TAMPERE UNIVERSITY OF TECHNOLOGY

JUKKA-PEKKA HUMALOJA
FREE BOUNDARY PROBLEM FOR HARMONIC FUNCTIONS

Master of Science Thesis

Examiners: Professor Seppo Pohjolainen and Senior Research Fellow Timo Hämäläinen
Examiner and topic approved by the Faculty Council of the Faculty of Natural Sciences on 4th December 2013

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Science and Engineering
HUMALOJA, JUKKA-PEKKA: Free Boundary Problem for Harmonic Functions
Master of Science Thesis, 57 pages, 10 Appendix pages
August 2014
Major subject: Mathematics
Examiners: Professor Seppo Pohjolainen and Senior Research Fellow Timo Hämäläinen
Keywords: free boundary problem, harmonic function, boundary element method, alternating-field technique, method of fundamental solutions

In this work we consider three different methods for solving a certain kind of free boundary problem for harmonic functions. The considered methods are the boundary element method, the alternating-field technique and the method of fundamental solutions. The main goal of this work is to find out if any of the above methods are eligible for solving the considered free boundary problem. We will also consider if the methods could be applied to solve similar free boundary problems in three dimensions.

When testing the performance of the different methods it turns out that the boundary element method does not seem suitable for solving the studied free boundary problem, while rather accurate solutions are obtained using the alternating-field technique and the method of fundamental solutions, out of which the former is found out to be the more reliable one. However, the alternating-field technique is restricted for two dimensional problems whereas the method of fundamental solutions could be relatively easily applied to three dimensional problems as well. Thus, out of the three considered methods the method of fundamental solutions should be the subject of further studies when considering thee dimensional free boundary problems.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Teknis-luonnontieteellinen koulutusohjelma
HUMALOJA, JUKKA-PEKKA: Vapaan reunan ongelma harmonisille funktioille
Diplomityö, 57 sivua, 10 liitesivua
Elokuu 2014
Pääaine: Matematiikka
Tarkastajat: professori Seppo Pohjolainen ja yliopistotutkija Timo Hämäläinen
Avainsanat: vapaan reunan ongelma, harmoninen funktio, reunaelementtimenetelmä, vuorottaiskenttämenetelmä, fundamentaaliratkaisujen menetelmä

Tässä työssä tarkastellaan kolmea eri menetelmää tietynlaisen vapaan reunan ongelman ratkaisemiseksi. Tarkasteltavat menetelmät ovat reunaelementtimenetelmä, vuorottaiskenttämenetelmä ja fundamentaaliratkaisujen menetelmä. Työn pääasiallinen tavoite on selvittää, voidaanko työssä tarkasteltavaa vapaan reunan ongelmaa harmonisille funktioille ratkaista käyttäen jotakin edellä mainituista menetelmistä. Lisäksi tarkastellaan, voitaisiinko menetelmiä soveltaa vastaavanlaisten kolmiulotteisten vapaan reunan ongelmien ratkaisemiseen.

Menetelmiä testattaessa käy ilmi, että reunaelementtimenetelmä ei vaikuttaisi soveltuvan tarkastellun vapaan reunan ongelman ratkaisemiseen, kun taas vuorottaiskenttämenetelmän ja fundamentaaliratkaisujen menetelmän tuottamat ratkaisut ovat varsin tarkkoja. Kahdesta viimeksi mainitusta menetelmästä vuorottaiskenttämenetelmä todetaan luotettavammaksi. Vuorottaiskenttämenetelmää voidaan kuitenkin käyttää ainoastaan kaksiulotteisten vapaan reunan ongelmien ratkaisemiseen, kun taas fundamentaaliratkaisujen menetelmä voitaisiin suhteellisen helposti muuntaa kolmiulotteisiin ongelmiin sopivaksi. Täten kaikista kolmesta menetelmästä fundamentaaliratkaisujen menetelmä olisi varteenotettavin vaihtoehto jatkotutkimuksen kohteeksi tarkasteltaessa kolmiulotteisia vapaan reunan ongelmia.

PREFACE

This work is done as a part of the project 'Inverse Problem with Free Boundary' in Tampere University of Technology, Department of Mathematics. It can more or less be considered as the final report for the first phase of the project.

First of all, I want to thank my supervisor Professor Seppo Pohjolainen for introducing me to such an interesting problem as the one considered in this work, for the opportunity to work full-time in the project while writing this thesis and for the guidance during the process. I also want to thank my supervisor Senior Research Fellow Timo Hämäläinen for his useful comments on my work, for the numerous MATLAB tips and for all the interesting conversations. Thirdly, I want to thank the other two representatives of the 'Inverse Problem with Free Boundary' project, whose help, feedback and ideas in the project meetings have contributed significantly to the progress of this work.

I want to thank Saku Suurniemi for his kind help in obtaining access to a certain book by Colton and Kress. Even though the very book does not appear in the references, it shed light on many subjects that were unclear by that time. I also want to thank my co-worker and friend Tuomas Myllykoski who was willing to help whenever needed, and who also brightened up many days with the well-timed coffee breaks.

Finally, I want to thank my parents and my sister for addressing their interest towards my work, for understanding the constraints on what can be told about it, and especially for all the support they have given during my studies.

Tampere, 23rd July 2014

Jukka-Pekka Humaloja
jukka-pekka.humaloja@tut.fi

CONTENTS

LIST OF SYMBOLS

$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{C}$	the set of complex numbers
z	a complex variable $z \in \mathbb{C}$, $z = x + iy$ such that $x, y \in \mathbb{R}$
$\mathbf{x}$	$\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$
$\mathbf{x} \cdot \mathbf{y}$	the dot product of $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \cdot \mathbf{y} = \sum\limits_{i=1}^{n} x_i y_i$
$\lvert\lvert\mathbf{x}\rvert\rvert$	the euclidian norm of $\mathbf{x}$, $\lvert\lvert\mathbf{x}\rvert\rvert = \sqrt{\mathbf{x} \cdot \mathbf{x}}$
$\dfrac{\partial f}{\partial x_k}$	the partial derivative of $f : \mathbb{R}^n \to X$ with respect to the variable x_k
R	a region R, $R \subset \mathbb{R}^n$
∇f	the gradient of $f : R \to \mathbb{R}$, $\nabla f = \left(\dfrac{\partial f}{\partial x_1}, \dfrac{\partial f}{\partial x_2}, \ldots, \dfrac{\partial f}{\partial x_n} \right)$
$\nabla^2 f$	the Laplacian of $f : R \to \mathbb{R}$, $\nabla^2 f = \nabla \cdot \nabla f$
∂R	the boundary of a region R
$\overline{R}$	the closure of a region R, $\overline{R} = R \cup \partial R$
$\mathbf{n}$	the unit exterior normal to a region R
$\dfrac{\partial f}{\partial n}$	the normal derivative of $f : R \to \mathbb{R}$, $\dfrac{\partial f}{\partial n} = \nabla f \cdot \mathbf{n}$
$\Phi(\mathbf{r}, \mathbf{r}_0)$	the fundamental solution of Laplace's equation with a singularity at the point $\mathbf{r}_0$
δ	the Dirac delta function
$C^2(R)$	the class of twice continuously differentiable functions in R
$f(D)$	the image of $f : D \to E$ in E
$[a,\, b]$	the closed interval from a to b, $[a,\, b] = \{x \in \mathbb{R} \,\vert\, a \leq x \leq b\}$

1. INTRODUCTION

This work considers a free boundary problem such as the one presented in Figure 1.1. A potential function $u : R \to \mathbb{R}$ satisfies Laplace's equation $\nabla^2 u = 0$ on the region R defined as

$$R = \{(x, y) \in \mathbb{R}^2 \,|\, x \in [a,\, b], \quad y \in [0,\, h(x)]\} \tag{1.1}$$

with a Neumann boundary condition $\partial u/\partial n = 0$ on the vertical boundaries $x = a$ and $x = b$, a Cauchy boundary condition $u = 0$ and $\partial u/\partial n = g(x)$ on the horizontal boundary $y = 0$ and a Dirichlet boundary condition $u = 1$ on the free boundary $y = h(x)$. Based on this information, we try to determine the function $h : [a,\, b] \to \mathbb{R}$.

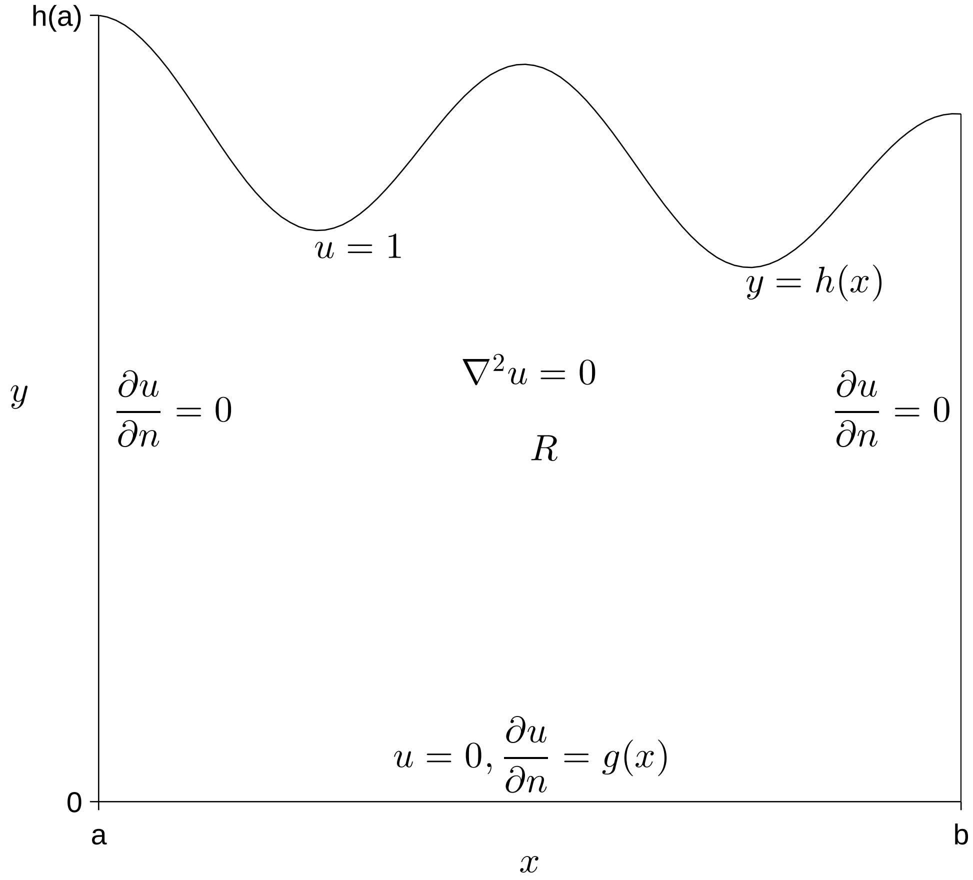
h(a)
$u = 1$
$y = h(x)$
y
$\frac{\partial u}{\partial n} = 0$
$\nabla^2 u = 0$
R
$\frac{\partial u}{\partial n} = 0$
$u = 0, \frac{\partial u}{\partial n} = g(x)$
0
a
b
x

Figure 1.1: *The problem on the region R.*

The main focus of this work is to study different numerical methods for solving the problem presented in Figure 1.1, while the analytic consideration of the problem is given less weight. The numerical methods studied are the boundary element method, the alternating-field technique and the method of fundamental solutions, which are all presented in the correspondingly named chapters. Furthermore, MATLAB implementations for testing the methods are given in the appendices.

Free boundary problems such as the one presented in Figure 1.1 are quite rarely considered in the literature per se, but a problem with a somewhat similar geometry is discussed in [14] by Nilson and Tsuei. On the other hand, there is a considerable range of publications concerning inclusion or cavity detection problems, such as [2], [11] and [15], and our problem can be related to those problems as we shall see in Section 2.4.

The outline of this work is as follows. In Chapter 2 we introduce some terminology related to this work and some results required to consider the numerical methods studied in this work. Chapters 3, 4 and 5 are dedicated to the different numerical methods, the boundary element method, the alternating-field technique and the method of fundamental solutions, respectively. In Chapter 6, we compare the performance of these methods in a few test cases and in Chapter 7, we conclude our work.

This work is done as a part of the project 'Inverse Problem with Free Boundary' at Tampere University of Technology, Department of Mathematics. Since a future goal of the project is to study the problem presented in Figure 1.1 in three dimensions, applicability to 3D is kept in mind when considering the different numerical methods.

2. THEORETICAL BACKGROUND

2.1 Laplace's Equation

Laplace's equation

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \ldots + \frac{\partial^2 u}{\partial x_n^2} = 0 \tag{2.1}$$

or equivalently

$$\nabla^2 u = 0 \tag{2.2}$$

arises in the study of steady state phenomena, i.e., phenomena that do not depend on time [19, pp. 161,172]. It is a special case of *Poisson's equation* [4, p. 222]

$$\nabla^2 u = f \tag{2.3}$$

for $f = 0$. In this work, Laplace's equation is considered in a *bounded*, *simply connected* region R [17, p. 475], the boundary of which is denoted by ∂R.

Equation 2.2 would have infinitely many solutions, but by specifying supplementary conditions in the form of *boundary conditions* the number of solutions can be reduced to at most one [19, p. 161]. There are three kinds of boundary conditions present in this work, namely [16, p. 194]

- *Dirichlet boundary conditions*: The value of u is known on the boundary ∂R
- *Neumann boundary conditions*: The normal derivative of u, i.e., $\partial u/\partial n$, is known on the boundary ∂R
- *Cauchy boundary conditions*: Both the value of u and $\partial u/\partial n$ are known on the boundary ∂R.

Different kinds of boundary conditions may be specified over different parts of the boundary ∂R, in which case the *boundary conditions* are called *mixed* [4, 208]. The problem of finding the solution to Laplace's equation satisfying certain boundary conditions is called a *boundary value problem* [19, p. 161].

Let us next introduce the *fundamental solution of Laplace's equation* [4, p. 242] and *harmonic functions* [19, p. 172] by the following definitions.

Definition 2.1. *A fundamental solution of Laplace's equation* (2.2) *in* $\mathbb{R}^n$ *with a* *singularity at* $\mathbf{r}_0$ *is any function* Φ *which satisfies Poisson's equation*

$$\nabla^2\Phi(\mathbf{r},\mathbf{r}_0) = \delta(\mathbf{r}-\mathbf{r}_0), \tag{2.4}$$

where δ *is the Dirac delta function.*

Definition 2.2. *Let* R *be a region in* $\mathbb{R}^n$*. A function* $u \in C^2(R)$ *which satisfies Laplace's equation in* R *is called harmonic in* R*.*

Note that a fundamental solution of Laplace's equation $\Phi(\mathbf{r},\mathbf{r}_0)$ is harmonic in a region R if and only if $\mathbf{r}_0 \notin R$.

We end this section by presenting Green's theorem in the plane [17, p. 457] and two of its corollaries [1, pp. 11, 50] concerning harmonic functions in a bounded, simply connected region R in $\mathbb{R}^2$ with the boundary ∂R. The line element of ∂R is denoted by $\mathrm{d}s(x,y)$.

Theorem 2.3. *Let* R *be a closed and bounded region in* $\mathbb{R}^2$ *whose boundary* ∂R *consists of finitely many simple closed curves which do not intersect each other and are sectionally smooth [17, p. 418]. Let* $\mathbf{F} : R \to \mathbb{R}^2$ *be such that* $\nabla \cdot \mathbf{F}$ *exists. Then*

$$\int\limits_{\partial R} \mathbf{F}\cdot\mathbf{n}\,\mathrm{d}s(x,y) = \iint\limits_{R} \nabla\cdot\mathbf{F}\,\mathrm{d}x\,\mathrm{d}y. \tag{2.5}$$

Corollary 2.4. *Let* R *be a bounded, simply connected region in* $\mathbb{R}^2$ *and let* u_1 *and* u_2 *be harmonic functions in* R*. Then*

$$\int\limits_{\partial R} \left(u_2\frac{\partial u_1}{\partial n} - u_1\frac{\partial u_2}{\partial n}\right)\mathrm{d}s(x,y) = 0. \tag{2.6}$$

Proof.

$$\begin{aligned}
&\int\limits_{\partial R} \left(u_2\frac{\partial u_1}{\partial n} - u_1\frac{\partial u_2}{\partial n}\right)\mathrm{d}s(x,y) \\
=\ &\int\limits_{\partial R} (u_2\nabla u_1\cdot\mathbf{n} - u_1\nabla u_2\cdot\mathbf{n})\mathrm{d}s(x,y) \\
=\ &\int\limits_{\partial R} [(u_2\nabla u_1 - u_1\nabla u_2)\cdot\mathbf{n}]\mathrm{d}s(x,y) \\
=\ &\iint\limits_{R} [\nabla\cdot(u_2\nabla u_1 - u_1\nabla u_2)]\mathrm{d}x\mathrm{d}y \\
=\ &\iint\limits_{R} (\nabla u_2\cdot\nabla u_1 + u_2\nabla^2 u_1 - \nabla u_1\cdot\nabla u_2 - u_1\nabla^2 u_2)\mathrm{d}x\mathrm{d}y \\
=\ &\iint\limits_{R} (\nabla u_1\cdot\nabla u_2 - \nabla u_1\cdot\nabla u_2)\mathrm{d}x\mathrm{d}y \\
=\ &0.
\end{aligned} \tag{2.7}$$

□

Corollary 2.5. *Let R be a bounded, simply connected region in $\mathbb{R}^2$ and let u be a harmonic function in R. Then*

$$\int\limits_{\partial R} \frac{\partial}{\partial n} u(x,y)\mathrm{d}s(x,y) = 0. \tag{2.8}$$

Proof.

$$\begin{aligned} \int\limits_{\partial R} \frac{\partial}{\partial n} u(x,y)\mathrm{d}s(x,y) &= \int\limits_{\partial R} \nabla u(x,y) \cdot \mathbf{n}\, \mathrm{d}s(x,y) \\ &= \iint\limits_{R} \nabla \cdot \nabla u(x,y)\mathrm{d}x\mathrm{d}y \\ &= \iint\limits_{R} \nabla^2 u(x,y)\mathrm{d}x\mathrm{d}y \\ &= 0. \end{aligned} \tag{2.9}$$

□

2.2 Free Boundary Problems

Let R be a region in $\mathbb{R}^n$ and let $u : R \to \mathbb{R}$ be a harmonic function in R. Let us divide the boundary of the region R in a fixed part ∂R_{fx} and a free part ∂R_{fr} such that $\partial R = \partial R_{fx} \cup \partial R_{fr}$. Let u satisfy certain boundary conditions in the fixed and in the free parts of the boundary ∂R. The goal of *free boundary problems* is to determine the location of ∂R_{fr} based on the information about the location of ∂R_{fx} and on the given boundary conditions for u. [7, pp. 80–81]

Free boundary problems are sometimes referred as *inverse boundary value problems*, since the formulations of these two problems are much alike. A typical example of a free boundary problem is an inclusion estimation problem where the harmonic function u satisfies a Cauchy boundary condition on the exterior boundary ∂R_{fx} of a region R, and in the interior of R there are inclusions of unknown shape, the boundaries of which form the free boundary ∂R_{fr}. On the free boundary, u satisfies either a Neumann or a Dirichlet boundary condition, depending on the application. [15, pp. 1, 3]

Free boundary problems are well known to be severely *ill-posed problems* [15, p. 1]. In order to clarify what that means, we define a *well-posed problem* [19, pp. 167–168] by

Definition 2.6. *A problem involving a partial differential equation is called well-posed if all the following requirements are satisfied:*

i A solution of the problem exists.

ii The solution is unique.

iii The solution depends continuously on the data of the problem.

If at least one of the requirements $i - iii$ is not satisfied, a problem is called *ill-posed* [10, p. 21]. In requirement *iii*, the phrase *solution depends continuously on the data* means that [19, p. 167] if u_1 and u_2 are solutions of a problem on a region R with different data f_1 and f_2, respectively, then

$$\forall \epsilon > 0 \; \exists \delta > 0 : \max_{\mathbf{r} \in \partial R} ||f_1(\mathbf{r}) - f_2(\mathbf{r})|| < \delta \Rightarrow \max_{\mathbf{r} \in \overline{R}} ||u_1(\mathbf{r}) - u_2(\mathbf{r})|| < \epsilon. \tag{2.10}$$

The ill-posed nature of free boundary problems arises from the fact that quite substantial changes in the free boundary may result extremely small changes in the data of the problem, which violates requirement *iii* in Definition 2.6 [15, p. 1]. The basic idea of solving such ill-posed problems is to use *regularization*, i.e., to modify the problem a little, so that the modified problem can be solved in a stable way and its solution is close to the solution of the original problem [10, p. 24]. Regularization will be further discussed within the presentations of the numerical methods in Chapters 3, 4 and 5.

2.3 Properties of Functions of Complex Variables

In this section, f is a function of the complex variable $z = x + iy$ with component functions u and v such that $f(z) = u(x, y) + iv(x, y)$. We shall begin this section with a few definitions that will introduce the concepts of *differentiability* [3, p. 54] and *holomorphic functions* [3, p. 70].

Definition 2.7. *Let a neighborhood of a point z_0 be contained in the domain of definition of a function f. The derivative of f at z_0, written $df(z_0)/dz$, is defined by*

$$\frac{d}{dz} f(z_0) = \lim_{z \to z_0} \frac{f(z) - f(z_0)}{z - z_0}, \tag{2.11}$$

provided that the limit exists. If the limit does exist, the function f is said to be differentiable at z_0.

Definition 2.8. *A function f of the complex variable z is holomorphic in an open set if it is differentiable at each point in that set.*

Before introducing further properties of holomorphic functions, we should present the differentiation formula for composite functions. Suppose that a function f is differentiable at z_0 and that a function g is differentiable at $f(z_0)$. Then the function $F(z) = g[f(z)]$ is differentiable at z_0. If we write $w = f(z)$ and $W = g(w)$, we obtain $dW/dz = dF(z)/dz$ by

$$\frac{dW}{dz} = \frac{dW}{dw}\frac{dw}{dz}. \tag{2.12}$$

The formula is known as the *chain rule* and it is analogous to the chain rule for differentiating composite functions of real variables. [3, pp. 57–58]

From now on, suppose that f is holomorphic in an arbitrary open set D. Then, the derivatives of f of all orders exist in D, and they are all holomorphic there [3, p. 160]. As a consequence, the component functions of f have continuous partial derivatives of all orders in D [3, p. 161].

Now, if we write $w = f(z)$ and $z = x + iy$, we may use the chain rule to point out that

$$\frac{\partial w}{\partial x} = \frac{df}{dz}\frac{\partial z}{\partial x} = \frac{df}{dz} \tag{2.13}$$

and that

$$\frac{\partial w}{\partial y} = \frac{df}{dz}\frac{\partial z}{\partial y} = i\frac{df}{dz}, \tag{2.14}$$

since $\partial z/\partial x = 1$ and $\partial z/\partial y = i$. Furthermore, if we write $w = u(x, y) + iv(x, y)$, we obtain

$$\frac{\partial w}{\partial x} = \frac{\partial u}{\partial x} + i\frac{\partial v}{\partial x} = \frac{df}{dz} \tag{2.15}$$

and

$$\frac{\partial w}{\partial y} = \frac{\partial u}{\partial y} + i\frac{\partial v}{\partial y} = i\frac{df}{dz}. \tag{2.16}$$

Eliminating df/dz from Equations (2.15) and (2.16) gives

$$i\left(\frac{\partial u}{\partial x} + i\frac{\partial v}{\partial x}\right) = \frac{\partial u}{\partial y} + i\frac{\partial v}{\partial y}, \tag{2.17}$$

and by equating the real and the imaginary parts from Equation (2.17) we obtain

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \tag{2.18}$$

which are the *Cauchy-Riemann equations.* [18, pp. 154–155]

Next, we will use the Cauchy-Riemann equations to prove the following theorem [3, p. 76].

Theorem 2.9. *If a function $f(z) = u(x, y) + iv(x, y)$ is holomorphic in an arbitrary open set D, then its component functions u and v are harmonic in D.*

Proof. Since f is assumed to be holomorphic in D, the first order partial derivatives of u and v must satisfy the Cauchy-Riemann equations throughout D. Differentiating both sides of these equations with respect to x gives

$$\frac{\partial}{\partial x}\frac{\partial u}{\partial x}=\frac{\partial}{\partial x}\frac{\partial v}{\partial y}\quad\text{and}\quad\frac{\partial}{\partial x}\frac{\partial u}{\partial y}=-\frac{\partial}{\partial x}\frac{\partial v}{\partial x},\tag{2.19}$$

and differentiating both sides with respect to y gives

$$\frac{\partial}{\partial y}\frac{\partial u}{\partial x}=\frac{\partial}{\partial y}\frac{\partial v}{\partial y}\quad\text{and}\quad\frac{\partial}{\partial y}\frac{\partial u}{\partial y}=-\frac{\partial}{\partial y}\frac{\partial v}{\partial x}.\tag{2.20}$$

Now, since u and v have continuous partial derivatives of all orders in D, according to Theorem IV in [17, p. 201] we have

$$\frac{\partial}{\partial y}\frac{\partial u}{\partial x}=\frac{\partial}{\partial x}\frac{\partial u}{\partial y}\quad\text{and}\quad\frac{\partial}{\partial x}\frac{\partial v}{\partial y}=\frac{\partial}{\partial y}\frac{\partial v}{\partial x},\tag{2.21}$$

and thus, we obtain from Equations (2.19), (2.20) and (2.21) that

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0\quad\text{and}\quad\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}=0,\tag{2.22}$$

i.e., u and v are harmonic in D. □

Note that in addition to u and v being harmonic in D, their first-order partial derivatives satisfy the Cauchy-Riemann equations throughout D. Such two functions are called *harmonic conjugates* in D [3, p. 77]. Conversely, it is shown in [3, pp. 63–65] that if the component functions of a function $f(z)=u(x,y)+iv(x,y)$ have continuous first-order partial derivatives and they satisfy the Cauchy-Riemann equations in an arbitrary open set D, then $f(z)$ is holomorphic in D.

Now, consider a harmonic function $u(x,y)$ on a simply connected domain D. We will show that there always exists a function $v(x,y)$ such that u and v are harmonic conjugates in D. In order to do this, let us recall a few results concerning line integrals.

Suppose that $P(x,y)$ and $Q(x,y)$ have continuous first-order partial derivatives in a simply connected domain D. If $\partial P/\partial y=\partial Q/\partial x$ everywhere in D, then the line integral

$$\int_C \Big(P(s,t)\mathrm{d}s+Q(s,t)\mathrm{d}t\Big)\tag{2.23}$$

is independent of the contour C as long as the contour lies entirely in D. Now, if we choose a fixed point (x_0,y_0) and a varying point (x,y), both in D, the integral represents a function

$$F(x,y) = \int\limits_{(x_0,y_0)}^{(x,y)} \Big(P(s,t)\mathrm{d}s + Q(s,t)\mathrm{d}t\Big), \tag{2.24}$$

the first-order partial derivatives of which are given by

$$\frac{\partial}{\partial x}F(x,y) = P(x,y), \quad \frac{\partial}{\partial y}F(x,y) = Q(x,y). \tag{2.25}$$

Note that the choice of the point (x_0, y_0) affects the value of F by an additive constant. [3, p. 352]

Let us now return to the given harmonic function $u(x, y)$. Since u is harmonic in D, it follows that

$$\frac{\partial}{\partial y}\bigg(-\frac{\partial u}{\partial y}\bigg) = \frac{\partial}{\partial x}\frac{\partial u}{\partial x} \tag{2.26}$$

everywhere in D, and that the first-order partial derivatives of $\partial u/\partial y$ and $\partial u/\partial x$ are continuous there. Thus, if (x_0, y_0) is a fixed point in D, the function

$$v(x,y) = \int\limits_{(x_0,y_0)}^{(x,y)} \bigg(-\frac{\partial}{\partial t}u(s,t)\mathrm{d}s + \frac{\partial}{\partial s}u(s,t)\mathrm{d}t\bigg) \tag{2.27}$$

is well defined in D, and according to Equation (2.25) we have

$$\frac{\partial}{\partial x}v(x,y) = -\frac{\partial}{\partial y}u(x,y), \quad \frac{\partial}{\partial y}v(x,y) = \frac{\partial}{\partial x}u(x,y), \tag{2.28}$$

which are the Cauchy-Riemann equations. Since the first-order partial derivatives of u are continuous, also the first-order partial derivatives of v must be continuous. Thus, the function $u(x, y) + iv(x, y)$ is holomorphic in D, implying that u and v are harmonic conjugates. It should be also noted that since the point $(x_0, y_0) \in D$ can be chosen arbitrarily, the value of v can be changed by an additive real constant without affecting u and v being harmonic conjugates. [3, pp. 352–353]

For the end of this section, we consider the inverse of a holomorphic function. If a function $f(z) = u(x, y) + iv(x, y)$ is invertible, holomorphic function in an open set D and if $df(z_0)/dz \neq 0$ at each point $z_0 \in D$, then f has a holomorphic inverse $F(w) = x(u, v) + iy(u, v)$ such that $F[f(z)] = z$ [3, pp. 348–349]. Since F is holomorphic in $f(D)$, x and y must be harmonic in $f(D)$ with respect to u and v [18, pp. 155–156]. Furthermore, as we saw earlier, F being holomorphic in $f(D)$ also implies that the first-order partial derivatives of its component functions x and

y must satisfy the Cauchy-Riemann equations in $f(D)$. In conclusion, we have

$$\frac{\partial^2 x}{\partial u^2} + \frac{\partial^2 x}{\partial v^2} = 0 \quad \text{and} \quad \frac{\partial^2 y}{\partial u^2} + \frac{\partial^2 y}{\partial v^2} = 0, \tag{2.29}$$

and

$$\frac{\partial x}{\partial u} = \frac{\partial y}{\partial v} \quad \text{and} \quad \frac{\partial x}{\partial v} = -\frac{\partial y}{\partial u} \tag{2.30}$$

in $f(D)$.

2.4 Conformal Mapping and the Method of Images

Consider a transformation $w = f(z)$, where f is a function of the complex variable $z = x + iy$. The transformation $w = f(z)$ is called *conformal* at a point z_0 if f is holomorphic there and $df(z_0)/dz \neq 0$. Furthermore, a transformation $w = f(z)$, defined on a domain D, that is conformal at each point in D is called a *conformal mapping.* The term conformal refers to the angle-preserving property of these mappings, that is, the magnitude and the sense of an angle between any two smooth arcs remain unchanged under a conformal mapping. [3, pp. 344–345]

Conformal mappings can be used in solving boundary value problems by mapping the domain of the problem to a more favorable one. The following theorem proves that harmonic functions are invariant under conformal mappings [3, p. 354]. The theorem actually holds for any domain, but for our purposes it is sufficient to consider only simply connected domains.

Theorem 2.10. *Suppose that a conformal mapping $\zeta = F(z) = \xi(x,y) + i\psi(x,y)$ maps a simply connected domain D_z in the z-plane onto a simply connected domain D_ζ in the ζ-plane. If $u(\xi,\psi)$ is a harmonic function in D_ζ, then the function $u[\xi(x,y),\psi(x,y)]$ is harmonic in D_z.*

Proof. Since the domain D_ζ is supposed to be simply connected, the harmonic function $u(\xi,\psi)$ has a harmonic conjugate $v(\xi,\psi)$ such that the function $f(\zeta) = u(\xi,\psi) + iv(\xi,\psi)$ is holomorphic in D_ζ. Now the composite function $f[F(z)]$ is holomorphic in D_z [3, p. 71], and thus, its real part $u[\xi(x,y),\psi(x,y)]$ is harmonic in D_z. □

When the domain of a boundary value problem is conformally mapped to a different one, the boundary conditions have to be transformed as well. Suppose that we have a Dirichlet boundary condition $u = u_0$ on a smooth arc Γ which is the image of a smooth arc C under a conformal mapping $\zeta = F(z) = \xi(x,y) + i\psi(x,y)$. Now we see that the value of the function $u[\xi(x,y),\psi(x,y)]$ at any point (x,y) on C is the same as the value of $u(\xi,\psi)$ at the point (ξ,ψ) on Γ. Since (ξ,ψ) is the image of

(x, y) under the mapping $F(z)$, the boundary condition $u = u_0$ holds for both Γ and C, i.e., Dirichlet boundary conditions are invariant under conformal mappings. [3, p. 356]

In the case of Neumann boundary conditions a transformation actually occurs. Consider the above mapping $F(z)$ and the arcs Γ and C, and denote the derivatives of u normal to Γ and C by $\partial u/\partial \eta$ and $\partial u/\partial n$, respectively. Then we have the relation

$$\frac{\partial}{\partial n} u[\xi(x, y), \psi(x, y)] = \left|\frac{dF(z)}{dz}\right| \frac{\partial}{\partial \eta} u(\xi, \psi) \tag{2.31}$$

which can be obtained by calculating the gradient of $u[\xi(x, y), \psi(x, y)]$ and using the fact that the mapping $F(z)$ is conformal. First of all, one should recall from Equation (2.15) that

$$\left|\frac{dF}{dz}\right| = \sqrt{\left(\frac{\partial \xi}{\partial x}\right)^2 + \left(\frac{\partial \psi}{\partial x}\right)^2}. \tag{2.32}$$

Now, if we consider the partial derivatives of u with respect to x and y, using the chain rule (2.12) we obtain

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \psi}\frac{\partial \psi}{\partial x} \quad \text{and} \quad \frac{\partial u}{\partial y} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial u}{\partial \psi}\frac{\partial \psi}{\partial y}. \tag{2.33}$$

Since $F(z)$ as a conformal mapping is a holomorphic function, the first order partial derivatives of its component functions ξ and ψ satisfy the Cauchy-Riemann equations (2.18) and thus, we may rewrite

$$\frac{\partial u}{\partial y} = -\frac{\partial u}{\partial \xi}\frac{\partial \psi}{\partial x} + \frac{\partial u}{\partial \psi}\frac{\partial \xi}{\partial x}. \tag{2.34}$$

Finally, calculating $||\nabla u[\xi(x, y), \psi(x, y)]||$ gives

$$\begin{aligned}
& ||\nabla u[\xi(x, y), \psi(x, y)]|| \\
= & \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2} \\
= & \sqrt{\left(\frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial \psi}\frac{\partial \psi}{\partial x}\right)^2 + \left(-\frac{\partial u}{\partial \xi}\frac{\partial \psi}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial \psi}\frac{\partial \xi}{\partial x}\right)^2} \\
= & \sqrt{\left[\left(\frac{\partial u}{\partial \xi}\right)^2 + \left(\frac{\partial u}{\partial \psi}\right)^2\right]\left[\left(\frac{\partial \xi}{\partial x}\right)^2 + \left(\frac{\partial \psi}{\partial x}\right)^2\right]} \\
= & ||\nabla u(\xi, \psi)|| \left|\frac{dF(z)}{dz}\right|.
\end{aligned} \tag{2.35}$$

Now, since $F(x)$ is conformal, the angle between $\nabla u(\xi, \psi)$ and the normal $\boldsymbol{\eta}$ is the same as the angle between $\nabla u[\xi(x,y), \psi(x,y)]$ and the normal $\mathbf{n}$. Thus, we have reached the relation given in equation (2.31). [3, pp. 348, 356–358, 360]

Now that we have justified means of using conformal mappings in solving boundary value problems for Laplace's equation, we may present a useful mapping that maps the semi-infinite strip $a \leq x \leq b$, $y \geq 0$ to the half unit disk $x^2 + y^2 \leq 1$, $x \geq 0$. The mapping is given by

$$\zeta = F(z) = \exp\left[\frac{i\pi}{b-a}\left(z - \frac{b+a}{2}\right)\right], \tag{2.36}$$

and when it is applied to the geometry of our problem, the resulting region is displayed in Figure 2.1. We see that the line $y = 0$ in the original problem maps to the half unit circle $\xi^2 + \psi^2 = 1$, $\xi \geq 0$ in the ζ-plane and the unknown boundary curve $y = h(x)$, denoted by $\mathbf{r}$ in the ζ-plane, in a sense cuts a part off of the half unit disk. The values of the function G are obtained by using the relation given in Equation (2.31), from which we obtain

$$G(\zeta) = \left|\frac{dF(z)}{dz}\right|^{-1} g(z) = \frac{b-a}{\pi} g(z), \tag{2.37}$$

where $\zeta = \xi + i\psi$ and $z = x + iy$.

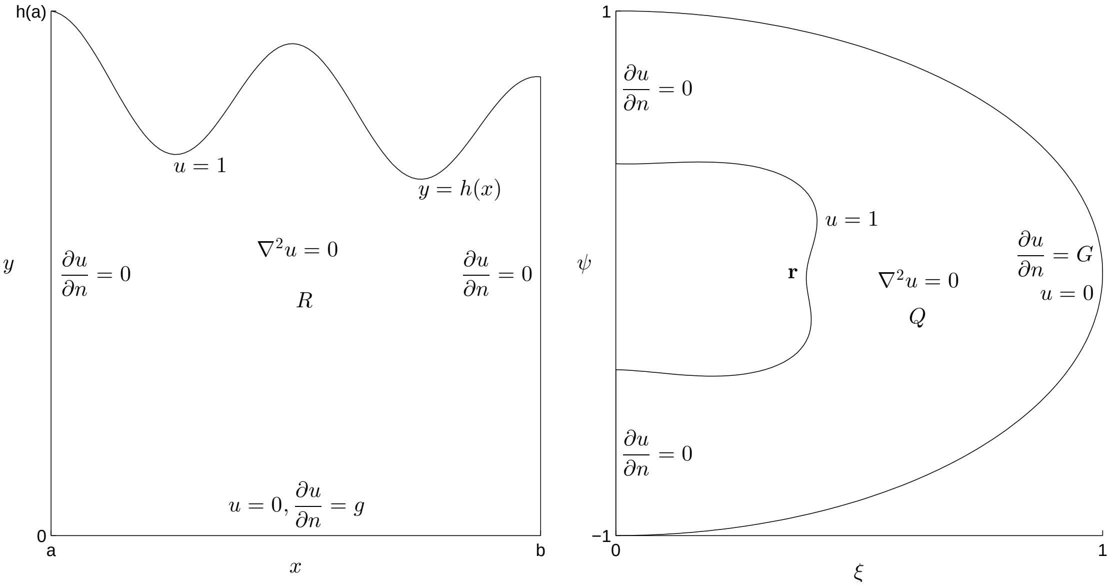
h(a)
$u = 1$
$y = h(x)$
y
$\frac{\partial u}{\partial n} = 0$
$\nabla^2 u = 0$
R
$\frac{\partial u}{\partial n} = 0$
$u = 0, \frac{\partial u}{\partial n} = g$
0
a
b
x
1
$\frac{\partial u}{\partial n} = 0$
$u = 1$
$\frac{\partial u}{\partial n} = G$
$u = 0$
ψ
$\mathbf{r}$
$\nabla^2 u = 0$
Q
$\frac{\partial u}{\partial n} = 0$
−1
0
1
ξ

Figure 2.1: *The geometry of the original problem and its conformal counterpart under the mapping $F(z)$.*

So far the conformal mapping has not shown any advantages, but in fact, the region in the ζ-plane can be reflected with respect to the line $\xi = 0$, and thus, we will obtain a simpler, *reduced* problem, the solution of which also satisfies the problem on the region Q. The geometry of the reduced problem is displayed in Figure 2.2, where $G_*(\xi, \psi) = G(|\xi|, \psi)$. [4, pp. 560–561]

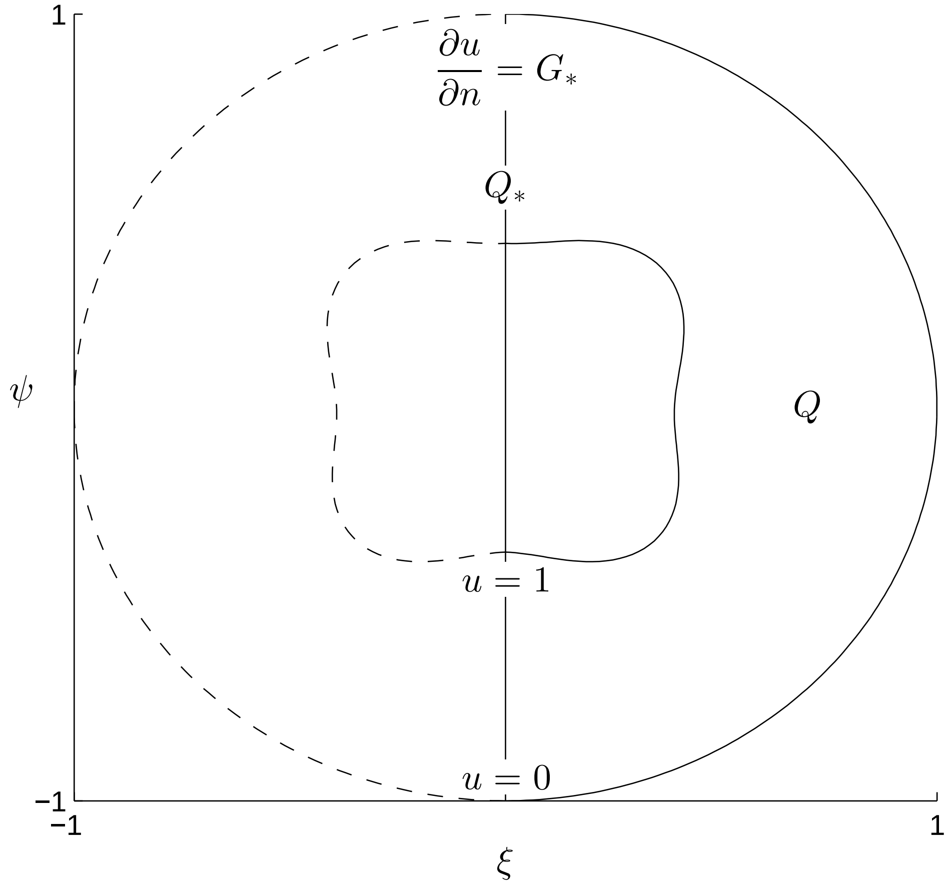

Figure 2.2: *Geometry of the reduced problem.*

Due to symmetry with respect to the line $\xi = 0$ it is actually rather intuitive that the solution of the boundary value problem on the region Q_* is also a solution to the boundary value problem on the region Q, i.e., the solution of the boundary value problem on the region Q_* intrinsically satisfies the condition $\partial u/\partial \xi = 0$ on the line $\xi = 0$. Similar symmetry methods can be also applied to boundary conditions of the form $u = 0$, which will be done within the method of fundamental solutions in Chapter 5 [4, pp. 560–561].

The geometry of the reduced problem in Figure 2.2 now corresponds to the geometry of a problem of finding obstacles within a circular host medium based on measurements on the exterior boundary, which is exactly what is considered, e.g., in [2], [11] and [15]. However, since the conformal mapping $F(z)$ cannot be applied to three dimensional problems, we do not actually test the methods presented in the above papers to the reduced problem but concentrate on finding methods to solve the problem in its original geometry. If one would actually solve the free boundary problem in the reduced form and construct the boundary curve $\mathbf{r}$ in the ζ-plane, one then needs the inverse of $F(z)$ given by

$$F^{-1}(\zeta) = -i\frac{b-a}{\pi}\ln\zeta + \frac{b+a}{2} \tag{2.38}$$

in order to map the solution back to the z-plane and thus, to obtain the reconstruction of the free boundary curve in the z-plane.

2.5 Required Basic Numerical Methods

In this section, we will present the basic numerical methods used within the implementations of the algorithms in Chapters 3, 4 and 5. These methods cover difference approximations of derivatives, numerical integration methods and interpolation by second degree polynomials.

Consider a function $v : D \to \mathbb{R}$ of x and y, where $D \subseteq \mathbb{R}^2$. The *forward difference*, the *backward difference* and the *central difference* approximations for the partial derivative of v with respect to x are given by [6, p. 145]

$$\frac{\partial}{\partial x}v(x,y) \approx \frac{v(x+h,y)-v(x,y)}{h}, \tag{2.39a}$$
$$\frac{\partial}{\partial x}v(x,y) \approx \frac{v(x,y)-v(x-h,y)}{h}, \tag{2.39b}$$
$$\frac{\partial}{\partial x}v(x,y) \approx \frac{v(x+h,y)-v(x-h,y)}{2h}, \tag{2.39c}$$

respectively, provided that $(x+h,y) \in D$ and $(x-h,y) \in D$ [19, p. 249]. Similarly, the second order partial derivative of v with respect to x can be approximated by [19, p. 252]

$$\frac{\partial^2}{\partial x^2}v(x,y) \approx \frac{v(x+h,y)-2v(x,y)+v(x-h,y)}{h^2}. \tag{2.40}$$

Approximations for the partial derivatives of v with respect to y are obtained analogously.

Next, consider a definite integral

$$\int_a^b f(x)\mathrm{d}x, \tag{2.41}$$

where $f : [a,b] \to \mathbb{R}$ and $a, b \in \mathbb{R}$ such that $a < b$. The integral can be approximated by the *trapezoidal rule* given by

$$\int_a^b f(x)\mathrm{d}x \approx h\left(\frac{f_0 + f_N}{2} + \sum_{k=1}^{N-1} f_k\right), \tag{2.42}$$

where $N \in \mathbb{N}$, $h = (b-a)/N$ and $f_k = f(a + kh)$ [6, p. 166]. A more accurate approximation for the integral is obtained by *Simpson's rule*, given by

$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{3}\left(f_0 + f_N + 2\sum_{k=1}^{\frac{1}{2}N-1} f_{2k} + 4\sum_{k=1}^{\frac{1}{2}N} f_{2k-1}\right), \tag{2.43}$$

where N is even, and h and f_k have the same interpretations as in the trapezoidal rule [6, p. 170].

For the end of this section, we will present a second degree polynomial $P(x)$ that interpolates a given function f at three given points. In order to do this, denote the three points where the value of f is known by x_0, x_1 and x_2. If we write the polynomial P in the form

$$P(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1). \tag{2.44}$$

where the coefficients c_0, c_1 and c_2 are given by

$$\begin{aligned} c_0 &= f(x_0), \\ c_1 &= \frac{f(x_1) - c_0}{x_1 - x_0}, \\ c_2 &= \frac{f(x_2) - c_0 - c_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}, \end{aligned} \tag{2.45}$$

it is easy to see that with such coefficients $P(x_i) = f(x_i)$ for $i = 0, 1, 2$, i.e., the polynomial P interpolates the function f at the points x_0, x_1 and x_2. [6, pp. 100–101]

2.6 About Optimization and Optimization Methods

In Chapters 3 and 5 we will see that solving the free boundary problem presented in Figure 1.1 by using the boundary element method or the method of fundamental solutions comes down to solving an *optimization problem.* Thus, in this section, we will briefly consider the basic concepts of optimization as well as certain optimization methods from MATLAB's Optimization Toolbox Release 2013b [13], among which we will choose the most suitable methods for solving the optimization problems we will encounter in Chapters 3 and 5.

First of all, we should clarify what is meant by an *optimization problem.* The goal of such a problem is to minimize a real-valued function f of N variables, that is, to find a point $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \text{ near } \mathbf{x}^*. \tag{2.46}$$

The function f is referred to as the *objective function* and the point $\mathbf{x}^*$ is called a *local minimizer*. Ideally we should find a *global minimizer*, i.e., a point $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{R}^N. \tag{2.47}$$

However, finding a global minimizer is different from, and much more complicated than finding a local minimizer. Hence, we will content ourselves with methods for finding a local minimizer and will refer to a local minimizer simply as a *minimizer*. [12, p. 3]

It is standard to express an optimization problem as

$$\min_{\mathbf{x}} f(\mathbf{x}). \tag{2.48}$$

Formally the problem presented in Equation (2.48) is called *unconstrained* since we impose no conditions on the minimizer $\mathbf{x}^*$. The *constrained* minimization problem is to find a minimizer over a set $U \subset \mathbb{R}^N$, and it is formally expressed as

$$\min_{\mathbf{x} \in U} f(\mathbf{x}). \tag{2.49}$$

It should be noted that in this work we will use the notation of Equation (2.48) for optimization problems in general and express the possible constraints in other ways. [12, p. 3]

We will now present the concept of *Least squares* problems, also known as *data fitting* problems. These problems are special cases of optimization problems where the objective function is of the form

$$f(\mathbf{x}) = ||\mathbf{F}(\mathbf{x})||^2, \tag{2.50}$$

where $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^M$ [13]. If $M > N$ the problem is called *overdetermined*, and if $M < N$ the problem is called *underdetermined* [12, p. 22]. The optimization problems we will encounter in Chapters 3 and 5 can be formulated as least squares problems. This aspect will give us more options when choosing the appropriate optimization method, since there are algorithms designed specifically for solving least squares problems [13].

We will next consider briefly the pros and cons of the different optimization methods available in MATLAB's Optimization Toolbox [13]. It should be mentioned that the optimization problems we will encounter will be nonlinear, which narrows our possibilities significantly. Additionally, in both Chapters 3 and 5 we are willing to impose at least some constraints on the minimizer, and hence, we will only consider methods for constrained nonlinear optimization.

MATLAB's Optimization Toolbox has a function called `fmincon` for constrained nonlinear optimization problems and a function called `fseminf` for semi-infinitely constrained nonlinear optimization problems. One can choose between multiple algorithms when using these functions, and the different possibilities are described in detail in the MATLAB documentation [13]. The advantage of these functions is that one can freely impose constraints on the minimizer in the form of linear and nonlinear equalities and inequalities, and by bounding the values of the components of the minimizer to desired intervals. However, if the optimization problem is in the form of a least squares problem, using these functions is not recommended according to the MATLAB documentation. [13]

For nonlinear least squares problems MATLAB's Optimization Toolbox has a function called `lsqnonlin` which is preferred over other nonlinear optimization methods when solving nonlinear least squares problems. However, unlike `fmincon` and `fseminf`, `lsqnonlin` cannot handle constraints in the form of linear or nonlinear equalities or inequalities. It is still possible to use these kinds of constraints in least squares problems, which will be discussed in Chapter 3. When using `fmincon`, one can choose between two algorithms, trust-region-reflective or Levenberg-Marquardt, the former of which cannot handle underdetermined problems but constraints can be imposed on the minimizer by bounding the values of its components to desired intervals, while the latter can handle also underdetermined problems but does not handle any kinds of constraints on the minimizer. [13]

All the previously mentioned functions and the algorithms they use are iterative methods for finding a local minimizer starting from an initial guess [13]. If the objective function of an optimization problem happens to have multiple local minima, which is most likely the case in the problems we will encounter later on, the minimizers found using these methods may depend in complex ways on the initial guess [12, p. 39]. It should be mentioned that MATLAB also has a Global Optimization Toolbox, but we will prefer the local optimization methods due to lower computational costs, even though the obtained solutions are not always optimal in the global sense.

When choosing the appropriate optimization functions and algorithms in Chapters 3 and 5 we will refer to the previously mentioned properties of the MATLAB functions considered in this section. Additionally, we will consult the MATLAB documentation concerning the Optimization Toolbox [13] about choosing the optimization function, algorithm and the other optimization options wisely. Care will also be taken when choosing the initial guess for the optimization algorithm.

3. BOUNDARY ELEMENT METHOD

3.1 Basis for the Method

In boundary element methods, the function $\Phi(\mathbf{r}, \mathbf{r}_0)$ given by the formula

$$\Phi(\mathbf{r}, \mathbf{r}_0) = \frac{1}{4\pi} \ln(||\mathbf{r} - \mathbf{r}_0||^2) = \frac{1}{4\pi} \ln[(x - x_0)^2 + (y - y_0)^2] \tag{3.1}$$

is referred as the *fundamental solution of the two-dimensional Laplace's equation* [1, p. 10]. We see that the function $\Phi(\mathbf{r}, \mathbf{r}_0)$ is, up to a multiplicative constant, a suitable choice as the fundamental solution of Laplace's equation also in terms of Definition 2.1. Now, if we consider a harmonic function u as a solution of a given boundary value problem in a given region R and define $\Phi(\mathbf{r}, \mathbf{r_0})$ according to Equation (3.1), the following equation holds true [1, pp. 12, 19]

$$\lambda(\mathbf{r}_0)u(\mathbf{r}_0) = \int\limits_{\partial R} \Big(u(\mathbf{r})\frac{\partial}{\partial n}\Phi(\mathbf{r}, \mathbf{r}_0) - \Phi(\mathbf{r}, \mathbf{r}_0)\frac{\partial}{\partial n}u(\mathbf{r})\Big)\mathrm{d}s, \tag{3.2}$$

where $\lambda(\mathbf{r}_0)$ is defined as follows [1, p. 19]

$$\lambda(\mathbf{r}_0) = \begin{cases} 1 & \text{if } \mathbf{r}_0 \in R \\ 1/2 & \text{if } \mathbf{r}_0 \text{ lies on the smooth part of } \partial R \\ 0 & \text{if } \mathbf{r}_0 \notin \overline{R} \end{cases}. \tag{3.3}$$

One should recall that when $\mathbf{r}_0 \notin \overline{R}$, the fundamental solution $\Phi(\mathbf{r}, \mathbf{r}_0)$ is harmonic in $\overline{R}$, and thus, the equality given in (3.2) with $\lambda(\mathbf{r}_0) = 0$ follows from Corollary 2.4 [1, p. 12]. The other two cases can be derived using Corollary 2.4 as well [1, pp. 13–18], but for our purposes the case $\mathbf{r}_0 \notin \overline{R}$ is sufficient. After all, we are not interested in the values of u but in determining the curve $y = h(x)$.

The method of finding the curve $y = h(x)$ is the following. We choose a number of points $\mathbf{r}_i \notin \overline{R}$, from which, based on equations (3.2) and (3.3), we will obtain a set of integral equations of the form

$$\int\limits_{\partial R} \Big(u(\mathbf{r})\frac{\partial}{\partial n}\Phi(\mathbf{r}, \mathbf{r}_i) - \Phi(\mathbf{r}, \mathbf{r}_i)\frac{\partial}{\partial n}u(\mathbf{r})\Big)\mathrm{d}s = 0.$$

In these equations, some of the values of u and $\partial u/\partial n$ are unknown (see Figure 1.1), along with the parametrization of the curve $y = h(x)$. Thus, by solving the set of integral equations we will obtain estimates for the missing boundary conditions, and more importantly, a parametrization for the curve $y = h(x)$. Details for obtaining the solution are given in the following sections.

3.2 Approximation for the Line Integral

When considering the geometry of the problem presented in Figure 1.1, the line integral along the boundary of R has to be divided into four segments [5, pp. 1076–1077]. Let us denote these segments by $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ and $\mathbf{r}_4$ such that $\mathbf{r}_1$ is the line from $(b, 0)$ to $(b, h(b))$, $\mathbf{r}_2$ is the line from $(a, 0)$ to $(b, 0)$, $\mathbf{r}_3$ is the line from $(a, 0)$ to $(a, h(a))$ and $\mathbf{r}_4$ is the curve $y = h(x)$ from $(a, h(a))$ to $(b, h(b))$. These segments can be oriented at will since the values of the line integrals are not affected by the reversal of orientation [5, p. 1076]. The denotation for the boundary segments is displayed in Figure 3.1.

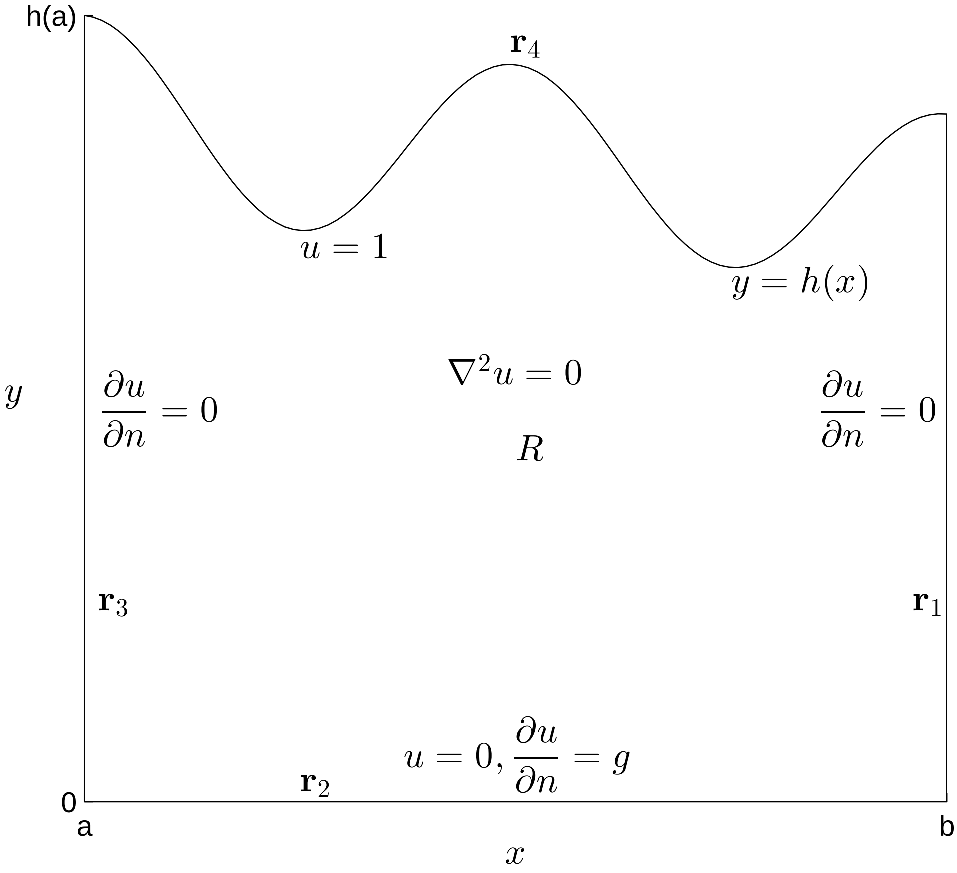
h(a)
$\mathbf{r}_4$
$u = 1$
$y = h(x)$
y
$\frac{\partial u}{\partial n} = 0$
$\nabla^2 u = 0$
R
$\frac{\partial u}{\partial n} = 0$
$\mathbf{r}_3$
$\mathbf{r}_1$
$u = 0, \frac{\partial u}{\partial n} = g$
$\mathbf{r}_2$
0
a
b
x

Figure 3.1: *Labeling for the boundary segments of ∂R.*

In order to evaluate the line integrals along the line segments, each segment $\mathbf{r}_i$ is divided into N_i linear boundary elements such that the values of u and $\partial u/\partial n$ can be approximated to be constant over each of these elements [1, pp. 19–20]. Then the value of the line integral along each segment can be approximated as a sum of the line integrals along the boundary elements [1, p. 20].

The segments $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ can clearly be parametrized by

$$\begin{aligned} \mathbf{r}_1(t) &= (b, t), \quad t \in [0, h(b)] \\ \mathbf{r}_2(t) &= (t, 0), \quad t \in [a, b] \\ \mathbf{r}_3(t) &= (a, t) \quad\ t \in [0, h(a)], \end{aligned} \tag{3.4}$$

and the boundary elements on those segments can be parametrized similarly by restricting the parameter t to a corresponding interval. Furthermore, the unit exterior normals to the segments $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$, and also to the boundary elements on those segments, are given by

$$\begin{aligned} \mathbf{n}_1(t) &= (1, 0) \\ \mathbf{n}_2(t) &= (0, -1) \\ \mathbf{n}_3(t) &= (-1, 0), \end{aligned} \tag{3.5}$$

respectively, satisfying the condition $d\mathbf{r}_i(t)/d(t) \cdot \mathbf{n}_i(t) = 0$ for every $i \in \{1, 2, 3\}$ [5, p. 861].

In the parametrization of the segment $\mathbf{r}_4$ we actually need to approximate the curve $y = h(x)$ by N_4 linear boundary elements. In order to give a proper parametrization, denote the values of x on the curve $y = h(x)$ by $x^{(0)}, x^{(1)}, \ldots, x^{(N_4)}$ such that $x^{(0)} = a$ and $x^{(N_4)} = b$. Similarly, denote the values of y by $y^{(0)}, y^{(1)}, \ldots, y^{(N_4)}$ such that $y^{(k)} = h(x^{(k)})$. Now, the points $(x^{(k-1)}, y^{(k-1)})$ and $(x^{(k)}, y^{(k)})$ are the endpoints of the kth linear boundary element of the segment $\mathbf{r}_4$, denoted by $\mathbf{r}_4^{(k)}$ [1, p. 19].

Next, determine the unit exterior normal to an element $\mathbf{r}_4^{(k)}$ by $\mathbf{n}_4^{(k)} = (n_x^{(k)}, n_y^{(k)}) = (y^{(k-1)} - y^{(k)}, x^{(k)} - x^{(k-1)})/l^{(k)}$, where $l^{(k)} = \sqrt{(x^{(k)} - x^{(k-1)})^2 + (y^{(k)} - y^{(k-1)})^2}$ is the length of $\mathbf{r}_4^{(k)}$. Now, if we parametrize $\mathbf{r}_4^{(k)}$ by

$$\mathbf{r}_4^{(k)}(t) = (x^{(k-1)} + tl^{(k)}n_y^{(k)}, y^{(k-1)} - tl^{(k)}n_x^{(k)}), \quad t \in [0, 1], \tag{3.6}$$

we see that (3.6) is the parametrization of a line from the point $(x^{(k-1)}, y^{(k-1)})$ to the point $(x^{(k)}, y^{(k)})$. Furthermore, for every element $\mathbf{r}_4^{(k)}$ we have $d\mathbf{r}_4^{(k)}(t)/dt \cdot \mathbf{n}_4^{(k)}(t) = 0$. [1, pp. 22–23]

Now that we have a parametrization for the boundary curve ∂R, we can evaluate the approximate line integrals along the line segments. However, before going into the actual integrals, one should recall that

$$\int\limits_C f(\mathbf{r})\mathrm{d}s = \int\limits_a^b f(\mathbf{r}(t)) \left|\left|\frac{d\mathbf{r}(t)}{dt}\right|\right| \mathrm{d}t, \tag{3.7}$$

where $\mathbf{r}(t)$ is a parametrization of the curve C [5, p. 1073], and note that

$$\frac{\partial}{\partial n}\Phi(\mathbf{r},\mathbf{r}_0) = \nabla\Phi(\mathbf{r},\mathbf{r}_0)\cdot\mathbf{n} = \frac{1}{2\pi}\frac{\mathbf{r}-\mathbf{r}_0}{||\mathbf{r}-\mathbf{r}_0||^2}\cdot\mathbf{n}. \tag{3.8}$$

Finally, consider the line integral along the segment $\mathbf{r}_1$. From Figure 3.1 we see that the boundary condition on the segment $\mathbf{r}_1$ is $\partial u/\partial n = 0$, so only the term containing u and $\partial\Phi/\partial n$ remains to be integrated. Furthermore, since the value of u is approximated to be constant over each boundary element, we only need to integrate

$$\begin{aligned}\int \frac{\partial}{\partial n}\Phi(\mathbf{r}_1,\mathbf{r}_0)\mathrm{d}s &= \frac{1}{2\pi}\int \frac{\mathbf{r}_1(t)-\mathbf{r}_0}{||\mathbf{r}_1(t)-\mathbf{r}_0||^2}\cdot\mathbf{n}_1(t)\left|\left|\frac{d\mathbf{r}_1(t)}{dt}\right|\right|\mathrm{d}t\\ &= \frac{1}{2\pi}\int \frac{(b-x_0,t-y_0)}{(b-x_0)^2+(t-y_0)^2}\cdot(1,0)\mathrm{d}t\\ &= \frac{1}{2\pi}\int \frac{b-x_0}{(b-x_0)^2+(t-y_0)^2}\mathrm{d}t\\ &= \frac{1}{2\pi}\arctan\left(\frac{t-y_0}{b-x_0}\right),\end{aligned} \tag{3.9}$$

where the integration can be done, e.g., using the `int` function in MATLAB. Thus, the whole integral along the segment $\mathbf{r}_1$ can be approximated by

$$\begin{aligned}&\frac{1}{2\pi}\int\limits_0^{y^{(N_4)}} u(b,t)\frac{b-x_0}{(b-x_0)^2+(t-y_0)^2}\mathrm{d}t\\ \approx\ &\sum_{k=1}^{N_1}\frac{u_b^{(k)}}{2\pi}\left[\arctan\left(\frac{t_k-y_0}{b-x_0}\right)-\arctan\left(\frac{t_{k-1}-y_0}{b-x_0}\right)\right],\end{aligned} \tag{3.10}$$

where $t_k = ky^{(N_4)}/N_1$ and $u_b^{(k)}$ is the approximated, unknown value of $u(b,t)$ on the kth element of the segment $\mathbf{r}_1$.

Moving on to the segment $\mathbf{r}_2$ where we have the boundary conditions $u = 0$ and $\partial u/\partial n = g(x)$, as shown in Figure 3.1. Again, since the value of $g(x)$ is approximated to be constant over each boundary element, the integral becomes

$$\begin{aligned}
-\int \Phi(\mathbf{r}_2,\mathbf{r}_0)\mathrm{d}s &= -\frac{1}{4\pi}\int \ln\Big(\,||\mathbf{r}_2(t)-\mathbf{r}_0||^2\,\Big)\left|\left|\frac{d\mathbf{r}_2(t)}{dt}\right|\right|\mathrm{d}t \\
&= -\frac{1}{4\pi}\int \ln\Big((t-x_0)^2+y_0^2\Big)\mathrm{d}t. \\
&= -\frac{1}{4\pi}\left[2y_0\arctan\left(\frac{t-x_0}{y_0}\right)-2t+(t-x_0)\ln\Big((t-x_0)^2+y_0^2\Big)\right],
\end{aligned} \tag{3.11}$$

where the integration can be done, e.g., using the `int` function in MATLAB. Thus, the whole integral along the segment $\mathbf{r}_2$ can be approximated by

$$\begin{aligned}
&-\frac{1}{4\pi}\int_a^b g(t)\ln[(t-x_0)^2+y_0^2]\mathrm{d}t \\
\approx\; &-\sum_{k=1}^{N_2}\frac{g^{(k)}}{4\pi}\left[2y_0\Bigg(\arctan\left[\frac{t_k-x_0}{y_0}\right]-\arctan\left[\frac{t_{k-1}-x_0}{y_0}\right]\Bigg)\right. \\
&+\ln\Big([t_k-x_0]^2+y_0^2\Big)(t_k-x_0)-2(t_k-t_{k-1}) \\
&\left.-\ln\Big([t_{k-1}-x_0]^2+y_0^2\Big)(t_{k-1}-x_0)\right],
\end{aligned} \tag{3.12}$$

where $t_k = a + k(b-a)/N_2$ and $g^{(k)} = (g(t_{k-1}) + g(t_k))/2$, i.e., the mean value of $g(x)$ at the endpoints of the element $\mathbf{r}_2^{(k)}$.

The line integral along the segment $\mathbf{r}_3$ is very similar to the one along the segment $\mathbf{r}_1$ and will not be considered in detail. Thus, as in (3.10), we obtain

$$\begin{aligned}
&\frac{1}{2\pi}\int_0^{y^{(0)}} u(a,t)\frac{x_0-a}{(a-x_0)^2+(t-y_0)^2}\mathrm{d}t \\
\approx\; &\sum_{k=1}^{N_3}\frac{u_a^{(k)}}{2\pi}\left[\arctan\left(\frac{t_{k-1}-y_0}{a-x_0}\right)-\arctan\left(\frac{t_k-y_0}{a-x_0}\right)\right],
\end{aligned} \tag{3.13}$$

where $t_k = ky^{(0)}/N_3$ and $u_a^{(k)}$ is the approximated, unknown value of $u(a,t)$ on the element $\mathbf{r}_3^{(k)}$.

It yet remains to approximate the line integral along the segment $\mathbf{r}_4$ where we have the boundary condition $u = 1$. We will split the integral into two parts and consider first the part containing u and $d\Phi(\mathbf{r},\mathbf{r}_0)/dt$. The integral along a single element $\mathbf{r}_4^{(k)}$ is the following

$$
\begin{aligned}
&\int \frac{\partial}{\partial n}\Phi(\mathbf{r}_4^{(k)}, \mathbf{r}_0)\mathrm{d}s \\
=\;& \frac{1}{2\pi}\int \frac{\mathbf{r}_4^{(k)}(t)-\mathbf{r}_0}{\left|\left|\mathbf{r}_4^{k}(t)-\mathbf{r}_0\right|\right|^2}\cdot \mathbf{n}_4(t)\left|\left|\frac{d\mathbf{r}_4^{(k)}(t)}{dt}\right|\right|\mathrm{d}t \\
=\;& \frac{1}{2\pi}\int \frac{(x^{(k-1)}+tl^{(k)}n_y^{(k)}-x_0, y^{(k-1)}-tl^{(k)}n_x^{(k)})}{(x^{(k-1)}+tl^{(k)}n_y^{(k)}-x_0)^2+(y^{(k-1)}-tl^{(k)}n_x^{(k)})^2}\cdot(n_x^{(k)}, n_y^{(k)})l^{(k)}\mathrm{d}t \\
=\;& \frac{1}{2\pi}\int \frac{(x^{(k-1)}-x_0)(y^{(k-1)}-y^{(k)})+(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})}{(x^{(k-1)}-x_0+t(x^{(k)}-x^{(k-1)}))^2+(y^{(k-1)}-y_0-t(y^{(k-1)}-y^{(k)}))^2}\mathrm{d}t,
\end{aligned}
\tag{3.14}
$$

which is of the form

$$
\frac{1}{2\pi}\int \frac{-c_1c_2+c_3c_4}{(c_1+tc_4)^2+(c_3+tc_2)^2}\mathrm{d}t = -\frac{1}{2\pi}\arctan\left(\frac{c_1c_4+c_2c_3+t(c_2^2+c_4^2)}{c_1c_2-c_3c_4}\right), \tag{3.15}
$$

where $c_1 = x^{(k-1)} - x_0$, $c_2 = y^{(k)} - y^{(k-1)}$, $c_2 = y^{(k-1)} - y_0$ and $c_4 = x^{(k)} - x^{(k-1)}$, and the integration can be done, e.g., using the `int` function in MATLAB. Finally, the whole integral can be approximated as the sum of the integrals along the elements. After substitutions we obtain

$$
\begin{aligned}
&\int\limits_{\mathbf{r}_4} \frac{\partial}{\partial n}\Phi(\mathbf{r}, \mathbf{r}_0)\mathrm{d}s \\
\approx\;& \frac{1}{2\pi}\sum_{k=1}^{N_4}\int\limits_0^1 \frac{(x^{(k-1)}-x_0)(y^{(k-1)}-y^{(k)})+(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})}{(x^{(k-1)}-x_0+t(x^{(k)}-x^{(k-1)}))^2+(y^{(k-1)}-y_0-t(y^{(k-1)}-y^{(k)}))^2}\mathrm{d}t \\
=\;& \frac{1}{2\pi}\sum_{k=1}^{N_4}\left[\arctan\left(\frac{(x^{(k-1)}-x_0)(x^{(k)}-x^{(k-1)})+(y^{(k-1)}-y_0)(y^{(k)}-y^{(k-1)})}{(x^{(k-1)}-x_0)(y^{(k)}-y^{(k-1)})-(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})}\right)\right. \\
&\left.-\arctan\left(\frac{(x^{(k)}-x_0)(x^{(k)}-x^{(k-1)})+(y^{(k)}-y_0)(y^{(k)}-y^{(k-1)})}{(x^{(k-1)}-x_0)(y^{(k)}-y^{(k-1)})-(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})}\right)\right],
\end{aligned}
\tag{3.16}
$$

where the points $(x^{(k)}, y^{(k)})$ are on the curve $y = h(x)$.

For the second part of the integral, the value of $\partial u/\partial n$ is approximated to be constant over each element $\mathbf{r}_4^{(k)}$. The integrand along a single element $\mathbf{r}_4^{(k)}$ is then

$$
\begin{aligned}
&-\int \Phi(\mathbf{r}_4^{(k)}, \mathbf{r}_0)\mathrm{d}s \\
=\;& -\frac{1}{4\pi}\int \ln\left(\left|\left|\mathbf{r}_4^{(k)}(t)-\mathbf{r}_0\right|\right|^2\right)\left|\left|\frac{d\mathbf{r}_4^{(k)}}{dt}\right|\right|\mathrm{d}t \\
=\;& -\frac{l^{(k)}}{4\pi}\int \ln\left([x^{(k)}-x_0+t(x^{(k)}-x^{(k-1)})]^2+[y^{(k)}-y_0+t(y^{(k)}-y^{(k-1)})]^2\right),
\end{aligned}
\tag{3.17}
$$

which is of the form

$$\begin{aligned}
&-\frac{l^{(k)}}{4\pi}\int \ln(c_2t^2+2c_1t+c_0)\mathrm{d}t\\
= \;&-\frac{l^{(k)}}{4\pi}\Bigg[t\Big(\ln(c_2t^2+2c_1t+c_0)-2\Big)\\
&+2\arctan\left(\frac{c_1+c_2t}{\sqrt{c_2c_0-c_1^2}}\right)\frac{\sqrt{c_2c_0-c_1^2}}{c_2}+\frac{c_1}{c_2}\ln(c_2t^2+c_1t+c_0)\Bigg],
\end{aligned} \tag{3.18}$$

where $c_2=(x^{(k)}-x^{(k-1)})^2+(y^{(k)}-y^{(k-1)})^2$, $c_1=(x^{(k-1)}-x_0)(x^{(k)}-x^{(k-1)})+(y^{(k-1)}-y_0)(y^{(k)}-y^{(k-1)})$ and $c_0=(x^{(k-1)}-x_0)^2+(y^{(k-1)}-y_0)^2$, and the integration can be done, e.g., using the `int` function in MATLAB. Finally, the whole integral can be approximated as the sum of the integrals along the boundary elements. After substitutions we obtain

$$\begin{aligned}
&\int\limits_{\mathbf{r}_4} -\Phi(\mathbf{r},\mathbf{r}_0)\frac{\partial}{\partial n}u(\mathbf{r})\mathrm{d}s\\
\approx\;& -\frac{1}{4\pi}\sum_{k=1}^{N_4}\int\limits_0^1\left(\frac{\partial u}{\partial x}(y^{(k-1)}-y^{(k)})+\frac{\partial u}{\partial y}(x^{(k)}-x^{(k-1)})\right)\\
&\times\ln\Big([x^{(k-1)}-x_0+t(x^{(k)}-x^{(k-1)})]^2+[y^{(k-1)}-y_0+t(y^{(k)}-y^{(k-1)})]^2\Big)\mathrm{d}t\\
=\;& -\frac{1}{4\pi}\sum_{k=1}^{N_4}\Bigg(\Bigg[\left(\frac{\partial u}{\partial x}\right)^{(k)}(y^{(k-1)}-y^{(k)})+\left(\frac{\partial u}{\partial y}\right)^{(k)}(x^{(k)}-x^{(k-1)})\Bigg]\\
&\times\Bigg[\ln\Big((x^{(k)}-x_0)^2+(y^{(k)}-y_0)^2\Big)-2+\frac{1}{(x^{(k)}-x^{(k-1)})^2+(y^{(k)}-y^{(k-1)})^2}\\
&\times\Bigg(2\Bigg[\arctan\Bigg(\frac{(x^{(k-1)}-x_0)(x^{(k)}-x^{(k-1)})+(x^{(k)}-x^{(k-1)})^2}{|(x^{(k-1)}-x_0)(y^{(k)}-y^{(k-1)})-(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})|}\\
&+\frac{(y^{(k-1)}-y_0)(y^{(k)}-y^{(k-1)})+(y^{(k)}-y^{(k-1)})^2}{|(x^{(k-1)}-x_0)(y^{(k)}-y^{(k-1)})-(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})|}\Bigg)\\
&-\arctan\left(\frac{(x^{(k-1)}-x_0)(x^{(k)}-x^{(k-1)})+(y^{(k-1)}-y_0)(y^{(k)}-y^{(k-1)})}{|(x^{(k-1)}-x_0)(y^{(k)}-y^{(k-1)})-(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})|}\right)\Bigg]\\
&\times\Big|(x^{(k-1)}-x_0)(y^{(k)}-y^{(k-1)})-(y^{(k-1)}-y_0)(x^{(k)}-x^{(k-1)})\Big|\\
&+\Big[(x^{(k-1)}-x_0)(x^{(k)}-x^{(k-1)})+(y^{(k-1)}-y_0)(y^{(k)}-y^{(k-1)})\Big]\\
&\times\Big[\ln\Big((x^{(k)}-x_0)^2+(y^{(k)}-y_0)^2\Big)-\ln\Big((x^{(k-1)}-x_0)^2+(y^{(k-1)}-y_0)^2\Big)\Big]\Bigg)\Bigg]\Bigg),
\end{aligned} \tag{3.19}$$

where the points $(x^{(k)},y^{(k)})$ are on the curve $y=h(x)$ and $(\partial u/\partial x)^{(k)}$ and $(\partial u/\partial y)^{(k)}$ are the approximate, unknown values of $\partial u(\mathbf{r}_4^{(k)}(t))/\partial x$ and $\partial u(\mathbf{r}_4^{(k)}(t))/\partial y$ on the element $\mathbf{r}_4^{(k)}$.

Now that we have obtained approximations for the line integrals along the segments $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ and $\mathbf{r}_4$, the line integral along ∂R is obtained by summing the integrals along the individual segments, i.e., by summing the approximations obtained in Equations (3.10), (3.12), (3.13), (3.16) and (3.19). For the sake of convenience the approximation for the whole integral along ∂R is not written here.

3.3 Solving the Set of Discretized Integral Equations

Now that we have means of approximating the line integral along ∂R, it remains to choose how we are going to solve the set of obtained, discretized integral equations. Basically, since we end up with a set of nonlinear equations of the form given in Equation (3.1) with different points $\mathbf{r}_i \notin \overline{R}$, one possibility would be simply to solve the set of equations, e.g., with MATLAB's `fsolve` function. However, it turned out that `fsolve` rarely manages to find a solution to our problem, and even when it does, the computational costs are unreasonably high. Additionally, when using `fsolve` it is not possible to take into account the different constraints the solution should satisfy. We will consider these constraints later on.

Since simply solving the set of integral equations is essentially out of question, another possibility is to approach the set of equations as an optimization problem. Consider the vector valued function

$$\mathbf{F}(\mathbf{r}_i, \mathbf{x}) = \int\limits_{\partial R} \left(u(\mathbf{r}) \frac{\partial}{\partial n} \Phi(\mathbf{r}, \mathbf{r}_i) - \Phi(\mathbf{r}, \mathbf{r}_i) \frac{\partial}{\partial n} u(\mathbf{r}) \right) \mathrm{d}s, \tag{3.20}$$

where $\mathbf{r}_i \notin \overline{R}$ are the selected points and the vector $\mathbf{x}$ stands for all the unknown variables in the approximated line integral. Since every element of $\mathbf{F}(\mathbf{r}_i, \mathbf{x})$ should be zero, a solution of the optimization problem

$$\min_{\mathbf{x}} ||\mathbf{F}(\mathbf{r}_i, \mathbf{x})|| \,, \tag{3.21}$$

where the norm is the euclidian norm, or

$$\min_{\mathbf{x}} ||\mathbf{F}(\mathbf{r}_i, \mathbf{x})||^2 \tag{3.22}$$

should also be close to a solution of the set of nonlinear equations $\mathbf{F}(\mathbf{r}_i, \mathbf{x}) = \mathbf{0}$.

If we choose the optimization problem given by Equation (3.21), we should choose either `fmincon` or `fseminf` from MATLAB's Optimization Toolbox as discussed in Section 2.6. On the other hand, if we choose the nonlinear least squares problem given by Equation (3.22), the preferred choice as the optimization function would then be `lsqnonlin`. Since we will also need to impose linear inequality constraints on the certain components of the minimizer, it would seem to be wise to choose the optimization problem of Equation (3.21) and choose either `fmincon` or `fseminf`, which could handle these constraints. However, numerical tests have shown that these functions rarely manage to find a minimizer in reasonable computation time, and thus, we will rather choose the nonlinear least squares problem of Equation (3.22) and use `lsqnonlin` which is at least computationally much more efficient.

Since we can choose arbitrarily many points $\mathbf{r}_i \notin \overline{R}$, the least squares problem given by Equation (3.22) can be made overdetermined so that we can freely choose between the trust-region-reflective and the Levenberg-Marquardt algorithm as discussed in Section 2.6. However, since we are willing to bound the values of the certain components of the minimizer to certain intervals, we should choose the trust-region-reflective algorithm which can handle these kinds of constraints (it is the default algorithm `lsqnonlin` uses in MATLAB Release 2013b [13]). In order to take into account the linear inequality constraints the minimizer should satisfy we will add regularization terms to the objective function $||\mathbf{F}(\mathbf{r}_i, \mathbf{x})||^2$ such that the regularization terms obtain relatively large values if a solution candidate $\mathbf{x}$ possesses unwanted properties and, conversely, that the regularization terms are rather small if the solution candidate satisfies the given constraints.

Let us now consider the constraints on the minimizer. From the properties of Laplace's equation we know that the values of the potential u should be strictly increasing along the lines $x = a$ and $x = b$ as functions of y. Thus, by adding a regularization term of the form

$$T_2(\lambda_2, \lambda_3, u_b) = \lambda_2 \sum_{k=1}^{N_1-1} \Big(\exp[-\lambda_3(u_b^{(k+1)} - u_b^{(k)})] \Big)^2, \tag{3.23}$$

to the objective function, where $\lambda_2 > 0$ and $\lambda_3 > 0$ are regularization parameters, we can control the values of u on the line $x = b$. Completely analogous term regularizes the values of u on the line $x = a$, namely

$$T_3(\lambda_4, \lambda_5, u_a) = \lambda_4 \sum_{k=1}^{N_3-1} \Big(\exp[-\lambda_5(u_a^{(k+1)} - u_a^{(k)})] \Big)^2, \tag{3.24}$$

where similarly $\lambda_4 > 0$ and $\lambda_5 > 0$. The values for the regularization parameters are chosen such that the regularization terms obtain suitable values with respect to the objective function $||\mathbf{F}(\mathbf{r}_i, \mathbf{x})||^2$. In the MATLAB implementation of this method we use $\lambda_2 = \lambda_4 = 0.01$, $\lambda_3 = N_1$ and $\lambda_5 = N_3$.

There should be yet one constraint on the values of the normal derivative of u on the boundary of the region R. Since u should be a harmonic function, we saw in Corollary 2.5 that the line integral of the normal derivative of u along the boundary of R should be zero. Since the normal derivative of u is already zero on the lines

$x = a$ and $x = b$, the constraint becomes

$$\int\limits_{\mathbf{r}_2} \frac{\partial}{\partial n} u(x,y) \mathrm{d}s(x,y) + \int\limits_{\mathbf{r}_4} \frac{\partial}{\partial n} u(x,y) \mathrm{d}s(x,y) = 0. \tag{3.25}$$

Furthermore, since the value of $\partial u/\partial n$ is approximated to be constant over boundary elements $\mathbf{r}_2^{(k)}$ and $\mathbf{r}_4^{(k)}$, the integrals along the segments $\mathbf{r}_2$ and $\mathbf{r}_4$ can be approximated as sums of the integrals over the boundary elements $\mathbf{r}_2^{(k)}$ and $\mathbf{r}_4^{(k)}$, respectively. We obtain

$$\int\limits_{\mathbf{r}_2} \frac{\partial}{\partial n} u(x,y) \mathrm{d}s(x,y) \approx \frac{b-a}{N_2} \sum_{k=1}^{N_2} g^{(k)}, \tag{3.26}$$

where $(b-a)/N_2$ is the length of each of the boundary elements $\mathbf{r}_2^{(k)}$, and

$$\int\limits_{\mathbf{r}_4} \frac{\partial}{\partial n} u(x,y) \mathrm{d}s(x,y) \approx \sum_{k=1}^{N_4} \left[\left(\frac{\partial u}{\partial x} \right)^{(k)} (y^{(k-1)} - y^{(k)}) + \left(\frac{\partial u}{\partial y} \right)^{(k)} (x^{(k)} - x^{(k-1)}) \right]. \tag{3.27}$$

Thus, in the least squares sense, the regularization term for the normal derivative of u is then

$$\begin{aligned} &T_1 \left(\lambda_1, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) \\ = \; &\lambda_1 \left(\frac{b-a}{N_2} \sum_{k=1}^{N_2} g^{(k)} + \sum_{k=1}^{N_4} \left[\left(\frac{\partial u}{\partial x} \right)^{(k)} (y^{(k-1)} - y^{(k)}) + \left(\frac{\partial u}{\partial y} \right)^{(k)} (x^{(k)} - x^{(k-1)}) \right] \right)^2, \end{aligned} \tag{3.28}$$

where $\lambda_1 > 0$ is a regularization parameter. In the MATLAB implementation we use $\lambda_1 = 1$.

Now that we have considered the constraints arising from the harmonic properties of u, we may present the regularized minimization problem as follows

$$\min_{\mathbf{x}} \left[||\mathbf{F}(\mathbf{r}_i, \mathbf{x})||^2 + T_1 \left(\lambda_1, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) + T_2(\lambda_2, \lambda_3, u_b) + T_3(\lambda_4, \lambda_5, u_a) \right], \tag{3.29}$$

where the elements of x, y, $\partial u/\partial x$, $\partial u/\partial y$, u_a and u_b correspond to the certain elements of the vector $\mathbf{x}$. In the MATLAB implementation the placement of the unknown variables in the vector $\mathbf{x}$ is the following

$$\begin{aligned} x_{k+1}:&\ \ y^{(k)}, && k\in\{0,2,\ldots,N_4\},\\ x_{N_4+1+k}:&\ \ u_b^{(k)}, && k\in\{1,2,\ldots,N_1\},\\ x_{N_4+N_1+1+k}:&\ \ u_a^{(k)} && k\in\{1,2,\ldots,N_3\},\\ x_{N_4+N_1+N_3+1+k}:&\ \ \left(\frac{\partial u}{\partial x}\right)^{(k)}, && k\in\{1,2,\ldots,N_4\},\\ x_{2N_4+N_1+N_3+1+k}:&\ \ \left(\frac{\partial u}{\partial y}\right)^{(k)}, && k\in\{1,2,\ldots,N_4\}, \end{aligned} \tag{3.30}$$

and the values of x on the segment $\mathbf{r}_4$ are preassigned by $x^{(k)} = a + k(b-a)/N_4$ with $k \in \{0,1,\ldots,N_4\}$, since for determining the curve $y = h(x)$ it is sufficient to consider only the values of $y^{(k)}$ at given points $x^{(k)}$.

In addition to the different constraints we may limit the possible values of certain unknown variables. From the geometry of the problem it is for example clear that all the values of y on the segment $\mathbf{r}_4$ should be positive. We also know that the values of u should be on the interval $[0,1]$. For the values of the partial derivatives of u on the segment $\mathbf{r}_4$ there are no clear limitations, but their values are at least somewhat restricted by the regularization term given in Equation (3.28).

As mentioned in Section 2.6, `lsqnonlin` yet needs an initial guess to start from. In order to find a reasonable initial guess, we solve the boundary value problem in a rectangle $x \in [a,b]$, $y \in [0,0.3]$, the solution of which is $u(x,y) = (10/3)y$, from which we also obtain the values for the partial derivatives of u on the segment $\mathbf{r}_4$. Thus, the initial guess in the MATLAB implementation is

$$\begin{aligned} y^{(k)} &= 0.3, && k\in\{0,1,\ldots,N_4\},\\ u_b^{(k)} &= \frac{1+2(k-1)}{2N_1}, && k\in\{1,2,\ldots,N_1\},\\ u_a^{(k)} &= \frac{1+2(k-1)}{2N_3}, && k\in\{1,2,\ldots,N_3\},\\ \left(\frac{\partial u}{\partial x}\right)^{(k)} &= 0, && k\in\{1,2,\ldots,N_4\},\\ \left(\frac{\partial u}{\partial y}\right)^{(k)} &= \frac{10}{3}, && k\in\{1,2,\ldots,N_4\}. \end{aligned} \tag{3.31}$$

It should be also noted that the points $\mathbf{r}_0 \notin R$ are chosen to be of the form $((k\pi)/300, -1)$, where $k \in \{1,2,\ldots,299\}$. Apart from the requirement $\mathbf{r}_0 \notin R$ the points are chosen rather arbitrarily since the locations of the points $\mathbf{r}_0$ should not otherwise affect the minimization procedure.

Now that we have considered all the necessary aspects of the method, we end this chapter by noting that a MATLAB implementation `ibem` of the method can be found in Appendix A. It should be noted that the `ibem` function simply returns the points (x, y) that reconstruct the boundary curve $y = h(x)$ based on the given data $g(x)$, while it does not actually check that the minimizer satisfies the given constraints or that a minimizer is even found. However, when the performance of the implemented method `ibem` is tested in Chapter 6, we will check that the obtained solutions to the test cases satisfy the constraints considered in this section and that the minimization procedure is successfully carried out in each of the cases.

4. ALTERNATING-FIELD TECHNIQUE

4.1 The Method in Outline

The basic idea of the alternating-field technique was originally introduced by Nilson and Tsuei in [14] where the technique is applied to a free boundary problem of electrochemical machining. The method presented in this chapter follows schematically the ideas presented in [14], but the actual alternating-field technique described in Section 4.4 is somewhat modified from the original one [14]. The method described in this chapter has been presented in the Fourth Finnish-Estonian Mathematical Colloquium [8] by the author and a paper has been written about the method for the 18th European Conference on Mathematics for Industry [9] by the author and the examiners of this work.

In this chapter we will consider the harmonic potential u as the real part of a holomorphic function $f(z) = u(x, y) + iv(x, y)$. This is justified since the region R of our problem is simply connected, and we saw in Section 2.3 that in such cases the harmonic potential u always has a harmonic conjugate v such that the function $f = u + iv$ is holomorphic.

After we determine v, the harmonic conjugate for u, and have formed the function $f(z) = u(x, y) + iv(x, y)$, we will consider the problem on the region $f(R)$. There we determine the inverse f^{-1} such that $z = f^{-1}[f(z)]$. As discussed in Section 2.3, if we assume that $df(z)/dz \neq 0$ at each point in R, then f^{-1} is holomorphic in $f(R)$ and we may write $f^{-1}(w) = x(u, v) + iy(u, v)$, where $w = u + iv$ and x and y are harmonic conjugates in $f(R)$.

We end up having boundary value problems for x and y on the region $f(R)$ which is a rectangle in the w-plane. We will also refer to the w-plane as the *inverted plane*. Instead of an unknown boundary curve we have unknown boundary conditions on a single boundary segment in the problems for x and y. The boundary segment with the unknown boundary conditions corresponds to the unknown boundary $y = h(x)$ in the original problem for u on R.

Finally, the unknown boundary conditions for x and y are determined using the alternating-field technique. The technique is an iterative procedure where, starting from an initial guess, we find convergent estimates for the unknown boundary conditions by solving Laplace's equation, by turns, for x and y. When the unknown boundary conditions are determined, we directly obtain the points (x, y) constructing the boundary curve $y = h(x)$. We will consider the different phases of this method in detail in the following sections. [14, pp. 271–272]

4.2 Finding a Harmonic Conjugate for the Potential Function

As we already established in Section 2.3, since the region R of our problem is simply connected, the harmonic potential u has a harmonic conjugate v in R such that the function $f(z) = u(x, y) + iv(x, y)$ is holomorphic in R. However, since we only have restricted knowledge about u in R, we cannot actually use the formula given in Equation 2.27 to determine the harmonic conjugate v. Fortunately, we will obtain adequate information about v solely based on the given boundary conditions for u using the Cauchy-Riemann equations (2.18).

Consider the Neumann boundary conditions $\partial u/\partial n = 0$ on the lines $x = a$ and $x = b$ (see Fig. 1.1). Since the unit exterior normals to those lines are $(-1, 0)$ and $(1, 0)$, respectively, the boundary conditions then imply

$$0 = \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \tag{4.1}$$

i.e., v is constant on the lines $x = a$ and $x = b$. Furthermore, we may apply the Cauchy-Riemann equations to the Neumann boundary condition $\partial u/\partial n = g(x)$ on the line $y = 0$. Since the unit exterior normal to the line $y = 0$ is $(0, -1)$, we obtain

$$g(x) = -\frac{\partial u}{\partial y} = \frac{\partial v}{\partial x}, \tag{4.2}$$

and thus, we may determine the change in the value of v along the line $y = 0$ by integrating $g(x)$. More importantly, since the v is constant on the lines $x = a$ and $x = b$, we may determine the difference in the values of v between those lines by computing $\int\limits_a^b g(x)\mathrm{d}x$. Furthermore, since we may freely add a real constant to v as noted in Section 2.3, we may actually determine $v(x = b) = 0$ and $v(x = a) = -\int\limits_a^b g(x)\mathrm{d}x$. Specifically, we obtain the value of v at any point on the line $y = 0$ from

$$v(x) = -\int\limits_x^b g(t)\mathrm{d}t, \quad x \in [a, b]. \tag{4.3}$$

We are yet to consider the Dirichlet boundary conditions $u = 0$ and $u = 1$ on the line $y = 0$ and on the curve $y = h(x)$, respectively. From the former boundary condition we obtain

$$0 = -\frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y}, \tag{4.4}$$

i.e., $\partial v/\partial n = 0$ on the line $u = 0$, and using analogous reasoning on the latter boundary condition we also obtain $\partial v/\partial n = 0$ on the curve $y = h(x)$. Thus, we have now determined boundary conditions for v corresponding to each of the given boundary conditions for u, from which we obtain the necessary information in order to move on to the next phase.

4.3 Mapping the Problem to the Inverted Plane

The obtained boundary conditions for v along with the given boundary conditions for u and the fact that u and v are harmonic conjugates in R give us a rather good insight into the behavior of the holomorphic function $f(z) = u(x, y) + iv(x, y)$ on the region R. Based on that insight we note that $f(z)$ actually is invertible, and hence the function $f^{-1}(w) = x(u, v) + iy(u, v)$ exists in $f(R)$. Based on the same insight we also note that at each point in R at least one of the partial derivatives of u (and v as well) is nonzero, implying that (See Equations (2.13) and (2.14)) $df(z)/dz \neq 0$ holds at each point in R and thus, the inverse $f^{-1}(w)$ is holomorphic in $f(R)$, as discussed in Section 2.3. However, while the assumption $df(z)/dz \neq 0$ does hold in R, it turns out in the test cases in Chapter 6 that the assumption does not hold in general at certain points on ∂R, which causes errors to the free boundary reconstructions.

Regardless, now that the function $f^{-1}(w) = x(u, v) + iy(u, v)$ is known to be holomorphic in $f(R)$, we may finally consider the problem in the inverted plane, i.e., in the w-plane. The region $f(R)$, denoted by Q in future, is a rectangle $u \in [0, 1]$, $v \in [0, V]$, where we denote $V = \int_a^b g(x)\mathrm{d}x$. The limits for u and v are obtained form their boundary conditions in the original problem on the region R. The problems for x and y on Q are displayed in Figure 4.1, where also the boundary conditions for x and y are given. The boundary conditions for x and y are considered next.

The Dirichlet boundary conditions for x on the lines $v = 0$ and $v = V$ seen in Figure 4.1 are obtained directly from the geometry of the original problem on the region R, since the boundary conditions for v on the lines $x = a$ and $x = b$ are exactly $v = V$ and $v = 0$, respectively. Similarly we obtain the boundary condition $y = 0$ on the line $u = 0$. Now we can again utilize the Cauchy-Riemann equations (2.30) to obtain the Neumann boundary conditions, for on the lines $v = 0$ and $v = V$ we have

$$0 = \frac{\partial x}{\partial u} = \frac{\partial y}{\partial v}, \tag{4.5}$$

from which we obtain $\partial y/\partial n = 0$ on the lines $v = 0$ and $v = V$. Similarly from the line $u = 0$ we obtain

$$0 = -\frac{\partial y}{\partial v} = -\frac{\partial x}{\partial u}, \tag{4.6}$$

i.e., $\partial x/\partial n = 0$ on the line $u = 0$.

The additional Dirichlet boundary condition $x = x(v)$ on the line $u = 0$ is the

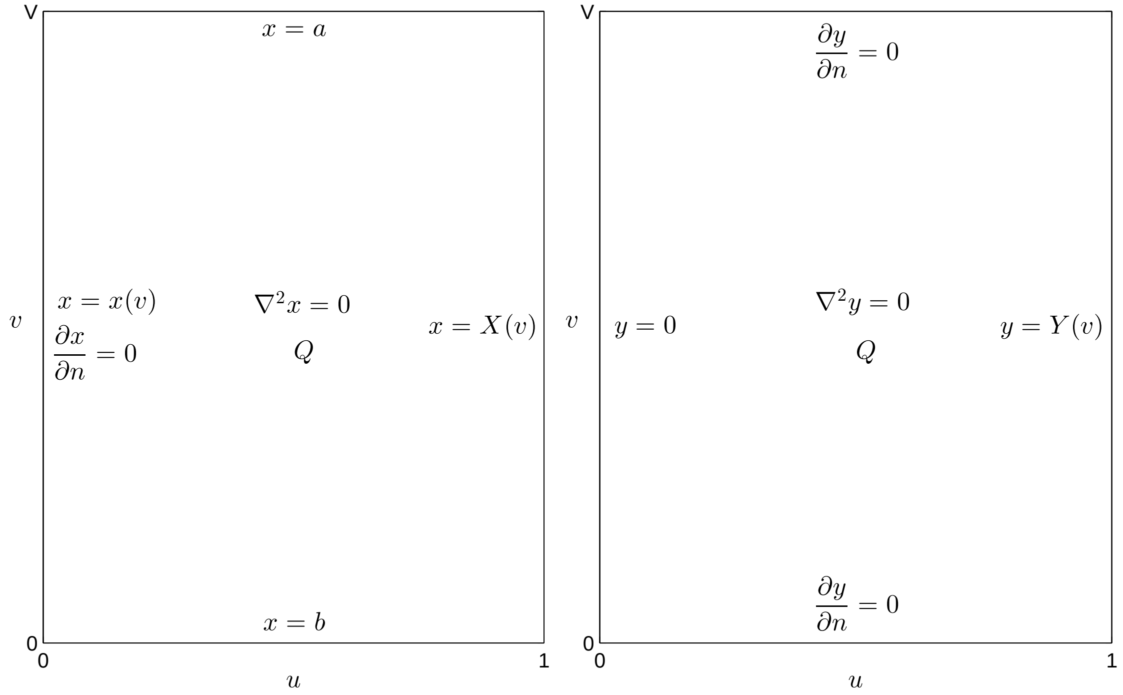

Figure 4.1: *The problems for x and y on the region Q.*

inverse of $v = v(x)$ given by Equation (4.3). In the implementation of this method, we assume knowing the values of the function $g(x)$ at some discrete points x_i, from which we may compute values for the function $v(x)$ at the points x_i using Equation (4.3) and the trapezoidal rule (2.42). Then we can use interpolation on the data $(x_i, v(x_i))$ to determine, which values of x result in the certain values of $v(x)$, from which we obtain the values of the function $x = x(v)$ at certain points v_j. In the implementation we use second degree polynomials as described in Section 2.5, where for each v_j we choose the three closest points $v(x_i)$ to it as interpolation points in order to approximate the value of $x(v_j)$. A MATLAB implementation `qinterp` of the interpolation procedure can be found in Appendix B.

There are still the Dirichlet boundary conditions $x = X(v)$ and $y = Y(v)$ on the line $u = 1$. These boundary conditions are unknown, since the boundary $u = 1$ corresponds to the unknown boundary $y = h(x)$ in the original problem for u. The unknown boundary conditions are determined using the alternating-field technique which is described in the next section. When the boundary conditions are determined, we obtain a set of points $(X(v), Y(v))$ with different values of v, and those are exactly the points that reconstruct the boundary curve $y = h(x)$ in the original problem on the region R.

4.4 Alternating-Field Technique on the Inverted Plane

For the alternating-field technique, the region Q is discretized by an $M \times N$ rectangular mesh, where $\Delta u = 1/(N-1)$ and $\Delta v = V/(M-1)$. The mesh points are denoted by (u_j, v_i) such that $u_1 = 0$, $u_N = 1$, $v_1 = V$ and $v_M = 0$, and the value of x (resp. y) at a point (u_j, v_i) is denoted by x_{ij} (resp. y_{ij}). Laplace's equation for x and y is solved as a set of linear equations, where the second order partial derivatives of x and y are approximated using Equation (2.40) and the Neumann boundary conditions are approximated using the backward (2.39b) or the forward difference approximation (2.39a), depending on the direction of the unit exterior normal on the corresponding boundaries. [14, p. 271]

The steps for the alternating-field technique are described below. Some steps are altered from the original presentation of Nilson and Tsuei in [14] due to differences in the problem geometries. As already mentioned in Section 4.1, in the alternating-field technique we find convergent estimates for the unknown boundary conditions $X(v)$ and $Y(v)$ by solving Laplace's equation, by turns, for x and y. The convergence of the estimates for $X(v)$ and $Y(v)$ is measured by the change in the arc length parameter s given by

$$s_i = \sum_{k=1}^{i} \sqrt{[X(v_{k+1}) - X(v_k)]^2 + [Y(v_{k+1}) - Y(v_k)]^2}, \tag{4.7}$$

i.e., s is the arc length parameter of the unknown boundary curve $y = h(x)$. [14, pp. 271–272]

The steps for the alternating-field technique are the following. [14, pp. 271–272]

0. Make an initial guess for $X(v)$. Note that $X(v_i) \in [a, b]$ for every $i \in \{1, 2, \ldots, M\}$ and that $X(v_1) = a$ and $X(v_M) = b$. Then perform the steps 1–6 in order to obtain the first iterates for $X(v)$ and $Y(v)$. When the arc length parameter s^* is computed in step 6, set $s = s^*$ and begin the procedure from step 1.

1. Assign boundary conditions for $x(u, v)$ field, i.e., set

$$\begin{aligned} &x_{1j} = a, \forall j \in \{1, 2, \ldots, N\}, && x_{Mj} = b, \forall j \in \{1, 2, \ldots, N\}, \\ &x_{i1} = x(v_i), \forall i \in \{1, 2, \ldots, M\} && x_{i2} = x_{i1}, \forall i \in \{1, 2, \ldots, M\}, \\ &x_{iN} = X(v_i), \forall i \in \{1, 2, \ldots, M\}. \end{aligned} \tag{4.8}$$

2. Solve Laplace's equation for x in Q.

3. Calculate new $Y(v)$ by the formula

$$\begin{aligned} Y(v_i) &= -\int_0^1 \frac{\partial x_{ij}}{\partial v}\mathrm{d}u, \quad && i \in \{2,3,\ldots,M-1\}, \\ Y(v_1) &= Y(v_2), && Y(v_M) = Y(v_{M-1}), \end{aligned} \tag{4.9}$$

   where $\partial x_{ij}/\partial v$ is approximated by the central difference formula (2.39c), i.e.,

$$\frac{\partial x_{ij}}{\partial v} \approx \frac{x_{(i-1)j} - x_{(i+1)j}}{2\Delta v} \tag{4.10}$$

   and the integral is evaluated using Simpson's rule (2.43).

4. Assign boundary conditions for $y(u,v)$ field, i.e., set

$$\begin{aligned} &y_{1j} = y_{2j}, \forall j \in \{1,2,\ldots,N\}, \quad && y_{Mj} = y_{(M-1)j}, \forall j \in \{1,2,\ldots,N\}, \\ &y_{i1} = 0, \forall i \in \{1,2,\ldots,M\}, && y_{iN} = Y(v_i), \forall i \in \{1,2,\ldots,M\}, \end{aligned} \tag{4.11}$$

   where $Y(v_i)$ is given by Equation (4.9).

5. Solve Laplace's equation for y in Q.

6. Calculate new $X(v)$ by the formula

$$\begin{aligned} X(v_i) &= x_{i1} + \int_0^1 \frac{\partial y_{ij}}{\partial v}\mathrm{d}u, \quad && i \in \{2,3,\ldots,M-1\}, \\ X(v_1) &= a, && X(v_M) = b, \end{aligned} \tag{4.12}$$

   where $\partial y_{ij}/\partial v$ is approximated by the central difference formula (2.39c), i.e.,

$$\frac{\partial y_{ij}}{\partial v} \approx \frac{y_{(i-1)j} - y_{(i+1)j}}{2\Delta v} \tag{4.13}$$

   and the integral is evaluated using Simpson's rule (2.43). Then calculate a new arc length parameter s^* from the newly obtained $X(v)$ and $Y(v)$ using Equation (4.7).

7. Check convergence for s, i.e., calculate $||s^* - s||^2$. If necessary, set $s = s^*$ and return to step 1. A new estimate for $X(v)$ is given by Equation (4.12).

Note that in Equations (4.9) and (4.12) we use the Cauchy-Riemann equations in order to obtain to the change in x and y between the lines $u = 0$ and $u = 1$. The chosen way of using the Cauchy-Riemann equations is different from the one in [14] because the way they are used in Equations (4.9) and (4.12) was found out to be more suitable for the boundary conditions of our problem. The usual stopping criterion for the procedure is to check, whether $||s^* - s||^2$ is sufficiently small. Other possibility would be to inspect the changes in $X(v)$ and $Y(v)$ individually and determine a stopping criterion based on them [14, p. 272].

A MATLAB implementation `aft` of the presented method is given in Appendix B, where the stopping criterion is chosen to be $||s^* - s||^2 <$ `tol`. The performance of the implemented method will be tested in Chapter 6, for which we will choose `tol` $= 10^{-10}$. One should note that in the implementation the initial guess for the unknown boundary condition $X(v)$ is chosen to be $x(v)$, i.e., the Dirichlet boundary condition on the line $u = 0$. Furthermore, one should note that no regularization is needed since the arc length parameter s is the only free parameter in the iterative procedure.

5. METHOD OF FUNDAMENTAL SOLUTIONS

5.1 Basis for the Method

The basic idea of the method of fundamental solutions is to find an approximate solution $u_N(\mathbf{r})$ to a boundary value problem on a region $\overline{R}$ as a linear combination of fundamental solutions of Laplace's equation, i.e.,

$$u_N(\mathbf{r}) = \sum_{j=1}^{N} c_j \Phi(\mathbf{r}, \mathbf{s}_j), \quad \mathbf{r} \in \overline{R}, \tag{5.1}$$

where N is the number of fundamental solutions used in the approximation. In the two dimensional case, a suitable choice for the fundamental solution $\Phi(\mathbf{r}, \mathbf{r}_0)$ is given by [2, p. 386]

$$\Phi(\mathbf{r}, \mathbf{r_0}) = -\ln(||\mathbf{r} - \mathbf{r_0}||) = -\frac{1}{2}\ln[(x - x_0)^2 + (y - y_0)^2], \tag{5.2}$$

which is also in accordance with Definition 2.1.

Note that the fundamental solution $\Phi(\mathbf{r}, \mathbf{r}_0)$ is singular at the point $\mathbf{r}_0$, and therefore the point $\mathbf{r}_0$ is called the *singularity of the fundamental solution*, a *singularity* for short. It then follows that the approximated solution u_N is singular at each point $\mathbf{s}_j$ and thus, in order to make u_N harmonic in R, the singularities $\{\mathbf{s}_j\}_{j=1}^N$ should be located in $\mathbb{R}^2 \setminus \overline{R}$. The exact locations of the singularities are either preassigned or determined along with the coefficients $\{c_j\}_{j=1}^N$ of the fundamental solutions so that the approximate solution u_N satisfies the given boundary conditions as well as possible. Usually the determination of the unknown variables is carried out by least squares fit of the boundary data. The resulting optimization problem is either linear or nonlinear depending on whether the locations of the singularities are preassigned or not, respectively. [7, p. 70]

5.2 Problem Formulation

When it comes to free boundary problems and the method of fundamental solutions, one should, along with the coefficients $\{c_j\}_{j=1}^N$ and usually the locations $\{\mathbf{s}_j\}_{j=1}^N$, determine the location of the free boundary as part of the procedure of finding the approximate solution u_N. One idea of locating the free boundary is to include a parametrization of the boundary curve to the procedure of finding u_N. In the least squares sense this would mean adding a term of the form $\sum_{k=1}^{M_{fr}}(u_N(\mathbf{r}_k)-1)^2$ to the functional to be minimized, where u_N is given by equation (5.1) and $\{\mathbf{r}_k\}_{k=1}^{M_{fr}}$ are the points located on the free boundary $y = h(x)$ where we have the Dirichlet boundary condition $u = 1$. [7, p. 81]

Another idea of locating the free boundary is to preassign the locations of some of the singularities with respect to the location of the free boundary, and thus, the location of the free boundary is obtained directly from the locations of the certain singularities. The basic idea of this method is visualized in Figure 5.1, where the locations $\{(x_j, y_j)\}_{j=1}^{M_{fr}}$ of the singularities are fixed with respect to the free boundary $y = h(x)$ such that $(x_j, ry_j) = (x_j, h(x_j))$, where the locations x_j can be preassigned and r is chosen such that $0 < r < 1$. For the illustration of Figure 5.1 we have chosen $d = 4/5$. [2, pp. 388–389]

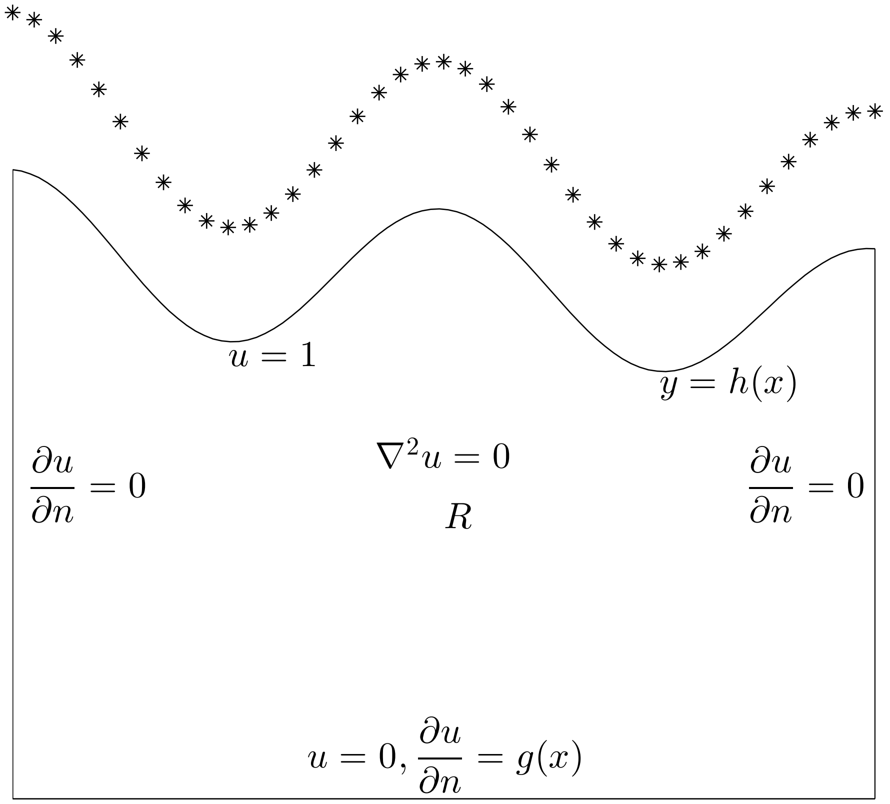
$u = 1$
$y = h(x)$
$\frac{\partial u}{\partial n} = 0$
$\nabla^2 u = 0$
R
$\frac{\partial u}{\partial n} = 0$
$u = 0, \frac{\partial u}{\partial n} = g(x)$

Figure 5.1: *The basic idea of preassigning the locations of the singularities with respect to the location of the free boundary. The singularities are marked with stars.*

Both of the above methods of determining the location of the unknown boundary were tested when trying out different implementations of this method. However, the obtained results were rather disappointing, and thus, we ended up choosing yet another way of locating the free boundary $y = h(x)$. The chosen method for the implementation is to leave the free boundary initially out of the minimization procedure, and when the approximate solution u_N is found, the free boundary curve is reconstructed from the equipotential lines $u_N = 1$. Even though the chosen method provides rather good approximations for the free boundary as we shall see in the test cases in Chapter 6, leaving the free boundary out of the minimization procedure leads to certain problems as well.

First of all, now that the free boundary is not present in the minimization procedure, we also lack the other endpoints of the vertical boundaries $x = a$ and $x = b$. Thus, since we cannot have a parametrization for the boundary points on the lines $x = a$ and $x = b$, we are practically forced to ignore the Neumann boundary condition $\partial u/\partial n = 0$ on those lines. Fortunately, numerical tests have shown that the boundary condition on the lines $x = a$ and $x = b$ is somewhat visible in the Neumann boundary condition $\partial u/\partial n = g(x)$ on the line $y = 0$. Thus, if the approximate solution u_N satisfies the condition $\partial u_N/\partial n = g(x)$ on the line $y = 0$ well, it is justifiable to assume that then the approximate solution u_N also satisfies the condition $\partial u_N/\partial n = 0$ on the lines $x = a$ and $x = b$ to some extent.

Second and a more problematic aspect in leaving the free boundary out of the minimization procedure is that, since the location of the free boundary is unknown during the minimization, we cannot assure that the singularities $\{\mathbf{s}_j\}_{j=1}^N$ are located outside the region R. Such unfavorably located singularities will likely cause notable errors on the reconstruction of the free boundary. An example of such an occurrence will be seen in one of the test cases in Chapter 6.

Now that we have abandoned the boundaries $y = h(x)$, $x = a$ and $x = b$, the remaining boundary condition for the approximate solution u_N to satisfy is the Cauchy boundary condition $u = 0$ and $\partial u/\partial n = g(x)$ on the line $y = 0$. We can make u_N satisfy the condition $u_N = 0$ intrinsically by using the method of images as suggested in Section 2.4. Namely, if the singularities are placed symmetrically with respect to the line $y = 0$ such that the coefficients of the fundamental solutions corresponding to the singularities at symmetrical locations are opposite, it is easy to see that the approximate solution u_N consisting of such singularity pairs is zero on the line $y = 0$. Thus, our fundamental solution is then of the form

$$\Phi^*(x, y, x_0, y_0) = -\frac{1}{2}\left(\ln[(x - x_0)^2 + (y - y_0)^2] - \ln[(x - x_0)^2 + (y + y_0)^2]\right), \quad (5.3)$$

where (x_0, y_0) and $(x_0, -y_0)$ are the locations of the symmetrically placed singularities. The approximate solution is then obtained as a linear combination of these fundamental solutions, i.e.,

$$u_N^*(x,y) = \sum_{j=1}^{N} c_j \Phi^*(x,y,x_j,y_j). \tag{5.4}$$

The idea of locating the singularities is illustrated in Figure 5.2.

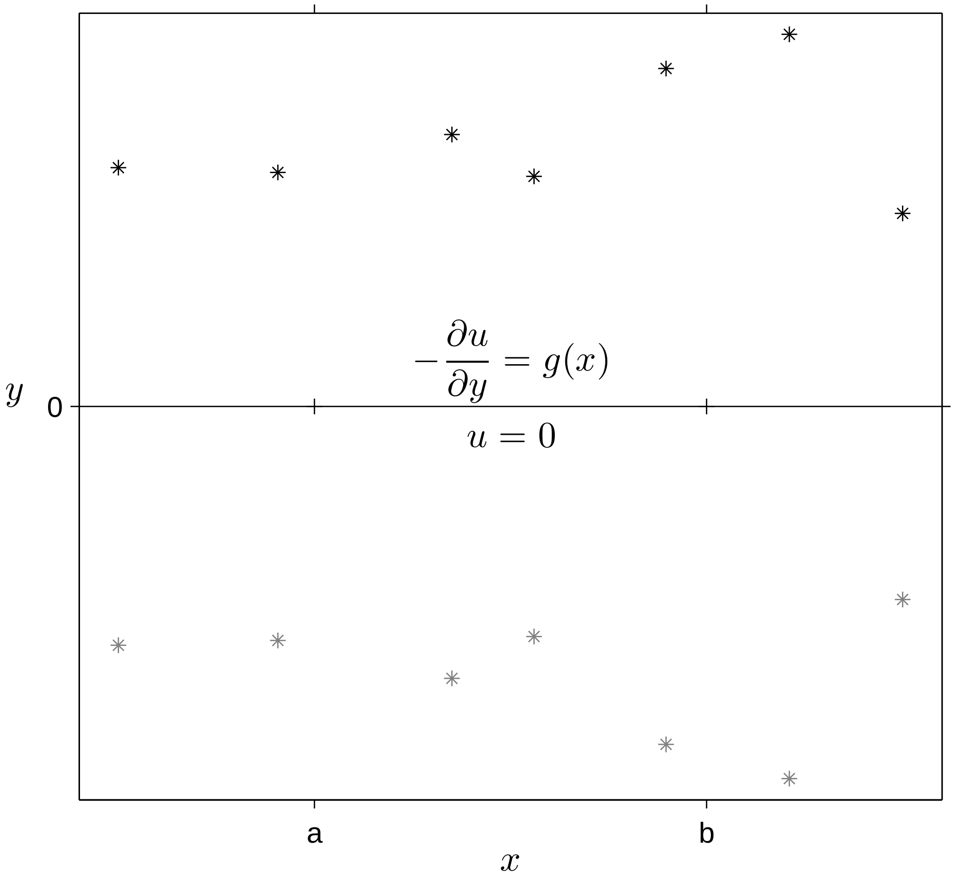

Figure 5.2: *Illustration of placing the singularities symmetrically with respect to the line $y = 0$. The singularities lying on the half plane $y < 0$ are marked with paler color for they are considered images of the singularities lying on the half plane $y > 0$.*

We still have to make the approximate solution u_N^* satisfy the boundary condition $\partial u/\partial n = g(x)$ on the line $y = 0$ as well as possible. This will be done by least squares minimization. Since the unit exterior normal to the line $y = 0$ is $(0, -1)$, we may symbolically calculate

$$-\frac{\partial u_N^*(x,y)}{\partial y} = \sum_{j=1}^{N} c_j \left[\frac{y - y_j}{(x-x_j)^2 + (y-y_j)^2} - \frac{y + y_j}{(x-x_j)^2 + (y+y_j)^2} \right], \tag{5.5}$$

from which we obtain

$$-\frac{\partial u_N^*(x,0)}{\partial y} = \sum_{j=1}^{N} \frac{-2c_j y_j}{(x-x_j)^2 + y_j^2}. \tag{5.6}$$

Now, if we let the singularities $\{(x_j, y_j)\}_{j=1}^N$ be located freely, the functional to be minimized can be written as

$$S(\mathbf{c}, \mathbf{x}, \mathbf{y}) = \sum_{i=1}^{M} \left(\sum_{j=1}^{N} \frac{-2c_j y_j}{(x_i - x_j)^2 + y_j^2} - g(x_i) \right)^2, \tag{5.7}$$

where the point $\{x_i\}_{i=1}^M$ are the points at which the value of $g(x)$ is known on the line $y = 0$.

5.3 Implementation of the Method of Fundamental Solutions

Since the functional $S(\mathbf{c}, \mathbf{x}, \mathbf{y})$ given in equation (5.7) is already in least squares form, the optimization problem to obtain the values of the unknown variables is simply given by

$$\min_{\mathbf{c},\mathbf{x},\mathbf{y}} S(\mathbf{c}, \mathbf{x}, \mathbf{y}). \tag{5.8}$$

The problem is nonlinear like the one considered in Section 3.3 within the boundary element method, and thus, we will use the same `lsqnonlin` function to solve the problem presented in Equation (5.8). Yet again, we will choose the trust-region-reflective algorithm for `lsqnonlin` in order to impose at least some constraints on the minimizer.

As a physical constraint on the minimizer we should require that all the coefficients $\{c_j\}_{j=1}^N$ of the fundamental solutions $\Phi^*(x, y, x_0, y_0)$ are positive, i.e., the fundamental solutions corresponding to the singularities located in the upper half plane $y > 0$ have positive coefficients and the ones located in the lower half plane $y < 0$ have negative coefficients. Furthermore, we should verify that the singularities corresponding to the fundamental solutions with positive coefficients are actually located in the upper half plane by requiring that the variables $\{y_j\}_{j=1}^N$ may obtain only positive values.

As an initial guess the values of the variables $\{y_j\}_{j=1}^N$ are all set to $1/2$ and the values of $\{x_j\}_{j=1}^N$ are evenly spaced over the interval $[a, b]$. The initial guess for all of the variables $\{c_j\}_{j=1}^N$ is $N^{-1} \int_a^b g(x)\mathrm{d}x$, where the integral is approximated by using Simpson's rule (2.43).

Regularization for the method of fundamental solutions in free boundary problems is discussed, e.g., by Borman et al. in [2]. However, the regularization methods discussed in [2] were not found either suitable or necessary in our problem. Instead, we will impose regularization on the number of singularity pairs used in constructing the approximate solution u_N^* (Recall that each fundamental solution $\Phi^*(x, y, x_0, y_0)$ has two singularities). We start of with a single pair, and the number of singularity pairs is increased if the approximate solution u_N^* does not satisfy the boundary condition $\partial u_N/\partial n = g(x)$ on the line $y = 0$ adequately well. This is achieved by checking whether the optimal value of the functional $S(\mathbf{c}, \mathbf{x}, \mathbf{y})$ is less than a certain tolerance. For the test cases in Chapter 6 we use the criterion $S(\mathbf{c}, \mathbf{x}, \mathbf{y}) < 10^{-3}$. Note that since the problem of locating the free boundary is ill-posed, the error on the boundary condition $\partial u_N/\partial n = g(x)$ does not necessarily correspond to the error on the free boundary reconstruction. The goal in using the least number of singularity pairs required for the criterion $S(\mathbf{c}, \mathbf{x}, \mathbf{y}) <$ `tol` is to obtain rather smooth reconstructions for the free boundary curve $y = h(x)$ and to keep the least squares problem (5.8) from becoming underdetermined so that the trust-region-reflective algorithm can be used.

When a suitable approximate solution u_N^* is found as a result of the least squares fit of the boundary data, the reconstruction for the free boundary $y = h(x)$ is obtained by searching the equipotential lines $u_N^* = 1$ which can be done using MATLAB's `contour` function. A MATLAB implementation `mfs` of the presented method is given in Appendix C, and the performance of the implemented method will be tested in Chapter 6. It should be noted that the least squares problem (5.8) will not become underdetermined in any of the test cases of Chapter 6, and thus, no problems will occur in using the trust-region-reflective algorithm.

6. TEST CASES, RESULTS AND DISCUSSION

6.1 General Aspects of the Test Cases

In this chapter, the performance of the methods presented in Chapters 3 – 5, or rather the performance of the implemented MATLAB functions found in Appendices A – C, is considered in five different test cases. Each of the test cases is considered individually in Sections 6.2 – 6.6, where we also present the graphical and numerical results obtained for the test cases using the three different methods. Then, in Section 6.7, we will discuss the performance of these methods in the test cases as a whole. Before going into the results, we will use the rest of this section to consider a few general aspects of the test cases.

In all the test cases the limits for x are $a = 0$ and $b = 1$ (see Fig. 1.1). The test cases are generated using MATLAB's `pdetool`, and the value of $\partial u/\partial n = g(x)$ on the line $y = 0$ is computed in each of the cases at 41 evenly spaced points on the interval $[0, 1]$. It should be noted that in `pdetool` the region R has to be constructed using polygons, circles and ellipses, and hence the analytic forms of the free boundary curves $y = h(x)$ may appear unnecessarily complicated in some of the test cases.

It should yet be noted that the results for all the test cases in Sections 6.2 – 6.6 are presented in a similar way. Both the graphical and the numerical presentation of the results will be considered in detail within Test Case 1 in Section 6.2, and one should consult that section in order to follow the presentation of the results in Sections 6.3 – 6.6.

6.2 Test Case 1 and Results

The free boundary curve for Test Case 1 is given by $y = h_1(x)$, where the function h_1 is defined over the interval $[0, 1]$ by

$$h_1(x) = \frac{3}{10}\sqrt{1 - \frac{32}{9}\left(x - \frac{1}{2}\right)^2}. \tag{6.1}$$

The curve $y = h_1(x)$ is smooth on the interval $x \in [0, 1]$, it has two minima at $x = 0$ and $x = 1$ and a single maximum at $x = 1/2$. It should be noted that the derivative of the function h_1 is nonzero at the points $x = 0$ and $x = 1$. The nonzero derivate at the endpoints of the interval $[0, 1]$ occurs also in Test Case 5 considered in Section

6.6, and we will not consider the effect of this property until discussing the results in general in Section 6.7.

A graphical presentation of the boundary curve $y = h_1(x)$ is given in Figure 6.1 where the curve is drawn with dashed line to all the three boxes. Each of these boxes contain also a graphical presentation of an approximation obtained for the free boundary curve using one of the three methods as listed below.

- The approximation obtained using the boundary element method `ibem` is drawn with solid line according to the linear boundary element approximation of the boundary curve $y = h(x)$.
- The approximation obtained using the alternating-field technique `aft` is drawn with dots corresponding to the obtained points $(X(v), Y(v))$.
- The approximation obtained using the method of fundamental solutions `mfs` is drawn with solid line according to the equipotential lines $u_N = 1$.

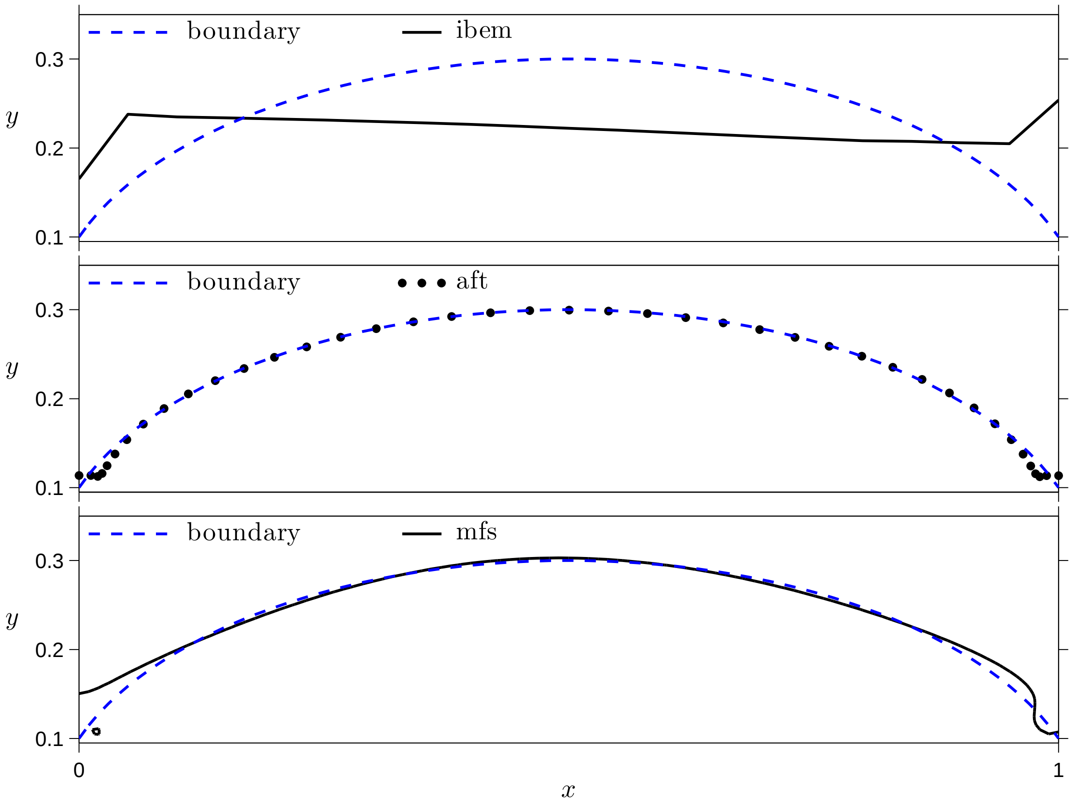

Figure 6.1: *The free boundary curve $y = h_1(x)$ (dashed line) in Test Case 1 and the approximations obtained for $y = h_1(x)$ using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs).*

When considering the free boundary approximations presented in Figure 6.1 we see that the approximation given by the `ibem` function does not resemble the actual boundary curve at all, the approximation given by the `aft` function seems quite accurate and the approximation given by the `mfs` function cannot even be presented as a curve $y = h(x)$ due to the equipotential loop near the point $(0, 0.1)$. The equipotential loop is caused by a singularity located inside it, and hence this case demonstrates well the problems that may arise if the singularities are located inside the region R. Such an approximation should strictly speaking be rejected, but for further consideration we simply content ourselves with ignoring the equipotential loop and consider the other curve the approximation of the free boundary.

Numerical data on the approximations obtained for the free boundary curve $y = h_1(x)$ is displayed in Table 6.1, where the average error norm $||Y - h(X)|| /n$, maximum absolute error $\max\{|Y - h(X)|\}$, maximum relative error $\max\{|Y - h(X)|/h(X)\}$ and the number of test points (X_j, Y_j), that is n, are given for each of the methods individually. The 'Other' column in the table contains information about the number of iterations k performed by the alternating-field technique `aft` and the number of singularity pairs N used by the method of fundamental solutions `mfs`.

Table 6.1: *Numerical data on the approximations obtained for the free boundary curve using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs) in Test Case 1.*

Method	$\frac{\|\|(Y-h(X))\|\|}{n}$	$\max\{\|Y - h(X)\|\}$	$\max\left\{\frac{\|Y-h(X)\|}{h(X)}\right\}$	n	Other
ibem	$1.41 \cdot 10^{-2}$	$1.54 \cdot 10^{-1}$	1.538	21	-
aft	$1.04 \cdot 10^{-3}$	$1.60 \cdot 10^{-2}$	0.137	41	$k = 111$
mfs	$7.31 \cdot 10^{-4}$	$5.04 \cdot 10^{-2}$	0.504	171	$N = 5$

The poor resemblance of the approximation obtained using the boundary element method `ibem`, which we saw in Figure 6.1, can also be seen in the data presented in Table 6.1, as all the errors on the `ibem` row are clearly larger than the ones on the other two rows. Additionally, the inaccuracy caused by the singularity located inside the region R in the approximation obtained using the method of fundamental solutions `mfs` can be seen in the table as the maximum relative error of over 50%. However, the average error norm of `mfs` is rather small, implying that for the most part the approximation is quite accurate. The smallest maximum errors in Test Case 1 are obtained using the alternating-field technique `aft`, which could be expected already based on Figure 6.1. Still, the smallest relative error is obtained using `mfs`, but one should note that the approximations obtained using the different methods are not necessarily comparable by the average error norm since the number of test points (X_j, Y_j) differs greatly between the methods.

6.3 Test Case 2 and Results

In Test Case 2 the free boundary curve is $y = h_2(x)$ where the function h_2 is defined by

$$h_2(x) = \begin{cases} \dfrac{1}{5} - \dfrac{1}{10}\sqrt{1-16x^2} & \text{if } 0 \le x < \dfrac{1}{4}, \\ \dfrac{1}{5} + \dfrac{1}{10}\sqrt{1-16\left(x-\dfrac{1}{2}\right)^2} & \text{if } \dfrac{1}{4} \le x < \dfrac{3}{4}, \\ \dfrac{1}{5} - \dfrac{1}{10}\sqrt{1-16(1-x)^2} & \text{if } \dfrac{3}{4} \le x \le 1. \end{cases} \tag{6.2}$$

The curve $y = h_2(x)$ is similar to the one in the previous section in a sense that it is smooth on the interval $x \in [0,1]$, it has two minima at $x = 0$ and $x = 1$ and a single maximum at $x = 1/2$. However, the derivative of the function h_2 at the points $x = 0$ and $x = 1$ is zero, where it differs from the function of the previous section.

A graphical presentation of the boundary curve $y = h_2(x)$ is given in Figure 6.2, where one can also see the approximations for the curve obtained using the three different methods. The graphical presentations of the curve $y = h_2(x)$ and of the approximations are given according to the way described in Section 6.2.

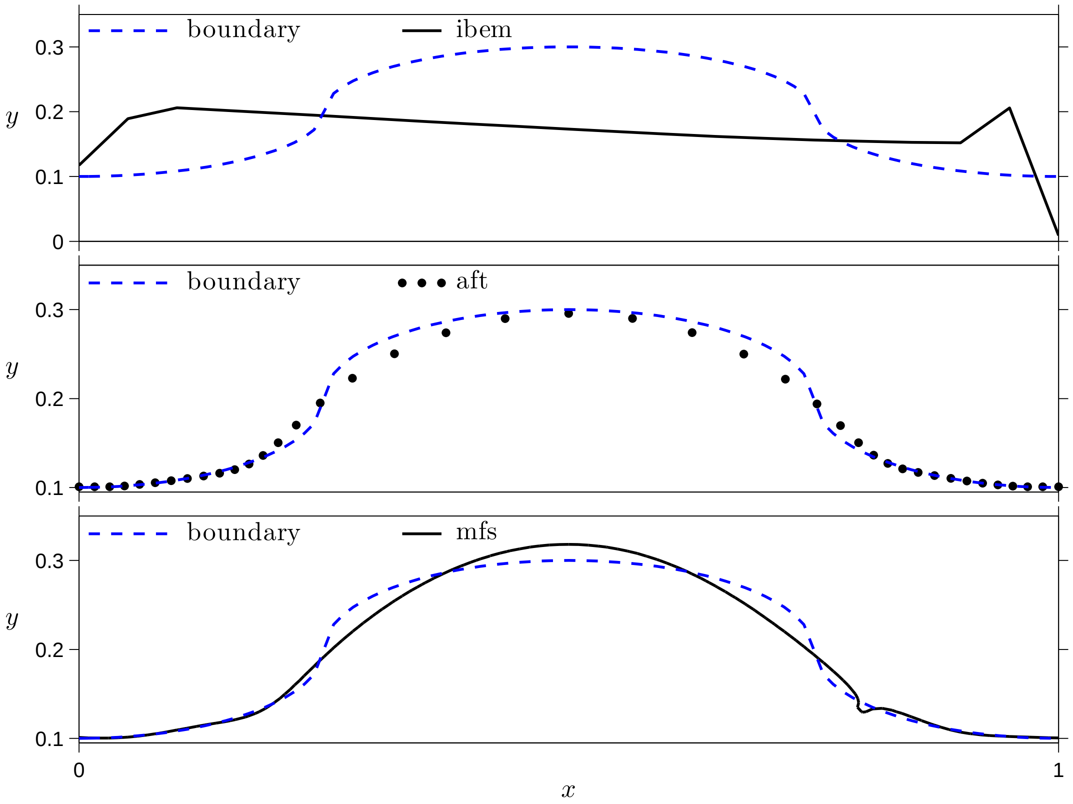

Figure 6.2: *The free boundary curve $y = h_2(x)$ (dashed line) in Test Case 2 and the approximations obtained for $y = h_2(x)$ using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs).*

Based on graphical results displayed in Figure 6.2 we see that the approximation obtained using the `ibem` function is again rather poor, while the functions `aft` and `mfs` result in quite sufficient approximations even though they could be more accurate at the certain parts of the boundary curve. Similar conclusions can be made based on Table 6.2, where the numerical data on Test Case 2 is presented. If necessary, one should consult Section 6.2 about the contents of the table.

Table 6.2: *Numerical data on the approximations obtained for the free boundary curve using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs) in Test Case 2.*

Method	$\frac{\|\|(Y-h(X))\|\|}{n}$	$\max\{\|Y-h(X)\|\}$	$\max\left\{\frac{\|Y-h(X)\|}{h(X)}\right\}$	n	Other
ibem	$1.94\cdot10^{-2}$	$1.29\cdot10^{-1}$	1.017	21	-
aft	$1.41\cdot10^{-3}$	$2.48\cdot10^{-2}$	0.106	41	$k=62$
mfs	$9.61\cdot10^{-4}$	$2.77\cdot10^{-2}$	0.117	189	$N=8$

As said, the numerical data presented in Table 6.2 is in accordance with the visuals of Figure 6.2, as the errors on the `ibem` row are clearly larger than the ones on the other two rows. The smallest maximum errors are again found on the `aft` row and the smallest relative error is again found on the `mfs` row. However, the differences in the errors between the rows `aft` and `mfs` in Table 6.2 are relatively small in general.

6.4 Test Case 3 and Results

The function h_3 for the free boundary curve $y = h_3(x)$ in Test Case 3 is given by

$$h_3(x) = \begin{cases} \dfrac{1}{2} - \dfrac{3}{20}\sqrt{1-\dfrac{400}{121}x^2} & \text{if } 0 \leq x < \dfrac{1}{2} \\ \dfrac{1}{2} - \dfrac{3}{20}\sqrt{1-\dfrac{400}{121}(1-x)^2} & \text{if } \dfrac{1}{2} \leq x \leq 1. \end{cases} \tag{6.3}$$

The curve $y = h_3(x)$ has two minima at $x = 0$ and $x = 1$ and a maximum at $x = 1/2$, where it is similar to the free boundary curves in the previous test cases. However, the curve $y = h_3(x)$ also has a non-smooth point at $x = 1/2$. The non-smooth point will have a similar effect as the nonzero derivative at the endpoints of the interval $[0, 1]$, and we will consider it as well in Section 6.7.

A graphical presentation of the free boundary curve $y = h_3(x)$ is given in Figure 6.3, where one can also see the approximations for the curve obtained using the three different methods. If necessary, one should consult Section 6.2 for the interpretation of the figure.

Based on Figure 6.3 it would seem that the approximation obtained using the

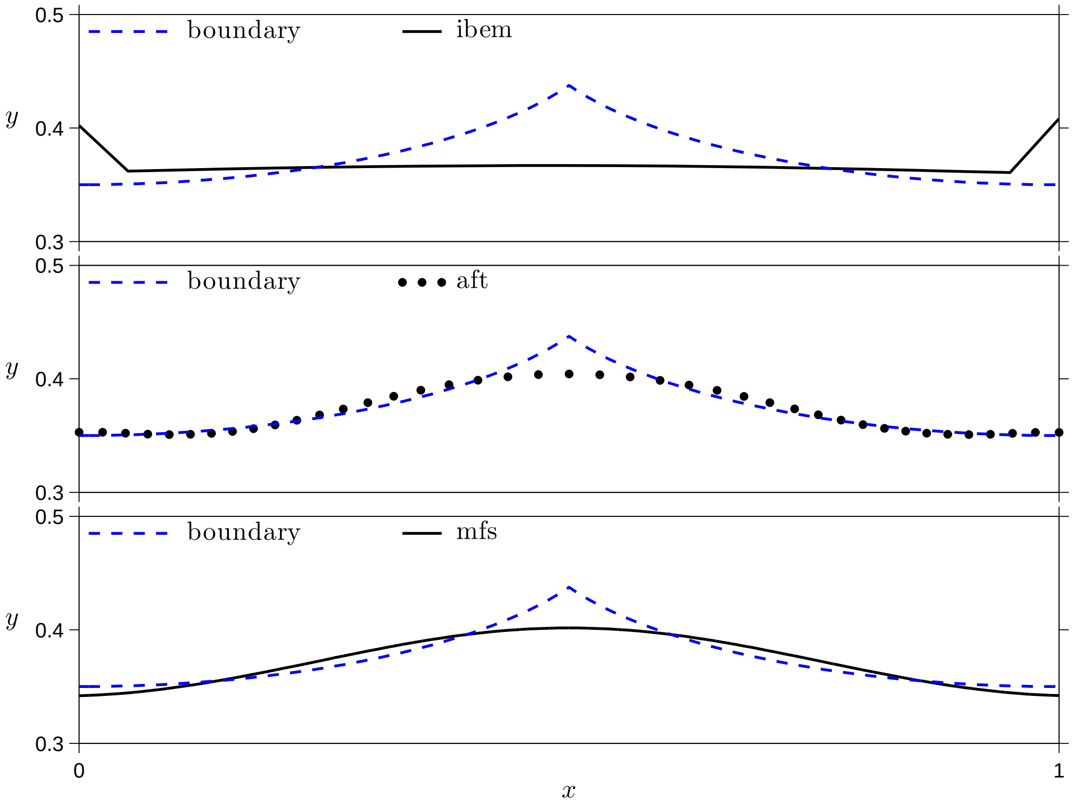

Figure 6.3: *The free boundary curve $y = h_3(x)$ (dashed line) in Test Case 3 and the approximations obtained for $y = h_3(x)$ using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs).*

`ibem` function is by far the most accurate, even though the approximation does not quite resemble the actual boundary curve. The approximations obtained using the functions `aft` and `mfs` seem to be quite accurate apart from the neighborhood of the non-smooth point $x = 1/2$ of the curve $y = h_3(x)$. The data presented in Table 6.3 again verifies these observations, as the errors on the `ibem` row are clearly smaller than in the two previous cases. The data on `aft` and on `mfs` do not differ significantly, but again the smallest maximum errors are obtained by `aft` and the smallest relative error is obtained by `mfs`.

Table 6.3: *Numerical data on the approximations obtained for the free boundary curve using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs) in Test Case 3.*

Method	$\frac{\|\|(Y-h(X))\|\|}{n}$	$\max\{\|Y-h(X)\|\}$	$\max\left\{\frac{\|Y-h(X)\|}{h(X)}\right\}$	n	Other
ibem	$6.52 \cdot 10^{-3}$	$7.06 \cdot 10^{-2}$	0.166	21	-
aft	$1.15 \cdot 10^{-3}$	$3.32 \cdot 10^{-2}$	0.076	41	$k = 135$
mfs	$8.01 \cdot 10^{-4}$	$3.59 \cdot 10^{-2}$	0.082	125	$N = 2$

6.5 Test Case 4 and Results

In Test Case 4 the function h_4 for the free boundary curve $y = h_4(x)$ is given by

$$h_4(x) = \begin{cases} \dfrac{2}{5} & \text{if } 0 \leq x < \dfrac{1}{4}, \\ \dfrac{3}{10} + \dfrac{1}{10}\sqrt{1-(4x-1)^2} & \text{if } \dfrac{1}{4} \leq x < \dfrac{1}{4} + \dfrac{5}{4\sqrt{29}}, \\ \dfrac{3}{10} + \dfrac{2}{5\sqrt{29}} - \dfrac{1}{10}\sqrt{1-\left(1+\dfrac{10}{\sqrt{29}}-4x\right)^2} & \text{if } \dfrac{1}{4} + \dfrac{5}{4\sqrt{29}} \leq x < \dfrac{1}{4} + \dfrac{5}{2\sqrt{29}}, \\ \dfrac{1}{5} + \dfrac{2}{5\sqrt{29}} & \text{if } \dfrac{1}{4} + \dfrac{5}{2\sqrt{29}} \leq x \leq 1. \end{cases} \tag{6.4}$$

The curve $y = h_4(x)$ is smooth on the interval $x \in [0, 1]$, the derivative of the function h_4 is zero at the endpoints of that interval, and the constant values $2/5$ and $1/5 + 2/(5\sqrt{29})$ are the maximum and minimum values of the function, respectively. Visualization for the curve $y = h_4(x)$ is given in Figure 6.4, where also the approximations obtained using the three different methods are displayed. If necessary, one should recall Section 6.2 for the interpretation of the figure.

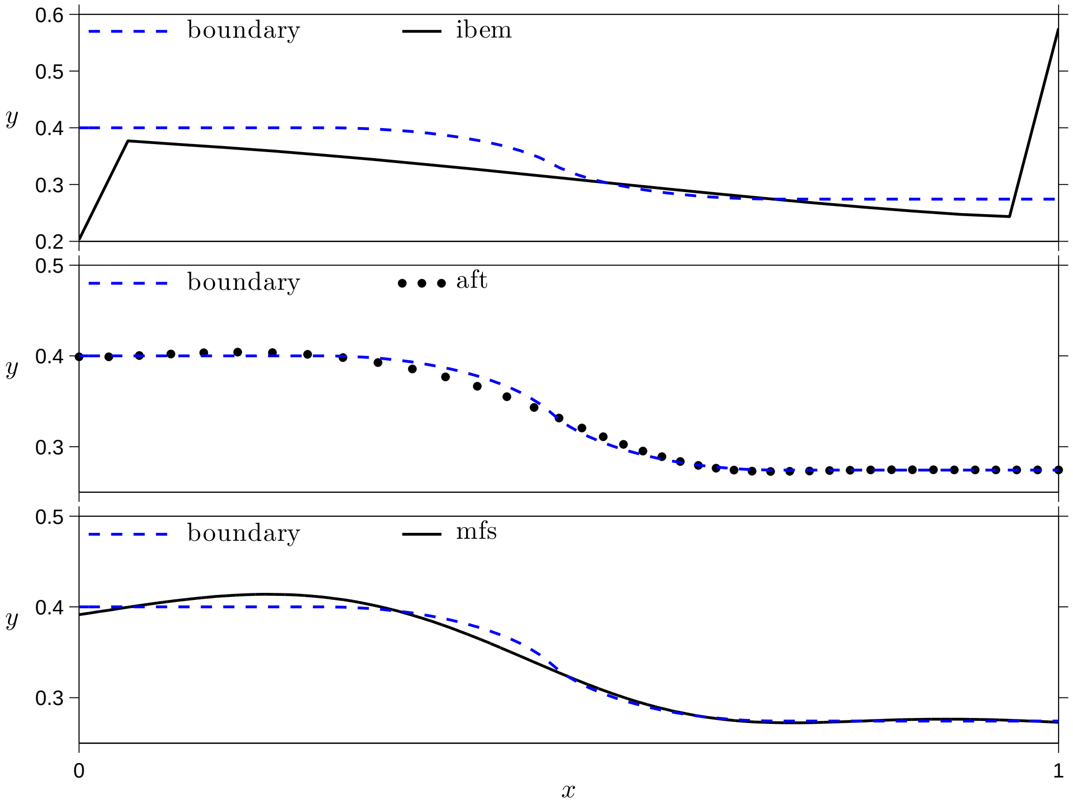

Figure 6.4: *The free boundary curve $y = h_4(x)$ (dashed line) in Test Case 4 and the approximations obtained for $y = h_4(x)$ using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs).*

In Figure 6.4 we see that for the first time the approximation obtained using the `ibem` function somewhat resembles the actual boundary curve. However, large errors occur in the approximation near the endpoints of the interval $[0, 1]$. Furthermore, the approximations obtained using the functions `aft` and `mfs` would seem to be by far the most accurate in this test case.

The numerical data corresponding to the results seen in Figure 6.4 is presented in Table 6.4. If necessary, one should consult Section 6.2 about the contents of the table.

Table 6.4: *Numerical data on the approximations obtained for the free boundary curve using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs) in Test Case 4.*

Method	$\frac{\|\|(Y-h(X))\|\|}{n}$	$\max\{\|Y-h(X)\|\}$	$\max\left\{\frac{\|Y-h(X)\|}{h(X)}\right\}$	n	Other
ibem	$1.84 \cdot 10^{-2}$	$3.01 \cdot 10^{-1}$	1.097	21	-
aft	$6.49 \cdot 10^{-4}$	$1.14 \cdot 10^{-2}$	0.031	41	$k = 157$
mfs	$5.98 \cdot 10^{-4}$	$1.43 \cdot 10^{-2}$	0.039	135	$N = 4$

Even though the approximation obtained by the `ibem` function somewhat resembled the actual boundary curve $y = h_4(x)$, the data in Table 6.4 shows that the errors are yet again relatively large and thus, the approximation is rather inaccurate. On the other hand, the errors on the rows `aft` and `mfs` are by far the smallest, which is in accordance with the observations we made about Figure 6.4. Even though the corresponding errors in rows `aft` and `mfs` of Table 6.4 are very close to each other, the smallest maximum errors are again obtained by `aft` and the smallest relative error is obtained by `mfs`.

6.6 Test Case 5 and Results

In Test Case 5 the free boundary curve $y = h_5(x)$ is a straight line, where the function h_5 is defined over the interval $[0, 1]$ by

$$h_5(x) = \frac{1}{5} + \frac{1}{5}x. \tag{6.5}$$

Clearly the curve $y = h_5(x)$ is smooth on the interval $x \in [0, 1]$, it has a maximum at $x = 1$ and a minimum at $x = 0$, and the derivative of the function $h_5(x)$ is nonzero at the endpoints of the interval $[0, 1]$, which was already mentioned in Section 6.2.

A graphical presentation of the boundary curve $y = h_5(x)$ is given in Figure 6.5, where one can also see the approximations for the curve obtained using the three different methods. Similar to the previous test cases, the graphical presentations of the curve $y = h_5(x)$ and of the approximations are given according to the way described in Section 6.2.

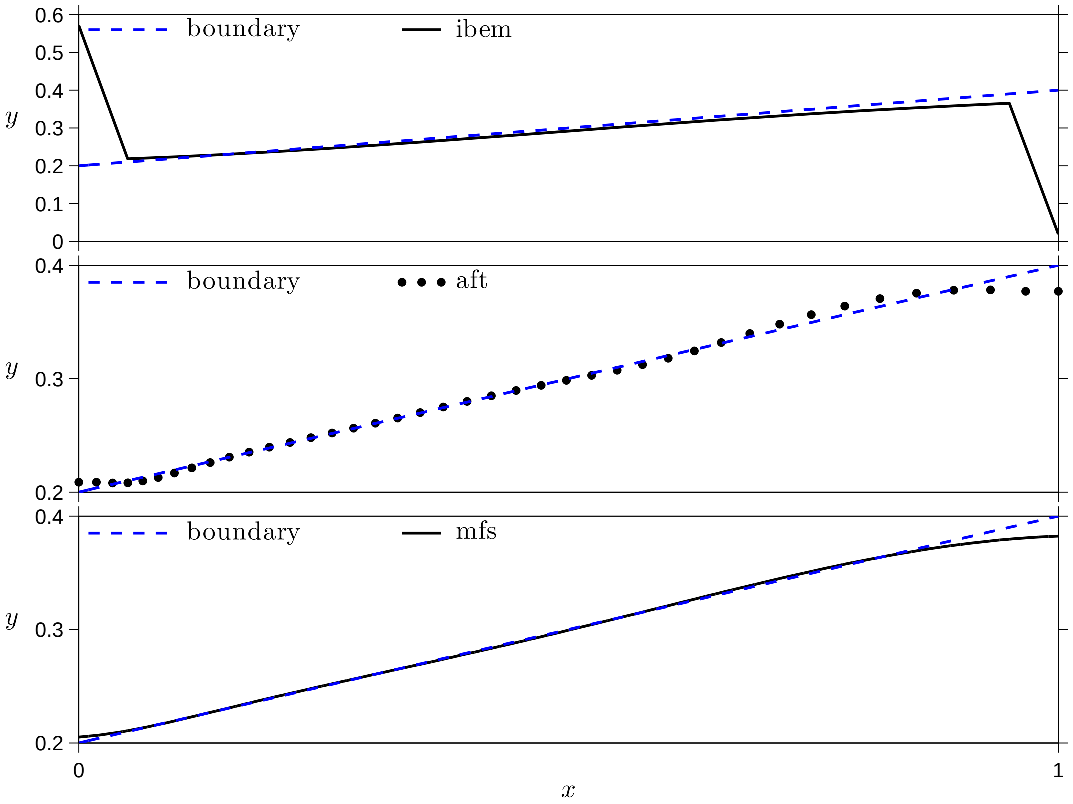

Figure 6.5: *The free boundary curve $y = h_5(x)$ (dashed line) in Test Case 5 and the approximations obtained for $y = h_5(x)$ using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs).*

We see in Figure 6.5 that the approximation obtained using the `ibem` actually follows the boundary curve $y = h_5(x)$ rather accurately apart from the large inaccuracies that occur near the endpoints of the interval $[0, 1]$. The result is somewhat similar to the one obtained in the previous test case, but here the resemblance is greater. The approximations obtained using the functions `aft` and `mfs` seem to be rather accurate for the most part, but a slight inaccuracy can be noticed in both the approximations near the line $x = 1$.

Numerical data on Test Case 5 is presented in Table 6.5. The data is once again presented according to the description in Section 6.2. We see from the table that even though the approximation obtained by `ibem` follows the actual boundary curve in Figure 6.5 rather accurately for the most part, the notable inaccuracies near the endpoints $x = 0$ and $x = 1$ result in large errors. When it comes to the functions `aft` and `mfs`, the errors are again relatively small, even if they were noticeable in Figure 6.5. It should yet be noted that this test case is the only one where the maximum errors obtained by `aft` were larger than the ones obtained by `mfs`.

Table 6.5: *Numerical data on the approximations obtained for the free boundary curve using the boundary element method (ibem), the alternating-field technique (aft) and the method of fundamental solutions (mfs) in Test Case 5.*

Method	$\frac{\|\|(Y-h(X))\|\|}{n}$	$\max\{\|Y-h(X)\|\}$	$\max\left\{\frac{\|Y-h(X)\|}{h(X)}\right\}$	n	Other
ibem	$2.54 \cdot 10^{-2}$	$3.81 \cdot 10^{-1}$	1.854	21	-
aft	$8.56 \cdot 10^{-4}$	$2.30 \cdot 10^{-2}$	0.057	41	$k = 134$
mfs	$3.09 \cdot 10^{-4}$	$1.76 \cdot 10^{-2}$	0.044	136	$N = 6$

6.7 Discussion on the Results

When considering the performance of the `ibem` function implemented for the boundary element method in all the test cases as a whole, it is justifiable to say that the free boundary reconstructions obtained using the `ibem` function were rather poor and did not give any reliable information about the free boundary curves in general. The apparent problem with the boundary element method is that not only is the parameterization of the free boundary $y = h(x)$ unknown but also the values of u and $\partial u/\partial n$ on certain boundaries are, which increases the amount of unknown variables in the minimization procedure. Even though regularization was used to take into account some constraints on the unknown variables, one should also validate that the values of the unknown u and $\partial u/\partial n$ are in accordance with the parametrization of the free boundary when a solution is found. However, it is not at all clear how this could be achieved in the minimization procedure, and even then, there is no guarantee that the free boundary reconstructions would be any better than what was obtained in the test cases. The method would be applicable to three dimensional problems but, due to the poor results in the two dimensional test cases, not much could be expected from its performance in 3D either.

Altogether, the `aft` function implemented for the alternating-field technique can be said to be the most reliable one of the three methods in the test cases. Inaccuracies did certainly occur, but some of them can be explained by the fact that the function $f(z) = u(x, y) + iv(x, y)$, where the real part u is the solution of the boundary value problem, is not conformal at certain points on ∂R as mentioned in Section 4.3.

First of all, consider Test Case 3 of Section 6.4 and the non-smooth point $x = 0.5$ of the curve $y = h_3(x)$, where the largest error occurred in the free boundary reconstruction. The angle on the curve $y = h_3(x)$ at the point $x = 0.5$ is clearly not preserved under the mapping $f(z) = u(x, y) + iv(x, y)$, since $f(z)$ maps the curve $y = h_3(x)$ to a straight line $u = 1$ which is smooth. In general, the curve $y = h(x)$ must not contain non-smooth points for the mapping $f(z) = u(x, y) + iv(x, y)$ to be conformal.

However, the curve $y = h(x)$ being smooth is not yet a sufficient condition for the mapping $f(z)$ to be conformal. Considering that the region R is mapped to a rectangle, the angles between the lines $x = a$ and $x = b$ and the curve $y = h(x)$ should be equal to the angles of a rectangle, i.e., $\pi/2$. The angles between the lines $x = a$ and $x = b$ and the curve $y = h(x)$ depend solely on the direction of the tangent of the curve, given by $(1, dh(x)/dx)$, at the points $x = a$ and $x = b$ [17, p. 422], and thus, the angles are $\pi/2$ exactly when

$$\frac{d}{dx}h(a) = \frac{d}{dx}h(b) = 0. \tag{6.6}$$

It can be seen in Figures 6.2 – 6.4 where the above condition is satisfied the relative errors near the points $x = 0$ and $x = 1$ are clearly smaller than the ones in Figures 6.1 and 6.5 where the condition is not satisfied. This can be seen most clearly in Figure 6.5 near the line $x = 1$.

Thus, for the most part the errors in the free boundary reconstructions obtained using the alternating-field technique occur due to the curve $y = h(x)$ not being smooth or the condition presented in Equation (6.6) being violated. Since these error sources are now known, the obtained free boundary reconstructions could in some cases be interpreted differently in order to take into account the possible errors. Regardless, the major drawback of the alternating-field technique is that it is restricted to two dimensional applications due to the fact that the certain properties of complex variables the technique utilizes cannot be used in higher dimensions.

The `mfs` function implemented for the method of fundamental solutions performed also rather well in the test cases. When compared to the performance of the alternating-field technique, the only remarkable difference occurred in the first test case where a singularity was located inside the region R. Furthermore, It was rather surprising that the free boundary reconstructions obtained using the functions `aft` and `mfs` were quite often so similar that even the inaccuracies occurred near the same parts of the actual free boundary curve $y = h(x)$. Altogether, the problems with the `mfs` implementation of the method of fundamental solutions are related to positioning the singularities favorably. The method is also very suitable for three dimensional applications since the only required change is to use the three dimensional fundamental solution [2, p. 386]

$$\Phi_3(\mathbf{r}, \mathbf{r}_0) = \frac{1}{||\mathbf{r} - \mathbf{r}_0||} \tag{6.7}$$

instead of the two dimensional one.

7. CONCLUSIONS

In this work we studied three different methods for solving a certain kind of free boundary problem for harmonic functions. MATLAB implementations were given for each of the methods and their performance was considered in numerical test cases.

The first method in this work was the boundary element method where the goal was to find the parametrization for the free boundary $y = h(x)$ by solving a set of integral equations which were actually reformulated as a least squares minimization problem. The results obtained using the boundary element method were clearly the weakest of the three methods, reasons for which may be the relatively large number of unknown variables in the minimization problem or insufficient regularization on certain unknown variables. The method could possibly be improved by further study, but considering that the other two methods already provided quite decent results it would not seem reasonable to put more effort into studying this method. Neither does the method stand out in its applicability to three dimensional problems since the method of fundamental solutions is applicable to 3D as well.

The second method was the alternating-field technique which is an iterative procedure for determining a parametrization for the free boundary $y = h(x)$ in the inverted plane. The harmonic potential u is considered as the real part of an holomorphic function $f(z) = u(x, y) + iv(x, y)$, and the problem is considered in the image of the region R under the mapping f. As a part of the process we needed to find the boundary conditions for v which is the harmonic conjugate of u on the region R.

The alternating-field technique was found to be the most reliable of the three methods when considering their performance in the test cases. Most of the errors in the free boundary reconstructions obtained using the technique where located in the neighborhood of points where the function $f(z)$ was not conformal, and thus, we might be able to prepare ourselves for errors in certain cases and find out a way to deal with them. However, the alternating-field technique is restricted only to two dimensional problems, and hence the development of the technique would not be beneficial if one is willing to find methods for solving three dimensional free boundary problems.

The third method was the method of fundamental solutions which is based on finding an approximate solution of the given boundary value problem as a linear combination of fundamental solutions of Laplace's equation. As a part of finding the approximate solution we also found a parametrization for the free boundary curve $y = h(x)$. Even though the implementation of the method of fundamental solutions given in this work could be characterized as a simple one, the results obtained in the test cases using the implementation were still rather good. However, the method should be further developed in order to obtain more reliable free boundary reconstructions. Considering that the method of fundamental solutions can be relatively easily applied to three dimensional problems, a further study of this method may very well be worth the effort.

In conclusion, out of the three studied methods the alternating-field technique was found to perform the most reliably in the two dimensional test cases, and a good runner-up was the method of fundamental solutions. However, if the main interest in the future is to study the free boundary problems considered in this work in three dimensions, the method of fundamental solutions is the only one of the methods studied in this work that is both applicable to three dimensions and has provided promising enough results in the two dimensional test cases, and thus, it should be the main focus of possible future work.

REFERENCES

[1] Ang, W.T. A Beginner's Course in Boundary Element Methods. Boca Raton, USA, 2007. Universal Publishers. 253 p.

[2] Borman, D. et al. The method of fundamental solutions for detection of cavities in EIT. Journal of Integral Equations and Applications 21(2009)3, pp. 383–406.

[3] Brown, J.W. & Churchill, R.V. Complex Variables and Applications. 7th edition. New York, USA, 2004. McGraw-Hill. 458 p.

[4] Datray R. & Lions, J.-L. Mathematical Analysis and Numerical Methods for Science and Technology. Berlin, Germany, 1990. Springer-Verlag. 695 p.

[5] Edwards, C.H. & Penney, D.E. Calculus 6e - Matrix version. New Jersey, USA, 2003. Prentice Hall. 1126 p.

[6] Eldén, L., Wittmeyer-Koch, L & Nielsen, H.B. Introduction to Numerical Computation - analysis and MATLAB® illustrations. Lund, Sweden, 2004. Studentliteratur. 375 p.

[7] Fairweather, G. & Karageorghis, A. The method of fundamental solutions for elliptic boundary value problems. Advances in Computational Mathematics 9(1998), pp. 69–95.

[8] Humaloja, J.-P., Hämäläinen, T. & Pohjolainen, S. Inverse Problem in Electrostatics with Free Boundary. The Fourth Finnish-Estonian Mathematical Colloquium, Helsinki, Finland, January 9-10, 2014.

[9] Humaloja, J.-P., Hämäläinen, T. & Pohjolainen, S. On Detecting the Shape of an Unknown Object in an Electric Field. The 18th European Conference on Mathematics for Industry, Taormina, Italy, June 9-13, 2014.

[10] Isakov, V. Inverse Problems for Partial Differential Equations. 2nd edition. New York, USA, 2006. Springer Science+Business Media, Inc. 344 p.

[11] Karamehmedovic, A. & Knudsen, K. Inclusion estimation from a single electrostatic boundary measurement. Inverse Problems 29(2013)2, 18 p.

[12] Kelley, C.T. Iterative Methods for Optimization. Philadelphia, USA, 1995. Society for Industrial and Applied Mathematics. 180 p.

[13] MATLAB and Optimization Toolbox Release 2013b, Natick, Massachusetts, USA, The MathWorks, Inc.

[14] Nilson, R.H. & Tsuei, Y.G. Free Boundary Problem of ECM by Alternating-Field Technique on Inverted Plane. Computer Methods in Applied Mechanics and Engineering 6(1975)3, pp. 265–282.

[15] Rundell, W. An equivalent source and probe-type method for inclusions from minimal Cauchy data. IAMCS Preprint Series 2009-093, 21 p.

[16] Stone, M. & Goldbart, P. Mathematics for Physics: A Guided Tour for Graduate Students. 1st edition. New York, USA, 2009. Cambridge University Press. 905 p.

[17] Taylor, A.E. & Mann, W.R. Advanced Calculus. 3rd edition. New York, USA, 1983. John Wiley & Sons, Inc. 732 p.

[18] Thom, A. & Apelt, C.J. Field Computations in Engineering & Physics. Bristol, Great Britain, 1961. D. Van Nostrand Company. 165 p.

[19] Zachmanoglou, E.C. & Thoe, D.W. Introduction to Partial Differential Equations with Applications. Baltimore, USA, 1976, The Williams & Wilkins Company. 405 p.

A. MATLAB FUNCTION FOR THE BOUNDARY ELEMENT METHOD

```
function [xout, yout] = ibem(gfun,a,b,x0,y0,n1,n3,n4)
% IBEM Function solves a free boundary problem using the boundary
% element method
%   gfun:       the function g(x) = -du/dy on y = 0 (row vector)
%   a, b:       limits for x
%   x0,y0:      chosen points (x0,y0) outside the region R
%   n1,n3,n4:   number of elements on segments r1,r3 and r4, resp.

gfun = gfun(1:end-1) + diff(gfun); % mean values of gfun
n2 = length(gfun); % number of elements on segment r2

% temporary variables
o = ones(length(x0),1);
k1 = 0:n1; o1 = ones(n1,1);
k2 = 0:n2; o2 = ones(n2,1);
k3 = 0:n3; o3 = ones(n3,1);
k4 = 0:n4; o4 = ones(n4,1);
xn2 = (b-a)*k2/n2; % x on segment r2
xn4 = (b-a)*k4/n4; % x on segment r4

% x(1:n4+1) = y_0:y_n4
% x(n4+2:n4+n1+1) = u(k) on x = b
% x(n4+n1+2:n4+n1+n3+1) = u(k) on x = a
% x(n4+n1+n3+2:2*n4+n1+n3+1) = dudx(k) on y = h(x)
% x(2*n4+n1+n3+2:3*n4+n1+n3+1) = dudy(k) on y = h(x)

% segment r1
F1 = @(x) 1/(2*pi)*sum(o*x(n4+2:n4+n1+1) ...
  .*(atan( (o*k1(2:end)*x(n4+1)/n1 - (o1*y0)').(b - (o1*x0)')) ...
  - atan( (o*k1(1:end-1)*x(n4+1)/n1 - (o1*y0)')./(b - (o1*x0)'))),2);
```

```
% segment r2
F2 = -1/(4*pi)*sum( (o*gfun) ...
  .*(2*(o2*y0)'.*atan( (o*xn2(2:end) - (o2*x0)')./( (o2*y0)')) ...
  + log( (o*xn2(2:end) - (o2*x0)').^2 + ( (o2*y0)').^2) ...
  .*(o*xn2(2:end) - (o2*x0)')...
  - 2*(o2*y0)'.*atan( (o*xn2(1:end-1) - (o2*x0)')./( (o2*y0)')) ...
  - log( (o*xn2(1:end-1) - (o2*x0)').^2 + ( (o2*y0)').^2) ...
  .*(o*xn2(1:end-1) - (o2*x0)') - 2*(b-a)/n2),2);

% segment r3
F3 = @(x) 1/(2*pi)*sum(o*x(n4+n1+2:n4+n1+n3+1) ...
  .*(atan( (o*k3(1:end-1)*x(1)/n3 - (o3*y0)')./(a - (o3*x0)')) ...
  - atan( (o*k3(2:end)*x(1)/n3 - (o3*y0)')./(a - (o3*x0)'))),2);

% segment r4 first half
F41 = @(x) 1/(2*pi)*sum(...
  atan( ( (o*xn4(1:end-1) - (o4*x0)')*(b-a)/n4 ...
  + (o*x(1:n4) - (o4*y0)').*(o*(x(2:n4+1)-x(1:n4)))) ...
  ./( (o*xn4(1:end-1) - (o4*x0)').*(o*(x(2:n4+1)-x(1:n4))) ...
  - (o*x(1:n4) - (o4*y0)')*(b-a)/n4)) ...
  - atan( ( (o*xn4(2:end) - (o4*x0)')*(b-a)/n4 ...
  + (o*x(2:n4+1) - (o4*y0)').*(o*(x(2:n4+1)-x(1:n4)))) ...
  ./( (o*xn4(1:end-1) - (o4*x0)').*(o*(x(2:n4+1)-x(1:n4))) ...
  - (o*x(1:n4) - (o4*y0)')*(b-a)/n4)),2);

% segment r4 second half
F42 = @(x) -1/(4*pi)*sum(o*(x(n4+n1+n3+2:2*n4+n1+n3+1) ...
  .*(x(1:n4) - x(2:n4+1)) + x(2*n4+n1+n3+2:3*n4+n1+n3+1)*(b-a)/n4)...
  .*(log( (o*xn4(2:end) - (o4*x0)').^2 ...
  + (o*x(2:n4+1) - (o4*y0)').^2) ...
  - 2 + (2*(atan( ( ( (b-a)/n4)^2 ...
  + (o*xn4(1:n4) - (o4*x0)').*(b-a)/n4 ...
  + o*(x(2:n4+1) - x(1:n4) ).^2 + (o*x(1:n4) - (o4*y0)') ...
  .*(o*(x(2:n4+1) - x(1:n4) ) ) )./abs( (o*x(1:n4) ...
  - (o4*y0)')*(b-a)/n4 - (o*(x(2:n4+1) ...
  - x(1:n4) ) ).*(o*xn4(1:n4) - (o4*x0)'))) ...
  - atan( ( (o*xn4(1:end-1) - (o4*x0)').*(b-a)/n4 ...
  + (o*x(1:n4) - (o4*y0)').*(o*(x(2:n4+1) - x(1:n4) ) ) ) ...
  ./abs( (o*x(1:n4) - (o4*y0)')*(b-a)/n4 ...
  - (o*(x(2:n4+1) - x(1:n4) ) ).*(o*xn4(1:n4) - (o4*x0)')))) ...
  .*abs( (o*x(1:n4) - (o4*y0)')*(b-a)/n4 ...
  - (o*(x(2:n4+1) - x(1:n4))).*(o*xn4(1:n4) - (o4*x0)') ) ...
  + ( (o*xn4(1:n4) - (o4*x0)')*(b-a)/n4 ...
  + (o*x(1:n4) - (o4*y0)').*(o*(x(2:n4+1) - x(1:n4)))) ...
  .*(log( (o*xn4(2:end) - (o4*x0)').^2 ...
  + (o*x(2:n4+1) - (o4*y0)').^2) ...
  - log( (o*xn4(1:n4) - (o4*x0)').^2 ...
  + (o*x(1:n4) - (o4*y0)').^2))) ...
  ./( ( (b-a)/n4)^2 + o*(x(2:n4+1) - x(1:n4) ).^2)),2);

% approximated line integral over the boundary of R
F = @(x) F1(x) + F2 + F3(x) + F41(x) + F42(x);
% regularization
%   the integral of du/dn over R should be zero
T1 = @(x) sum(x(n4+n1+n3+2:2*n4+n1+n3+1).*(x(1:n4) - x(2:n4+1)) ...
    + x(2*n4+n1+n3+2:3*n4+n1+n3+1)*(b-a)/n4) + sum(gfun)*(b-a)/n2;
%   the values of u should be strictly increasing on x = a and x = b
T2 = @(x) 0.1*exp(-n1*diff(x(n4+2:n4+n1+1) ) );
T3 = @(x) 0.1*exp(-n3*diff(x(n4+2:n4+n1+1) ) );
T = @(x) [T1(x); T2(x)'; T3(x)'];
S = @(x) [F(x); T(x)]; % function for minimization

% initial guess for the minimization (solution to h(x) = 0.3)
ini = [0.3*ones(1,n4+1), 1/(2*n1):1/n1:(1-1/(2*n1) ), ...
    1/(2*n3):1/n3:(1-1/(2*n3) ), zeros(1,n4), 10/3*ones(1,n4)];
% lower and upper bounds for minimization: y > 0, u in [0, 1]
lb = [zeros(1,n4+1), zeros(1,n1+n3), -inf*ones(1,2*n4)];
ub = [inf*ones(1,n4+1), ones(1,n1+n3), inf*ones(1,2*n4)];
% increase the number of maximum function evaluations
options = optimset('maxfunevals',100000);
% solve x by nonlinear least squares minimization
x = lsqnonlin(S,ini,lb,ub,options);
% values of xout are evenly spaced from a to b, values of yout
% correspond to the n4+1 first elements in the solution vector x
xout = xn4; yout = x(1:n4+1);
end
```

B. MATLAB FUNCTION FOR THE ALTERNATING-FIELD TECHNIQUE

```
function [X, Y] = aft(gfun,a,b, tol)
% AFT Function solves a free boundary problem using the alternating-
% field technique
%   gfun: the function g(x) = -du/dy on y = 0
%   a, b: limits for x
%   tol:  tolerance for the stopping criterion

% determine the number of mesh points
M = numel(gfun); N = M;
% compute values for the function v = v(x) using the trapezoidal rule
vx = zeros(M,1);
for k=1:(M-1)
    vx(k) = -(b-a)/(M-1)*(0.5*(gfun(k) ...
            + gfun(M)) + sum(gfun(k+1:M-1)));
end
% compute mesh size
du = 1/(N-1); dv = vx(1)/(M-1);
v = vx(1):-dv:0; % determine mesh points for v
% compute the values for the function x = x(v) by interpolation
x0 = qinterp(linspace(a,b,M),vx,v);
X = x0; % initial guess for the unknown boundary condition x = X(v)
% compute the coefficient matrix A and the vector b such that Ax = b
[A, Bx] = lapAx(M,N,du,dv,a,X,b,x0);
% LU decomposition for the matrix A
[Lx, Ux, Px] = lu(A);
% solve x from the system of linear equations Ax = b
x = Ux\(Lx\(Px*Bx));
% reshape x to an M times N matrix
x = reshape(x,M,N);
% calculate an estimate for the unknown boundary condition y = Y(v)
Y = newY(M,N,du,dv,x);
```

```
% compute the coefficient matrix A and the vector b such that Ay = b
[A, By] = lapAy(M,N,du,dv,Y);
% LU decomposition for the matrix A
[Ly, Uy, Py] = lu(A);
% solve y from the system of linear equations Ay = b
y = Uy\(Ly\(Py*By));
% reshape y to an M times N matrix
y = reshape(y,M,N);
% calculate a new estimate for the boundary condition x = X(v)
X = newX(M,N,du,dv,y,x0);
% initial estimate for the arc lenth parameter
s = cumsum(sqrt(diff(X).^2 + diff(Y).^2));
while true
    % replace the changed values of X(v) by new ones and solve x
    Bx(end-M+1:end) = X;
    x = Ux\(Lx\(Px*Bx));
    x = reshape(x,M,N);
    Y = newY(M,N,du,dv,x);
    % replace the changed values of Y(v) by new ones and solve y
    By(end-M+1:end) = Y;
    y = Uy\(Ly\(Py*By));
    y = reshape(y,M,N);
    X = newX(M,N,du,dv,y,x0);
    % calculate new arc length parameter sn and check convergence
    sn = cumsum(sqrt(diff(X).^2 + diff(Y).^2));
    if  norm(s - sn)^2 < tol
        break
    end
    s = sn; % replace s by sn and return to step 1
end

function newf = qinterp(f,x,newx)
% QINTERP interpolates f at x by second degree polynomials

newf = zeros(length(newx),1);
% the endpoints of x and newx are known to be the same
newf(1) = f(1); newf(end) = f(end);
% find the three closest points to newx(l)
l = 2; k = 1;
```

```
while l < length(newx)
    if k < length(x) - 1
        if sum(abs(newx(l) - x(k-1:k+1))) > ...
            sum(abs(newx(l) - x(k:k+2)))
            k = k + 1;
            continue
        end
    end
    % compute the coefficients of the interpolating polynomial
    c0 = f(k-1);
    c1 = (f(k) - c0)/(x(k) - x(k-1));
    c2 = (f(k+1) - c0 - c1*(x(k+1) - x(k-1)))/ ...
        ((x(k+1) - x(k-1))*(x(k+1) - x(k)));
    % determine the polynomial and compute its value at newx(l)
    P = @(t) c0 + c1*(t - x(k-1)) + c2*(t - x(k-1))*(t - x(k));
    newf(l) = P(newx(l));
    l = l + 1;
end

end

function [A, B] = lapAx(M,N,dx,dy,b,X,a,x0)
% LAPAX Laplacian matrix and solution vector for x
%   M, N:         number of mesh points
%   dx, dy:       mesh size
%   b, X, a, x0: Dirihlet boundary conditions for x
%   note also the Neumann boundary condition on the line u = 0

d0 = [ones(1,2*M), ...
    repmat([1, -2*(dx^2 + dy^2)*ones(1,M-2), 1], 1, N-3), ...
    ones(1,M)]'; % main diagonal
d1p = [0, zeros(1,2*M), ...
    repmat([0, dx^2*ones(1,M-2), 0], 1, N-3), ...
    zeros(1,M-1)]'; % 1st diagonal
d1n = [zeros(1,2*M-1), ...
    repmat([0, dx^2*ones(1,M-2), 0], 1, N-3), ...
    zeros(1,M+1)]'; % -1st diagonal
dMp = [zeros(1,3*M), ...
    repmat([0, dy^2*ones(1,M-2), 0], 1, N-3)]'; % M:th diagonal
```

```
dMn = [zeros(1,M), ...
    repmat([0, dy^2*ones(1,M-2), 0], 1, N-3), ...
    zeros(1,2*M)]'; % -M:th diagonal
% create sparse matrix A with the given diagonals
D = [dMn, d1n, d0, d1p, dMp];
d = [-M, -1, 0, 1, M];
A = spdiags(D,d,M*N,M*N);
% check dimensions of X and x0
if diff(size(X)) < 1
    X = X';
end
if diff(size(x0)) < 1
    x0 = x0';
end
% solution vector b
B = [x0, x0, repmat([b, zeros(1,M-2), a], 1, N-3), X]';
end

function [A, B] = lapAy(M,N,dx,dy,Y)
% LAPAY Laplacian matrix and solution vector for y
%   M, N:   number of mesh points
%   dx, dy: mesh points
%   Y:      Dirihlet bounary condition for y
%   note also y = 0 on the line u = 0 and the Neumann
%   boundary conditions on the lines v = 0 and v = V

d0 = [ones(1,1*M), ...
    repmat([1, -2*(dx^2 + dy^2)*ones(1,M-2), 1], 1, N-2), ...
    ones(1,M)]'; % main diagonal
d1p = [zeros(1,M+1), ...
    repmat([-1, dx^2*ones(1,M-2), 0], 1, N-2), ...
    zeros(1,M-1)]'; % 1st diagonal
d1n = [zeros(1,M-1), ...
    repmat([0, dx^2*ones(1,M-2), -1], 1, N-2), ...
    zeros(1,M+1)]'; % -1st diagonal
dMp = [zeros(1,2*M), ...
    repmat([0, dy^2*ones(1,M-2), 0], 1, N-2)]'; % M:th diagonal
dMn = [repmat([0, dy^2*ones(1,M-2), 0], 1, N-2), ...
    zeros(1,2*M)]'; % -M:th diagonal
```

```
% create sparse matrix A with the given diagonals
D = [dMn, d1n, d0, d1p, dMp];
d = [-M, -1, 0, 1, M];
A = spdiags(D,d,M*N,M*N);
% check dimensions of Y
if diff(size(Y)) < 1
    Y = Y';
end
% solution vector b
B = [zeros(1,M*(N-1)), Y]';
end

function Y = newY(M,N,du,dv,x)
% NEWY Calculates new estimate for the bondary condition y = Y(v)

Y = zeros(M,1);
for k = 2:(M-1)
    Y(k) = -0.5*du/(3*dv)*(x(k-1,1) - x(k+1,1) ...
        + 2*sum(x(k-1,3:2:N-2) - x(k+1,3:2:N-2)) ...
        + 4*sum(x(k-1,2:2:N-1) - x(k+1,2:2:N-1)) ...
        + (x(k-1,N) - x(k+1,N)));
end
Y(1) = Y(2); Y(M) = Y(M-1);
end

function X = newX(M,N,du,dv,y,x0)
% NEWX Calculates new estimate for the boundary condition x = X(v)

X = zeros(M,1);
for k = 2:(M-1)
    X(k) = x0(k) + 0.5*du/(3*dv)*((y(k-1,1) - y(k+1,1)) ...
        + 2*sum(y(k-1,3:2:N-2) - y(k+1,3:2:N-2)) ...
        + 4*sum(y(k-1,2:2:N-1) - y(k+1,2:2:N-1)) ...
        + (y(k-1,N) - y(k+1,N)));
end
X(1) = x0(1); X(M) = x0(M);
end

end
```

C. MATLAB FUNCTION FOR THE METHOD OF FUNDAMENTAL SOLUTIONS

```
function [X, Y] = mfs(gfun,a,b,tol)
% MFS Function solves a free boundary problem using the method
% of fundamental solutions
%   gfun: the function g(x) = -du/dy on y = 0 (column vector)
%   a, b: limits for x
%   tol:  tolerance for the number of singularities used

% integrating -g(x) from a to b in order to obtain an initial
% guess for the coefficients c_j (Simpson's rule)
intg = (a-b)/(3*length(gfun))*(gfun(1) + gfun(end) ...
    + 2*sum(gfun(3:2:end-2) ) + 4*sum(gfun(2:2:end-1) ) );
M = length(gfun); on1 = ones(M,1); % temporary variables
xn1 = linspace(a,b,M); % points where g(x) is known
% increase the number of maximum function evaluations and iterations
options = optimset('maxfunevals',100000,'maxiter',10000);
N = 1; % the number of singularity pairs, initiatively one

while true
% initial guess for the minimization procedure
ini = [intg/N*ones(1,N), 0.5*(b-a)*ones(1,N), linspace(a,b,N)];
% lower and upper bounds for minimization: c_j > 0, y > 0
lb = [zeros(1,N), zeros(1,N), -inf*ones(1,N)];
ub = [inf*ones(1,N), inf*ones(1,N), inf*ones(1,N)];
om = ones(N,1); % temporary variable

% indices for the unknown variables and the objective function
% x(1:M) = c_1:M
% x(M+1:2M) = y_1:M
% x(2*M+1:3*M) = x_1:M
S = @(x) sum(-2*on1*x(1:N).*(on1*x(N+1:2*N) ) ...
    ./( ( (om*xn1)' - on1*x(2*N+1:3*N)).^2 ...
    + (on1*x(N+1:2*N)).^2),2) - gfun;
% solve x by least squares minimization
x = lsqnonlin(S,ini,lb,ub,options);

if norm(S(x))^2 < tol
    break % break if the chosen tolerance is reached
end
N = N + 1; % otherwise increase the number of singularity pairs
end
% initialize search grid for the isoline(s) u_N = 1
[Xg, Yg] = meshgrid(linspace(a,b,101),linspace(0,(b-a)/2,101));
% compute the values of u_N on the search grid points
Z = u_N(Xg, Yg, x(1:N), x(2*N+1:3*N), x(N+1:2*N));
% plot the isoline(s) u_N = 1 and get the contour matrix C
[C, ~] = contour(Xg,Yg,Z,[1 1],'color','k');
% get the points X and Y from the contour matrix C
X = C(1,2:end);
Y = C(2,2:end);

function u = u_N(X,Y,c,xm,ym)
% U_N Function computes the value of u_N at points (X,Y)
%   X,Y:   points (X,Y) where the value of u_N is computed
%   c:     the coefficients c_j of the fundamental solutions
%   xm,ym: locations (xm,ym) for the singularities

% function handle for u_N
u_N = @(x0,y0) 0.5*sum(c ...
    .*(log( (x0 - xm).^2 + (y0 + ym).^2) ...
    - log( (x0 - xm).^2 + (y0 - ym).^2) ) );

u = zeros(size(X)); % initialization
% compute the values of u_N at the points (X,Y)
for k = 1:numel(X)
    u(k) = u_N(X(k),Y(k));
end
end

end
```