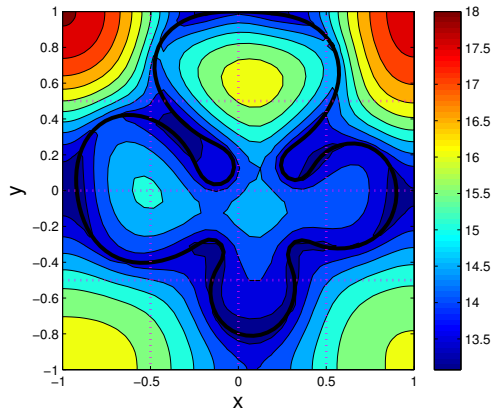
(c) Initial temperature for $\mathcal{C}_4^{(0)}$



(d) Final temperature for $\mathcal{C}_4^{(0)}$

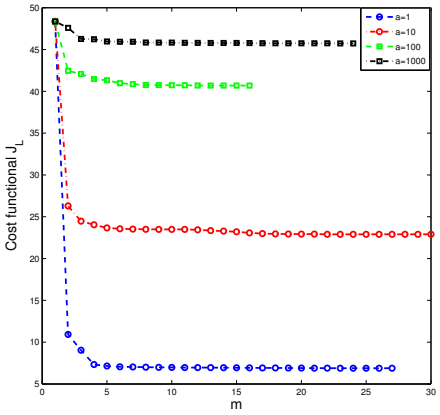


(e) Initial temperature for $\mathcal{C}_5^{(0)}$

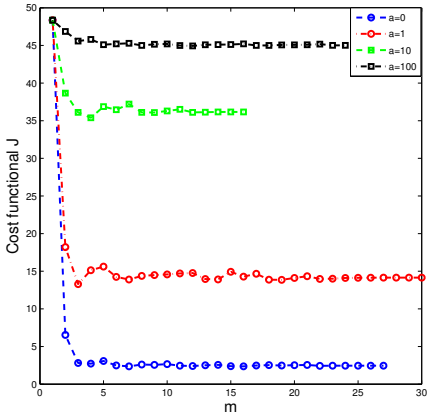


(f) Final temperature for $\mathcal{C}_5^{(0)}$

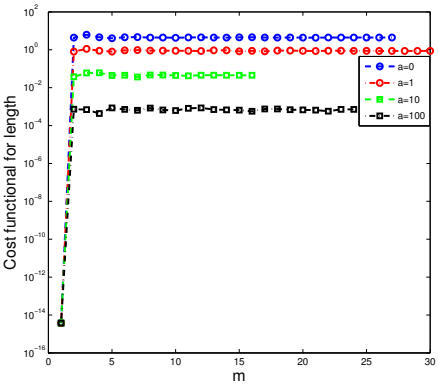
Figure 5.19: Initial and final temperatures for CASE 7 when $\mathcal{C}_3^{(0)}$, $\mathcal{C}_4^{(0)}$ and $\mathcal{C}_5^{(0)}$ are applied (the grid corresponds to cell pattern of $\overline{u}_4$).



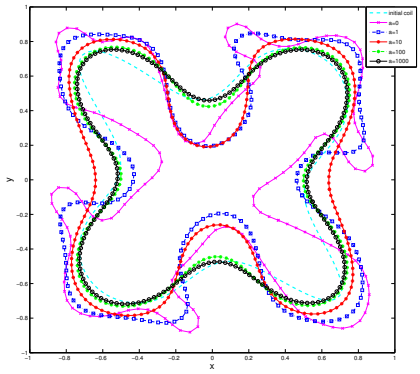
(a) Evolution of functional $\mathcal{J}_L^{(m)}$



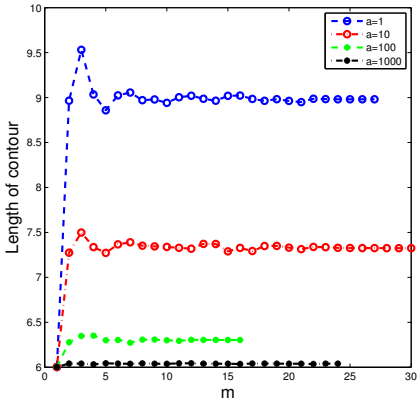
(b) Evolution of functional $\mathcal{J}^{(m)}$



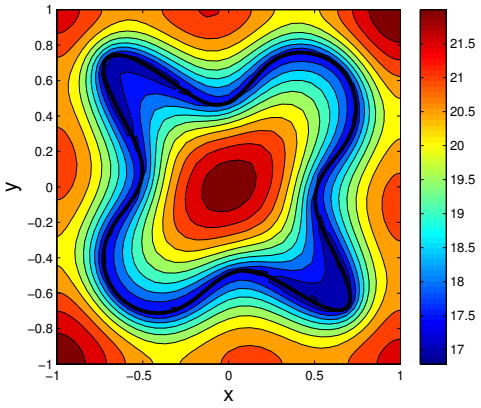
(c) Evolution of the length penalty



(d) Evolution of contour $\mathcal{C}^{(m)}$

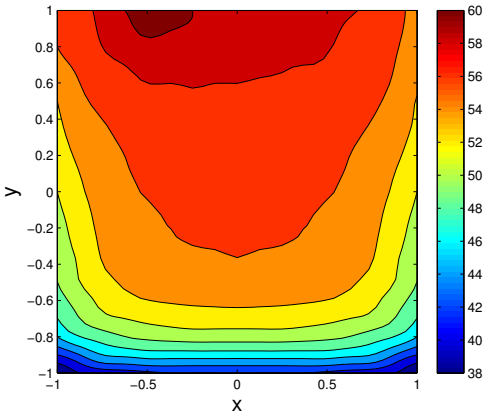


(e) Evolution of the length $L^{(m)}$

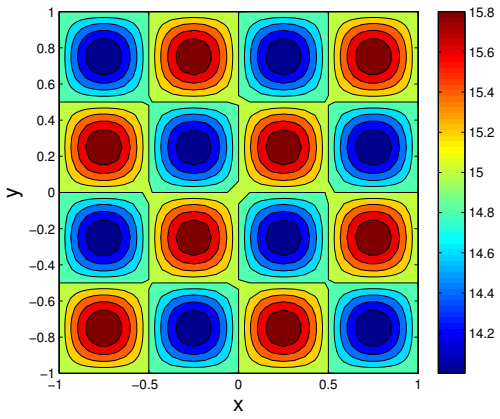


(f) Initial temperature

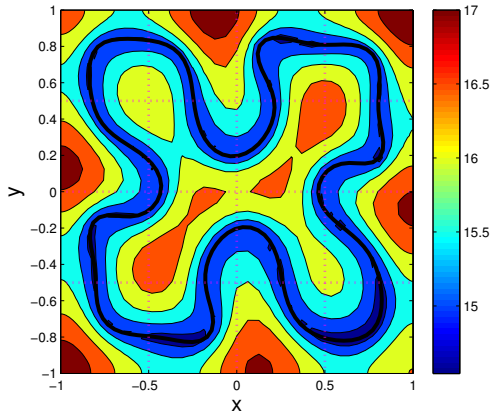
Figure 5.20: Optimization results for CASE 8 when different α's are applied; heat sources is q_3 and desired temperature is $\overline{u}_4$.



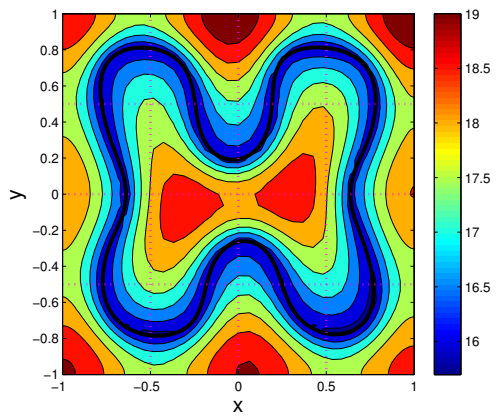
(a) Heat sources q_3



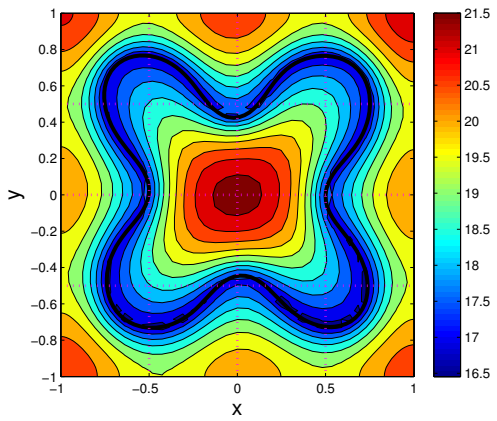
(b) Desired temperature $\overline{u}_4$



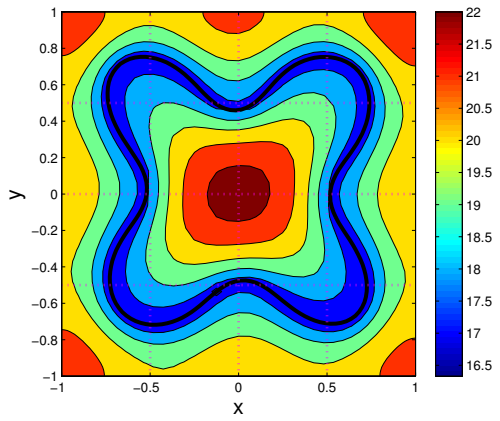
(c) Final temperature for $\alpha = 1$



(d) Final temperature for $\alpha = 10$

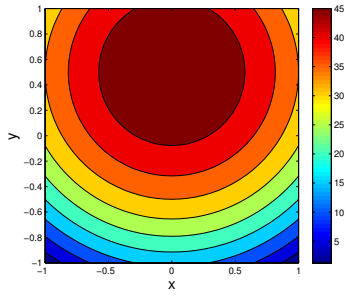


(e) Final temperature for $\alpha = 10^2$

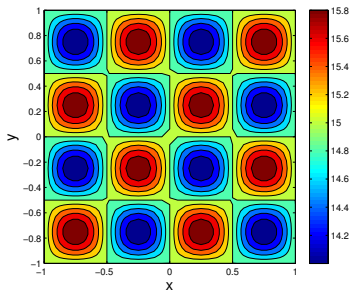


(f) Final temperature for $\alpha = 10^3$

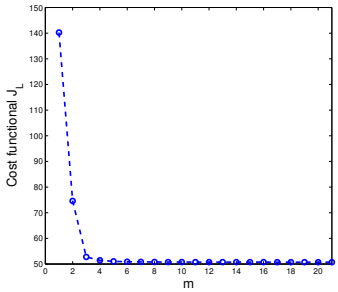
Figure 5.21: Final temperatures for $\alpha = 1$, $\alpha = 10$, $\alpha = 10^2$, $\alpha = 10^2$ in CASE 8 (the grid corresponds to cell pattern of $\overline{u}_4$).



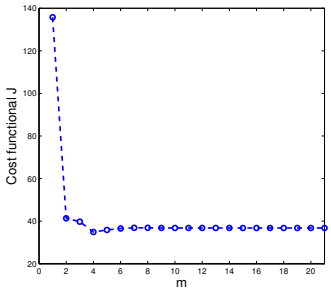
(a) Heat sources q_1



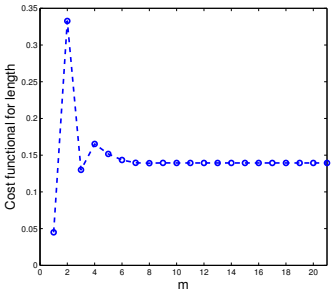
(b) Desired temperature $\overline{u}_4$



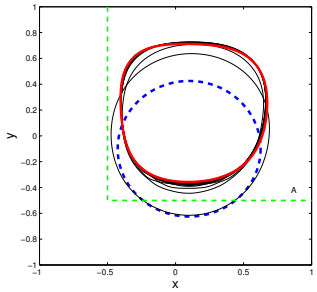
(c) Evolution of functional $\mathcal{J}_L^{(m)}$



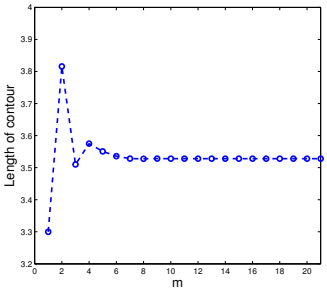
(d) Evolution of functional $\mathcal{J}^{(m)}$



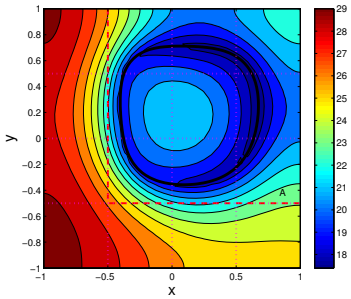
(e) Evolution of the length penalty



(f) Evolution of contour $\mathcal{C}^{(m)}$



(g) Evolution of the length $L^{(m)}$



(h) Final temperature (the grid corresponds to cell pattern of $\overline{u}_4$)

Figure 5.22: Optimization results for CASE 9 when subdomain of interest is $[-\frac{1}{2}, 1] \times [-\frac{1}{2}, 1]$ and $\alpha = 100$; heat sources is q_1 and desired temperature is $\overline{u}_4$.

Chapter 6

Conclusion and Future Work

This work is motivated by the optimal design of the liquid coolant system used in today's HEV battery pack. We introduced a 2D steady-state heat conduction governing system to model the heat transfer process in the battery pack. The battery pack was modelled by a 2D isolated square domain with some prescribed heat sources representing the battery heating. The coolant channel was simplified to a 1D close coil with constant temperature and thus no liquid flow was considered in the mathematical model. The problem is to find the shape of the coolant channel such that the temperature on some region is close to a given temperature distribution. To do so, a cost functional measuring the difference between the real and desired temperature over a certain region of interest was proposed in the least square sense. A PDE-constrained optimization was formulated and an optimal condition based on shape-differential calculus was introduced. The governing model was solved computationally using a combination of spectral method and boundary integral techniques. Whereas the optimal shape of coil was found along the conjugate gradient which is based on the shape gradient of cost functional w. r. t the shape of coil. The shape gradient of our functional is obtained by adjoint analysis and shape-differential calculus. Then we performed the validation for the solution of our governing model and adjoint system. By κ-test the

validation of our shape gradient was also confirmed. Finally, the optimization results based on different settings were presented.

Using the validation in section 5.1 we could see that our proposed approach of combining spectral methods and boundary integral techniques works well in solving the direct PDE system ($\mathcal{D}$) and the adjoint PDE system ($\mathcal{A}$). By increasing the resolution for both the domain and contour we can obtain a high accuracy result.

The κ-test for shape gradient evaluation also provided us with confidence in our adjoint-based shape gradient. Because the resolution can be a key factor affecting the accuracy of the gradient, we believe that by using a dense grid an accurate shape gradient can be obtained.

The optimal results in section 5.3 show that when different combinations of heat sources q and desired temperatures $\overline{u}$ are present, the optimal shape of coil $\tilde{\mathcal{C}}$ is not clear by intuition. Moreover, our obtained optimal shapes $\tilde{\mathcal{C}}$ lead to a reasonably small difference between the desired temperatures distribution and actual temperatures distribution in some chosen subdomain A. For the optimal results, we can have different local minimum depending on the choice of initial contour. In the present of length constraint, the change of contour is limited thus the optimal result is not that promising compared with free-constraint result. In addition, the algorithm is not adaptive to very large selection of contour length L since the probability of twisting can be high if we have a long contour packed into a finite domain.

For the future improvement, we will consider multiple channels with endings on the boundary of domain instead of single channel with a shape of closed curve. Moreover, we will incorporate convection by modelling the actual liquid flow in the channel.

Bibliography

[1] AUTO21 Research Project, *Multidisciplinary Optimization of Hybrid and Electric Vehicle Batteries. http://www.auto21.ca/en/subcontent.php?page=ae2507.* Project Leader: B. Protas, McMaster University.

[2] R. Haberman, *Elementary Applied Partial Differential Equations: with Fourier Series and Boundary Value Problems*,Prentice-Hall Inc, New Jersey, [1987].

[3] A. Ashrafizadeh, G. D. Raithby and G. D. Stubley, *Direct Design of Shape*. Numerical Heat Transfer, Part B, 41: 501-520, [2002].

[4] M. J. Colaco, H. R. B. Orlande and G. S. Dulikravich, *Inverse and Optimization Problems in Heat Transfer*. Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol.28 no.1, [2006].

[5] C. H. Lan, C. H. Cheng and C. Y. Wu, *Shape Design for Heat Conduction Problems using Curvilinear Grid Genegration, Conjugate Gradient and Redistribution Methods*. Numerical Heat Transfer, Part A, 39: 487-510, [2001].

[6] C. H. Huang and B. H. Chao, *An inverse geometry problem in identifying irregular boundary configrations*. International Journal of Heat and Mass Transfer,Vol. 40, No. 9: 2045-2053, [1997].

[7] C. H. Huang and T. Y. Hsiung, *An inverse design problem of estimating optimal*

shape of cooling passages in turbine blades. International Journal of Heat and Mass Transfer, Vol. 42, No. 23, 4307-4319, [1999].

[8] C. H. Huang and C. Y. Liu, *A three-dimensional inverse geometry problem in estimating simultaneously two interfacial configurations in a composite domain*. International Journal of Heat and Mass Transfer, Vol. 53, No. 1-3, 48-57, [2010].

[9] C. H. Huang and C. C. Shih, *A shape identification problem in estimating simultaneously two interfacial configrations in a multiple region domain*. Applied Thermal Engineering, 26: 77-88, [2006].

[10] J. Nocedal and S. J. Wright,*Springer Series in Operations Research: Numerical Optimization*. Springer-Verlag, New York, [1999].

[11] J. Haslinger and R. A. E. Mäkinen, *Introduction to Shape Optimization: Theory, Approximation, and Computation*. siam, Philadelphia, [2003].

[12] S. Schmidt and V. Schulz, *Shape Derivatives for General Objective Functions and the Incompressible Navier-Stokes Equations*. Control and Cybernetics, 39(3): 677-713, [2010].

[13] H. L. Royden, *Real Analysis*. Macmillan Publishing Company, [1988]

[14] B. Protas, *Remarks on Symbolic Generation of Adjoint Systems in PDE Optimization Problems*. Preprint, [2010].

[15] B. Protas, T. Bewley and G. Hagen, *A computational framework for the regularization of adjoint analysis in multiscale PDE systems*. Journal of Computational Physics, 195(1), 49-89, [2004].

[16] L. N. Trefethen,*Spectral Methods in Mathlab*. siam, Philadelphia, [2000].

[17] R. Kress,*Linear Integral Equations*. Springer-Verlag, New York, [1999].

[18] K. Atkinson and W. Han,*Theoretical Numerical Analysis: A Functional Analysis Framework*. Springer-Verlag, New York, [2001].

[19] W. Hackbusch, *Integral Equations: Theory and Numerical Treatment*. Birkhäuser, [1995].

[20] U.S. Kim, C.B. Shin and C.S. Kim, *Modeling for the scale-up of a lithium-ion polymer battery*. Journal of Power Sources, 189:841-846, [2009].