

Condition numbers in the boundary element method : shape and solvability

Citation for published version (APA):

Dijkstra, W. (2008). *Condition numbers in the boundary element method : shape and solvability*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR633956>

DOI:

[10.6100/IR633956](https://doi.org/10.6100/IR633956)

Document status and date:

Published: 01/01/2008

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Condition Numbers in the Boundary
Element Method:
Shape and Solvability

Copyright ©2008 by W. Dijkstra, Eindhoven, The Netherlands.
All rights are reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of the author.
Printed by Eindhoven University Press
Cover design: Creanza media

cover background: Victor Vasarely

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Dijkstra, Willem

Condition numbers in the boundary element method : shape and solvability /
door Willem Dijkstra. -
Eindhoven: Technische Universiteit Eindhoven, 2008.

Proefschrift. -ISBN 978-90-386-1245-4

NUR 919

Subject headings: boundary element methods / integral equations ; numerical
methods / numerical simulation / Navier-Stokes equations / glass
2000 Mathematics Subject Classification:
65N38, 74S15, 65R20, 35Q30, 76D05

Condition Numbers in the Boundary Element Method: Shape and Solvability

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 17 april 2008 om 16.00 uur

door

Willem Dijkstra

geboren te Driebruggen

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr. R.M.M. Mattheij

Copromotor:

Dr. M.E. Hochstenbach

*Apply your heart to instruction,
and your ears to words of knowledge*

Proverbs 23:12 (NKJ)

Contents

1	Introduction	1
1.1	Problem setting	1
1.2	Boundary Element Method	2
1.3	Condition number	4
1.4	Condition numbers of the BEM-matrices	5
1.5	Objectives	6
1.6	Outline of the thesis	8
2	Boundary Element Method	10
2.1	Integral Equations	10
2.2	Operator theory	15
2.3	Algebraic Equations	17
2.4	Matrix Elements	20
2.5	Numerical integration	22
3	Laplace equation at two-dimensional domain	24
3.1	Eigenvalues of \mathcal{K}^s and \mathcal{K}^d	24
3.2	Eigenvalues of the matrices	27
3.3	Dirichlet problem	29
3.4	Neumann problem	32
3.5	Mixed boundary conditions	34
3.6	Decoupled equations	43
4	Logarithmic capacity	47
4.1	Introduction	47
4.2	Logarithmic capacity	50
4.3	Dirichlet problem	51
4.4	Mixed problem	53
4.5	Neumann problem	55
4.6	Examples	57

5	Two-dimensional Stokes flow	65
5.1	Boundary integral equations for 2D Stokes flow	65
5.2	Eigensystem of the single layer operator \mathcal{G} at a circle	67
5.3	Invertibility of single layer operator on general domain	70
5.4	Invertibility of operator on general domain with mixed conditions .	73
5.5	Numerical examples	76
5.6	Blowing problem in 2D	86
6	Three-dimensional Stokes flow	94
6.1	Simulating the blowing of glass containers with the BEM	94
6.2	Mathematical model	96
6.3	Boundary integral equations	103
6.4	Numerical solution	106
6.5	Time integration and post-processing	108
7	Results	111
7.1	Glass blowing	111
7.2	Curvature driven flow	123
7.3	Parameter analysis	128
A	Curvature approximation	134
B	Contact problem	142
C	Smoothing techniques	149
C.1	Laplacian smoothing	149
C.2	Smoothing scalar function	151
C.3	Smoothing vector field	152
	Bibliography	155
	Index	162
	Summary	166
	Samenvatting	169
	Acknowledgements	172
	Curriculum vitae	173

Chapter 1

Introduction

*In mathematics you don't understand
things. You just get used to them.*
J. von Neumann [99]

1.1 Problem setting

Today many commercially available mathematical software tools exist that solve a large variety of mathematical problems. Often these tools are being used in a “black box” approach. The user inserts a mathematical problem and obtains a solution for it, without too much knowledge of what is happening behind the scene. Of course, software manufacturers claim that there is no need to have detailed information on the working of their products. Regardless of the mathematical problem that is inserted, the software will produce the correct solution. The question is whether this is always true.

Typically, in software tools a mathematical problem is defined on an object that has a certain size and shape. The user assumes that the success of solving the problem does not depend on this size and shape. For instance, if a correct solution is obtained for a square object, we expect a correct solution for the same mathematical problem for a rectangular object. If a correct solution is obtained for a circular object, we expect a correct solution for a circular object that is twice as large.

The mathematical software tools employ several numerical methods that solve mathematical problems. One such method is the boundary element method (BEM), which is the topic of this thesis. The success of this particular method to solve certain mathematical problems may depend on the size and shape of the objects on which the problems are defined. If the BEM is able to solve a problem on a certain object, it is not necessarily able to solve the problem on an object that is slightly larger or

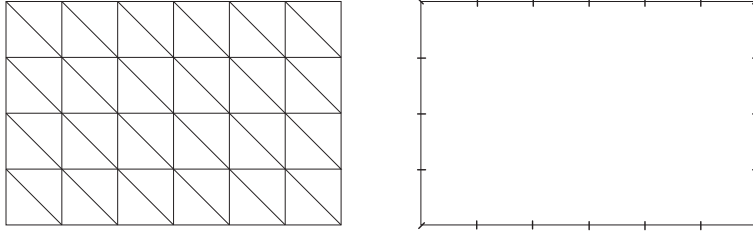


Figure 1.1: *In most numerical methods both the interior and the boundary of a domain need to be discretised (left). In the boundary element method only the boundary needs to be discretised (right).*

smaller. This has its effect on the software that is built upon the BEM. The user of such software must be aware that solutions provided by the software may not be correct. Hence in these cases detailed information on the working of the software, i.e. on the numerical method, is essential in judging the correctness of solutions. This thesis addresses the working of the BEM and investigates under which conditions it produces correct solutions.

1.2 Boundary Element Method

The BEM is a numerical method that approximates solutions of boundary value problems (BVPs). The method is a relatively young method as its birth can be placed in the sixties. Compared to the finite element method (FEM), the development of the BEM has been substantially slower. One reason for this slower development in the BEM is the limited availability of fundamental solutions of the BVPs. Another reason is likely to be the involvement of singular integral equations that need to be solved. Today these equations are well-understood, and the number of application fields in which the BEM is used is large, although not as large as for the finite element method.

The most important aspect in which the BEM distinguishes itself from other numerical methods is the fact that only the boundary of a domain needs to be discretised. In many other numerical methods, such as the FEM, finite differences or the finite volume method, in addition to the boundary, the interior of the domain also needs to be discretised (Figure 1.1). As a consequence of the boundary discretisation, the BEM is a suitable method for problems on external domains, or domains that have a free or moving boundary. Also problems in which singularities or discontinuities occur can be handled efficiently by the BEM. Another advantage of the BEM is that variables and their derivatives, for instance temperature and its flux, are computed with the same degree of accuracy.

Integral equations constitute the foundation of the BEM, and have been known for more than a century. In particular, it is known for a long time that the solutions of BVPs can also be expressed as solutions of an integral equation. As early as 1903 Fredholm already used discretised integral equations for potential problems [43]. His work can be considered the basis for the indirect formulation of the BEM; the functions that appear in the *indirect* formulation do not have a physical meaning, though physical quantities can be derived from these functions. The basis of the *direct* formulation can be placed at Somigliana [86] in 1886, who presented an integral equation relating displacements and stresses. A large number of books and papers have appeared on the subject of integral equations in potential and elasticity theory by mathematicians, such as Kellogg [58], Muskhelishvili [74], Mikhlin [73] and Kupradze [63]. Their results are, however, limited to simple problems as the integral equations have to be solved with analytical procedures and without the aid of computers.

The breakthrough in the development of the BEM came in the nineteen sixties. Jaswon [55] and Symm [90] discretised the integral equations for two-dimensional potential problems by approximating the boundary of a domain by a set of straight lines. At each line element the functions are approximated by constants. Their method has a semi-direct formulation, as the functions need to be differentiated or integrated to obtain physical quantities. A direct formulation has been introduced by Rizzo [80], who also used discretised integral equations to relate displacements and tractions in two-dimensional elasticity theory. The extension to three dimensions has been given by Cruse [31], using triangular elements to describe the domain boundary.

In the late sixties and early seventies the number of applications for which boundary elements are used grew. This constituted a firm foundation for the further development of the BEM and proved that the BEM is a powerful and accurate technique. At this stage attention was also paid to the error and convergence analysis of the BEM. An important contribution came from Hsiao and Wendland [54], who performed such error and convergence analysis for the *Galerkin* formulation of boundary integral equations. As opposed to the Galerkin formulation, the *point collocation* formulation yields easier approximations of integral equations. The error analysis for this type of boundary elements was performed by Arnold, Saranen and Wendland [3, 4, 83] during the eighties.

The first book covering the numerical solution of boundary integral equations has been published by Jaswon and Symm [56] in 1977. Not much later Brebbia [9] used the terminology “Boundary Elements” for the first time as opposed to “Finite Elements”. As of now, the BEM proves to be an effective alternative to solve many engineering problems from a variety of application fields, for instance acoustics, fracture mechanics, potential theory, elasticity theory, viscous flows, thermodynamics, etc.

1.3 Condition number

One way to measure the ability of a numerical method to accurately solve mathematical problems is by monitoring the so-called *condition number*. In this thesis we compute or estimate the condition numbers that appear in the BEM to see whether this method is able to solve mathematical problems accurately.

To explain the meaning of the condition number we first need to address the terms *well-conditioned* and *ill-conditioned*. In general a problem is called well-conditioned if a small change in the input data does not result in a large change in the problem's solution. A problem is called ill-conditioned if a small change in the input data causes a large change in the solution. Depending on how one defines "large" and "small", this classification enables us to divide problems into well-conditioned and ill-conditioned problems. However it does not provide any information on the degree of ill-conditioning. The condition number does precisely that.

Within the setting of this thesis, the condition number ranges from one to infinity. If the condition number is equal to one, then a problem is very well-conditioned. If a problem is singular, the condition number is infinitely large. For problems that approach a singular problem, the condition number approaches infinity. Hence such problems are very ill-conditioned.

In this thesis we study the condition number of linear systems of algebraic equations, which are problems of the form

$$\mathbf{Ax} = \mathbf{b}. \tag{1.1}$$

Such systems are the result of discretising the integral equations that appear in the BEM. The success of solving the linear system depends to a large extent on the condition number of the system matrix \mathbf{A} . If the condition number of this matrix is very large, then the linear system is difficult to solve accurately. Moreover, if the condition number is large, the solution \mathbf{x} is sensitive to perturbations in the input data \mathbf{b} .

It is unclear when the concept of condition number, and related to that the term ill-conditioned, was introduced. In 1948 Turing [92] mentioned that

... the expression 'ill-conditioned' is sometimes used merely as a term of abuse applicable to matrices or equations, but it seems often to carry a meaning somewhat similar to that defined below.

Evidently the term ill-conditioned was already in use at that time. In his paper Turing introduced the norm $N(\mathbf{A})$ and the maximum coefficient $M(\mathbf{A})$ of a matrix \mathbf{A} by

$$\begin{aligned} N(\mathbf{A}) &:= \left(\sum_{i,j} a_{ij}^2 \right)^{1/2}, \\ M(\mathbf{A}) &:= \max_{i,j} |a_{ij}|, \end{aligned} \tag{1.2}$$

in which we recognize $N(\mathbf{A})$ as the Frobenius norm and $M(\mathbf{A})$ as the maximum norm where the matrix \mathbf{A} is seen as a vector of numbers. With these two quantities Turing defined the *N-condition number* as $\frac{1}{n}N(\mathbf{A})N(\mathbf{A}^{-1})$ and the *M-condition number* as $nM(\mathbf{A})M(\mathbf{A}^{-1})$, where n is the size of the matrix. He claimed that these condition numbers are a measure of the degree of ill-conditioning in a matrix. Ironically, the quantity that is now known as the spectral norm of a matrix was also defined by Turing under the name *maximum expansion*. He did not use this quantity to define a related condition number however. Therefore the number that is nowadays referred to as the condition number does not exactly match the condition numbers defined by Turing.

1.4 Condition numbers of the BEM-matrices

Today the BEM is widely used in many application fields. Regrettably the issue of conditioning is often neglected. Usually when solving a BVP with the BEM, it is assumed that the condition number of the resulting system matrix is modest. The question is whether this is true.

First we need to remark that a BVP that is ill-posed will automatically lead to BEM-matrices that have large condition numbers. These BVPs are therefore not the most interesting problems to investigate. A more interesting question is whether well-posed BVPs can lead to BEM-matrices that have large condition numbers. It is this last class of BVPs that we focus on in this thesis.

Until now little attention has been given to the condition number of BEM-matrices. It has been proven that the condition number of the system matrix is at least order N , where N is the number of boundary elements on a two-dimensional domain [97]. This holds for the BEM-matrices that correspond to potential problems with Dirichlet boundary conditions. Similar results are derived by others [21], who have shown that some small modifications to the algebraic set of equations can improve the conditioning of the linear system. In a detailed study for two specific domains, namely the circle and the ellipse, the BEM is applied to the Laplace equation with Dirichlet boundary conditions [19, 20]. For both domains analytical expressions for the condition number of the BEM-matrix are derived. These expressions show the dependence of the condition number on the radius of the circle or the aspect ratio of the ellipse. The Laplace equation on a circle with Dirichlet boundary conditions has been the topic of several other papers [22, 24]; special attention is given to the so-called *local condition number*. It is claimed that this local condition number is a more accurate indicator for the sensitivity of a linear system than the ordinary condition number, which gives too pessimistic estimates of

the sensitivity.

To investigate the condition number of the BEM-matrices, it is useful to study the underlying boundary integral equations (BIEs), which form the basis of the BEM. If the BIE is singular, we may expect that its discrete counterpart, the linear system, is at least ill-conditioned. In that case the condition number is large and the linear system is difficult to solve accurately.

For the BIE arising from the Laplace equation some interesting results can be found in literature. It was observed that the BIE for the 2D Laplace equation with Dirichlet boundary conditions is singular on a domain of certain size [53, 56, 75, 85]. If the BIE is singular, the homogeneous BIE has a non-trivial solution. As a consequence we can add a multiple of this homogeneous solution to the solution of the inhomogeneous BIE, which is henceforth not unique. This introduces an extraordinary phenomenon; the size of a domain affects the uniqueness of the solution of the BIE.

Singular BIEs also occur for BVPs for vector valued functions, for instance for the plane elastostatic problem. By explicitly evaluating the BEM-matrices and computing their condition numbers it is shown that two sizes of the domain exist for which the BIE is not uniquely solvable [51, 62]. This numerical observation is formalized to a general theory, stating that for any 2D domain two sizes exist for which the BIE for the plane elastostatic problem is singular [27, 95]. There exists a number of ways to obtain nonsingular BIEs [50], for instance by using the hypersingular formulation of the BIE for the plane elasticity equations [16]. For this formulation no sizes exist for which the BIE is singular.

In essence the equations for plane elasticity are equal to the Stokes equations for viscous flows in 2D. Hence the developed theory for plane elasticity also applies to the Stokes equations in 2D. This implies that the BIE for the Stokes equations suffers from the same singularities [41].

The BVPs that we mentioned above, i.e. the Laplace equation, the elastostatic equations and the Stokes equations, are well-posed problems when Dirichlet boundary conditions are prescribed. Still, when solved with the BEM, ill-conditioned matrices appear at certain domains. This thesis provides further investigation on this phenomenon.

1.5 Objectives

The objectives of this thesis are twofold. First we want to obtain a better understanding of the conditioning of linear systems that occur in the BEM. The second objective is to study the effectiveness of the BEM for a particular application;

the simulation of the blowing phase in the industrial production process of glass bottles and jars.

1.5.1 Solvability

Almost all research that has been performed on condition numbers and boundary integral equations concerns BVPs with Dirichlet boundary conditions. For BVPs with mixed boundary conditions hardly any results are present. BVPs with mixed boundary conditions is therefore one of the topics of this thesis. We investigate the condition numbers of the matrices that appear when the BEM is used to solve such BVPs.

For the BEM-matrix arising from the Laplace equation with mixed boundary conditions on a circle it is possible to estimate its condition number. We show that this matrix is well-conditioned, except for the unit circle and circles that are close to the unit circle. In these cases the condition number is (infinitely) large.

For the BEM-matrix for the Laplace equation with mixed boundary conditions on an arbitrary 2D domain it can be shown that there is one specific scaling of that domain for which the condition number is infinitely large. The scaling for which this happens is called the *critical scaling* and the corresponding domain the *critical domain*.

The BEM-matrix for the Stokes equations with mixed boundary conditions on an arbitrary domain can also have an infinitely large condition number for certain domains. As the corresponding BIE consists of two equations, there exist two scalings of the domain at which the condition number becomes infinitely large.

There are several ways to avoid the infinitely large condition numbers at critical domains. The simplest remedy is to rescale the domain to another size such that the condition number is bounded. Another option is to add an extra equation to the linear system that guarantees low condition numbers. This extra equation is a compatibility condition that stems from the BVP. A drawback of this option is that we have to solve a system with a rectangular matrix, which requires different solution techniques. A third option is to slightly modify the fundamental solution of the BVP. By including a scaling parameter in this fundamental solution it can be shown that the condition numbers of the BEM-matrices remain bounded at the critical domains.

1.5.2 Blowing of glass

The singular BIEs that we mentioned above typically occur in a 2D setting. This is a direct consequence of the logarithmic nature of the fundamental solution for BVPs that contain the Laplace operator. Hence prudence is called for when one applies the BEM to BVPs on a 2D domain. In a 3D setting the fundamental solution for BVPs with the Laplace operator does not have a logarithmic term. Therefore singular BIEs

similar to those in 2D do not occur in 3D. This allows us to safely apply the BEM to BVPs in 3D that contain the Laplace operator.

As a special application we consider the blowing problem of viscous fluids. In this problem a viscous fluid is positioned in a mould and blown to a desired shape. This blowing process takes place, amongst others, in the industrial manufacturing of glass bottles and jars. The flow of the fluid is governed by the Stokes equations and can be solved with the BEM. This problem typically involves a free boundary. We will investigate whether the BEM is an appropriate numerical method to solve such a problem.

We are aware of several formulations of the BEM. In this thesis we choose for the direct symmetric collocation formulation. The direct formulation involves functions that have a physical meaning, whereas the indirect formulation uses auxiliary functions that have no physical meaning. The symmetric formulation, involving the single and double layer operators, is more commonly used than the non-symmetric formulation, which incorporates the hypersingular operator. We prefer the collocation method above the Galerkin method. Again the collocation method is more commonly used and it does not require a second integration step like the Galerkin method does.

1.6 Outline of the thesis

The thesis starts with an introduction on the BEM in **Chapter 2**. We demonstrate how a BVP can be translated into a BIE, and after discretisation of the domain boundary, into a system of linear equations. We illustrate this for the case of the Laplace equation, but the techniques used to derive the linear system are similar for other BVPs. We also present a number of fundamental results on the boundary integral operators that appear in the BIE.

In the **Chapters 3** and **4** we study the BEM-matrices for the Laplace equation on two-dimensional domains. **Chapter 3** concentrates on the Laplace equation on a circular domain with mixed boundary conditions. The eigenvalues of the corresponding BEM-matrix are approximated, which results in an estimate for the condition number of the BEM-matrix. **Chapter 4** generalizes the results to Laplace equations on arbitrary 2D domains. For this general class of problems it is not possible to estimate the condition number of the BEM-matrix accurately, though it is proven that for certain domains the condition number is infinitely large. This holds for both Laplace equations with Dirichlet conditions and mixed conditions. This phenomenon is confirmed by a number of numerical examples on circles, ellipses, squares and triangles. The large condition numbers can be avoided by making small

modifications to the standard boundary element formulation. We present a number of remedies that guarantee low condition numbers.

The extension to BVPs for vector-valued functions is described in **Chapter 5**. Here we focus on the Stokes equations on a 2D domain. Again it is shown that for certain domains the condition number of the corresponding BEM-matrix becomes infinitely large. This happens both for the Stokes equations with Dirichlet conditions and mixed conditions. We present a number of numerical examples that illustrate this phenomenon. To avoid condition numbers that are infinitely large at certain domains we list several remedies that are more or less similar to the remedies for the Laplacian case.

The domains for which the condition numbers become infinitely large only occur in 2D. Therefore we can safely apply the BEM on a 3D problem. In the **Chapters 6** and **7** we simulate the blowing phase of glass containers. In **Chapter 6** we present the mathematical model that describes this blowing problem. The starting point are the Navier-Stokes equations that describe the flow of a fluid in 3D. These equations can be reduced to the Stokes equations as the fluid is a creeping viscous flow. It is shown how to transform the Stokes equations to a set of BIEs. After discretisation of the domain we obtain a system of linear equations. In this way we can compute the velocity of the glass at any point in time. We use a time integration method to track the position of the glass surface as time evolves. **Chapter 7** gives numerical results for the blowing problem. We simulate the blowing of several containers for a number of different moulds. We also show another application that can be simulated with the help of the mathematical model for the blowing problem; the evolution of viscous drops. Such drops, regardless of their initial size, deform to a spherical drop. We illustrate this process for drops that have the initial shape of an ellipsoide and a beam. We conclude this chapter by investigating the role of the various forces that appear in the blowing problem, such as gravity, surface tension and frictional forces.

Chapter 2

Boundary Element Method

This chapter introduces the basics of the boundary element method (BEM). The method aims at approximating solutions of boundary value problems (BVPs). In particular we use the Laplace equation on a 2D domain as an example to present the BEM formulation. For other BVPs the BEM formulation can be obtained in a similar manner. First we transform the BVP into a boundary integral equation using Green's second identity. Then we discretise the boundary of the domain and obtain a linear system of algebraic equations. Finally we pay attention to the calculation of the matrices that appear in these algebraic equations. The BEM that we present here is the collocation method in a direct formulation, which means that the variables in the method represent physical quantities.

2.1 Integral Equations

We consider a simply connected domain Ω in \mathbb{R}^2 with boundary $\Gamma = \partial\Omega$. Denote by \mathbf{n} the outward normal on Γ . The function $u(\mathbf{x}) = u(x, y)$ for $\mathbf{x} \in \Omega$ is the solution of the Laplace equation, i.e.

$$\nabla^2 u := \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad \mathbf{x} \in \Omega. \quad (2.1)$$

As we will study mixed boundary conditions in this thesis, we divide the boundary Γ into two parts, $\Gamma = \Gamma_u \cup \Gamma_q$. At Γ_u we pose Dirichlet boundary conditions and at Γ_q we pose Neumann conditions. We introduce the notation $q := \partial u / \partial n$ as the normal derivative of u on Γ . Then the BVP for u with mixed boundary conditions reads

$$\begin{cases} \nabla^2 u = 0, & \mathbf{x} \in \Omega, \\ u = \tilde{u}, & \mathbf{x} \in \Gamma_u, \\ q = \tilde{q}, & \mathbf{x} \in \Gamma_q, \end{cases} \quad (2.2)$$

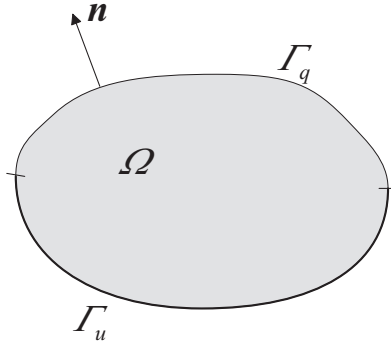


Figure 2.1: The domain Ω with boundary Γ , which is divided into a Dirichlet part Γ_u and a Neumann part Γ_q .

where \tilde{u} and \tilde{q} are known functions representing the boundary data.

Let \mathbf{x}_P and \mathbf{x}_Q be two points in Ω . The Euclidean distance between \mathbf{x}_P and \mathbf{x}_Q is

$$r(\mathbf{x}_P, \mathbf{x}_Q) := \|\mathbf{x}_P - \mathbf{x}_Q\|_2 = \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2}. \quad (2.3)$$

A fundamental solution for the Laplace operator ∇^2 is given by

$$G(\mathbf{x}_P, \mathbf{x}_Q) := \frac{1}{2\pi} \log \frac{1}{r(\mathbf{x}_P, \mathbf{x}_Q)}, \quad \mathbf{x}_P, \mathbf{x}_Q \in \Omega, \quad (2.4)$$

which means that

$$\nabla_Q^2 G(\mathbf{x}_P, \mathbf{x}_Q) = -\delta(\mathbf{x}_P - \mathbf{x}_Q), \quad \mathbf{x}_P, \mathbf{x}_Q \in \Omega. \quad (2.5)$$

The subscript Q of the Laplace operator denotes differentiation to \mathbf{x}_Q and $\delta(\mathbf{x})$ is the Dirac-delta function. Note that $G^\alpha(\mathbf{x}_P, \mathbf{x}_Q) := 1/2\pi \log(\alpha/r(\mathbf{x}_P, \mathbf{x}_Q))$, $\alpha \in \mathbb{R}^+$, is also a fundamental solution for the Laplace operator. The parameter α can be chosen as a characteristic length scale of the domain Ω , thus making the argument of the logarithm dimensionless.

Green's second identity for two functions u and v states that

$$\begin{aligned} \int_{\Omega} \left(u(\mathbf{x}_Q) \nabla^2 v(\mathbf{x}_Q) - v(\mathbf{x}_Q) \nabla^2 u(\mathbf{x}_Q) \right) d\Omega = \\ \int_{\Gamma} \left(u(\mathbf{x}_Q) \frac{\partial v}{\partial n}(\mathbf{x}_Q) - v(\mathbf{x}_Q) \frac{\partial u}{\partial n}(\mathbf{x}_Q) \right) d\Gamma. \end{aligned} \quad (2.6)$$

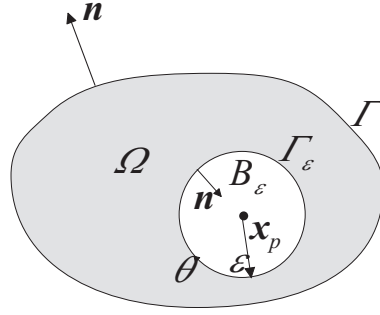


Figure 2.2: The point x_P lies in the interior of the domain.

For u we substitute the solution of the BVP (2.2), and for v we substitute the fundamental solution $G(\mathbf{x}_P, \mathbf{x}_Q)$, where \mathbf{x}_P is regarded as a parameter. As $\nabla^2 u(\mathbf{x}_Q) = 0$ in Ω , Green's second identity yields

$$\int_{\Omega} u(\mathbf{x}_Q) \nabla_Q^2 G(\mathbf{x}_P, \mathbf{x}_Q) d\Omega_Q = \int_{\Gamma} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) - G(\mathbf{x}_P, \mathbf{x}_Q) q(\mathbf{x}_Q) \right) d\Gamma_Q. \quad (2.7)$$

The integrals that appear in this identity must be evaluated carefully, as the fundamental solution G has a logarithmic singularity at $\mathbf{x}_Q = \mathbf{x}_P$. Hence the location of \mathbf{x}_P greatly influences the outcome of the integrals.

First we consider the case that \mathbf{x}_P is in the interior of the domain Ω . In that case we position a small circle B_ε with radius ε and boundary Γ_ε around the point \mathbf{x}_P , such that the entire circle is in the interior of Ω , as is shown in Figure 2.2. In this way G does not have a singularity in the domain $\Omega - B_\varepsilon$ and we have $\nabla_Q^2 G = 0$ in this new domain. Green's second identity applied to the domain $\Omega - B_\varepsilon$ becomes

$$\int_{\Gamma + \Gamma_\varepsilon} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) - G(\mathbf{x}_P, \mathbf{x}_Q) q(\mathbf{x}_Q) \right) d\Gamma_Q = 0, \quad \mathbf{x}_P \in \Omega, \quad (2.8)$$

Note that no domain integrals appear in this identity. The remaining boundary integral consists of two contributions; an integral over Γ and an integral over the circular boundary Γ_ε . The outward normal on Γ_ε is in the direction of the point \mathbf{x}_P . Therefore the local coordinate θ that describes Γ_ε , runs in clockwise direction over

the boundary, $0 \leq \theta < 2\pi$. To calculate the integral over Γ_ε we note that

$$\frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) \Big|_{\mathbf{x}_Q \in \Gamma_\varepsilon} = -\frac{\partial G}{\partial r}(\mathbf{x}_P, \mathbf{x}_Q) \Big|_{\mathbf{x}_Q \in \Gamma_\varepsilon} = \frac{1}{2\pi r} \Big|_{\mathbf{x}_Q \in \Gamma_\varepsilon} = \frac{1}{2\pi\varepsilon}. \quad (2.9)$$

Hence the integral over Γ_ε amounts to

$$\begin{aligned} & \int_{\Gamma_\varepsilon} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) - G(\mathbf{x}_P, \mathbf{x}_Q) q(\mathbf{x}_Q) \right) d\Gamma_Q \\ &= \frac{1}{2\pi} \int_0^{2\pi} \left(u(\mathbf{x}_Q) \frac{1}{\varepsilon} - \log \frac{1}{\varepsilon} q(\mathbf{x}_Q) \right) \varepsilon d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \left(u(\mathbf{x}_Q) + \varepsilon \log \varepsilon q(\mathbf{x}_Q) \right) d\theta. \end{aligned} \quad (2.10)$$

As $\varepsilon \log \varepsilon \rightarrow 0$ for $\varepsilon \downarrow 0$, the last integral approaches $u(\mathbf{x}_P)$ when ε tends to zero. Substituting this in (2.8) we obtain

$$u(\mathbf{x}_P) + \int_{\Gamma} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) - G(\mathbf{x}_P, \mathbf{x}_Q) q(\mathbf{x}_Q) \right) d\Gamma_Q = 0, \quad (2.11)$$

for $\mathbf{x}_P \in \Omega$. This identity relates the values of u in internal points \mathbf{x}_P to values of u and q at the boundary.

Next we consider a point \mathbf{x}_P at the boundary Γ . Again we position a small circle B_ε with radius ε and boundary Γ_ε around the point \mathbf{x}_P . A part of the circle B_ε lies within the domain Ω ; this part is denoted by B'_ε . Likewise, the part of the boundary Γ_ε that lies within Ω is denoted by Γ'_ε , as is shown in Figure 2.3.

We apply Green's second identity on the new domain $\Omega - B'_\varepsilon$. In this domain $\nabla^2 u(\mathbf{x}_Q) = 0$ and $\nabla_Q^2 G(\mathbf{x}_Q, \mathbf{x}_P) = 0$, since the singular point $\mathbf{x}_Q = \mathbf{x}_P$ is outside the domain. The boundary of the domain $\Omega - B'_\varepsilon$ is given by $\Gamma + \Gamma'_\varepsilon - C_\varepsilon$, where C_ε is the part of the boundary Γ that lies within B'_ε . Hence Green's second identity results in

$$\int_{\Gamma + \Gamma'_\varepsilon - C_\varepsilon} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_Q, \mathbf{x}_P) - G(\mathbf{x}_Q, \mathbf{x}_P) q(\mathbf{x}_Q) \right) d\Gamma_Q = 0, \quad (2.12)$$

which can be split into an integral over $\Gamma - C_\varepsilon$ and an integral over Γ'_ε . When ε tends to zero, the integral over $\Gamma - C_\varepsilon$ becomes an integral over the whole boundary Γ , i.e

$$\begin{aligned} & \lim_{\varepsilon \rightarrow 0} \int_{\Gamma - C_\varepsilon} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_Q, \mathbf{x}_P) - G(\mathbf{x}_Q, \mathbf{x}_P) q(\mathbf{x}_Q) \right) d\Gamma_Q \\ &= \int_{\Gamma} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_Q, \mathbf{x}_P) - G(\mathbf{x}_Q, \mathbf{x}_P) q(\mathbf{x}_Q) \right) d\Gamma_Q. \end{aligned} \quad (2.13)$$

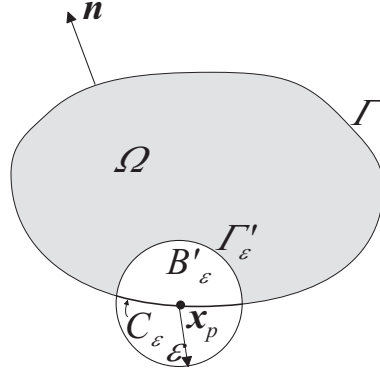


Figure 2.3: The point \mathbf{x}_P lies on the boundary of the domain.

If the boundary is smooth and ε is small, the circle segment Γ'_ε will be the half of the circle Γ_ε . Therefore, when ε tends to zero, the integral over Γ'_ε equals half the integral over Γ_ε , i.e.

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} \int_{\Gamma'_\varepsilon} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_Q, \mathbf{x}_P) - G(\mathbf{x}_Q, \mathbf{x}_P) q(\mathbf{x}_Q) \right) d\Gamma_Q \\ = \frac{1}{2} \int_{\Gamma_\varepsilon} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_Q, \mathbf{x}_P) - G(\mathbf{x}_Q, \mathbf{x}_P) q(\mathbf{x}_Q) \right) d\Gamma_Q. \end{aligned} \quad (2.14)$$

As shown in the case in which \mathbf{x}_P is in the interior of Ω , the latter integral approaches $u(\mathbf{x}_P)$ as $\varepsilon \rightarrow 0$. Substituting this result and (2.13) into (2.12) yields

$$\frac{1}{2} u(\mathbf{x}_P) + \int_{\Gamma} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) - G(\mathbf{x}_P, \mathbf{x}_Q) q(\mathbf{x}_Q) \right) d\Gamma_Q = 0, \quad (2.15)$$

for $\mathbf{x}_P \in \Gamma$. This identity relates the boundary values of u and q . No interior points appear in the equation and thus we have obtained a boundary integral equation. By solving this integral equation we find the values of u and q at the boundary. Knowing these values, equation (2.11) can be used to directly evaluate u at internal points. This is one of the benefits of the BEM, i.e. internal values are obtained without having to solve additional integral equations.

We can summarize the results in (2.11) and (2.15) with the integral equation

$$c(\mathbf{x}_P) u(\mathbf{x}_P) + \int_{\Gamma} \left(u(\mathbf{x}_Q) \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) - G(\mathbf{x}_P, \mathbf{x}_Q) q(\mathbf{x}_Q) \right) d\Gamma_Q = 0, \quad (2.16)$$

where the function $c(\mathbf{x}_P)$ is given by

$$c(\mathbf{x}_P) := \begin{cases} 1, & \mathbf{x}_P \in \Omega, \\ \frac{1}{2}, & \mathbf{x}_P \in \Gamma. \end{cases} \quad (2.17)$$

For $\mathbf{x}_P \in \Omega$ and $\mathbf{x}_Q \in \Gamma$ we introduce two kernel functions,

$$\begin{aligned} K^s(\mathbf{x}_P, \mathbf{x}_Q) &:= G(\mathbf{x}_P, \mathbf{x}_Q) = \frac{1}{2\pi} \log \frac{1}{r(\mathbf{x}_P, \mathbf{x}_Q)}, \\ K^d(\mathbf{x}_P, \mathbf{x}_Q) &:= \frac{\partial G}{\partial n_Q}(\mathbf{x}_P, \mathbf{x}_Q) = \frac{1}{2\pi} \frac{\langle \mathbf{x}_P - \mathbf{x}_Q, \mathbf{n}_Q \rangle}{r^2(\mathbf{x}_P, \mathbf{x}_Q)}, \end{aligned} \quad (2.18)$$

where $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ is the standard inner product. These definitions turn the integral equation into

$$c(\mathbf{x}_P)u(\mathbf{x}_P) + \int_{\Gamma} \left(K^d(\mathbf{x}_P, \mathbf{x}_Q)u(\mathbf{x}_Q) - K^s(\mathbf{x}_P, \mathbf{x}_Q)q(\mathbf{x}_Q) \right) d\Gamma_Q = 0. \quad (2.19)$$

At this point we introduce the *single and double layer potential*, given by

$$(\mathcal{K}^s q)(\mathbf{x}_P) := \int_{\Gamma} K^s(\mathbf{x}_P, \mathbf{x}_Q)q(\mathbf{x}_Q)d\Gamma_Q, \quad (2.20a)$$

$$(\mathcal{K}^d u)(\mathbf{x}_P) := \int_{\Gamma} K^d(\mathbf{x}_P, \mathbf{x}_Q)u(\mathbf{x}_Q)d\Gamma_Q, \quad (2.20b)$$

respectively. The operators \mathcal{K}^s and \mathcal{K}^d are called the *single and double layer operator*. With the potentials, we write the integral equation (2.19) in short-hand notation,

$$(c\mathcal{I} + \mathcal{K}^d)u = \mathcal{K}^s q, \quad (2.21)$$

where \mathcal{I} is the identity operator. In the sequel we assume that the functions u and q are as smooth as is required for the mathematical processes that they are involved in. The boundary integral operators \mathcal{K}^s and \mathcal{K}^d are well-known and their continuity properties have been investigated in detail [29]. Some basic results are presented in the next section.

2.2 Operator theory

The boundary integral operators \mathcal{K}^s and \mathcal{K}^d have been the topic of extensive study. In this section we list some basic results for these operators.

Theorem 2.1 *Let Ω be a bounded domain in \mathbb{R}^2 with smooth boundary Γ . The single layer operator \mathcal{K}^s maps functions from the Sobolev space $H^r(\Gamma)$ isomorphically to $H^{r+1}(\Gamma)$.*

Proof. See [14, p. 258, 287]. ■

Theorem 2.2 *Let Ω be a bounded domain in \mathbb{R}^2 with smooth boundary Γ . The boundary integral operator \mathcal{K} given by*

$$\mathcal{K}f := \int_{\Gamma} \frac{\partial G}{\partial n_y} f(\mathbf{y}) d\Gamma_y + \frac{1}{2}f = (\mathcal{K}^d + \frac{1}{2}\mathcal{I})f, \quad (2.22)$$

is a so-called Fredholm operator with index zero that maps $H^r(\Gamma)$ to $H^r(\Gamma)$.

Proof. See [14, p. 263, 289]. ■

If the Fredholm operator \mathcal{K} has index zero it follows that the kernel of \mathcal{K} and the kernel of its adjoint \mathcal{K}^* have the same dimension. In Chapter 5 we make use of this concept to apply the well-known Fredholm alternative.

Theorem 2.3 *The single layer operator \mathcal{K}^s is a compact and self-adjoint operator.*

Sketch of proof. It can be proven that for k in $L_2([a, b] \times [a, b])$ and satisfying $\overline{k(s, t)} = k(t, s)$ almost everywhere, the integral operator \mathcal{K} defined by

$$(\mathcal{K}f)(t) := \int_a^b k(t, s) f(s) ds \quad (2.23)$$

is compact and self-adjoint on $L_2([a, b])$ [46]. After parameterisation of the boundary, the operator \mathcal{K}^s defined in this chapter can be written in this form, and thus \mathcal{K}^s is compact and self-adjoint. Note that also the single layer operator for the Stokes equations, which will be introduced in Chapter 5, can be written in a similar way as (2.23). Hence the single layer operator for the Stokes equations is a compact and self-adjoint operator. ■

Theorem 2.4 *The eigenvalues of the single layer operator \mathcal{K}^s have an accumulation point 0.*

Sketch of proof. As \mathcal{K}^s is compact and self-adjoint the spectral theorem can be applied, which is formulated as follows [46]. Let \mathcal{K} be a compact self-adjoint

operator on a Hilbert space H . Then there exists an orthonormal system ϕ_1, ϕ_2, \dots of eigenvectors of \mathcal{K} and corresponding eigenvalues $\lambda_1, \lambda_2, \dots$ such that for all $x \in H$,

$$\mathcal{K}x = \sum_k \lambda_k (x, \phi_k) \phi_k, \quad (2.24)$$

where (\cdot, \cdot) is an appropriate inner product on H . If $\{\lambda_k\}$ is an infinite sequence, then it converges to zero. We remark that the spectral theorem can also be applied to the single layer operator for the Stokes equations. ■

The last theorem indicates that the eigenvalues of the boundary integral operator \mathcal{K}^s converge to zero, regardless of the domain Ω . In the next chapter we will analytically compute the eigenvalues of \mathcal{K}^s for a circular domain and we will see that indeed the eigenvalues accumulate at 0.

2.3 Algebraic Equations

To transform the integral equation into a system of algebraic equations, we start with the integral equation as given in (2.19) and choose $\mathbf{x}_P \in \Gamma$,

$$\frac{1}{2}u(\mathbf{x}_P) + \int_{\Gamma} K^d(\mathbf{x}_P, \mathbf{x}_Q)u(\mathbf{x}_Q)d\Gamma_Q = \int_{\Gamma} K^s(\mathbf{x}_P, \mathbf{x}_Q)q(\mathbf{x}_Q)d\Gamma_Q. \quad (2.25)$$

We select N points $\mathbf{y}_k, k = 1, \dots, N$, at the boundary Γ . Two consecutive points are connected by a straight line element Γ_k , which is called a *boundary element*. The center of each element is referred to as a *collocation node* \mathbf{x}_k . Then we replace \mathbf{x}_P in (2.25) by the l -th collocation node $\mathbf{x}_l, l = 1, \dots, N$, and replace the integral over Γ by a sum of integrals over Γ_k , yielding

$$\frac{1}{2}u(\mathbf{x}_l) + \sum_{k=1}^N \int_{\Gamma_k} K^d(\mathbf{x}_l, \mathbf{x}_Q)u(\mathbf{x}_Q)d\Gamma_Q = \sum_{k=1}^N \int_{\Gamma_k} K^s(\mathbf{x}_l, \mathbf{x}_Q)q(\mathbf{x}_Q)d\Gamma_Q, \quad (2.26)$$

for $l = 1, \dots, N$. At each element Γ_k the functions u and q are approximated by the constant coefficients $u_k := u(\mathbf{x}_k)$ and $q_k := q(\mathbf{x}_k)$ respectively. This gives us

$$\frac{1}{2}u_l + \sum_{k=1}^N u_k \int_{\Gamma_k} K^d(\mathbf{x}_l, \mathbf{x}_Q)d\Gamma_Q = \sum_{k=1}^N q_k \int_{\Gamma_k} K^s(\mathbf{x}_l, \mathbf{x}_Q)d\Gamma_Q, \quad (2.27)$$

for $l = 1, \dots, N$. One can also choose to approximate u and q by linear, quadratic, or even higher order functions. This does not make the BEM much more complicated,

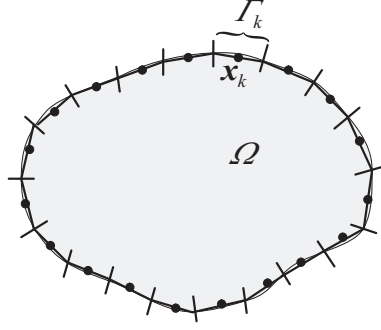


Figure 2.4: The boundary of the domain is approximated by N linear elements.

though the amount of work needed to solve the equations increases. In the subsequent chapters we will use constant or linear approximations only.

We introduce $N \times N$ matrices \mathbf{G} and \mathbf{H} by

$$\begin{aligned} G_{lk} &:= \int_{\Gamma_k} K^s(\mathbf{x}_l, \mathbf{x}_Q) d\Gamma_Q, \quad l, k = 1, \dots, N, \\ H_{lk} &:= \int_{\Gamma_k} K^d(\mathbf{x}_l, \mathbf{x}_Q) d\Gamma_Q, \quad l, k = 1, \dots, N, \end{aligned} \quad (2.28)$$

and vectors \mathbf{u} and \mathbf{q} by

$$\begin{aligned} \mathbf{u} &:= [u_1, \dots, u_N]^T, \\ \mathbf{q} &:= [q_1, \dots, q_N]^T. \end{aligned} \quad (2.29)$$

Then equation (2.27) can be written as

$$\frac{1}{2}\mathbf{u} + \mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q}. \quad (2.30)$$

With $\tilde{\mathbf{H}} := \frac{1}{2}\mathbf{I} + \mathbf{H}$ we have

$$\tilde{\mathbf{H}}\mathbf{u} = \mathbf{G}\mathbf{q}. \quad (2.31)$$

This linear system consists of N algebraic equations, whereas there are $2N$ coefficients, namely the coefficients u_k and q_k at the N elements Γ_k . However, the BVP (2.2) gives us Dirichlet and Neumann boundary conditions for u and q at the boundary. If Γ_k is a boundary element at which a Neumann condition is given, then q_k is a known coefficient while u_k is an unknown coefficient. Vice versa, if Γ_k is a boundary element at which a Dirichlet condition is given, then u_k is known and

q_k unknown. In this way the boundary conditions eliminate N coefficients and we obtain a system of N equations for the remaining N unknown coefficients.

At this point all coefficients u_k are at the left-hand side of (2.31), including the ones that are known. All coefficients q_k are at the right-hand side, including the ones that are unknown. Obviously, we want to have all the unknown coefficients on the same side to solve the equations efficiently. For this goal we need to reorder the equations in a suitable way. The l -th equation of (2.31) is given by

$$\tilde{H}_{l1}u_1 + \tilde{H}_{l2}u_2 + \dots + \tilde{H}_{lN}u_N = G_{l1}q_1 + G_{l2}q_2 + \dots + G_{lN}q_N. \quad (2.32)$$

Suppose that u_1 is given via a Dirichlet condition at the first element Γ_1 while q_1 is unknown. To move the term with u_1 in (2.32) to the right-hand side and the term with q_1 to the left-hand side, we subtract $\tilde{H}_{l1}u_1$ and $G_{l1}q_1$ from the equation, yielding

$$-G_{l1}q_1 + \tilde{H}_{l2}u_2 + \dots + \tilde{H}_{lN}u_N = -\tilde{H}_{l1}u_1 + G_{l2}q_2 + \dots + G_{lN}q_N. \quad (2.33)$$

In this way we move all unknown coefficients to the left-hand side and all known coefficients to the right-hand side. Without loss of generality, we may assume that the first m boundary elements have Dirichlet conditions and the remaining $N - m$ have Neumann conditions. (We can always obtain such a situation by renumbering the elements.) After reordering, the l -th equation is given by

$$\begin{aligned} -G_{l1}q_1 - \dots - G_{lm}q_m + \tilde{H}_{lm+1}u_{m+1} + \dots + \tilde{H}_{lN}u_N \\ = -\tilde{H}_{l1}u_1 - \dots - \tilde{H}_{lm}u_m + G_{lm+1}q_{m+1} + \dots + G_{lN}q_N, \end{aligned} \quad (2.34)$$

We define $N \times N$ matrices \mathbf{A} and $\tilde{\mathbf{G}}$ by

$$\begin{aligned} \mathbf{A} &:= \begin{pmatrix} -G_{11} & \cdots & -G_{1m} & \tilde{H}_{1m+1} & \cdots & \tilde{H}_{1N} \\ \vdots & & \vdots & \vdots & & \vdots \\ -G_{N1} & \cdots & -G_{Nm} & \tilde{H}_{Nm+1} & \cdots & \tilde{H}_{NN} \end{pmatrix}, \\ \tilde{\mathbf{G}} &:= \begin{pmatrix} -\tilde{H}_{11} & \cdots & -\tilde{H}_{1m} & G_{1m+1} & \cdots & G_{1N} \\ \vdots & & \vdots & \vdots & & \vdots \\ -\tilde{H}_{N1} & \cdots & -\tilde{H}_{Nm} & G_{Nm+1} & \cdots & G_{NN} \end{pmatrix}, \end{aligned} \quad (2.35)$$

and vectors \mathbf{x} and \mathbf{b} by

$$\begin{aligned} \mathbf{x} &:= [q_1, \dots, q_m, u_{m+1}, \dots, u_N]^T, \\ \mathbf{b} &:= [u_1, \dots, u_m, q_{m+1}, \dots, q_N]^T. \end{aligned} \quad (2.36)$$

With these definitions the equations can be written in the following way,

$$\mathbf{Ax} = \tilde{\mathbf{G}}\mathbf{b}. \quad (2.37)$$

The proces described above can be formalized by introducing matrices \mathbf{P}_1 and \mathbf{P}_2 ,

$$\mathbf{P}_1 := \begin{bmatrix} \mathbf{I}_m \\ \emptyset \end{bmatrix}, \mathbf{P}_2 := \begin{bmatrix} \emptyset \\ \mathbf{I}_{N-m} \end{bmatrix}, \quad (2.38)$$

where \mathbf{I}_k is the identity matrix of size k . The matrix \mathbf{P}_1 has size $N \times m$ and is a matrix that selects the first m columns from a matrix. Likewise, the matrix \mathbf{P}_2 is of size $N \times (N - m)$ and selects the last $N - m$ columns from a matrix. With these matrices, \mathbf{A} and $\tilde{\mathbf{G}}$ are constructed from $\tilde{\mathbf{H}}$ and \mathbf{G} with

$$\begin{aligned} \mathbf{A} &= [-\mathbf{G}\mathbf{P}_1 \mid \tilde{\mathbf{H}}\mathbf{P}_2], \\ \tilde{\mathbf{G}} &= [-\tilde{\mathbf{H}}\mathbf{P}_1 \mid \mathbf{G}\mathbf{P}_2], \end{aligned} \quad (2.39)$$

By introducing $\mathbf{f} := \tilde{\mathbf{G}}\mathbf{b}$, the system in (2.37) is written as

$$\mathbf{A}\mathbf{x} = \mathbf{f}. \quad (2.40)$$

This notation gives the linear system of equations in the standard form. The matrix \mathbf{A} is a dense matrix, but in many cases the matrix is not very large. For such a matrix a direct solver can be used to solve the linear system.

2.4 Matrix Elements

The matrices \mathbf{A} and $\tilde{\mathbf{G}}$ are constructed from the matrices \mathbf{H} and \mathbf{G} , which are obtained by evaluating the integrals given in (2.28). Suppose that the element Γ_k is a straight line from $\mathbf{x}_0 := (x_0, y_0)^T$ to $\mathbf{x}_1 := (x_1, y_1)^T$. A parameterisation of this element is given by

$$\mathbf{x}(\xi) := \frac{1}{2}(\mathbf{x}_0 + \mathbf{x}_1) + \frac{1}{2}\xi(\mathbf{x}_1 - \mathbf{x}_0), \quad (2.41)$$

where ξ is a local coordinate at the element, $-1 \leq \xi \leq 1$. The Jacobian of the parameterisation is

$$J(\xi) := \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2} = \frac{1}{2}\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} =: \frac{L_k}{2}, \quad (2.42)$$

where L_k is the length of element Γ_k .

Substitution of the parameterisation and the Jacobian in the integrals of (2.28)

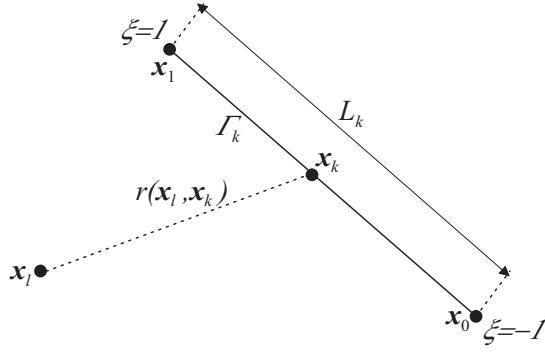


Figure 2.5: The local representation of an element.

yields the following expressions for the matrix elements,

$$\begin{aligned}
 G_{lk} &= \frac{1}{2\pi} \int_{\Gamma_k} \log \frac{1}{r(\mathbf{x}_l, \mathbf{x}_Q)} d\Gamma_Q \\
 &= -\frac{L_k}{4\pi} \int_{-1}^1 \log \left\| \mathbf{x}_l - \frac{\mathbf{x}_0 - \mathbf{x}_1}{2} - \xi \frac{\mathbf{x}_1 - \mathbf{x}_0}{2} \right\| d\xi, \\
 H_{lk} &= \frac{1}{2\pi} \int_{\Gamma_k} \frac{\langle \mathbf{x}_l - \mathbf{x}_Q, \mathbf{n}_Q \rangle}{r^2(\mathbf{x}_l, \mathbf{x}_Q)} d\Gamma_Q \\
 &= \frac{L_k}{4\pi} \int_{-1}^1 \frac{\langle \mathbf{x}_l - \frac{\mathbf{x}_0 - \mathbf{x}_1}{2} - \xi \frac{\mathbf{x}_1 - \mathbf{x}_0}{2}, \mathbf{n}_Q \rangle}{\left\| \mathbf{x}_l - \frac{\mathbf{x}_0 - \mathbf{x}_1}{2} - \xi \frac{\mathbf{x}_1 - \mathbf{x}_0}{2} \right\|^2} d\xi. \tag{2.43}
 \end{aligned}$$

When $l \neq k$, the integrands are nonsingular and we can evaluate the integrals by using standard numerical integration, see Section 2.5.

When $l = k$, the collocation point \mathbf{x}_l is in the center of the element over which is integrated. As a consequence the integrands have a (logarithmic) singularity and we cannot use standard numerical integration schemes. However, in this case we can evaluate the integrals analytically. Since the integrand in G_{ll} is symmetric in r we only need to parametrise one half of the element, i.e.

$$\mathbf{x}(\xi) := \mathbf{x}_l - \xi(\mathbf{x}_l - \mathbf{x}_1), \tag{2.44}$$

where $0 \leq \xi \leq 1$. The Jacobian of this parameterisation is

$$J(\xi) := \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2} = \sqrt{(x_l - x_1)^2 + (y_l - y_1)^2} = \frac{1}{2}L_l. \tag{2.45}$$

This leads to the following expression for the matrix element G_{ll} ,

$$\begin{aligned} G_{ll} &= \frac{1}{2\pi} \int_{\Gamma_l} \log \frac{1}{r(\mathbf{x}_l, \mathbf{x}_Q)} d\Gamma_Q = -2 \cdot \frac{1}{2\pi} \int_0^1 \log\left(\frac{1}{2} L_l \xi\right) d\xi \cdot \frac{1}{2} L_l \\ &= \frac{L_l}{2\pi} \left(1 + \log \frac{2}{L_l}\right). \end{aligned} \quad (2.46)$$

To compute the matrix element H_{ll} , the inner product $\langle \mathbf{x}_l - \mathbf{x}_Q, \mathbf{n}_Q \rangle$ has to be evaluated. Here the vector $\mathbf{x}_l - \mathbf{x}_Q$ coincides with the element Γ_l , as both \mathbf{x}_l and \mathbf{x}_Q are at Γ_l . Hence this vector is perpendicular to the normal \mathbf{n}_Q at the element Γ_l , and consequently $\langle \mathbf{x}_l - \mathbf{x}_Q, \mathbf{n}_Q \rangle = 0$, for all $\mathbf{x}_Q \in \Gamma_l$. This implies that the diagonal elements H_{ll} vanish,

$$H_{ll} = 0. \quad (2.47)$$

2.5 Numerical integration

After parameterisation of the boundary elements, the integrals that have to be evaluated are of the form

$$\int_{-1}^1 f(\xi) d\xi. \quad (2.48)$$

If f is nonsingular on the interval $[-1, 1]$, we may approximate the integral with a standard *Gauss-Legendre quadrature* scheme,

$$\int_{-1}^1 f(\xi) d\xi \approx \sum_{i=1}^m w_i f(\xi_i), \quad (2.49)$$

where ξ_i are the *knots* and w_i the *weights*. If f has a weak or logarithmic singularity at the interval $[-1, 1]$, we either have to resort to analytical expressions as described in the previous section, or use special numerical integration schemes for integrals with weak or logarithmic singularities. For an integral with a kernel that has a logarithmic singularity the following approximation is often used [34, 88],

$$\int_0^1 f(\xi) \log \xi d\xi \approx \sum_{i=1}^m \tilde{w}_i f(\tilde{\xi}_i), \quad (2.50)$$

where $\tilde{\xi}_i$ and \tilde{w}_i are the knots and weights for the quadrature rule with logarithmic weight function. Note that the integration interval for this approximation runs from

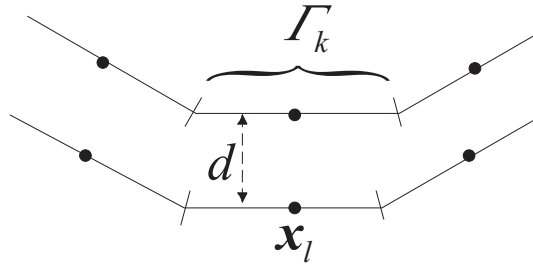


Figure 2.6: Two boundary parts are separated by a distance d .

zero to one, so a transformation of the original integration interval is required to apply this quadrature rule.

One other situation in which the numerical integration needs special attention is a geometry with a thin structure. If two boundaries or parts of a boundary are separated by a small distance d , some integrals become near singular and are difficult to evaluate numerically. This is caused by the fact that, for small d , a collocation node x_l at one boundary approaches an element Γ_k over which is integrated at the other boundary, see Figure 2.6. This phenomenon is called *thin-shape breakdown* (TSB) and has been reported for the Helmholtz boundary integral equation [32, 69]. In this thesis we show that a similar phenomenon occurs for the BIE for the Stokes equations in 2D (Section 5.6). In this case, the Gauss-Legendre quadrature is not accurate enough. A more efficient way to evaluate the integrals is by using an adaptive numerical integration scheme.

Chapter 3

Laplace equation at two-dimensional domain

In the current chapter and the subsequent chapter we investigate the boundary integral equation (BIE) for the Laplace equation on a two-dimensional domain. It turns out that for certain sizes and shapes of the domain this BIE is singular. In this chapter we concentrate on Laplace equations on circular domains. Due to the symmetry of these domains the boundary integral operators can be analysed easily. In particular it is possible to compute the eigenvalues of the integral operators analytically. Moreover, in the case of Dirichlet or Neumann boundary conditions it is possible to use these eigenvalues to derive a accurate estimate for the condition number of the BEM-matrices. Also for mixed boundary conditions an estimate for the condition number of the BEM-matrix can be obtained by combining the information from the Dirichlet and Neumann problems.

3.1 Eigenvalues of \mathcal{K}^s and \mathcal{K}^d

In many BIEs the *single and double layer operator* appear. The analysis of these boundary integral operators is a well-chartered area. Many papers discuss the spectral properties of the Laplace and Helmholtz integral operators as well as the eigenvalues of the corresponding discrete operators [1, 15]. When constructing preconditioners for the BEM-matrices, the spectral properties of the boundary integral operators also need to be addressed [71, 81, 87]. We use the spectral properties of the boundary integral operators to investigate the condition numbers of their discrete counterparts, the BEM-matrices. First we compute the eigenvalues of the operators for a circle.

For a circular domain Ω with radius R it is possible to compute the eigenvalues

of the boundary integral operators \mathcal{K}^s and \mathcal{K}^d analytically. First we introduce polar coordinates (ρ, θ) and (ρ', θ') , and write the points \mathbf{x} and \mathbf{x}' in Ω as

$$\begin{aligned}\mathbf{x} &= \rho (\cos \theta, \sin \theta)^T, \\ \mathbf{x}' &= \rho' (\cos \theta', \sin \theta')^T, \quad 0 < \rho, \rho' < R, \quad 0 < \theta, \theta' < 2\pi.\end{aligned}\quad (3.1)$$

Using these new coordinates the distance $r(\mathbf{x}, \mathbf{x}')$ between two points \mathbf{x} and \mathbf{x}' at the boundary Γ is given by

$$r^2(\mathbf{x}, \mathbf{x}') = 2R^2[1 - \cos(\theta - \theta')], \quad \mathbf{x}, \mathbf{x}' \in \Gamma, \quad (3.2)$$

while the normal $\mathbf{n} \equiv \mathbf{n}(\theta')$ at a point $\mathbf{x}' \in \Gamma$ is given by $\mathbf{n} = [\cos \theta', \sin \theta']^T$. Substitution of these expressions in the single and double layer potentials yields

$$\begin{aligned}(\mathcal{K}^s q)(\theta) &= \frac{-R}{4\pi} \int_0^{2\pi} \left[2 \log R + \log(2 - 2 \cos(\theta - \theta')) \right] q(\theta') d\theta', \\ (\mathcal{K}^d u)(\theta) &= \frac{-1}{4\pi} \int_0^{2\pi} u(\theta') d\theta'.\end{aligned}\quad (3.3)$$

The eigenvalues of the double layer operator \mathcal{K}^d are easily computed. We subsequently insert for u the functions 1 , $\cos(k\theta)$, and $\sin(k\theta)$, $k = 1, 2, \dots$, and find

$$\begin{aligned}\mathcal{K}^d 1 &= -\frac{1}{4\pi} \int_0^{2\pi} d\theta' = -\frac{1}{2}, \\ \mathcal{K}^d \cos(k\theta) &= -\frac{1}{4\pi} \int_0^{2\pi} \cos(k\theta') d\theta' = 0, \\ \mathcal{K}^d \sin(k\theta) &= -\frac{1}{4\pi} \int_0^{2\pi} \sin(k\theta') d\theta' = 0.\end{aligned}\quad (3.4)$$

From this we conclude that $-1/2$ is an eigenvalue of \mathcal{K}^d with eigenfunction $u = 1$, and 0 is an eigenvalue with eigenfunctions $u = \cos(k\theta)$ and $u = \sin(k\theta)$, $k = 1, 2, \dots$

To determine the eigenvalues of the single layer operator \mathcal{K}^s we introduce the function $f(x) := \log(2 - 2 \cos x)$. The Fourier series of this function is

$$f(x) \sim -\sum_{n=1}^{\infty} \frac{2}{n} \cos(nx). \quad (3.5)$$

eigenvalues \mathcal{K}^s	eigenfunctions \mathcal{K}^s	eigenvalues \mathcal{K}^d	eigenfunctions \mathcal{K}^d
$-R \log R$	1	$-\frac{1}{2}$	1
$\frac{R}{2k}$	$\sin(k\theta)$	0	$\sin(k\theta)$
	$\cos(k\theta)$		$\cos(k\theta)$

Table 3.1: The eigenvalues and eigenfunctions of \mathcal{K}^s and \mathcal{K}^d for Ω a circle with radius R .

We set $x := \theta - \theta'$ and substitute the series in the single layer potential to obtain

$$\begin{aligned}
(\mathcal{K}^s q)(\theta) &= -\frac{R}{4\pi} \int_0^{2\pi} \left[2 \log R - \sum_{n=1}^{\infty} \frac{2}{n} \cos n(\theta - \theta') \right] q(\theta') d\theta' \\
&= -\frac{R}{4\pi} \int_0^{2\pi} \left[2 \log R - \sum_{n=1}^{\infty} \frac{2}{n} \left(\cos(n\theta) \cos(n\theta') \right. \right. \\
&\quad \left. \left. + \sin(n\theta) \sin(n\theta') \right) \right] q(\theta') d\theta'. \tag{3.6}
\end{aligned}$$

Also for q we insert the functions 1, $\cos(k\theta)$, and $\sin(k\theta)$, $k = 1, 2, \dots$. Using some well-known results for integrals of products of trigonometric functions, we find

$$\begin{aligned}
\mathcal{K}^s 1 &= -\frac{R}{4\pi} \int_0^{2\pi} 2 \log R d\theta' = -R \log R, \\
\mathcal{K}^s \cos(k\theta) &= -\frac{R}{4\pi} \int_0^{2\pi} \left(-\frac{2}{k} \cos(k\theta) \cos^2(k\theta') \right) d\theta' = \frac{R}{2k} \cos(k\theta), \\
\mathcal{K}^s \sin(k\theta) &= -\frac{R}{4\pi} \int_0^{2\pi} \left(-\frac{2}{k} \sin(k\theta) \sin^2(k\theta') \right) d\theta' = \frac{R}{2k} \sin(k\theta). \tag{3.7}
\end{aligned}$$

From this we conclude that $-R \log R$ is an eigenvalue of \mathcal{K}^s with eigenfunction $q = 1$, and $R/2k$ is an eigenvalue with eigenfunctions $q = \cos(k\theta)$ and $q = \sin(k\theta)$, for $k = 1, 2, \dots$

Table 3.1 gives an overview of the eigenvalues and eigenfunctions of the operators \mathcal{K}^s and \mathcal{K}^d for a circle with radius R , cf. [1, 15].

3.2 Eigenvalues of the matrices

In this section we investigate the eigenvalues of the BEM-matrices \mathbf{G} and \mathbf{H} , which originate from the operators \mathcal{K}^s and \mathcal{K}^d , see Chapter 2. In particular we are interested in the correspondence between the eigenvalues of the integral operators and the eigenvalues of the matrices.

First we investigate the correspondence between the eigenvalues of the integral operator \mathcal{K}^s and the eigenvalues of \mathbf{G} . We do this for the case $R = 1$. Using the results from the previous section, we know that the integral operator has the following eigenvalues,

$$\lambda_k(\mathcal{K}^s) \in \{1/2, 1/4, 1/6, 1/8, \dots, 0\}. \quad (3.8)$$

N	$\lambda_1(\mathbf{G})$	$\lambda_2(\mathbf{G})$	$\lambda_3(\mathbf{G})$	$\lambda_4(\mathbf{G})$	$\lambda_N(\mathbf{G})$
8	0.5122	0.2472	0.1686	0.1482	$-2.0 \cdot 10^{-2}$
16	0.5031	0.2480	0.1627	0.1207	$-3.9 \cdot 10^{-3}$
32	0.5008	0.2493	0.1652	0.1230	$-8.9 \cdot 10^{-4}$
64	0.5002	0.2498	0.1663	0.1244	$-2.1 \cdot 10^{-4}$
128	0.5000	0.2500	0.1666	0.1248	$-5.1 \cdot 10^{-5}$

Table 3.2: The four largest eigenvalues of \mathbf{G} and the smallest for $R = 1$.

Let N be the number of boundary elements at Γ . Then \mathbf{G} has size $N \times N$ and has N eigenvalues $\lambda_1(\mathbf{G}) \geq \dots \geq \lambda_N(\mathbf{G})$. In Table 3.2 we give the four largest eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and λ_4 and the smallest eigenvalue λ_N of the matrix \mathbf{G} for several values of N . We observe that the eigenvalues of \mathbf{G} approximate the eigenvalues of \mathcal{K}^s . The numerical test shows that the largest eigenvalue of \mathbf{G} converges to the corresponding eigenvalue of \mathcal{K}^s with $\mathcal{O}(N^{-2})$. The other eigenvalues in Table 3.2 converge to the corresponding eigenvalues of \mathcal{K}^s slower. In general the convergence for the smallest eigenvalue is the slowest.

Further on in this chapter we need the eigenvalues of \mathbf{G} to compute the condition number of \mathbf{G} , in particular the largest and smallest eigenvalue. Table 3.2 shows that we can approximate these eigenvalues by the corresponding eigenvalues of \mathcal{K}^s . We observe that the eigenvalues are approximated rather well.

In Figure 3.1 we show the relative error made by approximating the largest and smallest eigenvalues of \mathbf{G} by the largest and smallest eigenvalues of \mathcal{K}^s . In this case we choose $R = 2$. We see that the error for the largest eigenvalue decreases rapidly to zero. The error for the smallest eigenvalue is much larger, approximately 16%. Later

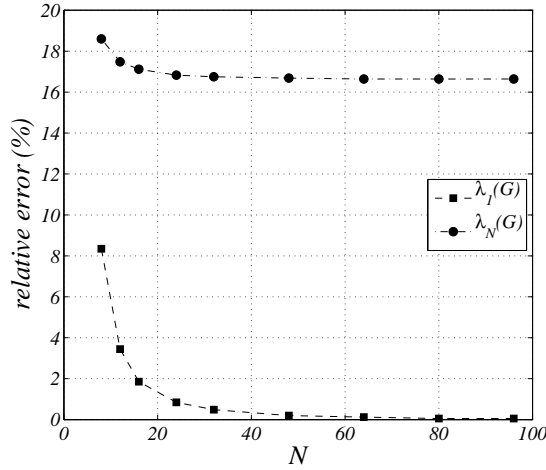


Figure 3.1: The errors for approximating the largest and smallest eigenvalue of \mathbf{G} by the largest and smallest eigenvalue of \mathcal{K}^s for $R = 2$.

on we will see which influence these errors have on the estimates of the condition number of the system matrices.

The results from Section 3.1 show that the boundary integral operator \mathcal{K}^d has only two distinct eigenvalues. Let $\lambda_1(\mathbf{H}) \geq \dots \geq \lambda_N(\mathbf{H})$ be the eigenvalues of \mathbf{H} , then we may expect that the first $N - 1$ eigenvalues will be (almost) equal. Hence it is sufficient to study $\lambda_1(\mathbf{H})$ and $\lambda_N(\mathbf{H})$. The eigenvalues of the integral operator \mathcal{K}^d are independent of R , so we may expect that the eigenvalues of the corresponding BEM-matrix \mathbf{H} are also independent of R . Hence we choose $R = 1$ and compute the eigenvalues of \mathbf{H} for several values of N . Table 3.3 shows the largest and smallest eigenvalue of \mathbf{H} , which approach the eigenvalues of \mathcal{K}^d as N goes to infinity. The numerical test shows that the rate by which this happens is $1/N$. Hence the eigenvalues of \mathcal{K}^d provide accurate estimates of the eigenvalues of the BEM-matrix \mathbf{H} . However, for the circular domain we can even compute the eigenvalues of \mathbf{H} analytically. Indeed, for the matrix elements of \mathbf{H} we find for $l \neq k$,

$$H_{lk} = \int_{\Gamma_k} K^d(\mathbf{x}_P, \mathbf{x}_Q) d\Gamma_Q = -\frac{1}{4\pi R} \int_{\Gamma_k} d\Gamma_Q = -\frac{1}{4\pi R} L_k, \quad (3.9)$$

with L_k the length of element Γ_k . Note that all elements have equal length, namely $L_k = 2R \tan(\pi/N)$. Substituting this in the matrix elements above results in

$$H_{lk} = -\frac{1}{2\pi} \tan \frac{\pi}{N}, \quad l \neq k. \quad (3.10)$$

N	$\lambda_N(\mathbf{H})$	$\lambda_1(\mathbf{H})$	abs. error $\lambda_1(\mathbf{H})$
8	$6.59 \cdot 10^{-2}$	-0.462	$3.85 \cdot 10^{-2}$
16	$3.17 \cdot 10^{-2}$	-0.475	$2.51 \cdot 10^{-2}$
32	$1.57 \cdot 10^{-2}$	-0.486	$1.41 \cdot 10^{-2}$
64	$7.82 \cdot 10^{-3}$	-0.493	$7.40 \cdot 10^{-3}$
128	$3.91 \cdot 10^{-3}$	-0.496	$3.80 \cdot 10^{-3}$
256	$1.95 \cdot 10^{-3}$	-0.498	$1.90 \cdot 10^{-3}$

Table 3.3: The eigenvalues of \mathbf{H} for $R = 1$.

Recall that the diagonal elements of \mathbf{H} are equal to zero (see (2.47)). Thus the matrix \mathbf{H} has a very simple structure, namely zeros on the diagonal, and the same non-zero number in all off-diagonal elements. For such a matrix the eigenvalues can be computed analytically, which results in

$$\begin{aligned}\lambda_1(\mathbf{H}) &= -\frac{N-1}{2\pi} \tan \frac{\pi}{N} \approx -\frac{1}{2} + \frac{1}{2N} + \mathcal{O}\left(\frac{1}{N^3}\right), \\ \lambda_N(\mathbf{H}) &= \frac{1}{2\pi} \tan \frac{\pi}{N} \approx \frac{1}{2N} + \mathcal{O}\left(\frac{1}{N^3}\right).\end{aligned}\quad (3.11)$$

One may verify that these expressions give the same eigenvalues as presented in Table 3.3.

3.3 Dirichlet problem

In the subsequent sections we investigate the condition number of the matrices that appear in the BEM. For a general $N \times N$ matrix \mathbf{A} , the condition number is defined as the ratio of the largest and smallest singular value,

$$\text{cond}(\mathbf{A}) := \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})}. \quad (3.12)$$

For symmetric matrices the singular values are equal to the absolute values of the eigenvalues. Hence when \mathbf{A} is symmetric, the condition number is computed as

$$\text{cond}(\mathbf{A}) := \frac{\max |\lambda(\mathbf{A})|}{\min |\lambda(\mathbf{A})|}. \quad (3.13)$$

Consider the Laplace equation on the circle with Dirichlet boundary conditions. In this case, $u = \tilde{u}$ is prescribed at the whole boundary. The BIE (2.21) reduces to

$$\mathcal{K}^s q = f. \quad (3.14)$$

Here $f := (\frac{1}{2}\mathcal{I} + \mathcal{K}^d)\tilde{u}$ is a known function depending on the boundary data \tilde{u} . Similarly the algebraic equations (2.30) reduce to

$$\mathbf{G}\mathbf{q} = \mathbf{f}, \quad (3.15)$$

where $\mathbf{f} := (\frac{1}{2}\mathbf{I} + \mathbf{H})\mathbf{u}$ is a known vector. As the N elements and nodes are uniformly distributed over the boundary Γ , the matrix \mathbf{G} is symmetric. In that case the condition number of \mathbf{G} can be computed as the ratio of largest and smallest eigenvalue of \mathbf{G} . Let N be even. Then the N eigenvalues of \mathbf{G} may be approximated by the first N eigenvalues of \mathcal{K}^s ,

$$-R \log R, \quad \frac{R}{2k}, \quad \frac{R}{N}, \quad (3.16)$$

with $k = 1, 2, \dots, N/2 - 1$ and where the eigenvalues $R/2k$ have geometric multiplicity two and the other eigenvalues geometric multiplicity one. The condition number can thus be approximated by

$$\begin{aligned} \text{cond}(\mathbf{G}) &\approx \frac{\max\left(\left(\max_{1 \leq k \leq N/2-1} \frac{R}{2k}\right), R |\log R|\right)}{\min\left(\left(\min_{1 \leq k \leq N/2-1} \frac{R}{2k}\right), R |\log R|\right)} \\ &= \frac{\max\left(\frac{1}{2}, |\log R|\right)}{\min\left(\frac{1}{N}, |\log R|\right)}. \end{aligned} \quad (3.17)$$

In Figure 3.2 we plot the approximation of (3.17) as a function of the radius R for four different values of N : 4, 8, 12, and 16. Note that the behaviour of the condition number as shown in the figure is in good agreement with the results in literature [19, 20]. For $R = 1$ the condition number jumps to infinity. This implies that the linear system $\mathbf{G}\mathbf{q} = \mathbf{f}$ is singular for the unit circle. In Chapter 4 we elaborate on modifications of the standard BEM formulation to avoid such singular systems. For $R \rightarrow 0$ the condition number also increases to infinity, reflecting the equations becoming singular when the domain shrinks to a single point. In Figure 3.2 a number of regimes can be distinguished, in which the behaviour of the condition number is different. To distinguish these regimes we write the estimate of the condition number in (3.17) as

$$\text{cond}(\mathbf{G}) \approx \begin{cases} \frac{1}{2|\log R|} & e^{-1/N} \leq R < 1 & \text{and } 1 < R \leq e^{1/N}, \\ \frac{N}{2} & e^{-1/2} \leq R \leq e^{-1/N} & \text{and } e^{1/N} \leq R \leq e^{1/2}, \\ N|\log R| & 0 \leq R \leq e^{-1/2} & \text{and } e^{1/2} \leq R < \infty. \end{cases} \quad (3.18)$$

To study the accuracy of the approximation in (3.17), we choose $R = 2$ and $N \geq 2$. In that case $e^{1/2} \leq R < \infty$ and the condition number of \mathbf{G} is estimated by

$$\text{cond}(\mathbf{G}) \approx N \log 2. \quad (3.19)$$

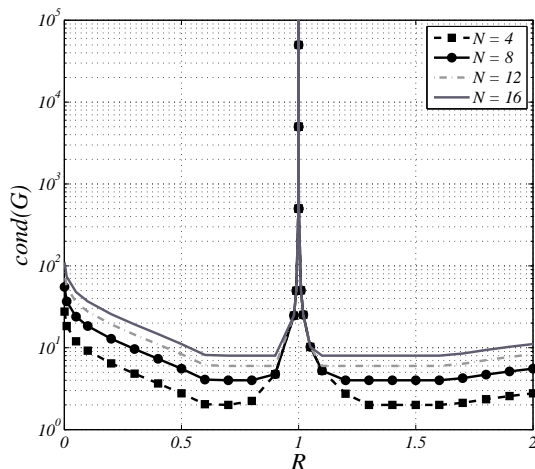


Figure 3.2: The approximation of the condition number of \mathbf{A} as a function of the radius R for several values of N .

N	$\text{cond}(\mathbf{G})$	estimate $N \log R$	error (%)	$\text{cond}(\mathbf{G})$ for modified fund. sol.	estimate $N/2$	error (%)
8	5.06	5.55	9.7	3.68	4	8.0
16	9.65	11.09	14.9	6.96	8	14.9
32	19.09	22.18	16.2	13.75	16	16.4
64	38.07	44.36	16.5	27.44	32	16.6
128	76.09	88.72	16.6	54.88	64	16.6

Table 3.4: The exact condition number of \mathbf{G} , and its approximation, for standard BEM and for BEM with a modified fundamental solution.

In the second and third column of Table 3.4, we give the true value of condition number of \mathbf{G} and its estimate for several values of N . In the fourth column we give the relative error between true value and its estimate. This error is related to the difference between the eigenvalues of the matrix \mathbf{G} and the eigenvalues of the operator \mathcal{K}^s .

In Figure 3.2 and (3.18) we observe that the condition number of \mathbf{G} has a minimal value of $N/2$ for $e^{-1/2} \leq R \leq e^{-1/N}$ and $e^{1/N} \leq R \leq e^{1/2}$. By rescaling the circle such that its new radius is in one of these two intervals, the condition number can be minimized. Another way to minimize the condition number, is by modifying the

fundamental solution of the Laplace operator by including a factor α ,

$$G^\alpha(r) = \frac{1}{2\pi} \log\left(\frac{\alpha}{r}\right). \quad (3.20)$$

This changes the eigenvalue $-R \log R$ of \mathcal{K}^s to $-R \log(R/\alpha)$. In that case the estimate for the condition number of \mathbf{G} becomes

$$\text{cond}(\mathbf{G}) \approx \frac{\max\left(\frac{1}{2}, |\log R/\alpha|\right)}{\min\left(\frac{1}{N}, |\log R/\alpha|\right)}. \quad (3.21)$$

By choosing $\alpha = Re^{-1/2}$ the nominator reduces to $1/2$ and the denominator to $1/N$. As a consequence we have $\text{cond}(\mathbf{G}) \approx N/2$, which is the smallest value reached in Figure 3.2. This agrees with the theory that the condition number of the BEM-matrix for a Laplace equation with Dirichlet conditions on any 2D domain can be minimized to $\mathcal{O}(N)$ [97]. In the last three columns of Table 3.4 we show the effect of the strategy of modifying the fundamental solution. We give the true condition number of \mathbf{G} with modified fundamental solution, the corresponding estimate of $N/2$, and the relative error. We observe that the condition number for the new matrix is approximately 25% smaller than the condition number of the original matrix. The most important gain however is that the condition number does not become infinitely large anymore.

3.4 Neumann problem

In this section we study the Laplace equation on a circle with Neumann boundary conditions. In this case $q = \tilde{q}$ is known at the whole boundary and the BIE (2.21) reduces to

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{K}^d\right)u = f. \quad (3.22)$$

Here $f := \mathcal{K}^s \tilde{q}$ is a known function depending on the boundary data. Similarly the algebraic equations (2.30) reduce to

$$\tilde{\mathbf{H}}\mathbf{u} = \mathbf{f}, \quad (3.23)$$

where $\mathbf{f} := \mathbf{G}\mathbf{q}$ is a known vector. As the N elements and nodes are uniformly distributed over the boundary Γ , the matrix $\tilde{\mathbf{H}}$ is symmetric. Hence the singular values of $\tilde{\mathbf{H}}$ are equal to the absolute values of the eigenvalues of $\tilde{\mathbf{H}}$, and consequently

$$\text{cond}(\tilde{\mathbf{H}}) = \frac{\sigma_{\max}(\tilde{\mathbf{H}})}{\sigma_{\min}(\tilde{\mathbf{H}})} = \frac{\lambda_1(\tilde{\mathbf{H}})}{\lambda_N(\tilde{\mathbf{H}})}. \quad (3.24)$$

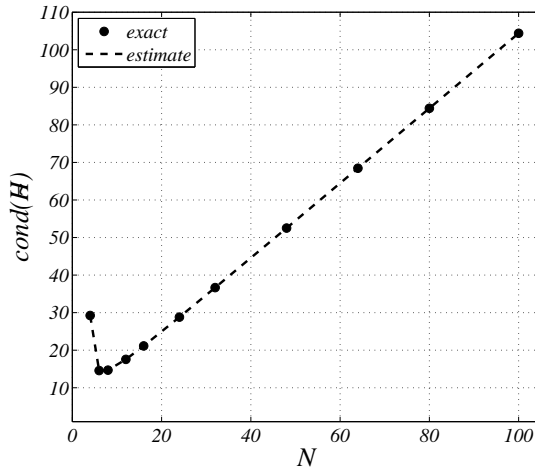


Figure 3.3: The condition number of $\tilde{\mathbf{H}}$ as a function of N .

Note that both $\lambda_1(\tilde{\mathbf{H}})$ and $\lambda_N(\tilde{\mathbf{H}})$ are positive. As the exact eigenvalues of \mathbf{H} are known and $\tilde{\mathbf{H}} = \frac{1}{2}\mathbf{I} + \mathbf{H}$ we thus find

$$\begin{aligned}\lambda_1(\tilde{\mathbf{H}}) &= \frac{1}{2} + \frac{1}{2\pi} \tan\left(\frac{\pi}{N}\right), \\ \lambda_N(\tilde{\mathbf{H}}) &= \frac{1}{2} - \frac{N-1}{2\pi} \tan\left(\frac{\pi}{N}\right).\end{aligned}\quad (3.25)$$

Consequently the condition number of $\tilde{\mathbf{H}}$ is equal to

$$\text{cond}(\tilde{\mathbf{H}}) = \frac{\pi + \tan \pi/N}{|\pi - (N-1) \tan \pi/N|}. \quad (3.26)$$

In Figure 3.3 we show the condition number of $\tilde{\mathbf{H}}$ as a function of N . For $N \geq 10$, the condition number shows a strong linear behaviour in N . Realizing that for large N we have $\tan \pi/N \approx \pi/N$, we find for the condition number

$$\text{cond}(\tilde{\mathbf{H}}) \approx \frac{\pi + \pi/N}{\pi - (N-1)\pi/N} = N + 1, \quad (3.27)$$

which confirms the linear behaviour from the figure.

The Laplace equation with Neumann boundary conditions is in essence an ill-posed problem. This is reflected by the zero eigenvalue of the boundary integral operator $\frac{1}{2}\mathcal{I} + \mathcal{K}^d$; the corresponding BIE (3.22) is singular. Hence we may expect a singular system for the discrete equations $\tilde{\mathbf{H}}\mathbf{u} = \mathbf{f}$, i.e. an infinitely large condition

number of $\tilde{\mathbf{H}}$. Nevertheless, the condition number of $\tilde{\mathbf{H}}$ only grows linearly with N . This is a consequence of the discretisation of the problem. The smallest eigenvalue of the discrete problem, i.e. the smallest eigenvalue of $\tilde{\mathbf{H}}$, is not exactly equal to zero, but approaches zero as the discretisation of the boundary becomes finer, i.e. the number of elements N increases.

3.5 Mixed boundary conditions

In this section we consider the Laplace equation on a circle with mixed boundary conditions. We assume that the first m elements of the boundary have Dirichlet boundary conditions and the last $N - m$ elements Neumann conditions. We can always reach such a situation by renumbering the elements. The BIE reads

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{K}^d\right)u = \mathcal{K}^s q, \quad (3.28)$$

while the set of algebraic equations is given by

$$\left(\frac{1}{2}\mathbf{I} + \mathbf{H}\right)\mathbf{u} = \mathbf{G}\mathbf{q}. \quad (3.29)$$

As described in Section 2.3 the latter equations can be written as

$$\mathbf{A}\mathbf{x} = \mathbf{f}, \quad (3.30)$$

where the matrix \mathbf{A} is constructed from the matrices \mathbf{G} and $\tilde{\mathbf{H}}$ by

$$\mathbf{A} = [-\mathbf{G}\mathbf{P}_1 \mid \tilde{\mathbf{H}}\mathbf{P}_2]. \quad (3.31)$$

In this section we derive an estimate for the condition number of the BEM-matrix \mathbf{A} .

Due to the symmetry of the boundary discretisation the matrices \mathbf{G} and $\tilde{\mathbf{H}}$ are *circulant* matrices [33]. Given the first row of such a matrix, one obtains the other rows by a cyclic shift of the first row. An important property of a circulant matrix \mathbf{X} is that it can be decomposed as $\mathbf{X} = \mathbf{F}^* \mathbf{\Lambda} \mathbf{F}$, where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of \mathbf{X} . The matrix \mathbf{F} is the so-called *Fourier matrix*, whose elements are defined by

$$F_{lk}^* := \frac{1}{\sqrt{N}} w^{(l-1)(k-1)}, \quad (3.32)$$

The asterisk denotes complex conjugation and $w := e^{2\pi i/N}$ is the N -th root of unity. The Fourier matrix \mathbf{F} is a unitary matrix. We apply the decomposition property of circulant matrices to \mathbf{G} and $\tilde{\mathbf{H}}$,

$$\begin{aligned} \mathbf{G} &= \mathbf{F}^* \mathbf{\Lambda}_G \mathbf{F}, \\ \tilde{\mathbf{H}} &= \mathbf{F}^* \mathbf{\Lambda}_H \mathbf{F}. \end{aligned} \quad (3.33)$$

Here Λ_G and Λ_H are diagonal matrices containing the eigenvalues of \mathbf{G} and $\tilde{\mathbf{H}}$ respectively. Here the eigenvalues of \mathbf{G} are replaced by the eigenvalues of \mathcal{K}^s and for the eigenvalues of $\tilde{\mathbf{H}}$ we use the exact expressions in (3.25). Substituting the decompositions for \mathbf{G} and $\tilde{\mathbf{H}}$ in (3.31), we write \mathbf{A} as

$$\mathbf{A} = \mathbf{F}^*[-\Lambda_G \mathbf{F} \mathbf{P}_1 \mid \Lambda_H \mathbf{F} \mathbf{P}_2]. \quad (3.34)$$

We define $\mathbf{F}_1 := \mathbf{F} \mathbf{P}_1$ and $\mathbf{F}_2 := \mathbf{F} \mathbf{P}_2$ to find

$$\mathbf{A} = \mathbf{F}^*[-\Lambda_G \mathbf{F}_1 \mid \Lambda_H \mathbf{F}_2]. \quad (3.35)$$

By introducing two other diagonal matrices Λ and \mathbf{D} by $\Lambda := \Lambda_G^{1/2} \Lambda_H^{-1/2}$ and $\mathbf{D} := \Lambda_G^{1/2} \Lambda_H^{1/2}$, we obtain *

$$\mathbf{A} = \mathbf{F}^* \mathbf{D}[-\Lambda \mathbf{F}_1 \mid \Lambda^{-1} \mathbf{F}_2]. \quad (3.36)$$

We also introduce QR-decompositions of $\Lambda \mathbf{F}_1$ and $\Lambda^{-1} \mathbf{F}_2$ as

$$\begin{aligned} \Lambda \mathbf{F}_1 &= \mathbf{Q}_1 \mathbf{U}_1, \\ \Lambda^{-1} \mathbf{F}_2 &= \mathbf{Q}_2 \mathbf{U}_2. \end{aligned} \quad (3.37)$$

The columns of \mathbf{Q}_1 and \mathbf{Q}_2 form bases of the subspaces which are spanned by the columns of $\Lambda \mathbf{F}_1$ and $\Lambda^{-1} \mathbf{F}_2$. The matrices \mathbf{U}_1 and \mathbf{U}_2 are upper triangular matrices. With these decompositions \mathbf{A} can be written as

$$\mathbf{A} = \mathbf{F}^* \mathbf{D} \underbrace{\begin{bmatrix} -\mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} \mathbf{U}_1 & \emptyset \\ \emptyset & \mathbf{U}_2 \end{bmatrix}}_{\mathbf{U}} = \mathbf{F}^* \mathbf{D} \mathbf{Q} \mathbf{U}. \quad (3.38)$$

Since the unitary matrix \mathbf{F} has condition number equal to 1 we find

$$\text{cond}(\mathbf{A}) \leq \text{cond}(\mathbf{D}) \text{cond}(\mathbf{Q}) \text{cond}(\mathbf{U}). \quad (3.39)$$

Hence to bound the condition number of \mathbf{A} we need estimates of the condition numbers of the matrices \mathbf{D} , \mathbf{Q} and \mathbf{U} .

Estimating $\text{cond}(\mathbf{D})$

The matrix \mathbf{D} is the product of two diagonal matrices of which we can approximate or determine the singular values, namely Λ_G and Λ_H . For convenience we list the

*Defining Λ and \mathbf{D} as the square root of Λ_G and Λ_H may yield complex numbers when a diagonal element of Λ_G or Λ_H is negative. However, in order to estimate the condition numbers, we only need to evaluate the *squared* singular values of Λ and \mathbf{D} , thus avoiding complex numbers.

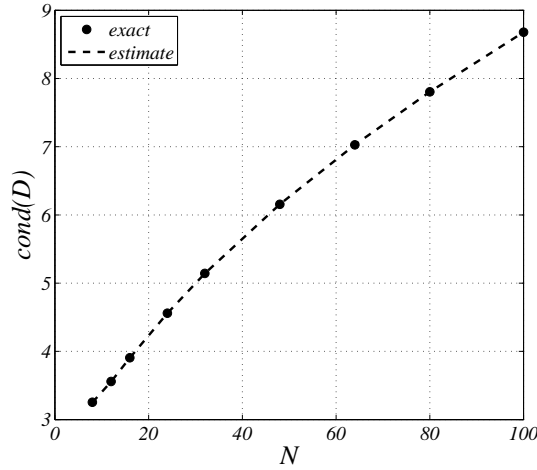


Figure 3.4: Condition number of \mathbf{D} as a function of N with $R = 1/2$ and $m = N/2$. Dots represent the exact values while the dashed line represents the estimate.

largest and smallest singular value of \mathbf{D} squared,

$$\begin{aligned}
 \sigma_1(\mathbf{D})^2 &= \sigma_1 \left(\mathbf{\Lambda}_G^{1/2} \mathbf{\Lambda}_H^{1/2} \right)^2 = \sigma_1 \left(\mathbf{\Lambda}_G \mathbf{\Lambda}_H \right)^2 = \max_l \left[\mathbf{\Lambda}_G \mathbf{\Lambda}_H \right]_{ll} \\
 &= \max \left\{ R |\log R| \left[\frac{1}{2} - \frac{N-1}{2\pi} \tan \frac{\pi}{N} \right], \frac{R}{4} + \frac{R}{4\pi} \tan \frac{\pi}{N} \right\}, \\
 \sigma_N(\mathbf{D})^2 &= \sigma_N \left(\mathbf{\Lambda}_G^{1/2} \mathbf{\Lambda}_H^{1/2} \right)^2 = \sigma_N \left(\mathbf{\Lambda}_G \mathbf{\Lambda}_H \right)^2 = \min_l \left[\mathbf{\Lambda}_G \mathbf{\Lambda}_H \right]_{ll} \\
 &= \min \left\{ R |\log R| \left[\frac{1}{2} - \frac{N-1}{2\pi} \tan \frac{\pi}{N} \right], \frac{R}{2N} + \frac{R}{2N\pi} \tan \frac{\pi}{N} \right\}.
 \end{aligned} \tag{3.40}$$

The condition number of \mathbf{D} is the square root of the ratio of these two expressions. Figure 3.4 shows the condition number of \mathbf{D} as a function of N . The dots give the exact value of the condition number while the dashed line represents the estimate as constructed in this section. We observe that there is a very good correspondence between exact values and the estimated values.

Estimating $\text{cond}(\mathbf{Q})$

The *Kantorovich-Wielandt angle* θ is found by taking pairs of orthogonal vectors \mathbf{x} and \mathbf{y} and calculating the smallest angle between their images under \mathbf{Q} [48],

$$\cos \theta := \max_{\mathbf{x} \perp \mathbf{y}} \frac{|(\mathbf{Q}\mathbf{x}, \mathbf{Q}\mathbf{y})|}{\|\mathbf{Q}\mathbf{x}\| \|\mathbf{Q}\mathbf{y}\|}. \quad (3.41)$$

The condition number of \mathbf{Q} is related to the Kantorovich-Wielandt angle by

$$\text{cond}(\mathbf{Q}) = \text{atan}(\theta/2), \quad (3.42)$$

It can be proven that the angle θ is the angle between the two subspaces spanned by the columns of $\Lambda \mathbf{F}_1$ and $\Lambda^{-1} \mathbf{F}_2$.

Lemma 3.1 *The Kantorovich-Wielandt angle θ is equal to the angle between the two subspaces spanned by the columns of $\Lambda \mathbf{F}_1$ and $\Lambda^{-1} \mathbf{F}_2$.*

Proof. The angle α between the two subspaces that are spanned by the columns of $\Lambda \mathbf{F}_1$ and $\Lambda^{-1} \mathbf{F}_2$ is defined as [8]

$$\cos \alpha := \max_{\xi_1 \in \mathcal{R}(\Lambda \mathbf{F}_1)} \max_{\xi_2 \in \mathcal{R}(\Lambda^{-1} \mathbf{F}_2)} \frac{|(\xi_1, \xi_2)|}{\|\xi_1\| \|\xi_2\|}. \quad (3.43)$$

To evaluate the Kantorovich-Wielandt angle we realize the following. The matrix \mathbf{Q} consists of two blocks, and therefore we select two special vectors \mathbf{x} and \mathbf{y} , namely $\mathbf{x} = [\mathbf{x}_1^T \mid 0, \dots, 0]^T$ and $\mathbf{y} = [0, \dots, 0 \mid \mathbf{y}_1^T]^T$, where $\mathbf{x}_1 \in \mathbb{R}^m$ and $\mathbf{y}_1 \in \mathbb{R}^{N-m}$. Clearly we have $\mathbf{x} \perp \mathbf{y}$. Moreover, we observe that $\mathbf{Q}\mathbf{x} = -\mathbf{Q}_1 \mathbf{x}_1$ and $\mathbf{Q}\mathbf{y} = \mathbf{Q}_2 \mathbf{y}_1$. We substitute this into the definition of the Kantorovich-Wielandt angle and find

$$\cos \theta = \max_{\mathbf{x}_1 \in \mathbb{R}^m} \max_{\mathbf{y}_1 \in \mathbb{R}^{N-m}} \frac{|(\mathbf{Q}_1 \mathbf{x}_1, \mathbf{Q}_2 \mathbf{y}_1)|}{\|\mathbf{Q}_1 \mathbf{x}_1\| \|\mathbf{Q}_2 \mathbf{y}_1\|}. \quad (3.44)$$

Recall that the columns of the matrices \mathbf{Q}_1 and \mathbf{Q}_2 form an orthogonal basis for the subspaces spanned by the columns of $\Lambda \mathbf{F}_1$ and $\Lambda^{-1} \mathbf{F}_2$. This means that we can introduce $\xi_1 \in \mathcal{R}(\Lambda \mathbf{F}_1)$ and $\xi_2 \in \mathcal{R}(\Lambda^{-1} \mathbf{F}_2)$ such that $\xi_1 = \mathbf{Q}_1 \mathbf{x}_1$ and $\xi_2 = \mathbf{Q}_2 \mathbf{y}_1$. Then (3.44) becomes

$$\cos \theta = \max_{\xi_1 \in \mathcal{R}(\Lambda \mathbf{F}_1)} \max_{\xi_2 \in \mathcal{R}(\Lambda^{-1} \mathbf{F}_2)} \frac{|(\xi_1, \xi_2)|}{\|\xi_1\| \|\xi_2\|}, \quad (3.45)$$

which is the definition of the angle between the subspaces. Thus the Kantorovich-Wielandt angle θ is equal to the angle α between the two subspaces. \blacksquare

Lemma 3.1 is used to prove the following theorem.

Theorem 3.2 *The condition number of \mathbf{Q} is equal to 1.*

Proof. The angle θ can be calculated from

$$\begin{aligned} \cos \theta &= \max_{\mathbf{x} \in \mathcal{R}(\mathbf{\Lambda F}_1)} \max_{\mathbf{y} \in \mathcal{R}(\mathbf{\Lambda}^{-1} \mathbf{F}_2)} \frac{|(\mathbf{x}, \mathbf{y})|}{\|\mathbf{x}\| \|\mathbf{y}\|} \\ &= \max_{\mathbf{x} \in \mathcal{R}(\mathbf{F}_1)} \max_{\mathbf{y} \in \mathcal{R}(\mathbf{F}_2)} \frac{|(\mathbf{\Lambda}^{-1} \mathbf{x}, \mathbf{\Lambda y})|}{\|\mathbf{\Lambda}^{-1} \mathbf{x}\| \|\mathbf{\Lambda y}\|} \\ &= \max_{\mathbf{x} \in \mathcal{R}(\mathbf{F}_1)} \max_{\mathbf{y} \in \mathcal{R}(\mathbf{F}_2)} \frac{|(\mathbf{\Lambda}^* \mathbf{\Lambda}^{-1} \mathbf{x}, \mathbf{y})|}{\|\mathbf{\Lambda}^{-1} \mathbf{x}\| \|\mathbf{\Lambda y}\|}. \end{aligned} \quad (3.46)$$

For $0 < R \leq 1$, $\mathbf{\Lambda}$ is a diagonal matrix with real elements, hence $\mathbf{\Lambda}^* = \mathbf{\Lambda}$, and we find

$$\cos \theta = \max_{\mathbf{x} \in \mathcal{R}(\mathbf{F}_1)} \max_{\mathbf{y} \in \mathcal{R}(\mathbf{F}_2)} \frac{|(\mathbf{x}, \mathbf{y})|}{\|\mathbf{\Lambda}^{-1} \mathbf{x}\| \|\mathbf{\Lambda y}\|}. \quad (3.47)$$

However, since $\mathcal{R}(\mathbf{F}_1) \perp \mathcal{R}(\mathbf{F}_2)$, the inner product between the vectors $\mathbf{x} \in \mathcal{R}(\mathbf{F}_1)$ and $\mathbf{y} \in \mathcal{R}(\mathbf{F}_2)$ is equal to zero. Hence $\cos \theta = 0$ and consequently $\text{cond}(\mathbf{Q}) = 1$.

For $R > 1$, the first diagonal element of $\mathbf{\Lambda}_G$ is negative, and hence the first diagonal element of $\mathbf{\Lambda} = \mathbf{\Lambda}_G^{1/2} \mathbf{\Lambda}_H^{-1/2}$ is imaginary. In that case

$$(\mathbf{\Lambda}^{-1} \mathbf{x}, \mathbf{\Lambda y}) = \left(-\frac{i}{a_1} \right) (ia_1)x_1y_1 + x_2y_2 + \dots + x_Ny_N = 0, \quad (3.48)$$

where $a_j, j = 1, \dots, N$, are the diagonal elements of $\mathbf{\Lambda}$. So also in the case $R > 1$ $\cos \theta = 0$ and $\text{cond}(\mathbf{Q}) = 1$. ■

Since the condition number of \mathbf{Q} is equal to one, it is interesting to note that \mathbf{Q} is a unitary matrix.

Corollary 3.3 *The matrix \mathbf{Q} is unitary.*

Proof. Recall that the matrix \mathbf{Q} consists of a unitary $N \times m$ block \mathbf{Q}_1 and a unitary $N \times (N - m)$ block \mathbf{Q}_2 . Accordingly we can split any vector $\mathbf{x} \in \mathbb{R}^N$ into two parts, $\mathbf{x} = [\mathbf{x}_1^T; \mathbf{x}_2^T]^T$. The matrix-vector product $\mathbf{Q}^* \mathbf{Qx}$ then reads

$$\mathbf{Q}^* \mathbf{Qx} = \begin{bmatrix} \mathbf{x}_1 - \mathbf{Q}_1^* \mathbf{Q}_2 \mathbf{x}_2 \\ -\mathbf{Q}_2^* \mathbf{Q}_1 \mathbf{x}_1 + \mathbf{x}_2 \end{bmatrix}. \quad (3.49)$$

The subspaces that are spanned by the columns of $\mathbf{\Lambda F}_1$ and $\mathbf{\Lambda}^{-1} \mathbf{F}_2$ are perpendicular. The matrices \mathbf{Q}_1 and \mathbf{Q}_2 are bases of these subspaces, and consequently $\mathbf{Q}_1^* \mathbf{Q}_2 = \mathbf{Q}_2^* \mathbf{Q}_1 = 0$. Thus we find $\mathbf{Q}^* \mathbf{Qx} = \mathbf{x}$. Likewise we can prove that $\mathbf{Q} \mathbf{Q}^* \mathbf{x} = \mathbf{x}$. Hence $\mathbf{Q}^* \mathbf{Q} = \mathbf{Q} \mathbf{Q}^* = \mathbf{I}$ and \mathbf{Q} is a unitary matrix. ■

Estimating $\text{cond}(\mathbf{U})$

To estimate the condition number of \mathbf{U} we need estimates of the singular values of \mathbf{U}_1 and \mathbf{U}_2 . For this observe that

$$\begin{aligned}\sigma_k(\mathbf{U}_1) &= \sigma_k(\mathbf{Q}_1\mathbf{U}_1) = \sigma_k(\mathbf{\Lambda}\mathbf{F}_1) \leq \sigma_k(\mathbf{\Lambda})\sigma_1(\mathbf{F}_1) = \sigma_k(\mathbf{\Lambda}), \\ \sigma_k(\mathbf{U}_2) &= \sigma_k(\mathbf{Q}_2\mathbf{U}_2) = \sigma_k(\mathbf{\Lambda}^{-1}\mathbf{F}_2) \leq \sigma_k(\mathbf{\Lambda}^{-1})\sigma_1(\mathbf{F}_2) = \sigma_k(\mathbf{\Lambda}^{-1}),\end{aligned}\tag{3.50}$$

for $k = 1, \dots, m$ and $k = 1, \dots, N - m$ respectively. Here we used the facts that \mathbf{Q}_i and \mathbf{F}_i have orthogonal columns and have singular values 1. We also made use of estimates of the singular values of products of matrices [47]. Furthermore, with $\mathbf{F}_1 = \mathbf{\Lambda}^{-1}\mathbf{Q}_1\mathbf{U}_1$ and $\mathbf{F}_2 = \mathbf{\Lambda}\mathbf{Q}_2\mathbf{U}_2$ we obtain

$$\begin{aligned}1 &= \sigma_k(\mathbf{F}_1) = \sigma_k(\mathbf{\Lambda}^{-1}\mathbf{Q}_1\mathbf{U}_1) \leq \sigma_1(\mathbf{\Lambda}^{-1})\sigma_k(\mathbf{Q}_1\mathbf{U}_1) = \sigma_1(\mathbf{\Lambda}^{-1})\sigma_k(\mathbf{U}_1), \\ 1 &= \sigma_k(\mathbf{F}_2) = \sigma_k(\mathbf{\Lambda}\mathbf{Q}_2\mathbf{U}_2) \leq \sigma_1(\mathbf{\Lambda})\sigma_k(\mathbf{Q}_2\mathbf{U}_2) = \sigma_1(\mathbf{\Lambda})\sigma_k(\mathbf{U}_2),\end{aligned}\tag{3.51}$$

for $k = 1, \dots, m$ and $k = 1, \dots, N - m$ respectively. This yields the following lower bounds,

$$\begin{aligned}\sigma_k(\mathbf{U}_1) &\geq \frac{1}{\sigma_1(\mathbf{\Lambda}^{-1})} = \sigma_N(\mathbf{\Lambda}), \quad k = 1, \dots, m, \\ \sigma_k(\mathbf{U}_2) &\geq \frac{1}{\sigma_1(\mathbf{\Lambda})} = \sigma_N(\mathbf{\Lambda}^{-1}), \quad k = 1, \dots, N - m.\end{aligned}\tag{3.52}$$

With (3.50) and (3.52) we have upper and lower bounds for the singular values of \mathbf{U}_1 and \mathbf{U}_2 . The singular values of \mathbf{U} are the singular values of \mathbf{U}_1 plus the singular values of \mathbf{U}_2 , i.e. $\sigma(\mathbf{U}) = \sigma(\mathbf{U}_1) \cup \sigma(\mathbf{U}_2)$. For the condition number of \mathbf{U} we obtain

$$\begin{aligned}\text{cond}(\mathbf{U}) &= \frac{\sigma_1(\mathbf{U})}{\sigma_N(\mathbf{U})} \leq \frac{\max\{\sigma_1(\mathbf{\Lambda}), \sigma_1(\mathbf{\Lambda}^{-1})\}}{\min\{\sigma_N(\mathbf{\Lambda}), \sigma_N(\mathbf{\Lambda}^{-1})\}} \\ &= \frac{\max\left\{\sigma_1(\mathbf{\Lambda}), \frac{1}{\sigma_N(\mathbf{\Lambda})}\right\}}{\min\left\{\sigma_N(\mathbf{\Lambda}), \frac{1}{\sigma_1(\mathbf{\Lambda})}\right\}} \\ &= \max\left\{\sigma_1(\mathbf{\Lambda})^2, \frac{1}{\sigma_N(\mathbf{\Lambda})^2}\right\}.\end{aligned}\tag{3.53}$$

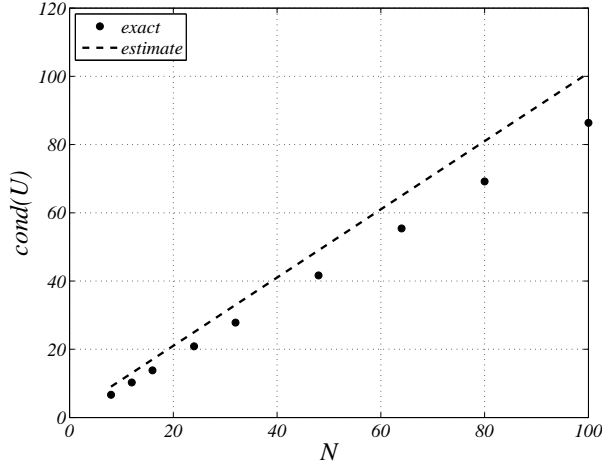


Figure 3.5: Condition number of \mathbf{U} as a function of N , where $R = 1/2$ and $m = N/2$. The dashed line is the estimate whereas the large dots give the exact value for several values of N .

Since $\mathbf{\Lambda}$ is the product of the square roots of $\mathbf{\Lambda}_G$ and $\mathbf{\Lambda}_H^{-1}$, we can derive its singular values, resulting in

$$\begin{aligned} \sigma_1(\mathbf{\Lambda})^2 &= \max \left\{ \frac{2\pi R |\log R|}{\pi - (N-1) \tan \pi/N}, \frac{\pi R}{\pi + \tan \pi/N} \right\}, \\ \sigma_N(\mathbf{\Lambda})^2 &= \min \left\{ \frac{2\pi R |\log R|}{\pi - (N-1) \tan \pi/N}, \frac{2\pi R/N}{\pi + \tan \pi/N} \right\}. \end{aligned} \quad (3.54)$$

We plot the condition number of \mathbf{U} and its approximation in Figure 3.5. As is seen from (3.53), the approximation provides an upper bound for the condition number of \mathbf{U} . The difference between the exact value and the estimate corresponds to the error that is made by approximating the smallest eigenvalue of \mathbf{G} by the smallest eigenvalue of \mathcal{K}^s .

Estimating $\text{cond}(\mathbf{A})$

The condition number of \mathbf{A} is estimated by the product of condition numbers of \mathbf{D} and \mathbf{U} , where the condition number of \mathbf{U} is obtained from the singular values of $\mathbf{\Lambda}$. Let us use a first order approximation for $\tan \pi/N$ to approximate the largest and smallest singular values of the matrices \mathbf{D} and $\mathbf{\Lambda}$ as given in (3.40) and (3.54). We

find

$$\begin{aligned}
\sigma_1(\mathbf{D})^2 &\approx \frac{R}{4N} \max(2|\log R|, N+1), \\
\sigma_N(\mathbf{D})^2 &\approx \frac{R}{2N} \min\left(|\log R|, 1 + \frac{1}{N}\right), \\
\sigma_1(\mathbf{A})^2 &\approx RN \max\left(2|\log R|, \frac{1}{N+1}\right), \\
\sigma_N(\mathbf{A})^2 &\approx 2RN \min\left(|\log R|, \frac{1}{N(N+1)}\right).
\end{aligned} \tag{3.55}$$

For a circle with radius $R = 1/2$ we obtain

$$\begin{aligned}
\text{cond}(\mathbf{D}) &\approx \sqrt{\frac{N+1}{2\log 2}}, \\
\text{cond}(\mathbf{U}) &\lesssim N+1,
\end{aligned} \tag{3.56}$$

which gives the following estimate for the condition number of \mathbf{A} ,

$$\text{cond}(\mathbf{A}) \lesssim \frac{1}{\sqrt{2\log 2}} (N+1)^{3/2}. \tag{3.57}$$

Figure 3.6 shows this estimate for the condition number. The dashed line gives the estimate as a function of N , whereas the dots give the exact value of the condition number for several values of N . For this example we choose $m = N/2$, i.e. as many elements with Dirichlet conditions as elements with Neumann conditions. However, the parameter m does not appear in the estimates, and will not play any role. We observe that the estimate is of higher order than the actual value of the condition number. In fact the condition number of \mathbf{A} turns out to be linear in N , while the estimate is of order $N^{3/2}$. Later on we will show how this discrepancy is caused.

Figure 3.7 gives the condition number of \mathbf{A} and its estimate as a function of R . We choose $N = 12$ and $m = 6$. Again the estimate is $\mathcal{O}(N^{3/2})$, while the actual value is $\mathcal{O}(N)$. The estimate does capture the large condition number at $R = 1$ though.

Let us recapitulate the steps that we have taken to bound the condition number of the matrix \mathbf{A} . We decomposed \mathbf{A} in a product of matrices and derived that $\text{cond}(\mathbf{A}) \leq \text{cond}(\mathbf{D})\text{cond}(\mathbf{U})$. We can evaluate or approximate the condition numbers of \mathbf{D} and \mathbf{U} very well, but the condition number of \mathbf{A} is over-estimated. This must be caused by the decomposition of \mathbf{A} . As a simple example, consider the matrices \mathbf{X} and \mathbf{Y} , given by

$$\mathbf{X} := \begin{bmatrix} 1 & 0 \\ 0 & \varepsilon \end{bmatrix}, \quad \mathbf{Y} := \begin{bmatrix} 1 & 0 \\ 0 & 1/\varepsilon \end{bmatrix}, \tag{3.58}$$

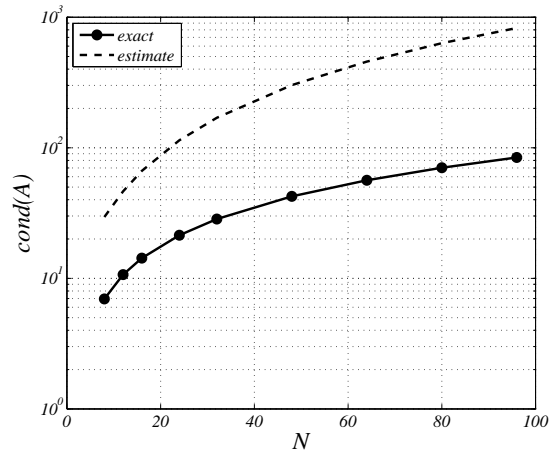


Figure 3.6: Condition number of \mathbf{A} as a function of N , where $R = 1/2$ and $m = N/2$. The dashed line is the estimate whereas the large dots give the exact condition number for several values of N .

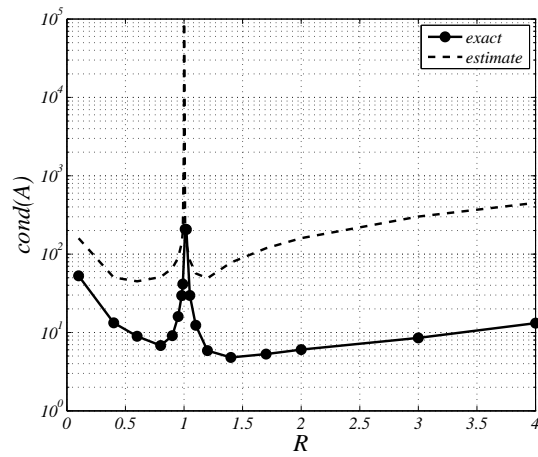


Figure 3.7: Condition number of \mathbf{A} as a function of R , where $N = 12$ and $m = 6$. The dashed line is the estimate whereas the large dots give exact values for several values of R .

where $0 < \varepsilon \ll 1$. Clearly both matrices have a large condition number $\sim \varepsilon^{-1}$, while the product matrix $\mathbf{Z} := \mathbf{X}\mathbf{Y}$ has condition number 1. Exploiting the fact that \mathbf{Z} is a product of \mathbf{X} and \mathbf{Y} yields,

$$\text{cond}(\mathbf{Z}) \leq \text{cond}(\mathbf{X})\text{cond}(\mathbf{Y}) = \frac{1}{\varepsilon^2}, \quad (3.59)$$

which highly over-estimates the condition number of \mathbf{Z} . Thus decomposing a matrix and multiplying the condition numbers of the several factor matrices does not necessarily yield a good approximation of the condition number of the original matrix.

3.6 Decoupled equations

The decomposition of the BEM-matrix \mathbf{A} that we derived in the previous section has been useful for finding an estimate for the condition number of \mathbf{A} . Besides this estimate the decomposition can also be used to decouple the equations from the BEM formulation. This allows us to separately retrieve the missing information of u at Γ_2 and q at Γ_1 . Recall equation (2.37), relating the boundary data \mathbf{b} to the unknown vector \mathbf{x} ,

$$\mathbf{A}\mathbf{x} = \tilde{\mathbf{G}}\mathbf{b}. \quad (3.60)$$

In the previous section we have shown that \mathbf{A} is decomposed as

$$\mathbf{A} = \mathbf{F}^*\mathbf{D}\mathbf{Q}\mathbf{U}. \quad (3.61)$$

The matrix $\tilde{\mathbf{G}}$ can be decomposed in a similar manner. First we write $\tilde{\mathbf{G}}$ as

$$\begin{aligned} \tilde{\mathbf{G}} &= [-\tilde{\mathbf{H}}\mathbf{P}_1 \mid \mathbf{G}\mathbf{P}_2] \\ &= [-\mathbf{F}^*\boldsymbol{\Lambda}_H\mathbf{F}\mathbf{P}_1 \mid \mathbf{F}^*\boldsymbol{\Lambda}_G\mathbf{F}\mathbf{P}_2] \\ &= \mathbf{F}^*[-\boldsymbol{\Lambda}_H\mathbf{F}_1 \mid \boldsymbol{\Lambda}_G\mathbf{F}_2] \\ &= \mathbf{F}^*\mathbf{D}[-\boldsymbol{\Lambda}^{-1}\mathbf{F}_1 \mid \boldsymbol{\Lambda}\mathbf{F}_2]. \end{aligned} \quad (3.62)$$

Realizing that the vector \mathbf{b} is constructed from the boundary data $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{q}}_2$, namely $\mathbf{b} = [\tilde{\mathbf{u}}_1; \tilde{\mathbf{q}}_2]$, we can write the right-hand side of (3.60) as

$$\tilde{\mathbf{G}}\mathbf{b} = \mathbf{F}^*\mathbf{D}(-\boldsymbol{\Lambda}^{-1}\mathbf{F}_1\tilde{\mathbf{u}}_1 + \boldsymbol{\Lambda}\mathbf{F}_2\tilde{\mathbf{q}}_2) =: \mathbf{F}^*\mathbf{D}\mathbf{f}. \quad (3.63)$$

Using the new expressions for \mathbf{A} and $\tilde{\mathbf{G}}\mathbf{b}$, equation (3.60) becomes

$$\mathbf{F}^*\mathbf{D}\mathbf{Q}\mathbf{U}\mathbf{x} = \mathbf{F}^*\mathbf{D}\mathbf{f}. \quad (3.64)$$

Multiplying by $\mathbf{Q}^*\mathbf{D}^{-1}\mathbf{F}$ yields

$$\mathbf{U}\mathbf{x} = \mathbf{Q}^*\mathbf{f}, \quad (3.65)$$

or

$$\begin{bmatrix} \mathbf{U}_1 & \emptyset \\ \emptyset & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{u}_2 \end{bmatrix} = [-\mathbf{Q}_1 \mid \mathbf{Q}_2]^*\mathbf{f} = \begin{bmatrix} -\mathbf{Q}_1^*\mathbf{f} \\ \mathbf{Q}_2^*\mathbf{f} \end{bmatrix}. \quad (3.66)$$

So we obtain a set of two linear systems,

$$\mathbf{U}_1\mathbf{q}_1 = -\mathbf{Q}_1^*\mathbf{f}, \quad (3.67a)$$

$$\mathbf{U}_2\mathbf{u}_2 = \mathbf{Q}_2^*\mathbf{f}. \quad (3.67b)$$

We are given data $\tilde{\mathbf{u}}_1$ on Γ_1 and $\tilde{\mathbf{q}}_2$ on Γ_2 . Assume that we are interested in \mathbf{u}_2 , i.e. the unknown coefficients of u at Γ_2 . In the original BEM formulation we would then calculate both u at Γ_2 and q at Γ_1 . With equation (3.67b) we can directly calculate u at Γ_2 without computing q at Γ_1 .

Let us summarize which steps we have to take in order to compute u at Γ_2 . First we need the Fourier matrix \mathbf{F} , whose two components \mathbf{F}_1 and \mathbf{F}_2 appear in the vector \mathbf{f} . We also need the diagonal matrix $\mathbf{\Lambda}$, which contains the eigenvalues of the matrices \mathbf{G} and $\tilde{\mathbf{H}}$ ($\mathbf{\Lambda} = \mathbf{\Lambda}_G^{1/2}\mathbf{\Lambda}_H^{-1/2}$). These eigenvalues are approximated by the eigenvalues of the single and double-layer potentials. Then we need a QR-decomposition of the matrix product $\mathbf{\Lambda}^{-1}\mathbf{F}_2$, which yields us the matrices \mathbf{Q}_2 and \mathbf{U}_2 . Finally we solve the system in (3.67b). Note that the matrix \mathbf{U}_2 in this system is an upper triangular matrix, so the system is solved very efficiently by backward elimination. The most costly step in this procedure is the QR-decomposition of $\mathbf{\Lambda}^{-1}\mathbf{F}_2$. However, note that $\mathbf{\Lambda}^{-1}\mathbf{F}_2$ is a $N \times (N - m)$ matrix. With the original BEM procedure we would have to find a QR-decomposition for a $N \times N$ matrix.

The method described above works similarly if we want to calculate q at Γ_1 without computing u at Γ_2 . Then we need a QR-decomposition for $\mathbf{\Lambda}\mathbf{F}_1$ and solve equation (3.67a).

Example 3.1

Consider the following problem on the circle Ω with radius R ,

$$\begin{aligned} \nabla^2 u &= 0, & \mathbf{x} &\in \Omega, \\ u &= R \cos \theta, & \mathbf{x} &\in \Gamma_1, \\ q &= \cos \theta, & \mathbf{x} &\in \Gamma_2, \end{aligned} \quad (3.68)$$

which has exact solution $u = R \cos \theta$ and $q = \cos \theta$ at Γ . We choose $R = 2$, $m = N - 1$, so all elements have Dirichlet boundary conditions, except for the last element. We calculate the value of u in this last element, following the procedure

N	BEM		BEM decoupled	
	time(s)	rel. error	time(s)	rel. error
100	0.44	$3.93 \cdot 10^{-4}$	0.10	$3.96 \cdot 10^{-4}$
200	1.85	$9.95 \cdot 10^{-4}$	0.39	$9.95 \cdot 10^{-5}$
300	4.56	$4.44 \cdot 10^{-5}$	1.27	$4.43 \cdot 10^{-5}$
400	9.14	$2.51 \cdot 10^{-5}$	2.97	$2.49 \cdot 10^{-5}$
500	15.80	$1.61 \cdot 10^{-5}$	5.64	$1.60 \cdot 10^{-5}$

Table 3.5: Calculation time and relative error for both BEM and decoupled BEM. Problem that is solved is almost purely Dirichlet ($m = N - 1$).

N	BEM	BEM decoupled
	time (s)	time (s)
100	0.43	0.12
200	1.82	0.56
300	4.63	1.86
400	9.13	4.08
500	15.80	7.79

Table 3.6: Calculation time for both BEM and decoupled BEM. Problem that is solved has mixed boundary conditions ($m = N/2$).

described above. In Table 3.5 we see that the computation time for finding u with the decoupled BEM is a factor 3 to 4 lower than finding u with the original BEM. At the same time the accuracy is equally good.

Now let us choose $m = N/2$ and calculate u at Γ_2 and q at Γ_1 separately using the decoupled BEM and compare this to the original BEM. The computation time is given in Table 3.6. Again we see that the decoupled BEM is faster than the original BEM. The reason for this is simple: it is more efficient to solve two small systems than one large system. \square

We can improve the efficiency even more, as we do not need to solve the systems in (3.67) by backward elimination. For this we multiply (3.67a) by \mathbf{Q}_1 ,

$$\mathbf{Q}_1 \mathbf{U}_1 \mathbf{q}_1 = -\mathbf{Q}_1 \mathbf{Q}_1^* \mathbf{f}, \quad (3.69)$$

and replace the product $\mathbf{Q}_1 \mathbf{U}_1$ by $\Lambda \mathbf{F}_1$,

$$\Lambda \mathbf{F}_1 \mathbf{q}_1 = -\mathbf{Q}_1 \mathbf{Q}_1^* \mathbf{f}. \quad (3.70)$$

We next multiply by $\mathbf{F}_1^* \Lambda^{-1}$ to find

$$\mathbf{q}_1 = -\mathbf{F}_1^* \Lambda^{-1} \mathbf{Q}_1 \mathbf{Q}_1^* \mathbf{f}. \quad (3.71)$$

The linear system in (3.67b) can be transformed in a similar manner, and the two systems in (3.67) are solved with

$$\mathbf{q}_1 = -\mathbf{F}_1^* \Lambda^{-1} \mathbf{Q}_1 \mathbf{Q}_1^* \mathbf{f}, \quad (3.72a)$$

$$\mathbf{u}_2 = \mathbf{F}_2^* \Lambda \mathbf{Q}_2 \mathbf{Q}_2^* \mathbf{f}. \quad (3.72b)$$

In this way \mathbf{q}_1 and \mathbf{u}_2 are found without using backward substitution. Nevertheless the expressions on the right-hand sides involve many matrix-vector multiplications and may therefore be not very efficient. Moreover we still need to find \mathbf{Q}_1 and \mathbf{Q}_2 , i.e. perform QR-decompositions.

For a pure Dirichlet problem, i.e. $m = N$, we have $\mathbf{Q}_1 = \mathbf{Q}$, which is a unitary matrix. Therefore the matrix product $\mathbf{Q}_1 \mathbf{Q}_1^*$ in (3.72a) is equal to the $N \times N$ matrix \mathbf{I}_N . Also $\mathbf{P}_1^T = \mathbf{I}_N$ and the vector \mathbf{f} is equal to $\mathbf{f} = -\Lambda^{-1} \mathbf{F} \tilde{\mathbf{u}}$. Hence, for the Dirichlet problem (3.72a) becomes

$$\mathbf{q} = -\mathbf{F}^* \Lambda^{-1} (-\Lambda^{-1} \mathbf{F} \tilde{\mathbf{u}}) = \mathbf{F}^* \Lambda^{-2} \mathbf{F} \tilde{\mathbf{u}} = \mathbf{F}^* \Lambda_G^{-1} \Lambda_H \mathbf{F} \tilde{\mathbf{u}}. \quad (3.73)$$

All matrices in this expression are known or can be approximated accurately. Thus we obtain \mathbf{q} by performing a simple matrix-vector product. In a similar manner we find \mathbf{u} from boundary data $\tilde{\mathbf{q}}$ with

$$\mathbf{u} = \mathbf{F}^* \Lambda_G \Lambda_H^{-1} \mathbf{F} \tilde{\mathbf{q}}. \quad (3.74)$$

Chapter 4

Logarithmic capacity

In this chapter we investigate the uniqueness of the solution of the boundary integral equation (BIE) for the Laplace equation on a general two-dimensional domain. Three different types of boundary conditions are distinguished: Dirichlet, mixed and Neumann boundary conditions. For a Laplace equation with Dirichlet boundary conditions it has been shown that a unique solution does not always exist, depending on the size of the domain. A similar result is proven for the case of mixed boundary conditions. The BIE for the Laplace equation with Neumann boundary conditions does never have a unique solution.

4.1 Introduction

It is well-known that the Laplace equation in differential form with either Dirichlet or mixed boundary conditions has a unique solution. However, when the Laplace equation is transformed to a BIE this is not so straightforward anymore. It is noted that the BIE for the Dirichlet Laplace equation does not always have a unique solution [53, 56, 75, 85]. Certain domains can be distinguished on which the BIE becomes singular and a non-trivial solution of the homogeneous equations can be found. A multiple of this solution can be added to the solution of the non-homogeneous equations, which is then no longer unique. For each domain there exists exactly one rescaled version of this domain for which the BIE becomes singular. This introduces an extraordinary phenomenon for the BIEs; uniqueness of the solutions depends on the scale of the domain.

The domains on which no unique solution can be guaranteed are related to the so-called *logarithmic capacity*. The logarithmic capacity is a real positive number being a function of the domain. This concept originates from the field of measure theory, but it also appears in potential theory. The concept of a capacity applied to

a single domain may be a bit confusing, as usually the *electrical* capacity is defined as a charge difference between two conducting objects. The logarithmic capacity however is related to a single domain.

In potential theory it is shown that when the logarithmic capacity of a domain is equal to one, then the homogeneous BIE for the Dirichlet Laplace equation at the boundary of that domain has a non-trivial solution [52, 66, 91]. This allows us to *a-priori* detect whether a BIE will become singular on a certain domain. Namely, we have to compute the logarithmic capacity and verify whether it is equal to one. Additionally, the logarithmic capacity also enables us to modify the BIE such that it does not become singular. We can scale the domain in such a way that the logarithmic capacity does not become equal to one. The BIE on the corresponding boundary will then be nonsingular.

The BIE for the Laplace equation with mixed boundary conditions did not receive much attention until now [39]. However, a similar phenomenon as for the Dirichlet case takes place for mixed conditions. For each domain there exists exactly one rescaled version of this domain for which the BIE becomes singular. This result is proven in Section 4.4.

Research has been done on the BIE for the biharmonic equation. It is shown that the BIE for the biharmonic equation with Dirichlet conditions on a circle does not admit a unique solution when the radius of the circle is equal to 1 or e^{-1} [23]. A few years later it was shown that a more general result is true: for any 2D domain there exists two critical scalings for which the BIE does not have a unique solution [44]. For the BIE for the biharmonic equation, the number of critical scaling can even increase to three or four for domains that consist of two or four separate squares [30]. Apart from rescaling the domain there are two options to guarantee nonsingular BIEs and a unique solution. The first option is to add two supplementary conditions [23], while the second option involves modified fundamental solutions for the biharmonic equation [28].

Another class of BIEs for which uniqueness properties has been investigated is the class of BIEs for the Helmholtz equation. It is derived that there is a countable set of critical wave numbers for which the condition number of both the integral operator as the related discrete operator becomes infinitely large [2, 59, 60, 61]. By introducing a coupling parameter between the various boundary integral operators it is shown that the condition numbers can be minimized. The singular BIE for the Helmholtz equation is different from the singular BIEs for the Laplace and biharmonic equation. The singularity appears at certain wave numbers, and not at certain critical scalings. Moreover singular BIEs also appear for the Helmholtz equation in 3D. This is not the case for the BIEs for the Laplace and biharmonic equation.

In this chapter we investigate both the BIE for the Laplace equation with Dirichlet

conditions and mixed conditions, and we also include the BIE for the Laplace equation with Neumann conditions. Thus we study three different problems. First we describe the setting of these three problems.

Let Ω be a simply connected domain in 2D whose boundary Γ is a closed curve. In the interior of Ω the Laplace equation holds for the unknown function $u = u(\mathbf{x})$,

$$\nabla^2 u = 0, \quad \mathbf{x} \in \Omega. \quad (4.1)$$

Recall that the *fundamental solution* G of the Laplace operator ∇^2 is given by

$$G(\mathbf{x}, \mathbf{y}) := \frac{1}{2\pi} \log \frac{1}{\|\mathbf{x} - \mathbf{y}\|}. \quad (4.2)$$

We denote by q the derivative of u with respect to the outward normal \mathbf{n} at Γ . Introduce the *single and double layer potential* by

$$\begin{aligned} (\mathcal{K}^s q)(\mathbf{x}) &:= \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) d\Gamma_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma, \\ (\mathcal{K}^d u)(\mathbf{x}) &:= \int_{\Gamma} \frac{\partial}{\partial n_{\mathbf{y}}} \{G(\mathbf{x}, \mathbf{y})\} u(\mathbf{y}) d\Gamma_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma, \end{aligned} \quad (4.3)$$

respectively. The BIE for the Laplace equation deduced in Chapter 2 reads (cf. [6])

$$\frac{1}{2}u + \mathcal{K}^d u = \mathcal{K}^s q, \quad \mathbf{x} \in \Gamma. \quad (4.4)$$

At each point on the boundary we prescribe either u or q . We distinguish three types of boundary conditions.

Dirichlet problem

$$u = \tilde{u}, \quad \mathbf{x} \in \Gamma. \quad (4.5)$$

Mixed problem

$$\begin{aligned} u &= \tilde{u}, \quad \mathbf{x} \in \Gamma_1, \\ q &= \tilde{q}, \quad \mathbf{x} \in \Gamma_2, \end{aligned} \quad (4.6)$$

Neumann problem

$$q = \tilde{q}, \quad \mathbf{x} \in \Gamma. \quad (4.7)$$

where $\Gamma_1 \cup \Gamma_2 = \Gamma$ and $\Gamma_1 \cap \Gamma_2 = \emptyset$.

4.2 Logarithmic capacity

To study the uniqueness properties of the Dirichlet and the mixed problem in the next section we need to introduce the notion of *logarithmic capacity*. We define the energy integral I by

$$I(q) := \int_{\Gamma} \int_{\Gamma} \log \frac{1}{\|\mathbf{x} - \mathbf{y}\|} q(\mathbf{x})q(\mathbf{y})d\Gamma_x d\Gamma_y \quad (4.8)$$

and the logarithmic capacity $C_l(\Gamma)$ is related to this integral by

$$-\log C_l(\Gamma) := \inf_q I(q). \quad (4.9)$$

Here the infimum is taken over all functions q with the restriction that

$$\int_{\Gamma} q(\mathbf{x})d\Gamma_x = 1. \quad (4.10)$$

Let us give a physical interpretation of the logarithmic capacity. For simplicity let the domain Ω be contained in the disc with radius $1/2$. In that case it can be shown that the integral $I(q)$ is positive. The function q can be seen as a charge distribution over a conducting domain Ω . Faraday demonstrated that this charge will only reside at the exterior boundary of the domain, in our case at Γ . We normalize q in such a way that the total amount of charge at Γ is equal to one, cf. condition (4.10). The function $\mathcal{K}^s q$ is identified as the potential due to the charge distribution q . Note that the integral I can also be written as

$$I(q) = 2\pi \int_{\Gamma} (\mathcal{K}^s q)(\mathbf{x})q(\mathbf{x})d\Gamma_x. \quad (4.11)$$

Hence I can be seen as the energy of the charge distribution q . The charge will distribute itself over Γ in such a way that the energy I is minimized. So the quantity $-\log C_l(\Gamma)$ is the minimal amount of energy. Hence the logarithmic capacity $C_l(\Gamma)$ is a measure for the capability of the boundary Γ to support a unit amount of charge.

For most boundaries the logarithmic capacity is not known explicitly. Only for a few elementary domains the logarithmic capacity can be calculated analytically [66]; we have listed some in Table 4.1.

There are also some useful properties [5, 52] that help us to determine or estimate the logarithmic capacity.

1. If Γ is the outer boundary of a closed bounded domain Ω , then $C_l(\Gamma) = C_l(\Omega)$. This agrees with the idea of Faraday's cage, mentioned above.
2. Denote by d_{Γ} the Euclidean diameter of Ω , then $C_l(\Gamma) \leq d_{\Gamma}$. Hence the radius of the smallest circle in which Γ is contained is an upper bound for the logarithmic capacity of Γ .

boundary Γ	logarithmic capacity $C_l(\Gamma)$
circle with radius R	R
square with side L	$\frac{\Gamma(\frac{1}{4})^2}{4\pi^{3/2}} L \approx 0.59017 \cdot L$
ellipse with semi-axes a and b	$(a + b)/2$
interval of length a	$\frac{1}{4}a$
isosceles right triangle side l	$\frac{3^{3/4}\Gamma(1/4)^2}{2^{7/2}\pi^{3/2}} l \approx 0.476 l$

Table 4.1: The logarithmic capacity of some domains. Note that $\Gamma(\cdot)$ represents the gamma-function.

3. If $\Gamma = \mathbf{x} + \alpha\Gamma_1$, then $C_l(\Gamma) = \alpha C_l(\Gamma_1)$. Hence the logarithmic capacity behaves linearly with respect to scaling and is invariant with respect to translation.
4. If $\Omega_1 \subset \Omega_2$, then $C_l(\Omega_1) \leq C_l(\Omega_2)$.
5. For a convex domain Ω ,

$$C_l(\Omega) \geq \left(\frac{\text{area}(\Omega)}{\pi} \right)^{1/2}. \quad (4.12)$$

If the properties from the list above do not supply accurate enough estimates, the logarithmic capacity can also be approximated numerically with the help of linear programming [82].

4.3 Dirichlet problem

For the BIE that arises from the Laplace equation with Dirichlet boundary conditions we have the following result.

Theorem 4.1 *There exists a nonzero q_e such that*

$$(\mathcal{K}^s q_e)(\mathbf{x}) = -\frac{1}{2\pi} \log C_l(\Gamma), \quad \mathbf{x} \in \Gamma. \quad (4.13)$$

Sketch of proof. In the following we briefly present the major steps in the proof of the theorem [52, 70, 98]. We observe that for the values of the energy integral (4.8) we have $-\infty < I(q) \leq \infty$. If the infimum of the energy integral is infinitely large, then by definition the logarithmic capacity is equal to zero.

Suppose that $C_l(\Gamma) > 0$ and thus $-\infty < I(q) < \infty$. It is proven [52, p. 282] that for each boundary Γ there exists a unique minimizer q_e of $I(q)$, i.e.

$$I(q_e) = \inf_q I(q) = -\log C_l(\Gamma) \text{ with } \int_{\Gamma} q_e(\mathbf{x}) d\Gamma_x = 1. \quad (4.14)$$

For the minimizer q_e the following result is proven [52, p. 287]. Let Γ be a closed bounded domain with positive logarithmic capacity and a connected complement. Then $2\pi\mathcal{K}^s q_e \leq -\log C_l(\Gamma)$ in the whole plane and $2\pi\mathcal{K}^s q_e = -\log C_l(\Gamma)$ at Γ , except possibly for a subset which has zero logarithmic capacity. ■

Theorem 4.1 leads to the following result.

Corollary 4.2 *If $C_l(\Gamma) = 1$ there exists a nonzero q_e such that $\mathcal{K}^s q_e = 0$.*

Thus in the specific case that $C_l(\Gamma) = 1$ the single layer operator \mathcal{K}^s admits an eigenfunction q_e with zero eigenvalue. Hence \mathcal{K}^s is not positive definite and the Dirichlet problem does not have a unique solution.

If we rescale the domain such that the Euclidean diameter is smaller than one, then the second property in Section 4.2 shows us that the logarithmic capacity will also be smaller than one. In this way we can guarantee the existence of a unique solution of the BIE.

Recall that the non-trivial solution q_e of the homogeneous BIE $\mathcal{K}^s q = 0$ has a contour integral equal to 1. At the same time we realize that a solution q of $\mathcal{K}^s q = 0$ has to satisfy

$$\int_{\Gamma} q d\Gamma = \int_{\Omega} \Delta u d\Omega = 0, \quad (4.15)$$

where we make use of Gauss' theorem. By adding this requirement for q to the BIE, we exclude the possibility that q_e is a solution of the homogeneous BIE. This provides a second strategy to ensure unique solutions of the BIE.

A third option to guarantee a unique solution is to adjust the integral operator \mathcal{K}^s . Note that the function G_α ,

$$G_\alpha(\mathbf{x}, \mathbf{y}) := \frac{1}{2\pi} \log \frac{\alpha}{\|\mathbf{x} - \mathbf{y}\|}, \quad \alpha \in \mathbb{R}^+, \quad (4.16)$$

is also a fundamental solution for the Laplace operator. The corresponding single layer potential reads

$$\mathcal{K}_\alpha^s q := \frac{1}{2\pi} \int_{\Gamma} \log \frac{\alpha}{\|\mathbf{x} - \mathbf{y}\|} q(\mathbf{y}) d\Gamma_y = \mathcal{K}^s q + \frac{\log \alpha}{2\pi} \int_{\Gamma} q d\Gamma. \quad (4.17)$$

For the minimizer q_e we get

$$\begin{aligned}\mathcal{K}_\alpha^s q_e &= \mathcal{K}^s q_e + \frac{\log \alpha}{2\pi} \int_\Gamma q_e d\Gamma = -\frac{1}{2\pi} \log C_l(\Gamma) + \frac{1}{2\pi} \log \alpha \\ &= \frac{1}{2\pi} \log \frac{\alpha}{C_l(\Gamma)}.\end{aligned}\quad (4.18)$$

This is only equal to zero if $\alpha = C_l(\Gamma)$. We may choose α any positive real number unequal to $C_l(\Gamma)$ and obtain $\mathcal{K}_\alpha^s q_e \neq 0$. In that case q_e is no longer an eigenfunction of the single layer potential operator with zero eigenvalue. Hence the BIE (4.4) with Dirichlet conditions is uniquely solvable if \mathcal{K}^s is replaced by \mathcal{K}_α^s [26, 70]. The advantage of this procedure is that we do not need a rescaling of the domain, nor do we have to add an extra equation. Furthermore, we do not need to know the logarithmic capacity explicitly; a rough estimate of the capacity suffices to choose α such that $\alpha \neq C_l(\Gamma)$.

There are also ways to ensure a unique solution of the BIE for the Laplace equation that can be used without having to know the logarithmic capacity. For instance, adding an extra collocation node at the interior or exterior of the domain can change the BIE in such a way that it is not singular any longer [17]. This depends on the location of the extra collocation node though. Another option is to use the hypersingular formulation of the BIE [18]. The hypersingular BIE is the normal derivative of the standard BIE and does not involve the single layer operator. As a consequence the BIE does not become singular at certain domains.

4.4 Mixed problem

To investigate the Laplace equation with mixed boundary conditions we have to rewrite the BIE in (4.4). For $i = 1, 2$ we introduce the functions $u_i := u|_{\Gamma_i}$ and $q_i := q|_{\Gamma_i}$ and the boundary integral operators

$$(\mathcal{K}_i^s q)(\mathbf{x}) := \int_{\Gamma_i} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) d\Gamma_y, \quad \mathbf{x} \in \Gamma, \quad (4.19a)$$

$$(\mathcal{K}_i^d u)(\mathbf{x}) := \int_{\Gamma_i} \frac{\partial}{\partial n_y} G(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) d\Gamma_y, \quad \mathbf{x} \in \Gamma. \quad (4.19b)$$

Note that the boundary conditions (4.6) provide $u_1 = \tilde{u}$ and $q_2 = \tilde{q}$. By distinguishing $\mathbf{x} \in \Gamma_1$ and $\mathbf{x} \in \Gamma_2$, we write (4.4) as a system of two BIEs,

$$\mathcal{K}_2^d u_2 - \mathcal{K}_1^s q_1 = \mathcal{K}_2^s \tilde{q} - \frac{1}{2} \tilde{u} - \mathcal{K}_1^d \tilde{u}, \quad \mathbf{x} \in \Gamma_1, \quad (4.20a)$$

$$\frac{1}{2} u_2 + \mathcal{K}_2^d u_2 - \mathcal{K}_1^s q_1 = \mathcal{K}_2^s \tilde{q} - \mathcal{K}_1^d \tilde{u}, \quad \mathbf{x} \in \Gamma_2. \quad (4.20b)$$

In this system all prescribed boundary data are at the right-hand side of the equations.

Theorem 4.3 *If $C_l(\Gamma) = 1$, the homogeneous equations of (4.20a) and (4.20b) have a non-trivial solution pair (q_1, u_2) .*

Proof. We have to find a non-trivial pair of functions (q_1, u_2) such that the left-hand sides of (4.20a) and (4.20b) are equal to zero when $C_l(\Gamma) = 1$. To this end we choose $u_2 \equiv 0$ and $q_1 = q_e|_{\Gamma_1} + h_1$, with the function h_1 satisfying

$$\mathcal{K}_1^s h_1 = \mathcal{K}_2^s q_e, \quad \mathbf{x} \in \Gamma. \quad (4.21)$$

With these choices both the left-hand sides of (4.20a) and (4.20b) are equal to

$$\begin{aligned} -\mathcal{K}_1^s q_1 &= -(\mathcal{K}_1^s q_e + \mathcal{K}_1^s h_1) = -(\mathcal{K}_1^s q_e + \mathcal{K}_2^s q_e) = -\mathcal{K}^s q_e \\ &= \frac{1}{2\pi} \log C_l(\Gamma) = 0. \end{aligned} \quad (4.22)$$

We still have to prove that it is possible to find a function h_1 that satisfies (4.21). First we note that the right-hand side of (4.21) is in $\langle q_e \rangle^\perp$, since

$$\begin{aligned} (\mathcal{K}_2^s q_e, q_e)_\Gamma &= \int_\Gamma \int_{\Gamma_2} G(\mathbf{x}, \mathbf{y}) q_e(\mathbf{y}) d\Gamma_y q_e(\mathbf{x}) d\Gamma_x \\ &= \int_{\Gamma_2} \int_\Gamma G(\mathbf{x}, \mathbf{y}) q_e(\mathbf{x}) d\Gamma_x q_e(\mathbf{y}) d\Gamma_y \\ &= (\mathcal{K}^s q_e, q_e)_{\Gamma_2} = -\frac{1}{2\pi} \log C_l(\Gamma) (1, q_e)_{\Gamma_2} = 0. \end{aligned} \quad (4.23)$$

Here $(\cdot, \cdot)_\Gamma$ stands for the inner product over the boundary Γ . As the right-hand side of (4.21) is in $\langle q_e \rangle^\perp$, we can generalize the question: is it possible to find a function h_1 such that $\mathcal{K}_1^s h_1 = \phi$ for all $\phi \in \langle q_e \rangle^\perp$? If so, then $\phi = \mathcal{K}_2^s q_e$ completes the proof.

For all functions $q \in \langle q_e \rangle^\perp$ with $q \neq 0$ we have $I(q) > I(q_e)$, since q_e is the unique minimizer of I . Using $I(q) = 2\pi(\mathcal{K}^s q, q)$ we find that

$$(\mathcal{K}^s q, q) > \frac{1}{2\pi} I(q_e) = -\frac{1}{2\pi} \log C_l(\Gamma) = 0, \quad q \in \langle q_e \rangle^\perp, \quad q \neq 0. \quad (4.24)$$

So \mathcal{K}^s is positive definite and invertible on the function space $\langle q_e \rangle^\perp$. This means that for all $\phi \in \langle q_e \rangle^\perp$ there is a function h with $\mathcal{K}^s h = \phi$. Let h be the solution of $\mathcal{K}^s h = \phi$, then it can be decomposed as

$$h = \begin{cases} h_1, & \mathbf{x} \in \Gamma_1, \\ h_2, & \mathbf{x} \in \Gamma_2. \end{cases} \quad (4.25)$$

Recall that we search for a function h_1 such that $\mathcal{K}_1^s h_1 = \phi$, for $\phi \in \langle q_e \rangle^\perp$. We add the function $\mathcal{K}_2^s h_2$ to this equation,

$$\mathcal{K}_1^s h_1 + \mathcal{K}_2^s h_2 = \phi + \mathcal{K}_2^s h_2, \quad (4.26)$$

which is equivalent to

$$\mathcal{K}^s h = \phi + \mathcal{K}_2^s h_2. \quad (4.27)$$

The right-hand side of this equation is in $\langle q_e \rangle^\perp$, since $\phi \in \langle q_e \rangle^\perp$ and

$$\begin{aligned} (\mathcal{K}_2^s h_2, q_e)_\Gamma &= \int_\Gamma \int_{\Gamma_2} G(\mathbf{x}, \mathbf{y}) h_2(\mathbf{y}) d\Gamma_y q_e(\mathbf{x}) d\Gamma_x \\ &= \int_{\Gamma_2} \int_\Gamma G(\mathbf{x}, \mathbf{y}) q_e(\mathbf{x}) d\Gamma_x h_2(\mathbf{y}) d\Gamma_y \\ &= (\mathcal{K}^s q_e, h_2)_{\Gamma_2} = -\frac{1}{2\pi} \log C_l(\Gamma) (1, h_2)_{\Gamma_2} = 0. \end{aligned} \quad (4.28)$$

Since \mathcal{K}^s is invertible on the function space $\langle q_e \rangle^\perp$, and the right-hand side of (4.27) is in $\langle q_e \rangle^\perp$, there exists a solution h of (4.27). The function h_1 is then the restriction of h to Γ_1 ■

Theorem 4.3 tells us that the BIE for the mixed problem does not have a unique solution when $C_l(\Gamma) = 1$, i.e. the BIE is singular. Moreover the division of Γ into a part Γ_1 with Dirichlet conditions and a part Γ_2 with Neumann conditions does not play a role in this. It does not make a difference whether we take Γ_1 very small or very large; the singular BIE relates solely to the whole boundary Γ .

To guarantee a unique solution for the mixed problem we have the same options as for the Dirichlet problem. The simplest remedy is to rescale the domain, thus avoiding a unit logarithmic capacity. A second option is to demand that the function q has a zero contour integral. Since part of q is already prescribed this yields the following condition for the unknown part of q ,

$$\int_{\Gamma_1} q_1 d\Gamma = - \int_{\Gamma_2} \tilde{q} d\Gamma. \quad (4.29)$$

As a last option to obtain nonsingular BIEs, we can also replace the single layer operator \mathcal{K}^s by \mathcal{K}_α^s , see Section 4.3.

4.5 Neumann problem

It is well known that the classic Neumann boundary value problem for the Laplace equation does not have a unique solution. Hence the corresponding BIE will also not have a unique solution. For completeness we prove the following theorem for the BIE (4.4) with Neumann boundary conditions in which it is shown that the Neumann problem has a solution which is unique up to a constant [25, p. 37].

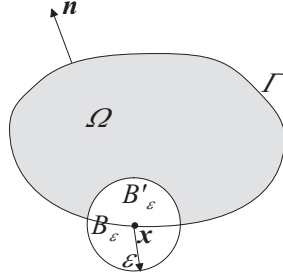


Figure 4.1: The point $x \in \Gamma$ is the center of a small circle B_ε with radius ε .

Theorem 4.4 For any closed curve Γ

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{K}^d\right)1 = 0. \quad (4.30)$$

Proof. To show that operator $\frac{1}{2}\mathcal{I} + \mathcal{K}^d$ applied to the constant function 1 yields zero, we need to prove that $\mathcal{K}^d 1 \equiv -\frac{1}{2}$ at the boundary. Let x be a point at the boundary Γ , then using Gauss' theorem we find

$$(\mathcal{K}^d 1)(x) = \int_{\Gamma} \frac{\partial}{\partial n_y} G(x, y) d\Gamma_y = \int_{\Omega} \nabla_y^2 G(x, y) d\Omega_y, \quad (4.31)$$

where the subscript y means integration or differentiation with respect to y . The fundamental solution is defined such that $\nabla_y^2 G(x, y) = -\delta(x - y)$. Hence, when integrating over y , we have to take special care at the point $y = x$. Let B_ε be a small circle with radius ε around the point x and let B'_ε be the part of that circle that lies inside Ω , i.e. $B'_\varepsilon = B_\varepsilon \cap \Omega$, see Figure 4.1. The domain integral in (4.31) can be split in

$$(\mathcal{K}^d 1)(x) = \int_{\Omega/B'_\varepsilon} \nabla_y^2 G(x, y) d\Omega_y + \int_{B'_\varepsilon} \nabla_y^2 G(x, y) d\Omega_y. \quad (4.32)$$

Within the domain Ω/B'_ε the fundamental solution does not have a singular point and thus $\nabla_y^2 G(x, y) = 0$ in this domain. As a consequence the first integral at the right-hand side of (4.32) is equal to zero. If the boundary Γ is smooth enough, the circle B'_ε is half the size of the circle B_ε . Likewise, if ε goes to zero, the integral over B'_ε in (4.32) is half the size of the same integral over B_ε . Hence we obtain

$$(\mathcal{K}^d 1)(x) = \frac{1}{2} \int_{B_\varepsilon} \nabla_y^2 G(x, y) d\Omega_y. \quad (4.33)$$

We use Gauss' theorem to transform the domain integral over B_ε to a boundary integral over Γ_ε , the boundary of the circle B_ε ,

$$(\mathcal{K}^d 1)(\mathbf{x}) = \frac{1}{2} \int_{\Gamma_\varepsilon} \frac{\partial}{\partial n_y} G(\mathbf{x}, \mathbf{y}) d\Gamma_y. \quad (4.34)$$

We introduce polar coordinates (r, θ) at the circle B_ε , the point \mathbf{x} being the local origin. Recall the definition of the fundamental solution (4.2) in which now $\|\mathbf{x} - \mathbf{y}\| \equiv r$ for $\mathbf{y} \in B_\varepsilon$. It is straightforward to see that

$$\frac{\partial}{\partial n_y} G(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi} \frac{\partial}{\partial r} \log \frac{1}{r} = -\frac{1}{2\pi} \frac{1}{r}. \quad (4.35)$$

Substituting this in the integral of (4.34) results in

$$(\mathcal{K}^d 1)(\mathbf{x}) = -\frac{1}{2} \int_0^{2\pi} \frac{1}{2\pi} \frac{1}{r} r d\theta = -\frac{1}{2}. \quad (4.36)$$

The direct consequence of this is that

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{K}^d\right)1 = 0, \quad (4.37)$$

with which Theorem 4.4 has been proven. ■

4.6 Examples

In this section we illustrate the results from the previous sections. We do this by calculating the condition number of the matrices that appear in the BEM. After discretisation of the boundary, the BIE transforms into a linear system of equations. If the BIE is singular, we may expect that the linear system is (almost) singular. As a consequence the condition number of the corresponding system matrix is very large.

For the BIE related to the Laplace equation with Dirichlet boundary equations we obtain the following linear system

$$\mathbf{G}\mathbf{q} = \mathbf{f}, \quad (4.38)$$

where $\mathbf{f} := \mathbf{f}(\tilde{u})$. We compute the condition number of \mathbf{G} for two cases: a circular domain with radius R and a square domain with side L . In both cases we choose $N = 36$ boundary elements.

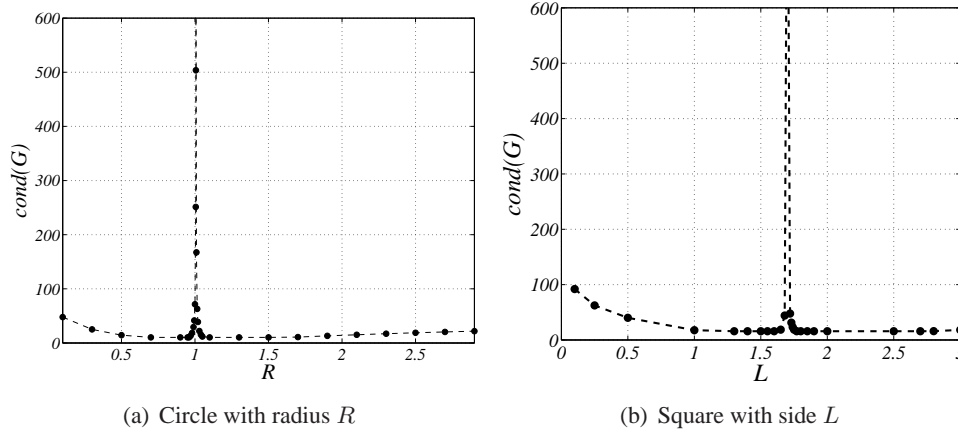


Figure 4.2: The condition number of \mathbf{G} for the Laplace equation with Dirichlet boundary conditions.

Example 4.1

For the circular domain the logarithmic capacity is equal to the radius of the circle, see Table 4.1. Thus if the radius R is equal to one also the logarithmic capacity is equal to one. In that case the BIE does not have a unique solution and the condition number of the matrix \mathbf{G} will be very large. In Figure 4.2(a) we show the condition number of \mathbf{G} as a function of the radius R . We observe that indeed the condition number goes to infinity when R approaches $R^* := 1$, cf. [19], [20]. Note that these observations are *a-posteriori* observations; first the matrix \mathbf{G} is constructed and then its condition number is computed. It is also possible to *a-priori* estimate the condition number for the Dirichlet problem on a circle [37, 38]. \square

Example 4.2

For the square domain the logarithmic capacity is approximately $0.59L$, see Table 4.1. Hence if the side length L is approximately equal to $L^* := 1/0.59 \approx 1.69$, then the logarithmic capacity is equal to one. Analogous to the case of the circle the condition number of the matrix \mathbf{G} is very large in that case. In Figure 4.2(b) the condition number of \mathbf{G} is plotted as a function of L . We observe that it grows to infinity when L approaches 1.69. \square

The scaling parameters R^* and L^* are called *critical* (or *degenerate*) *scalings*. Throughout this thesis we will adopt the term critical scaling. The corresponding

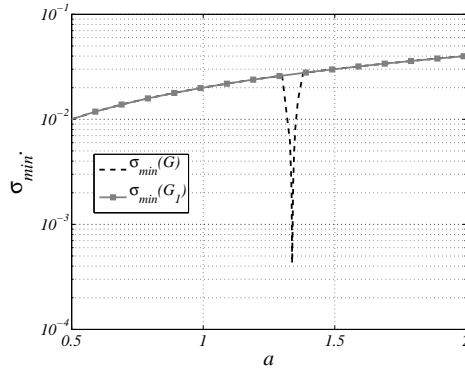


Figure 4.3: The smallest singular value of the matrices \mathbf{G} (circles) and \mathbf{G}_1 (diamonds) for the Laplace problem with mixed boundary conditions on an ellipsoidal domain.

domain is referred to as the *critical domain*. A description for the boundary of a critical domain which frequently appears in literature is Γ -contour [56].

It is obvious that we cannot use the boundary element formulation if the logarithmic capacity is equal to one. In Section 4.3 we suggested to search for solutions q of the Dirichlet BIE that also satisfy the requirement (4.15), i.e. have contour integral equal to zero. Translating this requirement to the discrete problem, we have to search for solutions \mathbf{q} that also satisfy $q_1 + \dots + q_N = 0$. Hence we have to solve the rectangular system

$$\underbrace{\begin{bmatrix} \mathbf{G} \\ \hline H \quad \dots \quad H \end{bmatrix}}_{\mathbf{G}_1} \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \\ 0 \end{bmatrix}. \quad (4.39)$$

Here the additional condition is multiplied by a non-zero scalar H in order to obtain a well-balanced matrix \mathbf{G}_1 . As we have a rectangular system, we require other solution techniques to solve the system, compared to the original system.

Example 4.3

To investigate the conditioning of the new system in (4.39), we compare the smallest singular values of \mathbf{G} and \mathbf{G}_1 . We illustrate this for an ellipsoidal domain with axis a and $a/2$. In Table 4.1 it is given that the logarithmic capacity of such an ellipse is equal to $3a/4$. Hence if we choose a equal to the critical value $a^* = 4/3$ the logarithmic capacity is equal to one. In Figure 4.3 both $\sigma_{\min}(\mathbf{G})$ (circles) and $\sigma_{\min}(\mathbf{G}_1)$ (diamonds) are plotted as a function of the scaling parameter a .

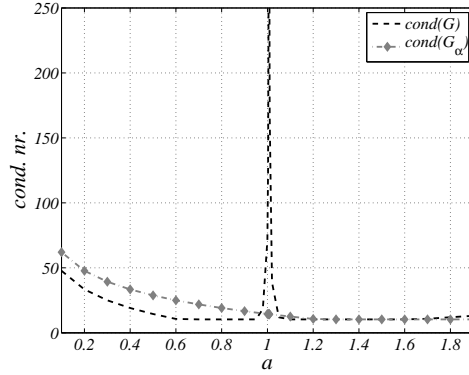


Figure 4.4: The condition number of \mathbf{G} (circles) and \mathbf{G}_α (diamonds) for a circular domain with radius a . Here $\alpha = 2$.

We observe that for $a = a^*$, $\sigma_{\min}(\mathbf{G})$ is going to zero, while $\sigma_{\min}(\mathbf{G}_1)$ remains $\mathcal{O}(10^{-2})$. Hence the system with \mathbf{G} is singular at $a = a^*$, while the system with \mathbf{G}_1 is nonsingular. \square

As explained in Section 4.3, a nonsingular BIE can also be obtained by replacing the integral operator \mathcal{K}^s by \mathcal{K}_α^s . This affects the matrix \mathbf{G} in the algebraic equations; the elements of the j -th column of \mathbf{G} is augmented with a factor $|\Gamma_j| \log \alpha / (2\pi)$, where $|\Gamma_j|$ is the length of the j -th boundary element. We denote this new matrix by \mathbf{G}_α .

Example 4.4

In Figure 4.4 we show the condition number of \mathbf{G} (circles) and \mathbf{G}_α (diamonds) for a circular domain with radius a . In this case we choose $\alpha = 2$. The condition number of the matrix \mathbf{G} goes to infinity as a approaches one, i.e. the unit circle. The condition number of \mathbf{G}_α remains bounded. \square

After discretisation of the boundary the BIE for the Laplace equation with mixed boundary conditions transforms in the linear system

$$\mathbf{G}\mathbf{q} = \left(\frac{1}{2}\mathbf{I} + \mathbf{H} \right) \mathbf{u}. \quad (4.40)$$

We assume that on the first part of the boundary Γ , represented by the first m ($0 \leq m \leq N$) elements, Dirichlet boundary conditions are given. On the remaining $N - m$ elements we have Neumann boundary conditions. This implies that the first m coefficients of \mathbf{u} and the last $N - m$ coefficients of \mathbf{q} are given. By

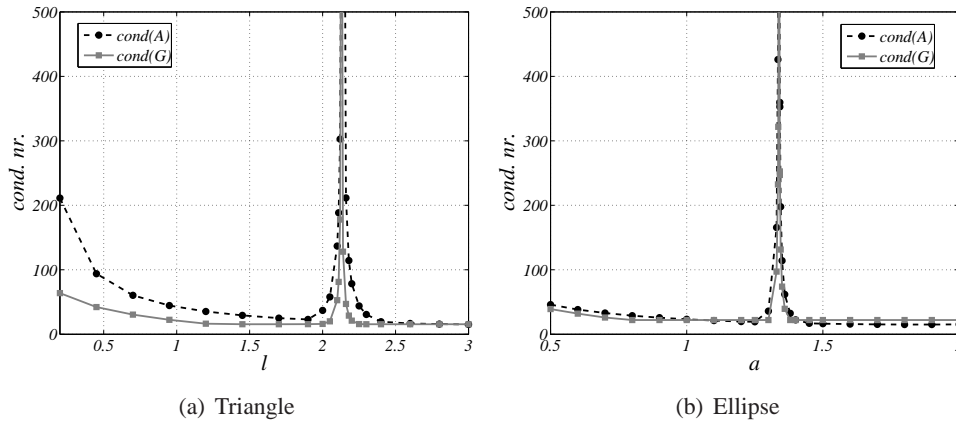


Figure 4.5: The condition number of the matrices \mathbf{A} (circles) and \mathbf{G} (squares) corresponding to the Laplace equation with mixed boundary conditions and Dirichlet boundary conditions respectively.

moving all unknown coefficients to the left-hand side and all known coefficients to the right-hand side in (4.40) we arrive at the standard form linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (4.41)$$

If the BIE is not uniquely solvable, then the linear system is singular, and the condition number of the matrix \mathbf{A} is very large. We illustrate this with two examples: a triangular domain and an ellipsoidal domain.

Example 4.5

The triangle is an isosceles right triangle with sides of length l . For such a triangle the logarithmic capacity is given by

$$C_l(\text{triangle}) = \frac{3^{3/4}\Gamma^2(1/4)}{2^{7/2}\pi^{3/2}} l \approx 0.476 l. \quad (4.42)$$

This implies that the condition number will be large when the scaling parameter l is close to the critical scaling $l^* := 1/0.476 \approx 2.1$. The ellipse has semi-axes of length a and $a/2$, which has logarithmic capacity equal to $3a/4$. Hence we may expect a large condition number when the scaling parameter a is close to $a^* := 4/3$.

In Figure 4.5 we show the condition numbers for the matrices \mathbf{A} and \mathbf{G} . We choose $N = 32$ and $m = N/2$, i.e. the number of elements with Dirichlet conditions is equal to the number of elements with Neumann conditions. The circles represent the condition number for the mixed problem, while the squares represent the condition number for the Dirichlet problem. We observe that the critical

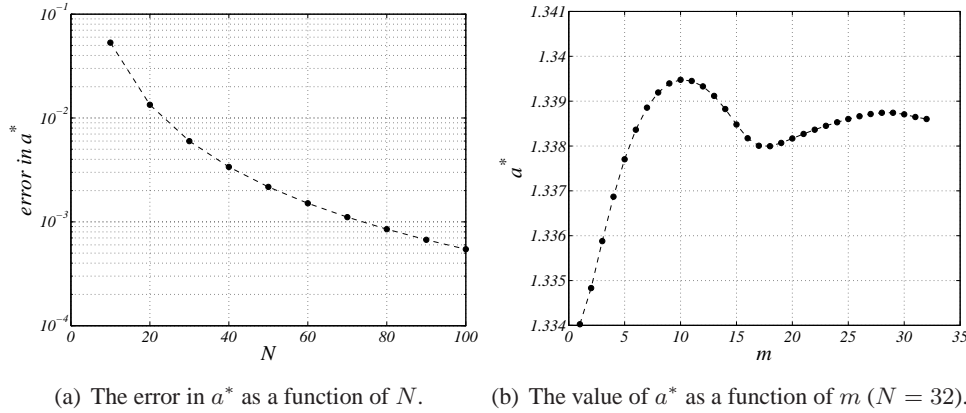


Figure 4.6: The critical scaling parameter a^* for which the condition number of \mathbf{A} goes to infinity.

scaling parameter is (almost) the same for both matrices \mathbf{A} and \mathbf{G} . The small difference that is present is caused by numerical inaccuracies due to the discretisation. Hence it does not matter whether we solve a Dirichlet problem or a mixed problem; for both problems there exists the same critical scaling of the domain such that the problems are not uniquely solvable. As was predicted for the triangle, the point where the condition number is very large is close to $l \approx 2.1$. For the ellipse we observe that indeed the point where the condition number is large is at $a \approx 1.3$. \square

Example 4.6

Figure 4.6 gives more details about the critical scaling for which the condition number goes to infinity. Again we consider an ellipsoidal domain with semi-axes of length a and $a/2$, having a critical scaling $a^* = 4/3$. This, of course, holds for a perfect ellipse. In reality we work with an approximation of an ellipse, namely a polygon with N sides. If N is large, the ellipse is approximated very well, and we expect to find a scaling parameter a^* that is close to that of the ellipse, i.e. $a^* = 4/3$. In Figure 4.6(a) we see the accuracy in a^* as a function of N . We observe that for large N the error between theoretical value and actual value gets very small.

In Section 4.4 it was already mentioned that the division of the boundary Γ into a Dirichlet and a Neumann part does not play a role in the singularity of the BIE. Figure 4.6(b) illustrates this. Here we vary m , the number of elements that have Dirichlet boundary conditions. The total number of elements is $N = 32$. Hence $m = 32$ corresponds to the Dirichlet problem, while $m = 1$ is a problem with Neumann conditions, except for one element. For each value of m we compute the critical

scaling a^* . We see that there is little change in the value of a^* as m varies between 1 and N . \square

As the mixed problem is ill-posed when the logarithmic capacity is equal to one, we may add an extra condition like we did for the Dirichlet problem. Since u satisfies the Laplace equation on the interior of the domain we know that q must have a zero contour integral. This leads to the following condition for the solution vector \mathbf{q} of the linear system in (4.40),

$$q_1 + \dots + q_m = -\gamma := -(\tilde{q}_{m+1} + \dots + \tilde{q}_N), \quad (4.43)$$

since part of the vector \mathbf{q} is already prescribed by the boundary condition at Γ_2 . Like we did for the Dirichlet case we formulate a new linear system in which the extra condition is incorporated,

$$\underbrace{\begin{bmatrix} \mathbf{A} \\ \hline H \dots H & 0 \dots 0 \end{bmatrix}}_{\mathbf{A}_1} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_N \\ -H\gamma \end{bmatrix}. \quad (4.44)$$

Here the additional condition is multiplied by a non-zero scalar H to obtain a well-balanced system matrix.

Example 4.7

We compare the smallest singular value $\sigma_{\min}(\mathbf{A})$ of the original matrix \mathbf{A} to the smallest singular value $\sigma_{\min}(\mathbf{A}_1)$ of the new matrix \mathbf{A}_1 for the example of the ellipse with semi-axes a and $a/2$. In Figure 4.7 we give the smallest singular values of the matrices \mathbf{A} (circles) and \mathbf{A}_1 (diamonds) as a function of the scaling parameter a . We observe that for the scaling $a = a^*$, $\sigma_{\min}(\mathbf{A})$ drops to zero, while $\sigma_{\min}(\mathbf{A}_1)$ remains $\mathcal{O}(10^{-2})$. Hence the system with \mathbf{A} is singular at $a = a^*$ while the system with \mathbf{A}_1 is nonsingular. \square

Another procedure to obtain a nonsingular BIE is to replace the integral operator \mathcal{K}^s by \mathcal{K}_α^s , see Section 4.4. This affects the matrix \mathbf{G} as described before, and as \mathbf{A} inherits a block from \mathbf{G} , also \mathbf{A} is affected. We denote the new matrix by \mathbf{A}_α .

Example 4.8

In Figure 4.8 we show the condition number of \mathbf{A} (circles) and \mathbf{A}_α (diamonds) for a circular domain with radius R . In this case we choose the parameter α equal to

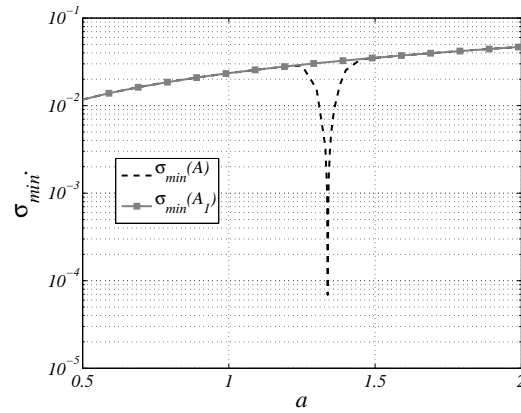


Figure 4.7: The condition number of the matrices \mathbf{A} (circles) and \mathbf{A}_1 (diamonds) for the Laplace problem with mixed boundary conditions on an ellipsoidal domain.

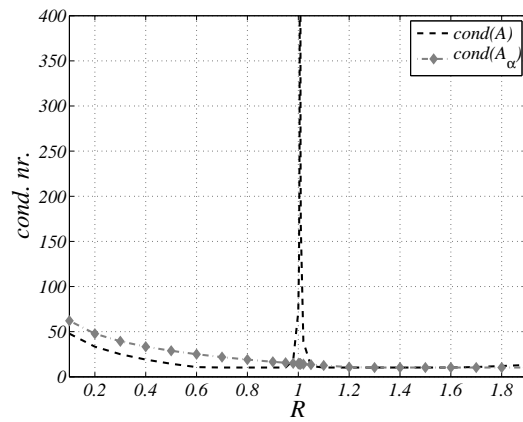


Figure 4.8: The condition number of \mathbf{A} (circles) and \mathbf{A}_α (diamonds) for a circular domain with radius R . Here $\alpha = 2$.

two. As we have seen before, the condition number of the matrix \mathbf{A} goes to infinity at $R = R^* := 1$. The condition number of \mathbf{A}_α remains bounded. \square

Another technique to ensure low condition numbers is by balancing the matrix elements of \mathbf{G} and \mathbf{H} with a suitable scaling parameter of the domain [72].

Chapter 5

Two-dimensional Stokes flow

It was observed in Chapter 3 and 4 that the boundary integral equation (BIE) for the Laplace equation may become singular at certain critical two-dimensional domains. The main cause for this lies in the logarithmic kernel that appears in the boundary integral operator. In the boundary integral operator for the Stokes equations on a two-dimensional domain the logarithmic kernel also appears. Therefore we may expect singular BIEs for the Stokes equations too. This is therefore the topic of this chapter.

The Stokes equations describe the flow of a viscous fluid. We briefly introduce the Stokes equations in differential form and the corresponding boundary integral formulation in dimensionless notation. For more details we refer to Chapter 6, in which the equations are derived in a three-dimensional setting. For the boundary conditions we either choose Dirichlet or mixed boundary conditions. We will show that for both cases there are critical sizes of the fluid domain for which the Stokes equations in boundary integral form are singular.

This chapter is an elaboration of earlier work of the author et al. [35] concerning condition numbers of BEM-matrices that appear for the 2D Stokes equations.

5.1 Boundary integral equations for 2D Stokes flow

The Laplace equation and the Stokes equations have at least one thing in common: the Laplace operator appears in both equations. As we have seen in previous chapters, the Laplace equation may lead to a singular BIE for certain critical domains. The question arises whether this is also the case for the Stokes equations: can the corresponding BIEs become singular on certain critical domains?

In this chapter we study the BIEs following from the Stokes equations. In particular we focus on the eigenvalues of the integral operators. It is shown that for certain critical domains these integral operators admit zero eigenvalues. Hence,

again we find that the BIEs become singular for a number of critical domains.

For the Laplace equation it is possible to *a-priori* determine the critical domains. For a number of simple domains, the logarithmic capacity can be used to exactly compute the critical size. For more involved domains the logarithmic capacity can be used to estimate the critical size of the domain. Unfortunately the critical domains for the Stokes equations do not coincide with the critical domains for the Laplace equations. Hence we cannot use the logarithmic capacity to *a-priori* determine the critical domains on which the BIEs for Stokes equations become singular. It is only by numerical experiments that we can distinguish the critical domains.

Let Ω be a two-dimensional simply-connected domain Ω with a smooth boundary Γ . The Stokes equations for a viscous flow in Ω read

$$\begin{aligned}\nabla^2 \mathbf{v} - \nabla p &= \mathbf{0}, \\ \nabla \cdot \mathbf{v} &= 0,\end{aligned}\tag{5.1}$$

where \mathbf{v} is the velocity field of the fluid and p its pressure. Let Γ be divided into a part Γ_1 on which the velocity \mathbf{v} is prescribed, and a part Γ_2 on which the pressure p is prescribed, $\Gamma = \Gamma_1 \cup \Gamma_2$. Hence the Stokes equations are subject to the boundary conditions

$$\begin{aligned}\mathbf{v} &= \tilde{\mathbf{v}}, \mathbf{x} \in \Gamma_1, \\ p &= \tilde{p}, \mathbf{x} \in \Gamma_2.\end{aligned}\tag{5.2}$$

Either Γ_1 or Γ_2 can be empty, leading to a purely Neumann or Dirichlet problem respectively. The Stokes equations in differential form can be transformed to a set of two BIEs [65, 75, 76]

$$\begin{aligned}\frac{1}{2}v_i(\mathbf{x}) + \int_{\Gamma} q_{ij}(\mathbf{x}, \mathbf{y})v_j(\mathbf{y})d\Gamma_{\mathbf{y}} \\ = \int_{\Gamma} u_{ij}(\mathbf{x}, \mathbf{y})b_j(\mathbf{y})d\Gamma_{\mathbf{y}}, \mathbf{x} \in \Gamma, i = 1, 2.\end{aligned}\tag{5.3}$$

Here a repeated index means summation over all possible values of that index. The vector function \mathbf{b} is the normal stress at the fluid boundary,

$$\mathbf{b} := \sigma(p, \mathbf{v})\mathbf{n},\tag{5.4}$$

with \mathbf{n} the outward unit normal at the boundary and the stress tensor σ defined by

$$\sigma_{ij}(p, \mathbf{v}) := -p\delta_{ij} + \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right).\tag{5.5}$$

Hence the boundary integral formulation involves two variables, the velocity \mathbf{v} and the normal stress \mathbf{b} . In correspondence to (5.2), at each point of the boundary either

\mathbf{v} or \mathbf{b} is prescribed,

$$\begin{aligned} \mathbf{v} &= \tilde{\mathbf{v}}, & \mathbf{x} &\in \Gamma_1, \\ \mathbf{b} &= -\tilde{p}\mathbf{n}, & \mathbf{x} &\in \Gamma_2. \end{aligned} \quad (5.6)$$

The kernels u_{ij} and q_{ij} in the integral equations are defined as

$$\begin{aligned} q_{ij}(\mathbf{x}, \mathbf{y}) &:= \frac{1}{\pi} \frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)n_k}{\|\mathbf{x} - \mathbf{y}\|^4}, \\ u_{ij}(\mathbf{x}, \mathbf{y}) &:= \frac{1}{4\pi} \left\{ \delta_{ij} \log \frac{1}{\|\mathbf{x} - \mathbf{y}\|} + \frac{(x_i - y_i)(x_j - y_j)}{\|\mathbf{x} - \mathbf{y}\|^2} \right\}, \end{aligned} \quad (5.7)$$

for $i, j = 1, 2$. We introduce boundary integral operators,

$$\begin{aligned} (\mathcal{G}\boldsymbol{\varphi})_i(\mathbf{x}) &:= \int_{\Gamma} u_{ij}(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) d\Gamma_{\mathbf{y}}, \\ (\mathcal{H}\boldsymbol{\psi})_i(\mathbf{x}) &:= \int_{\Gamma} q_{ij}(\mathbf{x}, \mathbf{y}) \psi_j(\mathbf{y}) d\Gamma_{\mathbf{y}}, \end{aligned} \quad (5.8)$$

which enables us to write (5.3) as

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{H}\right)\mathbf{v} = \mathcal{G}\mathbf{b}. \quad (5.9)$$

The operators \mathcal{G} and \mathcal{H} are called the *single and double layer operator* for the Stokes equations. For the Dirichlet problem the velocity \mathbf{v} at the boundary is given ($\Gamma_2 = \emptyset$) and we would like to reconstruct the normal stress \mathbf{b} at the boundary. To this end we need to invert the operator \mathcal{G} . This can only be done when all eigenvalues of \mathcal{G} are unequal to zero. In this chapter we investigate under which conditions \mathcal{G} admits a zero eigenvalue.

For the mixed problem the velocity at Γ_1 is prescribed and the normal stress at Γ_2 is prescribed. We would like to reconstruct the unknown velocity at Γ_2 and the unknown normal stress at Γ_1 . After rearranging known and unknown terms (see Section 5.4) we again need to invert a boundary integral operator. This can only be done when all eigenvalues of the operator are unequal to zero. We will show that zero eigenvalues occur under the same conditions as for the Dirichlet problem.

This chapter does not address the Stokes equations with Neumann conditions. It is well known that the corresponding BIEs do not have a unique solution, due to the existence of rigid body motions. The Completed Double Layer BEM [75] shows one way to get around this problem.

5.2 Eigensystem of the single layer operator \mathcal{G} at a circle

For a circular domain it is possible to compute the eigenvalues of the single layer operator \mathcal{G} analytically. These eigenvalues will show that for a circle with a particular

radius the BIE becomes singular. For general domains it is not possible to compute the eigenvalues of \mathcal{G} analytically. However the BIEs become also singular at certain domains. This phenomenon for arbitrary domains is presented in the next section. In the current section we compute eigenfunctions and eigenvalues of the operator \mathcal{G} for a circular boundary Γ . Here we make use of the fact that the first term in the kernel u_{ij} is related to the fundamental solution for the Laplace operator in 2D. This results in the following decomposition of the single layer operator \mathcal{G} for the Stokes equations,

$$(\mathcal{G}\varphi)_i(\mathbf{x}) = \frac{1}{2}\delta_{ij}(\mathcal{K}^s\varphi_j)(\mathbf{x}) + \frac{1}{4\pi} \int_{\Gamma} \frac{(x_i - y_i)(x_j - y_j)}{\|\mathbf{x} - \mathbf{y}\|^2} \varphi_j(\mathbf{y}) d\Gamma_y. \quad (5.10)$$

The operator \mathcal{K}^s is the single layer operator for the Laplace equation. For Γ a circle with radius R , the single-layer potential \mathcal{K}^s admits the following eigensystem [38],

$$\begin{aligned} \mathcal{K}^s \cos(kt) &= \frac{R}{2k} \cos(kt), \\ \mathcal{K}^s \sin(kt) &= \frac{R}{2k} \sin(kt), \\ \mathcal{K}^s 1 &= -R \log R, \end{aligned} \quad (5.11)$$

with $k \in \mathbb{N}$. We introduce the following polar coordinates,

$$\mathbf{x} := R[\cos t, \sin t], \quad \mathbf{y} := R[\cos s, \sin s], \quad 0 \leq t, s < 2\pi. \quad (5.12)$$

Then $\|\mathbf{x} - \mathbf{y}\|^2 = 2R^2(1 - \cos(t - s))$ and we find

$$\begin{aligned} u_{11}(t, s) &= \frac{1}{8\pi} \left\{ -\log [2R^2(1 - \cos(t - s))] + \frac{(\cos t - \cos s)^2}{1 - \cos(t - s)} \right\}, \\ u_{12}(t, s) &= \frac{1}{8\pi} \frac{(\cos t - \cos s)(\sin t - \sin s)}{1 - \cos(t - s)}, \\ u_{21}(t, s) &= \frac{1}{8\pi} \frac{(\sin t - \sin s)(\cos t - \cos s)}{1 - \cos(t - s)} = u_{12}(t, s), \\ u_{22}(t, s) &= \frac{1}{8\pi} \left\{ -\log [2R^2(1 - \cos(t - s))] + \frac{(\sin t - \sin s)^2}{1 - \cos(t - s)} \right\}. \end{aligned} \quad (5.13)$$

Using the integral kernels in polar coordinates given in (5.13) it is straightforward to compute the eigenvalues and eigenvectors of the operator \mathcal{G} . The results are given in Table 5.1.

We observe that the function $\mathbf{b}(\mathbf{x}) = \mathbf{n} = [\cos t, \sin t]^T$, representing the unit outward normal, is an eigenfunction with eigenvalue zero. This can be explained as follows: when we apply a stress in the direction of the normal at each point of the boundary, each stress having the same magnitude, the net contribution will be equal

eigenvalue	eigenfunctions
0	$\begin{bmatrix} \cos t \\ \sin t \end{bmatrix}$
$\frac{R}{2}$	$\begin{bmatrix} -\sin t \\ \cos t \end{bmatrix}$
$\frac{R}{4}$	$\begin{bmatrix} \cos t \\ -\sin t \end{bmatrix}, \begin{bmatrix} \sin t \\ \cos t \end{bmatrix}$
$\frac{R}{4k}, (k = 2, 3, \dots)$	$\begin{bmatrix} \cos(kt) \\ \sin(kt) \end{bmatrix}, \begin{bmatrix} \cos(kt) \\ -\sin(kt) \end{bmatrix},$ $\begin{bmatrix} \sin(kt) \\ \cos(kt) \end{bmatrix}, \begin{bmatrix} -\sin(kt) \\ \cos(kt) \end{bmatrix}$
$-\frac{1}{2}R \log R + \frac{1}{4}R$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Table 5.1: Eigenvalues and eigenfunctions of the single layer operator \mathcal{G} .

to zero. This phenomenon is not restricted to the circle, but applies to any shape of the boundary, as will be proven in Section 5.3. Hence, when solving the BIE (5.9) with Dirichlet boundary conditions, one has to exclude the normal from the solution space.

For the case $R = R^* := \exp(1/2)$, we see that there is another zero eigenvalue, which has an eigenspace with dimension two. The eigenfunctions correspond to a uniform stress distribution; at each point of the boundary an equal stress is applied in the same direction. This particular stress distribution will cause a translation of the body. The circular boundary with $R = R^*$ is called a *critical boundary* and the circle itself a *critical domain*. The radius or scale R^* is called the *critical scale* or *degenerate scale*. Equation (5.9) with Dirichlet conditions at the critical boundary cannot be solved as the operator \mathcal{G} admits a zero eigenvalue. The corresponding BIE is singular and does not have a unique solution.

The situation in which the BIE is not uniquely solvable for a circle with critical scale also occurs for the Dirichlet Laplace equation in 2D [19, 20, 22, 24, 38]. It is shown that the single-layer operator for the unit circle admits a zero eigenvalue, and consequently it cannot be inverted. Hence, the same phenomenon appears for both the Laplace and Stokes equations. The critical scale however is not the same for the Laplace and Stokes equations.

5.3 Invertibility of single layer operator on general domain

In this section we study the solvability of the BIE (5.9) with Dirichlet conditions on a smooth closed boundary Γ . We search for eigenfunctions \mathbf{b} of the boundary integral operator \mathcal{G} with zero eigenvalue, hence $\mathcal{G}\mathbf{b} = 0$. If such eigenfunctions exist, the boundary integral operator \mathcal{G} is not invertible and the integral equation (5.9) is not uniquely solvable. First we show that at least one such eigenfunction with zero eigenvalue exists.

Theorem 5.1 *For any smooth boundary Γ the outward unit normal $\mathbf{n}(\mathbf{x})$ is an eigenfunction of the boundary integral operator \mathcal{G} with eigenvalue zero.*

Proof. The i -th component of $\mathcal{G}\mathbf{n}$ equals

$$\begin{aligned}
 (\mathcal{G}\mathbf{n})_i &= \int_{\Gamma} u_{ij}(\mathbf{x}, \mathbf{y}) n_j(\mathbf{y}) d\Gamma_y \\
 &= \frac{1}{4\pi} \int_{\Gamma} \left[\delta_{ij} \log \frac{1}{\|\mathbf{x} - \mathbf{y}\|} + \frac{(x_i - y_i)(x_j - y_j)}{\|\mathbf{x} - \mathbf{y}\|^2} \right] n_j(\mathbf{y}) d\Gamma_y \\
 &= \frac{1}{4\pi} \int_{\Omega} \frac{\partial}{\partial x_j} \left[\delta_{ij} \log \frac{1}{\|\mathbf{x} - \mathbf{y}\|} + \frac{(x_i - y_i)(x_j - y_j)}{\|\mathbf{x} - \mathbf{y}\|^2} \right] d\Omega_y \\
 &= - \int_{\Omega} \frac{\partial}{\partial x_j} u_j^i d\Omega_y = - \int_{\Omega} \nabla \cdot \mathbf{u}^i d\Omega = 0.
 \end{aligned} \tag{5.14}$$

Here the vector \mathbf{u}^i is the velocity due to a *Stokeslet* [65], i.e. the velocity field induced by a point force in the e_i -direction. This velocity field satisfies the incompressibility condition $\nabla \cdot \mathbf{u}^i = 0$. ■

In Section 5.2 we already saw that for a circular boundary the normal vector \mathbf{n} is an eigenfunction of \mathcal{G} with eigenvalue zero. The current theorem generalizes this result to arbitrary smooth closed boundaries. In the sequel of this section we assume that the solutions of the Dirichlet problem (5.9) are sought in a function space that excludes the normal. Hence the eigenfunctions \mathbf{b} of \mathcal{G} we are looking for are perpendicular to \mathbf{n} .

We now show that for each boundary Γ there exist (at most) two critical scalings of the boundary such that the operator \mathcal{G} in the Dirichlet problem (5.9) is not invertible. This phenomenon has been observed and proven [41] and we will partly present the analysis here. In analogy to Section 5.2 the scaling for which the operator \mathcal{G} is not invertible is called a *critical scaling*, and the corresponding boundary a *critical boundary*. The domain that is enclosed by the critical boundary is referred to as the *critical domain*.

Theorem 5.2 For all given functions f and constant vectors d the system of equations

$$\begin{cases} \mathcal{G}b + c = f, \\ \int_{\Gamma} b d\Gamma = d, \end{cases} \quad (5.15)$$

has a unique solution pair (b, c) , where b is a function and c a constant vector.

Sketch of proof. The main idea is to show that the operator that maps the pair (b, c) to the left-hand side of (5.15) is an isomorphism [41]. ■

We proceed by introducing the two unit vectors $e_1 = [1, 0]^T$ and $e_2 = [0, 1]^T$. Theorem 5.2 guarantees that two pairs (b^1, c^1) and (b^2, c^2) exist that are the unique solutions to the two systems

$$\begin{cases} \mathcal{G}b^1 + c^1 = 0, \\ \int_{\Gamma} b^1 d\Gamma = e_1, \end{cases} \quad \begin{cases} \mathcal{G}b^2 + c^2 = 0, \\ \int_{\Gamma} b^2 d\Gamma = e_2. \end{cases} \quad (5.16)$$

We define the matrix C_{Γ} as $C_{\Gamma} := [c^1 | c^2]$.

Theorem 5.3 If $\det(C_{\Gamma}) = 0$, then the operator \mathcal{G} is not invertible.

Proof. Suppose that $\det(C_{\Gamma}) = 0$, then the columns c^1 and c^2 are dependent, say $c^1 = \alpha c^2$ for some $\alpha \in \mathbb{R}$, $\alpha \neq 0$. In that case

$$\begin{aligned} \mathbf{0} &= (\mathcal{G}b^1 + c^1) - \alpha(\mathcal{G}b^2 + c^2) \\ &= \mathcal{G}(b^1 - \alpha b^2) + \alpha c^2 - \alpha c^2 \\ &= \mathcal{G}(b^1 - \alpha b^2). \end{aligned} \quad (5.17)$$

The function $b^1 - \alpha b^2$ cannot be equal to zero, since this requires $\int_{\Gamma} (b^1 - \alpha b^2) d\Gamma$ to be equal to zero, while we have

$$\int_{\Gamma} (b^1 - \alpha b^2) d\Gamma = e_1 - \alpha e_2 \neq \mathbf{0}. \quad (5.18)$$

So $b^1 - \alpha b^2$ is an eigenfunction of \mathcal{G} with zero eigenvalue. This eigenfunction cannot be equal to the normal \mathbf{n} , since \mathbf{n} also requires $\int_{\Gamma} \mathbf{n} d\Gamma = 0$. ■

Corollary 5.4 There are (at most) two critical scalings of the domain Γ for which the operator \mathcal{G} is not invertible.

Proof. We rescale the domain Γ by a factor a , i.e. $\Gamma \rightarrow a\Gamma$. With the definition of the operator \mathcal{G} it can be shown that

$$\mathcal{G}b \rightarrow \mathcal{G}^a b := -\frac{1}{4\pi} \int_{\Gamma} a \log a b d\Gamma + a\mathcal{G}b. \quad (5.19)$$

Then the two systems in (5.16) change into

$$\begin{cases} a\mathcal{G}\mathbf{b}^j + \mathbf{c}^j - \frac{1}{4\pi} \int_{\Gamma} a \log a \mathbf{b}^j d\Gamma = \mathbf{0}, \\ a \int_{\Gamma} \mathbf{b}^j d\Gamma = \mathbf{e}_j, j = 1, 2. \end{cases} \quad (5.20)$$

Define $\mathbf{b}_a^j := a\mathbf{b}^j$ for $j = 1, 2$, then we obtain

$$\begin{cases} \mathcal{G}\mathbf{b}_a^j + \mathbf{c}^j - \frac{1}{4\pi} \int_{\Gamma} \log a \mathbf{b}_a^j d\Gamma = \mathbf{0}, \\ \int_{\Gamma} \mathbf{b}_a^j d\Gamma = \mathbf{e}_j, j = 1, 2. \end{cases} \quad (5.21)$$

Substituting the second equation into the first equation, we get

$$\begin{cases} \mathcal{G}\mathbf{b}_a^j + \mathbf{c}^j - \frac{1}{4\pi} \log a \mathbf{e}_j = \mathbf{0}, \\ \int_{\Gamma} \mathbf{b}_a^j d\Gamma = \mathbf{e}_j, j = 1, 2. \end{cases} \quad (5.22)$$

These systems have the same form as the original systems in (5.16), except for the change $\mathbf{c}^j \rightarrow \mathbf{c}^j - \frac{1}{4\pi} \log a \mathbf{e}_j$ for $j = 1, 2$. Define the new matrix $\mathbf{C}_{a\Gamma}$ by

$$\mathbf{C}_{a\Gamma} := \mathbf{C}_{\Gamma} - \frac{1}{4\pi} \log a \mathbf{I}_2, \quad (5.23)$$

then \mathcal{G}^a is not invertible when $\det(\mathbf{C}_{a\Gamma}) = 0$. Hence, when $\frac{1}{4\pi} \log a$ is an eigenvalue of \mathbf{C}_{Γ} , the operator \mathcal{G}^a is not invertible. This implies that, when \mathbf{C}_{Γ} has two distinct eigenvalues, there are two critical scalings a for which \mathcal{G}^a is not invertible. If \mathbf{C}_{Γ} has one eigenvalue with double multiplicity these critical scalings coincide. ■

The result of Corollary 5.4 shows that the BIEs for the Dirichlet Stokes equations become singular for certain sizes of the domain. As a consequence, the equations are not uniquely solvable. This solvability problem is an artifact of the boundary integral formulation; the Stokes equations in differential form always have a unique solution. In Chapter 3 and 4, and in literature [53, 56, 75, 85], a similar phenomenon is observed for the Laplace equation with Dirichlet conditions; in its differential form the problem is well-posed, while the corresponding BIE is not solvable at critical boundaries.

For the Laplace equation the critical scaling is related to the logarithmic capacity of the domain. By calculating or estimating the logarithmic capacity one can determine or estimate the critical domains, without computing BEM matrices and evaluating their condition numbers. It is this *a-priori* information that allows us to modify the standard BEM formulation such that the BIE become uniquely solvable (see Section 5.3).

For the Stokes equations, there does not exist an equivalent to the logarithmic capacity. Hence we cannot *a-priori* determine the critical domains. One way to determine the critical domains is by computing the BEM-matrices and evaluating

Section 5.4 Invertibility of operator on general domain with mixed conditions

their condition numbers. If the condition number jumps to infinity for a certain domain, then this domain is a critical domain. Hence, this strategy requires the solution of many BEM problems.

Another possibility to determine the critical domains is by solving the systems in (5.16). This yields the matrix \mathbf{C}_Γ , and subsequently the matrix $\mathbf{C}_{a\Gamma}$. By calculating the eigenvalues of the latter matrix the critical scalings can be found. Again we have to solve two non-standard BEM problems to compute the critical scalings.

REMARK: The BIEs for the Stokes flow in 2D are similar to the equations for plane elasticity. Hence the BIEs for the latter equations suffer from the same solvability problems as the Stokes equations. A proof of this phenomenon for plane elasticity is found in literature [95] and is similar to the proof sketched above.

5.4 Invertibility of operator on general domain with mixed conditions

In the previous section we showed that the boundary integral operator \mathcal{G} for the Dirichlet Stokes equations is not invertible for all domains. In this section we show that this phenomenon extends to the Stokes equations with mixed boundary conditions.

The starting point is again the BIE for the Stokes equations,

$$\frac{1}{2}\mathbf{v} + \mathcal{H}\mathbf{v} = \mathcal{G}\mathbf{b}, \text{ at } \Gamma. \quad (5.24)$$

Suppose that the boundary Γ is split into two parts, $\Gamma = \Gamma_1 \cup \Gamma_2$. On Γ_1 we prescribe the velocity \mathbf{v}^1 while the normal stress \mathbf{b}^1 is unknown. On Γ_2 we prescribe the normal stress \mathbf{b}^2 while the velocity \mathbf{v}^2 is unknown. The boundary integral operators \mathcal{G} and \mathcal{H} are split accordingly,

$$\begin{aligned} [\mathcal{G}\mathbf{b}]_i &= \int_{\Gamma} u_{ij} b_j d\Gamma = \int_{\Gamma_1} u_{ij} b_j^1 d\Gamma + \int_{\Gamma_2} u_{ij} b_j^2 d\Gamma =: [\mathcal{G}^1 \mathbf{b}^1]_i + [\mathcal{G}^2 \mathbf{b}^2]_i, \\ [\mathcal{H}\mathbf{v}]_i &= \int_{\Gamma} q_{ij} v_j d\Gamma = \int_{\Gamma_1} q_{ij} v_j^1 d\Gamma + \int_{\Gamma_2} q_{ij} v_j^2 d\Gamma =: [\mathcal{H}^1 \mathbf{v}^1]_i + [\mathcal{H}^2 \mathbf{v}^2]_i. \end{aligned} \quad (5.25)$$

With these notations the BIE is written in the following way,

$$\frac{1}{2}\mathbf{v}^k + \mathcal{H}^1 \mathbf{v}^1 + \mathcal{H}^2 \mathbf{v}^2 = \mathcal{G}^1 \mathbf{b}^1 + \mathcal{G}^2 \mathbf{b}^2, \text{ at } \Gamma_k, \quad k = 1, 2. \quad (5.26)$$

We arrange the terms in such a way that all unknowns are at the left-hand side and all

knowns are at the right-hand side,

$$\begin{aligned}\mathcal{H}^2\mathbf{v}^2 - \mathcal{G}^1\mathbf{b}^1 &= \mathcal{G}^2\mathbf{b}^2 - \mathcal{H}^1\mathbf{v}^1 - \frac{1}{2}\mathbf{v}^1, \text{ at } \Gamma_1, \\ \frac{1}{2}\mathbf{v}^2 + \mathcal{H}^2\mathbf{v}^2 - \mathcal{G}^1\mathbf{b}^1 &= \mathcal{G}^2\mathbf{b}^2 - \mathcal{H}^1\mathbf{v}^1, \text{ at } \Gamma_2.\end{aligned}\quad (5.27)$$

Now we can define an operator \mathcal{A} that assigns to the pair $(\mathbf{b}^1, \mathbf{v}^2)$ the two functions at the left-hand side of (5.27),

$$\begin{bmatrix} \mathbf{b}^1 \\ \mathbf{v}^2 \end{bmatrix} \xrightarrow{\mathcal{A}} \begin{bmatrix} \mathcal{H}^2\mathbf{v}^2 - \mathcal{G}^1\mathbf{b}^1 \\ \frac{1}{2}\mathbf{v}^2 + \mathcal{H}^2\mathbf{v}^2 - \mathcal{G}^1\mathbf{b}^1 \end{bmatrix}\quad (5.28)$$

To study the invertibility of this operator we need to study the homogeneous version of the equations in (5.27),

$$\begin{aligned}\mathcal{H}^2\mathbf{v}^2 - \mathcal{G}^1\mathbf{b}^1 &= 0, \text{ at } \Gamma_1, \\ \frac{1}{2}\mathbf{v}^2 + \mathcal{H}^2\mathbf{v}^2 - \mathcal{G}^1\mathbf{b}^1 &= 0, \text{ at } \Gamma_2.\end{aligned}\quad (5.29)$$

Theorem 5.5 *There are (at most) two scalings of Γ such that the homogeneous equations (5.29) have a non-trivial solution, i.e. \mathcal{A} is not invertible.*

Proof. From the Dirichlet problem we know that there are (at most) two scalings of Γ for which \mathcal{G} is not invertible. So there are a_I and $a_{II} \in \mathbb{R}$ and vector functions \mathbf{q}_I and \mathbf{q}_{II} such that

$$\mathcal{G}\mathbf{q}_k = \mathbf{0}, \text{ at } a_k\Gamma, \quad k = I, II. \quad (5.30)$$

The scalings a_k and functions \mathbf{q}_k may coincide but this does not affect the proof of the theorem. Denote the nullspace of the operator \mathcal{G} by $N(\mathcal{G})$. The normal vector \mathbf{n} is always in $N(\mathcal{G})$. If Γ is a critical boundary then also \mathbf{q}_I and \mathbf{q}_{II} are in $N(\mathcal{G})$. Assume that Γ is such a critical boundary. Let $\mathbf{q} \in N(\mathcal{G})$, that is $\mathbf{q} = \alpha_1\mathbf{q}_I + \alpha_2\mathbf{q}_{II} + \alpha_3\mathbf{n}$, for some α_1, α_2 and α_3 in \mathbb{R} , and $\mathcal{G}\mathbf{q} = \mathbf{0}$. Consider the homogeneous equations (5.29). We will show that there is a non-trivial pair $(\mathbf{b}^1, \mathbf{v}^2)$ that satisfies these equations. We choose $\mathbf{v}^2 = \mathbf{0}$ and $\mathbf{b}^1 = \mathbf{q}^1 + \mathbf{h}^1$. Here \mathbf{q}^i is the restriction of \mathbf{q} to Γ_i , $i = 1, 2$, and \mathbf{h}^1 is still unknown. Substituting these functions in the left-hand sides of the homogeneous equations yields for both equations

$$-\mathcal{G}^1\mathbf{b}^1 = -\mathcal{G}^1\mathbf{q}^1 - \mathcal{G}^1\mathbf{h}^1 = \mathcal{G}^2\mathbf{q}^2 - \mathcal{G}^1\mathbf{h}^1, \quad (5.31)$$

where we made use of $\mathcal{G}\mathbf{q} = \mathcal{G}^1\mathbf{q}^1 + \mathcal{G}^2\mathbf{q}^2 = \mathbf{0}$. If we can find a function \mathbf{h}^1 such that $\mathcal{G}^1\mathbf{h}^1 = \mathcal{G}^2\mathbf{q}^2$, then the left-hand sides of the homogeneous equations yield zero and we have found a non-trivial solution. Our task is then to prove that a function \mathbf{h}^1 exists such that

$$\mathcal{G}^1\mathbf{h}^1 = \mathcal{G}^2\mathbf{q}^2. \quad (5.32)$$

Section 5.4 Invertibility of operator on general domain with mixed conditions

A function \mathbf{b} is in $N(\mathcal{G})^\perp$ if it is perpendicular to $\mathbf{q} \in N(\mathcal{G})^\perp$. Hence, the inner product of \mathbf{q} and \mathbf{b} should be equal to zero, i.e.

$$\int_{\Gamma} q_i(\mathbf{x}) b_i(\mathbf{x}) d\Gamma_x = 0. \quad (5.33)$$

First we note that the right-hand side of (5.32) is contained in $N(\mathcal{G})^\perp$, since

$$\begin{aligned} \int_G q_i [\mathcal{G}^2 \mathbf{q}^2]_i d\Gamma_x &= \int_{\Gamma} q_i(\mathbf{x}) \int_{\Gamma_2} u_{ij}(\mathbf{x}, \mathbf{y}) q_j^2(\mathbf{y}) d\Gamma_y d\Gamma_x \\ &= \int_{\Gamma_2} q_j^2(\mathbf{y}) \int_{\Gamma} u_{ij}(\mathbf{x}, \mathbf{y}) q_i(\mathbf{x}) d\Gamma_x d\Gamma_y \\ &= \int_{\Gamma_2} q_j^2(\mathbf{y}) [\mathcal{G} \mathbf{q}]_j d\Gamma_y = 0. \end{aligned} \quad (5.34)$$

So we may generalize our task: prove that a solution \mathbf{h}^1 of

$$\mathcal{G}^1 \mathbf{h}^1 = \phi \quad (5.35)$$

exists, for any $\phi \in N(\mathcal{G})^\perp$. If so, then $\phi = \mathcal{G}^2 \mathbf{q}^2$ completes the proof. It is known that the operator \mathcal{G} is a Fredholm operator with index zero [29]. Hence the Fredholm alternative can be applied [70, p. 37]. This states that the homogeneous equation $\mathcal{G} \mathbf{h} = \mathbf{0}$ either has the trivial solution, or a set of non-trivial linearly independent solutions. We are in the second situation, as the nullspace of \mathcal{G} is non-empty and is spanned by \mathbf{n} , \mathbf{q}_I and \mathbf{q}_{II} . The Fredholm alternative further states that $\mathcal{G} \mathbf{h} = \phi$ is solvable if and only if ϕ is perpendicular to the solutions of $\mathcal{G}^* \mathbf{f} = \mathbf{0}$, where \mathcal{G}^* denotes the adjoint operator of \mathcal{G} . However, the single layer operator for the Stokes equations is self-adjoint, so $\mathcal{G}^* = \mathcal{G}$. Consequently the solvability condition says that ϕ has to be perpendicular to the solutions of $\mathcal{G} \mathbf{h} = \mathbf{0}$. This is the case since we defined $\phi \in N(\mathcal{G})^\perp$. Consequently there exists an \mathbf{h} with $\mathcal{G} \mathbf{h} = \phi$, for $\phi \in N(\mathcal{G})^\perp$. We split this \mathbf{h} into two parts,

$$\mathbf{h} = \begin{cases} \mathbf{h}^1, & \text{at } \Gamma_1, \\ \mathbf{h}^2, & \text{at } \Gamma_2. \end{cases} \quad (5.36)$$

Recall that we search for \mathbf{h}^1 such that $\mathcal{G}^1 \mathbf{h}^1 = \phi$. We add $\mathcal{G}^2 \mathbf{h}^2$ to both sides of this equation, obtaining

$$\mathcal{G}^1 \mathbf{h}^1 + \mathcal{G}^2 \mathbf{h}^2 = \phi + \mathcal{G}^2 \mathbf{h}^2, \quad (5.37)$$

or shorter

$$\mathcal{G} \mathbf{h} = \phi + \mathcal{G}^2 \mathbf{h}^2. \quad (5.38)$$

The right-hand side of this equation is in $N(\mathcal{G})^\perp$, since $\phi \in N(\mathcal{G})^\perp$ and

$$\begin{aligned} \int_G q_i [\mathcal{G}^2 \mathbf{h}^2]_i d\Gamma_x &= \int_\Gamma q_i(\mathbf{x}) \int_{\Gamma_2} u_{ij}(\mathbf{x}, \mathbf{y}) h_j^2(\mathbf{y}) d\Gamma_y d\Gamma_x \\ &= \int_{\Gamma_2} h_j^2(\mathbf{y}) \int_\Gamma u_{ij}(\mathbf{x}, \mathbf{y}) q_i(\mathbf{x}) d\Gamma_x d\Gamma_y \\ &= \int_{\Gamma_2} h_j^2(\mathbf{y}) [\mathcal{G}\mathbf{q}]_j d\Gamma_y = 0, \end{aligned} \quad (5.39)$$

so $\mathcal{G}^2 \mathbf{h}^2$ is also in $N(\mathcal{G})^\perp$. Recall that $\mathcal{G}\mathbf{h} = \phi + \mathcal{G}^2 \mathbf{h}^2$ is solvable if the right-hand side is in $N(\mathcal{G})^\perp$, so there exists an \mathbf{h} satisfying (5.38). Then we may construct \mathbf{h}^1 by simply restricting \mathbf{h} to Γ_1 , i.e. $\mathbf{h}^1 = \mathbf{h}|_{\Gamma_1}$. ■

This result shows that also the BIE for the Stokes equations with mixed boundary conditions may become singular. This happens for the same critical boundaries as for the Stokes equations with Dirichlet boundary conditions. Hence the mixed problem inherits the singularities from the Dirichlet problem. The division of the boundary into a Dirichlet and a Neumann part does not play a role in this.

Note that the Laplace equation exhibits the same behaviour. The boundary integral equation for the Laplace equation with mixed boundary conditions also inherits the solvability problems from the BIE for the Dirichlet case [39].

5.5 Numerical examples

To solve the BIEs (5.9), the boundary Γ is discretised into a set of N linear elements. At each element the velocity \mathbf{v} and normal stress \mathbf{b} are approximated linearly. In this way the BIEs are transformed into a linear system of algebraic equations (for details about the discretisation we refer to any BEM handbook [6, 10]). We introduce vectors \mathbf{v} and \mathbf{b} of length $2N$ containing the coefficients of \mathbf{v} and \mathbf{b} at the nodal points. Then the system of equations can be written in short-hand notation as

$$\left(\frac{1}{2}\mathbf{I} + \mathbf{H}\right)\mathbf{v} = \mathbf{G}\mathbf{b}. \quad (5.40)$$

Here \mathbf{G} and \mathbf{H} are the discrete counterparts of the single and double layer operator.

For the Dirichlet problem, the coefficients of the velocity vector \mathbf{v} are given, say $\mathbf{v} = \tilde{\mathbf{v}}$, and we need to solve

$$\mathbf{G}\mathbf{b} = \mathbf{f}(\tilde{\mathbf{v}}) := \left(\frac{1}{2}\mathbf{I} + \mathbf{H}\right)\tilde{\mathbf{v}} \quad (5.41)$$

to find the unknown coefficients of the normal stress vector \mathbf{b} .

If the boundary integral operator \mathcal{G} is not invertible then its discrete counterpart, the matrix \mathbf{G} is ill-conditioned. To visualize this we compute the condition number of \mathbf{G} : if the condition number is infinitely large, then the matrix is not invertible. As a consequence, the linear system (5.41) is singular and cannot be solved for arbitrary right-hand side vectors \mathbf{f} . If the condition number is bounded but very large, then still the problem (5.41) is difficult to solve numerically.

In the following examples we construct the matrix \mathbf{G} for a certain boundary Γ and compute the condition number of this matrix. Then we rescale the boundary Γ by a factor a , i.e. $\Gamma \rightarrow a\Gamma$. Again we compute the condition number of the matrix \mathbf{G} . We do this for several values of the scaling parameter a . According to the theory in the previous sections there are two critical scalings for which the integral operator \mathcal{G} is not invertible. For these two scalings also the matrix \mathbf{G} is not invertible, or at least very ill-conditioned. Hence we expect that the condition number of \mathbf{G} jumps to infinity at these two scalings. The scaling for which such large condition numbers appear is again called a *critical scaling* and has ideally the same value as the critical scaling defined for the BIE in the previous section. However, due to the discretisation of the equations, the critical scaling for the discrete problem may be slightly different from the critical scaling for the BIE. In the limit $N \rightarrow \infty$ these differences vanish. Analogously to the critical scaling, we define the *critical boundary* to be the boundary for which the condition number gets very large.

Example 5.1

In Figure 5.1 we show the condition number as a function of the scale a . We do this for an ellipse with aspect ratio 0.4 (to which we refer to as ellipse 1) and for an ellipse with aspect ratio 0.7 (to which we refer to as ellipse 2). We observe that for both cases two critical scalings exist for which the condition number goes to infinity. Moreover, these critical scalings differ significantly for the two ellipses. For ellipse 1 we find critical scalings 1.9 and 2.9 approximately, while for ellipse 2 we find 1.8 and 2.1. Hence the shape of the ellipse, i.e. its aspect ratio, greatly influences the values of the critical scalings.

Figure 5.2 visualises all ellipses for which the condition number of \mathbf{G} is very large, i.e. all critical ellipses with different aspect ratios. At the horizontal axis we put the length a of the horizontal semi-axis of the ellipse, at the vertical axis the length b of the vertical semi-axis of the ellipse. We compute the condition number of \mathbf{G} for several values of a and b . We call the values of a and b for which the condition number goes to infinity the *critical sizes* and the corresponding ellipse the critical ellipse. At the *critical sizes* we plot a dot in the (a, b) -plane of Figure 5.2. We see that the critical sizes lie on two curved lines, which are symmetric around the line $a = b$. It can be concluded that for an ellipse with fixed aspect ratio $d := a/b \neq 1$,

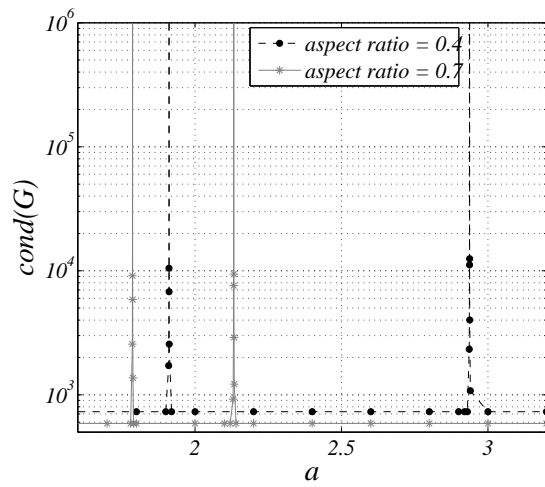


Figure 5.1: Condition number of \mathbf{G} for an ellipsoidal domain with aspect ratios 0.4 and 0.7 as a function of scaling parameter a .

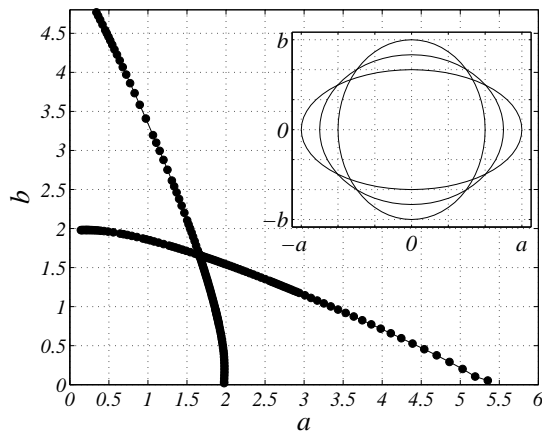


Figure 5.2: The critical sizes of an ellipse for which the condition number of \mathbf{G} is very large.

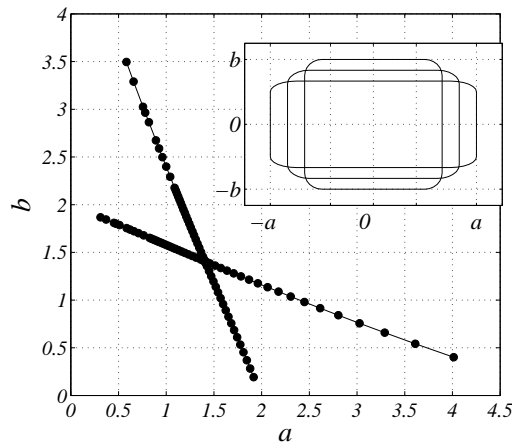


Figure 5.3: The critical sizes of a rectangular domain with rounded corners.

two critical sizes exist. For a circle, where $d = 1$, only one critical size exists. The values corresponding to this critical size are approximately $a = b = 1.65$, which agrees with the critical scale $\exp(1/2) \approx 1.649$ that we found in Section 5.2. \square

Example 5.2

In Figure 5.3 we show the critical sizes for a rectangle with rounded corners. The results look similar to those of the ellipse; again there are always two critical sizes, except when the aspect ratio of the domain is equal to one. \square

Example 5.3

In the case of mixed boundary conditions we may rearrange terms in the linear system (5.40) and put all known coefficients at the right-hand side and all unknown coefficients at the left-hand side. Then we obtain a linear system of the form

$$\mathbf{Ax} = \mathbf{g}. \tag{5.42}$$

The matrix \mathbf{A} consists of a block from the matrix \mathbf{G} and a block from the matrix \mathbf{H} . We compute the condition number of the matrix \mathbf{A} for the case of an ellipsoidal boundary. This ellipse is approximated by 16 linear elements. At the first eight elements we impose Dirichlet boundary conditions and at the last eight elements we impose Neumann boundary conditions. Then we rescale the boundary by a factor a and compute the condition number for the new situation.

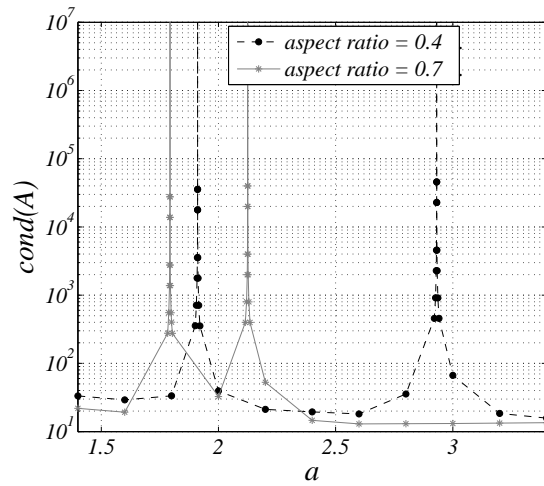


Figure 5.4: Condition number of \mathbf{A} for an ellipsoidal domain with aspect ratios 0.4 and 0.7 with mixed boundary conditions, as a function of scaling a .

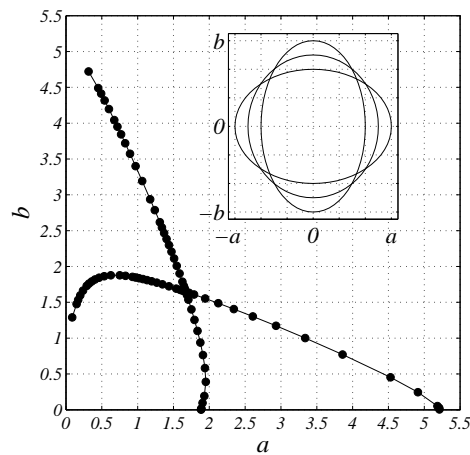


Figure 5.5: The critical sizes of an ellipse with mixed boundary conditions for which the condition number of \mathbf{A} is very large.

Figure 5.4 shows the condition number of the matrix for an ellipse with aspect ratio 0.4 and an ellipse with aspect ratio 0.7. We see that there exist two critical scalings for each ellipse. For these critical ellipses the condition number of \mathbf{A} becomes infinitely large.

The critical sizes of the ellipse in the case of mixed conditions are close to the

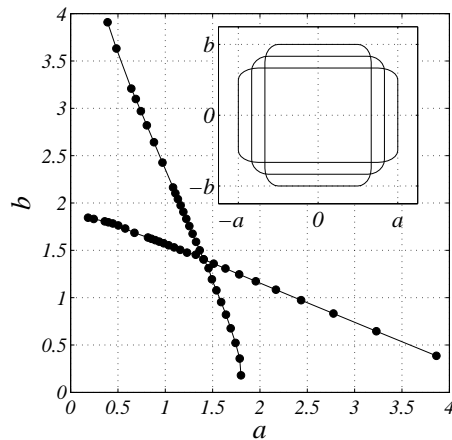


Figure 5.6: The critical sizes of rectangle with rounded corners with mixed boundary conditions for which the condition number of \mathbf{A} is very large.

critical sizes for the case of Dirichlet conditions. In Figure 5.5 we show all critical sizes of the ellipse for the case of mixed conditions. One can observe that this plot resembles the plot in Figure 5.2 for Dirichlet conditions. This confirms the idea that the BIE for the mixed case inherits the solvability problem from the BIE for the Dirichlet case. Note that the graph in Figure 5.6 is not symmetric anymore, as was the case for Dirichlet boundary conditions. This asymmetries are caused by the fact that the Dirichlet elements and Neumann elements are not distributed symmetrically over the boundary. \square

Example 5.4

In Figure 5.6 we show the critical sizes of a rectangular domain with rounded corners with mixed boundary conditions. The results look similar to those of rectangle with Dirichlet conditions (Figure 5.3). \square

Example 5.5

We now turn our attention to a time-dependent problem. We study a viscous drop of fluid of ellipsoidal shape that deforms to a circle due to surface tension. The evolution of the boundary of the drop is governed by the Stokes equations and can be solved using the BEM [78, 93]. The velocity \mathbf{v} of the boundary and the normal stress \mathbf{b} at the boundary are related by the BIE (5.9). The normal stress \mathbf{b} is proportional to the

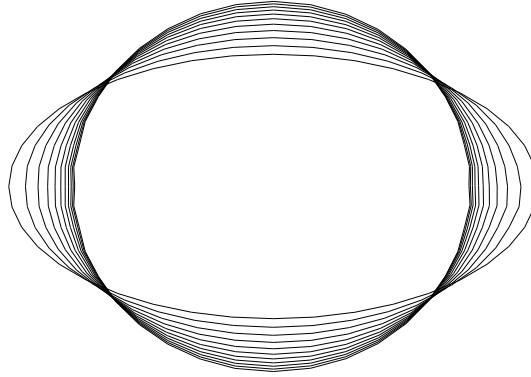


Figure 5.7: An ellipse with aspect ratio 0.5 deforming to a circle.

mean curvature κ of the boundary,

$$\mathbf{b} = -2\gamma\kappa\mathbf{n}, \quad (5.43)$$

with γ the surface tension and \mathbf{n} the outward normal at the boundary.

At time level $t = t_1$ we compute the velocity \mathbf{v} at all discretisation nodes \mathbf{x} . We fix a time step Δt and update the boundary of the drop with an Euler forward step,

$$\mathbf{x} \rightarrow \mathbf{x} + \Delta t\mathbf{v}(\mathbf{x}), \quad (5.44)$$

obtaining a new boundary. For this new boundary we again compute the velocity, and perform another Euler forward step. In this way we can study the shape evolution of the boundary. In Figure 5.7 we see the evolution of the ellipse to a circle. The initial shape is an ellipse with aspect ratio 0.5 and the longest semi-axis has a length 1.2. We choose 40 points to discretise the boundary, the size of the time step is $\Delta t = 0.375$ and we compute 10 time steps.

In the problem described above the normal stress is prescribed and the velocity has to be reconstructed with equation (5.9). We can also formulate a problem in which we try to reconstruct the normal stress given the boundary velocity. It is this problem that we study in this example. To solve this problem we need to invert the matrix \mathbf{G} at each time step. We know that there are certain critical boundaries for which the matrix \mathbf{G} will not be invertible. In the problem of the deforming ellipse we go through a whole range of ellipses with different shapes, and we risk to encounter one or more of those critical boundaries.

In Figure 5.8 we show the condition number of the matrix \mathbf{G} at each time step for a certain ellipse. An increase in the condition number indicates that the boundary

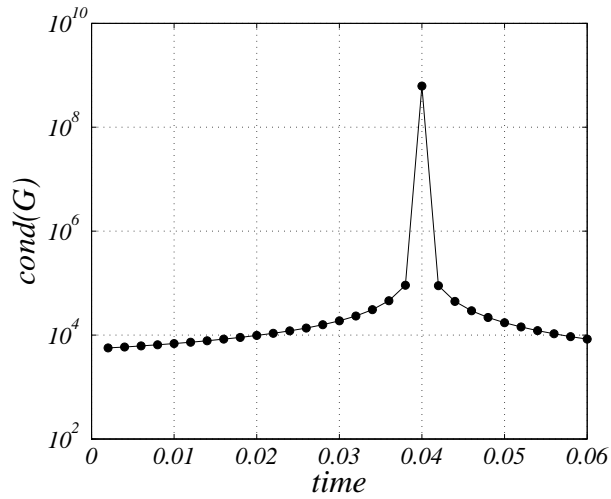


Figure 5.8: The condition number of G at each time step without scaling.

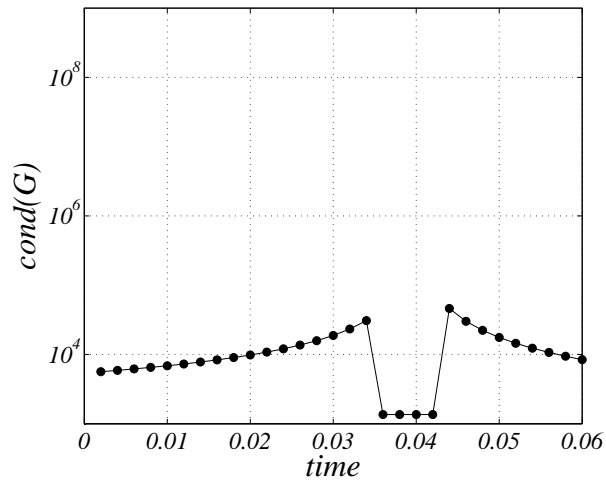


Figure 5.9: The condition number of G at each time step with temporary scaling when the domain is close to a critical domain.

under consideration is close to a critical boundary. We see that there is one time level in which such a critical boundary is reached. In this case we choose 20 discretisation points, a time step size $\Delta t = 0.002$ and 30 time steps.

There is a simple way to avoid a critical boundary. When the size of the boundary gets close to the critical boundary, we temporarily scale the domain. In Figure 5.9

	without scaling	with scaling	difference
area	10.59729837	10.5972852	$1.3150e - 05$
length	12.97535126	12.9751271	$2.2412e - 04$
a	2.759578290	2.75949155	$8.6740e - 05$
b	1.241866240	1.24190168	$-3.5440e - 05$

Table 5.2: *Difference in the BEM with and without temporarily rescaling. Listed are the surface area, the length of the boundary and the length of the semi-axes at time $t = 0.06$.*

we show the outcome of this strategy. At each time step between $t = 0.038$ and $t = 0.046$ we scale the domain by a factor 1.02. Then we solve the BEM problem for each of these time steps and the solutions are rescaled by a factor $1/1.02$. As a consequence the condition number of the BEM-matrices in these time steps does not become very large. The scaling of the domain during some time steps does hardly affect the outcome of the test. In Table 5.2 we give the surface area, the length of the boundary and the sizes of the semi-axes at the final time level $t = 0.06$. Their values hardly change when we perform temporary scaling.

Of course, for this strategy certain knowledge is needed about the critical boundaries, although we do not need to know the exact critical boundary. We only have to make sure we do not get too close to it. In the current test we only scaled at a restricted number of time steps. In general one could scale at every time step, thus excluding the possibility that a critical boundary is encountered.

The size of the initial ellipse strongly affects the condition number of the matrix at each time level as the ellipse deforms. In Figure 5.10 we show the size (a and b) of an ellipse while it deforms. The size of the initial ellipse is shown with a large dot, whereas the sizes it takes as it deforms to a circle are denoted by the trajectory starting in the dot. All ellipses deform to a circle, which is seen from the fact that all trajectories converge to the straight dashed line where $a = b$.

The dashed-dotted lines represent all critical ellipses, c.f. Figure 5.2. When the trajectory of an ellipse crosses one of the two dashed-dotted lines, it means that the ellipse at that time level is a critical ellipse. For this particular ellipse the condition number will be large. The trajectories of some ellipses never cross such a line of critical ellipses. This means that they never get a critical size as they deform to a circle. For other ellipses there is one point along their trajectory where the ellipse is a critical ellipse. It is also possible that an ellipse becomes critical twice during the deformation. An example is the ellipse with initial size $a = 4$ and $b = 0.8$. The trajectory of this ellipse crosses the line with critical ellipses twice. Figure 5.11 shows the condition number for this particular case. It can be seen that the condition number gets very large at the beginning and at the end of the time interval. \square

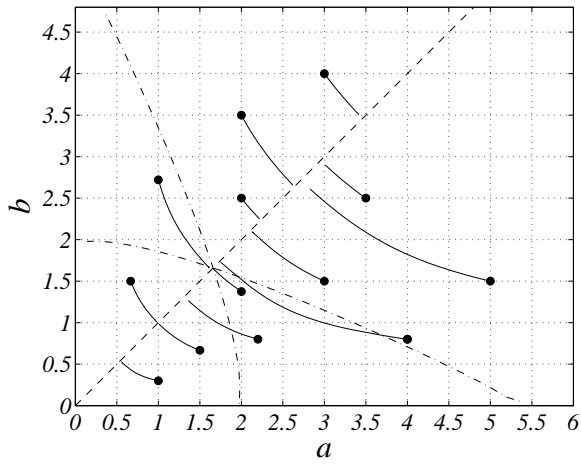


Figure 5.10: Each solid line represents the size of an ellipse as it deforms to a circle. The initial size, i.e. the lengths a and b of the semi-axes, is denoted with a large dot. The dashed-dotted lines represent the critical sizes. The dashed line represents all circles.

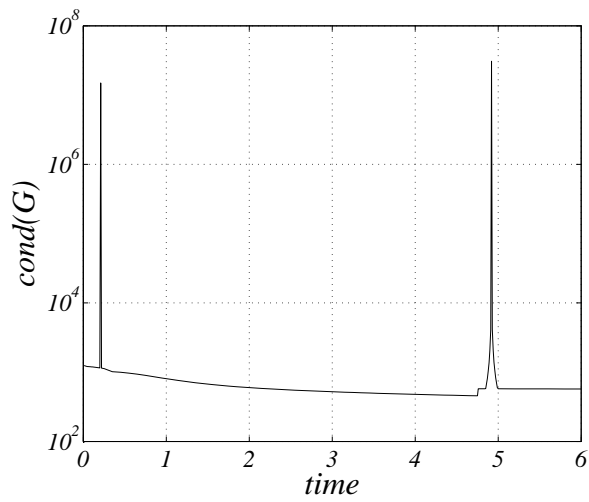


Figure 5.11: The condition number for the ellipse with initial values $a = 4$ and $b = 0.8$. During the deformation to a circle this particular ellipse gets a critical size twice.

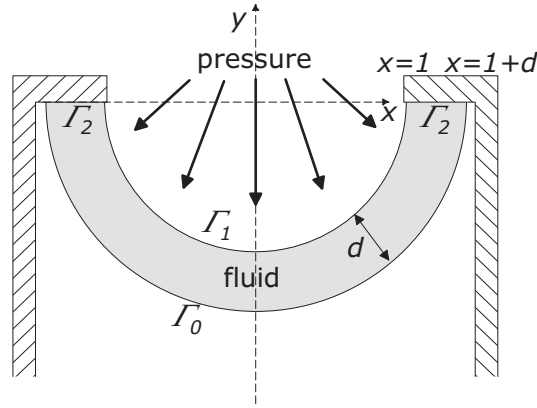


Figure 5.12: A viscous fluid is positioned into a mould. The boundary of the fluid consists of two semi-circles having radii 1 and $1 + d$.

5.6 Blowing problem in 2D

It is shown that the Helmholtz boundary integral equation for domains with thin parts causes numerical instabilities, especially as the thickness of the thin part approaches zero; this phenomenon is called *thin-shape breakdown* [32, 69]. In this section we investigate whether a similar phenomenon exists for the BIEs for the Stokes equations in 2D. In particular we want to know how the BEM-matrices are affected by a part of the domain whose thickness approaches zero.

To this end we study a special application of the Stokes equations, a problem that occurs in the blowing of glass bottles, as explained in Chapter 6. In this problem an amount of viscous fluid is positioned into a mould as depicted in Figure 5.12. At the top $y = 0$ (Γ_2) the fluid is fixed, while it is free to move everywhere else. At the boundary Γ_1 a pressure p_1 is applied, which causes the fluid to move downwards and sideways, into the mould. In the current section we investigate the blowing problem for an arbitrary viscous fluid. We do not address the time-dependent problem: for a certain initial shape we compute the velocity field of the fluid.

The flow of viscous fluids is described by the Stokes equations. The BEM can solve these equations and yields the velocity field at the boundary of the fluid. In this section we investigate whether the BEM-matrices that appear when solving the blowing problem become singular. We assume that the fluid is rotationally symmetric such that we can investigate the flow in a 2D setting. The initial boundary of the fluid

consists of two concentric semi-circles separated by a distance d , as can be seen in Figure 5.12. The inner circle has radius 1, so the outer circle has radius $1 + d$. The idea is that d will always be small compared to the radii of the circles, thus creating a thin structure.

The boundary value problem that has to be solved for the blowing problem is given by (for more details we refer to Chapter 6)

$$\begin{aligned}
\nabla^2 \mathbf{v} - \nabla p &= \mathbf{0}, \text{ at } \Omega, \\
\nabla \cdot \mathbf{v} &= 0, \text{ at } \Omega, \\
\sigma \mathbf{n} &= \mathbf{0}, \text{ at } \Gamma_0, \\
\sigma \mathbf{n} &= -p_1 \mathbf{n}, \text{ at } \Gamma_1, \\
\mathbf{v} &= \mathbf{0}, \text{ at } \Gamma_2.
\end{aligned} \tag{5.45}$$

Hence we prescribe the normal stress at the part of the boundary where the pressure is applied. At the other part of the free boundary we prescribe zero normal stress. The BEM transforms the boundary value problem into a linear system,

$$\left(\frac{1}{2}\mathbf{I} + \mathbf{H}\right)\mathbf{v} = \mathbf{G}\mathbf{b}, \tag{5.46}$$

where \mathbf{v} and \mathbf{b} are vectors with the coefficients of the velocity and the normal stress at each of the N nodes along the boundary. In this example we use linear elements; hence the velocity and normal stress vary linearly over each element. As the two boundary parts Γ_2 are very small, each part is modelled by one element only. Since the boundary of the domain mainly consists of two parts, i.e. the inner and the outer semi-circle, the matrix \mathbf{G} can be written as a block matrix,

$$\mathbf{G} = \left[\begin{array}{c|c} \mathbf{G}_1 & \mathbf{G}_2 \\ \hline \mathbf{G}_3 & \mathbf{G}_4 \end{array} \right]. \tag{5.47}$$

The block matrix \mathbf{G}_1 corresponds with integration over the inner semi-circle Γ_1 while all collocation nodes are at Γ_1 . Similarly \mathbf{G}_4 is related solely to the outer circle Γ_0 . Hence to compute these blocks we do not require any information from the opposite boundary part. The block matrices \mathbf{G}_2 and \mathbf{G}_3 correspond to integration over one semi-circle while the collocation nodes are at the opposite semi-circle. Let us assume that the number of boundary elements N is an even number.

If the unknowns in the vector \mathbf{b} are ordered as

$$\mathbf{b} = [b_x^1, \dots, b_x^{N/2}, b_y^1, \dots, b_y^{N/2}, b_x^N, \dots, b_x^{N/2+1}, b_y^N, \dots, b_y^{N/2+1}]^T, \tag{5.48}$$

d	$\ \mathbf{G} - \hat{\mathbf{G}}\ _\infty$	d	$\ \mathbf{G} - \hat{\mathbf{G}}\ _\infty$
0.1	$3.26 \cdot 10^{-2}$	0.0025	$1.72 \cdot 10^{-3}$
0.05	$2.06 \cdot 10^{-2}$	0.001	$7.57 \cdot 10^{-4}$
0.025	$1.23 \cdot 10^{-2}$	0.0005	$3.90 \cdot 10^{-4}$
0.01	$5.78 \cdot 10^{-3}$	0.00025	$1.99 \cdot 10^{-4}$
0.005	$3.18 \cdot 10^{-3}$	0.0001	$9.87 \cdot 10^{-5}$

Table 5.3: The matrix \mathbf{G} approaches the matrix $\hat{\mathbf{G}}$ as the thickness d goes to zero.

and the equations in the BEM are ordered in the following way,

$$\begin{array}{l}
 \text{equation at } \mathbf{x}_1 \\
 \vdots \\
 \text{equation at } \mathbf{x}_{N/2} \\
 \hline
 \text{equation at } \mathbf{x}_1 \\
 \vdots \\
 \text{equation at } \mathbf{x}_{N/2} \\
 \hline
 \text{equation at } \mathbf{x}_N \\
 \vdots \\
 \text{equation at } \mathbf{x}_{N/2+1} \\
 \hline
 \text{equation at } \mathbf{x}_N \\
 \vdots \\
 \text{equation at } \mathbf{x}_{N/2+1}
 \end{array}, \tag{5.49}$$

it can be shown that the matrix \mathbf{G} approaches the singular matrix $\hat{\mathbf{G}}$ in the limit $d \downarrow 0$,

$$\lim_{d \downarrow 0} \mathbf{G} = \hat{\mathbf{G}} := \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_1 \\ \mathbf{G}_1 & \mathbf{G}_1 \end{bmatrix}. \tag{5.50}$$

Table 5.3 shows the max-norm of $\mathbf{G} - \hat{\mathbf{G}}$ for decreasing values of d . We see that as $d \downarrow 0$ indeed $\mathbf{G} \rightarrow \hat{\mathbf{G}}$. Thus the matrix \mathbf{G} can be seen as a perturbation of the matrix $\hat{\mathbf{G}}$ around the point $d = 0$. Table 5.3 suggests that the difference between \mathbf{G} and $\hat{\mathbf{G}}$ depends linearly on d .

It is straightforward to see that any vector \mathbf{x} of the form $\mathbf{x} = [\mathbf{x}_1 \ -\mathbf{x}_1]^T$, $\mathbf{x}_1 \in \mathbb{R}^N$, is an eigenvector of $\hat{\mathbf{G}}$ with eigenvalue 0. Hence the eigenvalue 0 has an eigenspace with dimension (at least) N , and is spanned by

$$\{[1, 0, \dots, 0, -1, 0, \dots, 0]^T, \dots, [0, \dots, 0, 1, 0, \dots, 0, -1]^T\}. \tag{5.51}$$

m	$\ \mathbf{G} - \mathbf{G}_S\ _\infty$	m	$\ \mathbf{G} - \mathbf{G}_S\ _\infty$
4	$2.0 \cdot 10^{-3}$	20	$5.8 \cdot 10^{-5}$
8	$5.2 \cdot 10^{-4}$	24	$2.6 \cdot 10^{-5}$
12	$2.1 \cdot 10^{-4}$	28	$2.7 \cdot 10^{-5}$
16	$1.1 \cdot 10^{-4}$	32	$1.7 \cdot 10^{-5}$

Table 5.4: The difference between the matrix \mathbf{G} , computed with the help of a Gauss-Legendre scheme with m knots, and the matrix \mathbf{G}_S , computed with the help of an adaptive Simpson scheme.

Consider the matrix \mathbf{G} as a function of d , $\mathbf{G} = \mathbf{G}(d)$, and introduce a function $F : \mathbb{R}^{2N} \times \mathbb{R} \rightarrow \mathbb{R}^{2N}$ by

$$F(\mathbf{x}, d) := \mathbf{G}(d)\mathbf{x}. \quad (5.52)$$

If $\mathbf{G}(d)$ is computed exactly, there exist N roots of the equation $F = 0$, namely (\mathbf{x}^*, d^*) , where $\mathbf{x}^* = [\mathbf{x}_1 \ -\mathbf{x}_1]^T$, $\mathbf{x}_1 \in \mathbb{R}^N$, and $d^* = 0$. However in reality \mathbf{G} cannot be computed exactly since we make use of numerical integration schemes to approximate integrals, so in fact we deal with a matrix $\mathbf{G} + \mathbf{E}$, with $\|\mathbf{E}\|$ small. If the numerical integration is performed very accurately, the matrix \mathbf{E} approaches the zero matrix. For the true matrix $\mathbf{G} + \mathbf{E}$ we find $M \leq N$ of roots of the equation $F = 0$, namely (\mathbf{x}^*, d^*) , where $\mathbf{x}^* \approx [\mathbf{x}_1 \ -\mathbf{x}_1]^T$, $\mathbf{x}_1 \in \mathbb{R}^N$, and d^* small, but not necessarily 0.

The numerical integration error that is made by the Gauss-Legendre quadrature scheme that we use here is particularly large for the blowing problem that we study. To illustrate this we first compute the BEM-matrix \mathbf{G} by using an adaptive Simpson scheme with accuracy 10^{-16} . Adaptive schemes put more effort in approximating the integrals near singularities. These schemes keep increasing the number of knots near singularities, until the required accuracy is obtained. The Gauss-Legendre scheme treats each subinterval of the integration path equally, and does not verify whether a certain accuracy is obtained. Therefore the adaptive Simpson scheme approximates integrals with a logarithmic singularity much more accurate than the Gauss-Legendre scheme. We denote the matrix whose integrals are approximated with an adaptive Simpson scheme by \mathbf{G}_S and compare it to the matrices \mathbf{G} that are computed by using a Gauss-Legendre scheme with m knots. In Table 5.4 we show the max-norm of $\mathbf{G} - \mathbf{G}_S$. We see that the matrices \mathbf{G} approach \mathbf{G}_S very slowly. Even with as much as $m = 32$ knots the difference between \mathbf{G} and \mathbf{G}_S is still 10^{-5} . Hence the Gauss-Legendre scheme is much less accurate than the adaptive Simpson scheme. The cause of this large error lies in the fact that the two semi-circles of the boundary are

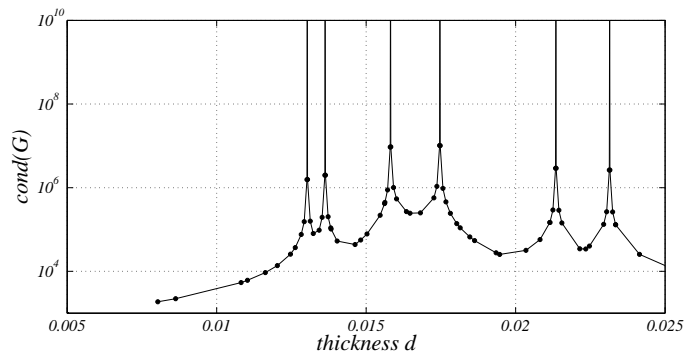
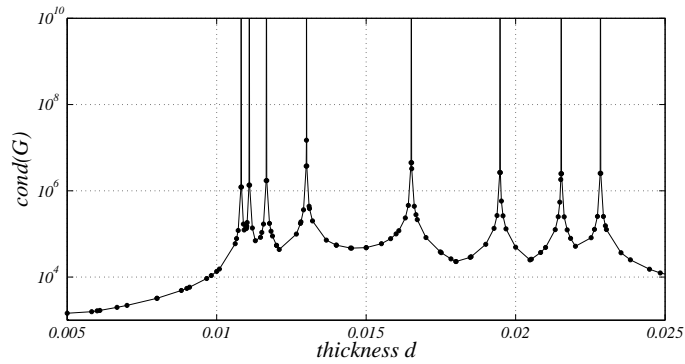
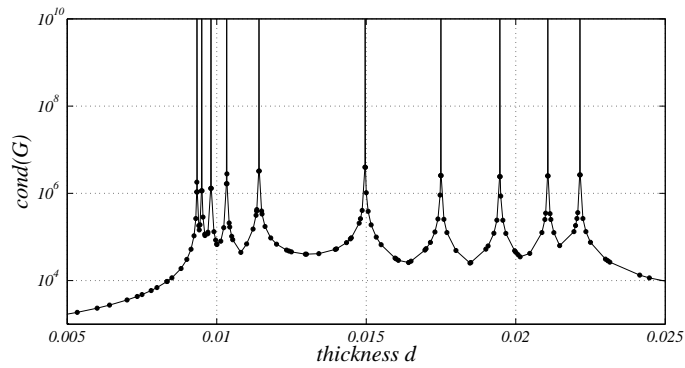
(a) $N = 18$ (b) $N = 20$ (c) $N = 22$

Figure 5.13: The condition number of the BEM-matrix \mathbf{G} as a function of the thickness d of the geometry.

separated by a small distance. In the computations of the BEM-matrix \mathbf{G} , integrals that involve the term $\log \|\mathbf{x} - \mathbf{y}\|$ have to be evaluated, where \mathbf{x} is a point at one semi-circle and \mathbf{y} a point at the opposite semi-circle. Hence when d approaches zero, the distance $\|\mathbf{x} - \mathbf{y}\|$ between \mathbf{x} and \mathbf{y} becomes very small, and a nearly logarithmic singularity appears in the integrals. The Gauss-Legendre scheme cannot approximate such integrals accurately enough [49, 84].

As a consequence, the matrix \mathbf{G} is not computed accurately, and the equation $F = 0$ has a number of roots (\mathbf{x}^*, d^*) in which the d^* are unequal to zero. To illustrate this we compute the BEM-matrix \mathbf{G} for several values of d and compute its condition number. In Figure 5.13(a) we show the condition number of \mathbf{G} as a function of d . For this example the boundary of the fluid is discretised with $N = 18$ boundary elements and the Gauss-Legendre scheme uses $m = 6$ knots. We observe that for 6 values of d the condition number jumps to infinity. These values of d correspond to the roots of $F = 0$.

If we increase the number of boundary elements N and compute the matrix elements of \mathbf{G} exactly, the number of roots of $F = 0$ also increases, as there are N such roots. If \mathbf{G} is not computed exactly, as is the case with the Gauss-Legendre scheme, we may also expect to find more roots of $F = 0$. Figure 5.13(b) and Figure 5.13(c) confirm this idea. For Figure 5.13(b) we choose $N = 20$ boundary elements and for Figure 5.13(c) we choose $N = 22$ boundary elements. For $N = 20$ we observe 8 values of d for which the condition number jumps to infinity, and for $N = 22$ even 10 of such values.

If we improve the accuracy of the numerical integration, we expect to approach the exact matrix \mathbf{G} better. As a consequence the roots of $F = 0$ should also approach the roots of the exact equation better. Hence the values of d^* should go to 0 if we improve on the numerical integration. Figure 5.14 confirms this idea. The matrices \mathbf{G} whose condition numbers are shown in Figure 5.14(a) are computed with a Gauss-Legendre scheme with $m = 6$ knots. For the matrices in Figure 5.14(b) and Figure 5.14(c) we use $m = 7$ and $m = 8$ knots respectively. Indeed we observe that the values of d for which the condition number jumps to infinity approach zero as m increases.

As we already noted before, the Gauss-Legendre schemes are not the most appropriate numerical integration schemes for this particular blowing problem. Adaptive schemes are better equipped for the blowing problem. In the next example we compute the BEM-matrix \mathbf{G} for several values of d using an adaptive Simpson scheme with accuracy 10^{-6} . With this accuracy the integrals in \mathbf{G} are approximated accurately enough compared to the Gauss-Legendre scheme, whose accuracy is only 10^{-5} for $m = 32$. The condition number of \mathbf{G} is plotted in Figure 5.15. We see that, except for $d \downarrow 0$, there is no other value of d for which the condition number of \mathbf{G} goes to infinity. The large condition number for $d \downarrow 0$ is expected, as \mathbf{G} approaches

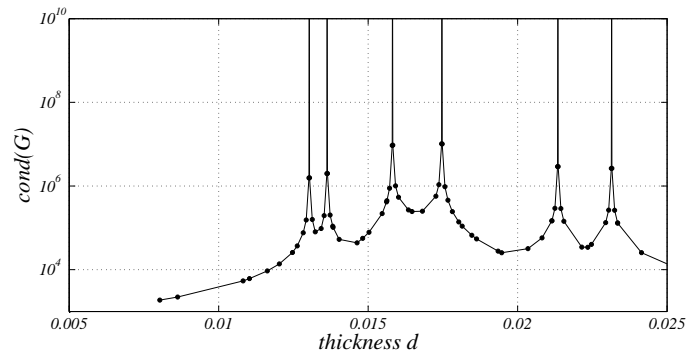
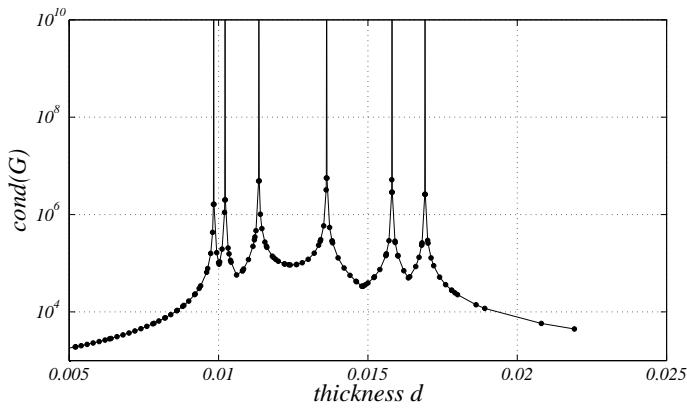
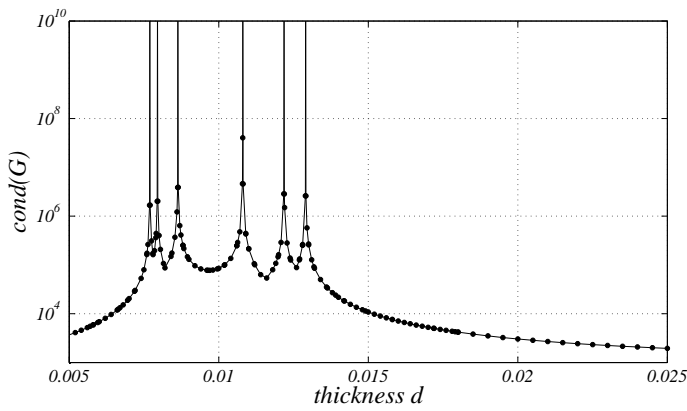
(a) $m = 6$ (b) $m = 7$ (c) $m = 8$

Figure 5.14: The condition number of the BEM-matrix \mathbf{G} as a function of the thickness d of the geometry. The number of boundary elements is $N = 18$.

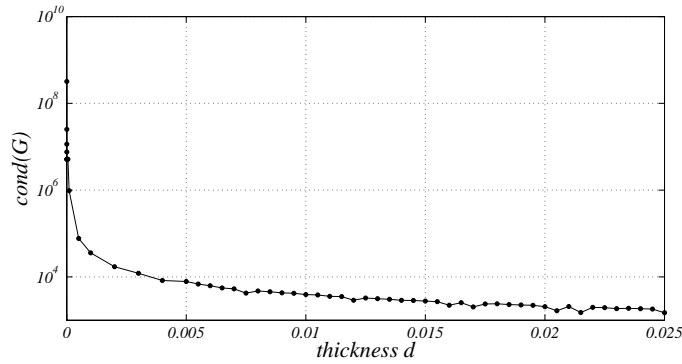


Figure 5.15: The condition number of the BEM-matrix \mathbf{G} as a function of thickness d . The number of boundary elements is $N = 18$. Numerical integration is performed with an adaptive Simpson scheme with accuracy 10^{-6} .

the singular matrix $\hat{\mathbf{G}}$ very well in these cases.

The blowing problem covered in this section shows that the numerical integration needs special attention in the case of thin domains, i.e. two boundary parts of the domain are separated by a small distance only. For such domains, in particular for the thin parts of such domains, the Gauss-Legendre quadrature scheme performs poorly when approximating the integrals that appear in the BEM-matrices. As a consequence, for some values of d the condition number of \mathbf{G} becomes infinitely large.

Chapter 6

Three-dimensional Stokes flow

In this chapter and the subsequent chapter we extend our research on the Stokes equations to three dimensions. We develop a mathematical model for a particular application: the blowing problem in 3D. We show that the BEM is a suitable numerical method to solve such problems. These chapters are an elaboration of earlier work of the author et al. [36, 40] concerning the numerical modelling of the blowing phase in the production of glass containers.

6.1 Simulating the blowing of glass containers with the BEM

In 3D the Stokes boundary integral equations cannot become singular at specific sizes or shapes of the domain, contrary to the Stokes boundary integral equations in 2D. The reason for this is the absence of a logarithmic term in the fundamental solution for the 3D Stokes equations. It is this logarithmic term that is the essence of the existence of critical domains in 2D. Hence we can safely construct a 3D BEM model for the class of blowing problems. In the blowing problem a viscous fluid is blown to a desired shape. We want to see whether the BEM is an appropriate numerical method to solve this blowing problem. As a special application we consider the blowing of glass containers.

The industrial production of glass containers like bottles and jars consists of several phases. First glass is molten in a furnace where the glass reaches temperatures between 1200 and 1600 °C. The molten glass is then cut into *gobs*, which are transported to a forming machine.

The gob falls into a mould, and a plunger is pushed into the mould, shaping the glass to an intermediate form called the *parison*. This phase of the production process

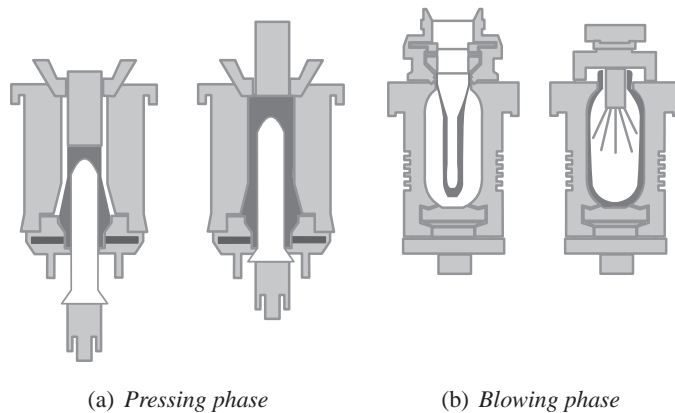


Figure 6.1: The production of glass containers consists of a pressing phase and a blowing phase.

is called the *pressing phase* (Figure 6.1(a)). The parison is put into a second mould in which it is allowed to creep in vertical direction (*sagging*) due to gravity for a short period of time. Then the parison is blown to its final shape by an inflow of pressurized air. This phase of the production process is called the *blowing phase* (Figure 6.1(b)).

For the glass industry it is important to optimize each phase of the production process. One can think of optimizing the shape of the parison, the speed of the plunger, the sagging time, the pressure of the air during the blowing phase, etc [68]. Experiments to tune these parameters are cumbersome, costly and time consuming. Therefore computer simulation of the various production phases can offer useful information to optimize the production.

In this thesis we study the flow of the glass during the blowing phase. We assume that the shape of the parison and the shape of the mould are given, and also the pressure of the inflowing air is prescribed. Given these settings the BEM computes the velocity at the boundary of the glass by solving the Stokes equations. Then we perform a time integration step to obtain the shape of the glass at the next time level. For this new shape we again compute the velocity at the boundary and perform a new time integration step. This iterative procedure enables us to study the shape evolution of the glass during the blowing phase. The computations are performed in three dimensions. This allows studying bottles and jars that are not rotationally symmetric, for instance due to small imperfections in the initial parison.

Numerical modelling of the production process of glass bottles and jars has been the topic of several papers. Mostly finite element methods are used to simulate the glass flow [12, 13], sometimes using a level set method to track the position of the

glass boundary [45]. In many cases rotationally symmetric parisons are modelled and computations are limited to two dimensions. To the author's knowledge our work is the first to address the blowing problem in three dimensions using the BEM.

During the blowing phase the temperature of the glass changes due to heat exchange with the mould. The viscosity of the glass depends on the temperature in an essentially non-linear fashion. As the viscosity appears in the Stokes equations, the heat and flow problem are coupled. However we show that during the blowing phase we may consider a homogeneous temperature. Consequently also the viscosity becomes homogeneous and the heat and flow problem can be addressed separately. In the papers mentioned above the heat problem is studied intensively. In this thesis we therefore focus on the flow problem.

In Figure 6.2 we schematically depict the glass and the mould. We assume that at the top (S_2) the glass touches the mould and cannot move. At the other parts where the glass touches the mould (S_3) it is allowed to slip along the mould. At the inlet of the mould (S_1) pressurized air flows in, while at the bottom of the mould (S_0) standard pressure is maintained.

Special attention has to be paid to the contact problem between the glass and the mould. Most papers assume a no-slip condition at the mould surface, i.e. glass cannot slip along the surface. In practice this is not the case. Sometimes the mould is even covered with a lubricating substance to improve the slip of the glass. Therefore we choose to work with a partial-slip boundary condition instead of a no-slip boundary condition.

The procedure described above results in a simulation tool that can be used to study the blowing phase for glass products. We have tested the simulation tool on several bottles and jars. The results of the tests are promising and may contribute to a better understanding of the production of bottles and jars.

6.2 Mathematical model

In this section we derive the mathematical model that describes the flow of a Newtonian fluid with high viscosity in three dimensions. Let the fluid domain be denoted by Ω , bounded by a closed surface S . The velocity and pressure of the fluid are denoted by \boldsymbol{v} and p respectively. Furthermore the fluid is characterized by the dynamic viscosity η , a surface tension γ and a typical length scale L . In general the viscosity η depends on the temperature of the glass. As the temperature may be space dependent, also the viscosity may be space dependent. However, later on we show that for the blowing problem the glass has a uniform temperature distribution, and henceforth the viscosity is uniform also.

The motion of the fluid is governed by two equations. The continuity equation expresses conservation of mass,

$$\nabla \cdot \mathbf{v} = 0, \quad (6.1)$$

where we assume that the density ρ of the fluid is constant and uniform, i.e. the fluid is incompressible. Vector fields that satisfy (6.1) are called *solenoidal*. The *Navier-Stokes equations* express conservation of momentum,

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\nabla \cdot \mathbf{v})\mathbf{v} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}, \quad (6.2)$$

where \mathbf{g} is a body force (here we consider only gravitational force $\mathbf{g} := -g\mathbf{e}_z$, with g the gravitational constant) and $\boldsymbol{\sigma}$ is the stress tensor. For the Newtonian fluid that we consider the following constitutive law for the stress tensor holds,

$$\sigma_{ij} := -p\delta_{ij} + 2\eta\mathcal{E}_{ij}. \quad (6.3)$$

Here δ_{ij} is the Kronecker delta and \mathcal{E} is the rate of deformation tensor, defined as

$$\mathcal{E}_{ij} := \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \quad (6.4)$$

With the constitutive law for $\boldsymbol{\sigma}$ the Navier-Stokes equations become

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\nabla \cdot \mathbf{v})\mathbf{v} = -\nabla p + \eta \nabla^2 \mathbf{v} + \rho \mathbf{g}. \quad (6.5)$$

We distinguish four types of boundary conditions at different parts of the surface, as can be seen in Figure 6.2. At the surfaces S_0 and S_1 the normal stress is related to the prescribed pressures p_0 and p_1 onto the surface and the surface tension γ ,

$$\begin{aligned} \boldsymbol{\sigma} \mathbf{n} &= -p_0 \mathbf{n} - \gamma \kappa \mathbf{n}, \text{ at } S_0, \\ \boldsymbol{\sigma} \mathbf{n} &= -p_1 \mathbf{n} - \gamma \kappa \mathbf{n}, \text{ at } S_1. \end{aligned} \quad (6.6)$$

The first term in the boundary condition accounts for the external pressure acting onto the surface. Here \mathbf{n} stands for the outward unit normal at the surface. The second term accounts for the surface tension due to the curvature of the surface. In the fluid all molecules attract one another. A molecule that is in the interior of the fluid domain is attracted by all its neighbours, so the average force it experiences is equal to zero. A molecule at the surface of the fluid has only neighbouring molecules at one side, and experiences a force into the fluid. For highly curved surfaces this force will be larger than for flat surfaces. The curvature of the surface is measured by the *mean curvature* κ with dimension L^{-1} . In Appendix A we present a way to approximate the curvature numerically.

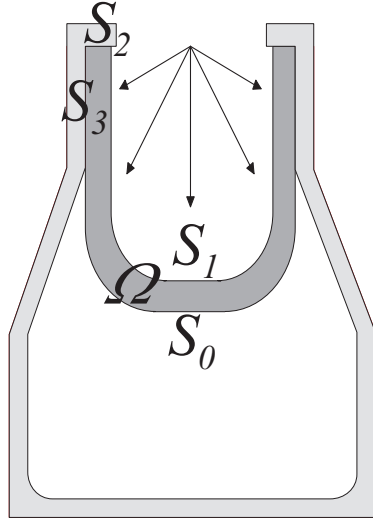


Figure 6.2: Cross-sectional view of the set-up of the blowing problem. The glass is positioned in a mould. The surface of the glass is divided into four parts.

At the surface S_2 we assume that the fluid is in contact with the mould. The fluid is not allowed to slip and hence we set the velocity equal to zero,

$$\mathbf{v} = \mathbf{0}, \text{ at } S_2. \quad (6.7)$$

At the surface S_3 the fluid is in contact with the mould, but is allowed to slip along the wall of the mould. This means that the velocity component in the normal direction to the wall is equal to zero, i.e. the fluid cannot penetrate the wall,

$$\mathbf{v} \cdot \mathbf{n} = 0, \text{ at } S_3. \quad (6.8)$$

The velocity components in the tangential directions may not be zero. Navier's slip law states that the tangential component of the velocity is related to the normal stress by [57, 67],

$$(\sigma \mathbf{n} + \beta_m \mathbf{v}) \cdot \mathbf{t} = 0, \text{ at } S_3. \quad (6.9)$$

Here \mathbf{t} is a vector in the tangential direction at the wall and β_m is a friction parameter. If $\beta_m \rightarrow 0$ there is no friction between fluid and wall. If $\beta_m \rightarrow \infty$ the friction between fluid and wall is very large and the fluid cannot flow along the wall. Condition (6.8) together with (6.9) yields the no-slip condition (6.7) when $\beta_m = \infty$.

We introduce a dimensionless pressure p' , velocity \mathbf{v}' and body force \mathbf{g}' by

$$p' := \frac{p - p_0}{p_1 - p_0}, \quad \mathbf{v}' := \frac{\mathbf{v}}{v_c} := \frac{\eta \mathbf{v}}{(p_1 - p_0)L}, \quad \mathbf{g}' := -\mathbf{e}_z. \quad (6.10)$$

We also define a characteristic time scale $t_c := L/v_c$. Using these characteristic variables and using L as a characteristic length, we rewrite the Navier-Stokes equations in dimensionless form as

$$\text{Re} \left(\frac{\partial \mathbf{v}'}{\partial t'} + (\nabla' \cdot \mathbf{v}') \mathbf{v}' \right) = -\nabla' p' + \nabla'^2 \mathbf{v}' + \frac{\text{Re}}{\text{Fr}} \mathbf{g}'. \quad (6.11)$$

Here the differential operator ∇' denotes differentiation with respect to the dimensionless spatial coordinates. The dimensionless numbers Re and Fr are the *Reynolds number* and the *Froude number* respectively, defined as

$$\begin{aligned} \text{Re} &:= \frac{\rho L v_c}{\eta}, \\ \text{Fr} &:= \frac{v_c^2}{gL}. \end{aligned} \quad (6.12)$$

For the type of fluids that we consider here, the Reynolds number is about 10^{-3} or smaller. For instance, for glass the Reynolds number can be computed with the values of the material properties and process parameters in Table 6.1, and we find $\text{Re} = 2.5 \cdot 10^{-4}$. This implies that the convective term at the left-hand side of (6.11) can be neglected. This reduces the Navier-Stokes equations to the *Stokes equations* with body force term,

$$\nabla'^2 \mathbf{v}' - \nabla' p' + \alpha \mathbf{g}' = 0, \quad (6.13)$$

where $\alpha := \text{Re}/\text{Fr}$. Together with the dimensionless form of (6.1)

$$\nabla' \cdot \mathbf{v}' = 0, \quad (6.14)$$

we obtain a system of four equations that describe the flow of the fluid. It can be verified that the dimensionless stress tensor σ' is given by

$$\sigma' := -\frac{p_0}{p_1 - p_0} \mathcal{I} + \frac{1}{p_1 - p_0} \sigma. \quad (6.15)$$

We also introduce a dimensionless curvature κ' by $\kappa' := L\kappa$. Substitution of σ' and κ' into the boundary conditions at S_0 and S_1 yields

$$\begin{aligned} \sigma' \mathbf{n} &= -\beta \kappa' \mathbf{n}, \text{ at } S_0, \\ \sigma' \mathbf{n} &= -(1 + \beta \kappa') \mathbf{n}, \text{ at } S_1, \end{aligned} \quad (6.16)$$

where the dimensionless number β is defined as

$$\beta := \frac{\gamma}{(p_1 - p_0)L}. \quad (6.17)$$

The boundary condition at S_2 becomes

$$\mathbf{v}' = \mathbf{0}, \text{ at } S_2. \quad (6.18)$$

It can be verified that substitution of σ' and \mathbf{v}' into the second part of the boundary condition at S_3 yields

$$-p_0 \mathbf{n} \cdot \mathbf{t} + (p_1 - p_0) \left(\sigma' \mathbf{n} + \frac{L\beta_m}{\eta} \mathbf{v}' \right) \cdot \mathbf{t} = 0. \quad (6.19)$$

The first term vanishes since $\mathbf{n} \cdot \mathbf{t} = 0$. We divide by $p_1 - p_0$ and obtain the following boundary conditions at S_3 ,

$$\begin{aligned} (\sigma' \mathbf{n} + \beta'_m \mathbf{v}') \cdot \mathbf{t} &= 0, \\ \mathbf{v}' \cdot \mathbf{n} &= 0, \text{ at } S_3, \end{aligned} \quad (6.20)$$

where the dimensionless number β'_m is defined as

$$\beta'_m := \frac{L\beta_m}{\eta}. \quad (6.21)$$

In the sequel we drop the $'$ to simplify notation.

We introduce a modified pressure \tilde{p} by [77, p. 164]

$$\tilde{p} := p + \alpha z, \quad (6.22)$$

where z is the vertical coordinate. Since $\nabla \tilde{p} = \nabla p + \alpha \mathbf{e}_z = \nabla p - \alpha \mathbf{g}$, the momentum balance simplifies to

$$\nabla^2 \mathbf{v} - \nabla \tilde{p} = 0, \text{ in } \Omega. \quad (6.23)$$

We may define a new stress tensor $\tilde{\sigma}$ by

$$\tilde{\sigma} = \tilde{\sigma}(\tilde{p}, \mathbf{v}) := -\alpha z \mathcal{I} + \sigma(p, \mathbf{v}). \quad (6.24)$$

Substitution of this new stress tensor into the boundary conditions at S_0 and S_1 yields

$$\begin{aligned} \tilde{\sigma} \mathbf{n} &= -(\alpha z + \beta \kappa) \mathbf{n}, \text{ at } S_0, \\ \tilde{\sigma} \mathbf{n} &= -(1 + \alpha z + \beta \kappa) \mathbf{n}, \text{ at } S_1. \end{aligned} \quad (6.25)$$

Thus we have moved the body force term from the Stokes equations to the boundary conditions. It can be verified that substitution of $\tilde{\sigma}$ into the second part of the boundary condition at S_3 yields

$$\alpha z \mathbf{n} \cdot \mathbf{t} + (\tilde{\sigma} \mathbf{n} + \beta_m \mathbf{v}) \cdot \mathbf{t} = 0. \quad (6.26)$$

The first term vanishes since $\mathbf{n} \cdot \mathbf{t} = 0$ and we obtain the following conditions at S_3 ,

$$\begin{aligned} (\tilde{\sigma}\mathbf{n} + \beta_m\mathbf{v}) \cdot \mathbf{t} &= 0, \\ \mathbf{v} \cdot \mathbf{n} &= 0, \text{ at } S_3, \end{aligned} \quad (6.27)$$

To summarize, the equations and boundary conditions in dimensionless form are given by

$$\begin{aligned} \nabla^2 \mathbf{v} - \nabla \tilde{p} &= 0, \text{ in } \Omega, \\ \nabla \cdot \mathbf{v} &= 0, \text{ in } \Omega, \\ \tilde{\sigma}\mathbf{n} &= -(\alpha z + \beta\kappa)\mathbf{n}, \text{ at } S_0, \\ \tilde{\sigma}\mathbf{n} &= -(1 + \alpha z + \beta\kappa)\mathbf{n}, \text{ at } S_1, \\ \mathbf{v} &= \mathbf{0}, \text{ at } S_2, \\ (\tilde{\sigma}\mathbf{n} + \beta_m\mathbf{v}) \cdot \mathbf{t} &= 0, \text{ at } S_3, \\ \mathbf{v} \cdot \mathbf{n} &= 0, \text{ at } S_3. \end{aligned} \quad (6.28)$$

In the sequel we will omit the $\tilde{\cdot}$.

This thesis does not aim addressing the heat change in the glass and the heat exchange between glass, air and mould. However, the temperature does enter the Stokes equations via the viscosity of the glass. In general the viscosity η depends on the temperature of the glass. Often in glass problems this dependence is modelled with the *Vogel-Fulcher-Tamman relation* [7], which is given by

$$\eta = \eta_0 \exp\left(\frac{B}{T - T_{VF}}\right). \quad (6.29)$$

Here T is the temperature of the glass and η_0 , B and T_{VF} are the so-called *Lakatos coefficients*.

We show that for the blowing problem it is not necessary to include the temperature dependence of the viscosity into our model. To see this consider the energy equation (still in dimension-full notation)

$$\begin{aligned} \rho c_p \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T &= k_c \nabla^2 T + \nabla \cdot (k_r(T) \nabla T) \\ &+ \eta ((\nabla \mathbf{v} + \nabla \mathbf{v}^T) : \nabla \mathbf{v}). \end{aligned} \quad (6.30)$$

The three terms on the right-hand side of this equation represent the conduction, the radiation and the viscous dissipation respectively. The parameters c_p , k_c and k_r are the specific heat, the conductivity and the Rosseland parameter respectively. We introduce a dimensionless temperature T' by $T = T_m + \Delta T T'$, with $\Delta T = T_g - T_m$, T_g being the initial temperature of the glass, and T_m being the temperature of the

parameter	value	dimension
L	0.01	m
v_c	0.01	m s^{-1}
ρ	2500	kg m^{-3}
η	1000	$\text{kg m}^{-1}\text{s}^{-1}$
c_p	1350	$\text{J kg}^{-1}\text{K}^{-1}$
k_c	1.5	$\text{W m}^{-1}\text{K}^{-1}$
T_g	1100	$^{\circ}\text{C}$
T_m	600	$^{\circ}\text{C}$
p_0	$1.00 \cdot 10^5$	$\text{kg m}^{-1}\text{s}^{-2}$
p_1	$1.38 \cdot 10^5$	$\text{kg m}^{-1}\text{s}^{-2}$
γ	0.3	kg s^{-2}

Table 6.1: Material properties and process parameters for the glass blowing problem.

mould. The dimensionless form of the heat equation reads (we immediately omit the ')

$$\begin{aligned} \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T &= \frac{1}{\text{Pe}} \nabla^2 T + \nabla \cdot \left(\frac{k_r(T)}{k_c} \frac{1}{\text{Pe}} \nabla T \right) \\ &+ \frac{\text{Ec}}{\text{Re}} ((\nabla \mathbf{v} + \nabla \mathbf{v}^T) : \nabla \mathbf{v}). \end{aligned} \quad (6.31)$$

Here the *Péclet number* and the *Eckert number* are defined as

$$\text{Pe} := \frac{\rho c_p v_c L}{k_c}, \quad \text{Ec} := \frac{v_c^2}{c_p \Delta T}. \quad (6.32)$$

Typical values of the parameters involved (see Table 6.1) yield $1/\text{Pe} \approx 10^{-3}$ and $\text{Ec}/\text{Re} \approx 10^{-7}$. Hence the terms on the right-hand side can be neglected and the energy equation reduces to

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = 0. \quad (6.33)$$

This implies that the temperature remains constant along streamlines. If we assume that the initial temperature of the glass is uniform in space and the mould has a uniform temperature too, it follows that the glass is isothermal. As a consequence, also the viscosity can be taken uniform. This shows that the Stokes equations and the heat equation are decoupled and can be solved separately. In this thesis, however, we are only interested in the flow problem.

6.3 Boundary integral equations

In this section the mathematical model summarized in (6.28) is transformed into a set of boundary integral equations [65]. We introduce a vector field $\mathbf{u}^k(\mathbf{x}, \mathbf{y})$ and a scalar function $q^k(\mathbf{x}, \mathbf{y})$ that satisfy the following Stokes equations,

$$\begin{aligned}\nabla_x^2 \mathbf{u}^k(\mathbf{x}, \mathbf{y}) - \nabla_x q^k(\mathbf{x}, \mathbf{y}) &= \delta(\mathbf{x} - \mathbf{y}) \mathbf{e}^k, \\ \nabla_x \cdot \mathbf{u}(\mathbf{x}, \mathbf{y}) &= 0.\end{aligned}\quad (6.34)$$

The vector \mathbf{e}^k is a unit vector directed along the k -th coordinate axis, $k = 1, 2, 3$. The subscript x means that differentiation is performed with respect to the spatial coordinate \mathbf{x} . The point \mathbf{y} is an arbitrary point in \mathbb{R}^3 . Physically the equations (6.34) describe the velocity and pressure due to a unit point source in the \mathbf{e}^k -direction located at the point \mathbf{y} . Using the requirements

$$u_j^k(\mathbf{x}, \mathbf{y}) = \mathcal{O}\left(\frac{1}{\|\mathbf{x} - \mathbf{y}\|}\right) \quad \text{and} \quad q^k(\mathbf{x}, \mathbf{y}) = o(1), \quad \text{for } \|\mathbf{x}\| \rightarrow \infty, \quad (6.35)$$

it can be seen that the solution of (6.34) is given by

$$\begin{aligned}u_j^k(\mathbf{x}, \mathbf{y}) &= -\frac{1}{8\pi} \left[\delta_{jk} \frac{1}{\|\mathbf{x} - \mathbf{y}\|} + \frac{(x_j - y_j)(x_k - y_k)}{\|\mathbf{x} - \mathbf{y}\|^3} \right], \\ q^k(\mathbf{x}, \mathbf{y}) &= -\frac{1}{4\pi} \frac{x_k - y_k}{\|\mathbf{x} - \mathbf{y}\|^3}.\end{aligned}\quad (6.36)$$

This solution is called the *fundamental singular solution* of the Stokes equations. The functions also satisfy the adjoint system,

$$\begin{aligned}\nabla_y^2 \mathbf{u}^k(\mathbf{x}, \mathbf{y}) + \nabla_y q^k(\mathbf{x}, \mathbf{y}) &= \delta(\mathbf{x} - \mathbf{y}) \mathbf{e}^k, \\ \nabla_y \cdot \mathbf{u}^k(\mathbf{x}, \mathbf{y}) &= 0.\end{aligned}\quad (6.37)$$

To proceed to a boundary integral formulation we require the *Green's identity for the Stokes equations*. Let \mathbf{u} and \mathbf{v} be two solenoidal vector fields, and p and q two sufficiently smooth scalar functions. Then the following integral identity holds [65]

$$\begin{aligned}\int_{\Omega} \left[\left(\nabla^2 v_i - \frac{\partial p}{\partial x_i} \right) u_i - \left(\nabla^2 u_i + \frac{\partial q}{\partial x_i} \right) v_i \right] d\Omega = \\ \int_S [\sigma_{ij}(p, \mathbf{v}) u_i n_j - \sigma_{ij}(-q, \mathbf{u}) v_i n_j] dS.\end{aligned}\quad (6.38)$$

Note that the stress tensor $\sigma(q, \mathbf{u})$ is equal to

$$\sigma_{ij}(q, \mathbf{u}) = -q\delta_{ij} + \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (6.39)$$

For the functions \mathbf{u} and q in the Green's identity we substitute the fundamental solutions $\mathbf{u}^k(\mathbf{x}, \mathbf{y})$ and $q^k(\mathbf{x}, \mathbf{y})$ and we consider these as functions of \mathbf{y} , thus satisfying the adjoint system (6.37). Furthermore we let \mathbf{v} and p be the solutions of the Stokes equations described in the previous section. Then we obtain the following integral identity,

$$v_k(\mathbf{x}) = \int_S \sigma_{ij}(-q^k, \mathbf{u}^k)_y v_i n_j dS_y - \int_S \sigma_{ij}(p, \mathbf{v}) u_i^k n_j dS_y, \quad \mathbf{x} \in \Omega, \quad (6.40)$$

for $k = 1, 2, 3$. We substitute the fundamental solution pair (q^k, \mathbf{u}^k) in the expression for σ , to obtain

$$\sigma_{ij}(-q^k, \mathbf{u}^k)_y = -\frac{3}{4\pi} \frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)}{\|\mathbf{x} - \mathbf{y}\|^4}. \quad (6.41)$$

We introduce a new variable \mathbf{b} ,

$$\mathbf{b} := \sigma(p, \mathbf{v})\mathbf{n}, \quad (6.42)$$

which represents the normal stress at the surface. Under the assumption that the surface of Ω is smooth, we let \mathbf{x} approach the surface. Then it can be deduced that

$$c_{ij}v_j(\mathbf{x}) + \int_S q_{ij}(\mathbf{x}, \mathbf{y})v_j(\mathbf{y})dS_y = \int_S u_{ij}(\mathbf{x}, \mathbf{y})b_j(\mathbf{y})dS_y, \quad i = 1, 2, 3. \quad (6.43)$$

Here the kernels q_{ij} and u_{ij} are defined as

$$\begin{aligned} q_{ij}(\mathbf{x}, \mathbf{y}) &:= \frac{3}{4\pi} \frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)n_k}{\|\mathbf{x} - \mathbf{y}\|^5} \\ u_{ij}(\mathbf{x}, \mathbf{y}) &:= \frac{1}{8\pi} \left[\delta_{ij} \frac{1}{\|\mathbf{x} - \mathbf{y}\|} + \frac{(x_i - y_i)(x_j - y_j)}{\|\mathbf{x} - \mathbf{y}\|^3} \right]. \end{aligned} \quad (6.44)$$

The coefficient c_{ij} depends on \mathbf{x} according to [65]

$$c_{ij} := \begin{cases} \delta_{ij} & \mathbf{x} \in \Omega, \\ \frac{1}{2}\delta_{ij} & \mathbf{x} \in \partial\Omega, \\ 0 & \text{elsewhere.} \end{cases} \quad (6.45)$$

From now on we choose $\mathbf{x} \in \partial\Omega$, which gives $c_{ij} = \frac{1}{2}\delta_{ij}$. We introduce the integral operators \mathcal{G} and \mathcal{H} ,

$$\begin{aligned} (\mathcal{G}\phi)_i &:= \int_S u_{ij}(\mathbf{x}, \mathbf{y})\phi_j(\mathbf{y})dS_y, \\ (\mathcal{H}\psi)_i &:= \int_S q_{ij}(\mathbf{x}, \mathbf{y})\psi_j(\mathbf{y})dS_y. \end{aligned} \quad (6.46)$$

These operators are called the *single and double layer operator* for the Stokes flow. With these operators the boundary integral equation (6.43) is written as steno

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{H}\right)\mathbf{v} = \mathcal{G}\mathbf{b}. \quad (6.47)$$

This boundary integral equation expresses the relation between the velocity \mathbf{v} at the surface of the fluid and the normal stress \mathbf{b} at the surface.

The double layer operator \mathcal{H} admits several eigenfunctions with eigenvalue zero. Hence the Neumann problem in which \mathbf{b} is prescribed and \mathbf{v} unknown at the whole boundary is not uniquely solvable. To overcome this non-uniqueness we follow the procedure of deflating the operator \mathcal{H} [94, p. 32]. The eigenfunctions of \mathcal{H} correspond to the six rigid body motions of Ω , including three translations and three rotations,

$$\begin{aligned} \varphi^m(\mathbf{x}) &:= \mathbf{e}^m, \quad m = 1, 2, 3, \\ \varphi^4(\mathbf{x}) &:= x_3\mathbf{e}^2 - x_2\mathbf{e}^3, \\ \varphi^5(\mathbf{x}) &:= -x_3\mathbf{e}^1 + x_1\mathbf{e}^3, \\ \varphi^6(\mathbf{x}) &:= x_2\mathbf{e}^1 - x_1\mathbf{e}^2. \end{aligned} \quad (6.48)$$

Here \mathbf{e}^m is the unit vector in the m -th direction. These rigid body motions are still in dimensionfull notation. We introduce the total surface area $|S|$, the center of mass \mathbf{x}^t and the moment of inertia I by

$$|S| := \int_S dS, \quad \mathbf{x}^t := \frac{1}{|S|} \int_S \mathbf{x} dS, \quad I := \int_S \|\mathbf{x} - \mathbf{x}^t\|^2 dS. \quad (6.49)$$

The dimensionless rigid body motions are given by

$$\begin{aligned} \hat{\varphi}^m &:= \frac{\varphi^m}{\sqrt{|S|}}, \quad m = 1, 2, 3, \\ \hat{\varphi}^4 &:= \frac{1}{\sqrt{I}} [(x_3 - x_3^t)\varphi^2 - (x_2 - x_2^t)\varphi^3], \\ \hat{\varphi}^5 &:= \frac{1}{\sqrt{I}} [-(x_3 - x_3^t)\varphi^1 + (x_1 - x_1^t)\varphi^3], \\ \hat{\varphi}^6 &:= \frac{1}{\sqrt{I}} [(x_2 - x_2^t)\varphi^1 - (x_1 - x_1^t)\varphi^2]. \end{aligned} \quad (6.50)$$

We define six projection operators \mathcal{P}^m by

$$\mathcal{P}^m := \hat{\varphi}^m(\cdot, \hat{\varphi}^m), \quad m = 1, \dots, 6. \quad (6.51)$$

We introduce the deflated form of the operator \mathcal{H} ,

$$\mathcal{H} + \mathcal{P} := \mathcal{H} + \sum_{m=1}^6 \mathcal{P}^m. \quad (6.52)$$

We replace the operator \mathcal{H} by the deflated operator $\mathcal{H} + \mathcal{P}$ in the boundary integral equation and obtain

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{H} + \mathcal{P}\right)\mathbf{v} = \mathcal{G}\mathbf{b}. \quad (6.53)$$

This boundary integral equation is uniquely solvable. The operator \mathcal{P} can also be written as

$$(\mathcal{P}\boldsymbol{\psi})_i = \int_S \varphi_{ij}(\mathbf{x}, \mathbf{y}) \psi_j(\mathbf{y}) dS_y, \quad (6.54)$$

where the kernel φ_{ij} is given by

$$\varphi_{ij} := \sum_{m=1}^6 \varphi_i^m(\mathbf{x}) \varphi_j^m(\mathbf{y}). \quad (6.55)$$

Note that deflating the operator \mathcal{H} is only necessary for pure Neumann problems, i.e. when \mathbf{b} is given at the whole boundary. For many applications in this thesis we have problems with mixed boundary conditions and we do not need to deflate the operator.

6.4 Numerical solution

In this section we transform the boundary integral equations (6.53) to a linear system of algebraic equations. The surface S is approximated by K linear triangular elements. Each element consists of three nodes $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$, which are located at the corners of the triangle. The total number of nodes is denoted by N . We introduce three linear shape functions,

$$\begin{aligned} \phi_1(\xi_1, \xi_2) &:= 1 - \xi_1 - \xi_2, \\ \phi_2(\xi_1, \xi_2) &:= \xi_1, \\ \phi_3(\xi_1, \xi_2) &:= \xi_2, \end{aligned} \quad (6.56)$$

where $0 \leq \xi_1, \xi_2 \leq 1$ and $\xi_1 + \xi_2 \leq 1$. Consider the k -th element S_k with nodes $\mathbf{x}^1, \mathbf{x}^2$ and \mathbf{x}^3 . The element S_k is parameterized by

$$\mathbf{y} = \mathbf{y}(\xi_1, \xi_2) = \phi_1 \mathbf{x}^1 + \phi_2 \mathbf{x}^2 + \phi_3 \mathbf{x}^3. \quad (6.57)$$

At each element S_k the functions \mathbf{v} and \mathbf{b} are linearly approximated with the same shape functions,

$$\begin{aligned} \mathbf{v}(\mathbf{y}) &= \phi_1 \mathbf{v}^1 + \phi_2 \mathbf{v}^2 + \phi_3 \mathbf{v}^3, \\ \mathbf{b}(\mathbf{y}) &= \phi_1 \mathbf{b}^1 + \phi_2 \mathbf{b}^2 + \phi_3 \mathbf{b}^3. \end{aligned} \quad (6.58)$$

Here $\mathbf{v}^s := \mathbf{v}(\mathbf{x}^s)$ is the velocity at the node \mathbf{x}^s and $\mathbf{b}^s := \mathbf{b}(\mathbf{x}^s)$ is the normal stress at the node \mathbf{x}^s . We approximate the surface integrals over S in (6.53) by a sum of integrals over the elements S_k , and substitute the approximations for \mathbf{v} and \mathbf{b} , yielding

$$\begin{aligned} \frac{1}{2}v_i(\mathbf{x}) &+ \sum_{k=1}^K \int_{S_k} q_{ij}(\mathbf{x}, \mathbf{y}) (\phi_1 v_j^1 + \phi_2 v_j^2 + \phi_3 v_j^3) dS_y \\ &+ \sum_{k=1}^K \int_{S_k} \varphi_{ij}(\mathbf{x}, \mathbf{y}) (\phi_1 v_j^1 + \phi_2 v_j^2 + \phi_3 v_j^3) dS_y \\ &= \sum_{k=1}^K \int_{S_k} u_{ij}(\mathbf{x}, \mathbf{y}) (\phi_1 b_j^1 + \phi_2 b_j^2 + \phi_3 b_j^3) dS_y, \quad \mathbf{x} \in S, \end{aligned} \quad (6.59)$$

for $i = 1, 2, 3$. In these three equations there are $3N$ velocity coefficients and $3N$ normal stress coefficients. At each node either the velocity or the normal stress is prescribed, leaving $3N$ unknown coefficients. Hence we need $3N$ equations to calculate these remaining $3N$ coefficients. By substituting $\mathbf{x} = \mathbf{x}^p$ in (6.59), $p = 1, \dots, N$, we obtain the necessary $3N$ equations.

Next we construct two coefficient vectors,

$$\begin{aligned} \mathbf{v} &:= [v_1^1, v_2^1, v_3^1, \dots, v_1^N, v_2^N, v_3^N]^T, \\ \mathbf{b} &:= [b_1^1, b_2^1, b_3^1, \dots, b_1^N, b_2^N, b_3^N]^T. \end{aligned} \quad (6.60)$$

This allows us to write (6.59) in a matrix-vector form,

$$(\mathbf{H} + \mathbf{\Phi})\mathbf{v} = \mathbf{G}\mathbf{b}. \quad (6.61)$$

To compute the matrices \mathbf{H} , $\mathbf{\Phi}$ and \mathbf{G} , we have to evaluate integrals of the form

$$\int_{S_k} q_{ij}(\mathbf{x}^p, \mathbf{y}) \phi_r dS_y, \quad \int_{S_k} \varphi_{ij}(\mathbf{x}^p, \mathbf{y}) \phi_r dS_y, \quad \int_{S_k} u_{ij}(\mathbf{x}^p, \mathbf{y}) \phi_r dS_y. \quad (6.62)$$

The integrals can be evaluated by using a Gaussian quadrature scheme, but special care has to be taken of the third type. When the node \mathbf{x}^p is in the surface element S_k the integrand is singular. In this case we perform an adaptive Gauss quadrature scheme to approximate the integral. The first integral also becomes singular when \mathbf{x}^p is in the surface element S_k . However in this case the integral can be calculated analytically and it can be shown that its contribution yields zero.

In the case of mixed boundary conditions, we either know the velocity coefficients at a node or the normal stress coefficients. Hence in (6.61) some of the unknowns are in the vector \mathbf{b} at the right-hand side and some of the knowns are

in the vector \mathbf{v} at the left-hand side. By moving the known coefficients to the right and the unknown coefficients to the left we arrive at the standard form linear system

$$\mathbf{A}\mathbf{x} = \mathbf{f}. \quad (6.63)$$

Here \mathbf{x} contains all unknown coefficients and \mathbf{f} is a vector with the prescribed boundary data.

When $S_3 \neq \{\emptyset\}$, i.e. a slip condition at the wall of the mould, there are nodes at which both the velocity and normal stress coefficients are unknown, though related via the slip conditions (6.27). Let \mathbf{t}^r , $r = 1, 2$, be the two tangential vectors at the wall at such a node $\mathbf{x} \in S_3$. Since $\mathbf{v} \cdot \mathbf{n} = 0$ at \mathbf{x} and $\{\mathbf{n}, \mathbf{t}^1, \mathbf{t}^2\}$ forms a local basis of \mathbb{R}^3 , we may write

$$\mathbf{v}(\mathbf{x}) = a_1 \mathbf{t}^1(\mathbf{x}) + a_2 \mathbf{t}^2(\mathbf{x}), \quad a_1, a_2 \in \mathbb{R}. \quad (6.64)$$

Substitution into $(\mathbf{b} + \beta_m \mathbf{v}) \cdot \mathbf{t}^r = 0$ yields $a_r = -(\mathbf{b} \cdot \mathbf{t}^r) / \beta_m$.

In this way we express \mathbf{v} in terms of \mathbf{b} . In the boundary integral equations we can replace \mathbf{v} by this expression and thus eliminate \mathbf{v} . We then solve the linear system, which is again of the form $\mathbf{A}\mathbf{x} = \mathbf{f}$. The solution yields \mathbf{b} at \mathbf{x} and we reconstruct \mathbf{v} at \mathbf{x} with the expression (6.64).

The matrix \mathbf{A} is a dense matrix and the linear system can be solved by using an LU-decomposition technique. Due to the dense nature of the matrix, this may become costly, especially when the size of the matrix is large. However the BEM reduces the dimension of the problem by one, as it involves variables at the surface only. Hence compared to finite element methods or finite volume methods, the number of unknowns is relatively low. Henceforth the matrix \mathbf{A} is also not as large as in other numerical methods. In our numerical tests the number of nodes N ranges from 750 to 1500. This leads to matrices of at most 4500×4500 entries. It turns out that computing these entries consumes most of the computation time, approximately 90%. Solving the linear system consumes only 5% of the time.

6.5 Time integration and post-processing

The movement of the boundary surface of the fluid domain is described by the velocity field $\mathbf{v}(\mathbf{x}, t)$ that is the outcome of the Stokes equations in Section 6.4. In fact we calculate the velocity at a set of N nodes at the boundary. To study the evolution of the boundary we need to solve an ordinary differential equation,

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{v}(\mathbf{x}, t). \quad (6.65)$$

At time $t = t^{\{n\}}$ we know the locations of the nodes $\mathbf{x}^{\{n\}}$ and the velocity at these nodes $\mathbf{v}(\mathbf{x}^{\{n\}}, t^{\{n\}}) =: \mathbf{v}^{\{n\}}$. We do not have any information of the nodes or

velocity in the future. Therefore we cannot make use of implicate time integration schemes to solve (6.65).

An option is to use an *Euler forward* scheme, in which we approximate the locations of the nodes at the next time level $t^{\{n+1\}}$ by

$$\mathbf{x}^{\{n+1\}} = \mathbf{x}^{\{n\}} + \Delta t \mathbf{v}^{\{n\}}, \quad (6.66)$$

where $\Delta t = t^{\{n+1\}} - t^{\{n\}}$. However this scheme is only first order accurate. Another option is to use a modified version of *Heun's method*, which is also called the *improved Euler method*. In this method the location of the node at time level $t^{\{n+1\}}$ is approximated by

$$\mathbf{x}^{\{n+1\}} = \mathbf{x}^{\{n\}} + \frac{1}{2} \Delta t \left[\mathbf{v}^{\{n\}} + \mathbf{v}(\mathbf{x}^{\{n\}} + \Delta t \mathbf{v}^{\{n\}}) \right]. \quad (6.67)$$

This method is known to be second order accurate [11]. However for this method we need the velocity \mathbf{v} at the next time level $t^{\{n+1\}}$ in the new location $\mathbf{x}^{\{n+1\}}$ of the node. As we remarked before we do not have information of future time levels. To get around this problem we first predict the location of the node at the next time level using an Euler forward step (6.66). For this predicted node $\mathbf{x}^{\{n+1\}}$ we again solve the Stokes equations and we obtain the velocity in this node at time $t^{\{n+1\}}$. Then we use Heun's method (6.67) to correct our prediction of $\mathbf{x}^{\{n+1\}}$.

In the procedure to numerically solve the Stokes equations as described in this chapter, two types of errors are made. First we make a discretisation error with the BEM. As we use linear elements this error is $\mathcal{O}(h^2)$, where h is the (maximal) element size. Second we make an error in the time integration, which is $\mathcal{O}(\Delta t)$ for Euler forward, and $\mathcal{O}(\Delta t^2)$ for Heun's method. We have to realize that to decrease the total error we have to decrease both the discretisation error and the time integration error. Therefore it does not help to use high order time integration schemes when the total error is dominated by the discretisation error. In our numerical tests the latter is the case. The discretisation error is larger than the time integration error and in that view it suffices to use Euler forward for time integration.

After we obtain the solution of the Stokes equations with the BEM there are several ways to improve the quality of the solution. First we perform velocity smoothing on the velocity field that is the outcome of the BEM. This smoothing step takes away irregularities in the velocity field that are not physical. The amount of velocity smoothing must be moderate, as it directly affects the solution of the Stokes equations. In Appendix C we give a more detailed description of the velocity smoothing process.

Another type of smoothing is Laplacian smoothing, which affects the discretisation of the fluid surface. Again it takes away irregularities, yielding a smoother surface. Appendix C introduces the Laplacian smoothing technique.

When we study the blowing problem for glass, the flow is restricted to the interior of the mould. In all our numerical examples this mould is given by a parametric representation. In Appendix B we develop a strategy to make sure that the fluid remains inside the mould, i.e. nodes of the discretised fluid surface cannot penetrate the wall of the mould. The basic idea is that, in first instance, these nodes are allowed to cross the wall. At the end of each time step we verify if any node has moved through the wall. If so, we determine a new location for such a node, and also account for the volume change caused by this relocation.

As the fluid surface expands in time, the triangular elements that constitute the discretised surface increase in size. This may lead to a very irregular and coarse surface discretisation. To avoid this we measure the lengths of the edges of the elements. If such an edge becomes larger than a certain tolerance value, the edge is subdivided into two new edges. A new node is introduced at the center of the subdivided edge, and the two elements sharing the edge are subdivided into four smaller elements. In this way the number of nodes and boundary elements increase in time, but it ensures the surface to remain smooth and regular.

Chapter 7

Results

This chapter shows numerical results from the mathematical model that was developed in the previous chapter. In Section 7.1 we present several numerical examples for the glass blowing problem. Section 7.2 shows examples for another type of problem, the evolution of viscous drops of fluid due to surface tension. In both sections all computations are performed in three dimensions, without making any assumptions on symmetry of the domains. We conclude this chapter with a parameter analysis in Section 7.3.

7.1 Glass blowing

The set-up of the blowing process of glass bottles and jars is schematically depicted in Figure 7.1. A preform of hot liquid glass, often called a *parison*, is positioned into a mould. Pressurized air flows into the mould causing the glass to move deeper into the mould. Eventually the whole mould is covered by a thin layer of glass. We simulate this blowing process with the BEM model that we developed in the previous chapter. We take several shapes for the mould and the parison and study the flow of the glass in time. In all simulations we assume that the top of the glass (S_2) is fixed to the mould and cannot move.

The material properties of glass and the process parameters of the blowing problem can be found in Table 6.1. With these properties the dimensionless numbers that appear in the model have the following values,

$$\alpha = 0.006, \quad \beta = 0.001. \quad (7.1)$$

This implies that the effect of gravity and surface tension is small compared to the effect of the inflow of pressurized air. Ideally the value of the dimensionless friction coefficient β_m is determined experimentally. Unfortunately these experiments are

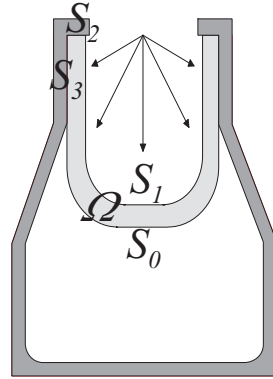


Figure 7.1: The parison is suspended in the mould, attached at S_2 . Pressurized air flows into the mould from above.

very complex to perform and we are not aware of any such experiments reported in literature. Therefore we perform a number of numerical tests to determine realistic values of β_m , see Example 7.6.

Example 7.1

The first simulation concerns a parison without mould. To keep a reference point, we still assume that the parison is fixed at the top. As there is no mould the glass is free to move in all directions. In Figure 7.2 we show six snapshots of the parison as it expands and Figure 7.3 shows a cross-sectional view at $y = 0$. The snapshots should be viewed from left to right, from top to bottom. The bulk of the glass is at the bottom of the parison, while the thinnest parts of the parison are at the sides. It is at these parts where the fluid flow is strong, while the bottom of the parison remains almost unchanged. Although the initial parison is rotationally symmetric, small a-symmetries appear during the simulation. This is a consequence of the fact that the surface mesh is not rotationally symmetric. If we decrease the mesh size the parison will become almost rotationally symmetric.

In principle this simulation can be continued for a long time. The glass will expand further, but numerical problems start to arise. First the layer of glass becomes very thin and special care has to be taken to ensure that the inner and outer surface remain separated by a small distance. Another reason to stop this particular simulation after some time is the increasing number of boundary elements. To ensure smooth surfaces the number of nodes and elements has to be increased significantly for large expansions of the glass. \square

Example 7.2

The next simulation shown in Figure 7.4 and 7.5 concerns a parison that is suspended in a mould. The air that flows in from above causes the glass to take the shape of the mould. When the glass touches the wall of the mould, it remains connected to the mould, though it is allowed to slip along the mould. In this example we take $\beta_m = 1$. The mould has a cylindrical shape with rounded corners. The reason to choose for rounded corners is twofold. First, in practice the corners of a bottle or jar are never 90-degree corners. Second, it turns out that it is very hard to make the glass fill the whole mould if the corners are straight. Even with rounded corners we still need to refine the mesh sufficiently to fill the corners with glass.

The cross-sectional view in Figure 7.5 shows that we get sharp corners at the top of the parison where the surface parts S_1 and S_2 touch. This is a direct consequence of the choice to keep the glass fixed at S_2 , while it is allowed to move at S_1 . In reality these sharp corners do not appear. \square

Example 7.3

Figure 7.6 and 7.7 show a simulation with a slightly more advanced mould. The lower part of the mould has a smaller width than the upper part of the mould. Again the corners are rounded and the glass is allowed to slip along the wall of the mould. In the previous two simulations the only driving force was the pressure of the air that is blown into the mould from above. In reality, in first instance the glass is subjected to gravity only. The glass will sag to the bottom of the mould and when it almost touches the bottom, air starts to blow into the mould. The simulation in Figure 7.6 and 7.7 distinguishes these two stages. The first three snapshots correspond to the sagging stage while the last three snapshots correspond to the blowing stage. \square

Example 7.4

The simulation that is presented in Figure 7.8 and 7.9 involves a parison and a mould that are not rotationally symmetric. We opt for square shapes with rounded corners. Hence the parison and mould are symmetric in the planes $x = 0$ and $y = 0$. However these symmetries are not exploited in the computations. Again we consider the two stages that occur in the production process: sagging and blowing. The first three snapshots correspond to the sagging stage while the last three snapshots correspond to the blowing phase. In this case it is clearly visible that the glass moves only in vertical direction during the sagging stage. In the blowing stage the glass also moves in radial direction. \square

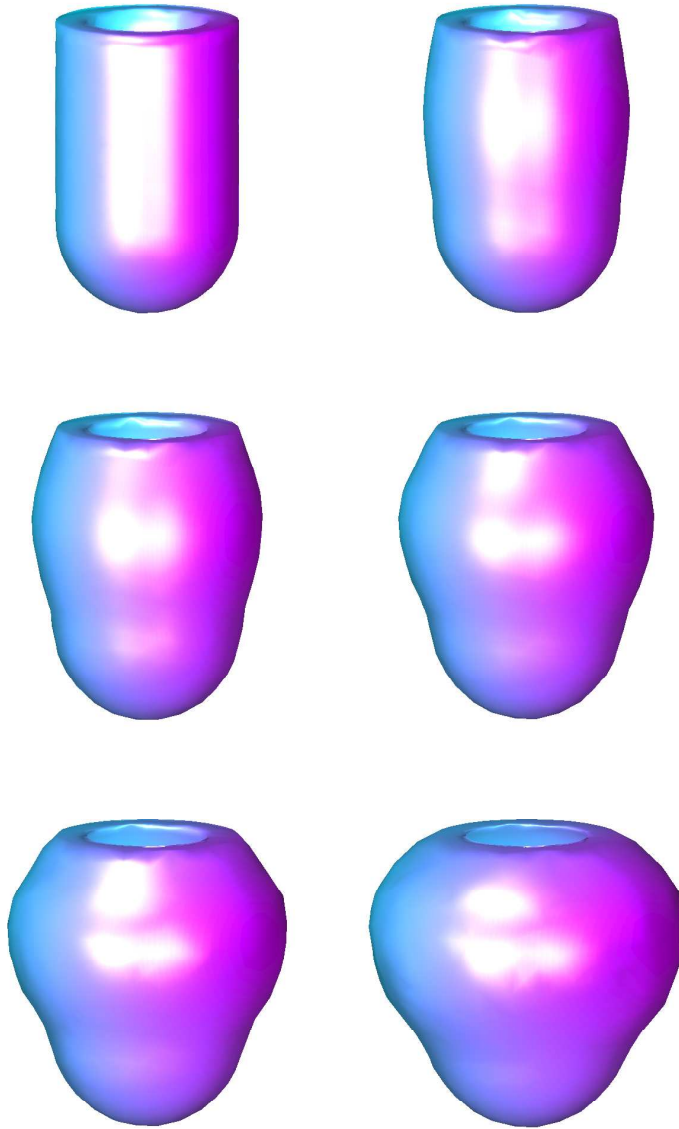


Figure 7.2: 3D Snapshots of the glass as it expands due to the air blowing in from above. For this simulation no mould is present, so the glass can expand freely in all directions. The only restriction is that the glass is fixed at the top of the parison.

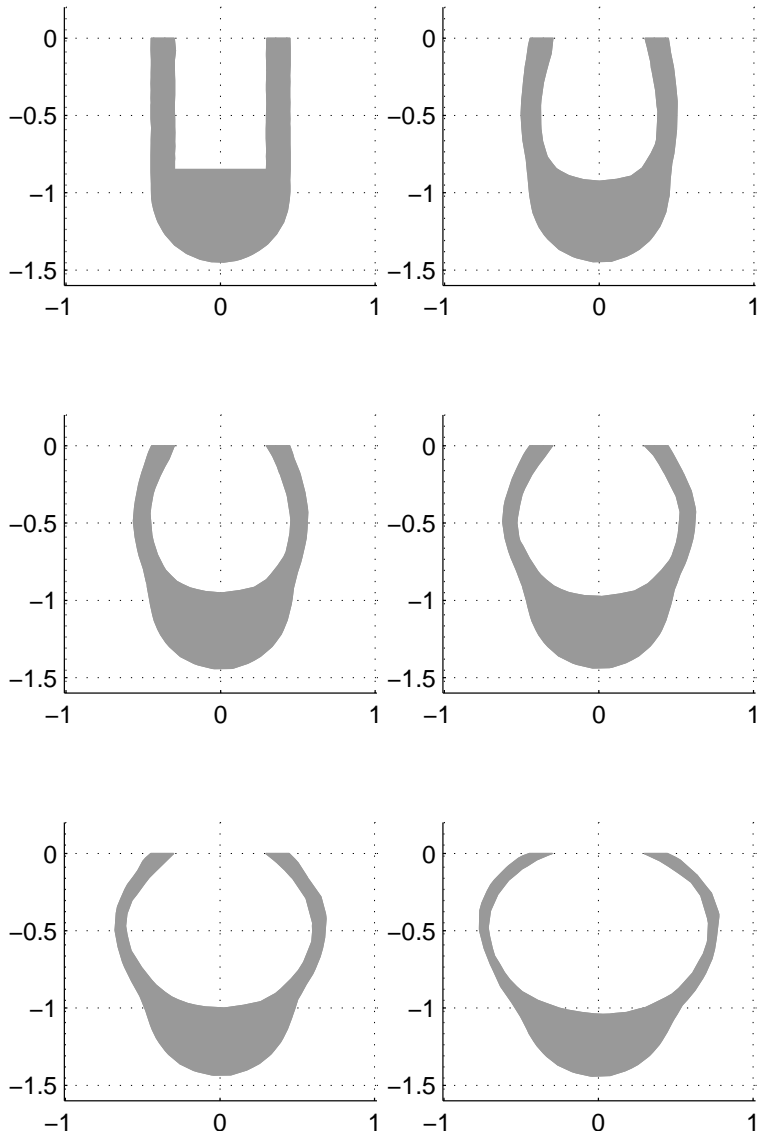


Figure 7.3: Cross-sectional view of Figure 7.2 at $y = 0$.

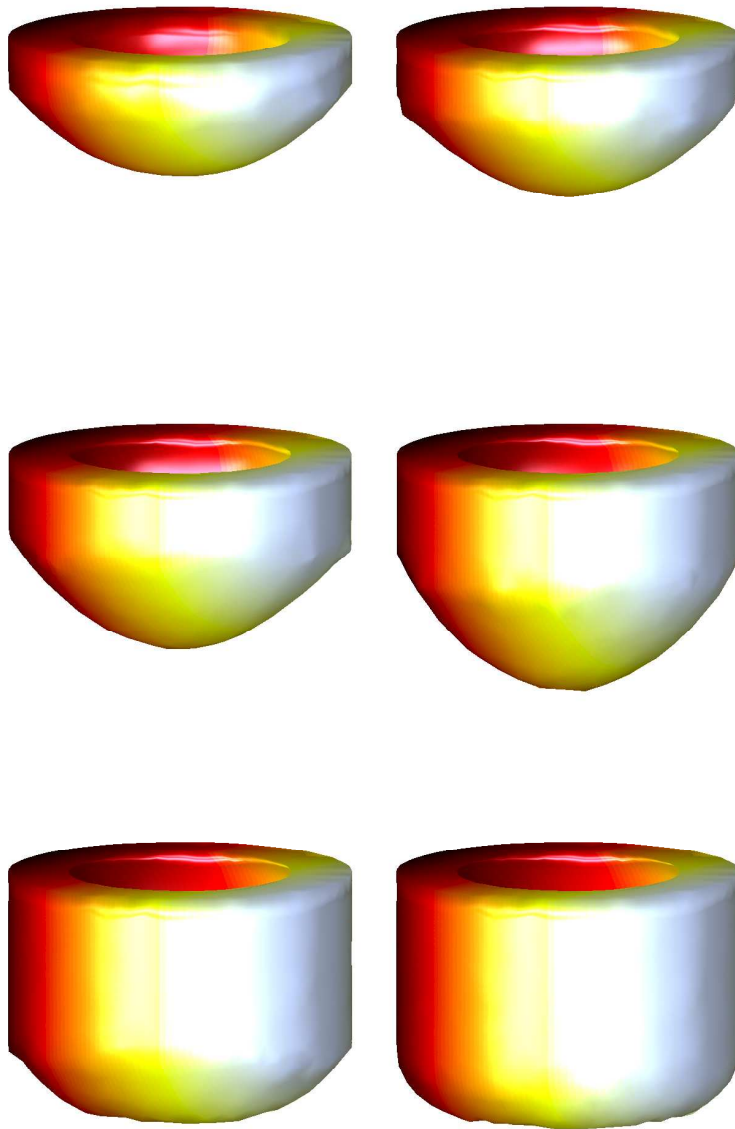


Figure 7.4: 3D Snapshots of the glass as it expands due to the air blowing in from above. The mould has a cylindrical shape with rounded corners. The glass is allowed to slip along the wall.

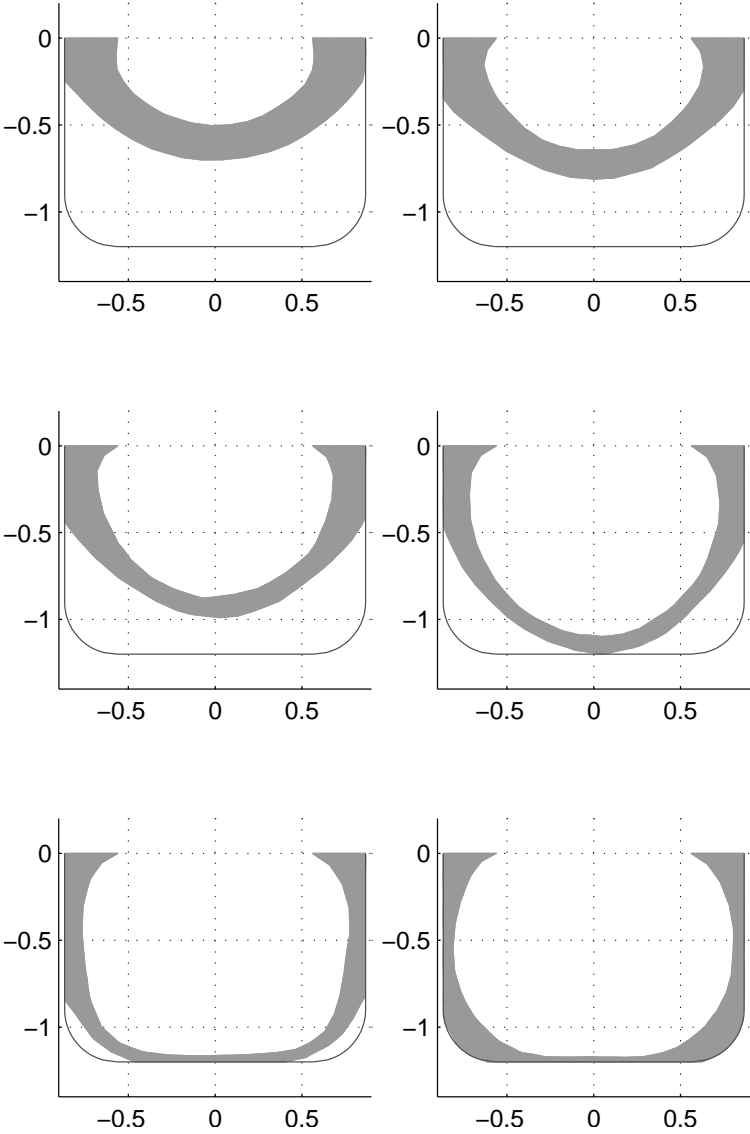


Figure 7.5: Cross-sectional view of Figure 7.4 at $y = 0$.

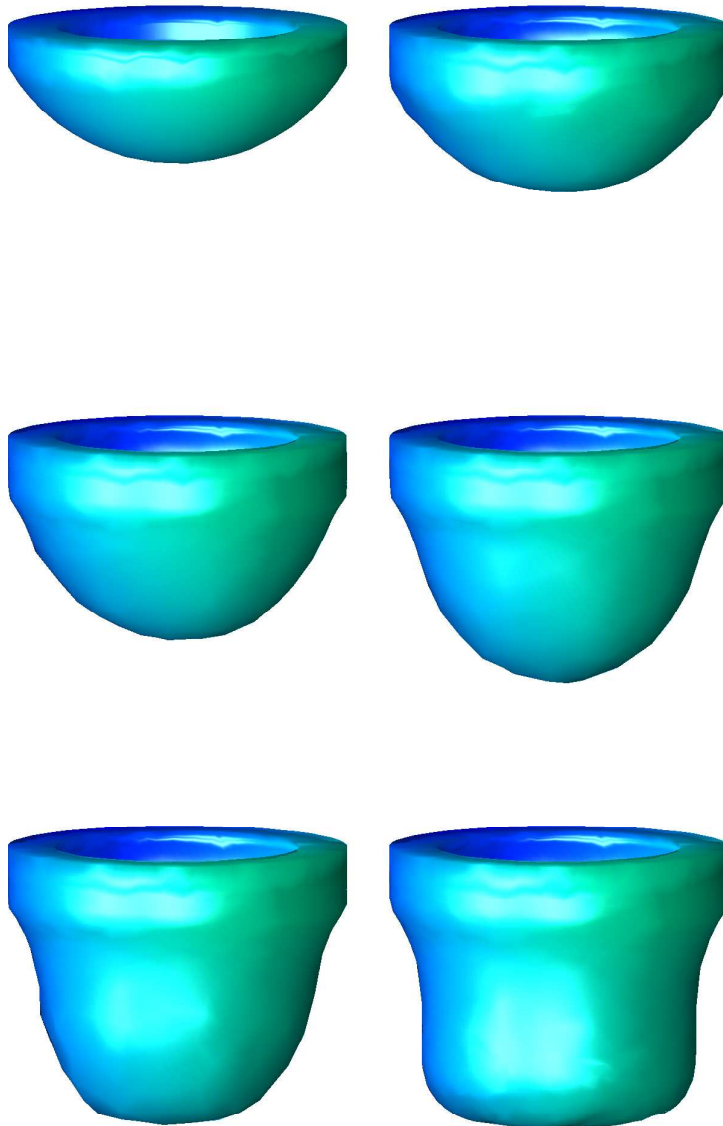


Figure 7.6: *During the first three snapshots the glass is sagging to the bottom of the mould. During the last three snapshots air is blowing into the mould from above. The glass is allowed to slip along the wall of the mould.*

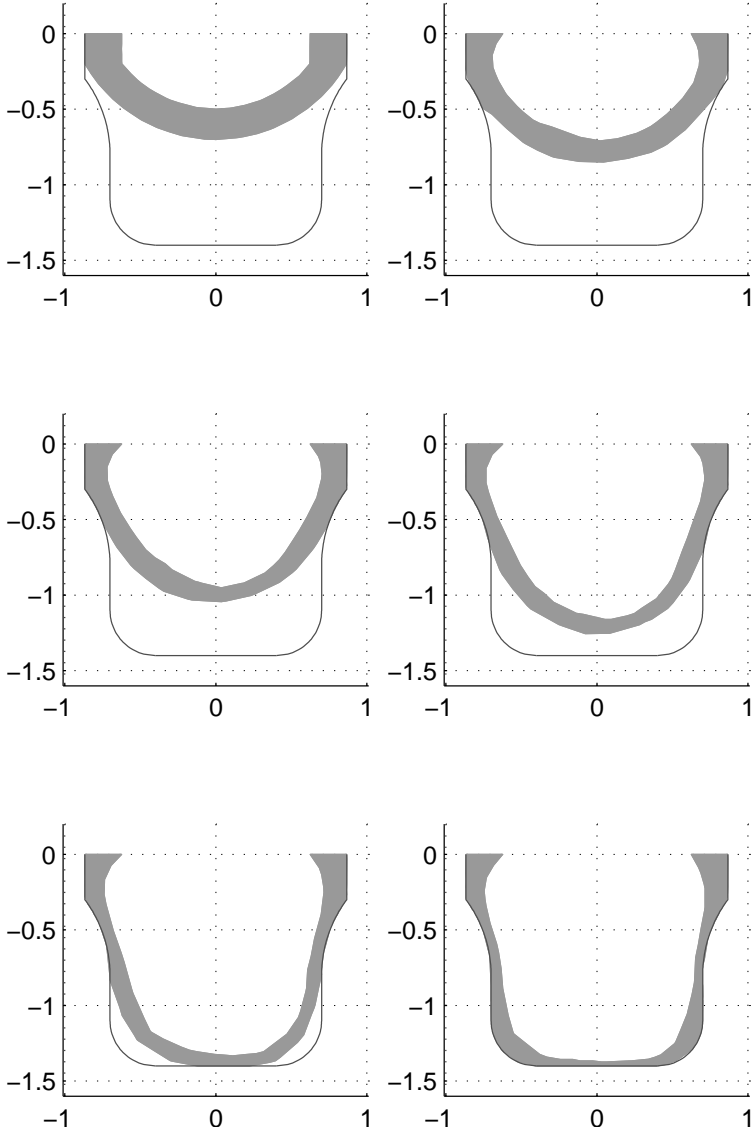


Figure 7.7: Cross-sectional view of Figure 7.6 at $y = 0$.

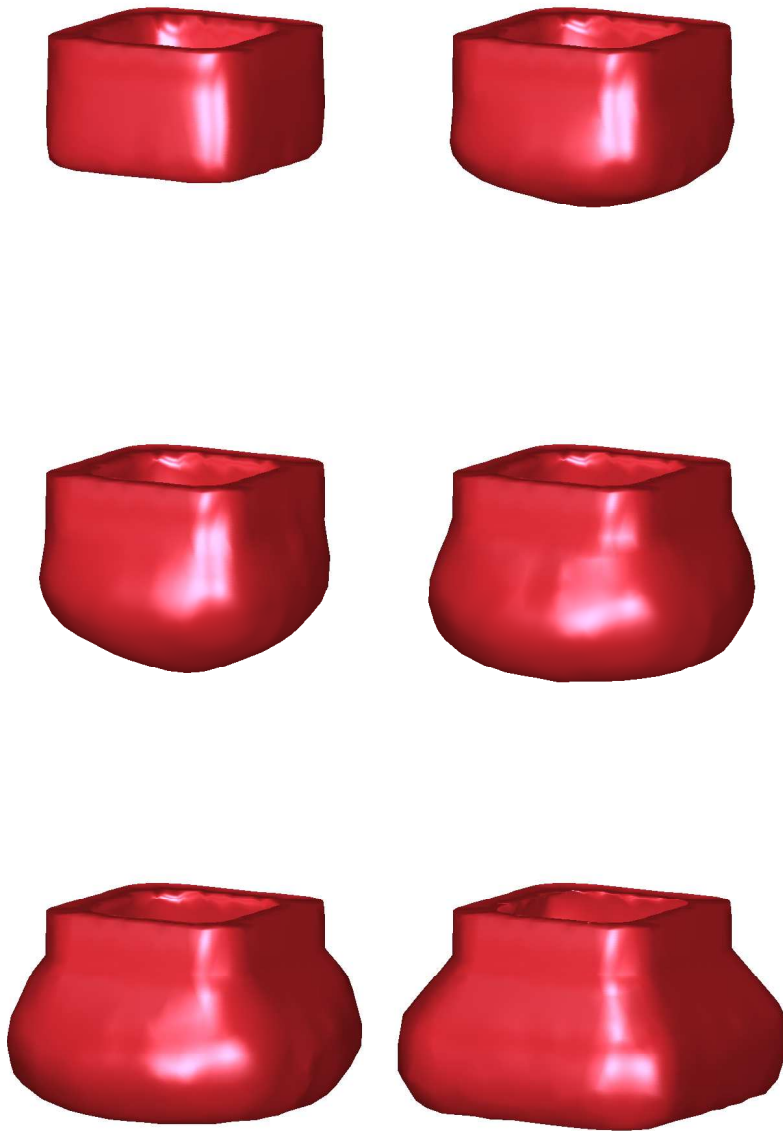


Figure 7.8: A square parison with rounded corners and a square mould with rounded corners. During the first three snapshots the glass is sagging to the bottom of the mould. During the last three snapshots air is blowing into the mould from above. The glass is not allowed to slip along the wall.

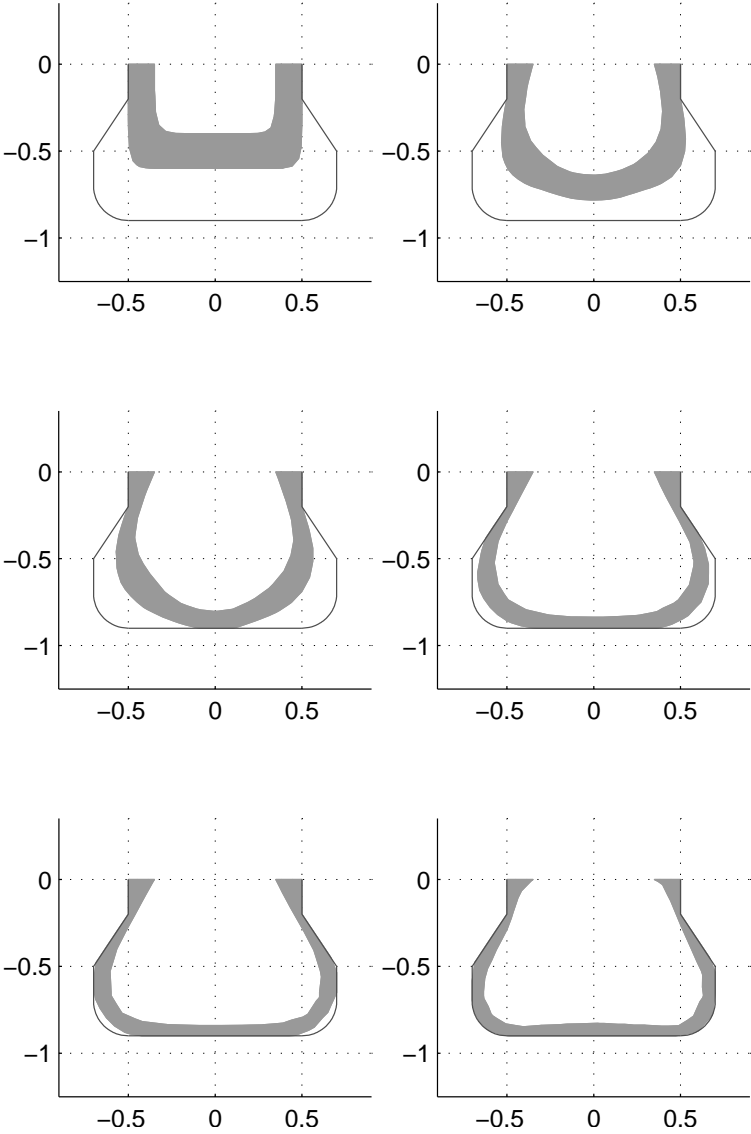


Figure 7.9: Cross-sectional view of Figure 7.8 at $y = 0$.

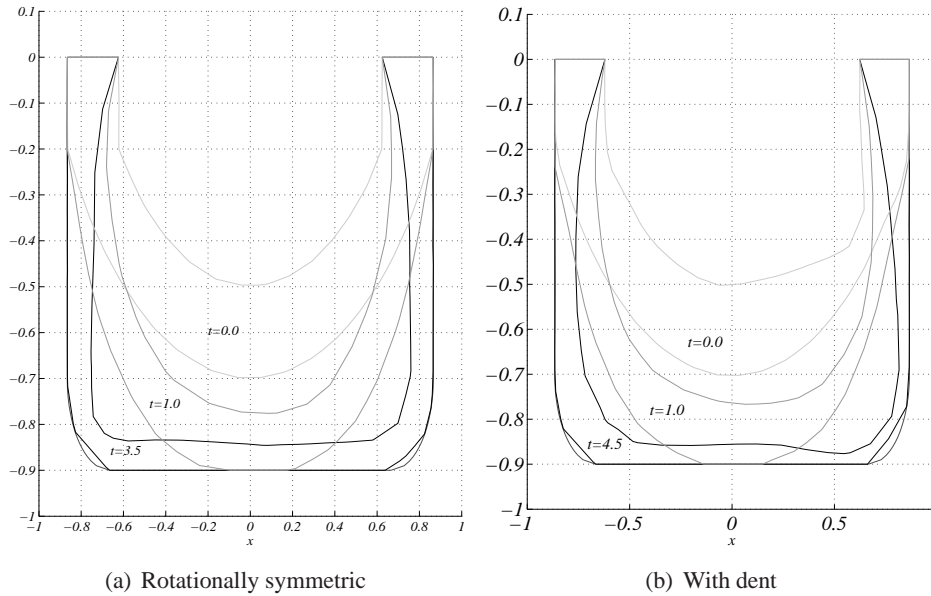


Figure 7.10: Comparison between a rotationally symmetric parison and a parison with a dent. The parisons are shown at three different time levels: $t = 0.0$, $t = 1.0$ and $t = t_{end}$.

Example 7.5

In Figure 7.10 we show the cross-sections of two parisons that are blown into a mould. The parison at the left-hand side, which we call parison 1, is perfectly rotationally symmetric, while the parison at the right-hand side, which we call parison 2, has a dent in the initial shape (at the right-hand side of the parison). Let t_{end} represent the (dimensionless) time level at which the glass product is finished, i.e. when the glass more or less covers the wall of the whole mould.

We compare the evolution of the two parisons at three different time levels: $t = 0.0$, $t = 1.0$ and $t = t_{end}$. The value of t_{end} differs for the two simulations. For parison 1 we find $t_{end} = 3.5$ while parison 2 has $t_{end} = 4.5$. Parison 2 takes approximately 28.9% longer to finish.

The dent is still present in the final product. The glass layer at the lower right corner of parison 2 is thinner than the glass layer at the same location of parison 1. Moreover the the glass layer at the lower left corner of parison 2 is thicker than the glass layer at the same location of parison 1. This is caused by the fact that the thin parts of the glass are easier to move than the thick parts. Hence the flow at the dent is relatively large. Since the total amount of glass needs to stay constant, the flow at other parts is relatively small. This is exactly what happens at the lower left corner of parison 1. \square

7.2 Curvature driven flow

In the absence of external forces, droplets of viscous fluid tend to evolve to a perfect sphere, regardless of the initial shape of the fluid. This is explained as follows. A molecule in the interior of the fluid is completely surrounded by other molecules. It experiences an attraction from all neighbouring molecules, which sums to zero. On the other hand, a molecule that is at the surface of the fluid experiences a net inward attraction. When the curvature of a convex surface is large, the inward attraction is large and when the curvature is small also the inward attraction is small. This introduces a potential energy of the fluid, which may be transformed to kinetic energy. For a spherically shaped fluid, the curvature is equal at each point of the surface. Hence the inward attraction is the same at each point. The fluid is in an equilibrium state and the potential energy will not be transformed to kinetic energy. In this section we show that the BEM model developed in Chapter 6 is also applicable for this type of evolution processes. Related work on two-dimensional fluid domains can be found in literature [78, 79, 94], in which both the direct and the indirect formulation of the BEM are used.

In this section we illustrate the evolution of viscous drops by looking at a number of fluid shapes that evolve to a sphere due to the surface tension. Using the notation from Chapter 6, we have $S_1 = S_2 = S_3 = \{\emptyset\}$ and $S \equiv S_0$, i.e. the fluid surface is a free surface. The boundary condition for this surface reads

$$\sigma \mathbf{n} = -(\alpha z + \beta \kappa) \mathbf{n}. \quad (7.2)$$

Since we do not want to take gravitational effects into account we take $\alpha = 0$, and as the only driving force of the flow is the surface tension, we may set $\beta = 1$.

Note that the choice $S \equiv S_0$ implies that we have to solve a Neumann problem. Hence we use the deflated form of the boundary integral operator \mathcal{H} , as is described in Section 6.3.

Example 7.6

In Figure 7.11 we see a beam-shaped volume of fluid evolving to a sphere. Figure 7.13(a) shows a cross-sectional view at $y = 0$ of the same fluid. The dimensions of the beam are 2, 2 and 1. The corners of the initial beam are rounded, as a viscous fluid will never have straight corners. For this particular simulation the initial number of nodes at the surface is $N = 500$ and the number of triangular elements is $K = 996$. The initial volume of the beam is 20.6465 and the volume of the final sphere to which it evolves is 20.6355. Hence we have a volume loss of less than 0.1%. Since the fluid approaches a sphere we can compute the radius that such a volume requires to find $r = 1.702$. The average radius of the nodes at the final surface is 1.714 with standard

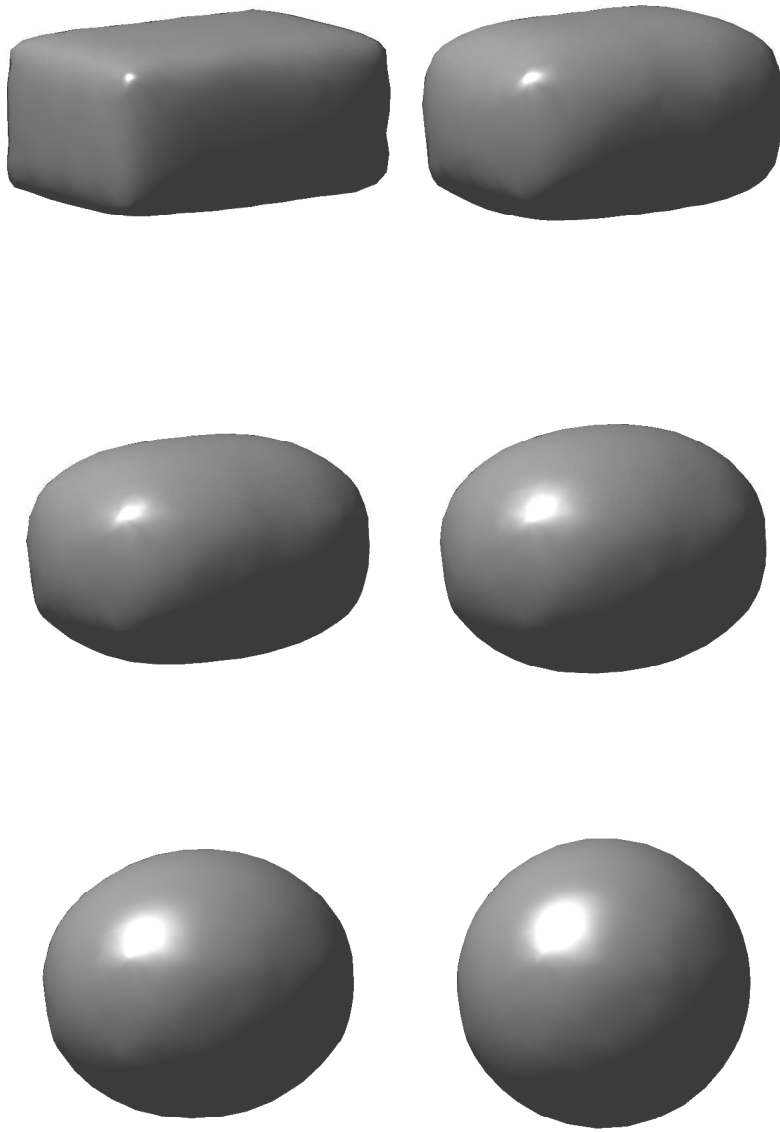


Figure 7.11: *A beam with sizes 2, 2 and 1 evolves to a sphere.*

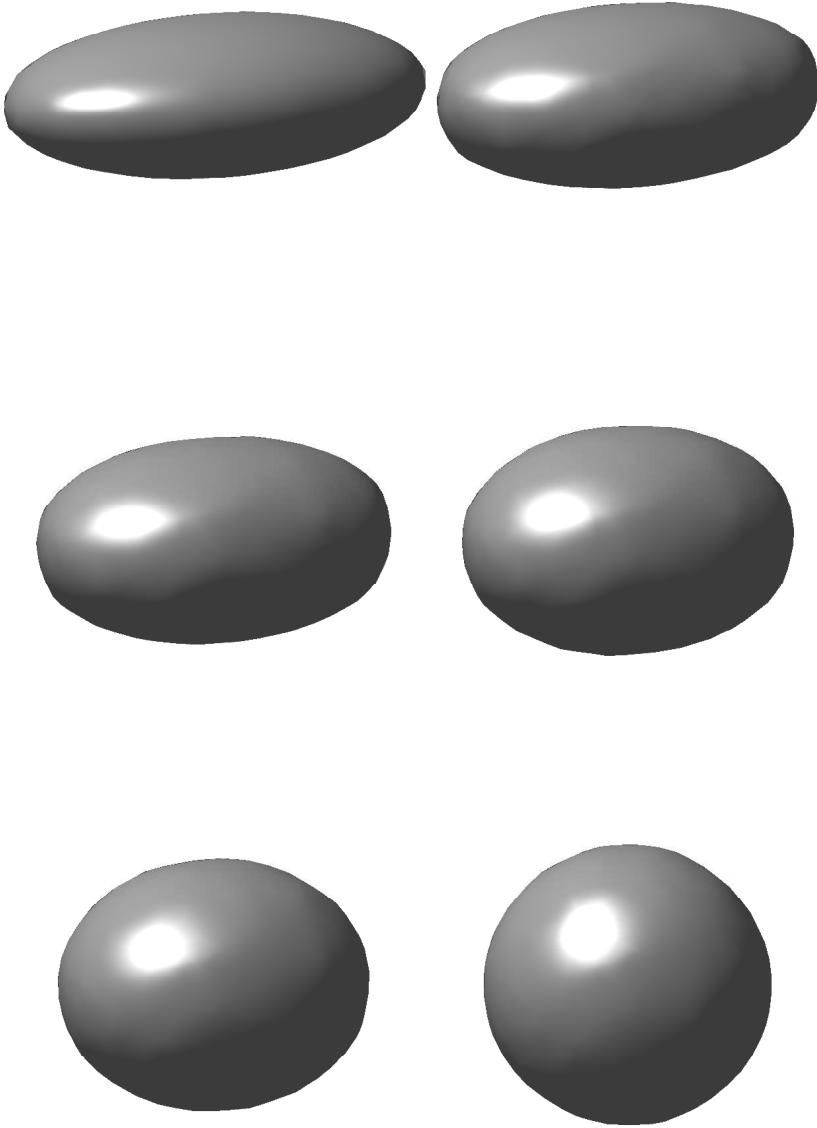


Figure 7.12: *An ellipse with semi-axes of length 3, 2 and 1 evolves to a sphere.*

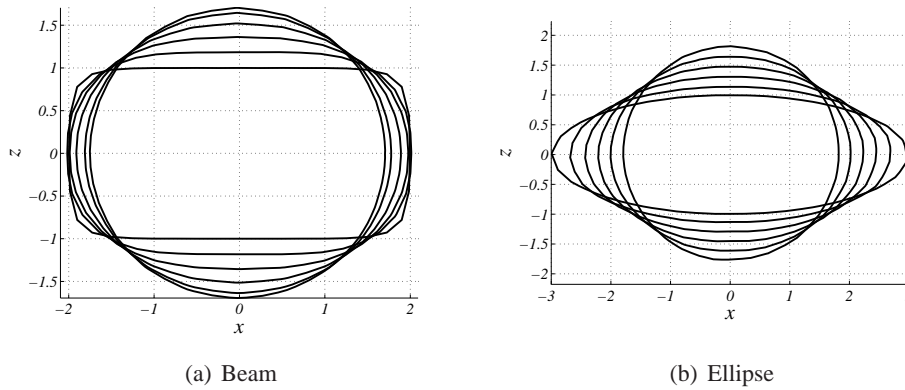


Figure 7.13: Cross-sectional views of Figure 7.11 and 7.12 at $y = 0$.

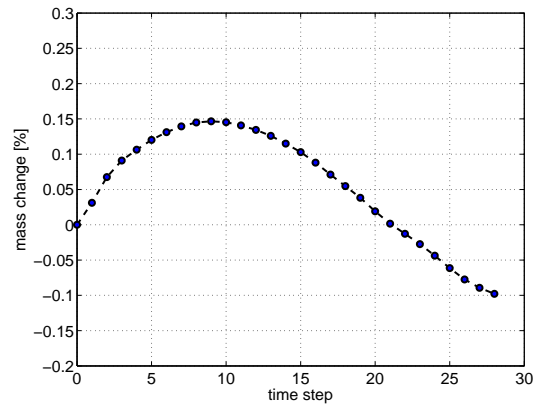


Figure 7.14: Volume change of the ellipsoid while evolving to a sphere.

deviation 0.025, which implies that the final surface is indeed a sphere with almost the appropriate radius. During the simulation we apply a remeshing technique that assures that all edges of the boundary elements are smaller than 0.39. This leads to a final discretisation with $N = 684$ nodes and $K = 1364$ elements. \square

Example 7.7

In the next example we watch an ellipsoidal volume of fluid evolve to a sphere (Figures 7.13(b) and 7.12). The initial ellipsoid has semi-axes of length 3, 2 and 1 and has $N = 500$ nodes and $K = 996$ elements at its surface. The volume of the

initial ellipse is equal to 24.56 while the volume of the final sphere is equal to 24.54. Hence the volume change is 0.08%. During the simulation we apply a remeshing technique that assures that all edges of the boundary elements are smaller than 0.4. This leads to a final discretisation with $N = 666$ nodes and $K = 1328$ elements. In Figure 7.14 we show the relative volume change of the ellipsoid during its evolution to a sphere. We observe that the volume first increases while later on it decreases again. \square

In Section 6.5 we mentioned two types of smoothing techniques to improve the quality of the BEM solution. We use Example 7.6 to study the effect of these smoothing techniques and to determine optimal settings for the smoothing.

The first technique is Laplacian smoothing, see Appendix C.1. This technique smooths the discretised surface and involves two parameters; the number of iterations N_1 and the weight w_1 . The second technique is the smoothing of a vector field at a surface, see Appendix C.3. This technique smooths the velocity field that is the solution of the Stokes equations. Involved are also two parameters; the number of iterations N_2 and the weight w_2 .

Similar to Example 7.6 we let a beam-shaped volume of fluid evolve to a sphere. We vary the smoothing parameters and determine for which values the final fluid approximates a sphere optimally.

The initial beam-shaped fluid has a volume of 20.6. When the radius of the final sphere is equal to $R := 1.702$, the sphere volume is also 20.6. This gives us a criterion to determine whether the discretised surface approximates a sphere accurately. We compute the distance of each node at the surface to the geometric center of the fluid. The average of these values should approximate 1.702. We also compute the standard deviation of these values. If the standard deviation is low, the fluid is close to a sphere. In this way we can study the effect of the smoothing techniques on the final geometry.

In the first test we choose $w_1 = w_2 = 0.4$ and vary the number of iterations. The results are shown in Table 7.1(a). We see that the average radius is close to the exact value in each case. The standard deviation is minimal for $N_1 = N_2 = 3$. In the second test we choose $N_1 = N_2 = 3$, i.e. the optimal choice from the previous test, and vary the weights w_1 and w_2 . The results are shown in Table 7.1(b). We observe that the best results are obtained when $w_1 = w_2 = 0.1$. In this case the radius is closest to the exact radius and the standard deviation is minimal.

Until now we chose the same number of iterations and weights for both smoothing techniques. In the next test we choose $w_1 = w_2 = 0.1$ and we take several combinations of numbers of iterations N_1 and N_2 . The results are shown in Table 7.2(a). Clearly the effect of the velocity smoothing is smaller than the effect of Laplacian smoothing.

In the last test we do not perform velocity smoothing ($N_2 = 0$) and we choose

N_1	N_2	\bar{r}	std(r)	w_1	w_2	\bar{r}	std(r)
0	0	1.7100	0.0550	0.1	0.1	1.7081	0.0110
1	1	1.7080	0.0162	0.4	0.4	1.7093	0.0139
3	3	1.7093	0.0139	0.7	0.7	1.7096	0.0154
6	6	1.7074	0.0213	1.0	1.0	1.7124	0.0242

Table 7.1: The mean radius and standard deviation of all nodes at the surface of the sphere for (a) $w_1 = w_2 = 0.4$ and (b) $N_1 = N_2 = 3$.

N_1	N_2	\bar{r}	std(r)	w_1	\bar{r}	std(r)
3	3	1.7090	0.0124	0.1	1.7081	0.0105
1	3	1.7131	0.0145	0.2	1.7078	0.0169
3	1	1.7081	0.0114	0.3	1.7071	0.0097
3	0	1.7081	0.0105	0.4	1.7089	0.0180
6	0	1.7085	0.0166	0.6	1.7115	0.0218

Table 7.2: The mean radius and standard deviation of all nodes at the surface of the sphere for (a) $w_1 = w_2 = 0.1$ and (b) $N_1 = 3$ and $N_2 = 0$.

$N_1 = 3$. We let the weight w_1 vary. The results are shown in Table 7.2(b). We see that the best results are obtained when $w_1 = 0.3$. In this case the average radius is very close to the exact radius and the standard deviation is minimal.

The main conclusions from these tests on smoothing techniques are:

- Velocity smoothing has less effect than Laplacian smoothing;
- The optimal number of iterations for Laplacian smoothing is $N_1 = 3$;
- The optimal weight for Laplacian smoothing is $w_1 = 0.3$.

In all simulations presented in this chapter we perform the smoothing techniques with the settings equal to or close to the optimal settings listed above.

7.3 Parameter analysis

In this section we investigate the role of several model parameters for the glass blowing problem. To this end we simulate the flow of a cup-shaped amount of glass.

Example 7.8

In the first example we let gravity act on the glass, which will cause the glass to sag downward. We study the significance of the surface tension during this process. The effect of surface tension is represented by the value of the dimensionless number β , which involves the surface tension. As gravity is the driving force we set $\alpha = 1$. The values of β that we use in this example lie in the range of 0.0 to 0.4. If $\beta = 0.0$, then the glass does not have surface tension. A large positive value of β implies a high surface tension.

We let the glass sag downwards for a fixed amount of time and plot the resulting shape in Figure 7.15. We do this for several values of β . The left pane of the figure gives a cross-section of the glass, while the right pane zooms in on the lowest part of this cross-section. Several lines are plotted, each representing a specific value of β . In the left pane, the various shapes, corresponding to the various values of β , more or less coincide. In the right pane we are able to distinguish the various shapes. We observe that the glass flow has moved downward the furthest for $\beta = 0.0$, while for large values of β the glass has not moved very far. Hence the surface tension slows down the flow of the glass.

Table 7.3 gives the height of the lowest point of the glass surface for several values of β . We compare these heights with the height for $\beta = 0.0$. We observe that, for $\beta = 1.0$, the glass surface is more than 2% higher than for $\beta = 0.0$. Hence the surface tension plays a small role in the case of gravity driven flow. Therefore we include surface tension in all numerical simulations that cover the sagging stage. \square

β	z_{min}	difference (%)
0.0	-0.8970	-
0.1	-0.8908	+0.69
0.2	-0.8858	+1.25
0.3	-0.8813	+1.75
0.4	-0.8765	+2.29

Table 7.3: The z -coordinate of the lowest node after sagging for several values of the surface tension β .

Example 7.9

In the next example we investigate the significance of the surface tension during the blowing stage. Hence the glass does not sag due to gravity in this case, but it flows due to a pressure difference between upper and lower layer of the glass. In this case

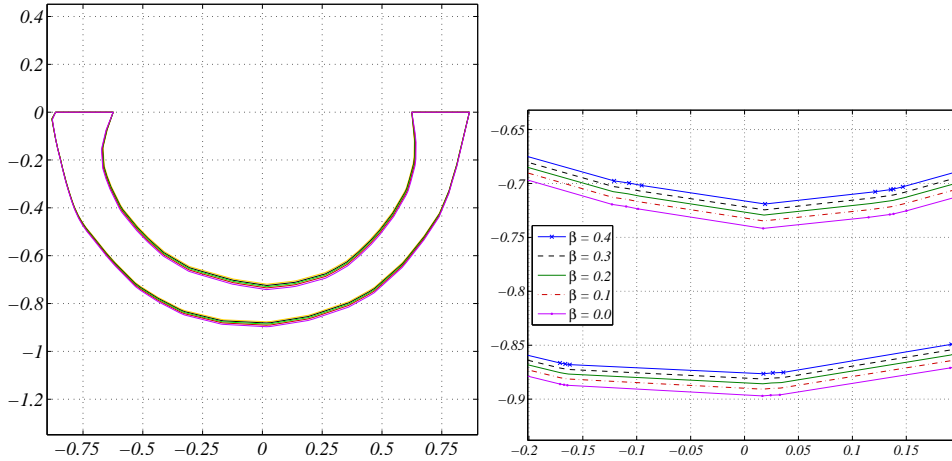


Figure 7.15: The effect of the surface tension on the flow of the glass compared to the effect of gravity. High values of β represent strong surface tension. In the left pane a cross-section of the glass is shown, the right pane zooms in on the lower part of this cross-section.

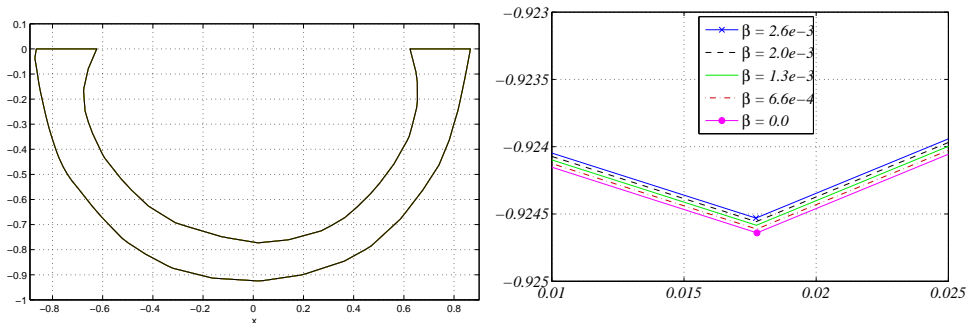


Figure 7.16: The effect of the surface tension on the flow of the glass compared to the effect of pressure. High values of β represent strong surface tension. In the left pane a cross-section of the glass is shown, the right pane zooms in on the lowest part of this cross-section.

we set $\alpha = 6.45 \cdot 10^{-3}$, which indicates that the contribution of the gravity is small compared to pressure. The values of β range from 0.0 to $2.6 \cdot 10^{-3}$. The left pane of Figure 7.16 shows a cross-section of the glass, while the right pane zooms in on the lowest part of this cross-section. We see that the various shapes that correspond to various values of β are almost similar. This shows that the surface tension is of no importance during the blowing stage. The flow is dominated by the pressure

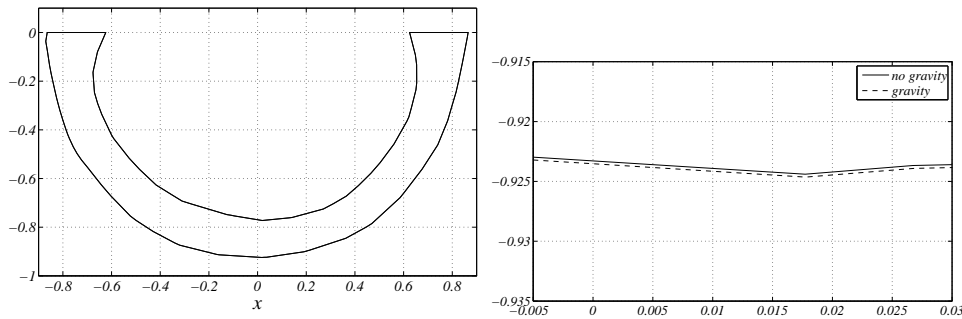


Figure 7.17: The effect of the gravity on the flow of the glass compared to the effect of pressure. In the left pane a cross-section of the glass is shown, the right pane zooms in on the lowest part of this cross-section.

difference. Therefore we do not need to include surface tension in the numerical simulations for the blowing stage. \square

Example 7.10

In this example we investigate whether we may also neglect gravity during the blowing stage. Figure 7.17 shows the shape of the glass in the blowing stage after some time. Two shapes are plotted: one shape corresponds to the simulation in which gravity is included, and the other shape corresponds to the simulation without gravity. The left pane gives a cross-section of the glass but we see no differences between the two shapes. In the right pane we zoom in on the lower part of the cross-section. Here we see that in the simulation where gravity is included, the glass moved downward a bit further than in the simulation without gravity. Still the differences are very small and we may conclude that gravity is negligible during the blowing stage. \square

Little is known about the friction parameter β_m . To the author's knowledge there are no references in literature to experiments in which the friction parameters for glass towards metals are determined. Therefore we perform a series of simulations in which we vary β_m . This shows how the flow of the glass depends on the friction parameter and helps us to determine realistic values of β_m .

Example 7.11

Figure 7.18 shows the new parison after a fixed number of time steps for several values of β_m . Both the mould and initial parison have a rotationally symmetric shape. In this simulation gravity is the only driving force for the flow of the glass.

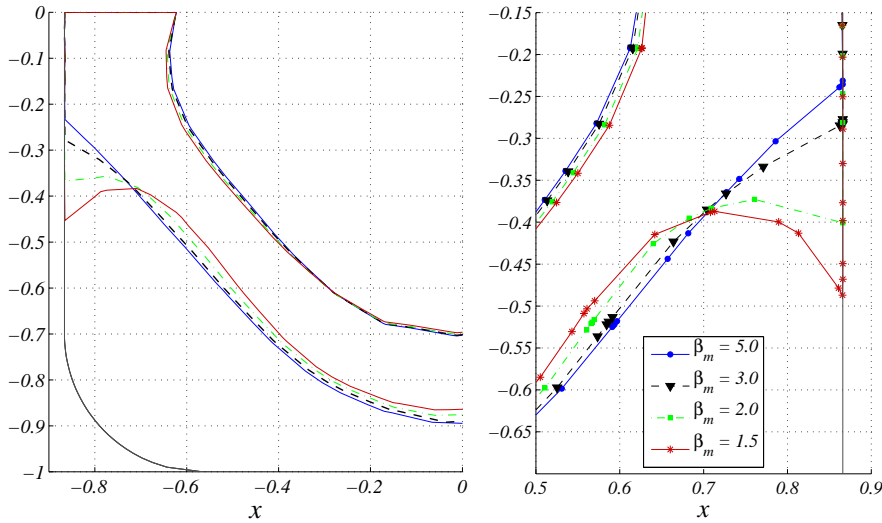


Figure 7.18: Deformation of the glass after a fixed number of time steps for different values of β_m .

Theoretically, for $\beta_m = \infty$ the friction between glass and mould is infinitely large and the glass does not slip at all along the wall of the mould; for $\beta_m = 0$ there is no friction, and the glass can flow freely along the wall of the mould. We observe that for $\beta_m = 5$ the glass has slipped a little along the wall of the mould. This slip becomes larger if the friction parameter is lowered further. For $\beta_m = 2$ and $\beta_m = 1.5$ we see that the glass that is in contact with the wall moves faster downward than the glass that does not touch the wall. Hence these values of β_m are not realistic. Appropriate choices for β_m lie in the range of 3 to ∞ . Note that we use the dimensionless friction parameter here, which depends on the length scale and the viscosity. Hence for examples that have other characteristic lengths and viscosities, suitable values for β_m may be slightly different than the values that we find for this example. \square

Example 7.12

Example 7.11 shows that a minimal value for the friction parameter is $\beta_m = 3$, based on observations of intermediate shapes during sagging. In the current example we elaborate a bit more on values of β_m that are in the interval $[5, \infty)$. For several values of β_m we simulate the blowing stage and plot the final shapes in Figure 7.19. We observe that the shapes vary significantly.

In Table 7.4 we give the height of the lowest point of the glass surface for several

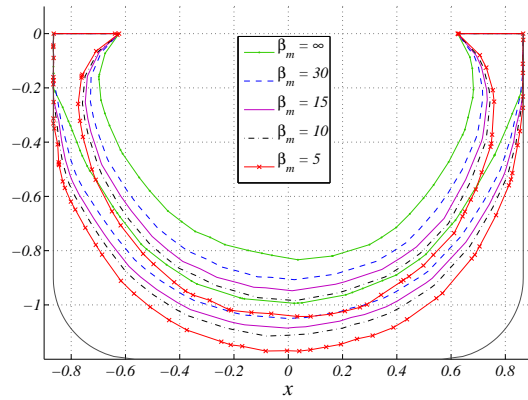


Figure 7.19: The effect of friction on the flow of the glass compared to the effect of pressure. High values of β_m represent much friction.

β_m	z_{min}	difference (%)
∞	-0.90	–
30	-1.05	-6
15	-1.09	-9
10	-1.12	-12
5	-1.17	-18

Table 7.4: The z -coordinate of the lowest node after the blowing stage for several values of the friction parameter β_m .

values of β_m . We compare these heights with the height for $\beta_m = 0$. We see that $\beta_m = 5$ gives a glass surface that is 18% lower than the surface for $\beta_m = \infty$. Hence the friction between glass and mould is very important and cannot be neglected. \square

Appendix A

Curvature approximation

This appendix describes the approximation of curvature parameters of a discretised surface. When a surface is given by a parametric representation these curvature parameters can be calculated analytically. However in many numerical applications the surface is a discretised surface, consisting of a large number of triangular elements and nodes. For such discretised surfaces it is not so straightforward to compute curvature parameters. However, there exists a number of strategies to approximate curvature parameters, of which the *paraboloid fit method* appears to be the most accurate one [89]. We use a similar method in which a *bicubic* polynomial is fitted through data points.

Assume that a surface S is uniquely described by $z = f(x, y)$, where (x, y) is in a closed and bounded set $D \subset \mathbb{R}^2$. The *Gaussian curvature* of the surface at a point $(x, y) \in D$ is defined as

$$K := \frac{f_{xx}f_{yy} - f_{xy}^2}{(1 + f_x^2 + f_y^2)^2} \Big|_{(x,y)}, \quad (\text{A.1})$$

and the *mean curvature* as

$$H := \frac{(1 + f_x^2)f_{xx} - 2f_x f_y f_{xy} + (1 + f_y^2)f_{yy}}{2(1 + f_x^2 + f_y^2)^{3/2}} \Big|_{(x,y)}. \quad (\text{A.2})$$

Related to these two curvature parameters are the *principal curvatures* κ_1 and κ_2 . They satisfy the relations

$$K = \kappa_1 \kappa_2, \quad H = \frac{1}{2}(\kappa_1 + \kappa_2). \quad (\text{A.3})$$

The principal curvatures can also be found as follows. Let x be a point at the surface S and let a vertical plane intersect the surface in the point x . The intersection of the surface and the plane yields a curve C . By rotating the planes around the

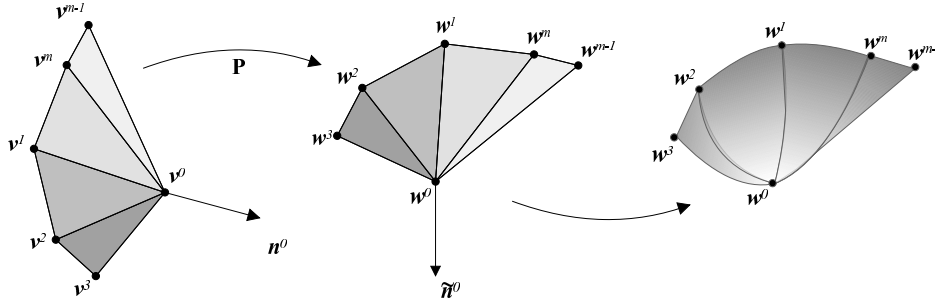


Figure A.1: The nodes v^0 and v^1, \dots, v^m are translated and rotated in such a way that v is in the origin and its normal vector points in the vertical direction.

vertical axis we get a set of curves C . For each curve we calculate its the second derivative at the point x . The largest second derivative is the largest principal curvature κ_1 and the smallest second derivative is the smallest principal curvature κ_2 .

The triangulated surface T is described by a set of nodes x_p at the surface S , $p = 1, \dots, N$ and a set of triangles T_k , $k = 1, \dots, K$, each triangle defined by three nodes. We want to approximate the curvature in each node x_p using only the information of the locations of neighbouring nodes. A node is a neighbour of x_p if they share a triangle T_k . The procedure to approximate the curvature goes as follows. Let v^0 be the node at which we want to approximate the curvature. Let v^1, \dots, v^m be the m neighbouring nodes of v^0 . First we translate and rotate the nodes v^0, v^1, \dots, v^m in such a way that v^0 is translated to the origin and the normal vector in v^0 at the surface points in vertical direction, see Figure A.1. Note that a translation and rotation of the surface does not affect the curvature parameters.

We fit a bicubic polynomial $p(x, y)$ through the nodes v^0, v^1, \dots, v^m using a least squares technique. The translation and rotation of the nodes guarantees that such a polynomial exists. The polynomial provides a local parameterisation of the surface of the form $z = p(x, y)$. Hence we can use (A.1) and (A.2) to calculate the curvature parameters in $(x, y) = (0, 0)$ at the parametric surface.

We describe the procedure above in more detail. Define the edge between v^0 and v^i by $e^i := v^0 - v^i$, $i = 1, \dots, m$. Let the k -th triangle be given by the nodes v^0, v^{k_1} and v^{k_2} . For the outward normal at this triangle we use the straightforward definition,

$$\mathbf{n}^k := \frac{\mathbf{e}^{k_1} \times \mathbf{e}^{k_2}}{\|\mathbf{e}^{k_1} \times \mathbf{e}^{k_2}\|}. \quad (\text{A.4})$$

The normal vector at the point v^0 can be defined in several ways. In each case the normal vector at v^0 is a weighted sum of the normals at the surrounding triangles. For the weights several options are available.

1. Average of normals at surrounding triangles,

$$\mathbf{n} := \frac{1}{m} \sum_{k=1}^m \mathbf{n}^k. \quad (\text{A.5})$$

2. If A_k is the area of the k -th triangle,

$$\mathbf{n} := \sum_{k=1}^m \frac{1}{A_k} \mathbf{n}^k. \quad (\text{A.6})$$

3. Define the weight θ_k by

$$\cos \theta_k := \frac{\langle \mathbf{e}^{k_1}, \mathbf{e}^{k_2} \rangle}{\|\mathbf{e}^{k_1}\| \|\mathbf{e}^{k_2}\|}, \quad (\text{A.7})$$

i.e. the angle between the two edges of a triangle that meet in \mathbf{v}^0 , and

$$\mathbf{n} := \sum_{k=1}^m \theta_k \mathbf{n}^k. \quad (\text{A.8})$$

Later on we will demonstrate the effect of the different definitions of \mathbf{n} on the curvature approximation. It turns out that the last definition gives the most accurate approximation.

We want to rotate the nodes $\mathbf{v}^0, \mathbf{v}^1, \dots, \mathbf{v}^m$ such that the normal in \mathbf{v}^0 points in either positive or negative vertical direction. For this goal we define two rotation matrices

$$\mathbf{R}_0 := \begin{bmatrix} \cos \alpha_1 & 0 & -\sin \alpha_1 \\ 0 & 1 & 0 \\ \sin \alpha_1 & 0 & \cos \alpha_1 \end{bmatrix}, \quad \mathbf{R}_1 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_2 & -\sin \alpha_2 \\ 0 & \sin \alpha_2 & \cos \alpha_2 \end{bmatrix}, \quad (\text{A.9})$$

which rotate the nodes over an angle α_1 with respect to the y -axis and over an angle α_2 with respect to the x -axis respectively. Write $\mathbf{n} = [n_1, n_2, n_3]^T$ and denote the rotation of \mathbf{n} over α_1 by $\mathbf{n}^1 := \mathbf{R}_0 \mathbf{n}$. We write $\mathbf{n}^1 = [n_1^1, n_2^1, n_3^1]^T$ and denote the rotation of \mathbf{n}^1 over α_2 by $\mathbf{n}^2 := \mathbf{R}_1 \mathbf{n}^1$. It can be shown that \mathbf{n}^2 points in positive or negative z -direction if α_1 and α_2 are chosen as follows,

$$\alpha_1 = \begin{cases} \arctan\left(\frac{n_1}{n_3}\right) & \text{if } n_3 \neq 0, \\ \frac{\pi}{2} & \text{if } n_3 = 0, \end{cases} \quad \alpha_2 = \begin{cases} -\arctan\left(\frac{n_2^1}{n_3^1}\right) & \text{if } n_3^1 \neq 0, \\ \frac{\pi}{2} & \text{if } n_3^1 = 0. \end{cases} \quad (\text{A.10})$$

Having determined the appropriate rotation angles α_1 and α_2 , we can rotate all nodes $\mathbf{v}^0, \mathbf{v}^1, \dots, \mathbf{v}^m$, using the rotation matrix $\mathbf{P} := \mathbf{R}_1 \mathbf{R}_0$. This yields a set of new nodes,

$$\begin{aligned}\tilde{\mathbf{w}}^0 &:= \mathbf{P} \mathbf{v}^0, \\ \tilde{\mathbf{w}}^i &:= \mathbf{P} \mathbf{v}^i, \quad i = 1, \dots, m.\end{aligned}\tag{A.11}$$

It may occur that at this stage the normal vector at $\tilde{\mathbf{w}}^0$ is pointing in the positive z -direction. In that case we simply mirror all nodes with respect to the (x, y) -plane. Thus the normal at $\tilde{\mathbf{w}}^0$ always point in the negative z -direction. Finally we translate all nodes over the same distance such that $\tilde{\mathbf{w}}^0$ is located in the origin. This yields the following set of nodes,

$$\begin{aligned}\mathbf{w}^0 &:= \tilde{\mathbf{w}}^0 - \tilde{\mathbf{w}}^0 = \mathbf{0}, \\ \mathbf{w}^i &:= \tilde{\mathbf{w}}^i - \tilde{\mathbf{w}}^0, \quad i = 1, \dots, m.\end{aligned}\tag{A.12}$$

Let $p(x, y) := a_1 x^2 + a_2 x y + a_3 y^2 + a_4 x^3 + a_5 y^3$ be a bicubic polynomial. The equation $z = p(x, y)$ describes a 2D surface. We use a linear least squares technique to find the polynomial p that gives the best fit of the nodes $\mathbf{w}^0, \mathbf{w}^1, \dots, \mathbf{w}^m$. In this way we create a parametric representation of a surface that locally approximates the discretised surface S . The Gaussian and mean curvature at the node \mathbf{v}^0 of the discretised surface are obtained by evaluating (A.1) and (A.2) with $f \equiv p$ at the point $(x, y) = (0, 0)$. They can be expressed in terms of the coefficients of the polynomial p ,

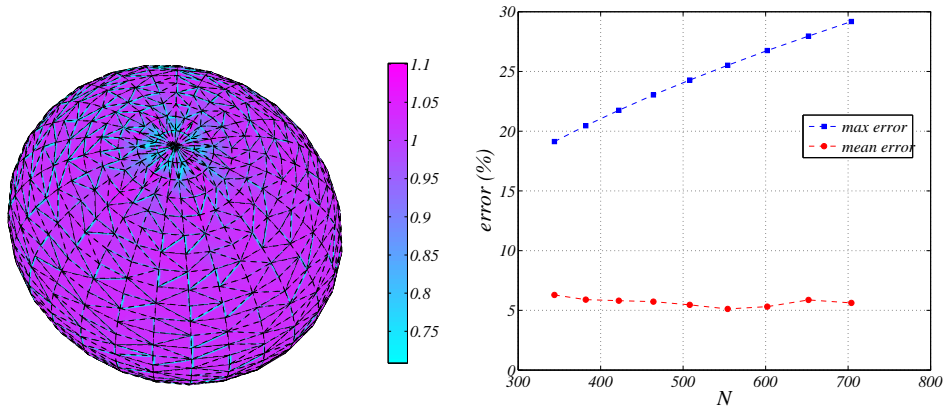
$$\begin{aligned}K &= 4a_1 a_3 - a_2^2, \\ H &= a_1 + a_3.\end{aligned}\tag{A.13}$$

We illustrate the procedure to approximate the curvature of a discretised surface for a sphere and an ellipsoid.

Example A.1

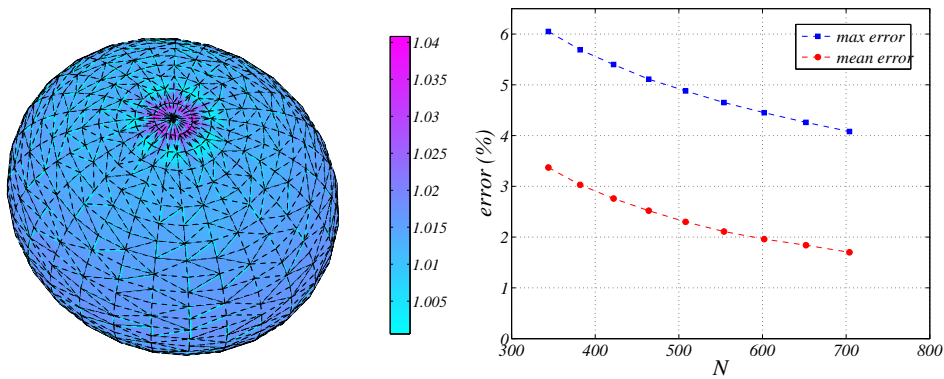
In Figure A.2(a) we show a unit sphere discretised with $N = 704$ nodes. It can be shown that the unit sphere has a Gaussian curvature equal to 1 in each point at the surface. The color at the surface represents the value of the Gaussian curvature as approximated by the polynomial fit method. We observe that the approximation is not equal to 1 everywhere.

We repeat the curvature approximation for the unit sphere with other numbers of nodes N . In Figure A.2(b) we show the maximum and mean error of the curvature approximation as a function of number of nodes N . To our surprise the mean error does not decrease if the number of nodes increases, i.e. the discretisation refines. Moreover the maximal error is even increasing with the number of nodes. \square



(a) A discretised sphere with $N = 704$ nodes. Color represents the Gaussian curvature. (b) The maximal and mean error in the curvature approximation of a sphere as a function of number of nodes N .

Figure A.2: Gaussian curvature approximation for a sphere. All neighbouring nodes are used in the approximation.



(a) A discretised sphere with $N = 704$ nodes. Color represents the Gaussian curvature. (b) The maximal and mean error in the curvature approximation of a sphere as a function of number of nodes N .

Figure A.3: Gaussian curvature approximation for a sphere. Only the five nearest neighbouring nodes are used in the approximation.

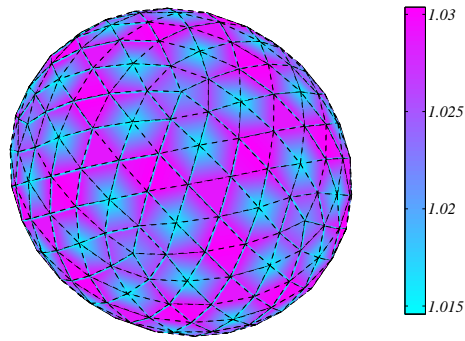


Figure A.4: A sphere with a more regular discretisation. Color represents the Gaussian curvature.

max nr. of neighbours	max. error (%)	mean error (%)
4	4.08	1.88
5	4.08	1.70
6	29.18	5.86
7	29.18	5.54
8	29.18	5.62
all	29.18	5.62

Table A.1: The maximal and mean error in the curvature approximation when a certain number of neighbours is used for creating a polynomial fit.

Example A.2

The way to improve the curvature approximation is by using only the nearest neighbours of a node when constructing the bicubic polynomial. In Figure A.3 we show the curvature approximation when at most five neighbours are used. We see that the errors are much smaller and that the accuracy increases when we take more nodes. We also investigate what number of neighbours gives the best curvature approximation. In Table A.1 we give the maximal and mean error when a certain number of neighbours is used. We see that for five neighbours the approximation is the most accurate. \square

weight	max. error (%)	mean error (%)
$\frac{1}{m}$	57.08	7.35
$\frac{1}{A_k}$	36.27	5.94
θ_k	4.08	1.70

Table A.2: The maximal and mean error in the curvature approximation for a sphere for each of the three definitions of the normal at a node. Number of nodes is $N = 704$.

Example A.3

The accuracy of the curvature approximation also depends on the discretisation of the surface. In Figure A.3 we see that the largest errors are at the top of the sphere, where we have a number of narrow triangular elements. In Figure A.4 we show a sphere with a more regular discretisation, i.e. all elements have approximately the same shape and size. Although the accuracy is more or less the same as for the sphere in Figure A.3, the largest errors are not concentrated at one area. \square

Example A.4

We mentioned that there are several possibilities to define the normal vector at a node. In Table A.2 we show the maximal and mean error in the curvature approximation for each of the three definitions. It is clear that the definition in which we use the weights θ_k results in a higher accuracy. \square

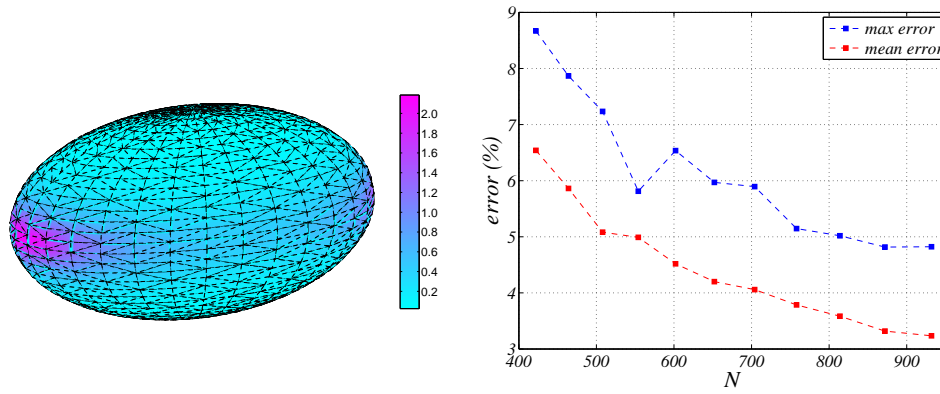
Also for an ellipsoid the Gaussian curvature can be calculated explicitly. For a point (x, y, z) at the surface of the ellipsoid the Gaussian curvature is given by

$$K = \frac{1}{a^2 b^2 c^2} \left(\frac{x^2}{a^4} + \frac{y^2}{b^4} + \frac{z^2}{c^4} \right)^{-2}, \quad (\text{A.14})$$

where a , b and c are the lengths of the semi-axes.

Example A.5

In Figure A.5(a) we show an ellipsoid where the semi-axes have length 3, 2 and 1, discretised with $N = 932$ nodes. The color at the surface represents the Gaussian curvature of the surface. We see that at the tips $x \pm 3$ the curvature is maximal, which agrees with the maximal value of the analytic expression for K . We repeat the curvature approximation for the ellipsoid with other numbers of nodes N . In Figure A.5(b) we give the maximal error and the mean error of the curvature



(a) A discretised ellipsoid with $N = 932$ nodes. (b) The maximal and mean error in the curvature approximation of the ellipsoid as a function of number of nodes N . Color represents the Gaussian curvature.

Figure A.5: *Gaussian curvature approximation for an ellipsoid. Only the five nearest neighbouring nodes are used in the approximation.*

approximation as a function of N . We see that for a fine discretisation of the surface ($N > 550$) the mean error is lower than 5%. \square

Appendix B

Contact problem

In this appendix we address the contact problem of a fluid with a wall. In many mathematical models for fluid flow it is assumed that the fluid is already in contact with a wall. At the contact area a slip or no-slip boundary condition is prescribed that ensures that the fluid remains in contact with the wall. For the blowing problem covered in this thesis the fluid may, in first instance, not yet be in contact with the wall. However as the fluid expands into the direction of the wall, at a certain moment it will touch the wall. From that moment on, the fluid will stay connected to the wall at that specific point. Thus a fluid particle may at first instance be part of a free boundary, while some time later it is fixed at a wall (although it may slide along the wall in the case of a slip boundary condition).

For a flow that is computed numerically the touching of the wall is difficult to implement. In boundary element methods, the fluid domain is described by a set of nodes and elements at the surface of the domain. Moreover the trajectory of a node or fluid particle is only known at a discrete set of time points. Thus it may occur that at time $t = t^n$ a fluid particle is not yet at the wall, while at time $t = t^{n+1}$ the particle has moved to the other side of the wall, see Figure B.1. Physically this is not possible, but it is very difficult to “tell” the fluid particle that it cannot move through the wall.

When an explicit time integration method is used to track the flow of the fluid, e.g. Euler forward, there is a simple way to avoid nodes moving through the wall. One can find an intermediate time $t^n < t^* < t^{n+1}$ such that the node that crossed the wall lies exactly at the wall at $t = t^*$. In this way we effectively decrease the time step of the time integration method to a suitable size. In practice one should thus verify all nodes at $t = t^{n+1}$ whether they moved through the wall. The node that has moved through the wall the furthest determines the new time step size. The drawback of the procedure is that the time steps can become very small, especially when the number of nodes is large.

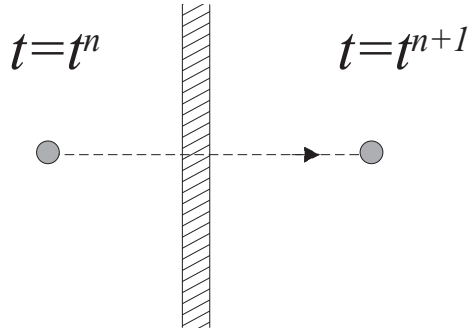


Figure B.1: At time $t = t^n$ a fluid particle is at the left-hand side of a wall, while at $t = t^{n+1}$ the particle is at the right-hand side of the wall.

In the sequel we present an alternative method to prevent fluid particles from moving through a wall for the 3D blowing problem. Let the surface S of a fluid domain be represented by N nodes and a set of triangular elements, the nodes lying at the corners of the triangles. Let W be a (possibly curved) surface representing the wall, for which a parametric representation is given. At time $t = t^n$ the coordinates of all nodes are given. With a numerical method the coordinates at time $t = t^{n+1}$ are computed. Assume that a node with coordinates \mathbf{x} has crossed the surface W during the time interval (t^n, t^{n+1}) . Let \mathbf{x}^{old} denote the old coordinates of \mathbf{x} at $t = t^n$. The strategy is to translate \mathbf{x} over a distance $d\mathbf{x}$, back into the direction of \mathbf{x}^{old} , such that $\mathbf{x} + d\mathbf{x}$ lies exactly at W .

The straight line from \mathbf{x}^{old} to \mathbf{x} can be parameterised by

$$\mathbf{y} = \mathbf{y}(\xi) = (1 - \xi)\mathbf{x}^{old} + \xi\mathbf{x}, \quad 0 \leq \xi \leq 1. \quad (\text{B.1})$$

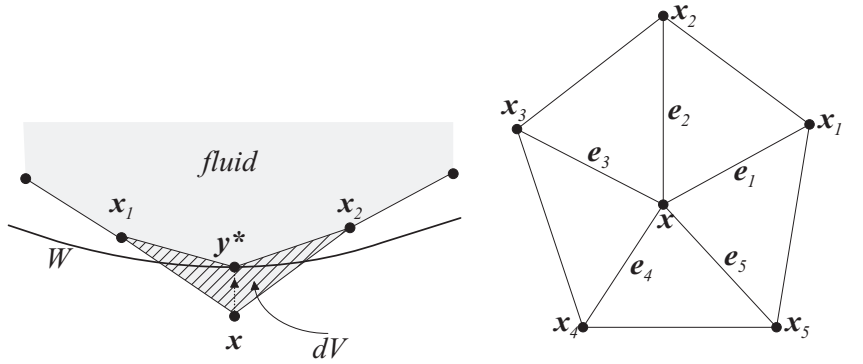
Using the parameterisation of the surface W it is possible to find the value ξ^* such that

$$\mathbf{y}^* := \mathbf{y}(\xi^*) \in W. \quad (\text{B.2})$$

The point \mathbf{y}^* can be considered as the point at W where the fluid particle crossed the wall during its travel from \mathbf{x}^{old} to \mathbf{x} . Therefore we relocate the node \mathbf{x} at the point \mathbf{y}^* . Thus the node has moved over a distance $d\mathbf{x}$ given by

$$d\mathbf{x} := \mathbf{x} - \mathbf{y}^*. \quad (\text{B.3})$$

By moving \mathbf{x} to \mathbf{y}^* at the surface W we lose an amount of fluid, see Figure B.2(a). This volume loss is unwanted and should be accounted for. First we need to exactly compute the volume loss. To this end we need the coordinates of \mathbf{x} , \mathbf{y}^* and their neighbours. Denote the nodes that share an edge of a triangular element with \mathbf{x} by



(a) By moving x back to the surface W a certain amount of fluid is lost (cross-sectional view). (b) The node x is surrounded by m triangular elements and m neighbouring nodes.

Figure B.2: Procedure to relocate x at the wall.

x^1, \dots, x^m , see Figure B.2(b). The volume loss dV due to the displacement $d\mathbf{x}$ is given by [64]

$$dV = \frac{1}{6} \langle d\mathbf{x}, \sum_{j=1}^m \mathbf{e}^j \times \mathbf{e}^{j+1} \rangle. \quad (\text{B.4})$$

Here \mathbf{e}^j is the edge connecting x and x^j , where $\mathbf{e}^{m+1} \equiv \mathbf{e}^1$.

For the blowing problem covered in this thesis the fluid domain is in fact a layer of fluid. For this particular shape we develop a strategy to compensate the volume loss. In Figure B.3(a) we give a schematic overview of the situation. The node x is located at the lower boundary of the fluid and is moved back to the point y^* at W . Let w be the node at the upper boundary that is closest to y^* . We replace w over a distance $d\mathbf{w}$ such that the volume loss is compensated. It is easy to see that $d\mathbf{w}$ has to satisfy

$$-dV = \frac{1}{6} \langle d\mathbf{w}, \sum_{j=1}^m \mathbf{f}^j \times \mathbf{f}^{j+1} \rangle, \quad (\text{B.5})$$

where \mathbf{f}^j is the edge between w and its j -th neighbour. Note that the vector quantity $d\mathbf{w}$ has to satisfy a scalar equation. This means that we have the freedom to choose the direction of $d\mathbf{w}$. We choose $d\mathbf{w} = w_0 \mathbf{n}^w$, where \mathbf{n}^w is the outward normal at w . If we substitute this into the condition for $d\mathbf{w}$ we find

$$w_0 = - \frac{6dV}{\langle \mathbf{n}^w, \sum_{j=1}^m \mathbf{f}^j \times \mathbf{f}^{j+1} \rangle}. \quad (\text{B.6})$$

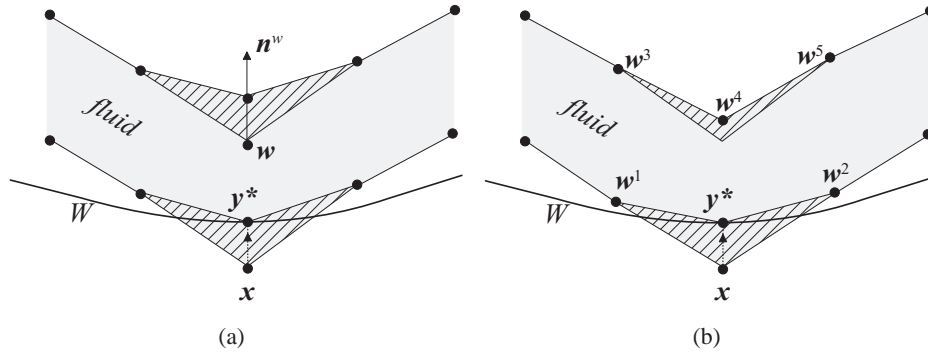


Figure B.3: One or multiple nodes are moved to compensate the volume loss.

In the procedure described above only one node w is relocated to compensate the volume loss caused by relocating x to the wall. The procedure can be improved by relocating several nodes w^1, \dots, w^m , each node compensating a part of the volume loss. Figure B.3(b) illustrates this idea for a 2D setting. The node x has moved through a wall, which is a curved line at a certain height. We distinguish 5 nodes w^i in the neighbourhood of x that are candidates to be relocated. Let α_i be 5 weights that sum to 1. Each of the nodes w^i is relocated over a distance $d w^i$ following the procedure described above, with the difference that the volume loss that w^i has to compensate is multiplied by the weight α_i . In this way the amount of fluid dV is distributed over a larger part of the total fluid, which is physically more correct.

Example B.1

We demonstrate the procedure described in this appendix for a thin layer of fluid that is slightly curved, see Figure B.4. In this example we only use a single node to compensate the volume loss. Located at the right is a straight wall, represented graphically by the light-gray surface. At time $t = t^n$ all nodes at the fluid surface are at the left side of the wall (Figure B.4(a)). A numerical method provides the coordinates of the nodes at the new time level $t = t^{n+1}$, see Figure B.4(b). We observe that a number of nodes has moved through the wall. We correct this by relocating these nodes at appropriate points at the wall. To account for the volume loss, we also relocate a number of nodes that are at the opposite side of the fluid layer. The new situation is shown in Figure B.4(c). We observe that the fluid perfectly touches the wall.

Figure B.5 shows a cross-section of the situation. The solid line represents the fluid layer, for which some nodes move to the right-hand side of the wall. The wall is represented by the vertical dashed line. In the third figure the fluid perfectly touches

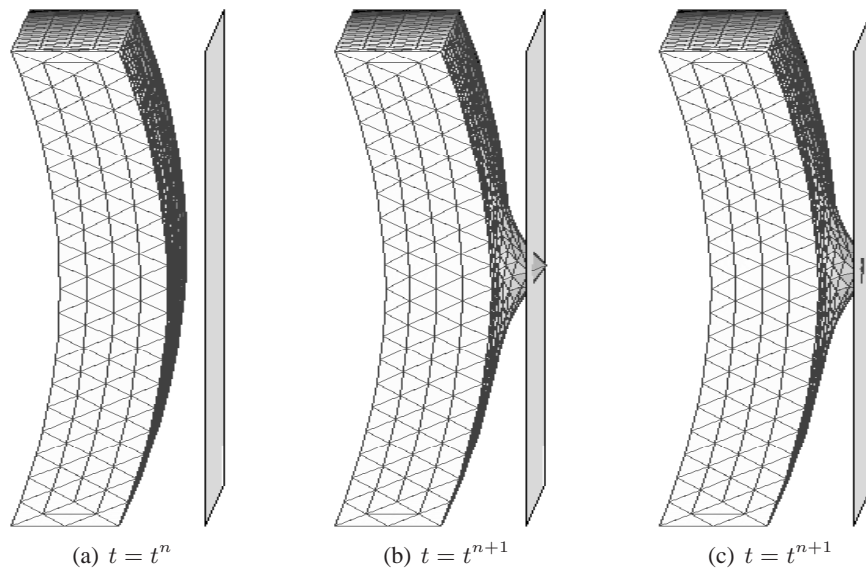


Figure B.4: A fluid layer near a wall (a). At time $t = t^{n+1}$ some nodes of the discretised fluid surface have moved through the wall (b). These nodes are relocated at the wall. Simultaneously some other nodes at the opposite side of the fluid layer are also relocated to compensate the volume loss (c).

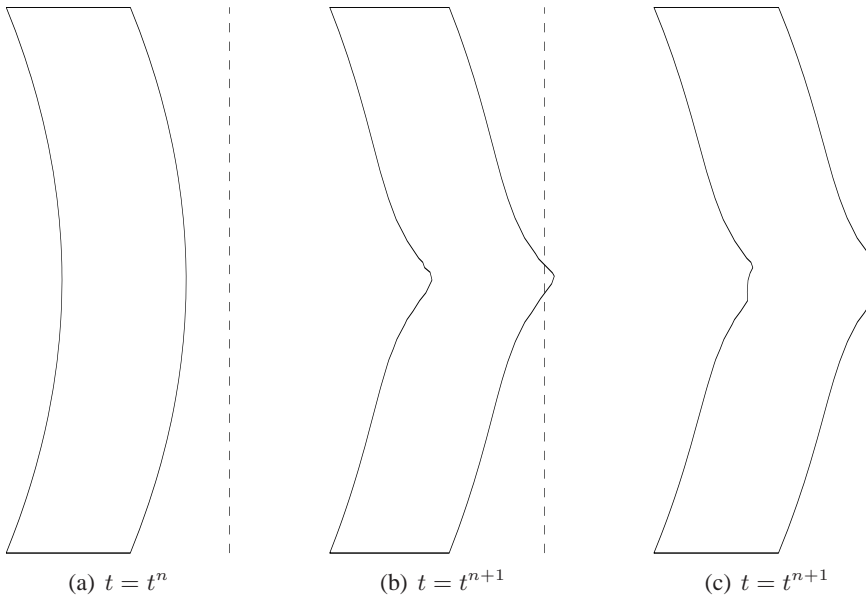


Figure B.5: Cross-sectional view of Figure B.4

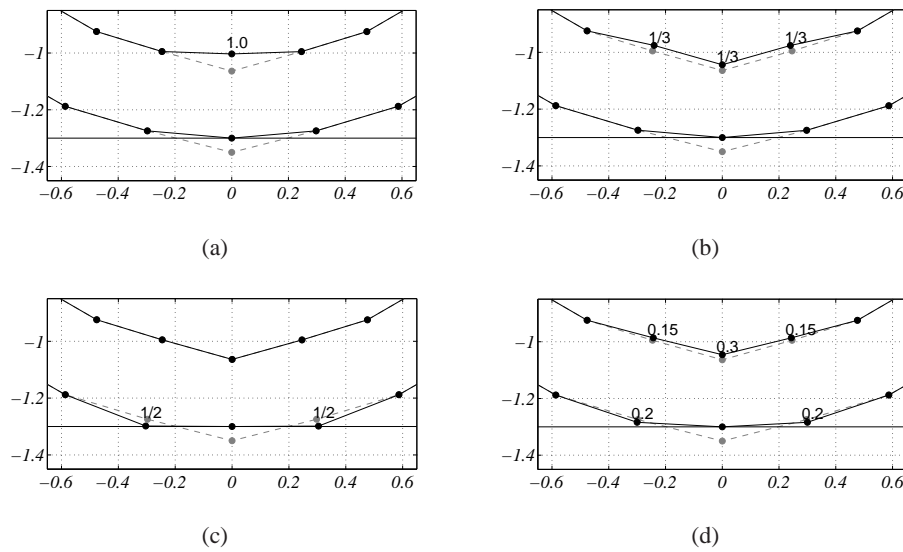


Figure B.6: Several nodes are used to compensate the volume loss caused by relocating x to the wall. The gray dashed line shows the fluid before, the black line the fluid after correction. The numbers near the nodes represent the weights α_i .

the wall. At the same time we see that a number of nodes at the opposite side of the fluid have also been relocated.

When a single node crosses the wall while all its neighbours stay at the other side, it turns out that the procedure described above is volume preserving. In the case that also some neighbours cross the wall, small errors are made. This is caused by the fact that we have to relocate nodes simultaneously, while the nodes share information. For instance, when a node moves over a small distance, the normal vectors at all neighbouring elements alter. Subsequently the normal vectors in all neighbouring nodes also alter. However the errors that are made due to these effects are relatively small. For the current example the volume loss is only $6.3 \cdot 10^{-2}\%$. \square

Example B.2

In this example we demonstrate the strategy to use multiple nodes to compensate the volume loss after relocating nodes at the wall. For a thin layer of fluid in 2D the lowest node has moved through a horizontal line, representing the wall. This node is relocated at the wall. To account for the volume loss we also relocate the five nearest nodes. By varying the weights α_i , we obtain various shapes of the fluid. Figure B.6

present several shapes, each one corresponding to a particular set of weights. The first shape is the result when only the node opposite x is relocated, as we did in the previous example. Apart for this particular choice, the other shapes only have small differences. It is a matter of taste to determine which shape, i.e. which set of weights, is preferable. \square

Appendix C

Smoothing techniques

In this appendix we present a number of smoothing techniques. The smoothing is applied to a triangulated surface directly, or to a scalar function or vector field at this discretised surface.

C.1 Laplacian smoothing

A well-known technique to smooth a triangulated surface is the technique called *Laplacian smoothing* [42, 64, 96]. Let S be a triangulated surface with N nodes and K triangular elements. The idea is that the coordinates of the nodes contain noise due to inaccuracies in the numerical method that determines these coordinates. The noise causes the surface to look unsmooth and irregular. Laplacian smoothing attempts to remove the noise, thus obtaining a smoother surface.

Let $\mathbf{x}^1, \dots, \mathbf{x}^m$ be the m neighbours of a node \mathbf{x} , i.e. the nodes that share a triangular element with \mathbf{x} . For each node \mathbf{x} we compute the geometric average \mathbf{x}_{av} of the neighbouring nodes,

$$\mathbf{x}_{av} := \frac{1}{m} \sum_{i=1}^m \mathbf{x}^i. \quad (\text{C.1})$$

If the node \mathbf{x} is too far away from \mathbf{x}_{av} , it is relocated to a weighted average of \mathbf{x} and \mathbf{x}_{av} ,

$$\mathbf{x} \rightarrow (1 - w)\mathbf{x} + w\mathbf{x}_{av}, \quad (\text{C.2})$$

where w is a suitably chosen weight, $0 \leq w \leq 1$. In this way \mathbf{x} is moved nearer to \mathbf{x}_{av} . If $w = 1$, \mathbf{x} is placed at \mathbf{x}_{av} , if $w = 0$, \mathbf{x} stays at its own place. We can relocate every node at the surface, in which case we apply *global smoothing*.

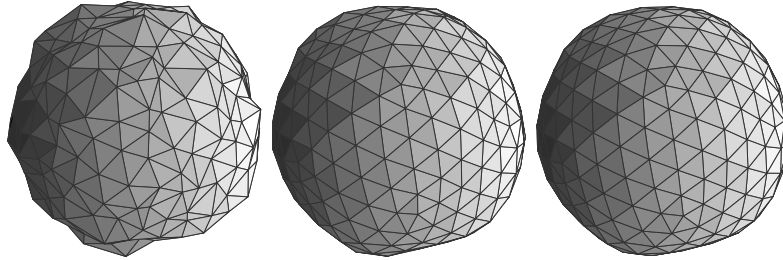


Figure C.1: A triangulated sphere whose coordinates contains noise is smoothed to an almost perfect sphere.

We may also replace x only if the distance to x_{av} exceeds a certain tolerance value. In that case we apply *local smoothing*. In other words, we smooth the surface only at nodes where it is most needed. For both global and local smoothing, the process can be repeated several times. In each iteration the surface gets smoother.

A side-effect of the smoothing is that the volume that the surface encloses decreases. This is a typical disadvantage of standard Laplacian smoothing. There are several modifications to the standard technique to avoid volume loss. The simplest modification is to restrict the movement of the node x to a direction perpendicular to the normal at the surface at x . Unfortunately this reduces the performance of the smoothing. Another possibility is to consider pairs of nodes that are connected by an edge [64]. The two nodes are relocated to new positions simultaneously. In this way we have more freedom to move the nodes to the desired locations, while conserving the volume. In the examples of this section we use the latter modification to Laplacian smoothing.

Example C.1

As an example we consider a triangulated unit sphere, as is shown in Figure C.1. Normally distributed noise is added to the coordinates of the nodes. By applying several iterations of Laplacian smoothing the noise is removed and a smooth surface appears. \square

Example C.2

For the Laplacian smoothing technique three parameters are needed. The weight w , the number of iterations n and the type of smoothing, which can be global or local. In Figure C.2 we let these parameters vary and study the performance of the smoothing. Again we consider the example of the noisy unit sphere. Ideally all nodes are at a

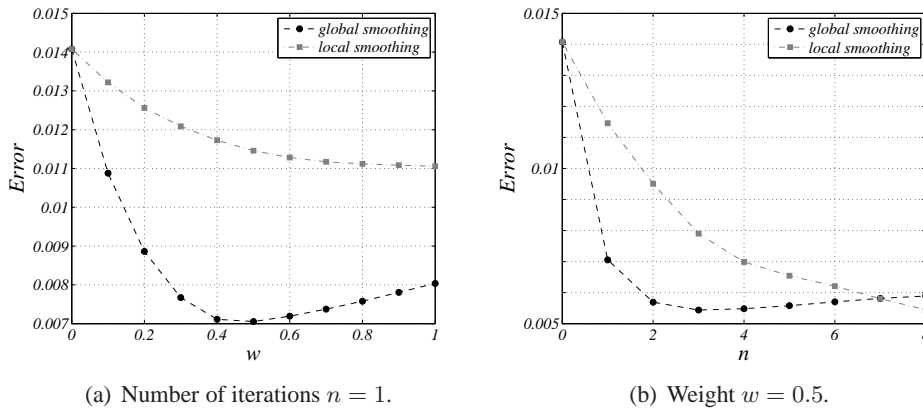


Figure C.2: The error of the discretised surface after Laplacian smoothing. The circles represent global smoothing while the squares represent local smoothing.

distance 1 from the origin. For the noisy sphere this is not the case and we can define the error at a node \mathbf{x} as the difference between $\|\mathbf{x}\|$ and 1.

In Figure C.2(a) we show the mean error as the weight w varies between zero and one. We both apply global smoothing (circles) and local smoothing (squares) and perform one iteration. We observe that global smoothing performs better than local smoothing. For global smoothing the best choice of the weight is $w = 0.5$.

In Figure C.2(b) we show the mean error as the number of iterations n varies between 0 and 8. The weight is kept at $w = 0.5$. For the local smoothing we see that the error decreases as we perform more iterations. However the calculation time grows linearly with the number of iterations. Since the smoothing process is mostly a post-processing step, the computations should not become too time-consuming. For global smoothing there is an optimum of $n = 3$ iterations, which leads to a relatively short computation time. \square

C.2 Smoothing scalar function

We use the technique of Laplacian smoothing to develop a technique to smooth a scalar function at a triangulated surface. Let ϕ be a smooth scalar function at the surface S . We assume that the values of ϕ at the nodes are computed numerically and contain noise. We want to remove this noise such that ϕ becomes smoother. For each node \mathbf{x} we compare the value $\phi(\mathbf{x})$ to the average function value $\phi_{av}(\mathbf{x})$ of the

function at the neighbours of \mathbf{x} ,

$$\phi_{av}(\mathbf{x}) := \frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}^i). \quad (\text{C.3})$$

The value of ϕ at \mathbf{x} is replaced by a weighted average of $\phi(\mathbf{x})$ and $\phi_{av}(\mathbf{x})$,

$$\phi(\mathbf{x}) \rightarrow (1 - w)\phi(\mathbf{x}) + w\phi_{av}(\mathbf{x}). \quad (\text{C.4})$$

If we do this for every node we apply global smoothing. We may also decide to change $\phi(\mathbf{x})$ only if the difference between $\phi(\mathbf{x})$ and $\phi_{av}(\mathbf{x})$ is larger than a certain tolerance value. In that case we apply local smoothing. The whole process can be repeated several times, in each iteration improving the smoothness of the scalar function ϕ .

Example C.3

To demonstrate the smoothing of a scalar function we consider the unit sphere. This time we do not add noise to the coordinates of the nodes of the triangulated sphere, so the sphere is a perfect sphere. As a test function we choose $\phi(\mathbf{x}) = x + y + z$ and add normally distributed noise to the function values at the nodes. In analogy to Laplacian smoothing three parameters exist: the weight w , the number of iterations n and the type of smoothing, which can be global or local. In Figure C.3 we let these parameters vary and study the performance of the smoothing. For this goal we define the following error measure. We consider the vector with exact function values of ϕ and the vector with smoothed function values of ϕ . Then we determine the mean difference between these two vectors.

In Figure C.3(a) we let the weight w vary between zero and one. We apply global smoothing (circles) and local smoothing (squares). As was the case for Laplacian smoothing, global smoothing gives better results than local smoothing. The optimal choice for w in the case of global smoothing is $w = 0.6$. In Figure C.3(b) we let the number of iterations n vary between 0 and 6. We observe that $n = 2$ or $n = 3$ iterations of smoothing yield the smallest error. \square

C.3 Smoothing vector field

In the same way as we smoothed a scalar function at a triangulated surface we may also smooth a vector field at a surface. Let \mathbf{v} be a smooth vector field at the surface S . We assume that for each node \mathbf{x} the vector $\mathbf{v}(\mathbf{x})$ is computed numerically

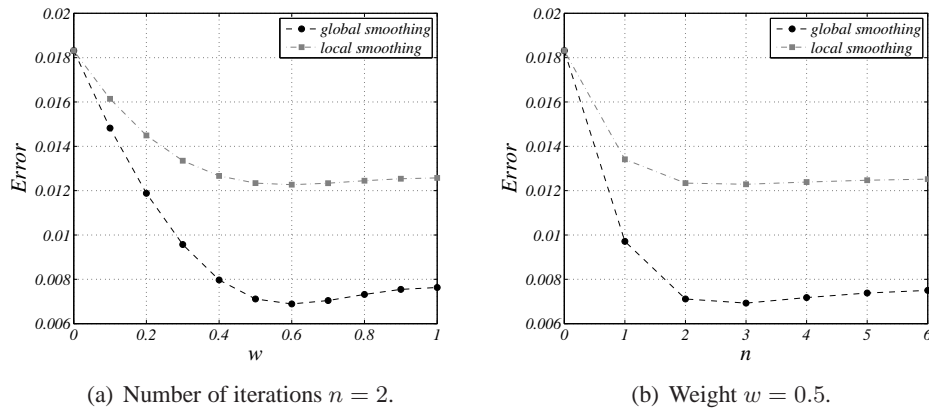
(a) Number of iterations $n = 2$.(b) Weight $w = 0.5$.

Figure C.3: The error in a scalar function at the discretised surface after smoothing. The circles represent global smoothing while the squares represent local smoothing.

and contains noise. We want to remove this noise such that \mathbf{v} becomes smoother. For each node \mathbf{x} we compare the vector $\mathbf{v}(\mathbf{x})$ to the average value $\mathbf{v}_{av}(\mathbf{x})$ of the vectors at the neighbours of \mathbf{x} ,

$$\mathbf{v}_{av}(\mathbf{x}) := \frac{1}{m} \sum_{i=1}^m \mathbf{v}(\mathbf{x}^i). \quad (\text{C.5})$$

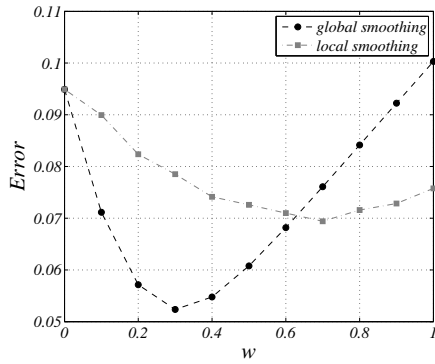
The value of \mathbf{v} at \mathbf{x} is replaced by a weighted average of $\mathbf{v}(\mathbf{x})$ and $\mathbf{v}_{av}(\mathbf{x})$,

$$\mathbf{v}(\mathbf{x}) \rightarrow (1 - w)\mathbf{v}(\mathbf{x}) + w\mathbf{v}_{av}(\mathbf{x}). \quad (\text{C.6})$$

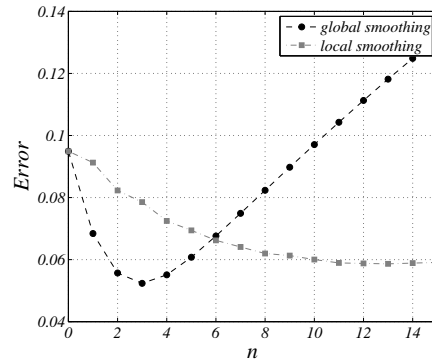
If we do this for all nodes we apply global smoothing. For local smoothing we look at the difference in magnitude and direction between $\mathbf{v}(\mathbf{x})$ and $\mathbf{v}_{av}(\mathbf{x})$. If one of these two is larger than a certain tolerance value, the velocity in \mathbf{x} is replaced by the weighted average. The whole process is repeated n times, in each iteration improving the smoothness of the vector field \mathbf{v} .

Example C.4

As an example we consider the smooth vector field $\mathbf{v} = [x^2, y^2, z^2]$ at the surface of the unit sphere. We add normally distributed noise to the values of \mathbf{v} at the nodes and apply the smoothing technique to obtain a smoother vector field. There exist three parameters: the weight w , the number of iterations n and the type of smoothing, which can be global or local. In Figure C.4 we let these parameters vary and study the performance of the smoothing. For this goal we define an error measure. At each node we take the Euclidean norm of the difference between the exact vector field and the smoothed vector field. Then we take the average of all these norms.



(a) Number of iterations is 3.



(b) Weight is 0.3.

Figure C.4: The reduction of the error at a vector field at the discretised surface due to Laplacian smoothing. The circles represent global smoothing while the squares represent local smoothing.

In Figure C.4(a) we let the weight w run from zero to one and apply global smoothing (circles) and local smoothing (squares). From the graph we see that global smoothing performs better than local smoothing for small weights. If a weight w is chosen larger than 0.6 it is better to use local smoothing. However the best results are obtained for global smoothing with a weight $w = 0.3$. Figure C.4(b) shows the error as a function of the number of iterations. For global smoothing the best choice is $n = 3$ iterations. If local smoothing is performed, $n = 12$ iterations give the lowest error. Performing more iterations does not lower the error. \square

Bibliography

- [1] S. Amini. On boundary integral operators for the Laplace and the Helmholtz equations and their discretisations. *Eng. Anal. Bound. Elem.*, 23:327–337, 1999.
- [2] S. Amini and S.M. Kirkup. Solution of Helmholtz equation in the exterior domain by elementary boundary integral methods. *J. Comput. Phys.*, 118:208–221, 1995.
- [3] D.N. Arnold and W.L. Wendland. On the asymptotic convergence of collocation methods. *Math. Comp.*, 41:349–381, 1983.
- [4] D.N. Arnold and W.L. Wendland. The convergence of spline collocation for strongly elliptic equations on curves. *Numer. Math.*, 47:317–341, 1985.
- [5] R.W. Barnard, K. Pearce, and A.Y. Solynin. Area, width, and logarithmic capacity of convex sets. *Pacific J. Math.*, 212:13–23, 2003.
- [6] A.A. Becker. *The Boundary Element Method in Engineering*. McGraw-Hill, Maidenhead, 1992.
- [7] R.C.G. Beerkens et al. *NCNG Handbook for Glass Technologists*. 1997.
- [8] A. Björck and G.H. Golub. Numerical methods for computing angles between linear subspaces. *Math. Comp.*, 27:579–594, 1973.
- [9] C.A. Brebbia. *The boundary element method for engineers*. Pentech Press, London, 1978.
- [10] C.A. Brebbia, J.C.F. Telles, and L.C. Wrobel. *The boundary element method in engineering*. McGraw-Hill Book Company, London, 1984.
- [11] J.C. Butcher. *The numerical analysis of ordinary differential equations*. Wiley, Chichester, 1987.
- [12] J.M.A. César de Sá. Numerical modelling of glass forming processes. *Eng. Comput.*, 3:266–275, 1986.

- [13] J.M.A. César de Sá, R.M. Natal Jorge, C.M.C. Silva, and R.P.R. Cardoso. A computational model for glass container forming processes. In *Europe Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering*, 1999.
- [14] G. Chen and J. Zhou. *Boundary Element Methods*. Academic Press, London, 1992.
- [15] J.T. Chen and Y.P. Chiu. On the pseudo-differential operators in the dual boundary integral equations using degenerate kernels and circulants. *Eng. Anal. Bound. Elem.*, 26:41–53, 2002.
- [16] J.T. Chen, S.R. Kuo, and J.H. Lin. Analytical study and numerical experiments for degenerate scale problems in the boundary element method for two-dimensional elasticity. *Internat. J. Numer. Methods Engrg.*, 26:1669–1681, 2002.
- [17] J.T. Chen, C.F. Lee, I.L. Chen, and J.H. Lin. An alternative method for degenerate scale problems in boundary element methods for the two-dimensional Laplace equation. *Eng. Anal. Bound. Elem.*, 26:559–569, 2002.
- [18] J.T. Chen, J.H. Lin, S.R. Kuo, and Y.P. Chiu. Analytical study and numerical experiments for degenerate scale problems in boundary element method using degenerate kernels and circulants. *Eng. Anal. Bound. Elem.*, 25:819–828, 2001.
- [19] S. Christiansen. Condition number of matrices derived from two classes of integral equations. *Mat. Meth. Appl. Sci.*, 3:364–392, 1981.
- [20] S. Christiansen. On two methods for elimination of non-unique solutions of an integral equation with logarithmic kernel. *Appl. Anal.*, 13:1–18, 1982.
- [21] S. Christiansen. Modifications of some first kind integral equations with logarithmic kernel to improve numerical conditioning. *Computing*, 34:221–242, 1985.
- [22] S. Christiansen and P.C. Hansen. The effective condition number applied to error analysis of certain boundary collocation methods. *J. Comp. Appl. Math.*, 54:15–35, 1994.
- [23] S. Christiansen and P. Hougaard. An investigation of a pair of integral equations for the biharmonic problem. *J. Inst. Maths. Applics*, 22:15–27, 1978.
- [24] S. Christiansen and J. Saranen. The conditioning of some numerical methods for the first kind boundary integral equations. *J. Comp. Appl. Math.*, 67:43–58, 1996.

-
- [25] C. Constanda. *A mathematical analysis of bending plates with transverse shear deformation*. Longman, Harlow, and Wiley, New York, 1990.
- [26] C. Constanda. On the solution of the Dirichlet problem for the two-dimensional Laplace equation. *Proc. Amer. Math. Soc.*, 119(3):877–884, 1993.
- [27] C. Constanda. On non-unique solutions of weakly singular integral equations in plane elasticity. *Quart. J. Mech. Appl. Math.*, 47:261–268, 1994.
- [28] C. Constanda. On the Dirichlet problem for the two-dimensional biharmonic equation. *Mat. Meth. Appl. Sci.*, 20:885–890, 1997.
- [29] M. Costabel. Boundary integral operators on Lipschitz domains: elementary results. *SIAM J. Math. Anal.*, 19(3):613–626, 1988.
- [30] M. Costabel and M. Dauge. On invertibility of the biharmonic single-layer potential operator. *Integr. Equ. Oper. Theory*, 24:46–67, 1996.
- [31] T.A. Cruse. Numerical solutions in three dimensional elastostatics. *Int. J. Solids Structures*, 5:1259–1274, 1969.
- [32] V. Cutanda, P.M. Juhl, and F. Jacobsen. On the modelling of narrow gaps using the standard boundary element method. *J. Acoust. Soc. Am.*, 109:1296–1303, 2001.
- [33] J. Davis. *Circulant Matrices*. John Wiley and Sons, New York, 1979.
- [34] P.J. Davis and P. Rabinowitz. *Numerical Integration*. Blaisdell Publishing Company, Waltham, 1967.
- [35] W. Dijkstra and R.M.M. Mattheij. Condition number of the BEM matrix arising from the Stokes equations in 2D. *Eng. Anal. Bound. Elem.* in press.
- [36] W. Dijkstra and R.M.M. Mattheij. Numerical modelling of the blowing phase in the production of glass containers. *Electron. J. Bound. Elem.* in press.
- [37] W. Dijkstra and R.M.M. Mattheij. Estimating the condition number of the matrices appearing in the boundary element method. In A.P. Selvadurai, C.L. Tan, and M.H. Aliabadi, editors, *Advances in Boundary Element Techniques VI*, pages 359–364, 2005.
- [38] W. Dijkstra and R.M.M. Mattheij. The condition number of the BEM-matrix arising from Laplace’s equation. *Electron. J. Bound. Elem.*, 4(2):67–81, 2006.

- [39] W. Dijkstra and R.M.M. Mattheij. A relation between the logarithmic capacity and the condition number of the BEM-matrices. *Comm. Numer. Methods Engrg.*, 23(7):665–680, 2007.
- [40] W. Dijkstra and R.M.M. Mattheij. Simulating the blowing of glass containers using the boundary element method. In V. Minutolo and M.H. Aliabadi, editors, *Advances in Boundary Element Techniques VIII*, pages 105–110, 2007.
- [41] V. Domínguez and F.J. Sayas. A BEM-FEM overlapping algorithm for the Stokes equation. *Appl. Math. Comput.*, 182:691–710, 2006.
- [42] D. Field. Laplacian smoothing and Delaunay triangulations. *Comm. Appl. Numer. Methods*, 4(6):709–712, 1988.
- [43] I. Fredholm. Sur une classe d'équations fonctionnelles. *Acta Numer.*, 27:365–390, 1903.
- [44] B. Fuglede. On a direct method of integral equations for solving the biharmonic Dirichlet problem. *ZAMM*, 61:449–459, 1981.
- [45] C.G. Giannopapa. Development of a computer simulation model for blowing glass containers. *ASME J. Manuf. Sci. Eng.*, 2008. in press.
- [46] I. Gohberg and S. Goldberg. *Basic Operator Theory*. Birkhäuser, Boston, 1981.
- [47] I.C. Gohberg and M.G. Kreĭn. *Introduction to the Theory of Linear Nonselfadjoint Operators*. Nauka, Moscow, 1969.
- [48] K. Gustafson. The geometrical meaning of the Kantorovich-Wielandt inequalities. *Lin. Alg. Appl.*, 296:143–151, 1999.
- [49] W. Hackbusch and S.A. Sauter. On numerical cubatures of nearly singular surface integrals arising in BEM collocation. *Computing*, 52:139–1994, 1994.
- [50] W. He, H. Ding, and H. Hu. A necessary and sufficient boundary integral formulation for plane elasticity problems. *Comm. Numer. Methods Engrg.*, 12:413–424, 1996.
- [51] U. Heise. Dependence of the round-off error in the solution of boundary integral equations on a geometrical scale factor. *Comput. Methods Appl. Mech. Engrg.*, 62:115–126, 1987.
- [52] E. Hille. *Analytic Function Theory: vol II*. Ginn, London, 1959.

- [53] G.C. Hsiao. On the stability of integral equations of the first kind with logarithmic kernels. *Arch. Ration. Mech. Anal.*, 94:179–192, 1986.
- [54] G.C. Hsiao and W.L. Wendland. A finite element method for some integral equations of the first kind. *J. Math. Anal. Appl.*, 58:449–481, 1979.
- [55] M.A. Jaswon. Integral equation methods in potential theory - i, Series A. *Proc. Roy. Soc. Lond.*, 275:23–32, 1963.
- [56] M.A. Jaswon and G. Symm. *Integral Equation Methods in Potential Theory and Elastostatics*. Academic Press, London, 1977.
- [57] V. John and A. Liakos. Time-dependent flow across a step: the slip with friction boundary condition. *Int. J. Numer. Meth. Fluids*, 50:713–731, 2006.
- [58] O.D. Kellogg. *Foundations of Potential Theory*. Springer, Berlin, 1929.
- [59] S.M. Kirkup. The influence of the weighting parameter on the improved boundary element solution of the exterior Helmholtz equation. *Wave Motion*, 93:93–101, 1992.
- [60] R. Kress. Minimizing the condition number of boundary integral operators in acoustics and electromagnetic scattering. *Quart. J. Mech. Appl. Math.*, 38:323–341, 1985.
- [61] R. Kress and W.T. Spassov. On the condition number of boundary integral operators for the exterior Dirichlet problem for the Helmholtz equation. *Numer. Math.*, 42:77–85, 1983.
- [62] G. Kuhn, G. Löbel, and I. Potrč. Kritisches Lösungsverhalten der direkten Randelementmethode bei logarithmischem Kern. *Z. angew. Math. Mech.*, 67:361–363, 1987.
- [63] V.D. Kupradze. *Potential Methods in Theory of Elasticity*. Daniel Davy, New York, 1965.
- [64] A. Kuprat, A. Khamayseh, D. George, and L. Larkey. Volume conserving smoothing for piecewise linear curves, surfaces and triple lines. *J. of Comp. Phys.*, 172:99–118, 2001.
- [65] O.A. Ladyzhenskaya. *The Mathematical Theory of Viscous Incompressible Flow*. Gordon and Beach, New York-London, 1963.
- [66] N.S. Landkof. *Foundations of Modern Potential Theory*. Springer Verlag, Berlin, 1972.

- [67] U. Lange. The use of remeshing methods in pressing simulations. In D. Krause and H. Loch, editors, *Mathematical Simulation in Glass Technology*, pages 317–325. Springer, 2002.
- [68] C. Marechal, P. Moreau, and D. Locheignies. Numerical optimization of a new robotized glass blowing process. *Engineering with Computers*, 19:233–240, 2004.
- [69] R. Martinez. The thin-shape breakdown (tsb) of the Helmholtz integral equation. *J. Acoust. Soc. Am.*, 90:2728–2738, 1991.
- [70] W. McLean. *Strongly elliptic systems and boundary integral equations*. Cambridge University Press, Cambridge, 2000.
- [71] W. McLean and T. Tran. A preconditioning strategy for boundary element Galerkin methods. *Numer. Methods Partial Differential Eq.*, 13:283–301, 1997.
- [72] J.L. Migeot. BEM-condition number of systems obtained when solving mixed boundary value potential problems. *Commun. Appl. Numer. Methods*, 2:429–435, 1986.
- [73] S.D. Mikhlin. *Integral Equations*. Pergamon Press, Oxford, 1957.
- [74] N.I. Muskhelishvili. *Some Basic Problems of Mathematical Theory of Elasticity*. Noordhoff, Holland, 1953.
- [75] H. Power and L.C. Wrobel. *Boundary integral methods in fluid mechanics*. Computational Mechanics Publications, Southampton, 1995.
- [76] C. Pozrikidis. *Boundary integral and singularity methods for linearized viscous flows*. Cambridge University Press, Cambridge, 1992.
- [77] C. Pozrikidis. *A practical guide to boundary element methods*. Chapman and Hall, Boca Raton, 2002.
- [78] A.R.M. Primo, L.C. Wrobel, and H. Power. An indirect boundary-element method for slow viscous flow in a bounded region containig air bubbles. *J. Engrg. Math.*, 37:305–326, 2000.
- [79] A.R.M. Primo, L.C. Wrobel, and H. Power. Low Reynolds number deformation of viscous drops in a bounded flow region under surface tension. *Math. Comput. Modelling*, 31:99–118, 2000.
- [80] F.J. Rizzo. An integral equation approach to boundary-value problems of classical elastostatics. *Q. Appl. Math.*, 25:83–95, 1967.

- [81] G.J. Rodin and O. Steinbach. Boundary element preconditioners for problems defined on slender domains. *SIAM J. Sci. Comput.*, 24:1450–1464, 2003.
- [82] J. Rostand. Computing logarithmic capacity with linear programming. *Experiment. Math.*, 6:221–238, 1997.
- [83] J. Saranen and W.L. Wendland. On the asymptotic convergence of collocation methods with spline functions of even degree. *Math. Comp.*, 45:91–108, 1985.
- [84] V. Sladek, J. Sladek, and M. Tanaka. Numerical integration of logarithmic and nearly logarithmic singularity in BEMs. *Appl. Math. Modelling*, 25:901–922, 2001.
- [85] I.H. Sloan and A. Spence. The Galerkin method for integral equations of the first kind with logarithmic kernel: theory. *IMA J. Numer. Anal.*, 8:105–122, 1988.
- [86] C. Somigliana. Sopra l’equilibrio di un corpo elastico isotrope. *Il Nuovo Cimento, serie III*, pages 181–185, 1886.
- [87] O. Steinbach and W.L. Wendland. The construction of some efficient preconditioners in the boundary element method. *Adv. in Comp. Math.*, 9:191–216, 1998.
- [88] A.H. Stroud and D. Secrest. *Gaussian quadrature formulas*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [89] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin. A comparison of Gaussian and mean curvatures estimation methods on triangular meshes. *Proc. Int. Conf. on Robotics and Automation*, pages 1021–1026, 2003.
- [90] G.T. Symm. Integral equation methods in potential theory - ii, Series A. *Proc. Roy. Soc. Lond.*, 275:33–46, 1963.
- [91] M. Tsuji. *Potential Theory in Modern Function Theory*. Maruzen, Tokyo, 1959.
- [92] A.M. Turing. Rounding-off errors in matrix processes. *Quart. J. Mech. Appl. Math.*, 1:287–308, 1948.
- [93] G.A.L. Vorst van der, R.M.M. Mattheij, and H.K. Kuiken. Boundary element solution for two-dimensional viscous sintering. *J. Comput. Phys.*, 100:50–63, 1992.
- [94] G.A.L. van der Vorst. *Modelling and numerical simulation of viscous sintering*. PhD thesis, Eindhoven University of Technology, 1994.

- [95] R. Vodička and V. Mantič. On invertibility of elastic single-layer potential operator. *J. Elasticity*, 74:147–173, 2004.
- [96] J. Vollmer, R. Mencl, and H. Müller. Improved Laplacian smoothing of noisy surface meshes. *Computer Graphics Forum*, 18(3):131–138, 1999.
- [97] W.L. Wendland and S. Christiansen. On the condition number of the influence matrix belonging to some first kind integral equations with logarithmic kernel. *Appl. Anal.*, 21:175–183, 1986.
- [98] Y. Yan and I.H. Sloan. On integral equations of the first kind with logarithmic kernels. *J. Integral Equations Appl.*, 1:549–579, 1988.
- [99] G. Zukav. *The Dancing W Li Masters: an Overview of the New Physics*, page 208. William Morrow and Company, New York, 1979.

Index

- \mathcal{G} , 67
- \mathcal{H} , 67
- \mathcal{K}^d , 15, 49
- \mathcal{K}^s , 15, 49
- \mathbf{A} , 19
- \mathbf{G} , 18, 107
- \mathbf{H} , 18, 107

- adaptive Simpson, 89
- algebraic equations, 17
- angle
 - between subspaces, 37
 - Kantorovich-Wielandt, 37

- beam, 123
- biharmonic equation, 48
- blowing phase, 95
- boundary element, 17
- boundary element method, 2, 10
- boundary elements, 106
- boundary integral equation, 6, 49, 66, 103
- boundary value problem, 10

- capacity
 - logarithmic, 47, 50
- circle, 24, 67
- circulant, 34
- collocation, 8
- collocation node, 17
- compact, 16
- condition number, 4, 29
 - local, 5

- M-, 5
- N-, 5
- contact problem, 142
- critical
 - boundary, 69
 - domain, 7, 59, 69
 - scaling, 7, 58, 69
- critical domain, 7, 59
- critical scaling, 7, 58, 69
- curvature
 - Gaussian, 134
 - mean, 82, 97, 134
 - principal, 134

- decomposition, 35
- decoupled equations, 43
- deflated operator, 106
- direct formulation, 8
- Dirichlet boundary conditions, 10
- Dirichlet problem, 49, 51
- double layer potential, 15, 24, 49, 67, 105
- dynamic viscosity, 96

- Eckert number, 102
- eigenvalues, 16, 24, 27
 - of \mathcal{K}^d , 24
 - of \mathcal{K}^s , 24
 - of \mathbf{G} , 27
 - of \mathbf{H} , 27
- elastostatic problem, 6
- element, 17
- elements, 106
- ellipsoid, 126, 140

- energy equation, 101
 Euler forward, 82, 109
- Fourier matrix, 34
 Fredholm, 16
 friction, 131
 friction parameter, 98
 Froude number, 99
 fundamental solution, 11, 49, 103
 modified, 11, 31
- Galerkin, 8
 Gauss-Legendre, 22, 89
 Gaussian curvature, 134
 global smoothing, 149
 gravity, 131
 Green's identity, 11, 103
- Helmholtz equation, 48
 Heun's method, 109
- ill-conditioned, 4
 improved Euler method, 109
- Jacobian, 20
- Kantorovich-Wielandt angle, 37
- knots
 of Gauss-Legendre scheme, 22
- Laplace
 equation, 10, 49
 operator, 11
 Laplace equation, 10, 49
 Laplacian smoothing, 109, 127, 149
 local condition number, 5
 local smoothing, 150
 logarithmic capacity, 47, 50
- M-condition number, 5
 mean curvature, 82, 97, 134
 mixed problem, 49, 53
- mould, 111
- N-condition number, 5
 Navier's slip law, 98
 Navier-Stokes equations, 97
 Neumann, 10
 Neumann problem, 49, 55
 node, 17
 normal stress, 66
 normal vector, 135
- Péclet number, 102
 parabolic fit method, 134
 parison, 94, 111
 pressing phase, 95
 principal curvature, 134
 projection operator, 105
- QR-decomposition, 35
- Reynolds number, 99
 rigid body motions, 105
 rotation matrix, 136
- sagging, 95, 113
 self-adjoint, 16
 Simpson
 adaptive, 89
 single layer potential, 15, 24, 49, 67, 105
 slip condition, 98
 smoothing
 Laplacian, 149
 scalar function, 151
 vector field, 152
 spectral theorem, 16
 sphere, 137
 Stokes equations, 6, 66, 99
 stress tensor, 66, 97
 surface tension, 81, 96, 129
- thin-shape breakdown, 23, 86
 triangular elements, 106

velocity smoothing, 109

viscosity, 96, 101

viscous drop, 81, 123

weights

 of Gauss-Legendre scheme, 22

well-conditioned, 4

Summary

The boundary element method (BEM) is an efficient numerical method that approximates solutions of various boundary value problems. Despite its success little research has been performed on the conditioning of the linear systems that appear in the BEM.

For a Laplace equation with Dirichlet boundary conditions a remarkable phenomenon is observed; the corresponding boundary integral equation (BIE) is singular for a certain critical size of the 2D domain. As a consequence the discrete counterpart of the BIE, the linear system, is singular too, or at least ill-conditioned. This is reflected by the condition number of the system matrix, which is infinitely large, or at least very large. When the condition number of the BEM-matrix is large, the linear system is difficult to solve and the solution of the system is very sensible to perturbations in the boundary data.

For a Laplace equation with mixed boundary conditions a similar phenomenon is observed. The corresponding BEM-matrix consists of two blocks; one block originates from the BEM-matrix belonging to the Dirichlet problem, the other block originates from the BEM-matrix belonging to the Neumann problem. The composite matrix inherits the solvability problems from the Dirichlet block. In other words, for the Laplace equation with mixed boundary conditions there exists also a critical size of the 2D domain for which the BEM-matrix has an infinitely large condition number. Hence the size and shape of the domain affects the solvability of the BEM problem.

The critical size of the domain for which the BIE becomes singular is related to the logarithmic capacity of the domain. The logarithmic capacity is a positive real number that is a function of the size and shape of the domain. If this logarithmic capacity is equal to one, the domain is a critical domain, and for this domain the BIE becomes singular. Thus by computing the logarithmic capacity we can *a-priori* determine whether the BIE will be singular or not. The logarithmic capacity depends linearly on the scale of the domain, and thus a domain with logarithmic capacity equal to one can always be found by rescaling the domain. Unfortunately the logarithmic capacity can only be computed analytically for a few simple domains; for more involved domains the logarithmic capacity can be estimated though.

There are several possibilities to avoid large condition numbers, i.e. singular BIEs that appear at critical domains. The first option is to rescale the domain such that the logarithmic capacity is unequal to one. One can also add a supplementary condition to the BIE and the linear system. A drawback of this option is that the linear system has more equations than unknowns and different techniques are required to solve the system. A third option is to slightly modify the fundamental solution of the Laplace operator. This fundamental solution directly appears in the BIE and it can be shown that a suitable modification yields BIEs that do not become singular.

The critical domains for which the BIEs become singular do not restrict to Laplace equations only. Also for BIEs applied to the biharmonic equation or the elastostatic equations and the Stokes equations such critical domains exist. As the last two equations are vectorial equations, also the corresponding BIE consists of two equations. As a consequence two critical domains can be found for which these BIEs become singular. To obtain nonsingular BIEs techniques similar to the Laplace case can be used. Unfortunately we cannot *a-priori* determine the sizes for which the BIEs becomes singular, and thus do not know to what size we should rescale the domain to obtain nonsingular BIEs.

The existence of critical domains is in essence caused by the logarithmic term in the fundamental solutions for the elliptic boundary value problems in 2D. This logarithmic term does not depend linearly on the size of the domain. When a domain is scaled, i.e. multiplied by a scale factor, the argument of the logarithm is also multiplied by this scale factor, but the logarithm turns this into an additive term. Thus the logarithm transforms multiplication into addition. This affects the BIEs in such a way that critical domains can appear. The fundamental solutions of boundary value problems in 3D do not contain a logarithmic term. Hence scaling of the domain does not affect the fundamental solution, and consequently also the BIE is not affected. Hence we may safely rescale 3D domains without the risk to encounter a critical domain.

An example in which a domain takes many different sizes and shapes is the blowing problem. In this problem a viscous fluid is blown to a desired shape. Typically the time is discretised into a set of discrete time steps, and at each step the shape of the fluid is computed by solving the Stokes equations. When attempting to simulate this problem in 2D, we meet a large number of 2D domains, and we risk that one of these domains is equal to or approaches a critical domain. In such a case the BEM will have difficulties with solving the Stokes equations for that particular domain.

When simulating the blowing problem in 3D, no critical domains are encountered. It turns out that the BEM is a very efficient numerical method for this particular 3D problem with a moving boundary. As we are merely interested in the shape of the fluid, we only need to know the flow of its boundary. The BEM does

exactly that; it does not compute the flow at the interior of the fluid. Furthermore it is rather easy to include other effects from the blowing problem in the model, such as gravity, surface tension and friction from the contact of the fluid with a wall. As only the boundary of the fluid is discretised, the system matrices that appear in the BEM are smaller than the system matrices that appear when solving the problem with a finite element method, for example. Though the BEM-matrices are dense, while the finite element matrices are sparse, the computational effort for the BEM is relatively low. In short, the BEM is a very appropriate numerical method when solving blowing problems.

Samenvatting

De boundary element method (BEM) is een efficiënte numerieke methode om oplossingen van randwaardeproblemen te benaderen. Ondanks haar succes is er weinig onderzoek gepleegd naar het goed of slecht geconditioneerd zijn van de stelsels van lineaire vergelijkingen die voorkomen in de BEM.

Voor de Laplace vergelijking met Dirichlet randvoorwaarden is een opmerkelijk verschijnsel geobserveerd; the bijbehorende randintegraalvergelijking (BIE) is singulier voor een bepaalde kritieke grootte van het 2D gebied. Daardoor is de discrete versie van de BIE, het stelsel van lineaire vergelijkingen, ook singulier, of tenminste slecht geconditioneerd. Dit wordt zichtbaar in het conditiegetal van de systeemmatrix, welke oneindig groot is, of tenminste heel erg groot. Als het conditiegetal van de BEM-matrix groot is, is het stelsel vergelijkingen moeilijk op te lossen en ook is de oplossing van het stelsel gevoelig voor perturbaties in de randvoorwaarden.

Voor een Laplace vergelijking in 2D met gemengde randvoorwaarden wordt een vergelijkbaar verschijnsel geobserveerd. The bijbehorende BEM-matrix bestaat uit twee blokken; een blok uit de BEM-matrix behorende bij het Dirichlet probleem, en een blok uit de BEM-matrix behorende bij het Neumann probleem. De samengestelde matrix erft de oplosbaarheidsproblemen van het Dirichlet blok. Daardoor bestaat er voor de Laplace vergelijking met gemengde randvoorwaarden ook een kritieke grootte van het gebied waarvoor het conditiegetal van de BEM-matrix oneindig groot wordt. Met andere woorden, de grootte van het gebied beïnvloedt de oplosbaarheid van het BEM probleem.

De kritieke grootte van een gebied waarvoor de BIE singulier wordt is gerelateerd aan de logaritmische capaciteit van het gebied. De logaritmische capaciteit is een positief reëel getal dat een functie is van de grootte en de vorm van het gebied. Als deze logaritmische capaciteit gelijk is aan één, dan is het gebied een kritiek gebied, en is de BIE singulier. Door de logaritmische capaciteit te berekenen kunnen we *a-priori* bepalen of een BIE singulier is. De logaritmische capaciteit hangt lineair af van de schaal van een gebied, en een gebied met logaritmische capaciteit gelijk aan één kan dus altijd gevonden worden door het gebied te herschalen. Helaas

kan de logaritmische capaciteit alleen analytisch worden uitgerekend voor een paar eenvoudige gebieden; voor ingewikkeldere gebieden kan de logaritmische capaciteit worden geschat.

Er zijn verschillende mogelijkheden om grote conditiegetallen, i.e. singuliere BIEs als gevolg van kritieke gebieden, te voorkomen. De eerste optie is het gebied te herschalen zodanig dat de logaritmische capaciteit ongelijk is aan één. Men kan ook een extra vergelijking toevoegen aan de BIE en het stelsel lineaire vergelijkingen. Een nadeel hiervan is dat we een stelsel krijgen met meer vergelijkingen dan onbekenden, en we hebben andere technieken nodig om deze stelsels op te lossen. Een derde optie is de fundamenteeloplossing van de Laplace operator aan te passen. Deze fundamenteeloplossing komt voor in de BIE en het kan aangetoond worden dat een geschikte aanpassing tot BIEs leidt die niet meer singulier zijn.

De kritieke gebieden waarvoor de BIEs singulier zijn beperken zich niet tot de Laplace vergelijking. Ook voor BIEs voor de biharmonische vergelijking of de elastostatische vergelijkingen en de Stokes vergelijkingen bestaan zulke kritieke gebieden. Aangezien de laatste twee vergelijkingen vectoriële vergelijkingen zijn, bestaat de bijbehorende BIE ook uit twee vergelijkingen. Dientengevolge kunnen er ook twee kritieke gebieden gevonden worden waarvoor deze BIEs singulier zijn. Om niet-singuliere BIEs te verkrijgen kunnen vergelijkbare methodes gebruikt worden als voor het geval van de Laplace vergelijking. Helaas kunnen we niet *a-priori* bepalen voor welke kritieke gebieden de BIEs singulier zijn, en dus weten we ook niet hoe we de gebieden moeten herschalen om niet-singuliere BIEs te verkrijgen.

Het verschijnsel van kritieke gebieden wordt in essentie veroorzaakt door de aanwezigheid van een logaritmische term in de fundamenteeloplossing voor elliptische randwaardeproblemen in 2D. Deze logaritmische term hangt niet-linear af van de grootte van het gebied. Als een gebied wordt geschaald, i.e. vermenigvuldigd met een schaalfactor, dan wordt het argument van de logaritme ook vermenigvuldigd met deze schaalfactor, maar de logaritme verandert dit in een optelling. Op deze manier verandert de logaritme een vermenigvuldiging in een optelling. Hierdoor worden de vergelijkingen dusdanig veranderd dat verschijnselen als kritieke gebieden kunnen optreden. De fundamenteeloplossing voor randwaardeproblemen in 3D bevat geen logaritmische term. Een herschaling van het gebied heeft daardoor geen significante invloed op de fundamenteeloplossing, en ook de BIE wordt niet essentieel beïnvloed. Hierdoor kunnen we een 3D gebied herschalen zonder het risico te lopen op een kritiek gebied te stuiten.

Een voorbeeld waarin een gebied veel verschillende groottes en vormen aanneemt is het blaasprobleem. In het blaasprobleem wordt een visceuze vloeistof in een gewenste vorm geblazen. De tijd wordt vaak gediscretiseerd in een aantal discrete tijdstappen, en in elke stap kan de vorm van de vloeistof berekend worden door de Stokes vergelijkingen op te lossen. Als we dit probleem in 2D simuleren doorlopen

we een groot aantal gebieden, en we lopen het risico dat we op een kritiek gebied stuiten of op een gebied dat bijna kritiek is. In zo'n geval zal de BEM moeite hebben om de Stokes vergelijkingen op te lossen.

Als we het blaasprobleem simuleren in 3D komen we geen kritieke gebieden tegen. Het blijkt dat de BEM een zeer efficiënte numerieke methode is voor dit type probleem in 3D met een bewegende rand. Omdat we eigenlijk alleen geïnteresseerd zijn in de vorm van de vloeistof, hoeven we alleen de stroming van de rand te weten. De BEM doet dit precies; ze berekent de stroming in het inwendige niet. Verder is het gemakkelijk om andere aspecten van het blaasprobleem toe te voegen aan het model, zoals zwaartekracht, oppervlaktespanning en wrijving als gevolg van contact tussen vloeistof en een wand. Omdat alleen de rand van de vloeistof gediscretiseerd wordt, zijn de systeemmatrices die voorkomen in de BEM veel kleiner dan de matrices die voorkomen in bijvoorbeeld de eindige elementen methode. Ook al zijn de BEM-matrices vol, terwijl eindige elementen matrices ijl zijn, de rekestijd voor de BEM is relatief kort. Kortom, de BEM is een uitermate geschikte numerieke methode om blaasproblemen te simuleren.

Acknowledgements

During my four years as a PhD student several people have supported me. These people have, each in his own way, contributed to the realization of this thesis. First of all I would like to express my gratitude towards my promotor prof. dr. Bob Mattheij for giving me the opportunity to work on a challenging and interesting mathematical subject for four years. He has guided and inspired me greatly, while at the same time allowing me a high level of independence. My copromotor dr. ir. Michiel Hochstenbach was very valuable in discussing technical aspects of my work, especially during the last year of my thesis. I am also greatly indebted towards dr. ir. Bas van der Linden who assisted me in programming my simulation tool, and also helped me on several other computational issues. I would like to thank prof. dr. Jan de Graaf for several fruitful discussions on the concept of logarithmic capacity.

I really enjoyed working in the CASA group at the Eindhoven University of Technology. I highly esteem the open atmosphere, the healthy working spirit, and all the agreeable social events. I would like to thank all CASA-members for making this possible. A special word of gratitude goes towards the colleagues with whom I shared the office. Bratislav, Irina, Luiza, Ali, and Maxim; thank you for the sound working environment during all these years. As one cannot live on mathematics only, I would like to thank Bas, Luiza, Ronald, Martijn and others for the numerous coffee and lunch breaks in which we have had many joyful conversations.

Although he is yet not old enough to read this, I want to thank Stephan, my son, for the new source of energy that I could tap into after his birth. Finally, I would like to thank my wife, Esther, for standing by my side with me for all these years. Without your support, both emotional and technical, and without your encouragements, I would never have been able to complete this thesis.

*In every thing give thanks: for this is the
will of God in Christ Jesus concerning you.*
1 Thessalonians 5:18 (NKJ)

Curriculum vitae

Willem Dijkstra was born in Driebruggen, The Netherlands, on April 15th 1979. He completed his pre-university education at the Kalsbeek College in Woerden in 1997 and started his Applied Mathematics studies at the Eindhoven University of Technology in the same year. During his studies he performed an internship at Thales Nederland in Hengelo, concerning antenna-arrays in radar technology. From September 2001 until January 2004 he followed the postgraduate program Mathematics for Industry at the Stan Ackermans Institute (SAI) of the Eindhoven University of Technology. His studies at the SAI were concluded with a final project at Agrotechnology and Food Sciences in Wageningen. The final project report was simultaneously also his Master thesis. The final project concerned modelling the moisture loss in chicken filet during cooking. From January 2004 until April 2008 he was a PhD student in the Centre for Analysis, Scientific computing and Applications (CASA) at the Eindhoven University of Technology. The research performed in this group has led to this thesis.