SHAPE OPTIMIZATION OF ELLIPTIC PDE PROBLEMS ON COMPLEX DOMAINS

SHAPE OPTIMIZATION OF ELLIPTIC PDE PROBLEMS ON COMPLEX DOMAINS

By

KATSIARYNA NIAKHAI, M.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 (2013) McMaster University

(Mathematics) Hamilton, Ontario

TITLE:	SHAPE OPTIMIZATION OF ELLIPTIC PDE PROBLEMS ON COMPLEX DOMAINS
AUTHOR:	Katsiaryna Niakhai, M. Sc. (Belarusian State University)
SUPERVISOR:	Dr. Bartosz Protas
NUMBER OF PAGES:	VI, 103

Abstract

This investigation is motivated by the problem of optimal design of cooling elements in modern battery systems. We consider a simple model of two-dimensional steady-state heat conduction described by elliptic partial differential equations (PDEs) and involving a one dimensional cooling element represented by an open contour. The problem consists in finding an optimal shape of the cooling element which will ensure that the solution in a given region is close (in the least square sense) to some prescribed target distribution. We formulate this problem as PDE-constrained optimization and the locally optimal contour shapes are found using the conjugate gradient algorithm in which the Sobolev shape gradients are obtained using methods of the shape-differential calculus combined with adjoint analysis. The main novelty of this work is an accurate and efficient approach to the evaluation of the shape gradients based on a boundary-integral formulation. A number of computational aspects of the proposed approach is discussed and optimization results obtained in several test problems are presented.

Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. This work would not have been accomplished without close guidance and assistance of my supervisor Dr. Bartosz Protas whose insight and solid experience helped to overcome many difficulties arising on the way to the final result. I thank him for the patience and encouragement throughout the course of this thesis.

I am indebted with my knowledge and experience to the professors in McMaster University who I took courses from and worked with as a teaching assistant: Dr. Walter Craig, Dr. Dmitry Pelinovsky, Dr. Lia Bronsard, Dr. Jean Pierre Gabardo, Dr. Eric Sawyer and Dr. Gail Wolkowicz.

My warm and sincere thanks also go to my colleagues: Dmitry Ponomarev, Anton Sakovich, Vladislav Bukstynov, Diego Ayala and Xiaohui Peng for their support, guidance and helpful suggestions.

Additionally, I am very thankful to professors Dr. Nataliya Paulianok, Dr. Anatoliy Kalinin, Dr. Rafail Gabasov, Dr. Sergey Markov and Dr. Valeriy Kraxotko who I had opportunity to learn from in the past and who also gave me support and valuable advice regarding academic career.

Last but not least, I would like to thank my mother Galina, husband Alexander and best friend Irina for encouraging me in difficult moments.

To the memory of my grandfather

Contents

1 Introduction **1**

1.1 Mathematical Modelling of Heat Transfer 2

1.2 Shape Optimization in Heat Transfer 3

1.3 Structure of the Thesis 5

2 Formulation of the Problem **6**

2.1 Mathematical Model 6

2.2 Problem of Optimal Shape Design 8

3 Shape Optimization Approach **11**

3.1 Sensitivity Analysis in Shape Optimization 11

3.1.1 Shape Calculus: Material and Shape Derivative 12

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 . . 17

3.3 Gradient Descent Method 19

3.3.1 Perturbation PDE System 19
3.3.2 Adjoint PDE System 21
3.3.3 Ensuring smoothness of the gradient 31

4 Numerical Implementation **33**

4.1 Spectral Methods for Non-periodic Domains 34
4.1.1 Spectral Differentiation for Non-periodic Domain 35
4.1.2 Spectral Collocation Methods 40
4.2 Clenshaw-Curtis Quadrature 42
4.3 Boundary Integral Equation 43
4.3.1 Single-layer Potential 44
4.4 Numerical Implementation 47
4.4.1 Discretizing Equation (4.21a) 48
4.4.2 Discretizing Equation (4.21b) 49
4.4.3 Discretizing Boundary Conditions (4.21c) 54
4.4.4 Discretization of $u_h(x)$ 57
4.4.5 Normal Derivatives on $\mathcal{C}$ and Curvature 57
4.4.6 Sobolev Gradient and Evaluation of Cost Functional 59
4.5 Optimization Algorithm 60
4.5.1 Conjugate Gradient Method 60
4.5.2 Optimization Algorithm 61

5 Results **62**

5.1 Validation of BIE Method 62

5.2 Validation of the Cost Functional Gradient 66

5.3 Optimization Results 70

6 Conclusion and Future Work **81**

A Shape-Derivative of the Arc-Length Coordinate **84**

B Alternative Formulation of the Shape Gradients **87**

B.1 PDE-based Shape-Differentiation and Derivation of the Adjoint System 90

B.2 PDE-based Shape-Differentiation and BIE-based Derivation of the Adjoint System 92

B.3 BIE-based Shape-Differentiation and Derivation of the Adjoint System 94

B.4 Summary of the Results 100

List of Figures

2.1 Sketch of the domain 7

4.1 Chebyshev points 35

4.2 2D Chebyshev grid 38

4.3 Tensor product grid 3x3 and enumeration pattern 39

4.4 Schematic illustration of the different grids used to discretize the contour $\mathcal{C}$: (circles) - Chebyshev grid, (squares) - uniform grid. 53

4.5 Sketch of the normal vector at the boundary position $\mathbf{x}_0^c$ 54

5.1 The accuracy of BIE method with density $\mu(t) = \cos(t)$: (asterisks) - the Clenshaw-Curtis quadrature approach, (circles) - the Chebyshev-Gauss approach, (solid line) - a line with the slope -3/4 64

5.2 The accuracy of BIE method with density $\mu(t) = \sin(t)$: (asterisks) - the Clenshaw-Curtis quadrature approach, (circles) - the Chebyshev-Gauss approach, (solid line) - a line with the slope -3/2 65

5.3 Initial contour $\mathcal{C}_0$ and perturbed contours (5.3) used in the κ-test . . . 68

5.4 κ-test 1, $\log_{10}|\kappa - 1|$ for $\nabla\mathcal{J}$ with 4 different perturbations 69

5.5 κ-test 2, $\log_{10}|\kappa - 1|$ for $\nabla\mathcal{J}$ with 4 different perturbations 70

5.6 Optimization results for CASE 1 75

5.6 (continued) Optimization results for CASE 1 76

5.7 Optimization results for CASE 2 77

5.7 (continued) Optimization results for CASE 2 78

5.8 Optimization results for CASE 3 79

5.8 (continued) Optimization results for CASE 3 80

B.1 Schematic of the gradient derivation: (double lines) - BIE transformation, (dashed lines) - shape differentiation, (dotted lines) - adjoint identity, (solid lines) - formulation of the gradient 88

List of Tables

5.1 Settings for different κ-tests defined in (5.1) 66

5.2 Parameters of the optimization problems 71

B.1 The gradients and adjoint systems derived in the different approaches . 100

Chapter 1

Introduction

This work is motivated by the problem of optimization of the heat transfer processes in automotive industry. In particular, we are investigating the shape of the cooling channel in Li-ion battery pack which provides the energy for hybrid electric vehicles (HEV). The main question discussed in this thesis is how we can design the channel, so that for some provided length we can achieve some target temperature distribution in the given area of the battery pack. Generally, the shape of a cooling element can depend on many different factors, such as the heat sources distribution, the material of cooling channel, etc. Here we concentrate on the fundamental effect of the geometry of the cooling units on the efficiency of the heat transfer processes in the battery pack. We consider a simple mathematical model of two-dimensional steady-state heat conduction involving a one-dimensional cooling element represented by an open contour. This investigation is an extension of the previous work [1], where the cooling channel was modelled as a closed contour. Optimization of geometry of the cooling element leads to shape optimization problem, which is treated using "optimize-then-discretize" approach, i.e., while we ultimately discretize the problem for the purpose of a numerical solution, the key components of our approach are derived in the continuous setting. As a consequence, they are independent on the specific discretization applied.

1.1 Mathematical Modelling of Heat Transfer

Heat transfer problems arise in many industrial and environmental processes, particularly in energy utilization, thermal processing, and thermal control. There are three types of heat transfer: heat transfer by means of molecular agitation within a material without any motion of the material as a whole (conduction), heat transfer by mass motion of a fluid such as air or water when the heated fluid is caused to move away from the source of heat, carrying energy with it (convection) and heat transfer through empty space by the propagation of electromagnetic waves (radiation). The problem we are considering in this work is a primarily heat conduction problem, so we start with a brief introduction to heat conduction and its mathematical modelling.

Conduction is the transport of thermal energy in solids and fluids due to short-range atomic interactions, modelled by the heat equation. Here we consider two dimensional (2D) parabolic partial differential equation derived from Fourier's law and conservation of energy:

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

where $\Omega \subset \mathbb{R}^2$ with the boundary $\partial\Omega$, $u(\mathbf{x}, t)$ is a temperature as a function of space and time, $q(\mathbf{x})$ is a prescribed heat sources in the domain Ω and $k > 0$ is the thermal conductivity of a given isotropic material at given conditions (assumed constant).

In problems of heat transfer, it is common to encounter the condition that no heat may enter or leave the boundary of the domain, i.e. that the domain is perfectly insulated. This corresponds to the Neumann boundary conditions where the derivative in the direction given by the unit normal vector $\mathbf{n}$ is zero

$$\frac{\partial u}{\partial n} = 0, \quad \text{on } \partial\Omega.$$

If we want to model a heat flow $f(\mathbf{x})$ on the boundary, we impose inhomogeneous

Neumann conditions $k\frac{\partial u}{\partial n} = f$. Alternatively, we can place the Dirichlet boundary conditions, when the temperature is known at the boundary, or some mixed or more complicated boundary conditions. For instance, we can impose Robin boundary conditions $au + b\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$ for some a, $b \neq 0$, as a linear combination of Dirichlet and Neumann boundary conditions, which arises if heat flux at the boundary is proportional to the temperature.

As times goes on, we arrive at some stationary state when the spatial distribution of temperature does not change any further, i.e. $\frac{\partial u}{\partial t} = 0$, the heat equation (1.1) becomes an elliptic PDE called Poisson's equation

$$-k\Delta u = q, \tag{1.2}$$

which, without the non-homogeneous term, becomes Laplace equation $-\Delta u = 0$. These are so-called steady-state cases and a solution u is the stationary solution.

1.2 Shape Optimization in Heat Transfer

To predict the response of dynamic systems from given parameters, data and source terms requires a mathematical model of the behaviour of the process under investigation and a physical theory linking the state variables of the model to data and parameters. This mathematical model is a so-called direct problem and usually defined by one or more coupled integral, ordinary or partial differential systems and sufficient number of boundary and initial conditions. If any of the parameters necessary to define a direct problem are unknown or known incompletely, an inverse problem arises. It specifies the desired outcomes and computes the unknown parameters and/or the data such that the resulting outcome is as close as possible to the desired one [2].

In shape optimization, the undetermined property is the shape of the domain, i.e.

optimization of the geometry is of primary interest. Introduction to mathematical and computational aspects of shape optimization is given in [3]. In [2] the authors focus on modelling and simulations with application to heat exchanger systems as well as computational methodologies such as finite-difference, finite-element and finite-volume methods, lattice Boltzmann numerical method, boundary element method. Several works which discuss the shape optimization problem are as follows: Huang and Wuchiu [4] study a shape design of interfacial surface between two conductive bodies in a three-dimensional multiple region domains, based on the desired system heat flux and domain volume. The design algorithm utilized the Levenberg-Marquardt method (LMM) and B-spline surface generation. In [5] the simplified conjugate-gradient method (SCGM) combined with the finite element method to optimize the shape of the slug inside the LED package. A three-dimensional inverse design problem is solved in [6] by using the LMM.

This work continues the investigation started in [1], where the case of closed cooling contour was considered. Here, we discuss an open channel with the ends on the boundary of the domain and consider not only some constant temperature of the contour, but also a linear change of the temperature along the channel. Replacing a closed contour (with periodic data) with an open contour with a nonperiodic data leads to a number of technical problems which need to be addressed in order to retain high accuracy of the computations.

The main difficulty in this investigation is related to numerical solution of the boundary integral equation with singular kernel on non-periodic domain, which arises as an equivalent formulation of the direct and adjoint problems. The approach to removing singularity applied in this work is similar to the procedures discussed in [7, 8] and [9]. Moreover, the collocation method from [7] is utilized. However, in the present work we emphasize spectrally accurate approximation of the regularized integral by the Clenshaw-Curtis quadrature.

1.3 Structure of the Thesis

In the next chapter we introduce the mathematical model of our problem and then cast it as a two-dimensional optimization problem. In Chapter 3 we review shape calculus basics and state the optimality conditions followed by the derivation of the cost functional gradient with the help of adjoint variables. An approach to the gradient approximation employing spectral methods is presented in Chapter 4, then the validation of the method is performed and computational examples are discussed in Chapter 5, while conclusions and outlook are deferred to Chapter 6. In Appendix A shape derivation of the arc length coordinate is considered and in Appendix B the alternative formulations of the gradient are discussed.

Chapter 2

Formulation of the Problem

2.1 Mathematical Model

In this work, we consider a simplified steady-state 2D model of our problem:

$$-k\Delta u_1 = q \qquad \text{in } \Omega_1, \tag{2.1a}$$

$$-k\Delta u_2 = q \qquad \text{in } \Omega_2, \tag{2.1b}$$

$$u_1 = u_2 \qquad \text{on } \mathcal{C}, \tag{2.1c}$$

$$k\Big(\frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n}\Big) = \gamma(u_1 - u_0) \quad \text{on } \mathcal{C}, \tag{2.1d}$$

$$k\frac{\partial u}{\partial n} = 0 \qquad \text{on } \partial\Omega, \tag{2.1e}$$

where $u_1 = u\chi_{\Omega_1}$, $u_2 = u\chi_{\Omega_2}$ and χ is the characteristic function of the corresponding set.

We assume that the battery pack is given by a 2D square region $\Omega \in \mathbb{R}^2$ with $\partial\Omega$ denoting its boundary. Our goal is to remove heat from the domain Ω by optimizing the shape of the cooling channel $\mathcal{C}$, which in our model is simplified to be a smooth

curve of total length L and a temperature u_0 (analogue of the cooling fluid temperature in the physical model).

Since the cooling channel goes across the battery pack, the endpoints of our contour $\mathcal{C}$ are attached to the boundary $\partial\Omega$. In addition, the curve is considered to be perpendicular to the boundary at these points to guarantee that it stays in the domain Ω during entire optimization process.

As shown in Figure 2.1, the contour $\mathcal{C}$ divides the domain into two subdomains Ω_1, $\Omega_2 \in \Omega$, with the boundaries $\mathcal{C} \cup \partial\Omega_1$ and $\mathcal{C} \cup \partial\Omega_2$ respectively.

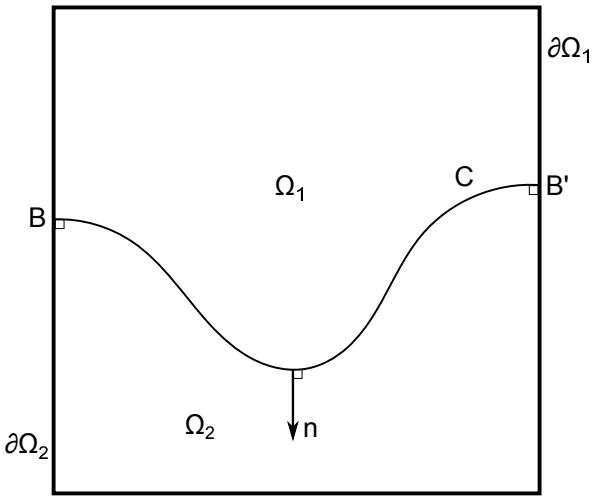


Figure 2.1: Sketch of the domain

Let us consider system (2.1) in more detail:

- The governing PDEs (2.1a) and (2.1b) describe the distribution of the temperature u in the domains Ω_1 and Ω_2 (i.e. $u_1 = u\chi_{\Omega_1}$ and $u_2 = u\chi_{\Omega_2}$ respectively), corresponding to the distribution of prescribed heat sources $q : \Omega \longrightarrow \mathbb{R}$ and the boundary conditions (2.1c) – (2.1e). We assume that $q(\mathbf{x})$ is at least a continuous function of $\mathbf{x} \in \Omega$. The thermal conductivity coefficient $k > 0$ depends on material properties of the battery pack.

- The boundary condition (2.1c) means that the temperature u is continuous across the contour $\mathcal{C}$.

- The relation (2.1d) is based on Newton's cooling law and the conservation of energy, showing that the net flux across the cooling channel is directly proportional to the difference between the outside temperature $u|_{\mathcal{C}}$ on the contour and the temperature of the channel $u_0 = u_0(s)$, $s \in \mathcal{C}$. Unlike thermal conductivity k, heat transfer coefficient γ is not a property of material and in our problem its considered to be a known constant.

- The reference temperature $u_0 = u_0(s)$ of the cooling contour $\mathcal{C}$ is varying along the contour. We assume that u_0 increases linearly with the length corresponding to the coolant liquid heating up as it absorbs heat, i.e.

$$u_0(s) = u_{in} + \frac{u_{out} - u_{in}}{L} s, \quad s \in [0, L], \tag{2.2}$$

  where u_{in} and u_{out} are the prescribed temperatures at the inlet and outlet and L is the length of the contour.

- Condition (2.1e), where $\mathbf{n}$ is the unit normal to the boundary $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ and pointing out of the domain, states that the boundary is thermally insulated.

2.2 Problem of Optimal Shape Design

Our optimization problem consists in finding a smooth curve $\mathcal{C}$ which minimizes the difference between the temperature u, corresponding to the given curve $\mathcal{C}$, and some prescribed temperature $\bar{u}$. The difference is taken in the least square sense over some subdomain $A \subseteq \Omega$. Thus, the cost functional representing the performance criterion

we want to optimize is defined as:

$$\mathcal{J}(\mathcal{C}, u) = \frac{1}{2}\int_A (u - \bar{u})^2 \, d\mathbf{x}. \tag{2.3}$$

The cost functional depends on u explicitly, whereas u itself depends on the shape of the contour $\mathcal{C}$ through the system of equations (2.1), which allows us to think of the cost functional as a function of one variable $\mathcal{C}$ only. In addition, since in the real-life applications the contour $\mathcal{C}$ representing the cooling element may not be arbitrary, we also consider the additional constraint on the contour length, namely,

$$\int_{\mathcal{C}} ds = L_0,$$

where L_0 is some prescribed length. Then, the problem of finding an optimal curve $\widehat{\mathcal{C}}$ can be formulated as PDE-constrained optimization in the following way:

$$\begin{aligned} &\min_{\mathcal{C}} \mathcal{J}(\mathcal{C}) \\ &\text{subject to: system (2.1)} \\ &\int_{\mathcal{C}} ds = L_0. \end{aligned} \tag{2.4}$$

Let us mention here the key elements required to address the given optimization problem. An approximation of a minimizer $\mathcal{C}$ can be found using the steepest descent gradient method, given some initial guess $\mathcal{C}_0$, i.e:

$$\begin{aligned} \mathbf{x}_{\mathcal{C}^{(i+1)}} &= \mathbf{x}_{\mathcal{C}^{(i)}} - \tau^{(i)} \nabla \mathcal{J}(\mathcal{C}^{(i)}), \quad i = 0, 1, 2, ... \\ \mathbf{x}_{\mathcal{C}^{(0)}} &= \mathbf{x}_{\mathcal{C}_0}, \end{aligned} \tag{2.5}$$

where the points $\mathbf{x}_{\mathcal{C}^{(i)}}$ represent the contour $\mathcal{C}^{(i)}$.

The gradient-based optimization is chosen since it gives a stable numerical method

even when no good initial guess for the control variable $\mathcal{C}$ is available. To apply the scheme (2.5) to our problem we need to define the gradient $\nabla\mathcal{J}$. This can be done by expressing $\nabla\mathcal{J}$ as a function of the solution of a suitably-defined adjoint system. This is a standard approach to solution of PDE-constrained optimization problems introduced in [10], which involves the shape calculus and the Riesz representation theorem [11] as the key tools. In addition to methods of the shape-differential calculus, our computational approach will be based on the combination of spectral methods and the boundary integral equation technique discussed in Section 4.4. The step $\tau^{(i)}$ is determined by the line minimization method introduced in Section 4.5.1.

As can be seen from (2.5), the central ingredient of the algorithm is computation of the cost functional gradient $\nabla\mathcal{J}$ with respect to our control variable, namely the shape of the contour $\mathcal{C}$. In the next chapter we discuss what is the optimality condition which guarantees the optimal shape of the contour $\mathcal{C}$ and how to define a suitable adjoint system which ensures that the gradient $\nabla\mathcal{J}$ respects the PDE constraints.

Finally, the main steps of the algorithm are:

1. Choose initial guess $\mathcal{C}^{(0)} = \mathcal{C}_0$, set iteration index to $i := 0$.
2. Solve direct problem (2.1).
3. Solve adjoint problem.
4. Compute the gradient $\nabla\mathcal{J}$.
5. Find the optimal step $\tau^{(i)}$.
6. Update $\mathcal{C}^{(i+1)}$ by (2.5).
Set $i := i + 1$.
Continue from step **2** until optimality conditions (to be determined in Section 3.2) is satisfied.

Chapter 3

Shape Optimization Approach

At the beginning of this chapter we clarify how to describe changes in the geometry and how to differentiate functions with varying domains of their definition. On the basis of these results we obtain the necessary optimality conditions for our optimization problem. Next, we demonstrate the derivation of a special adjoint system which provides us with an expression for the gradient - a crucial instrument required by the gradient descent method (Section 3.3) for the numerical treatment of the problem.

3.1 Sensitivity Analysis in Shape Optimization

Sensitivity analysis in shape optimization deals with computations of derivatives of solutions to direct problems and cost functionals with respect to shape variations. Based on this information we can derive necessary optimality conditions satisfied by solution to optimization problems. Sensitivity analysis also plays an important role in computations, providing us with the gradient information required by the gradient type methods. The key tool of sensitivity analysis, the material derivative approach, is presented with an assumption that all data we need are sufficiently smooth.

3.1.1 Shape Calculus: Material and Shape Derivative

In this section we introduce the notion of the shape and material derivatives and provide the differentiation results for some model partial differential and integral operators based on the monograph [3].

Let Λ be a bounded open set of $\mathbb{R}^2$, with a regular boundary Γ and let $\psi(x) : \Lambda \longrightarrow \mathbb{R}$ be a real function.

Consider the following model equations:

$$A(\psi) = f \quad \text{in } \Lambda, \tag{3.1a}$$

$$B(\psi) = g \quad \text{on } \Gamma. \tag{3.1b}$$

Here A and B denote some linear partial differential operators with constant coefficients, for example, A could be the Laplace operator and B is the Neumann or Dirichlet boundary conditions, f and g are some functions defined on $\mathbb{R}^2$.

As a prototype of cost functionals consider the integrals:

$$J(\Lambda) = \int_{\Lambda} \psi(\mathbf{x})\, d\mathbf{x}, \tag{3.2a}$$

$$G(\Gamma) = \int_{\Gamma} \psi(\mathbf{x})\, ds. \tag{3.2b}$$

We are interested in the behaviour of ψ, J and G, more specifically, how to differentiate these functions if the domain of their definition varies. Let us denote a new configuration of the domain Λ at some pseudo-time $\tau \in \mathbb{R}$ as Λ_τ. To formalize the

change in the geometry of Λ we introduce a mapping

$$\begin{aligned} T_\tau &: \Lambda \longrightarrow \mathbb{R}^2 \\ T_\tau &= I \cdot \mathbf{x} + \tau \boldsymbol{\mathcal{V}} \quad \forall \mathbf{x} \in \Lambda, \end{aligned}$$

where I is a $d \times d$ identity operator and $\boldsymbol{\mathcal{V}} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ is a vector field of an appropriate smoothness. Note that when $\tau = 0$, we have $T_\tau = I$, so we can write $\Lambda = \Lambda_0 = T_0(\Lambda)$ and $\Lambda_\tau = T_\tau(\Lambda)$. Moreover, for τ small enough, T_τ is close to identity I and Λ_τ is an open set with regular boundary $\Gamma_\tau = T_\tau(\Gamma)$, close to Λ. In this set-up the function $\psi_\tau : \Lambda_\tau \longrightarrow \mathbb{R}^2$ can be viewed as

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

Then, its material derivative is given by the chain rule as total derivative at $\tau = 0$:

$$\dot{\psi} = \frac{d}{d\tau} \psi(\tau, \mathbf{x} + \tau \boldsymbol{\mathcal{V}}) \Big|_{\tau=0} = \Big[\frac{\partial \psi(\tau, \mathbf{x}_\tau)}{\partial \tau} + \boldsymbol{\nabla} \psi(\tau, \mathbf{x}_\tau) \cdot \boldsymbol{\mathcal{V}}(\mathbf{x}) \Big] \Big|_{\tau=0} = \psi' + \boldsymbol{\nabla} \psi \cdot \boldsymbol{\mathcal{V}}, \tag{3.4}$$

where the partial derivative $\psi' := \frac{\partial \psi}{\partial \tau} \Big|_{\tau=0}$ is known as the shape derivative. Note that the symbol ψ now has two meanings: it designates the function ψ defined by (3.3) and designates its restriction to the pseudo-time level $\tau = 0$. Under certain smoothness assumptions the shape and spatial derivatives commute [3]:

$$\left(\frac{\partial \psi}{\partial x_i} \right)' = \frac{\partial}{\partial x_i} (\psi').$$

So now, if our model PDE system (3.1) depends on τ, i.e.:

$$A\Big(\psi(\tau, \mathbf{x}_\tau)\Big) = f_\tau \quad \text{in } \Lambda_\tau, \tag{3.5a}$$

$$B\Big(\psi(\tau, \mathbf{x}_\tau)\Big) = g_\tau \quad \text{on } \Gamma_\tau, \tag{3.5b}$$

we can differentiate it by taking the shape derivative of both sides of (3.5) and exchange the order of the shape and spatial derivatives on the left hand side (LHS) of the equations which leads to the so-called sensitivity or perturbation PDEs:

$$A(\psi') = f' \quad \text{in } \Lambda, \tag{3.6a}$$

$$B(\psi') = g' \quad \text{on } \Gamma. \tag{3.6b}$$

Next, we provide the differentiation results for integrals (3.2) in which both the domains and integrands depend on τ, i.e.:

$$J(\Lambda_\tau) = \int_{\Lambda_\tau} \psi(\tau, \mathbf{x}_\tau)\, d\mathbf{x}_\tau,$$

$$G(\Gamma_\tau) = \int_{\Gamma_\tau} \psi(\tau, \mathbf{x}_\tau)\, ds_\tau.$$

The material derivatives of these integrals can be viewed as the directional derivatives characterizing the behaviour of J and G when Λ 'moves' in the direction $\boldsymbol{\mathcal{V}}$ defining T_τ.

Lemma 1.

$$\dot{J}(\Lambda; \boldsymbol{\mathcal{V}}) = \int_\Lambda \dot{\psi}\, d\mathbf{x} + \int_\Lambda \psi\, div \boldsymbol{\mathcal{V}}\, d\mathbf{x}, \tag{3.7a}$$

$$\dot{J}(\Lambda; \boldsymbol{\mathcal{V}}) = \int_\Lambda \psi'\, d\mathbf{x} + \int_\Gamma \psi \boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, d\mathbf{x}, \tag{3.7b}$$

$$\dot{G}(\Gamma; \boldsymbol{\mathcal{V}}) = \int_\Gamma \psi'\, ds + \int_\Gamma \left(\frac{\partial \psi}{\partial n} + \kappa \psi \right) \boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, ds, \tag{3.7c}$$

where $\dot{\psi} := \dot{\psi}(0, \mathbf{x})$ *and* $\psi' := \psi'(0, \mathbf{x})$ *are the material and shape derivatives of* ψ *respectively and* κ *denotes the curvature of* Γ.

The proof of this Lemma can be found in [3].

In case of more complicated integrals, for example

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

where the integrand F is a function of ψ and pseudo-time τ, the material derivative of $\mathcal{E}$ can be derived by applying the results of Lemma 1 and the chain rule:

$$\dot{\mathcal{E}} = \left.\frac{d\mathcal{E}}{d\tau}\right|_{\tau=0} = \int_\Lambda \left\{ \frac{\partial F(0,\psi)}{\partial \tau} + \frac{\partial F(0,\psi)}{\partial \psi} \cdot \psi' \right\} d\mathbf{x} + \int_\Gamma F(0,\psi)\, \boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, d\mathbf{x}. \tag{3.8}$$

In particular, if F does not depend on τ explicitly, we get:

$$\dot{\mathcal{E}} = \int_\Lambda \frac{\partial F(0,\psi)}{\partial \psi} \cdot \psi'\, d\mathbf{x} + \int_\Gamma F(0,\psi)\, \boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, d\mathbf{x}. \tag{3.9}$$

So far, we have discussed the results on curvilinear integrals over closed contours. We conclude the discussion of the shape calculus by presenting the result for the case of integrals defined over open contours [12].

Theorem 1. *Let ψ be a smooth function defined on perturbations $\Gamma(\tau, \boldsymbol{\mathcal{V}})$ of a smooth arc $\Gamma(0) := \Gamma(0, \boldsymbol{\mathcal{V}}) = \widehat{BB'}$. Then*

$$\Big(\int_{\Gamma(\tau,\boldsymbol{\mathcal{V}})} \psi\, ds \Big)' = \int_{\Gamma(0)} \psi'\, ds + \int_{\Gamma(0)} \Big(\kappa\psi + \frac{\partial \psi}{\partial n} \Big) \boldsymbol{\mathcal{V}} \cdot \mathbf{n}\, ds + [\, \psi\, \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\,] \Big|_B^{B'}, \tag{3.10}$$

where $\boldsymbol{\nu}$ is the unit vector tangent to $\Gamma(0)$.

3.2 Optimality Conditions

3.2.1 Optimality Conditions without Length Constraint

Recall that we are looking for an optimal curve $\widehat{\mathcal{C}}$, such that the minimum of the cost functional (2.4) is achieved.

The necessary condition that characterizes the minimizer $\widehat{\mathcal{C}}$ of the cost functional is the vanishing of its shape derivative

$$\mathcal{J}'(\widehat{\mathcal{C}};\zeta) = 0 \tag{3.11}$$

for all perturbations ζ affecting the contour. In particular, in the case when no length constraint imposed on the contour, we have:

$$\mathcal{J}'(\widehat{\mathcal{C}};\zeta) = \int_A (u-\bar{u})u'\,d\mathbf{x} = 0. \tag{3.12}$$

Note that the solution u will depend on the shape of the contour $\mathcal{C}$, i.e. $u = u(\mathcal{C})$. We emphasize that, while the elliptic PDEs and the boundary conditions in system (2.1) are linear in the dependent variables u_1 and u_2, problem (2.1) is in fact geometrically nonlinear with respect to the shape of the contour $\mathcal{C}$. Therefore, problem (2.4) may admit several local minimizers and so the condition (3.12) can characterize local minimizer $\widehat{\mathcal{C}}$ only.

On the other hand, according to the Riesz representation theorem [13], there is a unique element $\nabla\mathcal{J} \in \mathcal{X}$ which satisfies the identity

$$\mathcal{J}'(\widehat{\mathcal{C}};\zeta) = \langle \nabla\mathcal{J}, \zeta\rangle_{\mathcal{X}} \quad \forall \zeta \in \mathcal{X}, \tag{3.13}$$

where $\langle\,\cdot\,,\,\cdot\,\rangle_{\mathcal{X}}$ denotes the inner product in some suitable Hilbert space $\mathcal{X}$. Since

$\zeta \in \mathcal{X}$ is arbitrary, combining identity (3.13) with the necessary conditions (3.12), we conclude that our optimality condition is

$$\nabla \mathcal{J} = 0. \tag{3.14}$$

From here we restrict ourselves to the space of square-integrable functions $\mathcal{X} = L^2(\Omega)$, but later we discuss the effect of choosing $\mathcal{X}$ to be a Sobolev space.

An expression for the gradient $\nabla \mathcal{J}$ cannot be obtained in a straightforward manner from (3.13) and (3.12), since $\mathcal{J}'(\widehat{\mathcal{C}}; \zeta)$ is not in the form consistent with $\langle \nabla^{L^2} \mathcal{J}, \zeta \rangle_{L^2}$ yet. The derivation of the gradient using suitably-defined adjoint variables is described in Section 3.3.2.

3.2.2 Optimality Conditions in the Presence of Length Constraint

Now, let us consider the case when the contour is subject to length constraint $\int_C ds = L_0$. We can take this constraint into account by adding a weight (penalty) term to the cost functional as follows:

$$\mathcal{J}_{L_0}(\mathcal{C}) = \mathcal{J}(\mathcal{C}) + \frac{1}{2}\,\alpha \left(\int_C ds - L_0 \right)^2, \tag{3.15}$$

where $\mathcal{J}(\mathcal{C})$ is the same as in (2.4), $\alpha \in \mathbb{R}^+$ is a constant determining the penalty on the violation of the length constraint. We note that the more systematic formulation of the constrained problem using Lagrange multipliers could be presented, but, given the geometric nonlinearity of the constraint $\int_C ds = L_0$, the Lagrange multipliers can be rather hard to compute accurately, so for simplicity in this work we chose weighted sum formulation (3.15).

Using (3.12) as the derivative of the first term and applying (3.7c) to differentiate the second term in the gradient we get:

$$\mathcal{J}'_{L_0}(\widehat{\mathcal{C}};\zeta) = \mathcal{J}'(\widehat{\mathcal{C}};\zeta) + \alpha\Big(\int_C ds - L_0\Big)\int_C \kappa\zeta\, ds. \tag{3.16}$$

Thus, applying identity (3.13) to the first term and writing the second term as L^2 inner product, which is already in the Riesz form, we obtain the following expression:

$$\mathcal{J}'_{L_0}(\widehat{\mathcal{C}};\zeta) = \langle \nabla^{L^2}\mathcal{J}, \zeta\rangle_{L^2} + \left\langle \alpha\Big(\int_C ds - L_0\Big)\kappa, \zeta \right\rangle_{L^2}. \tag{3.17}$$

If we denote by $\nabla^{L^2}_{L_0}\mathcal{J}$ the gradient corresponding to the length-constraint case, by the same argument as above we can write

$$\mathcal{J}'_{L_0}(\widehat{\mathcal{C}};\zeta) = \langle \nabla^{L^2}_{L_0}\mathcal{J},\, \zeta\rangle_{L^2} \tag{3.18}$$

and our optimality condition is

$$\mathcal{J}'_{L_0}(\widehat{\mathcal{C}};\zeta) = 0 \quad \forall \zeta \in H^1.$$

Combining with (3.17) we derive the gradient expression as

$$\nabla^{L^2}_{L_0}\mathcal{J} = \nabla^{L^2}\mathcal{J} + \alpha\Big(\int_C ds - L_0\Big)\kappa$$

and an equivalent optimality condition can be seen to be

$$\nabla^{L^2}_{L_0}\mathcal{J} = 0.$$

The optimality conditions derived in this section are utilized by the gradient descent

method to determine that the minimizer of the cost functional, i.e. to find the locally optimal contour $\widehat{\mathcal{C}}$.

3.3 Gradient Descent Method

To find a local minimum of the cost functional we need to make steps in the steepest descent direction, i.e. in the direction of $-\nabla\mathcal{J}$. Up to this point, we have described the optimality condition that should be satisfied by the gradient, but were not able to provide an explicit expression for the gradient itself. The connection between the shape derivative of the cost functional and its gradient was established with the help of the Riesz formulation. In this section we show that some useful relations can be derived by analysing the sensitivity of each variable involved in the problem definition. Moreover, together with adjoint analysis, these relations provide a formula for the gradient as required by the algorithm.

3.3.1 Perturbation PDE System

Given a vector field $\boldsymbol{\mathcal{V}}$ on Ω such that $\boldsymbol{\mathcal{V}}(\partial\Omega) = 0$, the flow map T_τ introduced in Section 3.1.1 can be written as $T_\tau = I + \tau\,\boldsymbol{\mathcal{V}}$, where I is the identity operator. Here we employ the assumption that the end points of the contour $\mathcal{C}$ are attached to the domain boundary $\partial\Omega$ at the right angle (see Figure 2.1), so the terms proportional to $(\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu})|_B^{B'}$ in (3.10) vanish identically. Therefore, only the normal component $\zeta := \boldsymbol{\mathcal{V}}\cdot\mathbf{n}$ of the perturbation velocity field on the contour $\mathcal{C}$ plays a role in expression for shape differentials (3.10).

Thus, $\mathcal{C}_\tau$ can now be defined as the set of points $\mathbf{x}_\mathcal{C}(\tau)$ such that:

$$\mathbf{x}_\mathcal{C}(\tau) = \mathbf{x} + \tau\zeta\mathbf{n}, \quad \mathbf{x}_\mathcal{C}(\tau) \in \mathcal{C}_\tau, \quad \mathbf{x} \in \mathcal{C}.$$

Since the shape of the domains Ω_1, Ω_2 depends on τ, then the functions u_1 and u_2 defined on these domains are functions of τ themselves and can be denoted $u_{1\tau}$, $u_{2\tau}$.

Thus, we can interpret our problem as parameter-dependent PDE system:

$$-k\Delta u_{1\tau} = q\chi_{\Omega_{1\tau}} \qquad \text{in } \Omega_{1\tau}, \tag{3.19a}$$

$$-k\Delta u_{2\tau} = q\chi_{\Omega_{2\tau}} \qquad \text{in } \Omega_{2\tau}, \tag{3.19b}$$

$$u_{1\tau} = u_{2\tau} \qquad \text{on } \mathcal{C}_\tau, \tag{3.19c}$$

$$k\Big(\frac{\partial u_{2\tau}}{\partial n} - \frac{\partial u_{1\tau}}{\partial n}\Big) = \gamma(u_{1\tau} - u_{0\tau}) \quad \text{on } \mathcal{C}_\tau, \tag{3.19d}$$

$$k\frac{\partial u_\tau}{\partial n} = 0 \qquad \text{on } \partial\Omega, \tag{3.19e}$$

where $u_{1\tau} = u\chi_{\Omega_{1\tau}}$ and $u_{2\tau} = u\chi_{\Omega_{2\tau}}$.

As will be seen later, the shape derivative of u' is an important element of the adjoint analysis. The set of relations satisfied by u' is obtained by taking material derivative of system (3.19) with respect to τ at $\tau = 0$ and then using formula (3.4).

Theorem 2. *The shape derivative u' of the solution u satisfies the following perturbation PDE system:*

$$-k\Delta u_1' = 0 \qquad \textit{in } \Omega_1, \tag{3.20a}$$

$$-k\Delta u_2' = 0 \qquad \textit{in } \Omega_2, \tag{3.20b}$$

$$u_2' - u_1' = \Big(\frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n}\Big)\zeta \qquad \textit{on } \mathcal{C}, \tag{3.20c}$$

$$k\Big(\frac{\partial u_2'}{\partial n} - \frac{\partial u_1'}{\partial n}\Big) - \gamma u_1' = \gamma\Big[\frac{\partial u_1}{\partial n} + \kappa(u_1 - u_0)\Big]\zeta - \gamma u_0' \quad \textit{on } \mathcal{C}, \tag{3.20d}$$

$$k\frac{\partial u'}{\partial n} = 0 \qquad \textit{on } \partial\Omega, \tag{3.20e}$$

where $\zeta = \boldsymbol{\mathcal{V}} \cdot \mathbf{n}$.

Details of this derivation is provided in [1].

3.3.2 Adjoint PDE System

The shape derivative of the cost functional (3.12) can now be transformed to a form consistent with Riesz identity (3.13) by introducing a suitable adjoint operator and the corresponding adjoint state.

Thus, we need to find the adjoint system

$$\begin{aligned} A^*(u^*) =& f^* \quad \text{in } \Omega, \\ B^*(u^*) =& g^* \quad \text{on } \mathcal{C}, \\ C^*(u^*) =& h^* \quad \text{on } \partial\Omega, \end{aligned}$$

such that $\int_A (u-\bar{u})u' d\,\mathbf{x} = \int_{\mathcal{C}} \boldsymbol{\nabla}^{L^2}\mathcal{J}\zeta\, ds$, which holds for some $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ depending on u^* and u. Suppose

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

$$u_2^* : \Omega_2 \longrightarrow \mathbb{R}$$

are two shape-independent functions, i.e. $(u_1^*)' = 0$ and $(u_2^*)' = 0$. Now, to define the adjoint system we need to perform a series of transformations, such as integrating the governing equations (3.19a) and (3.19b) against, respectively, u_1^* and u_2^* over Ω_1 and Ω_2, performing integration by parts and shape differentiation. First, we multiply (3.19a) by u_1^*, integrate over Ω_1 and then apply Green's formula:

$$\begin{aligned} & -k\int_{\Omega_{1\tau}} \Delta\, u_{1\tau} u_1^*\, d\mathbf{x} \\ &= -k\left[\int_{\partial\Omega_1} \frac{\partial u_{1\tau}}{\partial n} u_1^*\, ds + \int_{\mathcal{C}_\tau} \frac{\partial u_{1\tau}}{\partial n} u_1^*\, ds - \int_{\Omega_{1\tau}} \nabla\, u_{1\tau}\cdot\nabla\, u_1^*\, d\mathbf{x}\right] \\ &= \int_{\Omega_{1\tau}} q u_1^*\, d\mathbf{x}. \end{aligned}$$

Next, we take the shape derivative of both sides of the equation at $\tau = 0$ and consider

(3.19e) which cancels the first integral on the right hand side:

$$
\begin{aligned}
&- k\frac{d}{d\tau}\left[\int_{\mathcal{C}_\tau}\frac{\partial u_{1\tau}}{\partial n}u_1^*\,ds - \int_{\Omega_{1\tau}}\nabla\, u_{1\tau}\cdot\nabla\, u_1^*\,d\mathbf{x}\right]\Bigg|_{\tau=0}\\
&= -k\Bigg\{\int_{\mathcal{C}}\Big(\frac{\partial u_1}{\partial n}u_1^*\Big)'\,ds + \int_{\mathcal{C}}\left(\frac{\partial}{\partial n}\Big(\frac{\partial u_1}{\partial n}u_1^*\Big) + \kappa\frac{\partial u_1}{\partial n}u_1^*\right)\zeta\,ds - \int_{\Omega_1}\nabla\, u_1'\cdot\nabla\, u_1^*\,d\mathbf{x}\\
&- \int_{\Omega_1}\nabla\, u_1\cdot\nabla\,(u_1^*)'\,d\mathbf{x} - \int_{\partial\Omega_1}\nabla\, u_1\cdot\nabla\, u_1^*\zeta\,ds - \int_{\mathcal{C}}\nabla\, u_1\cdot\nabla\, u_1^*\zeta\,ds\Bigg\}\\
&= \int_{\Omega_1}(qu_1^*)'\,d\mathbf{x} + \int_{\partial\Omega_1}qu_1^*\,\zeta\,ds + \int_{\mathcal{C}}qu_1^*\,\zeta\,ds.
\end{aligned}
$$

Consider the fact that $(u_1^*)' = 0$, $q' = 0$ and $\zeta = 0$ on $\partial\Omega_1$ and apply Green's formula to $\int_{\Omega_1}\nabla\, u_1'\cdot\nabla\, u_1^*\,d\mathbf{x}$:

$$
\begin{aligned}
&- k\Bigg\{\int_{\mathcal{C}}\frac{\partial u_1'}{\partial n}u_1^*\,ds + \int_{\mathcal{C}}\left(\frac{\partial}{\partial n}\Big(\frac{\partial u_1}{\partial n}u_1^*\Big) + \kappa\frac{\partial u_1}{\partial n}u_1^*\right)\zeta\,ds + \int_{\Omega_1}u_1'\Delta\, u_1^*\,d\mathbf{x}\\
&- \int_{\mathcal{C}}\frac{\partial u_1^*}{\partial n}u_1'\,ds - \int_{\partial\Omega_1}\frac{\partial u_1^*}{\partial n}u_1'\,ds - \int_{\mathcal{C}}\nabla\, u_1\cdot\nabla\, u_1^*\zeta\,ds\Bigg\}\\
&= \int_{\mathcal{C}}qu_1^*\,\zeta\,ds.
\end{aligned}
$$

After grouping the terms we get

$$
\begin{aligned}
&- k\Bigg\{\int_{\Omega_1}u_1'\Delta\, u_1^*\,d\mathbf{x} + \int_{\mathcal{C}}\left(\frac{\partial u_1'}{\partial n}u_1^* - \frac{\partial u_1^*}{\partial n}u_1'\right)ds\\
&+ \int_{\mathcal{C}}\left(\frac{\partial}{\partial n}\Big(\frac{\partial u_1}{\partial n}u_1^*\Big) + \kappa\frac{\partial u_1}{\partial n}u_1^* - \nabla\, u_1\cdot\nabla\, u_1^*\right)\zeta\,ds - \int_{\partial\Omega_1}\frac{\partial u_1^*}{\partial n}u_1'\,ds\Bigg\} \qquad (3.21)\\
&= \int_{\mathcal{C}}qu_1^*\,\zeta\,ds.
\end{aligned}
$$

Next, we multiply (3.19b) by u_2^* and integrate over Ω_2 and then apply Green's

formula:

$$
\begin{aligned}
&- k \int_{\Omega_{2\tau}} \Delta\, u_{2\tau} u_2^*\, d\mathbf{x} \\
&= -k \left[\int_{\partial\Omega_2} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, ds - \int_{\mathcal{C}_\tau} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, ds - \int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, u_2^*\, d\mathbf{x} \right] \\
&= \left[\text{ by (3.19}e\text{)}, \quad \frac{\partial u_{2\tau}}{\partial n} u_2^* = 0 \text{ on } \partial\Omega_2 \right] \\
&= k \left[\int_{\mathcal{C}_\tau} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, ds + \int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, u_2^*\, d\mathbf{x} \right] \\
&= \int_{\Omega_{2\tau}} q u_2^*\, d\mathbf{x}.
\end{aligned}
$$

Then, we take the shape derivative of both sides of the equation at $\tau = 0$:

$$
\begin{aligned}
&k \frac{d}{d\tau} \left[\int_{\mathcal{C}_\tau} \frac{\partial u_{2\tau}}{\partial n} u_2^*\, ds + \int_{\Omega_{2\tau}} \nabla\, u_{2\tau} \cdot \nabla\, u_2^*\, d\mathbf{x} \right] \Bigg|_{\tau=0} \\
&= k \Bigg\{ \int_{\mathcal{C}} \Big(\frac{\partial u_2}{\partial n} u_2^* \Big)'\, ds + \int_{\mathcal{C}} \left(\frac{\partial}{\partial n} \Big(\frac{\partial u_2}{\partial n} u_2^* \Big) + \kappa \frac{\partial u_2}{\partial n} u_2^* \right) \zeta\, ds + \int_{\Omega_2} \nabla\, u_2' \cdot \nabla\, u_2^*\, d\mathbf{x} \\
&+ \int_{\Omega_2} \nabla\, u_2 \cdot \nabla\, (u_2^*)'\, d\mathbf{x} + \int_{\partial\Omega_2} \nabla\, u_2 \cdot \nabla\, u_2^* \zeta\, ds - \int_{\mathcal{C}} \nabla\, u_2 \cdot \nabla\, u_2^* \zeta\, ds \Bigg\} \\
&= \int_{\Omega_2} (q u_2^*)'\, d\mathbf{x} + \int_{\partial\Omega_2} q u_2^* \zeta\, ds - \int_{\mathcal{C}} q u_2^* \zeta\, ds.
\end{aligned}
$$

Combining with $(u_2^*)' = 0$, $q' = 0$ and $\zeta = 0$ on $\partial\Omega_2$, applying Green's formula to $\int_{\Omega_2} \nabla\, u_2' \cdot \nabla\, u_2^*\, d\mathbf{x}$ and rearranging the terms:

$$
\begin{aligned}
&k \Bigg\{ - \int_{\Omega_2} u_2' \Delta\, u_2^*\, d\mathbf{x} + \int_{\mathcal{C}} \left(\frac{\partial u_2'}{\partial n} u_2^* - \frac{\partial u_2^*}{\partial n} u_2' \right) ds \\
&+ \int_{\mathcal{C}} \left(\frac{\partial}{\partial n} \Big(\frac{\partial u_2}{\partial n} u_2^* \Big) + \kappa \frac{\partial u_2}{\partial n} u_2^* - \nabla\, u_2 \cdot \nabla\, u_2^* \right) \zeta\, ds + \int_{\partial\Omega_2} \frac{\partial u_2^*}{\partial n} u_2'\, ds \Bigg\} \qquad (3.22) \\
&= - \int_{\mathcal{C}} q u_2^* \zeta\, ds.
\end{aligned}
$$

Finally, we add (3.21) and (3.22), and then group the terms to obtain:

$$k\Bigg\{ -\int_{\Omega_1} u_1'\Delta\, u_1^*\, d\mathbf{x} - \int_{\Omega_2} u_2'\Delta\, u_2^*\, d\mathbf{x} + \int_{\mathcal{C}} \left(\frac{\partial u_2'}{\partial n} u_2^* - \frac{\partial u_1'}{\partial n} u_1^* + \frac{\partial u_1^*}{\partial n} u_1' - \frac{\partial u_2^*}{\partial n} u_2' \right) ds$$
$$+ \int_{\mathcal{C}} \left[\frac{\partial}{\partial n}\Big(\frac{\partial u_2}{\partial n} u_2^* - \frac{\partial u_1}{\partial n} u_1^* \Big) + \kappa \Big(\frac{\partial u_2}{\partial n} u_2^* - \frac{\partial u_1}{\partial n} u_1^* \Big) + \Big(\nabla\, u_1 \cdot \nabla\, u_1^* - \nabla\, u_2 \cdot \nabla\, u_2^* \Big) \right] \zeta\, ds$$
$$+ \int_{\partial\Omega_2} \frac{\partial u_2^*}{\partial n} u_2'\, ds + \int_{\partial\Omega_1} \frac{\partial u_1^*}{\partial n} u_1'\, ds \Bigg\} = \int_{\mathcal{C}} (q u_1^* - q u_2^*)\, \zeta\, ds. \tag{3.23}$$

At this point let us choose

$$-k\Delta u_1^* = (u - \bar{u})\chi_{A_1}, \tag{3.24}$$
$$-k\Delta u_2^* = (u - \bar{u})\chi_{A_2}, \tag{3.25}$$

where $A_1 = A \cap \Omega_1$ and $A_2 = A \cap \Omega_2$. Then, the first two terms in (3.23) can be transformed into

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

We also choose the following boundary conditions:

$$\begin{aligned} \frac{\partial u_1^*}{\partial n} &= 0 \quad \text{on } \partial\Omega_1, \\ \frac{\partial u_2^*}{\partial n} &= 0 \quad \text{on } \partial\Omega_2, \end{aligned} \tag{3.26}$$

and so (3.23) can be written as:

$$\mathcal{J}'(\mathcal{C};\zeta\mathbf{n}) = B,$$

where

$$\begin{aligned} B = &- k\int_{\mathcal{C}}\left(\frac{\partial u_2'}{\partial n}u_2^* - \frac{\partial u_1'}{\partial n}u_1^* + \frac{\partial u_1^*}{\partial n}u_1' - \frac{\partial u_2^*}{\partial n}u_2'\right)ds + \int_{\mathcal{C}}(qu_1^* - qu_2^*)\,\zeta\,ds \\ &- k\int_{\mathcal{C}}\left[\frac{\partial}{\partial n}\Big(\frac{\partial u_2}{\partial n}u_2^* - \frac{\partial u_1}{\partial n}u_1^*\Big) + \kappa\Big(\frac{\partial u_2}{\partial n}u_2^* - \frac{\partial u_1}{\partial n}u_1^*\Big)\right. \\ &\left. + \Big(\nabla u_1\cdot\nabla u_1^* - \nabla u_2\cdot\nabla u_2^*\Big)\right]\zeta\,ds. \end{aligned} \tag{3.27}$$

Now, we focus on the first term in (3.27) which is the only one not in the Riesz form yet. Since it is not immediately obvious how adjoint boundary conditions should be chosen in this case, we follow the algorithm for symbolic generation of adjoint systems from [14]. First, let us define two symbolic vectors:

$$\begin{aligned} \mathbf{V}' &:= \left[u_1', \frac{\partial u_1'}{\partial n}, u_2', \frac{\partial u_2'}{\partial n}\right]^T, \\ \mathbf{V}^* &:= \left[u_1^*, \frac{\partial u_1^*}{\partial n}, u_2^*, \frac{\partial u_2^*}{\partial n}\right]^T. \end{aligned} \tag{3.28}$$

Then, boundary conditions (3.20c), (3.20d) on $\mathcal{C}$ of the perturbed system (3.20) can be rewritten in the matrix form:

$$\mathbf{M}\,\mathbf{V}' = \begin{bmatrix} \frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n} \\ \gamma\left[\frac{\partial u_1}{\partial n} + \kappa\,(u_1 - u_0) - \tilde{A}\right] \end{bmatrix}\zeta, \tag{3.29}$$

where

$$\mathbf{M} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ -\gamma & -k & 0 & k \end{bmatrix} \tag{3.30}$$

and $\tilde{A}$ is the integral operator representing the derivative of $u_0(s)$ with respect to the arc length s given by:

$$u_0'(s;\zeta) = \tilde{A}\zeta = \frac{u_{out} - u_{in}}{L} \int_0^L \left[H(s - \tilde{s}) - \frac{s}{L} \right] \kappa\, \zeta\, d\tilde{s}, \tag{3.31}$$

where $H(s)$ is the Heaviside function. The complete details of the derivation of (3.31) are provided in Appendix A.

We can think of $\mathbf{V'}$ as an unknown solution for underdetermined system (3.29) and solve it for $\mathbf{V'}$. Note that we have 2 degrees of freedom, so we introduce 2 free variables: v_1 and v_2. By ansatz from [14] we have

$$\mathbf{V'} = \zeta \mathbf{p_0} + v_1 \mathbf{p_1} + v_2 \mathbf{p_2},$$

where $\mathbf{p_0} \in \text{Range}(\mathbf{M}^T)$ is a particular solution of the system $\mathbf{M}\,\mathbf{V'} = \mathbf{W}\zeta$, where $\mathbf{W}$ is the matrix on the right hand side (RHS) of (3.29) and $\text{span}\{\mathbf{p_1}, \mathbf{p_2}\} = \text{Ker}(\mathbf{M})$.

Thus

$$\begin{aligned} \mathbf{V'} &= \zeta\, \mathbf{p_0} + v_1 \mathbf{p_1} + v_2 \mathbf{p_2} \\ &= \zeta \begin{bmatrix} \frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n} \\ -\frac{\gamma}{k}\left[\frac{\partial u_2}{\partial n} + \kappa\,(u_1 - u_0) - \tilde{A} \right] \\ 0 \\ 0 \end{bmatrix} + v_1 \begin{bmatrix} 1 \\ -\frac{\gamma}{k} \\ 1 \\ 0 \end{bmatrix} + v_2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{3.32}$$

Now, let us denote the first integrand of B as b_1 and write it in a matrix form using

a suitably defined matrix $\mathbf{A}$:

$$\begin{aligned} b_1 &= \frac{\partial u_2'}{\partial n} u_2^* - \frac{\partial u_1'}{\partial n} u_1^* + \frac{\partial u_1^*}{\partial n} u_1' - \frac{\partial u_2^*}{\partial n} u_2' \\ &= \mathbf{V}^{*T} \mathbf{A} \mathbf{V}', \end{aligned} \tag{3.33}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}. \tag{3.34}$$

Using the expression for $\mathbf{V}'$ from (3.32) we get:

$$\begin{aligned} b_1 &= \mathbf{V}^{*T} \mathbf{A} [\zeta\, \mathbf{p_0} + v_1 \mathbf{p_1} + v_2 \mathbf{p_2}] \\ &= (\mathbf{V}^{*T} \mathbf{A} \mathbf{p_0}) \zeta + (\mathbf{V}^{*T} \mathbf{A} \mathbf{p_1}) v_1 + (\mathbf{V}^{*T} \mathbf{A} \mathbf{p_2}) v_2. \end{aligned} \tag{3.35}$$

Notice that, if we set $\mathbf{V}^{*T} \mathbf{A} \mathbf{p_1} = 0$ and $\mathbf{V}^{*T} \mathbf{A} \mathbf{p_2} = 0$ due to the definition of $\mathbf{p_1}$ and $\mathbf{p_2}$, then our b_1 is a linear form of ζ and so B will be in the Riesz form. Enforcing these conditions we get:

$$\begin{aligned} \mathbf{V}^{*T} \mathbf{A} \mathbf{p_1} &= \begin{bmatrix} u_1^*, \frac{\partial u_1^*}{\partial n}, u_2^*, \frac{\partial u_2^*}{\partial n} \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -\frac{\gamma}{k} \\ 1 \\ 0 \end{bmatrix} \\ &= \frac{\partial u_1^*}{\partial n} + \frac{\gamma\, u_1^*}{k} - \frac{\partial u_2^*}{\partial n} = 0, \end{aligned}$$

$$\mathbf{V}^{*T}\mathbf{A}\mathbf{p_2} = \left[u_1^*, \frac{\partial u_1^*}{\partial n}, u_2^*, \frac{\partial u_2^*}{\partial n}\right] \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$= -u_1^* + u_2^* = 0.$$

Therefore, we determined the following boundary conditions of the adjoint system:

$$k\left(\frac{\partial u_2^*}{\partial n} - \frac{\partial u_1^*}{\partial n}\right) = \gamma\, u_1^*, \tag{3.36}$$

$$u_2^* - u_1^* = 0. \tag{3.37}$$

Thus, the first integral in B is in the desired form

$$\begin{aligned}
&- k\int_{\mathcal{C}} (\mathbf{V}^{*T}\mathbf{A}\mathbf{p_0})\zeta\, ds \\
&= -k\int_{\mathcal{C}} \left\{ \left(\frac{\partial u_2}{\partial n} - \frac{\partial u_1}{\partial n}\right)\frac{\partial u_1^*}{\partial n} + \frac{\gamma\, u_1^*}{k}\left[\frac{\partial u_2}{\partial n} + \kappa\,(u_1 - u_0) - \tilde{A}\right]\right\}\zeta\, ds \\
&= [\text{ by } (2.1d)] = \int_{\mathcal{C}} \left\{ -\gamma(u_1 - u_0)\frac{\partial u_1^*}{\partial n} - \gamma\, u_1^*\left[\frac{\partial u_2}{\partial n} + \kappa\,(u_1 - u_0) - \tilde{A}\right]\right\}\zeta\, ds \\
&= \int_{\mathcal{C}} \left\{ -\gamma\,(u_1 - u_0)\left(\kappa\, u_1^* + \frac{\partial u_1^*}{\partial n}\right) - \gamma\, u_1^*\frac{\partial u_2}{\partial n}\right\}\zeta\, ds - \int_{\mathcal{C}} \gamma u_1^*\tilde{A}\zeta\, ds \\
&= \int_{\mathcal{C}} \left\{ -\gamma\,(u_1 - u_0)\left(\kappa\, u_1^* + \frac{\partial u_1^*}{\partial n}\right) - \gamma\, u_1^*\frac{\partial u_2}{\partial n}\right\}\zeta\, ds \\
&- \frac{\gamma(u_{out} - u_{in})}{L}\int_0^L \left\{ \int_0^L \left[H(\tilde{s} - s) - \frac{\tilde{s}}{L}\right] u_1^*(\tilde{s})\, d\tilde{s}\right\}\kappa\,\zeta\, ds,
\end{aligned} \tag{3.38}$$

where the integral operator $\tilde{A}$ was expanded (cf. (3.31)).

Also note that by (3.36) the second integral in B vanishes. It remains to consider the third integral. Since it is already in the Riesz form, let us simplify it by expressing

its integrand b_2 in the local curvilinear form:

$$\begin{aligned} b_2 &= \left[\frac{\partial}{\partial n}\Big(\frac{\partial u_2}{\partial n}u_2^* - \frac{\partial u_1}{\partial n}u_1^*\Big) + \kappa\Big(\frac{\partial u_2}{\partial n}u_2^* - \frac{\partial u_1}{\partial n}u_1^*\Big) + \Big(\nabla u_1 \cdot \nabla u_1^* - \nabla u_2 \cdot \nabla u_2^*\Big)\right] \\ &= \frac{\partial^2 u_2}{\partial n^2}u_2^* + \frac{\partial u_2}{\partial n}\frac{\partial u_2^*}{\partial n} - \frac{\partial^2 u_1}{\partial n^2}u_1^* - \frac{\partial u_1}{\partial n}\frac{\partial u_1^*}{\partial n} + \kappa\,\frac{\partial u_1}{\partial n}u_2^* - \kappa\,\frac{\partial u_1}{\partial n}u_1^* \\ &+ \left(\frac{\partial u_1}{\partial s}, \frac{\partial u_1}{\partial n}\right) \cdot \left(\frac{\partial u_1^*}{\partial s}, \frac{\partial u_1^*}{\partial n}\right) - \left(\frac{\partial u_2}{\partial s}, \frac{\partial u_2}{\partial n}\right) \cdot \left(\frac{\partial u_2^*}{\partial s}, \frac{\partial u_2^*}{\partial n}\right) \\ &= \frac{\partial^2 u_2}{\partial n^2}u_2^* - \frac{\partial^2 u_1}{\partial n^2}u_1^* + \kappa\,\frac{\partial u_2}{\partial n}u_2^* - \kappa\,\frac{\partial u_1}{\partial n}u_1^* + \frac{\partial u_1}{\partial s}\frac{\partial u_1^*}{\partial s} - \frac{\partial u_2}{\partial s}\frac{\partial u_2^*}{\partial s}. \end{aligned}$$

To deal with $\frac{\partial^2}{\partial n^2}$ terms we use Laplace-Beltrami operator which is defined as:

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

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

$$\frac{\partial^2}{\partial n^2} = \Delta\, - \frac{\partial^2}{\partial s^2} - \kappa\,\frac{\partial}{\partial n}. \tag{3.39}$$

This gives us

$$b_2 = \Big(\Delta u_2 - \frac{\partial^2 u_2}{\partial s^2}\Big)u_2^* - \Big(\Delta u_1 - \frac{\partial^2 u_1}{\partial s^2}\Big)u_1^* + \frac{\partial u_1}{\partial s}\frac{\partial u_1^*}{\partial s} - \frac{\partial u_2}{\partial s}\frac{\partial u_2^*}{\partial s}.$$

Next, we notice that since $u_1 = u_2$ and $u_1^* = u_2^*$ on $\mathcal{C}$ we get $\frac{\partial^i u_1}{\partial s^i} = \frac{\partial^i u_2}{\partial s^i}$, $i = 1, 2$ and $\frac{\partial u_1^*}{\partial s} = \frac{\partial u_2^*}{\partial s}$. Together with $\Delta u_1 = \Delta u_2$ on $\mathcal{C}$, which follows from smooth extension of the Laplacian of u_1 and u_2 to $\mathcal{C}$, we get that b_2 vanishes and so the third integral in B is zero.

Finally, we have

$$\begin{aligned} B = \int_{\mathcal{C}} \Big\{ -\gamma\,(u_1 - u_0)\Big(\kappa u_1^* + \frac{\partial u_1^*}{\partial n}\Big) - \gamma\,\frac{\partial u_2}{\partial n}\,u_1^* \Big\}\,\zeta\,ds \\ -\frac{\gamma(u_{out}-u_{in})}{L}\int_0^L \Big\{ \int_0^L \Big[H(\tilde{s}-s) - \frac{\tilde{s}}{L}\Big]\,u_1^*(\tilde{s})\,d\tilde{s}\Big\}\kappa\,\zeta\,ds. \end{aligned} \tag{3.40}$$

Since $B = \mathcal{J}'(\mathcal{C};\zeta\mathbf{n}) = \langle \boldsymbol{\nabla}^{L^2}\mathcal{J},\,\zeta\rangle$, our L^2 gradient is:

$$\begin{aligned} \boldsymbol{\nabla}^{L^2}\mathcal{J} = &-\gamma\,(u_1 - u_0)\Big(\kappa u_1^* + \frac{\partial u_1^*}{\partial n}\Big) - \gamma\,\frac{\partial u_2}{\partial n}\,u_1^* \\ &-\frac{\gamma\kappa(u_{out}-u_{in})}{L}\int_0^L \Big[H(\tilde{s}-s) - \frac{\tilde{s}}{L}\Big]\,u_1^*(\tilde{s})\,d\tilde{s} \quad \text{ on } \mathcal{C}. \end{aligned} \tag{3.41}$$

In the presence of length constraint the gradient is given by:

$$\begin{aligned} \boldsymbol{\nabla}^{L^2}_{L_0}\mathcal{J} = &-\gamma\,(u_1 - u_0)\Big(\kappa u_1^* + \frac{\partial u_1^*}{\partial n}\Big) - \gamma\,\frac{\partial u_2}{\partial n}\,u_1^* \\ &-\frac{\gamma\kappa(u_{out}-u_{in})}{L}\int_0^L \Big[H(\tilde{s}-s) - \frac{\tilde{s}}{L}\Big]\,u_1^*(\tilde{s})\,d\tilde{s} + \alpha\,\Big(\int_{\mathcal{C}} ds - L_0\Big)\kappa \quad \text{ on } \mathcal{C}. \end{aligned} \tag{3.42}$$

Ultimately, our adjoint system is:

$$-k\Delta u_1^* = (u-\bar{u})\chi_{A_1} \quad \text{in } \Omega_1, \tag{3.43a}$$

$$-k\Delta u_2^* = (u-\bar{u})\chi_{A_2} \quad \text{in } \Omega_2, \tag{3.43b}$$

$$u_2^* - u_1^* = 0 \quad \text{on } \mathcal{C}, \tag{3.43c}$$

$$k\Big(\frac{\partial u_2^*}{\partial n} - \frac{\partial u_1^*}{\partial n}\Big) = \gamma\,u_1^* \quad \text{on } \mathcal{C}, \tag{3.43d}$$

$$k\frac{\partial u^*}{\partial n} = 0 \quad \text{on } \partial\Omega, \tag{3.43e}$$

$$\text{where } u = \begin{cases} u_1, & \mathbf{x} \in \Omega_1 \\ u_2, & \mathbf{x} \in \Omega_2 \end{cases} \quad \text{and } u^* = \begin{cases} u_1^*, & \mathbf{x} \in \Omega_1 \\ u_2^*, & \mathbf{x} \in \Omega_2 \end{cases}.$$

We emphasize that in order to evaluate cost functional gradient (3.42) direct and adjoint problems (2.1) and (3.43), respectively, need to be solved. The appropriate numerical scheme is discussed in the next chapter.

3.3.3 Ensuring smoothness of the gradient

The perturbation of $\mathcal{C}$ in the direction of the gradient $\boldsymbol{\nabla}^{L^2}\mathcal{J}$ might not lead to a smooth contour, since the L^2 space only guarantees square integrability. To ensure the smoothness of the perturbed contour $\mathcal{C}$, we can use an operator from some Sobolev space, in particular the Hilbert space H^1, to extract a smoother gradient. According to the Riesz representation theorem, there exists the corresponding gradient $\boldsymbol{\nabla}^{H^1}\mathcal{J}$, which satisfies the following identity [15]:

$$\mathcal{J}'(\mathcal{C};\zeta\mathbf{n}) = \langle \boldsymbol{\nabla}^{H^1}\mathcal{J}, \zeta\rangle_{H^1} = \int_{\mathcal{C}} \Big(\boldsymbol{\nabla}^{H^1}\mathcal{J} + l^2 \frac{\partial \boldsymbol{\nabla}^{H^1}\mathcal{J}}{\partial s}\frac{\partial \zeta}{\partial s}\Big)\, ds, \tag{3.44}$$

where l is a constant sometimes referred to as the Sobolev coefficient playing the role of an adjustable length-scale parameter. Applying integration by parts, we get

$$\mathcal{J}'(\mathcal{C};\zeta\mathbf{n}) = \langle \boldsymbol{\nabla}^{H^1}\mathcal{J}, \zeta\rangle_{H^1} = \int_{\mathcal{C}} \Big(\boldsymbol{\nabla}^{H^1}\mathcal{J} - l^2 \frac{\partial^2 \boldsymbol{\nabla}^{H^1}\mathcal{J}}{\partial s^2}\Big)\zeta\, ds + l^2 \frac{\partial \boldsymbol{\nabla}^{H^1}\mathcal{J}}{\partial s}\,\zeta\Big|_B^{B'}.$$

Combining this result with the Riesz representation in L^2, namely,

$$J'(\mathcal{C};\zeta\mathbf{n}) = \langle \boldsymbol{\nabla}^{L^2}\mathcal{J}, \zeta\rangle_{L^2} = \int_{\mathcal{C}} \boldsymbol{\nabla}^{L^2}\mathcal{J}\,\zeta\, ds$$

we can derive the following system characterizing the Sobolev gradient $\boldsymbol{\nabla}^{H^1}\mathcal{J}$:

$$\begin{cases} (1 - l^2\partial_s^2)\boldsymbol{\nabla}^{H^1}\mathcal{J} & = \boldsymbol{\nabla}^{L^2}\mathcal{J} \quad \text{on } \mathcal{C} \\ l^2 \frac{\partial \boldsymbol{\nabla}^{H^1}\mathcal{J}}{\partial s} & = 0 \qquad\quad \text{at } B, B'. \end{cases} \tag{3.45}$$

Similar result holds for the gradient corresponding to the case with the length constraint. Note that $l \to 0$ recovers the standard L^2 gradient, whereas $l \to \infty$ recovers the H^1 gradient based on the inner product corresponding to the seminorm (3.44), i.e. larger l values result in the higher smoothness of the Sobolev gradient.

More specifically, incorporating derivatives into the inner product, has the effect of scale-depending filtering [15]. The procedure introduced above is equivalent to applying a low-pass filter to the L^2 gradient with the quantity l^{-2} representing the "cut-off" scale.

The extraction of gradients in spaces of smoother functions, such as Sobolev space H^1, is a well-known technique in adjoint-based optimization of PDEs [15, 16], where it is treated as a form of preconditioning.

Chapter 4

Numerical Implementation

To obtain an optimal contour $\mathcal{C}$, corresponding to a local minimum of our cost functional, we need to be able to evaluate the gradient (3.42) at each iteration of the algorithm. Since the expression for $\nabla\mathcal{J}$ involves the solutions of both direct and adjoint problems (2.1) and (3.43), the corresponding systems of PDEs have to be solved for any new configuration of the contour $\mathcal{C}$. Since these systems have in fact essentially identical structure, we will focus our discussion on the solution of the first one. We observe that both problems (2.1) and (3.43) can be treated as combinations of two Poisson problems (defined in Ω_1 and in Ω_2) which are coupled via some mixed boundary conditions on the contour $\mathcal{C}$ separating the two domains. Given the linearity of equations (2.1a)—(2.1b), we split problem (2.1) into two subproblems: a potential problem associated with the complex boundary conditions (2.1d) and another elliptic problem arising from the presence of the source term q, which are then coupled using a suitable interpolation scheme. Since the solution methods for these subproblems are adapted to their analytic structure, we achieve for each of them the highest possible (spectral) accuracy.

We begin this chapter with a brief introduction to spectral methods and the bound-

ary integral equation technique which provide theoretical preliminaries for dealing with our subproblems, followed by the computational scheme developed for their numerical solution. The major steps are summarized in the optimization algorithm provided at the end of this chapter.

4.1 Spectral Methods for Non-periodic Domains

In this section we will see how to construct spectral methods for bounded, non-periodic domains. Suppose we work with non-periodic functions defined on $[-1, 1]$. One way is to extend the functions periodically and treat them as periodic applying trigonometric (Fourier) interpolation in equispaced points. The disadvantage of such approach is the loss of spectral accuracy, since in general a smooth function becomes non-smooth when periodically extended. Instead, a polynomial interpolation on unevenly spaced points is the right choice [17]. The common property of appropriate sets is the density of grid points distribution:

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

The most common set and the one we will use is *Chebyshev points*:

$$x_j = \cos(j\pi/N), \quad j = 0, ..., N.$$

Geometrically, these points can be seen as the projections on $[-1, 1]$ of equispaced points $j\pi/N$, $j = 0, ..., N$ on the upper half of the unit circle (see Figure 4.1). Note that the points are numbered from right to left.

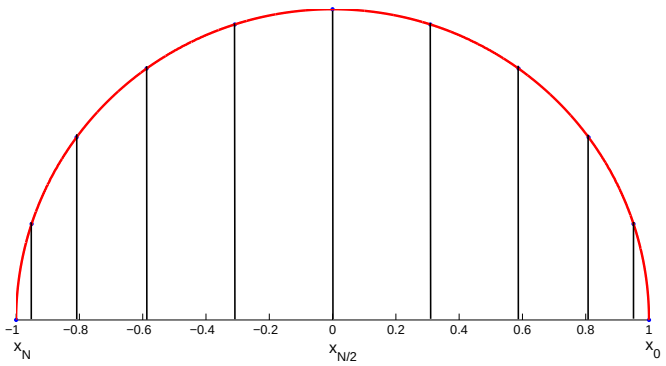


Figure 4.1: Chebyshev points

4.1.1 Spectral Differentiation for Non-periodic Domain

Now, we will use Chebyshev points to construct Chebyshev differentiation matrices and demonstrate how to apply them to numerically solve a certain class of PDEs. Let us first discuss how we can approximate a discrete derivative $\mathbf{w}$ of a function v having only the values v_j at the Chebyshev points $x_j = \cos(j\pi/N)$, $j = 0, ..., N$. Following [17], this can be done in two steps:

- Let p be the unique polynomial of degree $\leq N$ with $p(x_j) = v_j$, $j = 0, ..., N$.
- Set $w_j = p'(x_j)$.

Since this operation is linear, it can be represented as multiplication of a vector $\mathbf{v} := [v_0, ..., v_N]$ by a $(N+1) \times (N+1)$ matrix denoted as $\mathbf{D}_N$:

$$\mathbf{w} = \mathbf{D}_N \mathbf{v}.$$

If we construct the polynomial p from step 1 given above, say, by writing it in the Lagrange form, and taking the derivative as in step 2, we can derive general formulas for the elements of $\mathbf{D}_N$ for an arbitrary N:

Lemma 2. [20] *For each $N \geq 1$, let the rows and columns of the $(N+1) \times (N+1)$ Chebyshev spectral matrix $\mathbf{D}_N$ be indexed from 0 to N. The entries of this matrix are*

$$\begin{aligned}
(\mathbf{D}_N)_{00} &= \frac{2N^2+1}{6}, \quad (\mathbf{D}_N)_{NN} = -\frac{2N^2+1}{6}, \\
(\mathbf{D}_N)_{ii} &= \frac{-x_i}{2(1-x_i^2)} \quad 1 \leqslant i \leqslant N-1, \\
(\mathbf{D}_N)_{ij} &= \frac{c_i(-1)^{i+j}}{c_j(x_i - x_j)} \quad i \neq j; 1 \leqslant i,j \leqslant N-1,
\end{aligned} \tag{4.1}$$

where

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

Note that the second derivative of v can be computed by applying $\mathbf{D}_N^2$ matrix which is the square of $\mathbf{D}_N$. Let us now consider an example of solving a simple ordinary differential equation (ODE) with Neumann and Dirichlet conditions at the left and right endpoints respectively.

Example 1:

$$u_{xx} = e^{4x}, \quad -1 < x < 1, \quad u_x(-1) = u(1) = 0. \tag{4.2}$$

We already know how to treat derivatives numerically with spectral accuracy with the help of the spectral differentiation matrix. Denote $\mathbf{v} = [u_0, ..., u_N]$ as the vector of unknowns corresponding to Chebyshev points $x_0, ..., x_N$ and $\mathbf{f} = [e^{4x_0}, ..., e^{4x_N}]$ as a vector corresponding to the right hand side of the equation evaluated on this grid. Then, the second derivative operator on the LHS evaluated at these points can be approximated by $\mathbf{D}_N^2\mathbf{v}$.

So now the question is how we can impose the boundary conditions. The condition at $x = 1$, which corresponds to the grid point $j = 0$, will be satisfied if we replace

the first row of the differentiation matrix $\mathbf{D}_N^2$ with the row $[1, 0, 0, ..., 0]$ and the first element of the vector $\mathbf{f}$ with 0. At $x = -1$, i.e. $j = N$, on the other hand we need to impose a condition involving the first derivative. To approximate the first derivative of u at this point we will replace the last row of $\mathbf{D}_N^2$ with the last row of the spectral differentiation matrix $\mathbf{D}_N$. The corresponding element of the vector $\mathbf{f}$ is assigned 0. So, in the end we will be solving $(N+1) \times (N+1)$ linear system of equations in which $N - 1$ equations enforce the governing equation at the interior grid points and the first and last equations provide the Dirichlet and the Neumann boundary conditions respectively. The RHS will be represented by the vector $[0, e^{4x_1}, ..., e^{4x_{N-1}}, 0]$.

After giving the idea of the approach for one-dimensional problem, we now consider how to extend the method to problems in several space dimensions. Due to the nature of our mathematical model, we restrict ourselves to 2D case.

To be more specific, let us consider the discretization of 2D Laplace operator:

$$\Delta u := u_{xx} + u_{yy}, \quad x, y \in [-1, 1].$$

The proper grid is based on Chebyshev points independently constructed in each direction and known as a *tensor product grid* (see Figure 4.2).

To express the operators, in particular the Laplacian, on such grid we use tensor (Kronecker) products, denoted by $\otimes$: if $\mathbf{B}$ and $\mathbf{C}$ are of dimensions $n \times m$ and $l \times r$, respectively, then $\mathbf{B} \otimes \mathbf{C}$ is $nl \times mr$ matrix with $n \times m$ block form, where the (i, j) block is $b_{ij}\mathbf{C}$.

Let us demonstrate how it works on a small size grid $N = 2$ (see Figure 4.3). Note, that the way we enumerate the nodes is crucial for the method - another ordering would, in general, give another equivalent operator representation. Here we begin with the upper-right grid point and move to the lower-left point: start at the level

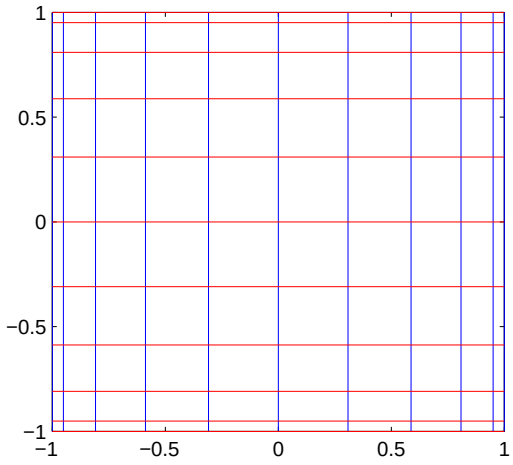


Figure 4.2: 2D Chebyshev grid

$y_0 = 1$ and enumerate points going from $x_0 = 1$ to $x_N = -1$, next, move to the level y_1 and repeat until $y_N = -1$ is reached. This idea is an extension of a numeration approach applied in 1D case and correlates with the spectral differentiation matrix construction.

We emphasize that by applying the enumeration technique described above, we reshape our data on $(N+1) \times (N+1)$ grid to be represented by a $(N+1)^2$ vector. This linear transformation sometimes referred as the vectorization of the matrix.

Suppose we get discrete representation of u at these grid points: $[v_1, ..., v_9]^T$. To approximate the Laplacian we need to differentiate spectrally in the x and y directions independently. With the help of the spectral differentiation matrix $\mathbf{D}_N^2$ we can derive the second derivative in the x direction as [17]:

$$u_{xx} \approx I \otimes \mathbf{D}_N^2 \mathbf{v},$$

where I is the $(N+1) \times (N+1)$ identity matrix.

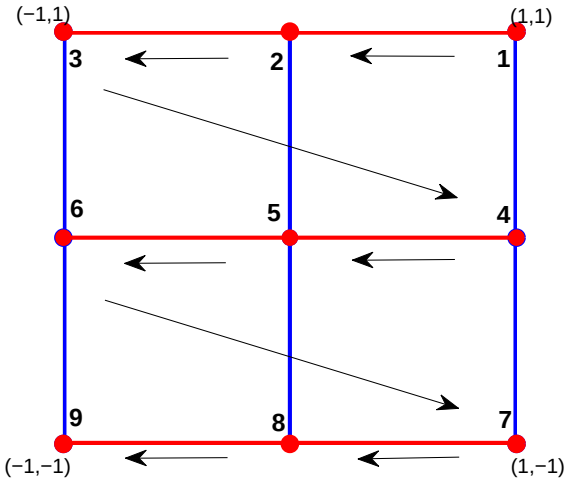


Figure 4.3: Tensor product grid 3x3 and enumeration pattern

Let us demonstrate this fact by expanding the Kronecker product for $N = 2$:

$$u_{xx} \approx \left[\begin{array}{c|c|c} \mathbf{D}_2^2 & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{D}_2^2 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{D}_2^2 \end{array}\right] \left[\begin{array}{c} v_1 \\ \vdots \\ v_9 \end{array}\right],$$

where $\mathbf{D}_2^2$ is a 3×3 matrix and $\mathbf{0}$ stands for a 3×3 zero block. On the other hand, the second derivative in the y direction is approximated by the following tensor product:

$$u_{yy} \approx \mathbf{D}_N^2 \otimes I\, \mathbf{v},$$

so that

$$u_{yy} \approx \left[\begin{array}{c|c|c} (\mathbf{D}_2^2)_{11} I & (\mathbf{D}_2^2)_{12} I & (\mathbf{D}_2^2)_{13} I \\ \hline (\mathbf{D}_2^2)_{21} I & (\mathbf{D}_2^2)_{22} I & (\mathbf{D}_2^2)_{23} I \\ \hline (\mathbf{D}_2^2)_{31} I & (\mathbf{D}_2^2)_{32} I & (\mathbf{D}_2^2)_{33} I \end{array}\right] \left[\begin{array}{c} v_1 \\ \vdots \\ v_9 \end{array}\right].$$

Thus, our discrete Laplacian is given by:

$$\Delta \approx \mathbf{\Delta}_N := I \otimes \mathbf{D}_N^2 + \mathbf{D}_N^2 \otimes I. \tag{4.3}$$

If we need to include some boundary conditions to such type of the problems, we should follow a similar idea as was developed for 1D case (see Example 1), i.e. replace the corresponding rows or columns in the final matrix, $\mathbf{\Delta}_N$ in our case, with the data from the matrix representing boundary conditions. For example, if we want to enforce $u_x = 0$ at $x = 1$ as a part of some boundary condition, we start with identifying the indices of the corresponding elements in the vector $\mathbf{v}$. From Figure 4.3 we get that v_1, v_4 and v_7 are given at the point $x = 1$. Thus, in the discrete Laplacian columns 1, 4 and 7 need to be replaced with the corresponding columns of the matrix $I \otimes \mathbf{D}_N$ which is an approximation of u_x on 2D Chebyshev grid.

4.1.2 Spectral Collocation Methods

The basic idea for solving differential and integral equations expressed in a general form $\mathcal{L}u = f$, where $\mathcal{L}$ is differential or integral operator, is to assume that unknown function $u(x)$ can be approximated by a sum of $N + 1$ basis functions $\varphi_n(x)$:

$$u(x) \approx u_N(x) = \sum_{n=0}^{N} a_n \varphi_n(x). \tag{4.4}$$

After substituting this series back into the equation the so-called residual function is defined:

$$R(x; a_0, ..., a_N) = \mathcal{L}u_N - f.$$

The residual function is identically equal to zero for the exact solution. To obtain an approximation to the solution, i.e. the coefficients $a_0, ..., a_N$ in (4.4), two main approaches exist:

- spectral or Galerkin

  The coefficients are computed by multiplying the residual $R(x, a_n)$ by given basis functions and integrating over the domain of definition (coefficients $a_0, ..., a_N$ are chosen such that the residual is orthogonal to the subset spanned by the basis functions)

- pseudospectral or collocation

  The coefficients are found by requiring that the residual vanish at the collocation points.

Since the pseudospectral method is equivalent to the spectral if the integrals of the latter is computed by Gaussian numerical quadrature [18], in this work we are using collocation method for the sake of computational efficiency.

Due to non-periodicity of our domain, the best choice for the basis set is Chebyshev polynomials of the first kind:

$$T_i(x) = \cos(i \arccos x), \quad i = 0, ..., N.$$

The optimal choice of the collocation points associated with the Chebyshev basis functions depends on the type of the problem we are solving, but the most common sets are the following:

- Gauss-Chebyshev points:

$$x_i = \cos\left(\frac{(2i+1)\pi}{2(N+1)}\right), \quad i = 0, ..., N.$$

  Note that this set of points exclude the end points of the interval $[-1, 1]$.

- Chebyshev-Lobatto points:

$$x_i = \cos\left(\frac{\pi i}{N}\right), \quad i = 0, ..., N.$$

  Now the end points are included, which allows us to enforce some boundary conditions, if applicable.

The most appropriate set of points for solving direct and adjoint problems (2.1), (3.43) is discussed in Section 4.4 devoted to numerical techniques.

4.2 Clenshaw-Curtis Quadrature

Let us discuss how the Chebyshev polynomials and points can be utilized for numerical evaluation of certain integrals. Consider an integral

$$I = \int_{-1}^{1} f(x)\, dx. \tag{4.5}$$

To obtain a spectrally accurate approximation of such type of the integral the Clenshaw-Curtis quadrature can be used:

$$I \approx \sum_{j=0}^{N} w_j\, f(x_j), \tag{4.6}$$

where $x_j = \cos(\frac{\pi j}{N})$, $j = 0, ..., N$ are Chebyshev-Lobatto grid points and w_j are the weights to be determined. Recall that Chebyshev polynomials are the images of cosine functions under the mapping $x = \cos(t)$. Thus, this grid is an evenly spaced in the trigonometric coordinate t. The integral I after the transformation $x = \cos(t)$ becomes:

$$I = \int_{0}^{\pi} f(\cos(t))\, \sin(t)\, dt$$

and the quadrature approximation is given by [18]:

$$
\begin{aligned}
I_N &= \sum_{j=0}^{N} w_j \, f(\cos(t_j)), \\
t_j &= \pi j/N, \quad j = 0, ..., N, \\
w_j &= \sin(t_j) \, \frac{2}{N} \sum_{m=0}^{N} \sin(m t_j) \, [1 - \cos(m\pi)]/m.
\end{aligned}
$$

Such a quadrature is exact for polynomials of degree $(N-1)$ [18]. One could argue that Gaussian quadrature with the weight function $w(x) \equiv 1$ could be a better choice, since it is exact when $f(x)$ is a polynomial of degree $(2N-1)$ or less. The discussion on this subject can be found in [19] and [18] and, as demonstrated in [19], the two formulas are about equally accurate for most integrands. The main benefit from the Clenshaw-Curtis quadrature for the algorithm employed in this work is the fact that the Chebyshev grid, where the integrand is evaluated, serves also as a main tool for discretizing the data in the direct and adjoint problems. On the other hand, Gaussian type quadratures with similar abscissas, such as Chebyshev-Gauss quadrature with the weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$, would result in the need for a discontinuous change of variables to adjust the integrand offsetting the advantage of this approach (see the concrete example in Section 4.4.2).

4.3 Boundary Integral Equation

The *boundary integral equation method* denotes the transformation of a linear partial differential equations with d spatial variables into an integral equation over a $(d-1)$-dimensional surface. It is typically applicable to elliptic PDE problems. The method starts from a differential equation $Lu = 0$ with suitable boundary conditions and looks for an equivalent formulation as integral equation. Here, we restrict ourselves

to the two-dimensional Laplace equation and demonstrate how to transform it into an integral equation.

4.3.1 Single-layer Potential

We observe that the solution of an inhomogeneous linear differential equation, in particular the governing Poisson equation (2.1a) in our problem, can be represented as a sum

$$u = u_p + u_h \tag{4.7}$$

consisting of the solution u_p of some inhomogeneous problem and some harmonic function u_h, satisfying Laplace equation and complicated boundary conditions (2.1d)-(2.1e). The solutions u_p and u_h can be thought as solutions of two different subproblems, one of which is governed by Poisson equation and another is governed by Laplace equation.

In this section we discuss remarkable properties of the solution of Laplace equation which will be utilized to introduce the boundary integral formulation of one of the subproblems satisfying boundary conditions (2.1d)-(2.1e).

Let us consider the case when $d = 2$, $\Gamma = \partial\Omega$ is a closed curve. For Laplace's equation $-\Delta\,\Phi = 0$ in $\mathbb{R}^2$ we have the following fundamental solution [20]

$$\omega(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi} \ln |\mathbf{x} - \mathbf{y}|, \tag{4.8}$$

which only depends on the Euclidean norm $|\mathbf{x} - \mathbf{y}|$ and the order of singularity at $\mathbf{x} = \mathbf{y}$ is the weakest possible.

The curvilinear integral

$$\Phi(\mathbf{x}) = \oint_\Gamma \omega(\mathbf{x}, \mathbf{y}) \mu(\mathbf{y})\, d\mathbf{y} \tag{4.9}$$

over Γ with fundamental solution ω defines the *single-layer potential* corresponding to differential operator L. The function μ is called the *density* of the single-layer potential. In particular, the single layer potential corresponding to Laplace's operator, according to (4.8), has the following form in $\mathbb{R}^2$

$$\Phi(\mathbf{x}) = -\frac{1}{2\pi} \oint_{\Gamma} \ln |\mathbf{x} - \mathbf{y}| \mu(\mathbf{y}) \, d\mathbf{y}. \tag{4.10}$$

The next lemma gives the fundamental result on integral representation of Laplace equation.

Lemma 3. [20] *Let Γ is a piecewise C^1 curve. The single-layer potential* (4.10) *satisfies the Laplace equation in $\mathbb{R}^2 \setminus \Gamma$, i.e.,*

$$-\Delta \, \Phi = 0 \qquad \mathbf{x} \notin \Gamma. \tag{4.11}$$

Next, we will demonstrate some properties of single-layer potential Φ which will be used in numerical implementation.

Lemma 4. *The single-layer potential Φ defined as* (4.9) *over Γ is continuous for $\mathbf{x} \in \mathbb{R}^2$ and analytic in $\mathbb{R}^2 \backslash \Gamma$.*

The proof of this result can be found in [20].

The next lemma employs the fact that for $\mathbf{x} \notin \Gamma$, the derivative of Φ may be taken under the integral sign.

Lemma 5. *The gradient of Φ is:*

$$\nabla \, \Phi = -\frac{1}{2\pi} \oint_{\Gamma} \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^2} \, \mu(\mathbf{y}) \, d\mathbf{y}. \tag{4.12}$$

Note that the kernel $\frac{\mathbf{x}-\mathbf{y}}{|\mathbf{x}-\mathbf{y}|^2}$ is not improperly integrable over Γ.

In addition, we present some results for $\frac{\partial \Phi_{\pm}}{\partial n}$ evaluated on the boundary:

Theorem 3. [20] *Let Γ be Hölder-continuously differentiable and the density function μ of single-layer potential Φ satisfies $f \in L^{\infty}(\Gamma)$. For any $\mathbf{x} \in \Gamma$, 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 $\mathbf{x}$ and is given by:*

$$\frac{\partial \Phi_{\pm}}{\partial n}(\mathbf{x}) := \lim_{\varepsilon \to 0^+} \mathbf{n}(\mathbf{x}) \cdot \nabla \Phi \left(\mathbf{x} \pm \varepsilon \, \mathbf{n}(\mathbf{x})\right), \tag{4.13}$$

2. *the jump relation holds as:*

$$\frac{\partial \Phi_+(\mathbf{x})}{\partial n} - \frac{\partial \Phi_-(\mathbf{x})}{\partial n} = -\mu(\mathbf{x}), \tag{4.14}$$

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

$$\frac{1}{2}\left[\frac{\partial \Phi_+(\mathbf{x})}{\partial n} + \frac{\partial \Phi_-(\mathbf{x})}{\partial n}\right] = -\frac{1}{2\pi} \int_{\Gamma} \frac{< \mathbf{n}(\mathbf{x}), \mathbf{x}-\mathbf{y} >}{|\mathbf{x}-\mathbf{y}|^2} \, \mu(\mathbf{y}) \, d\mathbf{y}, \tag{4.15}$$

where $< \cdot, \cdot >$ denotes the Euclidean inner product.

Equation (4.15) can also been written in the form

$$\frac{1}{2}\left[\frac{\partial \Phi_+(\mathbf{x})}{\partial n} + \frac{\partial \Phi_-(\mathbf{x})}{\partial n}\right] = -\frac{1}{2\pi} \int_{\Gamma} \frac{\partial}{\partial n_x} \ln |\mathbf{x}-\mathbf{y}| \, \mu(\mathbf{y}) \, d\mathbf{y}. \tag{4.16}$$

Moreover, the right hand-side of (4.15) has an integrand with a removable singularity, namely [20]:

$$\lim_{\mathbf{y} \to \mathbf{x}} \frac{< \mathbf{n}(\mathbf{x}), \mathbf{x}-\mathbf{y} >}{|\mathbf{x}-\mathbf{y}|^2} = \frac{\kappa}{2} \quad \mathbf{x}, \mathbf{y} \in \Gamma, \tag{4.17}$$

where κ is the signed curvature of Γ.

Theorem 3 together with (4.17) provides us with specific techniques necessary to apply the BIE method to the direct and adjoint problems.

4.4 Numerical Implementation

In this section we reformulate our direct problem (2.1) in terms of the solutions u_p and u_h using relation (4.7) and employ the properties of the single-layer potential, introduced in the previous section, to satisfy mixed boundary condition (2.1d).

Now the direct problem (2.1) is given by the following system:

$$
\begin{aligned}
-k\Delta u_p &= q(\mathbf{x}) && \text{in } \Omega, && (4.18a)\\
\Delta u_h &= 0 && \text{in } \Omega \setminus \mathcal{C}, && (4.18b)\\
u_h\Big|_1 &= u_h\Big|_2 && \text{on } \mathcal{C}, && (4.18c)\\
k\left(\frac{\partial u_h}{\partial n}\Big|_1 - \frac{\partial u_h}{\partial n}\Big|_2\right) &= \gamma(u-u_0) && \text{on } \mathcal{C}, && (4.18d)\\
\frac{\partial u_p}{\partial n} &= -\frac{\partial u_h}{\partial n} && \text{on } \partial\Omega, && (4.18e)
\end{aligned}
$$

where $u_h\Big|_1 = u_h\chi_{\Omega_1}$, $u_h\Big|_2 = u_h\chi_{\Omega_2}$ and

$$
u_h(\mathbf{x}) = -\frac{1}{2\pi}\oint_{\mathcal{C}} \ln|\mathbf{x}-\mathbf{x}_{\mathcal{C}}|\mu(\mathbf{x}_{\mathcal{C}})d\sigma \quad \text{on } \Omega\setminus\mathcal{C} \tag{4.19}
$$

in which μ is the density function.

Considering the jump relation (4.14) and the boundary condition (4.18d), the density $\mu(\mathbf{x})$ satisfies:

$$
-\mu(\mathbf{x}) = \frac{\gamma}{k}(u-u_0(\mathbf{x})) \quad \forall \mathbf{x}\in\mathcal{C}. \tag{4.20}
$$

Next we plug (4.7) and the single-layer potential into (4.20) and get a singular boundary integral equation

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

of Fredholm type II, satisfied by density μ.

Since by Lemma 4 u_h is continuous across $\mathcal{C}$, then (4.18c) is satisfied by definition and we can write system (4.18) in an equivalent form with $u_p(\mathbf{x})$ and $\mu(\mathbf{x})$ as unknowns:

$$-k\Delta u_p = q(\mathbf{x}) \qquad \text{in } \Omega, \tag{4.21a}$$

$$-\mu(\mathbf{x}) + \frac{\gamma}{2\pi k} \oint_{\mathcal{C}} \ln|\mathbf{x} - \mathbf{x}_{\mathcal{C}}| \mu(\mathbf{x}_{\mathcal{C}})\, d\sigma = \frac{\gamma}{k}(u_p(\mathbf{x}) - u_0(\mathbf{x})) \quad \forall \mathbf{x} \in \mathcal{C}, \tag{4.21b}$$

$$\frac{\partial u_p}{\partial n} = -\frac{\partial u_h}{\partial n} \qquad \mathbf{x} \in \partial\Omega. \tag{4.21c}$$

In the later sections we will use this system to approximate the solution of direct problem (2.1).

4.4.1 Discretizing Equation (4.21a)

To discretize u_p on Ω we introduce a matrix $\mathbf{U}$ where $\mathbf{U}_{ij} = u_p(x_i, y_j)$, $i, j = 0, ..., N$ is an approximation of u_p on 2D Chebyshev grid. Following the ideas from Section 4.1.1, we rearrange the terms in matrix $\mathbf{U}$ to obtain a vector $\tilde{\mathbf{U}}$ and apply the same technique to the right-hand side q, denoting the resulting column vector $\tilde{\mathbf{Q}}$. Now, using spectral differentiation matrix (4.3), the Laplacian of u in (4.21a) can be numerically approximated as:

$$-k\, \boldsymbol{\Delta}_{\mathbf{N}}\, \tilde{\mathbf{U}} = \tilde{\mathbf{Q}}. \tag{4.22}$$

Note that the method of reordering the elements to get the vectors $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{Q}}$ is crucial and should be the same for both sides of the equation and in further numerical

evaluations.

4.4.2 Discretizing Equation (4.21b)

To proceed with discretization of BIE (4.21b) first we need to convert the line integral in (4.21b) into a definite integral. This can be achieved if we parametrize the contour $\mathcal{C}$ in terms of $t \in [0, \pi]$ by a map $\mathbf{z} : \mathbb{R} \to \mathbb{R}^2$ such that:

$$\mathbf{x} = \mathbf{z}(t) := [z_1(t), z_2(t)], \quad \mathbf{x} \in \mathcal{C}, \quad t \in [0, \pi].$$

There are 2 steps which lead to such parametrization. First, we introduce arc length parameter s and the transformation $\theta : s \in [0, L] \to \mathbf{x} \in \mathcal{C}$ with $||\theta'(s)|| = 1$ which converts a line integral over the curve to a definite integral from 0 to L.

Next, define $s(t) = \frac{Lt}{\pi},\ t \in [0, \pi],\ s'(t) = \frac{L}{\pi}$ which maps $t \in [0, \pi]$ to $s \in [0, L]$. Thus, we have $\mathbf{z} := \theta \circ s$, such that $||\mathbf{z}'(t)|| = ||\theta'(t)|| = ||\theta'(s)||s'(t) = \frac{L}{\pi}$.

Let us denote $\mathbf{x}$ and $\mathbf{x}_\mathcal{C}$ as $\mathbf{z}(t)$ and $\mathbf{z}(t')$, respectively. Then, we have our equation transformed to:

$$-\mu(\mathbf{x}) + \frac{\gamma L}{2\pi^2 k} \int_0^\pi \ln |\mathbf{z}(t) - \mathbf{z}(t')| \mu(t')\, dt' = \frac{\gamma}{k}(u_p(t) - u_0(t)) \ , \quad t \in [0, \pi]. \tag{4.23}$$

We have to address the singularity in this line integral as $t \to t'$. Let us split the integral in the following way:

$$-\mu(\mathbf{x}) + \frac{\gamma L}{2\pi^2 k} \underbrace{\int_0^\pi \ln \left| \frac{\mathbf{z}(t) - \mathbf{z}(t')}{\cos(t) - \cos(t')} \right| \mu(t')\, dt'}_{\text{(I)}} + \frac{\gamma L}{2\pi^2 k} \underbrace{\int_0^\pi \ln |\cos(t) - \cos(t')| \mu(t')\, dt'}_{\text{(II)}}$$
$$= \frac{\gamma}{k}(u_p(t) - u_0(t)), \quad t \in [0, \pi]. \tag{4.24}$$

Now the singularity in the integral (I) can be removed by replacing the kernel with $\ln\left|\frac{||\mathbf{z}'(t')||}{\sin(t')}\right|$ when $t \to t'$.

Note that (II) is the integral containing the principal part of the logarithmic kernel, while (I) contains a regular one. The numerical evaluation of the integral (II) with the logarithmic-kernel is neither accurate nor simple in general, so that we will overcome this problem by analytic evaluation of (II) on suitably chosen discrete points. Following [7], we define an approximating subspace $X_M = \text{span}\{T_m(t) = \cos(mt), m = 0, 1, 2, ..., M,\ 0 \le t \le \pi\}$ and approximate the unknown density function as

$$\mu(t) = \mu_M(t) = \sum_{m=0}^{M} a_m T_m(t), \quad 0 \le t \le \pi,$$

where $T_m(t)$ are Chebyshev polynomials.

To treat integral (II) we use the observation that, for $0 \le t \le \pi$,

$$\int_0^\pi T_m(t') \ln|\cos(t) - \cos(t')|\, dt' = \int_{-1}^{1} \frac{T_m(\cos^{-1}(\xi))}{\sqrt{1-\xi^2}} \ln|\xi - x|\, d\xi = \begin{cases} -\pi \ln(2), & \text{if } m = 0 \\ -\frac{\pi}{m} T_m(t), & \text{if } m \ge 1, \end{cases}$$

where we noted that $\cos(t) = \xi$.

Then, we have (4.23) in the form

$$\begin{aligned} &-\sum_{m=0}^{M} a_m T_m(t) + \frac{\gamma L}{2\pi^2 k} \sum_{m=0}^{M} a_m \int_0^\pi \ln\left|\frac{\mathbf{z}(t) - \mathbf{z}(t')}{\cos(t) - \cos(t')}\right| T_m(t')\, dt' - \\ &\frac{\gamma L}{2\pi^2 k}\left(a_0 \pi \ln(2) + \sum_{m=1}^{M} \frac{a_m \pi}{m} T_m(t)\right) = \frac{\gamma}{k}(u_p(t) - u_0(t)), \quad t \in [0, \pi]. \end{aligned} \tag{4.25}$$

So, now to find $\mu_M(t)$ we need to find the coefficients a_m, $m = 0, 1, .., M$ from

$$-\sum_{m=0}^{M} a_m T_m(t) + \frac{\gamma L}{2\pi^2 k}\sum_{m=0}^{M} a_m I_m(t) - \frac{\gamma L}{2\pi k}\left(a_0 \ln(2) + \sum_{m=1}^{M}\frac{a_m}{m}T_m(t)\right) = \frac{\gamma}{k}(u_p(t) - u_0(t)), \tag{4.26}$$

where

$$I_m(t) = \int_0^{\pi} \ln\left|\frac{\mathbf{z}(t) - \mathbf{z}(t')}{\cos(t) - \cos(t')}\right| T_m(t')\, dt'. \tag{4.27}$$

Let us discuss the numerical treatment of this integral.

Option 1: Note that the change of variables $\tau = \cos(t)$ with $dt = \frac{-1}{\sqrt{1-\tau^2}}\, d\tau$ and $\mathbf{z}(t) = \mathbf{z}(\cos^{-1}(\cos(t))) = \mathbf{z}(\cos^{-1}(\tau)) = \tilde{\mathbf{z}}(\tau)$ transforms (4.27) to

$$I_m(\tau) = \int_{-1}^{1} \ln\left|\frac{\tilde{\mathbf{z}}(\tau) - \tilde{\mathbf{z}}(\tau')}{\tau - \tau'}\right| \frac{T_m(\cos^{-1}(\tau'))}{\sqrt{1-\tau'^2}}\, d\tau'.$$

The weight function $\frac{1}{\sqrt{1-\tau^2}}$ suggests the use of the Chebyshev-Gauss quadrature to approximate the integral. However, investigating the smoothness of the integrand $\ln\left|\frac{\tilde{\mathbf{z}}(\tau)-\tilde{\mathbf{z}}(\tau')}{\tau-\tau'}\right| T_m(\cos^{-1}(\tau'))$ and, in particular, $T_m(\cos^{-1}(\tau')) = \cos(m\cos^{-1}(\tau'))$, whose derivative given by $\frac{m\sin(m\cos^{-1}(\tau))}{\sqrt{1-\tau^2}}$ has a singularity at $\tau = \pm 1$, we conclude that the integrand does not obey the smoothness requirements on the interval $\tau \in [-1, 1]$ necessary to guarantee the spectral convergence of the quadrature.

Option 2: We can also make use of the spectral methods for the integrals by representing the curve $\mathbf{z}(t)$, $t \in [0, \pi]$ in some new variable $y \in [-1, 1]$ via a linear change of variables. Let $t = \frac{y+1}{2}\pi$, $dt = \pi/2\, dy$. Thus, (4.27) is equivalent to

$$I_m(t) = \frac{\pi}{2}\int_{-1}^{1} \ln\left|\frac{\mathbf{z}(t) - \mathbf{z}(\frac{y'+1}{2}\pi)}{\cos(t) - \cos(\frac{y'+1}{2}\pi)}\right| \cos\left(\frac{m(y'+1)}{2}\pi\right) dy'. \tag{4.28}$$

To approximate this integral with the spectral accuracy we apply the Clenshaw-

Curtis quadrature on the Chebyshev nodes $\{y_i = \cos(\frac{\pi i}{M}),\ i = 0, 1, .., M\}$ [17] :

$$I_m(t) = \frac{\pi}{2} \sum_{i=0}^{M} w_i \rho_m(t, y_i), \tag{4.29}$$

where $\rho_m(t, y_i) = \ln \left| \frac{\mathbf{z}(t) - \mathbf{z}(\frac{y_i+1}{2}\pi)}{\cos(t) - \cos(\frac{y_i+1}{2}\pi)} \right| \cos\left(\frac{m(y_i+1)}{2}\pi\right)$.

This approach avoids the regularity issue of Option 1 and gives the benefit of spectral accuracy.

Considering (4.26) and (4.28) we conclude that two overlapping discretizations of the curve $\mathcal{C}$ are needed for solving BIE (4.23):

- uniform in the arc length :

$$s_j = \frac{L t_j}{\pi}, \quad t_j = \frac{2j+1}{2(M+1)}\pi, \quad j = 0, 1, .., M, \tag{4.30}$$

  These points will serve for the collocation approach applied to (4.26). We emphasize here that there are two equivalent representations of Chebyshev polynomials T_m, namely $T_m(t) = \cos(mt)$, $0 \le t \le \pi$ and $T_m(x) = \cos(m \arccos x)$, $-1 \le x \le 1$, so we can think of collocation points $x_j = \cos(t_j)$ as Chebyshev in x or as uniform in t. Since both sets of points result in the uniform grid in arc length $s(t) = \frac{Lt}{\pi}$ on the curve $\mathcal{C}$, we will refer to these sets as a "uniform" grid.

- Chebyshev in the arc length :

$$s_i = \frac{L}{2}(y_i + 1), \quad y_i = \cos\left(\frac{\pi i}{M}\right), \quad \hat{t}_i = \frac{y_i + 1}{2}\pi, \quad i = 0, 1, .., M \tag{4.31}$$

  involved in the Clenshaw-Curtis quadratures such as (4.29). We note that Chebyshev nodes $\{y_i\} \in [-1, 1]$ correspond to Chebyshev points in the arc length $s_i = \frac{L}{2}(y_i + 1) \in [0, L]$, $i = 0, 1, .., M$. To highlight this property we will refer to

Figure 4.4: Schematic illustration of the different grids used to discretize the contour $\mathcal{C}$: (circles) - Chebyshev grid, (squares) - uniform grid.

these points as Chebyshev grid.

Figure 4.4 illustrates two grids constructed to discretize the problem.

Alltogether, the collocation method and Clenshaw-Curtis quadrature transform (4.26) to the system of $M+1$ equations with respect to the expansion coefficients a_m, $m = 0, 1, .., M$:

$$\begin{aligned} -\sum_{m=0}^{M} a_m T_m(t_j) + \frac{\gamma L}{4\pi k} \sum_{m=0}^{M} a_m \sum_{i=0}^{M} w_i \ln \left| \frac{\mathbf{z}(t_j) - \mathbf{z}(\hat{t}_i)}{\cos(t_j) - \cos(\hat{t}_i)} \right| \cos(m\,\hat{t}_i) - \\ -\frac{\gamma L}{2\pi k} \left(a_0 \ln(2) + \sum_{m=1}^{M} \frac{a_m}{m} T_m(t_j) \right) = \frac{\gamma}{k} (u_p(t_j) - u_0(t_j)), \quad i, j = 0, 1, .., M. \end{aligned} \tag{4.32}$$

In the matrix form problem (4.32) can be represented as

$$[\mathbb{T} + c_1 \mathbb{K}_1 + c_2 \mathbb{K}_2] \cdot \mathbf{a} = [\frac{\gamma}{k} (\tilde{\mathbf{U}}_0 - \mathbb{P} \cdot \mathbf{U})],$$

where $\mathbb{T} = [T_m(t_j)]$ is a matrix of Chebyshev polynomials evaluated on a uniform grid, $\mathbf{a} = [a_0, ..., a_M]$ and $\mathbb{K}_1 = [I_{jm}]$ is a matrix representing the integral with removable singularity (4.28):

$$I_{jm} = \sum_{i=0, t_j \neq \hat{t}_i}^{M} w_i \ln \left| \frac{\mathbf{z}(t_j) - \mathbf{z}(\hat{t}_i)}{\cos(t_j) - \cos(\hat{t}_i)} \right| T_m(\hat{t}_i) + w_j \ln \left| \frac{L}{\pi \sin(t_j)} \right| T_m(t_j).$$

Operator $\mathbb{K}_2 = [\frac{T_m(t_j)}{m}]$, in which the entries corresponding to $m = 0$ are replaced with

$\ln(2)$, represents the improper integral, $\mathbb{P}$ is an interpolation operator, $\tilde{\mathbf{U}}_0$ is a vector given by elements $u_0(t_j)$, $j = 0, ..., M$ and coefficients $c_1 = -\frac{\gamma L}{4\pi k}$ and $c_2 = \frac{\gamma L}{2\pi k}$.

Finally, together with (4.22) we get the following matrix representation of (4.21a) and (4.21b):

$$\begin{bmatrix} -\mathbf{\Delta}_N & \mathbf{0} \\ \frac{\gamma}{k}\mathbb{P} & \mathbb{T} + c_1\mathbb{K}_1 + c_2\mathbb{K}_2 \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{U}} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \frac{1}{k}\tilde{\mathbf{Q}} \\ \frac{\gamma}{k}\tilde{\mathbf{U}}_0 \end{bmatrix}. \tag{4.33}$$

4.4.3 Discretizing Boundary Conditions (4.21c)

It remains to include the boundary conditions

$$\frac{\partial u_p}{\partial n} = -\frac{\partial u_h}{\partial n}, \qquad \text{on } \partial\Omega$$

into the matrix representation (4.33) of direct problem (4.21).

Suppose we are at the boundary position $\mathbf{x}_0^c = (-1, y_0)$ (see Figure 4.5) which corresponds to the c-th element in the rearranged vector $\mathbf{U}$. Since we are at $x = -1$,

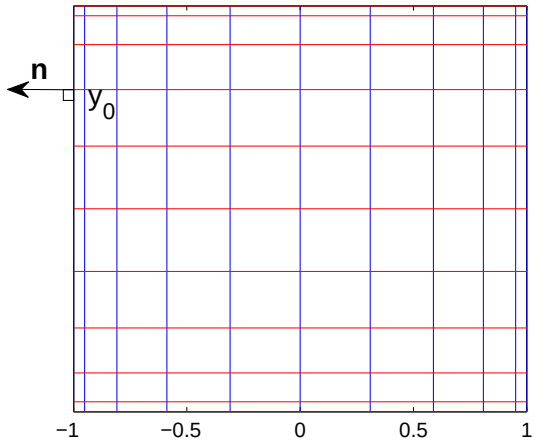


Figure 4.5: Sketch of the normal vector at the boundary position $\mathbf{x}_0^c$

we have $\mathbf{n} = (-1, 0)$ and :

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

Then (4.21c) has the form:

$$\frac{\partial u_p}{\partial x} = \frac{\partial u_h}{\partial n}\Big|_{x_0^c}.$$

Thus, for the RHS of (4.21c) combined with (4.12) we have

$$\frac{\partial u_h}{\partial n}\Big|_{x_0^c} = \nabla u_h \cdot \mathbf{n}\Big|_{x_0^c} = -\frac{1}{2\pi}\int_{\mathcal{C}} \frac{\langle \mathbf{n}, \mathbf{x} - \mathbf{x}_{\mathcal{C}} \rangle}{|\mathbf{x} - \mathbf{x}_{\mathcal{C}}|^2} \mu(\mathbf{x}_{\mathcal{C}})\, d\sigma\Big|_{x_0^c}.$$

Next, we transform the integral to the definite integral over $[-1, 1]$, substitute $\mathbf{n} = (-1, 0)$ and the expansion of density function μ in terms of the Chebyshev polynomials:

$$\frac{\partial u_h}{\partial n}\Big|_{x_0^c} = -\frac{L}{4\pi}\sum_{m=0}^{M} a_m \int_{-1}^{1} \frac{1 + z_1(\frac{y+1}{2}\pi)}{|x_0^c - \mathbf{z}(\frac{y+1}{2}\pi)|^2}\, T_m\Big(\frac{y+1}{2}\pi\Big)\, dy.$$

Then, we apply the Clenshaw-Curtis quadrature:

$$\frac{\partial u_h}{\partial n}\Big|_{x_0^c} \approx -\frac{L}{4\pi}\sum_{m=0}^{M} a_m \sum_{i=0}^{M} w_i \frac{1 + z_1(\hat{t}_i)}{|x_0^c - \mathbf{z}(\hat{t}_i)|^2}\, T_m(\hat{t}_i) = \mathbf{b}^{(c)} \cdot \mathbf{a},$$

where $\hat{t}_i = \frac{y_i+1}{2}\pi$ and $\mathbf{b}^{(c)}$ is a row vector with the elements:

$$\mathbf{b}_j^{(c)} = \frac{L}{4\pi}\, w_i \frac{1 + z_1(\hat{t}_i)}{|x_0^c - \mathbf{z}(\hat{t}_i)|^2}\, T_m(\hat{t}_i), \quad j = 1, ..., M.$$

Recall that for the LHS of (4.21c) we have

$$\frac{\partial u_p}{\partial x} \approx \mathbf{D}_N \otimes I_{N+1} \cdot \tilde{\mathbf{U}}, \tag{4.34}$$

where the numerical differentiation technique from Section 4.1.1 is applied.

Now, linking the LHS and RHS together at the c-th position we have:

$$-(\mathbf{D}_N \otimes I_{N+1})_c \, \tilde{\mathbf{U}} + \mathbf{b}^{(c)} \cdot \mathbf{a} = 0, \tag{4.35}$$

where $(\mathbf{D}_N \otimes I_{N+1})_c$ is the c-th row of $\mathbf{D}_N \otimes I_{N+1}$. To impose the boundary condition at $\mathbf{x}_0^c$ we have to replace the c-th row of $\mathbf{\Delta}_N$ in (4.33) with $-(\mathbf{D}_N \otimes I_{N+1})_c$ and the c-th row of zero matrix $\mathbf{0}$ with $\mathbf{b}^{(c)}$. 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 (4.21) is:

$$\begin{bmatrix} -\mathbf{\Delta}'_N & \mathbb{B} \\ \frac{\gamma}{k}\mathbb{P} & \mathbb{T} + c_1\mathbb{K}_1 + c_2\mathbb{K}_2 \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{U}} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \frac{1}{k}\tilde{\mathbf{Q}}' \\ \frac{\gamma}{k}\tilde{\mathbf{U}}_0 \end{bmatrix} \tag{4.36}$$

in which $\mathbf{\Delta}'_N$, $\mathbb{B}$ and $\tilde{\mathbf{Q}}'$ correspond to, respectively, $\mathbf{\Delta}_N$, $\mathbf{0}$ and $\tilde{\mathbf{Q}}$ in (4.33) after imposing the boundary conditions. Thus, we have described the way we solve direct problem (2.1).

Note that for the adjoint system (3.43) the same strategy is applied. The adjoint variable u^* is now given by the sum $u^* = u_h^* + u_p^*$ and the vector $\mathbf{a}^*$ corresponds to the expansion coefficients of the density function of single-layer potential u_h^* in the Chebyshev polynomials. This leads to the algebraic system similar to (4.36), but with different right-hand side:

$$\begin{bmatrix} -\mathbf{\Delta}'_N & \mathbb{B} \\ \frac{\gamma}{k}\mathbb{P} & \mathbb{T} + c_1\mathbb{K}_1 + c_2\mathbb{K}_2 \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{U}}^* \\ \mathbf{a}^* \end{bmatrix} = \begin{bmatrix} \frac{1}{k}\tilde{\mathbf{Q}}^* \\ \mathbf{0} \end{bmatrix}, \tag{4.37}$$

where $\tilde{\mathbf{U}}^*$ corresponds to the rearranged vector representing u_p^* and $\tilde{\mathbf{Q}}^*$ is the reordering of $(u - \bar{u})\chi_{A_i}$, $i = 1, 2$, evaluated on 2D Chebyshev grid after imposing boundary conditions of the adjoint system.

4.4.4 Discretization of $u_h(x)$

By solving (4.21) we obtain an approximation of u_p in Ω and u_h on $\mathcal{C}$. In order to derive the values of temperature distribution u in domain Ω by (4.7) it remains to find u_h in $\Omega \setminus \mathcal{C}$. Recall that by (4.19)

$$u_h(\mathbf{x}) = -\frac{1}{2\pi} \oint_{\mathcal{C}} \ln|\mathbf{x} - \mathbf{x}_{\mathcal{C}}| \mu(\mathbf{x}_{\mathcal{C}})\, d\sigma \quad \text{in } \Omega \setminus \mathcal{C}.$$

To remove the singularity as $\mathbf{x} \longrightarrow \mathbf{x}_{\mathcal{C}}$ we use the same splitting technique as for the equation (4.21b) and then collocate it on the same grid. In terms of operators introduced in Section 4.4.2 we have

$$\mathbf{U}_h = \Big[-\frac{L}{4\pi}\mathbb{K}_1 + \frac{L}{2\pi}\mathbb{K}_2 \Big] \cdot \mathbf{a},$$

where $\mathbf{U}_h = u_h(t_i)$, $i = 0, 1, ..., M$ given on uniform grid (4.30). Note that $u_h(\mathbf{x})$ for $\mathbf{x} \in \Omega \setminus \mathcal{C}$ is given by *regular* integral (4.19), so we can directly apply the Clenshaw-Curtis quadrature:

$$u_h(t) \approx -\frac{L}{4\pi} \sum_{i=0}^{M} w_i \, \ln|\mathbf{z}(t) - \mathbf{z}(\hat{t}_i)| \, \mu(\hat{t}_i),$$

where $\hat{t}_i$, $i = 0, 1, .., M$ is Chebyshev grid (4.31).

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

Solutions of the direct and adjoint problems serve to compute the gradient (3.42):

$$\begin{aligned} \boldsymbol{\nabla}^{L^2}_{L_0}\mathcal{J} = &- \gamma\,(u_1 - u_0)\Big(\kappa u_1^* + \frac{\partial u_1^*}{\partial n}\Big) - \gamma\,\frac{\partial u_2}{\partial n}\,u_1^* \\ &- \frac{\gamma\kappa(u_{out} - u_{in})}{L} \int_0^L \left[H(\tilde{s} - s) - \frac{\tilde{s}}{L} \right] u_1^*(\tilde{s})\, d\tilde{s} + \alpha\,\Big(\int_{\mathcal{C}} ds - L_0 \Big)\kappa \quad \text{on } \mathcal{C}, \end{aligned}$$

where $\alpha = 0$ corresponds to the problem without the length constraint. Since u_1 and u_1^* can be obtained in a straightforward manner from the solutions of the corresponding systems, we focus on evaluation of the normal derivatives $\frac{\partial u_2}{\partial n}$ and $\frac{\partial u_1^*}{\partial n}$, entering the formula for the gradient (3.42). By construction, the normal derivatives $\frac{\partial u_1}{\partial n}$ and $\frac{\partial u_2}{\partial n}$ are given by:

$$\begin{aligned} \frac{\partial u_1}{\partial n} &= \frac{\partial u_h^-}{\partial n} + \frac{\partial u_p}{\partial n}, \\ \frac{\partial u_2}{\partial n} &= \frac{\partial u_h^+}{\partial n} + \frac{\partial u_p}{\partial n}, \end{aligned} \tag{4.38}$$

where we applied the fact that $u_p \in C^2(\Omega)$, i.e. $\frac{\partial u_p}{\partial n} := \frac{\partial u_p^-}{\partial n} = \frac{\partial u_p^+}{\partial n}$. Let us first treat the normal derivatives of u_h. Using the properties (4.14) and (4.15) of density function μ on $\mathbf{x} \in \mathcal{C}$, we obtain:

$$\frac{\partial u_h^+}{\partial n} = -\frac{1}{2\pi} \int_{\mathcal{C}} \frac{\langle \mathbf{n}(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle}{|\mathbf{x} - \mathbf{y}|^2} \, \mu(\mathbf{y}) \, d\mathbf{y} - \frac{1}{2} \, \mu(\mathbf{x}), \tag{4.39a}$$

$$\frac{\partial u_h^-}{\partial n} = -\frac{1}{2\pi} \int_{\mathcal{C}} \frac{\langle \mathbf{n}(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle}{|\mathbf{x} - \mathbf{y}|^2} \, \mu(\mathbf{y}) \, d\mathbf{y} + \frac{1}{2} \, \mu(\mathbf{x}). \tag{4.39b}$$

To approximate the integral on the right hand side, we transform the domain of integration to $[-1, 1]$ and after applying the Clenshaw-Curtis formula we have:

$$-\frac{1}{2\pi} \int_{\mathcal{C}} \frac{\langle \mathbf{n}(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle}{|\mathbf{x} - \mathbf{y}|^2} \, \mu(\mathbf{y}) \, d\mathbf{y} \approx -\frac{L}{4\pi} \sum_{l=0}^{M} w_l \, \frac{\langle \mathbf{n}(\mathbf{x}), \mathbf{z}(t) - \mathbf{z}(\hat{t}_l) \rangle}{|\mathbf{z}(t) - \mathbf{z}(\hat{t}_l)|^2} \, \mu(\hat{t}_l), \tag{4.40}$$

where $\hat{t}_l$ are the Chebyshev points. Next, we collocate (4.39a), (4.39b) at Chebyshev points $\hat{t}_i$, and replace all terms where $i = l$ with $\frac{\kappa}{2} \, \mu(\hat{t}_i)$ followed from (4.17). Finally, to compute the curvature κ recall that the curve $\mathcal{C}$ is given by the map $\mathbf{z}(t) = [z_1(t), z_2(t)]$, so

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

To calculate the derivatives of $\mathbf{z}$ with spectral accuracy we apply the technique described in Section 4.1.1.

Now, it remains to discuss $\frac{\partial u_p}{\partial n}$ which is given by $\frac{\partial u_p}{\partial n} = \nabla u_p \cdot \mathbf{n}$. The gradient of u_p can be approximated with spectral accuracy by the Chebyshev differentiation method introduced in Section 4.1.1 as $(\mathbf{U}_x, \mathbf{U}_y) \approx (I_{N+1} \otimes \mathbf{D}_N \mathbf{U},\, \mathbf{D}_N \otimes I_{N+1} \mathbf{U})$. The values of $\frac{\partial u_p}{\partial n}$ for $\mathbf{x} \in \mathcal{C}$ are derived by applying interpolation operator $\mathbb{P}$ to the gradient: $\frac{\partial u_p}{\partial n} \approx \mathbb{P}\left(\mathbf{U}_x, \mathbf{U}_y\right) \cdot \mathbf{n}$. To evaluate $\frac{\partial u_1^*}{\partial n}$ and $\frac{\partial u_2^*}{\partial n}$ we follow the same idea.

4.4.6 Sobolev Gradient and Evaluation of Cost Functional

After solving the direct and adjoint PDE systems, and then computing $\frac{\partial u_2}{\partial n}$ and $\frac{\partial u_1^*}{\partial n}$ we can evaluate the gradient $\boldsymbol{\nabla}^{L^2} \mathcal{J}$ given by (3.41). As was mentioned in Section 3.3.3, the resulting gradient is not useful for actual contour modification, since it does not guarantee the smoothness of the updated contour. For this purpose, we introduced Sobolev gradient $\boldsymbol{\nabla}^{H^1} \mathcal{J}$ as a solution of the PDE with Neumann boundary condition (3.45).

For the numerical solution of given PDE on bounded non-periodic domain, we apply spectral differentiation matrix $\mathbf{D}_N^2$ to discretize the second derivative operator, so that the governing equation is given by:

$$\Big(I_N - \frac{4l^2}{L^2}\, \mathbf{D}_N^2\Big) \tilde{\boldsymbol{\nabla}}^{H^1} \mathcal{J} = \tilde{\boldsymbol{\nabla}}^{L^2} \mathcal{J},$$

where $\tilde{\boldsymbol{\nabla}}^{H^1} \mathcal{J}$ and $\tilde{\boldsymbol{\nabla}}^{L^2} \mathcal{J}$ are column vectors representing the gradients and the factor $\frac{4}{L^2}$ is the result of curve parametrization in $[-1, 1]$. To enforce Neumann boundary conditions we replace the first and last row of $(I_N - \frac{4l^2}{L^2}\, \mathbf{D}_N^2)$ with corresponding rows of $\mathbf{D}_N$, and the first and last elements of the vector $\tilde{\boldsymbol{\nabla}}^{L^2} \mathcal{J}$ on the right hand side with zeros.

4.5 Optimization Algorithm

4.5.1 Conjugate Gradient Method

To minimize the cost functional (2.3) and to update the shape of the contour at each iteration we use the following gradient-descent algorithm:

$$\begin{aligned} \mathbf{x}_{\mathcal{C}^{(i)}} &= \mathbf{x}_{\mathcal{C}^{(i-1)}} - \tau^{(i-1)} \nabla \mathcal{J}(\mathcal{C}^{(i-1)}), \quad i = 1, 2, ... \\ \mathbf{x}_{\mathcal{C}^{(0)}} &= \mathbf{x}_{\mathcal{C}_0}, \end{aligned} \tag{4.41}$$

where the contour $\mathcal{C}_0$, which meets the domain boundary at the right angle (see Section 2.1), represents the initial guess and $\tau^{(i-1)}$ is the length of the step along the descent direction at the $(i-1)$-th iteration. To find this step we need to solve a line minimization problem:

$$\tau^{(i)} = \text{argmin}_{\tau>0} \mathcal{J}(\mathcal{C}^{(i)} - \tau \nabla \mathcal{J}(\mathcal{C}^{(i)})). \tag{4.42}$$

For one-dimensional minimization in (4.42) we use Brent's method [21]. It combines a golden section search designed to handle the worst possible case of function minimization and, if function is nicely parabolic near to the minimum, the parabola fitted through any three points to take us very close to minimum in a single step. To find the optimal step $\tau^{(i)}$, Brent's method needs to evaluate the cost functional in (4.42) multiple times for different values of τ. This requires solving our direct problem for each such evaluation and therefore, the search of the parameter $\tau^{(i)}$ is the most expensive part of the algorithm.

We emphasize that, while for the sake of brevity of notation formula (4.41) represents the steepest descent approach, we use the Polak-Ribiere version of the nonlinear conjugate gradient method [22] to obtain the results presented in Section 5.3. Thus,

we use the Polak-Ribiere formula

$$\beta_i = \frac{\nabla \mathcal{J}^T(\mathcal{C}^{(i)})\Big(\nabla \mathcal{J}(\mathcal{C}^{(i)}) - \nabla \mathcal{J}(\mathcal{C}^{(i-1)})\Big)}{\nabla \mathcal{J}^T(\mathcal{C}^{(i-1)})\nabla \mathcal{J}(\mathcal{C}^{(i-1)})} \tag{4.43}$$

to update the conjugate direction

$$P^{(i)} = -\nabla \mathcal{J}(\mathcal{C}^{(i)}) + \beta_i P^{(i-1)}.$$

4.5.2 Optimization Algorithm

In this section we summarize the iterative algorithm which solves our optimization problem.

Algorithm 1.

Step 0. Choose an initial guess for the contour $\mathcal{C}^{(0)} = \mathcal{C}_0$. Set $i = 0$.

Step 1. Obtain the gradient $\nabla^{L_2}\mathcal{J}(\mathcal{C}^{(i)})$ by solving direct and adjoint problems (2.1), (3.43).

Step 2. Compute smoother (Sobolev) gradient $\nabla^{H^1}\mathcal{J}(\mathcal{C}^{(i)})$ as a solution of the system (3.45). Set $P^{(i)} = -\nabla^{H^1}\mathcal{J}(\mathcal{C}^{(i)})$.

Step 3. Proceed by bracketing the local minimum of the cost functional $\mathcal{J}(\mathcal{C}^{(i)} + \tau P^{(i)})$ and apply linear search method to find $\tau^{(i)} = \text{argmin}_{\tau>0}\mathcal{J}(\mathcal{C}^{(i)} + \tau P^{(i)})$.

Step 4. Update the curve $\mathcal{C}^{(i+1)} = \mathcal{C}^{(i)} + \tau^{(i)} P^{(i)}$.

Step 5. Compute β_{i+1} by Polak-Ribiere formula (4.43).
Update the conjugate gradient $P^{(i+1)} = -\nabla^{H^1}\mathcal{J}(\mathcal{C}^{(i+1)}) + \tilde{\beta}_{i+1}P^{(i)}$,
where $\tilde{\beta}_{i+1} = \max(\beta_{i+1}, 0)$ and $\tilde{\beta}_{i+1}$ is set to 0 every 20 iterations.

Step 6. Set $i = i + 1$.
Exit if $\| \nabla^{H^1}\mathcal{J}(\mathcal{C}^{(i)}) \|_{L_2} < Tol$ which is equivalent to the optimality condition $\nabla \mathcal{J} = 0$ reached with some tolerance Tol.
Else: go to **Step 1**.

Chapter 5

Results

5.1 Validation of BIE Method

Recall that our direct problem (4.18) has an equivalent representation (4.21) with the boundary integral equation (4.21b). In this section we describe the test which validates the numerical approach developed for BIE (4.21b), which we restate here for completeness:

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

This is a Fredholm integral equation of the second kind with a weakly singular kernel.

The development of an appropriate numerical scheme for this BIE is a key to construction of highly-accurate discretization of the whole system (4.21). So we have to choose an efficient way to approximate the integral with removable singularity (4.27). Our Option 1 was a Gaussian-type approach, namely the Chebyshev-Gauss quadrature. We have shown analytically that the lack of smoothness of the integrand prevents us from getting spectrally accurate result for this type of the numerical scheme. Our

Option 2 was the Clenshaw-Curtis quadrature which from the analytical point of view should give us the desired rate of convergence. Note that we solve BIE (4.21b) by the Chebyshev collocation approach combined with one of the quadratures, so the error of the methods is a sum of the error of collocation approach and the error of the quadrature itself.

Let us compare the accuracy of both methods: one involving the Gaussian quadrature and the other with the Clenshaw-Curtis quadrature, by solving BIE (4.21b) in isolation from other equations in (4.21) with different "manufactured" solutions. By comparing a known actual solution and the numerical solutions obtained on successively refined grids, the method of the manufactured solution allows us to demonstrate that the equations are solved to the theoretical order of accuracy of the discretization methods applied.

This approach consists of the following steps:

1. choose some function to be a solution of the equation,
2. substitute this function back to the equation and find corresponding right hand side of the equation that would guarantee our solution satisfies the equation,
3. solve numerically the equation with the customized RHS derived in step 2,
4. compare the numerical solution from step 3 with analytic function that was chosen in step 1.

It is a non-trivial task to compute the singular integral in (4.21b) analytically in a general case, so we restrict ourselves to the contour $\mathcal{C}$ given by $\mathbf{z}(t) = [\sin(t), \cos(t)]$, $t \in [0, \pi]$ which leads to a simpler form of the singular kernel $\ln|\mathbf{z}(t) - \mathbf{z}(t')|$.

Consider now two manufactured solutions: $\tilde{\mu}_1(t) = \cos(t)$ and $\tilde{\mu}_2(t) = \sin(t)$. In the first case, Figure 5.1 demonstrates the spectral rate of convergence of the collocation

method with the Clenshaw-Curtis quadrature, while the Gaussian quadrature only shows an algebraic accuracy. Note that since both methods employ the same collocation approach, the difference in their performance is due to the quadrature that is used in each method. So, as was predicted based on the smoothness of the integrands in Option 1 and Option 2, the numerical scheme with the Chebyshev-Gauss quadrature cannot compete with the Clenshaw-Curtis quadrature due to locally singular change of variables required to transform the integral to the Gaussian type.

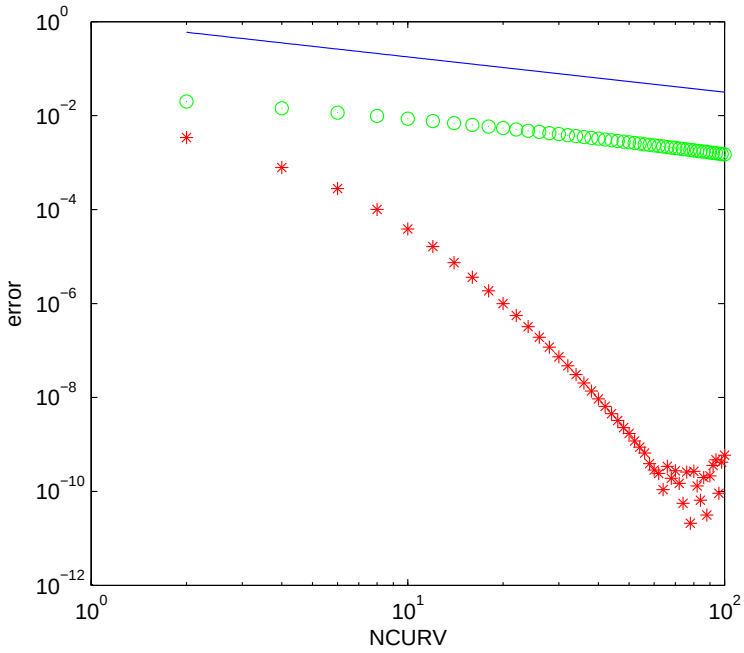


Figure 5.1: The accuracy of BIE method with density $\mu(t) = \cos(t)$: (asterisks) - the Clenshaw-Curtis quadrature approach, (circles) - the Chebyshev-Gauss approach, (solid line) - a line with the slope -3/4

On the other hand, in the second case when $\tilde{\mu}_2(t) = \sin(t)$ both approaches give only algebraic accuracy with Clenshaw-Curtis method showing slightly better results (see Figure 5.2). In this case, the lack of smoothness comes from the representation of $\tilde{\mu}_2(t)$. Expansion of density function μ in the truncated Chebyshev series involved

in the collocation approach requires $\mu(\tau)$, where $\tau = \cos(t)$, $t \in [0, \pi]$, $\tau \in [-1, 1]$, to be smooth in τ to provide spectrally accurate results. Note that in the present case $\tilde{\mu}_2(\tau) = \sqrt{1-\tau^2}$, $\tau \in [-1, 1]$ and its first derivative has a singularity at the end points $\tau = -1$ and $\tau = 1$. So here the algebraic errors in both options come from the limitations of the collocation method. We can nonetheless conclude that, the approach that employs the Clenshaw-Curtis quadrature has better properties than the one that uses the Gaussian quadrature, especially in cases when the smoothness requirements of the collocation method are resulting in the spectral convergence.

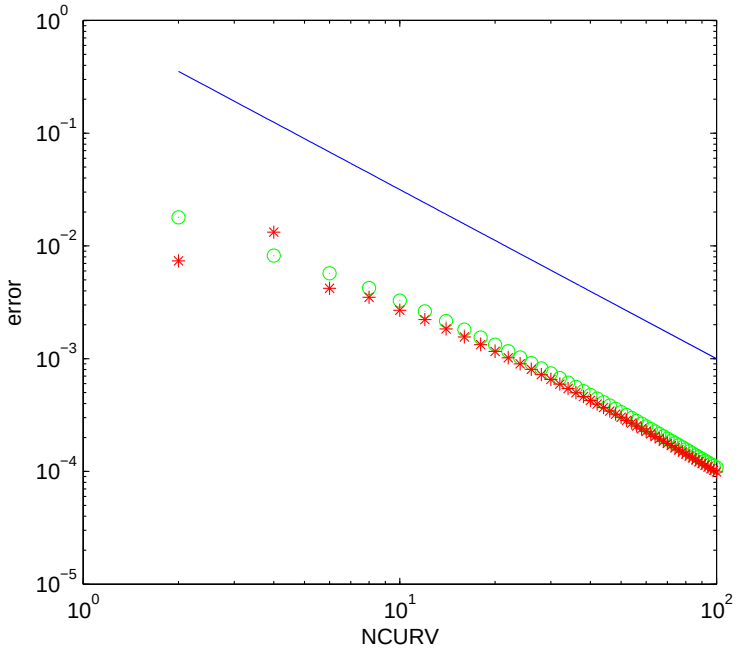


Figure 5.2: The accuracy of BIE method with density $\mu(t) = \sin(t)$: (asterisks) - the Clenshaw-Curtis quadrature approach, (circles) - the Chebyshev-Gauss approach, (solid line) - a line with the slope -3/2

5.2 Validation of the Cost Functional Gradient

In this section we present the test which demonstrates consistency of the gradient computations. Recall that, according to Riesz representation (3.13), for any perturbation ζ of contour $\mathcal{C}$, the shape derivative of the cost functional can be given by the following inner product in L_2:

$$\mathcal{J}'_\epsilon(\mathcal{C};\, \zeta\mathbf{n}) = \int_{\mathcal{C}} \boldsymbol{\nabla}^{L_2} \mathcal{J}\zeta \, ds.$$

We define the quantity

$$\kappa(\epsilon) \;:=\; \frac{\mathcal{J}'_\epsilon(\mathcal{C};\, \zeta\mathbf{n})}{\int_{\mathcal{C}} \boldsymbol{\nabla}^{L_2} \mathcal{J}\zeta \, ds}, \tag{5.1}$$

where $\mathcal{J}'_\epsilon(\mathcal{C};\, \zeta\mathbf{n})$ denotes a finite-difference approximation of the directional derivative with respect to the shape of $\mathcal{C}$ in the direction of some perturbation ζ:

$$\mathcal{J}'_\epsilon(\mathcal{C};\, \zeta\mathbf{n}) = \frac{\mathcal{J}(\mathcal{C} + \epsilon\,\zeta\mathbf{n}) - \mathcal{J}(\mathcal{C})}{\epsilon}.$$

Deviation of $\kappa(\epsilon)$ from 1 is a measure of the inconsistency of the gradient.

To test our gradient, we plot this quantity for a range of ϵ values, starting from very small and increasing by several orders of magnitude. We summarize our test settings in the table below.

	Contour $\mathcal{C}$	$u_0(s)$	Perturbations	Resolution (N, M)	Domain
κ-test 1	$\mathcal{C}_0$	$u_0(s) \equiv 10$	ζ_1, ζ_2, ζ_3 and ζ_4	(30, 50), (41, 100), (52, 300)	Ω
κ-test 2	$\mathcal{C}_0$	$u_0(s) = 5 + 10s$	ζ_1, ζ_2, ζ_3 and ζ_4	(30, 50), (41, 100), (52, 300)	Ω

Table 5.1: Settings for different κ-tests defined in (5.1)

Here we briefly comment on Table 5.1. The second column is the cooling contour we are using for the tests shown in Figure 5.3. The contour $\mathcal{C}_0$ represents a straight line at the position $x = 0.15$. The third column contains the information about the reference temperature along the contour (cf. (2.2)). For the first test we are using constant value u_0, for the second test we assume that the temperature is changing along the contour linearly. The fourth column specifies all the perturbations applied to the contour during the test. In each case we are using four different perturbations $\zeta_j(t)$, $j = 1, 2, 3, 4$, for the contour displacement given as:

$$\zeta_j(t) = \begin{cases} 1, & -1 \le t \le -\frac{1}{2}, \\ \cos\left(j\frac{(2t+1)\pi}{2}\right), & -\frac{1}{2} \le t \le \frac{1}{2}, \\ (-1)^j, & \frac{1}{2} \le t \le 1, \end{cases} \qquad j = 1, 2, 3, 4. \tag{5.2}$$

Thus, the points of the perturbed contour satisfy the following identity

$$\mathbf{x}_{\mathcal{C}}^j = \mathbf{x}_{\mathcal{C}} + \epsilon\, \zeta_j\, \mathbf{n}, \quad j = 1, 2, 3, 4. \tag{5.3}$$

For the sake of clarity we illustrate these perturbations in Figure 5.3.

The fifth column specifies the resolutions applied in different cases where we use a 2D Chebyshev grid of size $(N + 1) \times (N + 1)$ for the domain and M Chebyshev points for the contour representation. The last column is the domain of our interest where $\Omega = [-1, 1] \times [-1, 1]$. For all κ-tests here we use the heat sources q and target temperature distribution $\bar{u}$ given by:

$$q(x, y) = 50 - 15x^2 - 15(y - 0.5)^2,$$
$$\bar{u}(x, y) = 15 + \sin(2\pi x)\sin(2\pi y).$$

Instead of plotting $\kappa(\epsilon)$ itself, we plot the values of $\log_{10}|\kappa(\epsilon)-1|$ to determine the accuracy of the κ-test in terms of significant digits.

Results of κ-test 1 are displayed in Figure 5.4. We observe that for different perturbations ζ_j, $j=1,2,3,4$, the value of κ stays very close to 1 for the intermediate values of ϵ. The deviation of κ from 1 for very small values of ϵ is the result of subtractive cancellation errors and for big values of ϵ it is the result of poor approximation of the first derivative of the cost functional by finite differences. For intermediate values of ϵ, where we might expect κ to be free from round-off and truncation errors, we do not observe a significant improvement in $\kappa(\epsilon)$ approximation, i.e., a "wedge" shape in the plot of ϵ, which is due to the presence of other computational errors such as interpolation errors. The closeness of κ to 1 confirms that the cost functional gradient $\nabla\mathcal{J}$ is evaluated accurately. Moreover, as we increase the resolution, we obtain a better estimate for the gradient.

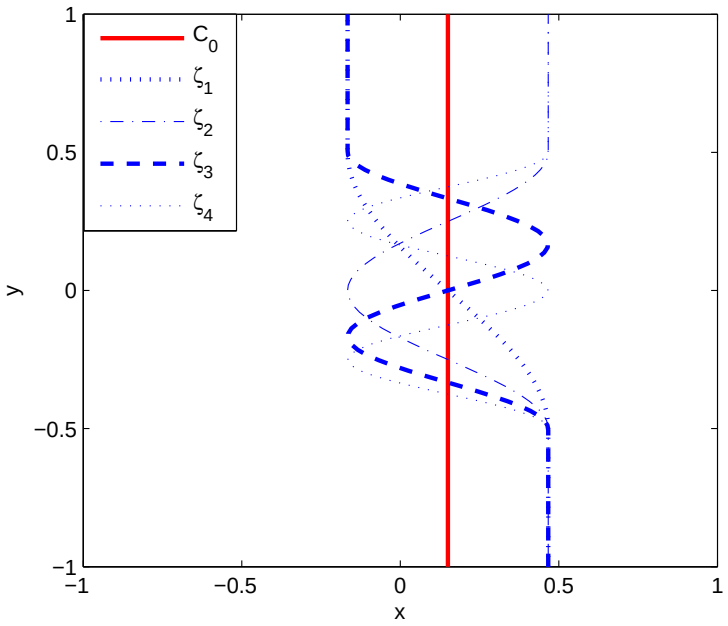
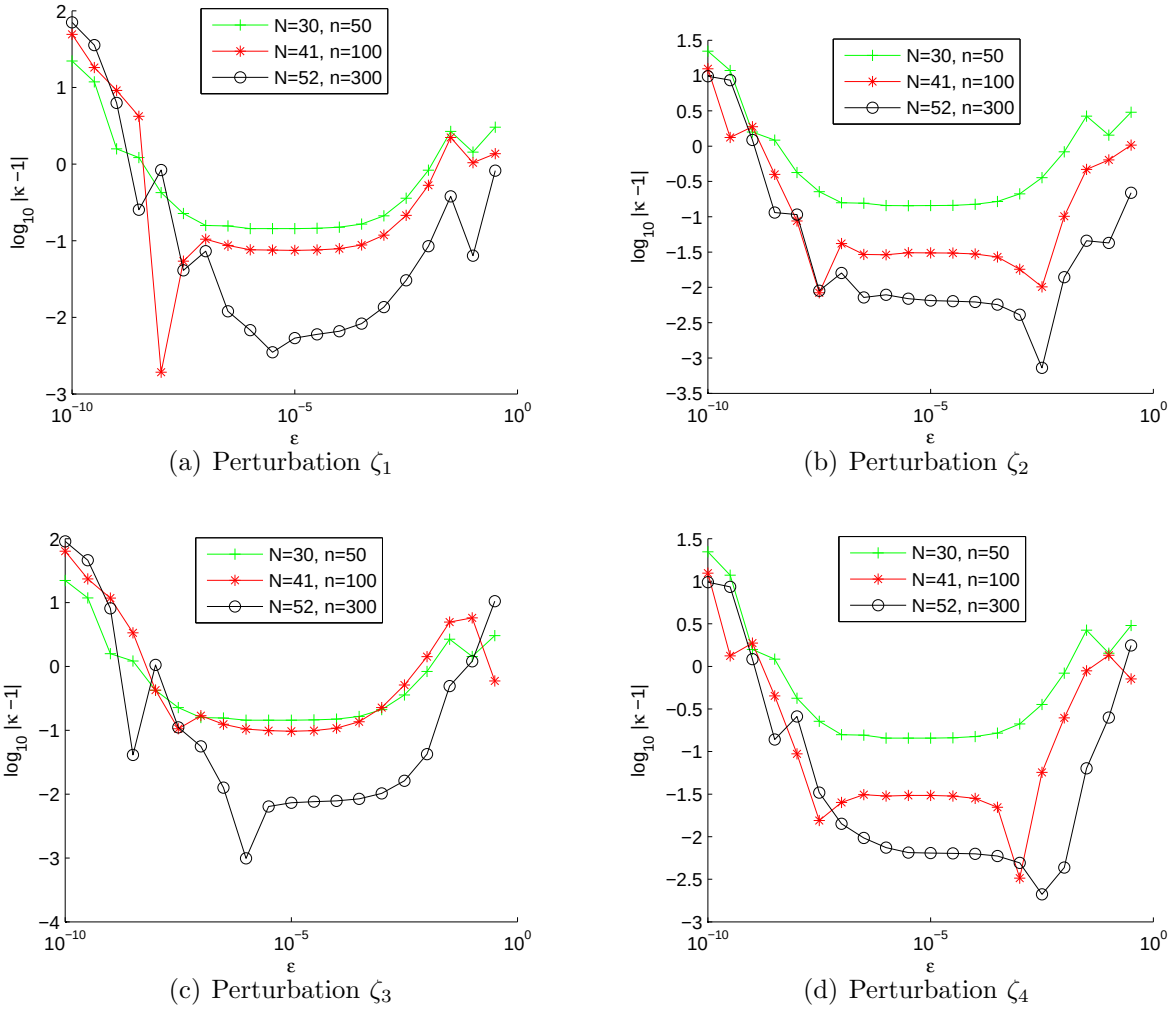


Figure 5.3: Initial contour $\mathcal{C}_0$ and perturbed contours (5.3) used in the κ-test

Results of κ-test 2 are illustrated in Figure 5.5. Here we see that κ also stays close to one, except for the regions where the round off errors or big approximation errors for the first derivative of $\mathcal{J}$ occur. Note that in both tests we have an accuracy up to 2 significant digits or higher. Finally, we conclude that the validation tests exhibit the anticipated results with refinement of resolution leading to increasing accuracy of the cost functional gradient $\nabla\mathcal{J}$.



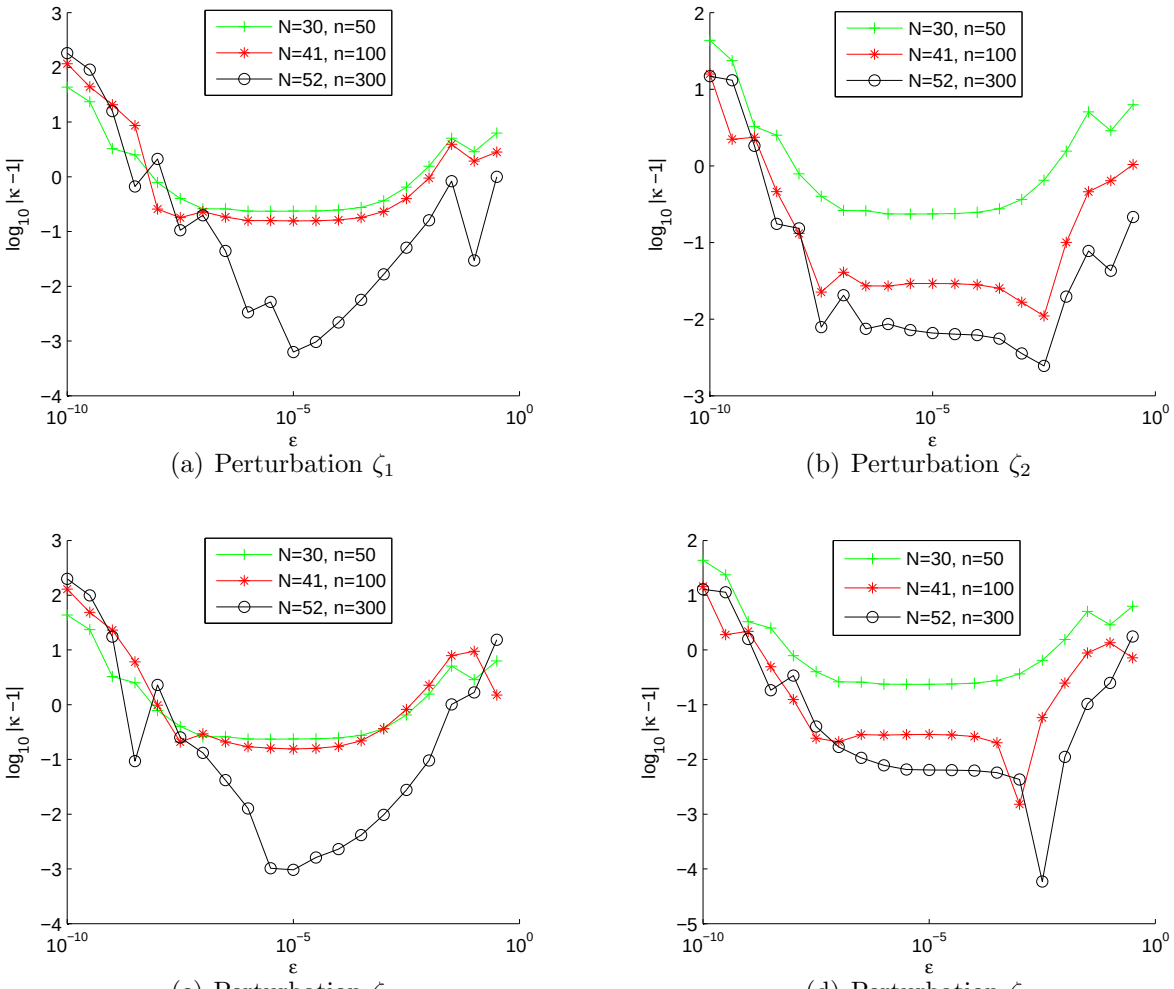
(a) Perturbation ζ_1

(b) Perturbation ζ_2

(c) Perturbation ζ_3

(d) Perturbation ζ_4

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



(a) Perturbation ζ_1

(b) Perturbation ζ_2

(c) Perturbation ζ_3

(d) Perturbation ζ_4

Figure 5.5: κ-test 2, $\log_{10}|\kappa - 1|$ for $\nabla\mathcal{J}$ with 4 different perturbations

5.3 Optimization Results

The results of the κ-tests provide us with the confidence in the accuracy of the gradient evaluation, so we can now obtain reliable optimization results. In this section we will study solutions of the following three optimization problems, as indicated in Table 5.2.

The second column indicates the heat sources distribution applied in each case,

	q	$\bar{u}$	α	l	L_0	$\mathcal{C}^{(0)}$	u_0
CASE 1	q_1	$\bar{u}_1$	0, 1, 10, 10^2, 10^3	0.2	4	$\mathcal{C}_1^{(0)}$	$u_0 = 10$
CASE 2	q_2	$\bar{u}_2$	0, 10, 10^2, 10^3, 10^4	0.2	2.3	$\mathcal{C}_2^{(0)}$	$u_0(s) = 20s$
CASE 3	q_2	$\bar{u}_3$	0, 1, 10, 10^2, 10^3	0.2	2.3	$\mathcal{C}_1^{(0)}$	$u_0 = 10$, $u_0 = 10 + 6s$, $u_0 = 10 + 9s$, $u_0 = 4 + 6s$, $u_0 = 1 + 9s$

Table 5.2: Parameters of the optimization problems

where q_1 is given by

$$q_1(x, y) = 50 - 30\cos\left(\frac{3\pi}{2}x\right)\cos\left(\frac{3\pi}{2}y\right), \quad (x, y) \in \Omega, \tag{5.4}$$

see Figure 5.6(a), and q_2 represents the data obtained from the temperature distribution determined experimentally in an actual battery cell [23] (cf. Figure 5.7(a)).

The third column are the desired temperature distributions $\bar{u}$ defined for $(x, y) \in \Omega$ as:

$$\begin{aligned}
&\bar{u}_1(x, y) = 20 - \sin(0.75x - 1.6)\sin(1.25y - 2.5),\\
&\bar{u}_2(x, y) = 40 + 20\sin\left(\frac{x\cos(\frac{\pi}{5}) - y\sin(\frac{\pi}{5})}{2} - 1.6\right)\sin\left(2y\cos\left(\frac{\pi}{5}\right) + 2x\sin\left(\frac{\pi}{5}\right) + 1.5\right),\\
&\bar{u}_3(x, y) = 30.
\end{aligned} \tag{5.5}$$

The plots of $\bar{u}_1$ and $\bar{u}_2$ are provided, respectively, in Figure 5.6(b) and Figure 5.7(b). The target temperature distribution $\bar{u}_3$ represents the constant temperature in the

domain Ω. The 4th column α is the weight for the length constraint (cf. (3.42)) and the fifth column is the Sobolev coefficient l (cf. Section 3.3.3). The next column is the total length L_0 for the cases where the length constraint enforced. The 7th column represents the contours we use as initial guesses for the optimization algorithm in each case:

$$\begin{aligned} \mathcal{C}_1^{(0)} &: x(t) = t, \\ &\quad y(t) = 0.78, \quad t \in [-1, 1] \\ \mathcal{C}_2^{(0)} &: x(t) = -0.78, \\ &\quad y(t) = t, \quad t \in [-1, 1]. \end{aligned} \tag{5.6}$$

The last column provides us with the reference temperature of the contour, which could be a constant or changing along the arc length. In all tests we consider Ω to be the domain of interest over which the cost functional is defined and $N = 30$, $M = 100$ represent the resolution for the domain and the number of the points used to discretize the contour, respectively.

In Figures 5.6—5.8 we summarize the optimization results obtained in each case. In all plots representing the initial and optimal temperature distribution the contour that generates the corresponding distribution is indicated with a thick solid curve. Also in all cases we use the thick solid line to highlight the various data related to the reference case $\alpha = 0$ (i.e., when no length constraint is present), such as the cost functional, the contour length evolution and the locally optimal contours.

Below we briefly describe the results obtained in the different cases.

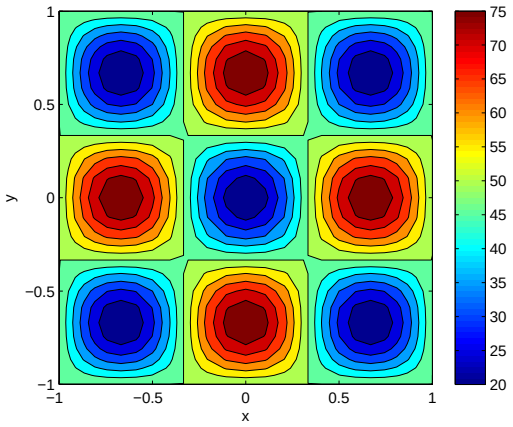
CASE 1: The target temperature field used in this case varies very slowly and stays close to a constant distribution (cf. Figure 5.6(b)). In the case when no length constraint was enforced (thick solid line in Fugure 5.6(d)) we observe a significant drop of the cost functional value from $O(10^4)$ to $O(10)$. The length of the curve evolves from an initial value of $L = 2$ to $L = 5.5$ (cf. 5.6(e)). When we introduce the

length constraint and increase α from 1 to 10^3 by factor of 10, we observe that the minimum value of the cost functional increases as α does, while the total length of the contour converges to the desired value which is set to be $L_0 = 4$. Thick dash-dotted line corresponds to $\alpha = 1$, data for $\alpha = 10$ is represented by thin dash-dotted line, thick dashed line denotes $\alpha = 100$ case and thin dashed line is used for $\alpha = 1000$ case. For the optimal temperature distribution when different α applied see Figure 5.6(g)—5.6(k).

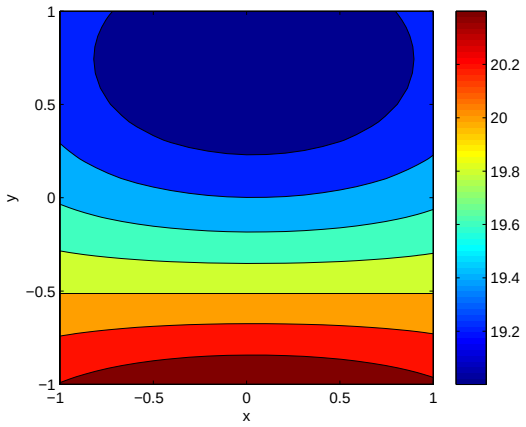
CASE 2: The heat sources in this case are represented by a typical distribution in an actual battery. The target temperature field is shown in Figure 5.7(b). The reference temperature u_0 is changing along the length of the contour. In the case when no length constraint was enforced (thick solid line in Fugure 5.7(d)) we observe a significant drop of the cost functional. The achieved minimum of the cost functional increases as value of the parameter α goes up, while the total length of the contour converges to the desired value of $L_0 = 2.3$. Thick dash-dotted line corresponds to $\alpha = 10$, data for $\alpha = 10^2$ is represented by thin dash-dotted line, thick dashed line denotes $\alpha = 10^3$ case and thin dashed line is used for $\alpha = 10^4$ case. The optimal temperature distributions corresponding to the locally optimal curves obtained with different weights of the length constraint are displayed in Figure 5.7(g)—5.7(k).

CASE 3: In this case we also use the heat sources distribution q_2, but with the constant target temperature distribution and $\mathcal{C}_1^{(0)}$ as initial guess. The data concerning this case is collected in Figure 5.8. First, using the contour shown in Figure 5.8(b) as the initial guess, we solve our optimization problem (2.4) assuming $u_0 = \text{const}$ and $\alpha = 0$ (solid line in Figures 5.8(c) and 5.8(e)). Then, using the obtained optimal shape as the initial guess, we solve problem (2.4) again, but now we allow u_0 to vary with the arc length s. We model changes in the inflow/outflow temperature of the coolant liquid by decreasing u_{in} or increasing

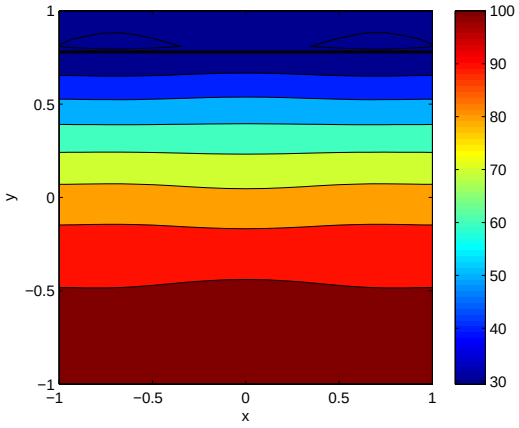
u_{out} (cf. (2.2)), which corresponds to, respectively, decreasing thickness of the dash-dotted and dashed contours in Figure 5.8(e). In Figure 5.8(c) we observe that in the initial optimization the cost functional drops by over three orders of magnitude during less than 10 iterations. Next, we consider the case with $u_{in} = 10$ and $u_{out} = 19$, and solve the optimization problem with the length constraint $L_0 = 2.3$ and increasing values of α. The resulting locally optimal contours are shown in Figure 5.8(f) and Figure 5.8(d) presents the evolution of the contour length $L(\mathcal{C}^{(n)})$ with iterations for different values of α (thinner lines correspond to increasing α). As expected, for increasing values of α, the contour length approaches the prescribed value L_0 while the contours themselves become less deformed. The temperature fields $u(x, y)$ obtained in the cases with $u_0 = 10$, $u_{in} = 1$ and $u_{out} = 10$, and $u_{in} = 10$ and $u_{out} = 19$, and without the length constraint are shown in Figures 5.8(g)—5.8(i). The last case with the length constraint and $\alpha = 1000$ is shown in Figure 5.8(j). We see that, with the exception of the case in which the inflow temperature u_{in} is quite low, the optimal contour shapes tend to weave around the two hot spots in the heat distribution (Figure 5.8(a)).



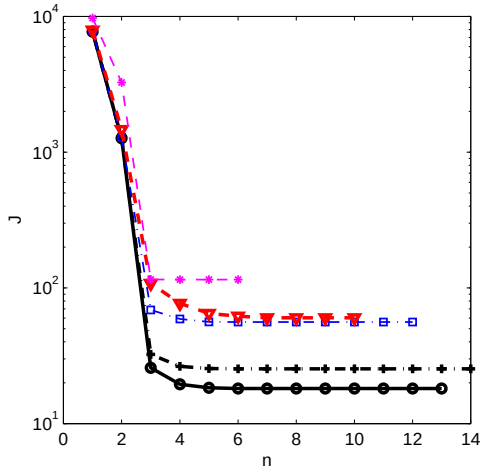
(a) Distribution of heat sources q_1



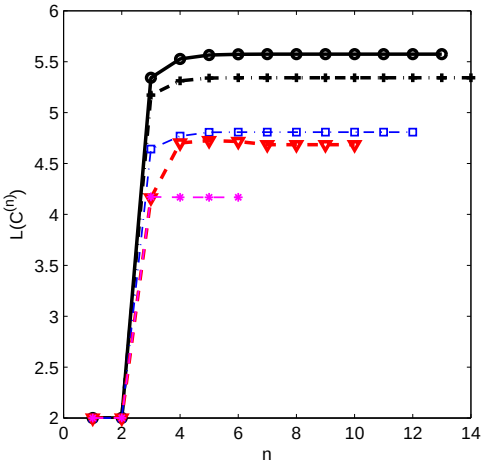
(b) Target temperature field $\bar{u}_1$



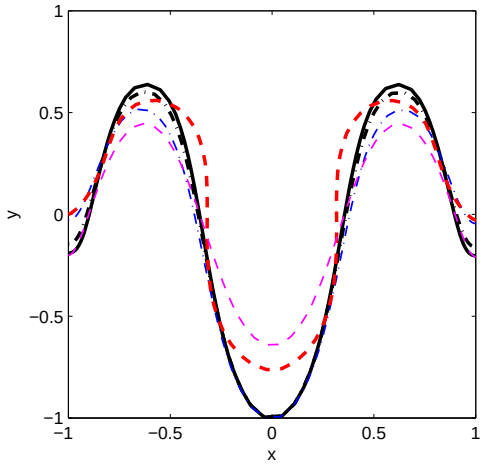
(c) Initial temperature field



(d) Evolution of cost functional $\mathcal{J}(\mathcal{C}^{(n)})$

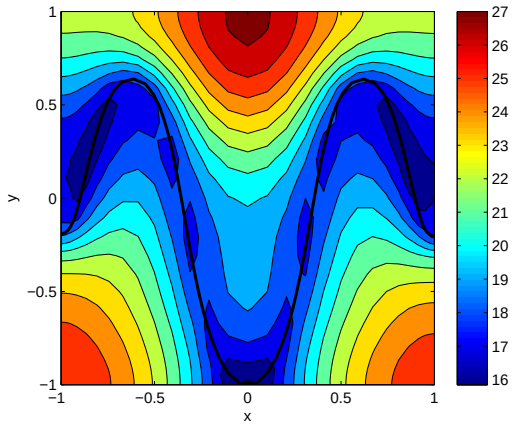


(e) Evolution of contour length $L(\mathcal{C}^{(n)})$

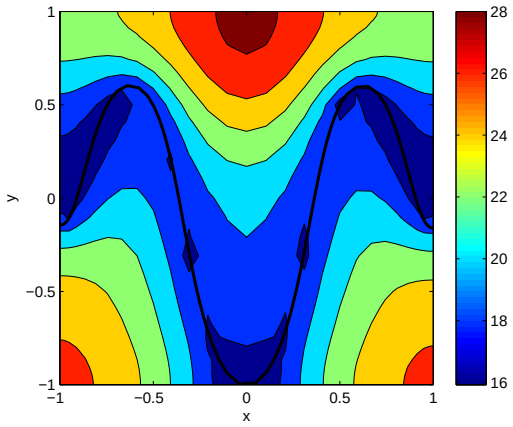


(f) Optimal contours $\widehat{\mathcal{C}}$

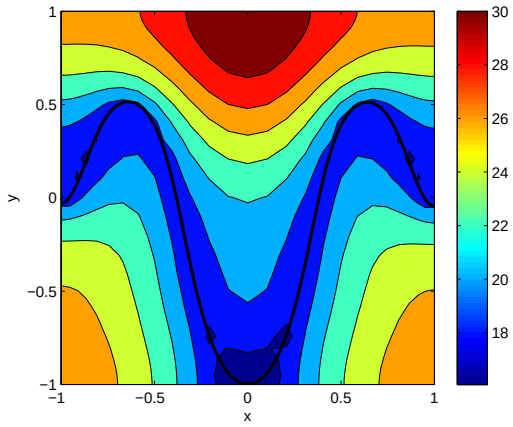
Figure 5.6: Optimization results for CASE 1



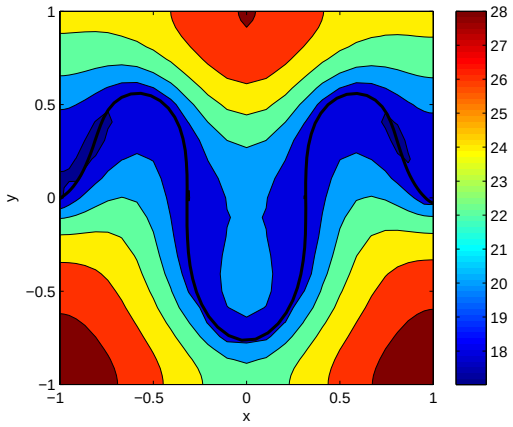
(g) Optimal temperature distribution, $\alpha = 0$



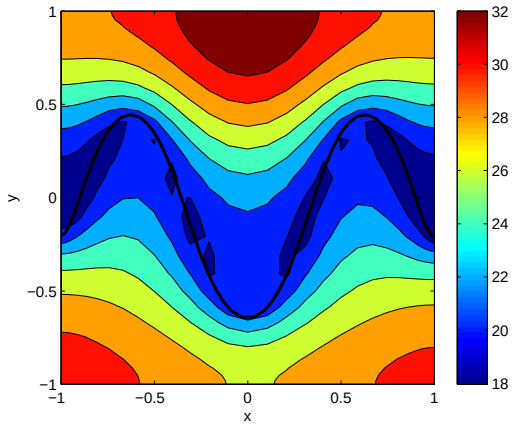
(h) Optimal temperature distribution, $\alpha = 1$



(i) Optimal temperature distribution, $\alpha = 10$

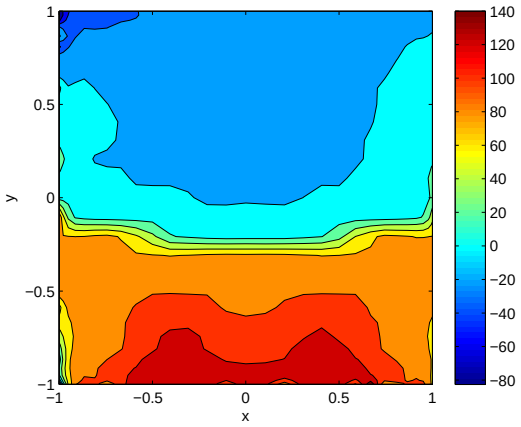


(j) Optimal temperature distribution, $\alpha = 100$

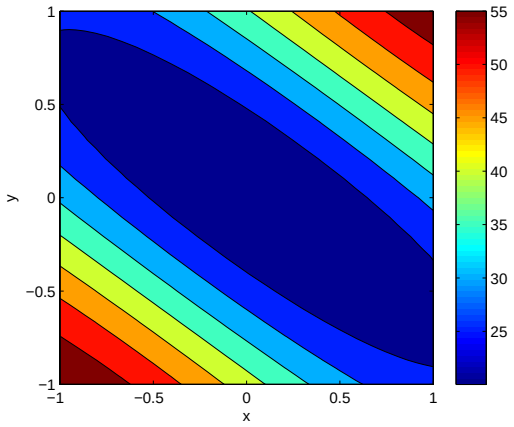


(k) Optimal temperature distribution, $\alpha = 1000$

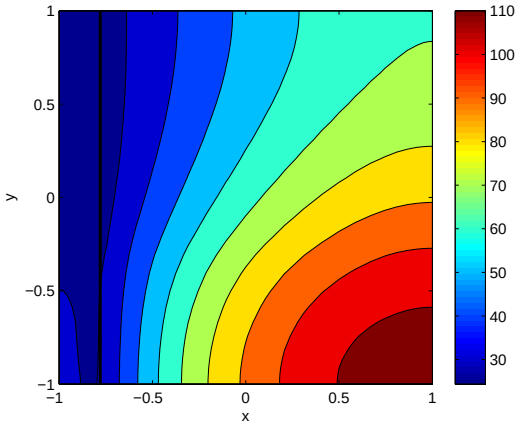
Figure 5.6: (continued) Optimization results for CASE 1



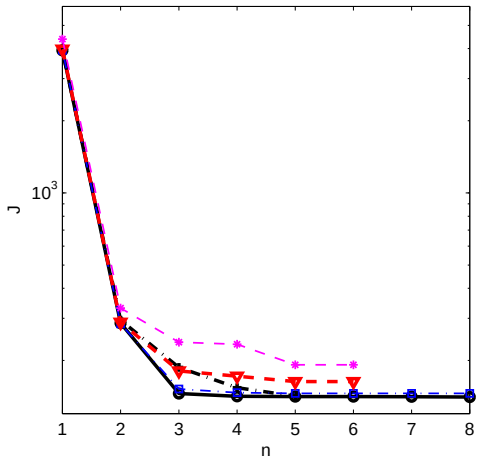
(a) Distribution of heat sources q_2



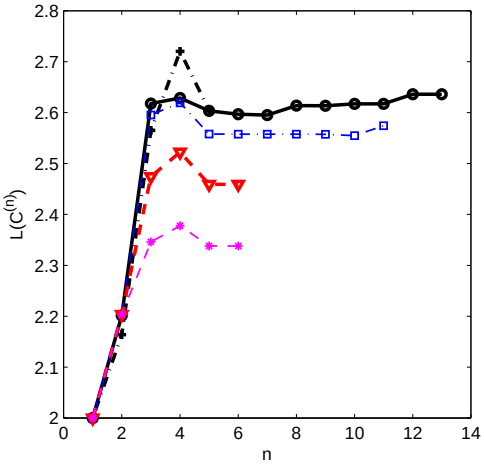
(b) Target temperature field $\bar{u}_2$



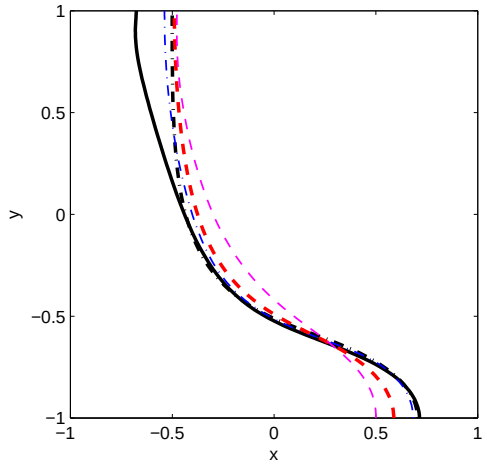
(c) Initial temperature field



(d) Evolution of cost functional $\mathcal{J}(\mathcal{C}^{(n)})$

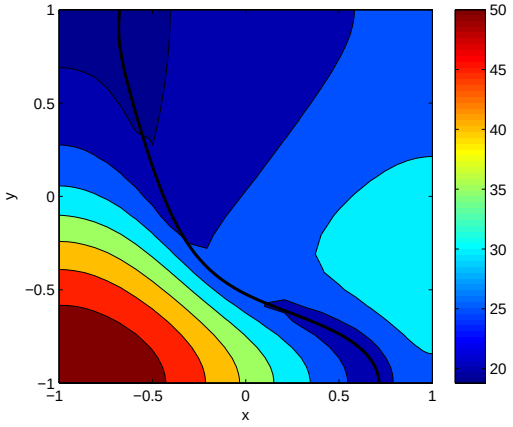


(e) Evolution of contour length $L(\mathcal{C}^{(n)})$

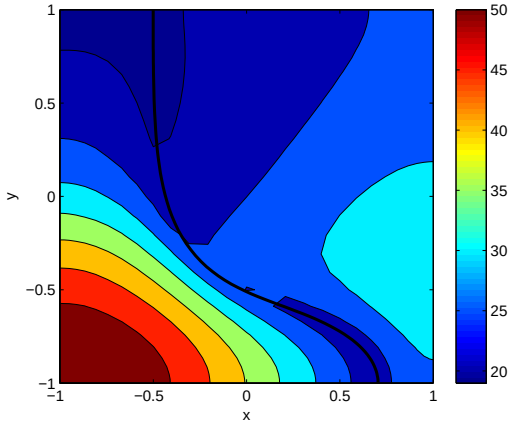


(f) Optimal contours $\widehat{\mathcal{C}}$

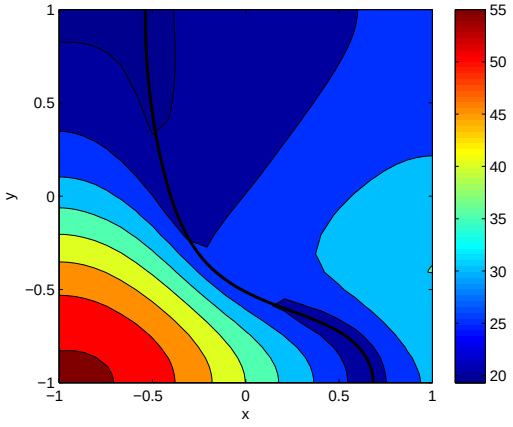
Figure 5.7: Optimization results for CASE 2



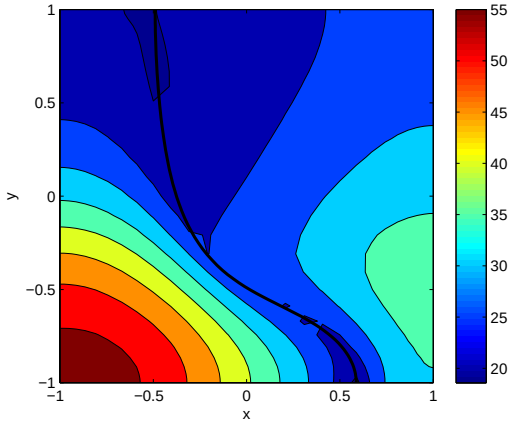
(g) Optimal temperature distribution, $\alpha = 0$



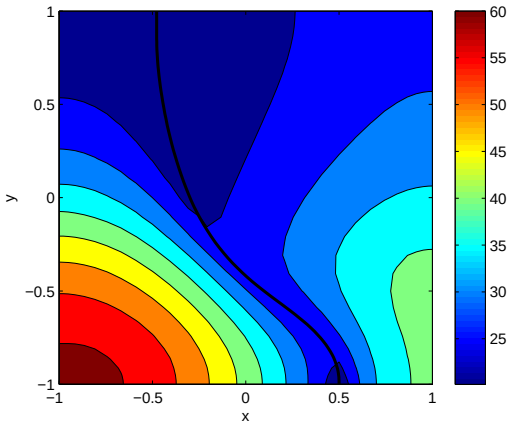
(h) Optimal temperature distribution, $\alpha = 10$



(i) Optimal temperature distribution, $\alpha = 100$

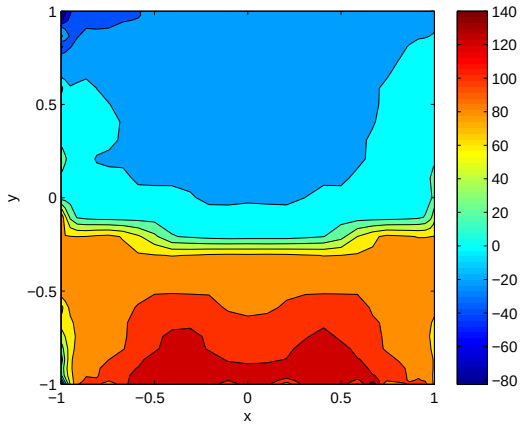


(j) Optimal temperature distribution, $\alpha = 10^3$

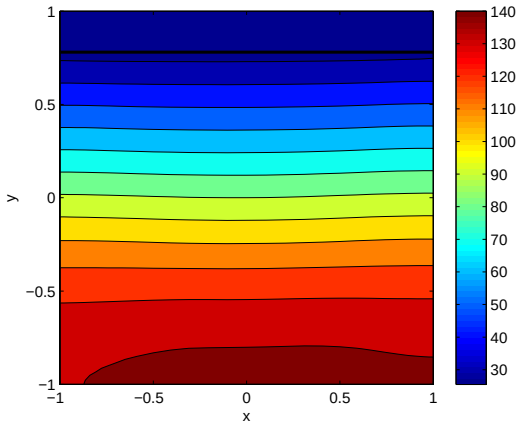


(k) Optimal temperature distribution, $\alpha = 10^4$

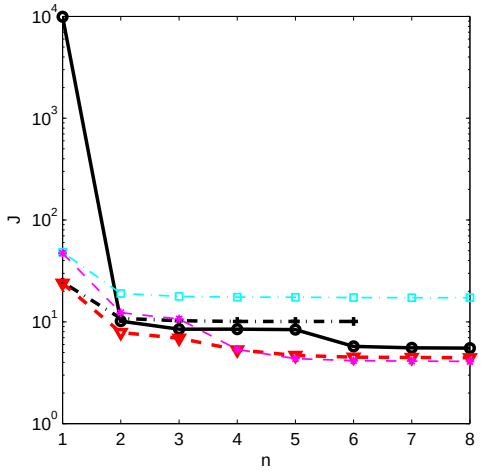
Figure 5.7: (continued) Optimization results for CASE 2



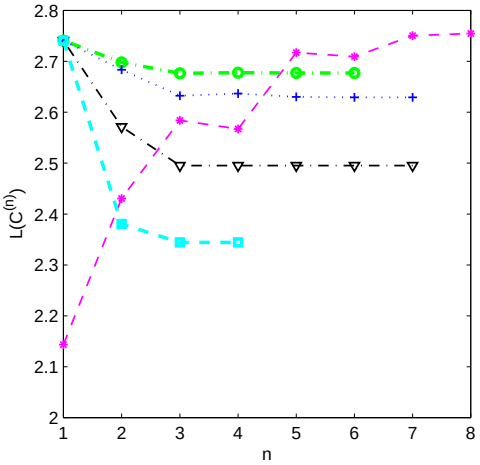
(a) Distribution of heat sources q_2



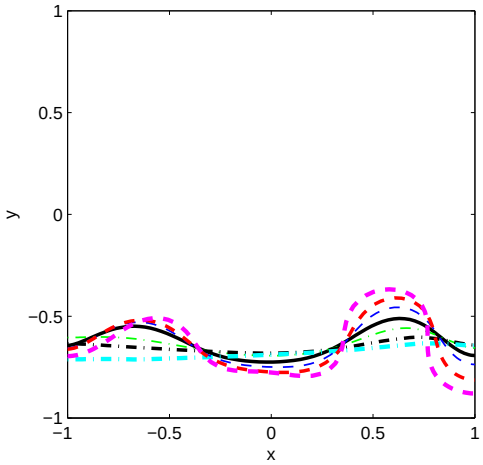
(b) Initial temperature field



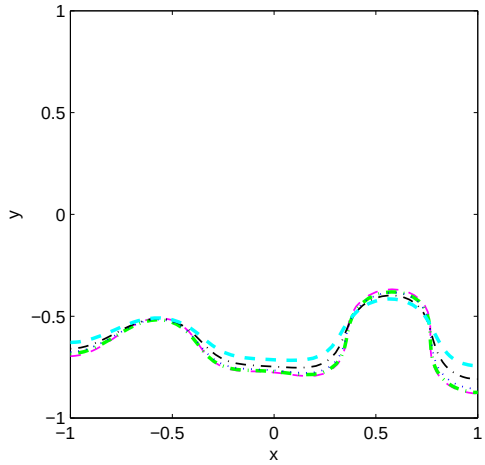
(c) Evolution of cost functional $\mathcal{J}(\mathcal{C}^{(n)})$



(d) Evolution of contour length $L(\mathcal{C}^{(n)})$

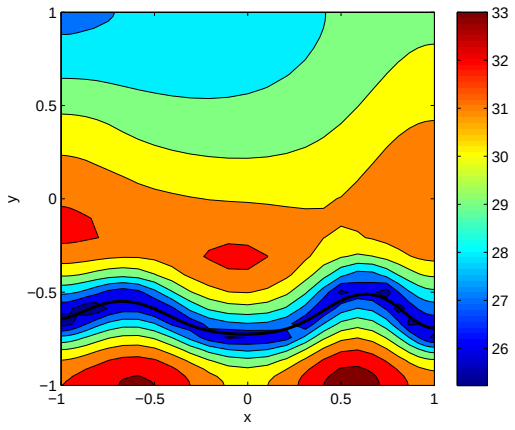


(e) Optimal contours $\widehat{\mathcal{C}}$ obtained without the length constraint

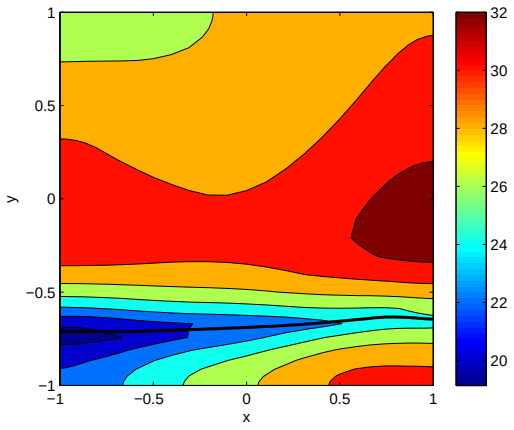


(f) Optimal contours $\widehat{\mathcal{C}}$ obtained with the length constraint

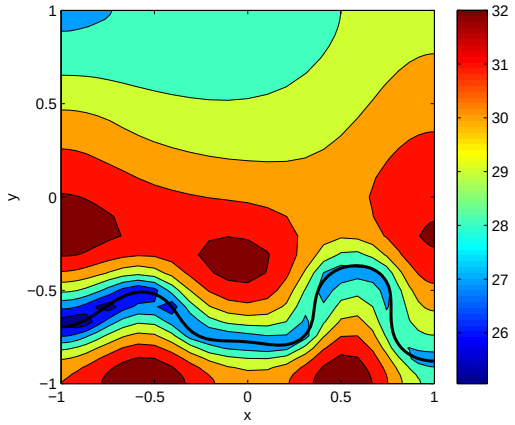
Figure 5.8: Optimization results for CASE 3



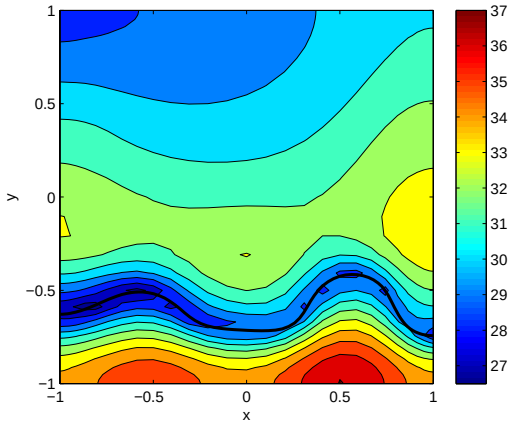
(g) Optimal temperature distribution, $u_0 = 10$, $\alpha = 0$



(h) Optimal temperature distribution, $u_{in} = 1$, $u_{out} = 10$, $\alpha = 0$



(i) Optimal temperature distribution, $u_{in} = 10$, $u_{out} = 19$, $\alpha = 0$



(j) Optimal temperature distribution, $u_{in} = 10$, $u_{out} = 19$, $\alpha = 1000$

Figure 5.8: (continued) Optimization results for CASE 3

Chapter 6

Conclusion and Future Work

This work is motivated by the problem of optimal design of the liquid coolant elements used in modern battery systems. We employed a modified version of the two-dimensional steady-state heat conduction model introduced in [1] to describe the heat transfer process in the battery pack. The battery pack was modelled by a two-dimensional isolated square domain with some prescribed heat sources representing the battery heating. The coolant channel was simplified to a one-dimensional open coil with end points attached to the domain boundary and the reference temperature varying linearly with the length corresponding to the coolant liquid heating up as it absorbs heat. The problem is to find the shape of the cooling channel such that the temperature in some given region is close to a target temperature distribution. The difference between the actual and desired temperature over a certain region of interest was measured in the least square sense by a suitably chosen cost functional. The problem was formulated as a PDE-constrained optimization and a necessary optimality condition based on the shape calculus was introduced. The problem was solved numerically using a combination of a spectral method and a boundary-integral approach. The locally optimal shape of the coil was found using the conjugate gradient method based on the gradients of the cost functional with respect to the shape of the

contour. The shape gradient of our functional is obtained using adjoint analysis and shape-differential calculus. The method we introduced to numerically evaluate the shape gradients based on a boundary-integral approach is the key novel contribution of this work. In particular, two overlapping discretizations of the contour were used to solve a singular BIE on an open contour with the spectral accuracy. Then, the validation of the boundary integral method was performed, whereas the consistency of our cost functional gradient was confirmed by κ-test. Finally, the optimization results based on different settings were presented.

Based on the validation results obtained in Section 5.1 we could see that our boundary integral technique works well for the solution of the Fredholm integral equation of the second kind with a weakly singular kernel. By increasing the number of points used for the contour discretization we can obtain a highly accurate result.

Validation based on the κ-test also provided us with confidence in our adjoint-based calculation of the shape gradient. Since the resolution can be a crucial factor affecting the accuracy of the gradient, we believe that by using sufficiently fine grids an accurate shape gradient can be obtained.

The optimal results in Section 5.3 demonstrate that when different combinations of heat sources q and desired temperature profiles $\bar{u}$ are considered, the optimal shape of the contour is generally not intuitive. Moreover, the optimal shapes $\widehat{\mathcal{C}}$ that we obtained lead to a reasonably small difference between the desired and actual temperatures distributions in a given domain. The presence of the length constraint results in limited contour variation and thus the optimal result is not that promising compared with the unconstrained cases. We note that the proposed approach may not be well adapted to handle problems with large values of L_0, since with increasing L_0 increases the possibility of self-intersecting due to a long contour packed into a finite domain.

Several alternative formulations of the shape gradients defined for a very simple

model are discussed in Appendix B including a PDE-based approach, method involving shape-differentiation at the PDEs level with a BIE-based derivation of the adjoint system and a BIE-based approach. For future improvements one may consider alternative gradient derivation techniques applied to more complicated problems, such as the problem discussed in this thesis. It is an interesting open problem to compare the accuracy of the shape gradients evaluated using these different approaches. In addition, the actual fluid flow in the cooling channel can be modelled more accurately and the possibilities for topological optimization of the cooling element will be considered.

Appendix A

Shape-Derivative of the Arc-Length Coordinate

The sensitivity analysis plays an important role in the construction of the adjoint system and, as a consequence, in the representation of cost functional gradient. In this appendix we discuss the derivation of an explicit formula for the shape derivative (i.e., sensitivity) u_0' of the reference temperature of the contour $\mathcal{C}$ which enters perturbation PDE (3.20). As $u_0(s)$ varies along the contour, i.e., depends on the arc length coordinate s, the shape derivative s' need to be addressed as well.

First of all, we need to define u_0'. As follows from (2.2), the shape derivative of u_0 is given by:

$$u_0'(s;\zeta) = \frac{(u_{out} - u_{in})(s'L - sL')}{L^2}. \tag{A.1}$$

We thus need to compute the shape derivatives of the total length L and the arc length s, respectively. By the definition of the length of the curve, we immediately obtain that:

$$L'(\mathcal{C};\zeta) = \left(\int_{\mathcal{C}} ds \right)' = \int_{\mathcal{C}} \kappa\zeta \, ds. \tag{A.2}$$

To proceed with the shape differentiation of s, we introduce the following parametrization of the contour $\mathcal{C}$:

$$\mathbf{x}(\xi) = [x(\xi),\, y(\xi)] \in \mathcal{C},$$

where the parameter $\xi \in [0, 1]$. Then, the arc length s can be expressed as

$$s(\mathbf{x}(\xi)) = \int_0^\xi \frac{ds}{d\xi}\, d\xi' = \int_0^\xi \left| \frac{d\mathbf{x}}{d\xi} \right| d\xi' = \int_0^\xi \sqrt{x'(\xi')^2 + y'(\xi')^2}\, d\xi'. \tag{A.3}$$

Now we can shape-differentiate (A.3) as follows:

$$s'(\mathbf{x}(\xi); \zeta) = \int_0^\xi \frac{d}{d\epsilon} \left| \frac{d}{d\xi}(\mathbf{x} + \epsilon\, \zeta\, \mathbf{n}) \right| \Bigg|_{\epsilon=0} d\xi'. \tag{A.4}$$

Let us consider the integrand in more detail. By the chain rule the term $\frac{d}{d\xi}(\mathbf{x}+\epsilon\, \zeta\, \mathbf{n})$ can be written as

$$\frac{d}{d\xi}(\mathbf{x} + \epsilon\, \zeta\, \mathbf{n}) = \frac{d\mathbf{x}}{d\xi} + \epsilon \left[\frac{d\zeta}{d\xi}\, \mathbf{n} + \zeta\, \frac{d\mathbf{n}}{d\xi} \right] = \frac{d\mathbf{x}}{d\xi} + \epsilon \left[\frac{d\zeta}{d\xi}\, \mathbf{n} + \kappa\zeta\, \frac{ds}{d\xi}\, \mathbf{t} \right], \tag{A.5}$$

where $\mathbf{t}$ denotes the tangent vector and the fact that $\frac{d\mathbf{n}}{d\xi} = \frac{ds}{d\xi}\frac{d\mathbf{n}}{ds}$ and Frenet's formula $\frac{d\mathbf{n}}{ds} = \kappa\, \mathbf{t}$ were applied. Therefore, the integrand is given by

$$\begin{aligned} \frac{d}{d\epsilon} \left| \frac{d}{d\xi}(\mathbf{x} + \epsilon\, \zeta\, \mathbf{n}) \right| \Bigg|_{\epsilon=0} = & \frac{d}{d\epsilon} \sqrt{\left(\frac{d\mathbf{x}}{d\xi} + \epsilon \Big[\frac{d\zeta}{d\xi}\, \mathbf{n} + \kappa\, \zeta \frac{ds}{d\xi}\, \mathbf{t} \Big] \right)^2} \Bigg|_{\epsilon=0} = \\ & \frac{1}{\left| \frac{d\mathbf{x}}{d\xi} \right|} \frac{d\mathbf{x}}{d\xi} \cdot \left(\frac{d\zeta}{d\xi}\, \mathbf{n} + \kappa\, \zeta\, \frac{ds}{d\xi}\, \mathbf{t} \right). \end{aligned} \tag{A.6}$$

We notice that $\frac{1}{\left| \frac{d\mathbf{x}}{d\xi} \right|} \frac{d\mathbf{x}}{d\xi} = \mathbf{t}$ and after we multiply it by the first term in brackets in (A.6), the product vanishes due to $\mathbf{t} \cdot \mathbf{n} = 0$ leaving us with the following formula for

the integrand in (A.4)

$$\frac{d}{d\epsilon}\left|\frac{d}{d\xi}(\mathbf{x}+\epsilon\,\zeta\,\mathbf{n})\right|\Big|_{\epsilon=0}=\kappa\,\zeta\,\frac{ds}{d\xi}. \tag{A.7}$$

Now, returning to the shape-derivative of the arc-length coordinate s (A.4), we have:

$$s'(\mathbf{x}(\xi);\zeta)=\int_0^{\xi}\kappa\,\zeta\,\frac{ds}{d\xi}\,d\xi'=\int_0^{s}\kappa\,\zeta\,d\tilde{s}. \tag{A.8}$$

Finally, we conclude with the expression for u_0' (cf. (A.1)):

$$\begin{aligned}u_0'(s;\zeta)=&\frac{(u_{out}-u_{in})}{L}\left[\int_0^{s}\kappa\,\zeta\,d\tilde{s}-\frac{s}{L}\int_0^{L}\kappa\,\zeta\,d\tilde{s}\right]=\\&\frac{(u_{out}-u_{in})}{L}\int_0^{L}\left[H(s-\tilde{s})-\frac{s}{L}\right]\kappa\,\zeta\,d\tilde{s},\end{aligned} \tag{A.9}$$

where $H(s)$ is the Heaviside function.

Appendix B

Alternative Formulation of the Shape Gradients

In this thesis we used a standard sensitivity analysis and a suitably defined adjoint system to derive the gradient. Then, to approximate the gradient numerically, we solved the direct and adjoint systems using their boundary integral formulations. The question we are asking here is how the formula for the gradient would change if from the beginning we worked with the BIE representation of the direct or perturbed systems instead of their PDE forms. It may turn out that such an alternative formulation would be easier to derive and/or more convenient for numerical approximation.

According to the Riesz representation theorem, there is a unique gradient $\nabla \mathcal{J}$, such that

$$\mathcal{J}'(\Gamma;\zeta) = \langle \nabla \mathcal{J}, \zeta \rangle_{\mathcal{X}} \quad \forall \zeta \in \mathcal{X}, \tag{B.1}$$

so the gradients derived by different approaches must be equivalent.

In this appendix we compare three different techniques for deriving the gradient of the cost functional as summarized schematically in Figure B.1.

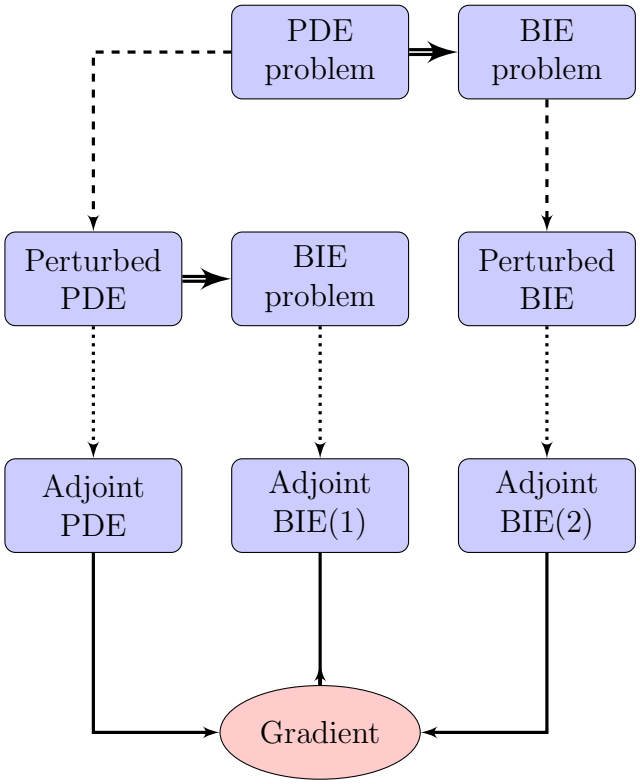


Figure B.1: Schematic of the gradient derivation: (double lines) - BIE transformation, (dashed lines) - shape differentiation, (dotted lines) - adjoint identity, (solid lines) - formulation of the gradient

To focus attention on the key issues, we consider an optimization problem for the following very simple system

$$-\Delta u = 0 \quad \text{in } \Omega, \tag{B.2a}$$

$$u = g \quad \text{on } \Gamma = \partial\Omega, \tag{B.2b}$$

where $\Omega \subset \mathbb{R}^2$ is a domain with the boundary Γ and g is some known function representing the Dirichlet boundary data.

The cost functional we want to minimize is

$$\mathcal{J}(\Gamma) = \frac{1}{2}\int_A \omega(\mathbf{x})(u-\bar{u})^2\, d\mathbf{x}, \tag{B.3}$$

where $\bar{u}$ is our target value of u (for example, a target temperature distribution), $\omega(\mathbf{x})$ is some given weight function defined in A and $A \subset \Omega$ is the region where we want to match our target field $\bar{u}$.

Recall that using the Riesz representation theorem, in Section 3.2 we derived an equivalent formulation of the optimality condition $\nabla\mathcal{J} = 0$, where the gradient is derived from identity (3.13), which we repeat here for completeness:

$$\mathcal{J}'(\Gamma;\zeta) = \langle \nabla\mathcal{J}, \zeta\rangle_{L_2} \quad \forall \zeta \in L_2. \tag{B.4}$$

The shape derivative of our cost functional (B.3) by Lemma 1 is given by

$$\mathcal{J}'(\Gamma;\zeta) = \int_A \omega(\mathbf{x})(u-\bar{u})u'\, d\mathbf{x} \tag{B.5}$$

and, since it is not in the desired Riesz form yet, we need to define a suitable adjoint system to extract the gradient from this expression.

Thus, in the following sections we derive expression for the cost functional gra-

dient of our model optimization problem applying three different techniques shown schematically in Figure B.1. Each column in this schematic represents a different approach and the different line types illustrate different operations, namely the double lines represent the BIE transformations of PDE problems, the dashed lines indicate the shape-differentiation operations, the dotted lines introduce the adjoint variables and the solid lines stand for the gradient formulations. The main difference between these approaches is the stage at which the problem is transformed to a boundary integral formulation and resulting in different types of equations which need to be solved to obtain the gradient in each approach.

B.1 PDE-based Shape-Differentiation and Derivation of the Adjoint System

In this section to obtain the gradient we work solely with partial differential equations, which is the most common approach to the gradient derivation. This technique (cf. left column in Figure B.1) relies on the PDE formulation of the direct problem, so the perturbation and adjoint systems are derived in the PDE form as well.

First, we derive our perturbation PDE system

$$-\Delta u' = 0 \quad \text{in } \Omega, \tag{B.6a}$$

$$u' + \frac{\partial u}{\partial n}\,\zeta = 0 \quad \text{on } \Gamma, \tag{B.6b}$$

where ζ is an arbitrary perturbation, $\mathbf{n}$ is an outer unit normal vector and u' is the shape derivative of u as before. This system follows directly from the shape calculus applied to the statement of the direct problem (B.2) and the fact that the boundary data g is considered to be independent on the shape of the boundary Γ.

Next, to derive the gradient formulation, we introduce an adjoint variable u^* defined in Ω and, after multiplying the governing equation (B.6a) by u^* and integrating over Ω, we get the following equality:

$$\int_\Omega \Delta u' \, u^* \, d\mathbf{x} = 0.$$

Using Green's identity, this expression is transformed to:

$$0 = \int_\Omega u' \, \Delta u^* \, d\mathbf{x} + \int_\Gamma \left(u^* \frac{\partial u'}{\partial n} - u' \frac{\partial u^*}{\partial n} \right) ds.$$

If we impose an adjoint equation

$$-\Delta u^* = (\omega(u - \bar{u}))\chi_A,$$

the first term in our identity can be replaced by the shape derivative of the cost functional (B.5), so that now we have:

$$\mathcal{J}'(\Gamma; \zeta) = -\int_\Gamma \left(u^* \frac{\partial u'}{\partial n} - u' \frac{\partial u^*}{\partial n} \right) ds.$$

After we express u' from (B.6b) and substitute it into the previous relation, we get

$$\mathcal{J}'(\Gamma; \zeta) = -\int_\Gamma u^* \frac{\partial u'}{\partial n} \, ds - \int_\Gamma \frac{\partial u}{\partial n} \frac{\partial u^*}{\partial n} \zeta \, ds,$$

where the second term in the Riesz form already. Finally, introducing the boundary condition

$$u^* = 0 \quad \text{on } \Gamma$$

we obtain

$$\mathcal{J}'(\Gamma; \zeta) = -\int_\Gamma \frac{\partial u}{\partial n} \frac{\partial u^*}{\partial n} \zeta \, ds.$$

Thus, using Riesz identity (B.1) the cost functional gradient is obtained in the form

$$\nabla \mathcal{J} = -\frac{\partial u}{\partial n}\frac{\partial u^*}{\partial n}, \tag{B.7}$$

where u^* is a solution of the following adjoint PDE:

$$-\Delta u^* = (\omega(u - \bar{u}))\chi_A \quad \text{in } \Omega, \tag{B.8a}$$

$$u^* = 0 \qquad\qquad\qquad\quad \text{on } \Gamma. \tag{B.8b}$$

B.2 PDE-based Shape-Differentiation and BIE-based Derivation of the Adjoint System

In this method (cf. central column in Figure B.1) we use the PDE formulation of the direct problem, then shape-differentiate it and transform the corresponding perturbed system into an equivalent BIE which is used for deriving a BIE-based formulation of the adjoint system.

As a starting point for this approach, we follow the method as above. The modifications begin after the perturbation system (B.6) is derived. Following the idea introduced in Section 4.3, the solution of the governing equation (B.6a) is given by a single-layer potential:

$$u'(\mathbf{x}) = -\frac{1}{2\pi}\int_\Gamma \ln|\mathbf{x} - \mathbf{y}|\rho'(\mathbf{y})\, ds_y, \quad \mathbf{x} \in \Omega,$$

where $\rho'(\mathbf{y})$, $\mathbf{y} \in \Gamma$, denotes the corresponding density function.

Thus, we obtain the following BIE as an equivalent formulation of the perturbation

system (B.6):

$$-\frac{1}{2\pi}\int_{\Gamma}\ln|\mathbf{x}-\mathbf{y}|\rho'(\mathbf{y})\,ds_y=-\frac{\partial u}{\partial n}\,\zeta(\mathbf{x}),\quad \mathbf{x}\in\Gamma, \tag{B.9}$$

which corresponds to (B.6b) and in which the integral is to be interpreted as an improper one. Introducing an adjoint variable $\rho^*:\Gamma\longrightarrow\mathbb{R}$ to facilitate the gradient derivation, multiplying (B.9) by it and integrating over the boundary Γ, we get:

$$-\frac{1}{2\pi}\int_{\Gamma}\rho^*(\mathbf{x})\Big[\int_{\Gamma}\ln|\mathbf{x}-\mathbf{y}|\rho'(\mathbf{y})\,ds_y\Big]ds_x=-\int_{\Gamma}\rho^*(\mathbf{x})\,\frac{\partial u}{\partial n}\,\zeta\,ds_x. \tag{B.10}$$

Let us stop here for a moment and give an expression for the cost functional derivative (B.5) in terms of the single-layer potential representation of u':

$$\mathcal{J}'(\Gamma;\zeta)=-\frac{1}{2\pi}\int_{A}\omega(u-\bar{u})\Big[\int_{\Gamma}\ln|\mathbf{x}-\mathbf{y}|\rho'(\mathbf{y})\,ds_y\Big]\,d\mathbf{x},$$

which after exchanging the order of integration becomes:

$$\mathcal{J}'(\Gamma;\zeta)=-\frac{1}{2\pi}\int_{\Gamma}\rho'(\mathbf{y})\Big[\int_{A}\omega(u-\bar{u})\ln|\mathbf{x}-\mathbf{y}|\,d\mathbf{x}\Big]\,ds_y. \tag{B.11}$$

Next, we rewrite the adjoint identity (B.10), also after changing the integration order:

$$-\frac{1}{2\pi}\int_{\Gamma}\rho'(\mathbf{y})\Big[\int_{\Gamma}\ln|\mathbf{x}-\mathbf{y}|\,\rho^*(\mathbf{x})\,ds_x\Big]\,ds_y=-\int_{\Gamma}\rho^*(\mathbf{x})\,\frac{\partial u}{\partial n}\,\zeta\,ds_x. \tag{B.12}$$

Note that, if we now introduce the following adjoint boundary integral equation

$$\int_{\Gamma}\ln|\mathbf{x}-\mathbf{y}|\,\rho^*(\mathbf{x})\,ds_x=\int_{A}\omega(u-\bar{u})\ln|\mathbf{x}-\mathbf{y}|\,d\mathbf{x}, \tag{B.13}$$

then with the help of identity (B.12), the shape derivative (B.11) of the cost functional

becomes equal to the following Riesz representation

$$\mathcal{J}'(\Gamma;\zeta) = -\int_\Gamma \rho^*(\mathbf{x})\,\frac{\partial u}{\partial n}\,\zeta\,ds_x.$$

Thus, the cost function gradient is given by

$$\nabla\mathcal{J} = -\rho^*(\mathbf{x})\,\frac{\partial u}{\partial n}, \tag{B.14}$$

where $\rho^*(\mathbf{x})$ is the solution of the adjoint boundary integral equation (B.13).

B.3 BIE-based Shape-Differentiation and Derivation of the Adjoint System

In this approach (cf. right column in Figure B.1), we work solely with boundary integral equations introducing BIE formulation of problem (B.2), which we shape-differentiate to obtain the perturbed equation and the corresponding adjoint BIE. In addition, to define our shape derivative properly, we introduce auxillary problem in the complement domain $\mathbb{R}^2 \setminus \Omega$.

We begin by formulating our model problem (B.2) as a BIE by introducing a single-layer potential

$$u(\mathbf{x}) = -\frac{1}{2\pi}\int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\mu(\mathbf{y})\,ds_y, \quad \mathbf{x}\in\Omega, \tag{B.15}$$

where $\mu(\mathbf{y})$, $\mathbf{y}\in\Gamma$, is the corresponding density function.

Thus, we obtain the direct problem in the following BIE form:

$$-\frac{1}{2\pi}\int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\mu(\mathbf{y})\,ds_y = g, \quad \mathbf{x}\in\Gamma, \tag{B.16}$$

which corresponds to (B.2b) with (B.2a) satisfied identically by representation (B.15).

Let us first discuss the shape differentiation of u given by (B.15), since it will play an important role in the derivation of the perturbation equation and will also enter the expression for the shape derivative of cost functional (B.5).

Recall the definition of the material derivative of $u(\mathbf{x})$

$$\dot{u} = u' + \boldsymbol{\nabla} u \cdot \boldsymbol{\mathcal{V}}. \tag{B.17}$$

Note that

$$\boldsymbol{\nabla} u(\mathbf{x}) \cdot \boldsymbol{\mathcal{V}} = -\frac{1}{2\pi} \int_\Gamma \nabla_x \ln|\mathbf{x}-\mathbf{y}| \mu(\mathbf{y}) ds_y \cdot \boldsymbol{\mathcal{V}} = -\frac{1}{2\pi} \int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x} \mu(\mathbf{y}) ds_y \cdot \zeta(\mathbf{x}). \tag{B.18}$$

To compute u', we refer to formula (3.7c) for the differentiation of the curvilinear integrals:

$$\begin{aligned} u'(\mathbf{x}) = &- \frac{1}{2\pi} \int_\Gamma \ln|\mathbf{x}-\mathbf{y}| \mu'(\mathbf{y}) ds_y - \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y} \mu(\mathbf{y})\, \zeta(\mathbf{y})\, ds_y \\ &- \frac{1}{2\pi} \left(\int_\Gamma \ln|\mathbf{x}-\mathbf{y}| \frac{\partial \mu}{\partial n_y} + \kappa \ln|\mathbf{x}-\mathbf{y}| \mu(\mathbf{y}) \right) \zeta(\mathbf{y})\, ds_y, \end{aligned} \tag{B.19}$$

where κ is the signed curvature of the boundary Γ.

Finally, after adding the intermediate results above, we have:

$$\begin{aligned} \dot{u} = &- \frac{1}{2\pi} \int_\Gamma \ln|\mathbf{x}-\mathbf{y}| \mu'(\mathbf{y}) ds_y - \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y} \mu(\mathbf{y})\, \zeta(\mathbf{y})\, ds_y \\ &- \frac{1}{2\pi} \left(\int_\Gamma \ln|\mathbf{x}-\mathbf{y}| \frac{\partial \mu}{\partial n_y} + \kappa \ln|\mathbf{x}-\mathbf{y}| \mu(\mathbf{y}) \right) \zeta(\mathbf{y})\, ds_y \\ &- \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x} \mu(\mathbf{y}) ds_y \cdot \zeta(\mathbf{x}). \end{aligned} \tag{B.20}$$

We observe that formula (B.20) contains the term $\frac{\partial \mu}{\partial n_y}$ which is not defined properly, since the density function $\mu(\mathbf{y})$, $\mathbf{y} \in \Gamma$, is not defined away from the boundary. To give sense to this expression, we need to define one-sided limits of u on the two opposite sides of the contour.

First, we smoothly extend the Laplacian to the boundary to get

$$-\Delta u_-|_\Gamma = 0, \tag{B.21}$$

where $u_- = u(\mathbf{x})$, $\mathbf{x} \in \bar{\Omega}$ is a solution to the interior problem.

Next, we define u_+ as a solution of the following exterior problem:

$$-\Delta u_+ = 0 \quad \text{in } \Omega_+, \tag{B.22a}$$

$$u_+ = g \quad \text{on } \Gamma, \tag{B.22b}$$

where Ω_+ stands for some exterior domain with respect to Ω, e.g., $\Omega_+ = \mathbb{R}^2 \setminus \Omega$.

Finally, we recall the property (4.14) of the density function

$$-\mu(\mathbf{x}) = \frac{\partial u_+(\mathbf{x})}{\partial n} - \frac{\partial u_-(\mathbf{x})}{\partial n}.$$

Thus, we can express its normal derivative as

$$\frac{\partial \mu(\mathbf{x})}{\partial n} = -\frac{\partial^2 u_+(\mathbf{x})}{\partial n^2} + \frac{\partial^2 u_-(\mathbf{x})}{\partial n^2}. \tag{B.23}$$

To deal with the second-order normal derivatives of u_+ and u_-, we use the Laplace-Beltrami operator introduced in Section 3.3.2. In this way we obtain the following

useful equality defined in the local curvilinear coordinates (s, n):

$$\frac{\partial^2}{\partial n^2} = \Delta - \frac{\partial^2}{\partial s^2} - \kappa \frac{\partial}{\partial n}. \tag{B.24}$$

Therefore, using expression (B.24) and problems (B.21), (B.22), the normal derivative of μ (cf. (B.23)) is given by

$$\frac{\partial \mu(\mathbf{x})}{\partial n} = -\left(- \frac{\partial^2 u_+}{\partial s^2} - \kappa \frac{\partial u_+}{\partial n} \right) + \left(- \frac{\partial^2 u_-}{\partial s^2} - \kappa \frac{\partial u_-}{\partial n} \right) = -\kappa \mu(\mathbf{x}), \tag{B.25}$$

where property (4.14) and the fact that the solution u is continuous across the boundary were applied.

Finally, we substitute (B.25) back to formulae (B.19), (B.20) and get simplified expressions for the shape and material derivatives of u, respectively:

$$u' = -\frac{1}{2\pi} \int_\Gamma \ln |\mathbf{x} - \mathbf{y}| \mu'(\mathbf{y}) ds_y - \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln |\mathbf{x} - \mathbf{y}|}{\partial n_y} \mu(\mathbf{y})\, \zeta(\mathbf{y})\, ds_y \tag{B.26}$$

and

$$\begin{aligned} \dot{u} = &- \frac{1}{2\pi} \int_\Gamma \ln |\mathbf{x} - \mathbf{y}| \mu'(\mathbf{y}) ds_y - \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln |\mathbf{x} - \mathbf{y}|}{\partial n_y} \mu(\mathbf{y})\, \zeta(\mathbf{y})\, ds_y \\ &- \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln |\mathbf{x} - \mathbf{y}|}{\partial n_x} \mu(\mathbf{y}) ds_y \cdot \zeta(\mathbf{x}). \end{aligned} \tag{B.27}$$

Based on this result and the fact that g is shape-independent, the following perturbed boundary integral equation is obtained:

$$\begin{aligned} -\frac{1}{2\pi} \int_\Gamma \ln |\mathbf{x} - \mathbf{y}| \mu'(\mathbf{y}) ds_y = \\ \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln |\mathbf{x} - \mathbf{y}|}{\partial n_y} \mu(\mathbf{y})\, \zeta(\mathbf{y})\, ds_y + \frac{1}{2\pi} \int_\Gamma \frac{\partial \ln |\mathbf{x} - \mathbf{y}|}{\partial n_x} \mu(\mathbf{y}) ds_y\, \zeta(\mathbf{x}). \end{aligned} \tag{B.28}$$

Now, to derive an expression for the cost functional gradient, we introduce an adjoint variable $\mu^*(\mathbf{x})$, $\mathbf{x} \in \Gamma$, multiply our perturbed equation (B.28) by μ^* and integrate over Γ:

$$-\frac{1}{2\pi}\int_\Gamma \mu^*(\mathbf{x}) \int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\mu'(\mathbf{y})\, ds_y\, ds_x = \\ \frac{1}{2\pi}\int_\Gamma \mu^*(\mathbf{x}) \int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\mu(\mathbf{y})\,\zeta(\mathbf{y})\, ds_y\, ds_x + \frac{1}{2\pi}\int_\Gamma \mu^*(\mathbf{x}) \int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x}\mu(\mathbf{y}) ds_y\,\zeta(\mathbf{x})\, ds_x.$$

After changing the order of integration, we have

$$\begin{aligned} -\frac{1}{2\pi}\int_\Gamma \mu'(\mathbf{y}) \left[\int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\mu^*(\mathbf{x})\, ds_x\right] ds_y = \\ \frac{1}{2\pi}\int_\Gamma \left[\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\mu^*(\mathbf{x})\, ds_x\right]\mu(\mathbf{y})\,\zeta(\mathbf{y})\, ds_y + \\ \frac{1}{2\pi}\int_\Gamma \left[\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x}\mu(\mathbf{y}) ds_y\right]\mu^*(\mathbf{x})\,\zeta(\mathbf{x})\, ds_x. \end{aligned} \tag{B.29}$$

Next, with the notion of simplified formula (B.26) for the shape derivative u', the shape derivative of the cost functional (B.5) can be written as:

$$\mathcal{J}'(\Gamma;\zeta) = \int_A \omega(\mathbf{x})(u-\bar{u})\left(-\frac{1}{2\pi}\int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\mu'(\mathbf{y})ds_y - \frac{1}{2\pi}\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\mu(\mathbf{y})\,\zeta(\mathbf{y})\, ds_y\right) d\mathbf{x}.$$

After exchanging the order of integration, $\mathcal{J}'$ becomes

$$\begin{aligned} \mathcal{J}'(\Gamma;\zeta) = &-\frac{1}{2\pi}\int_\Gamma \mu'(\mathbf{y})\left[\int_A \omega(\mathbf{x})(u-\bar{u})\ln|\mathbf{x}-\mathbf{y}|\, d\mathbf{x}\right] ds_y \\ &-\frac{1}{2\pi}\int_\Gamma \left[\int_A \omega(\mathbf{x})(u-\bar{u})\frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\, d\mathbf{x}\right]\mu(\mathbf{y})\zeta(\mathbf{y})\, ds_y. \end{aligned} \tag{B.30}$$

We note that the second term is in the Riesz form (cf. (B.1)).

Let us now take a look at the first term in (B.30). Imposing the adjoint equation

$$\int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\mu^*(\mathbf{x})\,ds_x = \int_A \omega(\mathbf{x})(u-\bar{u})\ln|\mathbf{x}-\mathbf{y}|\,d\mathbf{x}, \tag{B.31}$$

and using identity (B.29) we can transform the first term on the RHS in (B.30) to the Riesz representation

$$\begin{aligned}-\frac{1}{2\pi}\int_\Gamma \mu'(\mathbf{y})\left[\int_A \omega(\mathbf{x})(u-\bar{u})\ln|\mathbf{x}-\mathbf{y}|\,d\mathbf{x}\right]ds_y = \\ \frac{1}{2\pi}\int_\Gamma\left[\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\mu^*(\mathbf{x})\,ds_x\right]\mu(\mathbf{y})\,\zeta(\mathbf{y})\,ds_y + \\ \frac{1}{2\pi}\int_\Gamma\left[\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x}\mu(\mathbf{y})ds_y\right]\mu^*(\mathbf{x})\,\zeta(\mathbf{x})\,ds_x.\end{aligned}$$

Moreover, since from (B.31) it follows that

$$\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\mu^*(\mathbf{x})\,ds_x = \int_A \omega(\mathbf{x})(u-\bar{u})\frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_y}\,d\mathbf{x},$$

the expression for $\mathcal{J}'$ can be simplified further and is finally given by

$$\mathcal{J}'(\Gamma;\zeta) = \frac{1}{2\pi}\int_\Gamma\left[\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x}\mu(\mathbf{y})ds_y\right]\mu^*(\mathbf{x})\,\zeta(\mathbf{x})\,ds_x. \tag{B.32}$$

From the Riesz representation theorem, the gradient expression

$$\nabla\mathcal{J} = \frac{\mu^*(\mathbf{x})}{2\pi}\int_\Gamma \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x}\mu(\mathbf{y})ds_y = -\mu^*(\mathbf{x})\frac{\partial u}{\partial n} \tag{B.33}$$

follows immediately, where μ^* solves the adjoint system (B.31).

We remark that in fact the right hand side of (B.33) is not an improper integral,

since according to Theorem 3 and properties (4.16), (4.17)

$$\lim_{\mathbf{y}\to\mathbf{x}} \frac{\partial \ln|\mathbf{x}-\mathbf{y}|}{\partial n_x} = \lim_{\mathbf{y}\to\mathbf{x}} \frac{\langle \mathbf{n}(\mathbf{x}), \mathbf{x}-\mathbf{y}\rangle}{|\mathbf{x}-\mathbf{y}|^2} = \frac{\kappa}{2}, \quad \mathbf{x},\, \mathbf{y} \in \Gamma.$$

B.4 Summary of the Results

In this section we give a brief summary of the expressions for the gradient obtained using the three different techniques introduced in sections B.1, B.2 and B.3.

	PDE-based derivation (Section B.1)	direct PDE, adjoint BIE (Section B.2)	BIEs-based derivation (Section B.3)
$\nabla\mathcal{J}$	$\nabla\mathcal{J} = -\frac{\partial u}{\partial n}\frac{\partial u^*}{\partial n}$ (B.7)	$\nabla\mathcal{J} = -\rho^*(\mathbf{x})\,\frac{\partial u}{\partial n}$ (B.14)	$\nabla\mathcal{J} = -\mu^*(\mathbf{x})\frac{\partial u}{\partial n}$ (B.33)
Adjoint System	$-\Delta u^* = RHS$ in Ω, $u^* = 0$ on Γ (B.8)	$G_\Gamma\rho^*(\mathbf{x}) = J_A(\omega, u)$ (B.13)	$G_\Gamma\mu^*(\mathbf{x}) = J_A(\omega, u)$ (B.31)

Table B.1: The gradients and adjoint systems derived in the different approaches

For the sake of the brevity of notation, in Table B.1 we denote $RHS = (\omega(u-\bar{u}))\chi_A$, $G_\Gamma\rho^*(\mathbf{x}) = \int_\Gamma \ln|\mathbf{x}-\mathbf{y}|\,\rho^*(\mathbf{x})\,ds_x$ and $J_A(\omega,u) = \int_A \omega(u-\bar{u})\ln|\mathbf{x}-\mathbf{y}|\,d\mathbf{x}$.

We emphasize that all variables entering the gradient formulation in the PDE-based approach are solutions of direct (B.2) and adjoint (B.8) PDE problems, whereas to obtain the gradient in the BIE-based approach we need to solve direct (B.16) and adjoint (B.31) BIEs. To find the variables entering the gradient formula in PDE-based shape-differentiation and BIE-based derivation of the adjoint equation case (cf. B.2), we have to deal with the direct equation in PDE form and the adjoint BIE.

We also note that the expressions for the gradient derived in the sections B.2 and B.3 are identical, since the adjoint variables μ^* and ρ^*, entering the gradient formulation, solve the identical adjoint equations (B.13), (B.31) and the integral in (B.33) is in fact $-\frac{\partial u}{\partial n}$ in (B.14).

Bibliography

[1] X. Peng, "Optimal geometry in a simple model of two-dimensional heat transfer," Master's thesis, McMaster University, 2011.

[2] A. Belmiloudi, *Heat Transfer: Mathematical Modelling, Numerical Methods and Information Technology*. InTech, 2011.

[3] J. Haslinger and R. Mäkinen, *Introduction to shape optimization: theory, approximation, and computation*, vol. 7. Society for Industrial Mathematics, 1987.

[4] C.-H. Huang and C.-T. Wuchiu, "A shape design problem in determining the interfacial surface of two bodies based on the desired system heat flux," *International Journal of Heat and Mass Transfer*, vol. 54, no. 11, pp. 2514–2524, 2011.

[5] D. T. Lin, C.-N. Huang, and C.-C. Chang, "The optimization of the heat removal on the led package," *Advanced Science Letters*, vol. 4, no. 6-7, pp. 6–7, 2011.

[6] C.-H. Huang and W.-L. Chang, "An inverse design method for optimizing design parameters of heat sink modules with encapsulated chip," *Applied Thermal Engineering*, vol. 40, pp. 216–226, 2012.

[7] B. Yun, S. Lee, and U. Choi, "A modified boundary integral method on open arcs in the plane," *Computers & Mathematics with Applications*, vol. 31, no. 11, pp. 37–43, 1996.

[8] K. E. Atkinson and I. H. Sloan, "The numerical solution of first-kind logarithmic-kernel integral equations on smooth open arcs," *Math. Comp*, vol. 56, no. 193, pp. 119–139, 1991.

[9] G. Rus and R. Gallego, "Hypersingular shape sensitivity boundary integral equation for crack identification under harmonic elastodynamic excitation," *Computer methods in applied mechanics and engineering*, vol. 196, no. 25, pp. 2596–2618, 2007.

[10] J. L. Lions, *Optimal control of systems governed by partial differential equations*, vol. 1200. Springer Berlin, 1971.

[11] M. S. Berger, *Nonlinearity and functional analysis: lectures on nonlinear problems in mathematical analysis*, vol. 74. Academic Press, 1977.

[12] O. Volkov and B. Protas, "An inverse model for a free-boundary problem with a contact line: Steady case," *Journal of Computational Physics*, vol. 228, no. 13, pp. 4893–4910, 2009.

[13] L. P. Lebedev and I. I. Vorovich, *Functional analysis in mechanics*. Springer, 2002.

[14] B. Protas, "An algorithm for symbolic generation of adjoint systems in pde optimization problems." Preprint, 2012.

[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*, vol. 195, no. 1, pp. 49–89, 2004.

[16] B. Protas, "Adjoint-based optimization of pde systems with alternative gradients," *Journal of Computational Physics*, vol. 227, no. 13, pp. 6490–6510, 2008.

[17] L. Trefethen, *Spectral methods in MATLAB*, vol. 10. Society for Industrial Mathematics, 2000.

[18] J. P. Boyd, *Chebyshev and Fourier spectral methods*. Dover publications, 2001.

[19] L. N. Trefethen, "Is gauss quadrature better than clenshaw-curtis?," *SIAM review*, vol. 50, no. 1, pp. 67–87, 2008.

[20] W. Hackbusch, *Integral equations: theory and numerical treatment*, vol. 120. Birkhäuser Basel, 1995.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

[22] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer verlag, 1999.

[23] 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*, vol. 189, no. 1, pp. 841–846, 2009.