

Generalised Stefan Problems

Linear analysis and computation

by

Roger David Donaldson

B.A.Sc., The University of British Columbia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

Department of Mathematics
Institute of Applied Mathematics

We accept this thesis as conforming
to the required standard

.....
.....
.....
.....

THE UNIVERSITY OF BRITISH COLUMBIA

December 2003

© Roger David Donaldson, 2003

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature) _____

Department of Mathematics
Institute of Applied Mathematics

The University Of British Columbia
Vancouver, Canada

Date _____

Abstract

We consider elliptic problems in which the domain is separated into two regions by a free boundary, on which mixed Dirichlet-Neumann conditions are specified. Led by the classical Stefan condition applied to change of phase models, we consider idealised numerical methods which evolve the interface by a trial method that uses the error in one of the boundary conditions as the normal velocity of the boundary. Using linear perturbation analysis of simple cases, we show exactly which interfacial conditions lead to well-posed problems and which choices of velocities lead to convergent methods. Moreover, some velocities lead to methods having superior numerical properties. Analysis of numerics representing the free boundary by a cubic spline fit is presented, followed by an example computation. Related ongoing work is introduced.

Contents

Abstract	ii
Contents	iii
List of Tables	v
List of Figures	vi
Preface	vii
1 Numerical Methods for Free Boundary Problems	1
1.1 Physical examples	3
1.2 Ice-water model	9
1.3 Solutions to free boundary problems	11
1.4 Convergence of explicit schemes	14
2 Linearised Problem	18
2.1 Base solutions	18
2.2 Criteria for well-posedness	20
2.3 Generalised Stefan velocities	24
2.4 Good velocity choices	26
3 Analysis of the Implementation	32
3.1 Fourier transform of the cubic spline	32
3.2 The cubic spline in generalised Stefan schemes	35
3.3 Smoothing Stefan velocity data	36
4 Computer Examples	39
5 Continuing Work	46
5.1 Generalised Stefan velocities for a level set method	46
5.2 Extensions to general operators and base boundaries	50
5.3 Boundary contact points	54
5.4 Summary	56
Bibliography	58

A	Calculation Details	61
A.1	Ellipticity of the porous media flow problem	61
A.2	Calculation of λ	63
A.3	The Fourier integral of the cubic spline	64
A.4	The Fourier transform of periodic functions	65
A.5	Series solution to model ice-water problem	66

List of Tables

2.1	Well-posedness for feedback conditions	24
4.1	Maximum stable step sizes for each discretisation	43

List of Figures

1.1	The dam seepage problem	4
1.2	The Alt-Caffarelli problem	5
1.3	Riabouchinsky flow	5
1.4	Flow of a Bingham fluid	7
1.5	Falling droplet	7
1.6	Two-phase flow in porous media	9
1.7	A model of the ice-water system	10
1.8	Forward Euler method stability	15
2.1	Normal to a curve with $\eta'(x) \ll 1$	21
2.2	Effect of velocity choice on the ice-water problem	30
2.3	Effect of velocity choice on the feedback problem	30
3.1	Sketch of $s_N(n) = s(n/N)$	35
4.1	Evolution of the ice-water boundary by the Neumann method	41
4.2	Evolution of the ice-water boundary by the Dirichlet method	42
4.3	Mean square error of numerical solutions	43
4.4	Oscillation in the Neumann method	44
4.5	Accuracy of the Neumann and Dirichlet methods	45
5.1	Boundary contact point geometry	55

Preface

The original motivation for this work is a problem in fuel cell water management. The fuel cell system facilitates a controlled reaction between hydrogen and oxygen to create electric energy and water, and understanding the formation and flow of this water in the fuel cell system plays a role in optimising fuel cell performance. This motivating problem seeks to determine the boundaries between liquid water, water vapour, and mixed liquid-vapour regimes within the porous media that permits water transport within the fuel cell. This problem is a free boundary problem, where not only state variables, but also phase boundaries, are unknown *a priori*. While the work of this thesis does not address the fuel cell problem directly, it analyses a novel strategy for handling the general class of free boundary problems of which the fuel cell problem is a member.

I wish to acknowledge my supervisor Brian Wetton for not only bringing this problem to me as a masters thesis project, but also for his ideas and support in its development. Work presented here is joint with Dr Wetton. I also wish to acknowledge NSERC, Ballard Power Systems Inc, and The University of British Columbia Department of Mathematics for providing financial support for this work. I look forward to seeing the application of ideas contained herein to fuel cell development in the near future.

Roger Donaldson

Vancouver, Canada
August 2003

Chapter 1

Numerical Methods for Free Boundary Problems

Partial differential equations (PDEs) often describe physical laws, such as conservation of mass, momentum, or energy. The solution of a PDE system gives information about the physical situation by describing how state variables, such as velocity or temperature, vary throughout a domain. In many cases, the boundaries of the domain are known, and boundary conditions, together with the PDE system, provide sufficient information to solve for the state variables. However, some physical situations do not specify the location of the domain boundaries, instead providing additional boundary information such that the state variables and the boundary location may be determined. Such problems are referred to as free boundary problems.

A typical free boundary problem statement includes a fixed physical boundary, but within the fixed domain, boundaries with unknown locations separate subdomains where different equations may apply to different sets of dependent variables. Even if the PDEs that apply in the subdomains are linear, free boundary problems are, on the whole, nonlinear, and are often difficult to solve numerically, let alone analytically [19, ch 7]. Free boundary problems can be time-dependent, and study of the time-dependent case has led to interest in so-called front tracking problems. However, efficiently determining steady-state solutions of free boundary problems is alone challenging, and the purpose of this thesis is to discuss a novel numerical method for solving steady-state free boundary problems, motivated by the classical Stefan velocity for phase change models.

While a variety of techniques exist for numerically solving free boundary problems, we look in particular at *trial methods*, also called fixed point methods. At a free boundary, in addition to the number of boundary conditions required to specify the problem were the boundary fixed, one further boundary condition must be provided to specify the free boundary position. Trial methods solve for the free boundary of a steady-state system as follows:

1. Choose an initial approximation to the free boundary.
2. Solve the resulting fixed boundary problem using all but one of the free boundary conditions.
3. The remaining condition will not be satisfied unless the position of the free boundary is accurate: use the residual in the remaining condition to

adjust the location of the boundary.

4. If the adjustment required is sufficiently small, stop; otherwise, adopt the new boundary and return to step 2.

Our focus is on the manner in which step 3 is carried out. We treat the error in the remaining condition as a normal boundary velocity, and use this velocity to track the boundary toward its steady state solution. In addition, we recognise that the choices of which boundary conditions are applied to solve the fixed boundary problem, and which is used as the velocity, are arbitrary. Some choices give the trial method superior numerical properties, while others give unstable methods. While we have already stated that free boundary problems are inherently nonlinear, addressing questions of uniqueness is beyond the scope of this thesis, and we instead confine our discussion to issues surrounding stability of the trial method using various velocity choices.

The work of this thesis uses tools of applied mathematics and computation accessible to senior undergraduate students of mathematics, physics or electrical engineering. Some of the signal processing tools may be unfamiliar to mathematics students, while some of the linear algebra and asymptotic analysis may be unfamiliar to students of physics or engineering. While this thesis is intended to be self-contained, the tools used, together with some suggested references, are as follows: The overall structure of the stability analysis is adapted from the theory of numerical solutions to ordinary differential equations (ODEs): for a discussion of numerical methods for, ODEs, cf. Dormand's text [10]. See Okendon et. al. [19, ch 7] for an introduction to free and moving boundary problems, including the form of our basic stability analysis. Motivation for the perturbation method used to evaluate stability can be found in Gustafson et. al. [15], and various smoothing and Fourier transform techniques can be found in signal analysis texts, cf. Hamming's book [16]. The physics of heat transfer may be studied from any number of engineering texts on the subject, but the development of the heat equation in any introductory PDEs book, such as that by Boyce and DiPrima [5] should suffice as a reference. Linear algebra is used extensively in our arguments: the text by Strang [28] is an excellent resource. For a comprehensive text on the finite element method, balancing theory with practical discussion, see Oden and Reddy [20]. Finally, Crank's text [8], contains an analytic treatment of classic free boundary problems, with its closing chapters discussing numerical methods.

We now proceed to several physical examples of free boundary problems. The first examples motivate the solution of free boundary problems in general, while the last of these examples provides motivation for our numerical method. This introductory chapter closes with a discussion of methods for solving free boundary problems, with focus on the convergence of our trial method. Chapter 2 evaluates the stability of the trial method applied to a linearised version of the free boundary problem, showing which free boundary problems are well-posed, and which choices of velocity conditions give superior numerical methods. Chapter 3 shows that the discretisation and smoothing operations in practical schemes preserve the numerical properties predicted by the idealised

linear analysis, and Chapter 4 presents the performance of the practical scheme in example computations. Chapter 5 discusses continuing work, introducing a hybrid scheme combining the benefits of a level set method with good numerical qualities of our trial method, and presents questions regarding points of contact between free boundaries and the physical boundary of the problem. Finally, an appendix provides details of some of the longer calculations.

1.1 Physical examples

Our analysis deals only with linear, constant coefficient boundary conditions. We also restrict our attention to the case of problems in two spatial dimensions where the elliptic operators on either side of the interface are the vector Laplacian, with some mild extensions possible as outlined in Chapter 5. However, we believe our analysis and the insights into robust and fast numerical methods that can be gained from it are extensible to much more general situations. We outline below some of the examples of the type of problems in our general class that motivated our work and to which our methods could be applied.

The examples are roughly in order of increasing complexity. Having a single dependent variable, the first four examples are the lowest order of the class of free boundary problems we consider. They consist of a scalar elliptic problem in a domain with a free boundary at which two mixed Dirichlet-Neumann conditions are applied. The next two examples have two state variables on either side of the free boundary, a total of four dependent variables, necessitating five free boundary conditions. Despite the range of complexity, all of these problems belong to the single class of free boundary problems we may treat using our trial method.

Dam seepage

A simple free boundary value problem is outlined in some detail in [8, ch 2]. Several techniques are also outlined for the solution of this problem in later chapters. It is the problem of finding the seepage region Ω (the region in which liquid water will be present) in a porous dam separating two reservoirs of water at different levels. A diagram of the situation is given in Figure 1.1. The reservoir levels y_1 and y_2 and the dam width L are given. The free boundary Γ ($y = f(x)$) must be determined. The equations that govern the system are

$$\Delta\phi = 0 \text{ in } \Omega \tag{1.1a}$$

$$\phi = y_1 \text{ on } x = 0 \tag{1.1b}$$

$$\phi = y_2 \text{ on } x = L \tag{1.1c}$$

$$\frac{\partial\phi}{\partial y} = 0 \text{ on } y = 0 \tag{1.1d}$$

$$\phi = y \text{ on } \gamma \tag{1.1e}$$

$$\phi = y \text{ on } \Gamma \tag{1.1f}$$

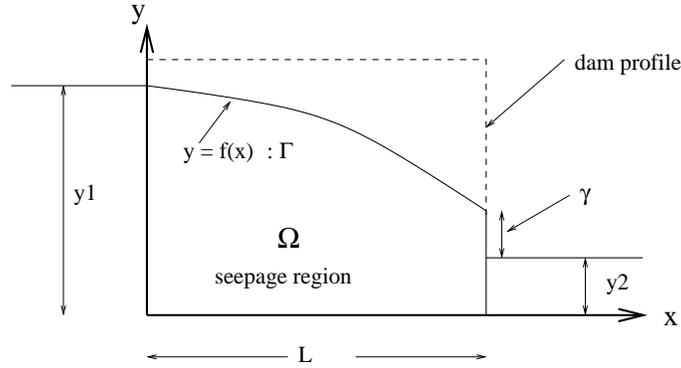


Figure 1.1: The dam seepage problem. The reservoir levels y_1 and y_2 and the dam width L are given. The free boundary Γ ($y = f(x)$) must be determined.

$$\frac{\partial \phi}{\partial n} = 0 \quad \text{on } \Gamma \quad (1.1g)$$

where ϕ is the modified velocity potential. Equation (1.1a) is the elliptic problem for the scalar quantity ϕ . One boundary condition (1.1b-1.1e) is given on each physical boundary. Two boundary conditions (1.1f,1.1g) (one Dirichlet and one Neumann) are given on the free boundary Γ . Thus, this is an example of the kind of general problem we consider in this work. We note that additional constraints on the problem need to be specified on γ and where the free boundary Γ meets the physical boundary at $x = 0$ and $x = L$ to obtain a physically reasonable solution. We refer the reader to [8] for details.

Alt-Caffarelli problem

Another scalar example is the Alt-Caffarelli problem [3], also known as the exterior Bernoulli problem [11]. Several applications in which this problem arises (fluid flow, optimal insulation, electrolytic galvanisation) are discussed in [11]. The schematic of this problem is given in Figure 1.2.

Here, the equations that apply are

$$\Delta u = 0 \quad \text{in } \Omega \quad (1.2a)$$

$$u = 1 \quad \text{on } \Sigma \quad (1.2b)$$

$$u = 0 \quad \text{on } \Gamma \quad (1.2c)$$

$$\frac{\partial u}{\partial n} = Q \quad \text{on } \Gamma \quad (1.2d)$$

where $Q < 0$ is given. Here, (1.2a) is the elliptic problem for the scalar u , (1.2b) is the single boundary condition on the physical boundary Σ and (1.2c,1.2d) are the two conditions (one Dirichlet and one Neumann) on the free boundary Γ . The conditions on the free boundary are very similar to those for the

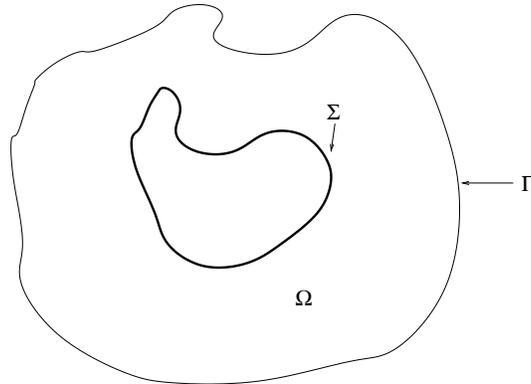


Figure 1.2: The Alt-Caffarelli problem. The physical boundary Σ is given. The free boundary Γ must be determined.

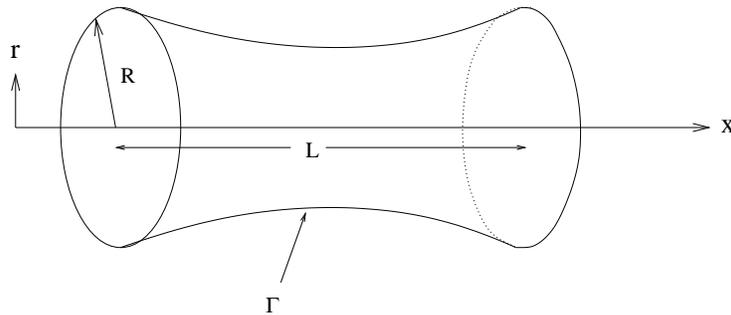


Figure 1.3: The Riabouchinsky flow past circular disks. Here, the disk size R and spacing L are given as well as the strength of the external flow past the disks. The free boundary Γ is to be determined.

dam problem above (1.1f, 1.1g). In fact the only difference at Γ is that the Alt-Caffarelli problem is forced through the Neumann condition and the dam problem through the Dirichlet condition. The physical boundary conditions are also different.

Riabouchinsky flow past circular disks

We discuss here a problem presented in [14] on the inviscid, irrotational flow past two disks, separated by a distance L , positioned perpendicular to the x -axis with their centres on this axis. A free surface Γ is to be found that connects the perimeters of the disks. This surface separates the external fluid flow from a region of cavitation. The situation is shown pictorially in Figure 1.3. The flow is axisymmetric and can be described by a stream function $\psi(x, r)$.

The equations that apply here are

$$\psi_{xx} + \psi_{rr} - \frac{1}{r}\psi_r = 0 \quad \text{in } \Omega \quad (1.3a)$$

$$\psi = 0 \quad \text{on the disks} \quad (1.3b)$$

$$\psi = 0 \quad \text{on } \Gamma \quad (1.3c)$$

$$\frac{\partial \psi}{\partial n} = r \quad \text{on } \Gamma \quad (1.3d)$$

where here Ω is the exterior flow domain. As before, we have one scalar elliptic equation in the domain (1.3a), one boundary condition at each physical boundary (1.3b), and two conditions at the free boundary (1.3c, 1.3d). Far field conditions must also be given to solve the problem. They can be considered physical boundary conditions at “ ∞ ”.

Hagen-Poiseuille flows of a Bingham fluid

Bingham fluids are particular non-Newtonian fluids that have a yield stress, a minimum stress below which they behave like solids, cf. [13]. We consider here the steady, pressure-driven flow of a Bingham fluid through a pipe of constant cross-sectional shape Σ , taken to be in the xy plane. We consider flows in which fluid moves only in the direction along the pipe with speed $w(x, y)$. It is possible that an unyielded plug forms away from the pipe walls as shown in Figure 1.4. This plug moves as a solid with common speed U , to be determined along with its shape Γ . It is also possible that unyielded regions form stuck to the walls of the pipe depending on the shape of Σ . These unyielded regions will have zero velocity. This is just an indication that for most free-boundary value problems from applications, the topology of the free boundary is not known *a priori* as in the preceding examples.

For just a single unyielded region away from the pipe boundary, the following describe the problem:

$$\nabla \cdot \left(\left[1 + \frac{B}{|\nabla w|} \right] \nabla w \right) = -\lambda \quad \text{in } \Omega \quad (1.4a)$$

$$w = 0 \quad \text{on } \Sigma \quad (1.4b)$$

$$w = U \quad \text{on } \Gamma \quad (1.4c)$$

$$\frac{\partial w}{\partial n} = 0 \quad \text{on } \Gamma \quad (1.4d)$$

where B is the yield stress of the fluid (a given physical parameter), $-\lambda$ is the given, imposed pressure gradient and U is the speed of the plug. Although nonlinear, the problem has the same structure as the previous ones: (1.4a) is a scalar elliptic equation, (1.4b) is a single condition on the physical boundary, and (1.4c, 1.4d) are conditions on the free surface. The velocity U is a global unknown to be determined as part of the problem.

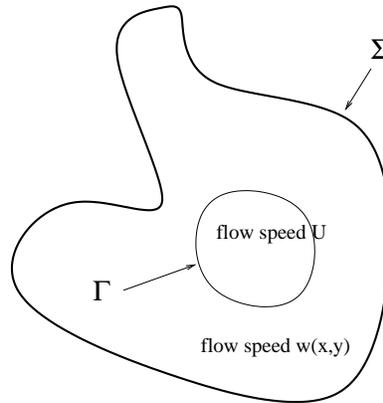


Figure 1.4: Pressure-driven Hagen-Poiseuille flow through a pipe of constant cross-sectional shape Σ . An unyielded region inside Γ can develop. Here, the material moves as a solid with constant speed U . Both the shape of Γ and U must be determined as part of the problem.

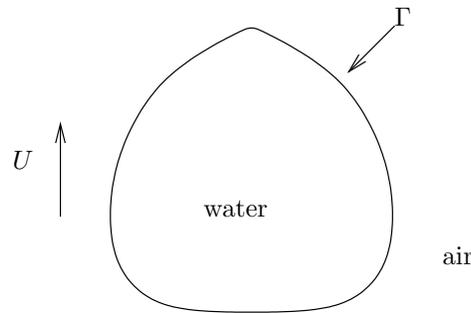


Figure 1.5: The falling droplet in a moving coordinate system.

Falling droplet

We consider the problem of a steady state droplet falling with speed U . With the speed given, the mass of the droplet falling with that steady state speed will be determined. We present the situation for a two-dimensional droplet to avoid the complexity of axisymmetric coordinates but the system has similar character in that more realistic case. We consider the droplet in a moving frame of speed $(0, -U)$ so it will appear stationary as shown in Figure 1.5.

The equations that apply here can be derived from equations for unsteady flow, cf [7].

$$\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \mu \Delta \mathbf{u} = 0 \quad \text{in both air and water} \quad (1.5a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in both air and water} \quad (1.5b)$$

$$\mathbf{u} \rightarrow (0, U) \text{ far from droplet} \quad (1.5c)$$

$$\left[p\hat{n} + \mu \frac{\partial \mathbf{u}}{\partial n} \right] = \kappa \sigma \hat{n} \text{ on } \Gamma \quad (1.5d)$$

$$[\mathbf{u}] = 0 \text{ on } \Gamma \quad (1.5e)$$

$$\mathbf{u} \cdot \hat{n} = 0 \text{ on } \Gamma \quad (1.5f)$$

where ρ is the density and μ the viscosity, both different for air and water; \mathbf{u} is the vector velocity; p is the pressure; $[\cdot]$ denotes the difference in quantities from the water to the air side of the free surface; \hat{n} is the unit normal direction of the free surface; κ is the curvature of the interface; and σ is the surface tension.

We can show that this system falls into our class of free boundary problems. First, it is appropriate to consider (1.5a, 1.5b) as a rank 2 system of elliptic equations in both air and water; it is possible to rephrase the problem in the right function spaces eliminating the pressure, see [30]. Equation (1.5c) represents two physical (far-field) boundary conditions for the air velocity. If we consider the surface tension as a forcing term, the stress-continuity equations (1.5d) are two mixed Dirichlet-Neumann conditions on Γ . Continuity of velocity (1.5e) gives two more Dirichlet conditions and the steady-state condition (1.5f) gives another. Thus, we have 5 total mixed Dirichlet-Neumann conditions for this system of four dependent variables, making this another example of our general class of problems.

Two-phase flow in porous media

We consider below a problem of two-phase flow in porous media, a two-dimensional version of the problem studied mathematically in [6], and by experiment in [34]. The problem considers a closed system containing a sand pack, water vapour and liquid water. If the system is heated at the top, it is possible to form two distinct zones separated by a free boundary. In the lower, two-phase zone, both vapour and liquid water can be found. In the upper zone, there is only water vapour. With low temperatures or enough total water in the system it is possible to have another free boundary to a third zone of pores filled with water, but we do not consider that case here. We consider the steady, two-dimensional case shown in Figure 1.6, taking the solution periodic in x for simplicity.

The variables pressure $P_v(x, t)$ and temperature $T(x, t)$ describe the flow in the upper region, and are governed by

$$\Delta T = 0 \quad (1.6a)$$

$$\nabla \cdot (C\mathbf{u}) = 0, \quad (1.6b)$$

where C and \mathbf{u} are functions of T and P_v . Flow in the lower region is described by temperature and saturation $s(x, y)$. s is the fraction of pore space occupied by liquid water. The equations are

$$\Delta T = -K_T \nabla(\rho_l u_l) \quad (1.6c)$$

$$\nabla(\rho_l u_l + \rho_v u_v) = 0. \quad (1.6d)$$

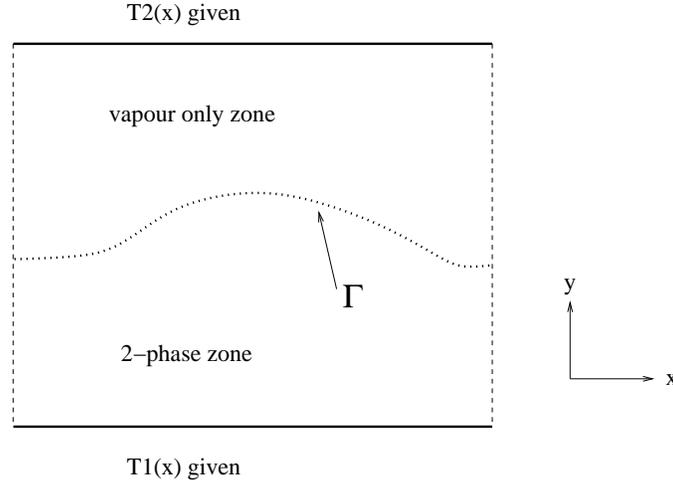


Figure 1.6: Two-phase flow in porous media. The upper temperature $T_1(x)$, the lower temperature $T_2(x)$ and the total water content W in the domain are given. The two-phase zone boundary Γ is to be determined.

where u_l , u_v , and ρ_v are functions of T and s . K_T is a constant. The relationships between the intermediate quantities and state variables give elliptic problems for T, P and s . Details are given in Appendix A.1.

On the free boundary, we have the following conditions:

$$s = 0 \quad (1.7a)$$

$$[T] = 0 \quad (1.7b)$$

$$[P_v] = 0 \quad (1.7c)$$

$$(\rho_v u_v)^+ \cdot \hat{n} = (\rho_v u_v + \rho_l u_l)^- \cdot \hat{n} \quad (1.7d)$$

$$\left[\frac{\partial T}{\partial n} \right] = K_T (\rho_l u_l)^-. \quad (1.7e)$$

These are five Dirichlet-Neumann conditions required for our problem of four variables. With additional conditions on the fixed boundaries, as well as a global condition, the problem is well-posed. This is too complex an example to develop our numerical scheme, but this problem demonstrates the range of problems to which our scheme applies. This problem is not addressed further in this thesis, but will be solved by the methods contained herein in future work.

1.2 Ice-water model

This final example is a two-dimensional model of temperature in an ice-water system with a phase interface, Γ . This example is a two-sided problem, but with

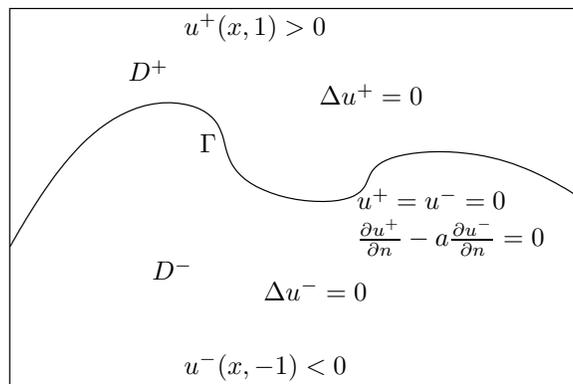


Figure 1.7: A model of the ice-water system. Two Dirichlet and one Neumann condition must be satisfied at Γ at steady state, although these conditions can be generalised to problems with other free boundary conditions.

a scalar elliptic problem on each side of the boundary for a total of two dependent variables, hence three conditions at the free boundary. From this last example, we develop our framework for making good choices for the velocity condition in our trial method.

Motivated by the physics, we take our first step in developing our numerical scheme for solving free boundary problems. Figure 1.7 illustrates the problem of a box filled with water, with temperature, u , held at $u^-(x, -1) < 0$ on the bottom boundary and $u^+(x, 1) > 0$ on the top boundary. A phase boundary, where $u = 0$, appears as $y = \eta(x)$, dividing the domain into an upper, water-filled subdomain, D^+ , and a lower, ice-filled subdomain, D^- . At steady state, the temperature obeys $\Delta u^\pm = 0$, where u^+ and u^- represent the temperatures above and below Γ , respectively. Thus, at Γ , we have the two conditions $u^+ = u^- = 0$. Since Γ is a free boundary, an additional condition is again required to completely specify the problem, and for this third condition, we look to the time-dependent problem.

In the time-dependent problem, the phase boundary moves locally as ice melts to water, or as water freezes. The normal velocity of the boundary, v_n , is given by the classical Stefan condition on heat flux normal to the boundary,

$$v_n = K \left(\frac{\partial u^+}{\partial n} - a \frac{\partial u^-}{\partial n} \right). \quad (1.8)$$

a is the ratio of thermal conductivities of ice to water and K is a physical constant of proportionality¹. At steady state, $v_n = 0$, and we use this as the third

¹In supercooled liquid interface problems, a small surface tension term is added to the RHS of (1.8). This term, while negligible in our situation, is needed in the supercooled case to obtain physically reasonable (well-posed) interface motion [24].

condition on Γ . This representation suggests a particular trial method for iteratively solving the free boundary problem by explicit time-stepping: Assume an approximation for Γ , solve the problem for u^+ and u^- in the resulting subdomains using the conditions $u^+ = u^- = 0$ at Γ , and then move Γ normally proportional to the Stefan velocity given by equation (1.8). When we discretise Γ , as is necessary in a numerical solve, this method treats the behaviour of each of the points of Γ as the solution to a time-dependent ODE. The iteration discretises “time” as the boundary evolves toward a steady state, yet as we are solving a steady-state free boundary problem, we give time no physical interpretation, but refer to it only in the sense of the progression of the solve.

This method converges, but, as we shall show, the stability of this method is dependent on the discretisation we choose for Γ ; finer discretisations for Γ require that shorter time steps be taken to maintain stability. The fixed-boundary problem must be solved each time Γ is adjusted, and, as this solve can be expensive, it is desirable to take time steps as large as possible, independent of the discretisation of the boundary. Fortunately, there is another option.

We can choose to set the Stefan velocity equal to zero for each fixed-boundary solve in the iteration. We enforce one other boundary condition, say, $u^+ = u^-$, and use the error in the remaining boundary condition, the value of $u^+ = u^-$, which we expect will ultimately tend toward zero, to determine the normal velocity of Γ . This is the essence of our approach:

If a good choice is made for the velocity condition, the stability of the method can be made independent of the boundary discretisation.

Note that out of deference to the classical Stefan velocity, we have adopted the term “velocity” to refer to the error in the boundary condition used to adjust the boundary at each iteration.

We return to the ice-water system to compare the numerical performance of the trial method using the physical Stefan velocity with the that using the alternative “Dirichlet” velocity in Chapter 4. We will use this system as motivation for considering the stability of problems with more general conditions at Γ . Our analysis will be restricted to the case where $\Delta u = 0$ is the problem in each subdomain. However, we expect that our stability predictions will hold for other elliptic operators, an important consideration for the other applications suggested. Before presenting a study of our numerical scheme, let us look at methods available for solving free boundary problems, with particular attention to their convergence.

1.3 Solutions to free boundary problems

We are not aware of any analytic theory covering our general class of free boundary problems, although there has been progress on individual problems. References to early work can be found in [8], including a lengthy discussion on the dam problem. Existence and regularity of the Alt-Caffarelli problem is shown

in [3], extensively using a reformulation of the problem as a functional minimisation. An exact solution for the ice-water example of the previous section does exist: the interface is the zero level set of a harmonic function after a reformulation shown in Appendix A.5. In [6] semi-analytic solutions and their regularity for the two-phase zone problem are presented, but only for a one-dimensional problem.

Given the difficulty in formulating analytic solutions for free boundary problems, seeking good numerical schemes for solving these problems is desirable. Several techniques are discussed in the literature, of which we introduce *capturing techniques* and *front tracking*. Front tracking encompasses both *shape optimisation* and our trial method, the latter of which we expand upon with respect to its stability.

Capturing techniques

A useful method for solving the type of problems we have considered is to reformulate the problem so that modified equations are solved in the whole domain (both sides of the interface). The interface is then recovered from the resulting solution. The Extended-Lagrangian formulation of Bingham fluids [13] and the entropy formulation for the Stefan problem [8] are examples of this approach.

Not all problems can be solved using this idea, although a variety of related approaches are available. The falling droplet problem can be approached using the methods developed in [7]. In their method, they discretise both the air and water as a uniform fluid with a transition zone in density and viscosity at the interface. This transition zone automatically approximates the interface conditions. The level set method of Osher and Sethian [23] is used to capture the interface location and provide the curvature for the surface tension term. The level set method is able to capture the topological changes of merging droplets and bubbles. In total, this is a delicate and sophisticated method. We return to the level set method in Chapter 5, where we combine it with our trial method to capture the good qualities of each approach.

It may be possible to reformulate the two phase flow problem and capture the interface by relaxing the saturation assumption in the two-phase zone and allowing condensation to happen at a finite rate proportional to the oversaturation [12]. However, this is not a straight-forward process either, and further study would be required to extend this idea to more general problems.

Front tracking and shape optimisation

Another numerical approach is to discretise the free boundary directly, forming a problem where the mismatch of an approximate boundary to the desired one is minimised. Such techniques are called front tracking, for their analogues in time-dependent free boundary problems. This is exactly the form of our trial method, whereby each point on the boundary discretisation is moved normally according to a chosen velocity.

To our knowledge, work exploring the effects the choice of velocity condition has on a trial method reduces to work by Garabedian [14], in which he addresses the one-sided Riabouchinsky flow problem (1.3a-1.3d). Garabedian's idea is instead of using the first or second boundary condition alone in the fixed boundary PDE solve, to choose a combination of the two conditions that does not change along the normal to Γ , a *stationary combination*. Garabedian uses the error in the Dirichlet condition $u = 0$ to adjust the boundary. For this particular problem, just as in Newton's method, Garabedian's method gives a quadratic rate of convergence. If $u = 0$ is used to solve the fixed boundary problem, with error in the Neumann condition used to adjust the boundary, the convergence rate is linear.

In their review paper on free boundaries of liquid in capillary tubes, Cuvelier and Schulkes [9] reproduce Garabedian's idea from a similar perspective. They determine how to adjust the free boundary by linearising solutions of the free boundary problem near its supposed location. The linearised version of the problem local to the boundary gives Garabedian's method, which, owing to its derivation, they refer to as *total linearisation*. They apply total linearisation to a one-sided free boundary problem similar to that considered by Garabedian.

Several other authors have since reproduced Garabedian's idea using free boundary adjustment by shape optimisation [11, 18, 31–33]. Shape optimisation solves free boundary problems by a variational approach, minimising an integral functional that integrates a cost function over the variable domain having a free boundary. Shape optimisation uses shape calculus, which takes derivatives of an integral functional with respect to changes in its domain of integration. Ultimately, then, shape optimisation uses Newton's method to determine the boundary location where the shape derivative is zero. For an introduction to the shape calculus, cf [26, 27].

Garabedian's idea manifests itself in the context of shape optimisation when the freedom in the variational formulation is used to keep one combination of free boundary conditions stationary as the boundary is adjusted. This is exactly Garabedian's approach, and results in the error in u being, from the variational perspective, the correct condition to use to adjust the free boundary. Again, these authors consider the one-sided free boundary problems addressed by Garabedian, and report the superior convergence properties of the Garabedian trial method.

To avoid the calculation of shape derivatives, which is really only feasible for fairly simple systems, other techniques using only the values of the interface residuals have been developed, cf. [8]. In certain situations, such as in [14], they can be made to work almost as well as shape optimisation (Newton) methods. In some situations they are able to move a poor initial estimate to the desired interface. In this work, we consider a general framework to analyse value-type methods and identify those with superior numerical properties.

Our method looks generally at choices for the velocity condition, and while rate of convergence, the primary motivation of the above authors, is important, we look instead at the stability of methods using different velocity conditions. We also generalise the problem to systems having many dependent variables

on either side of the free boundary, a feature particularly important in our application to phase change models. We treat the free boundary iteration as a semi-discretisation, tracking the evolution of the boundary in time. This allows us to evaluate the convergence of free boundary solves in the framework of numerical methods for ordinary differential equations (ODEs).

1.4 Convergence of explicit schemes

The trial method is an *explicit scheme*: the future location of Γ in the iteration depends only on its current (and, in a more general setting, previous) location. Such schemes are convenient to compute, and often converge rapidly, but are prone to instability if not applied carefully. That explicit iterative solutions to free boundary problems are prone to instability is known, cf. [25], and we must ask how stability depends on features of the problem. This section formalises the compromise between stability and convergence rate in terms of the problem stiffness. We begin with a simple example.

Consider the ODE

$$\frac{dy}{dt} = \lambda y, \quad (1.9)$$

where $y(0)$ is given. λ is the growth rate of the problem, and constraints on how we solve this problem depend on the size of λ . The lowest order explicit scheme we can use for this problem is the forward Euler method, for which we have the approximation

$$y^{n+1} = y^n + \lambda k y^n, \quad (1.10)$$

where k is the time step size, and if $y^0 = y(0)$, y^n approximates $y(nk)$.

If $\text{Re}(\lambda) < 0$, the exact solution decays, so, too, we should expect the approximation to decay. Hence, we apply the criterion that the growth rate, $|y_{n+1}/y_n| < 1$, so that

$$|1 + \lambda k| < 1. \quad (1.11)$$

If this condition is violated, the numerical solution will grow, an unacceptable representation of the decaying exact solution. We use condition (1.11) as the stability condition for the forward Euler method applied to this problem, and we note that our choice for k depends on the intrinsic growth (decay) rate of the problem at hand. Figure 1.8 is a sketch of the stability regime for the forward Euler method as a function of λk . If λ is purely real, we require $-2 < \lambda k < 0$, and if we would like y^n to decay to zero as quickly as possible, we choose

$$k = -\frac{2}{\lambda} \quad (1.12)$$

In which case $y^1 = 0$. This is different than considering how to choose k such that our numerical solution best approximates the exact solution². In our trial

²Higher order Runge-Kutta schemes for approximating (1.9), those that obtain better accuracy for a given k , do exist, but, in determining steady-state solutions, because we are not concerned with the accuracy of the decay dynamics, higher order schemes provide us no advantage over the forward Euler method.

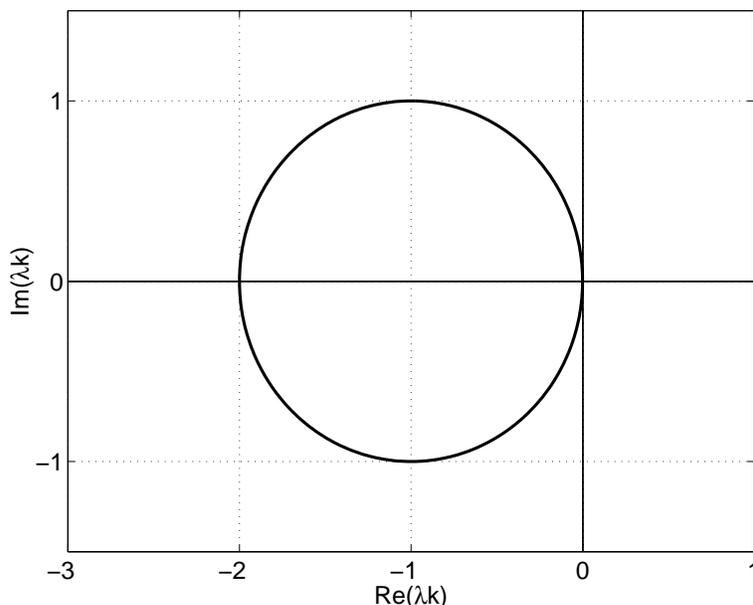


Figure 1.8: Stability regime of the forward Euler method. The method is stable as long as λk lies inside the circle. Note the allowance for both real and complex λ .

method, y represents the error in the free boundary location, and we wish to see this error reduced to zero in as few iterations as possible. This simple development applies to solving for the scalar $y = y(t)$. Let us extend this idea to our trial method.

Consider our numerical method. Let \mathbf{y} be a vector representing the y -locations of N points discretising the free boundary. At each iteration, we adjust each y -value of the boundary to a new location by

$$\mathbf{y}^{n+1} = \mathbf{y}^n + kv_n. \quad (1.13)$$

Again, v_n is the normal velocity of the boundary, as determined by the velocity condition. (Strictly speaking, we should be moving each boundary point normally to Γ , but this is a good approximation if Γ is fairly flat. We will make this assumption rigorous in the next chapter.) $v_n = \mathbf{A}^n(\mathbf{y})$, $\mathbf{A}^n : \mathbb{R}^N \rightarrow \mathbb{R}^N$, is a function that computes the normal velocities of the boundary points based on the solution to the associated fixed boundary problem at iteration n . $\mathbf{A}^n(\mathbf{y})$, therefore, encompasses the PDE solve at iteration n and depends on the choice of velocity condition.

Let us assume that \mathbf{A}^n is linear. This implies that the solution to the free boundary problem is $\mathbf{y} = 0$. This assumption is carried through the analysis of succeeding chapters and, despite the fact that free boundary problems are

inherently nonlinear, choosing linear \mathbf{A}^n does serve as a good model for the behaviour of our iterative scheme. We omit the n superscript from here on, with the implicit understanding that \mathbf{A} may change from iteration to iteration. In a practical problem, the solution will not likely be $\mathbf{y} = 0$, but if we instead regard \mathbf{y} as the error, the difference between our approximation and the exact solution, the assumption that \mathbf{A} is linear for small error is reasonable and using this model makes sense.

Suppose that \mathbf{A} is negative definite, so that the velocity decays as \mathbf{y} approaches the solution, $\mathbf{y} = 0$. Furthermore, suppose that \mathbf{A} has eigenvalues $\lambda_M = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N = \lambda_m < 0$. Thus, equation (1.13) becomes

$$\mathbf{y}_i^{n+1} = \mathbf{y}_i^n + k\mathbf{A}\mathbf{y}_i^n = \mathbf{y}_i^n(1 + k\lambda_i) \quad (1.14)$$

for each eigenvalue λ_i and each eigenvector \mathbf{y}_i^n . We would like to choose k that keeps the solve stable while advancing it as quickly as possible. That is, we wish to know the value of k that minimises the maximum growth rate r of the problem, dependent on $\{\lambda_i\}$. Hence,

$$r = \min_{k>0} \left\{ \max_{\lambda \in \{\lambda_i\}} \{|1 + k\lambda|\} \right\}. \quad (1.15)$$

To compute r and the k at which r is attained, we observe that all values $|1 + k\lambda|$ lie between $|1 + k\lambda_m|$ and $|1 + k\lambda_M|$. Since $\lambda_M \leq \lambda_m < 0$, the minimising k occurs when

$$\begin{aligned} -(1 + k\lambda_m) &= 1 + k\lambda_M, \\ k &= \frac{-2}{\lambda_m + \lambda_M}, \end{aligned} \quad (1.16)$$

and so

$$r = \frac{\kappa - 1}{\kappa + 1}, \quad (1.17)$$

where $\kappa = \lambda_M/\lambda_m$ is the condition number of \mathbf{A} . κ is also referred to as the problem *stiffness*, large values of κ representing stiff problems. Thus, equation (1.17) shows that when $\kappa \gg 1$, $r \approx 1$ and $\mathbf{y}^{n+1} \approx \mathbf{y}^n$. Since linearity of \mathbf{A} implies that $\mathbf{y} = 0$ is the solution, such stiff problems converge slowly. However, when $\kappa \approx 1$, $r \ll 1$ and for any \mathbf{y}^n , $\mathbf{y}^{n+1} \approx 0$, indicating a rapidly converging problem.

Having the freedom to choose the velocity condition for the trial method is synonymous with having the freedom to choose stiffness qualities of \mathbf{A} . Equation (1.16) tells us how to best choose k for a given \mathbf{A} , but more importantly, expression (1.17) for r tells us what the eigenvalues of optimal \mathbf{A} should be. The closer κ is to 1, the faster the convergence to $\mathbf{y} = 0$ will be for optimal k . The central idea of this work is to capitalise on the freedom in our choice of velocity condition to minimise the stiffness of the problem.

In the next chapter, we calculate the eigenvalues of the problem with general free boundary conditions where the calculation is done for a linearised problem

over an infinite domain. In this setting, the problem has an infinite number of eigenvalues, but we can immediately anticipate how the distribution of eigenvalues affects the stiffness of related discrete problems. Chapter 3 applies the results from the linear analysis to a practical scheme with a discretised free boundary, hence a discrete set of eigenvalues, which more closely follows the motivation from this introductory discussion.

Chapter 2

Linearised Problem

Our trial method for tracking boundary points is an ODE-type semi-discretisation. We use a linearised version of the problem to determine conditions for stable numerical schemes that use explicit semi-discretised front tracking methods to evolve the free boundary. The stability analysis takes a known base solution, perturbs the free boundary in that solution by a single wave-mode, and checks for growth or decay of that mode. The wave-modes carried by Γ depend on its discretisation, but for this initial analysis, we compute how the stiffness, hence maximum step size, depends on wavenumber, and treat the discretised boundary in Chapter 3. This procedure is equivalent to a spectral analysis of the linearised problem.

We shall see several major results in this linear analysis. Before proceeding to the stability analysis, we examine the existence of base solutions for general linear boundary conditions. This investigation will indicate that the behaviour of the free boundary can depend not only on the boundary conditions themselves, but also on global conditions of the problem. Using these base solutions, we use the form of the perturbation analysis to show which boundary and global conditions give well-posed problems: conditions at the free boundary cannot be specified arbitrarily. Finally, after developing the relevant formalism, we shall come to the point of our analysis having direct impact on the numerics: we predict the effect that velocity choice has on the stability of the explicit iterative scheme for determining the free boundary location.

2.1 Base solutions

We consider the problem in \mathbb{R}^2 . While the ice-water problem has only two dependent variables, u^+ and u^- , we are by no means bound by this restriction, and instead of 2 elliptic problems, we consider m elliptic problems for each of m dependent variables. Each of the variables may apply variously above and below the free boundary. We assume that $\Delta \mathbf{u} = 0$ in both D^+ and D^- (the upper and lower subdomains, respectively), and that the free boundary, Γ , can be represented as a function, $y = \eta(x)$. We consider the domain to be infinite in y and in x . We assume that the $m+1$ conditions at Γ are linear in \mathbf{u} and \mathbf{u}_n , that is,

$$\mathbf{G}\bar{\mathbf{u}} = \mathbf{b} \tag{2.1}$$

where \mathbf{G} is an $m+1$ -by- $2m$ matrix. Hence, \mathbf{G} accounts for the additional condition needed to allow for the freedom in the boundary location. We assume

no redundancy in the boundary conditions, so \mathbf{G} is full rank. \mathbf{b} is a constant vector representing any driving conditions at Γ , and $\tilde{\mathbf{u}} = [\mathbf{u}_n, \mathbf{u}]$, the normal derivatives and values of \mathbf{u} at the boundary. Boundary normals are taken in the positive y -direction. In the case of the ice-water system, for example,

$$\mathbf{G} = \begin{bmatrix} 1 & -a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

$\mathbf{b} = \mathbf{0}$, and $\tilde{\mathbf{u}} = [u_n^+, u_n^-, u^+, u^-]^T$. Equations (2.1) may be row-reduced without losing boundary data, so this form for linear boundary conditions makes it clear why conditions for one-sided free boundary problems, those for which $m = 1$, ultimately reduce to those studied by Garabedian: with only two boundary conditions to specify and only one dependent variable, the only choices are $u = \text{const}$ and $u_n = \text{const}$.

We assume asymptotic expansions for the boundary $y = \eta(x)$ and the associated solution \mathbf{u} :

$$\eta \sim \eta_0 + \epsilon\eta_1 + \dots \quad (2.3)$$

$$\mathbf{u} \sim \mathbf{u}_0 + \epsilon\mathbf{u}_1 + \dots \quad (2.4)$$

Our goal is to determine base solutions, \mathbf{u}_0 , for the case where the base boundary is $y = \eta_0(x) = 0$.

Our differential operator is the Laplacian in each subdomain, so base solutions are of the form $\mathbf{u} = y\mathbf{r}^1 + \mathbf{r}^0$, where the constant vector $\tilde{\mathbf{r}} = [\mathbf{r}^1, \mathbf{r}^0]^T$ solves

$$\mathbf{G}\tilde{\mathbf{r}} = \mathbf{b}. \quad (2.5)$$

$\tilde{\mathbf{r}} = \tilde{\mathbf{r}}_H + \tilde{\mathbf{r}}_P$, where the particular solution, $\tilde{\mathbf{r}}_P$, solves

$$\mathbf{G}^T \mathbf{G} \tilde{\mathbf{r}}_P = \mathbf{G}^T \mathbf{b}, \quad (2.6)$$

and

$$\mathbf{G}\tilde{\mathbf{r}}_H = 0, \quad (2.7)$$

so $\tilde{\mathbf{r}}_H$ belongs to the nullspace of \mathbf{G} . \mathbf{G} is $m+1$ -by- $2m$, so its nullspace has dimension $m-1$. Including $m-1$ far-field conditions on \mathbf{u} specifies the base solution completely. As a result, stability will depend on global far-field conditions as well as local conditions at Γ . Note that if $\mathbf{b} = 0$, $\tilde{\mathbf{r}}_P = 0$, and only $m-2$ far-field conditions are required for a stability analysis, as these will determine the base solution to within multiplication by a constant. At worst, this constant may be chosen to have the incorrect sign, and if an instability owing to the base solution is determined, changing the sign of any resulting boundary velocity will be the obvious corrective action. This is the case for the ice-water system, where $m = 2$ and the global condition is the total energy flux across the boundary. In problems with more than two dependent variables, however, the base solution depends on a linear combination of null vectors of \mathbf{G} and local stability (ie at Γ) is affected by the global constraints (global fluxes) that determine the

correct combination of these null vectors for the problem. The local problem cannot determine these fluxes: they are determined by global conditions, and we consider them to be given.

2.2 Criteria for well-posedness

Since we have generalised our boundary conditions, it is not clear which conditions \mathbf{G} and global fluxes give well-posed problems. A problem is well-posed if a solution exists, that solution is unique, and if the solution depends continuously on the data. We well-posedness spectrally, by driving the boundary conditions by a single wave-mode, in which case the boundary conditions (2.5) become

$$\mathbf{G}\tilde{\mathbf{u}} = \mathbf{b} + \epsilon \mathbf{f} e^{i\alpha x}. \quad (2.8)$$

We seek a perturbation solution of form (2.4). Given the form of the driving function, we suppose that $\eta_1 = h e^{i\alpha x}$, and, since $\Delta \mathbf{u} = 0$,

$$\mathbf{u}_1 = \mathbf{c} e^{\pm|\alpha|y} e^{i\alpha x}, \quad (2.9)$$

where we choose the appropriate sign for $|\alpha|$ to ensure decay as $y \rightarrow \pm\infty$. This reduces solving the problem with boundary condition (2.8) to solving for \mathbf{c} and h . Thus, we suggest the following definition:

When the solution $[\mathbf{c}, h]$ exists and is unique for all driving conditions defined by \mathbf{f} and all α , we state that the problem with boundary conditions \mathbf{G} and given global fluxes is linearly well-posed.

Following the development in [19], we use the asymptotic expansions (2.3) and (2.4) with the assumption that $\epsilon \ll 1$ to approximate the conditions that must hold at $\eta_0 = 0$ given that we wish to satisfy the boundary conditions on $\eta(x)$. First, we expand $\mathbf{u}(x, \eta(x))$ as a Taylor series, recalling that $\eta_0(x) = 0$. We shall suppress the t -dependence of η for clarity.

$$\begin{aligned} \mathbf{u}(x, \eta(x)) &= \mathbf{u}(x, \eta_0(x) + \epsilon \eta_1(x) + \dots) = \\ &= \mathbf{u}(x, 0) + \epsilon \eta_1(x) \mathbf{u}_y(x, 0) + \mathcal{O}(\epsilon^2). \end{aligned} \quad (2.10)$$

Now we substitute the asymptotic expansion for \mathbf{u} , and our assumed $\eta_1(x)$, so that at the boundary

$$\mathbf{u}(x, \eta(x)) = \mathbf{u}_0(x, 0) + \epsilon [\mathbf{u}_1(x, 0) + h e^{i\alpha x} \mathbf{u}_{0y}(x, 0)] + \mathcal{O}(\epsilon^2). \quad (2.11)$$

We also need \mathbf{u}_n in order to approximate $\tilde{\mathbf{u}}$. Now, $\mathbf{u}_n = \nabla \mathbf{u} \cdot \hat{n}$, so we need to compute \hat{n} , the unit normal vector to Γ . Referring to the sketch in Figure 2.1, we note that the slope $\eta'(x) = \epsilon \eta_1'(x) \ll 1$, so the unit tangent vector is $\hat{t} \propto (1, \eta'(x))$ and so $\hat{n} \propto (-\eta'(x), 1)$. The x -component of \mathbf{u}_n is $\mathcal{O}(\epsilon)$ while the y -component is $\mathcal{O}(1)$. Hence, to lowest order, $\mathbf{u}_n = \mathbf{u}_y$, so

$$\mathbf{u}_n(x, \eta(x)) = \mathbf{u}_{0y}(x, 0) + \epsilon [\mathbf{u}_{1y}(x, 0) + h e^{i\alpha x} \mathbf{u}_{0yy}(x, 0)] + \mathcal{O}(\epsilon^2). \quad (2.12)$$

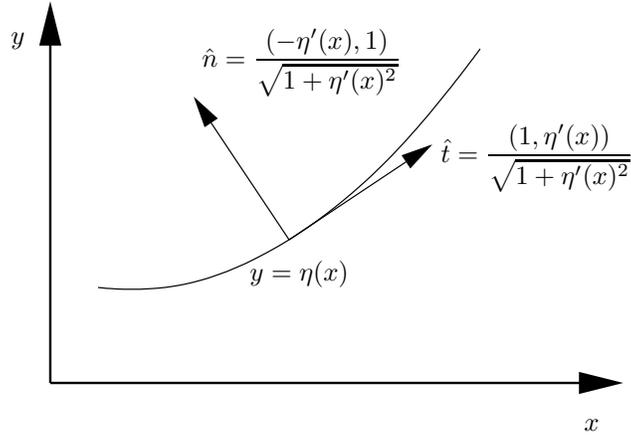


Figure 2.1: Calculation of the normal to a function $y = \eta(x)$ showing that when $\eta'(x) \ll 1$, $\hat{t} \sim (1, 0)$ and $\hat{n} \sim (0, 1)$. The $\eta(x)$ sketched here is excessively curved in order to clearly distinguish \hat{t} and \hat{n} . In our analysis, however, $\eta(x)$ is nearly flat.

Setting $\tilde{\mathbf{u}} = [\mathbf{u}_n, \mathbf{u}]^T$, equation (2.8) becomes, to $\mathcal{O}(\epsilon)$,

$$\mathbf{G}\tilde{\mathbf{u}}_0 + \epsilon\mathbf{G}[\tilde{\mathbf{u}}_1 + h e^{i\alpha x} \tilde{\mu}_0] = \mathbf{b} + \epsilon\mathbf{f}e^{i\alpha x}, \quad (2.13)$$

where $\tilde{\mu}_0 = [\mathbf{u}_{0yy}, \mathbf{u}_{0y}]^T$ is dependent on \mathbf{u}_0 , the base solution, and $\tilde{\mathbf{u}}_0$ balances the $\mathcal{O}(1)$ terms by equation (2.5). $\tilde{\mathbf{u}}_1$ depends linearly on \mathbf{c} , so

$$\tilde{\mathbf{u}}_1(x, 0) = e^{i\alpha x} \mathbf{K}\mathbf{c}, \quad (2.14)$$

where

$$\mathbf{K} = \begin{bmatrix} \pm|\alpha| & & & \\ & \ddots & & \\ & & \pm|\alpha| & \\ 1 & & & \\ & \ddots & & \\ & & & 1 \end{bmatrix} \quad (2.15)$$

is a $2m$ -by- m matrix dependent on the specific elliptic differential operators at hand, the Laplacian in this case. The sign of each $\pm|\alpha|$ term is chosen such that the $\mathcal{O}(\epsilon)$ solution decays to zero far from Γ . Hence, looking again at equation (2.9), this means that $-|\alpha|$ applies to u above Γ , and $+|\alpha|$ applies to u below. Thus, the $\mathcal{O}(\epsilon)$ equation becomes

$$\mathbf{G}\mathbf{K}\mathbf{c} + h\mathbf{G}\tilde{\mu}_0 = \mathbf{f}. \quad (2.16)$$

This system is full-rank, hence solvable for \mathbf{c} and h , as long as $\mathbf{G}\tilde{\mu}_0$ does not lie in the columnspace of $\mathbf{G}\mathbf{K}$. $\mathbf{G}\mathbf{K}$ is $m+1$ -by- m and is full rank, hence its left

nullspace is spanned by a single vector $\mathbf{w} = \ker[(\mathbf{G}\mathbf{K})^T]$. So for solvability, and thus for well-posedness, we require that

$$\mathbf{w}^T \mathbf{G} \tilde{\mu}_0 \neq 0 \quad (2.17)$$

for each α , where \mathbf{w} depends on α . Recall that $\tilde{\mu}_0$ depends on the boundary conditions defined by \mathbf{G} and \mathbf{b} , as well as the given global fluxes.

Let us consider two examples. The first will use the boundary conditions for the ice-water model, equations (2.2), a problem we know to be well-posed, and the second will use a set of conditions directly relating the values of the dependent variables at the boundary to their normal derivatives in order to make some connection between our mathematics and physically reasonable problems.

The ice-water problem. The ice-water problem has two dependent variables, one above the boundary and one below, hence we use the operator matrix

$$\mathbf{K} = \begin{bmatrix} -|\alpha| & 0 \\ 0 & |\alpha| \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.18)$$

Recalling the boundary conditions \mathbf{G} for the ice-water model (2.2),

$$\mathbf{G}\mathbf{K} = \begin{bmatrix} -|\alpha| & -a|\alpha| \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.19)$$

so that $\mathbf{w} = [1, |\alpha|, a|\alpha|]^T$. The base solution for the ice-water problem is, as previously stated, represented by the nullspace of \mathbf{G} which is $[1, 1/a, 0, 0]^T$. Thus, with the global flux a constant, we may take $\tilde{\mu}_0 = [0, 0, 1, 1/a]^T$, and our well-posedness condition is

$$\mathbf{w}^T \mathbf{G} \tilde{\mu}_0 = 2|\alpha| \neq 0. \quad (2.20)$$

This is certainly true for $|\alpha| > 0$. (The fact that the solvability condition does not hold for $\alpha = 0$ is not concerning or surprising, as in such case, the perturbation is $\epsilon \mathbf{f} = \text{const}$ with is parallel to the base driving condition \mathbf{b} , and may be added to the base driving condition prior to determining the base solutions.) That the solvability condition holds here is an indication that the problem is well-posed, as we should expect from its physical interpretation.

Feedback conditions. Again, we consider a system having two dependent variables, one above Γ and the other below. However, let us investigate another set of boundary conditions,

$$\mathbf{G} = \begin{bmatrix} \delta^+ & 0 & 1 & 0 \\ 0 & a\delta^- & 0 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad (2.21)$$

where again $a > 0$ is the ratio of thermal conductivities of the upper and lower regions. \mathbf{G} again applies to $\tilde{\mathbf{u}} = [u_n^+, u_n^-, u^+, u^-]^T$. With $\delta^\pm = \pm 1$ and

$\delta^- = \pm 1$, the first two conditions are *feedback conditions* on the variables, where the normal derivative of the solution at the boundary (often interpreted as proportional to a rate of energy flow) depends on the value of the solution. Two feedback types, positive feedback, where an increase in u increases u_n , and negative feedback, where an increase in u decreases u_n , can be represented, depending on the signs of δ^\pm . Given our convention that the \hat{n} always points upward, these first two conditions represent negative, or stabilising feedback if $\delta^+ = -1$ and $\delta^- = 1$, and represent positive, or destabilising feedback if $\delta^+ = 1$ and $\delta^- = -1$. A discussion following the calculation clarifies the physical interpretation and addresses the other two cases.

The third condition is relatively arbitrary: in this case we enforce continuity of the solution across Γ . Leaving the physical argument briefly, let us see what the mathematics says about what the signs of δ^\pm must be to satisfy the well-posedness condition.

We apply the Laplacian in each subdomain, so we use the same operator matrix, \mathbf{K} , as for the ice-water example, equation (2.18), so that

$$\mathbf{GK} = \begin{bmatrix} -|\alpha|\delta^+ + 1 & 0 \\ 0 & a|\alpha|\delta^- + 1 \\ 1 & -1 \end{bmatrix}, \quad (2.22)$$

and so

$$\mathbf{w} = \begin{bmatrix} a|\alpha|\delta^- + 1 \\ |\alpha|\delta^+ - 1 \\ (a|\alpha|\delta^- + 1)(|\alpha|\delta^+ - 1) \end{bmatrix}. \quad (2.23)$$

The nullspace of \mathbf{G} is $[-\delta^+, -\delta^-/a, 1, 1]^T$ so $\tilde{\mu}_0 = [0, 0, -\delta^+, -\delta^-/a]^T$. Working through the algebra, including identifying $(\delta^+)^2 = (\delta^-)^2 = 1$, we have the well-posedness criterion

$$\mathbf{w}^T \mathbf{G} \tilde{\mu}_0 = |\alpha|[(\delta^+ - a\delta^-)|\alpha| - 2] \neq 0 \text{ for } \alpha \neq 0. \quad (2.24)$$

Hence, requiring that zeros of equation (2.24) lie on the negative real line is sufficient for well-posedness. That is,

$$p(\delta^+, \delta^-) := \delta^+ - a\delta^- < 0. \quad (2.25)$$

There are four possible combinations of δ^+, δ^- , and we summarise these combinations in Table 2.1.

The results in this table, while a direct result of the mathematics, are consistent with the physical interpretation of feedback conditions. Suppose u^\pm do indeed represent temperature. If temperature decreases, we should expect heat to flow into the boundary. Rate of heat flow out of a region is proportional to $-\nabla u \cdot \hat{n}^*$, where \hat{n}^* is the outward unit normal. Hence well-posed boundary conditions should satisfy something like $u_n^* + u = \text{const}$. The third case in our table sets this very condition on both u^+ and u^- , remembering that for u^+ , the direction of the outward normal to the boundary appears reversed with respect

δ^+	δ^-	$p(\delta^+, \delta^-)$	comment
1	1	$1 - a$	well-posed for $a > 1$
1	-1	$1 + a$	never well-posed
-1	1	$-1 - a$	well-posed for all a
-1	-1	$-1 + a$	well-posed for $a < 1$

Table 2.1: Well-posedness for various feedback conditions. Recall that $a > 0$. Physical interpretations of these results are in the text.

to the region D^+ , since by our convention, \hat{n} always points upward. This situation is stabilising, and is called *negative feedback*. If the opposite condition is imposed, we have *positive feedback*, the second case in the table, and the problem is ill-posed.

The two conditional cases in Table 2.1 are a result of the coupling between the two regions by the continuity condition at the boundary. These cases are well-posed for certain values of a , because if too much energy leaves or enters the boundary from one region for a given u , energy can leave or enter the boundary from the other region to maintain negative feedback overall. Since the degree of this compensation depends on a , so do these conditional cases.

In the case of feedback of a single dependent variable at a fixed boundary, a similar analysis shows that our well-posedness condition (2.17) is the same as the requirement that $\alpha\beta \geq 0$ for the Robin condition $\alpha u_n + \beta u = g$ to provide a unique solution to the usual elliptic boundary value problem, cf. [19].

2.3 Generalised Stefan velocities

With the linear well-posedness of problems for constant coefficient boundary conditions established, we can discuss the stability of numerical methods for free boundary problems. In particular, motivated by our discussion of stability in the introductory chapter, we wish to compute the decay rate or stiffness of the problem. The stiffness, λ , is a function of the wave-modes carried by the boundary, hence, we determine stability by computing the decay rate of perturbations to the boundaries of base solutions of the problem. Recalling that the base boundary is $y = \eta_0(x, t) = 0$, we set $\eta(x, t) = \epsilon\eta_1(x, t) = \epsilon h(t)e^{i\alpha x}$. Our goal is to determine conditions for the decay of the amplitude $h(t)$, dependent on our choice of velocity in our explicit solve. The perturbed problem is linear, so we assume exponential decay, $h'(t) = \lambda h(t)$, and seek conditions for $\text{Re}(\lambda) < 0$.

To satisfy the conditions (2.5) at the boundary, we apply the same calculation as that for determining well-posedness where specifically we choose $\mathbf{f} = 0$, the perturbation now being in the boundary location rather than in a driving function at the boundary. With this change, the $\mathcal{O}(\epsilon)$ equation (2.16) becomes

$$\mathbf{G}\mathbf{K}\mathbf{c} + h\mathbf{G}\tilde{\mu}_0 = 0. \quad (2.26)$$

Solving m of these equations determines \mathbf{c} . We set the error in the remaining

equation to equal $h'(t)$, which, to lowest order, is v_n , the normal velocity of the boundary. Our freedom now is in which equation, or linear combination of equations, we choose from equations (2.26) as the velocity for adjusting Γ at each iteration.

To lowest order, v_n equals the vertical boundary velocity, so

$$v_n = \frac{d\eta}{dt} = \epsilon e^{i\alpha x} \frac{dh}{dt}. \quad (2.27)$$

We specify our velocity choice by the $m+1$ -unit vector \mathbf{v} , so h obeys

$$\mathbf{v}^T \mathbf{G} \mathbf{K} \mathbf{c} + h \mathbf{v}^T \mathbf{G} \tilde{\mu}_0 = \frac{dh}{dt} = \lambda h. \quad (2.28)$$

For example, if the classical Stefan velocity (1.8) is used for the velocity in the ice-water problem, the velocity vector, $\mathbf{v} = [1, 0, 0]^T$. If instead we choose the error in $u^+ = u^-$ as the velocity, the velocity vector is $\mathbf{v} = [0, 1, 1]$. We examine these particular choices further in the computer examples in Chapter 4.

Finally, we isolate λ . If the conditions from \mathbf{G} used to solve the fixed-boundary problem are orthogonal to \mathbf{v} , using orthonormal matrices to manipulate \mathbf{v}^T and \mathbf{w}^T from equation (2.17) to eliminate $\mathbf{G} \mathbf{K}$ gives

$$\lambda = \frac{\mathbf{w}^T \mathbf{G} \tilde{\mu}_0}{\mathbf{w}^T \mathbf{v}}. \quad (2.29)$$

If the conditions from \mathbf{G} used to solve the fixed-boundary problem are not orthogonal to \mathbf{v} , we can write the collection of velocity conditions and fixed-boundary solve conditions as

$$\mathbf{M} = \begin{bmatrix} \mathbf{v}^T \\ \mathbf{A}^T \end{bmatrix}, \quad (2.30)$$

where the m columns of \mathbf{A} represent the combination of boundary conditions in \mathbf{G} used to compute the fixed-boundary problem. Note that \mathbf{M} is full rank. Defining $\mathbf{e}_1 = [1, 0, \dots, 0]^T$,

$$\lambda = \frac{\mathbf{w}^T \mathbf{G} \tilde{\mu}_0}{\mathbf{w}^T \mathbf{M}^{-1} \mathbf{e}_1}. \quad (2.31)$$

Details of this calculation, including the modification required when \mathbf{v} is not orthogonal to all of the fixed-boundary conditions, can be found in the appendix.

λ is a ratio of polynomials in $|\alpha|$. Since we require that all modes decay, λ must not change sign; restricting the numerator and denominator never to equal zero is sufficient to guarantee this. The numerator of (2.29) and (2.31) is exactly the expression (2.17), and, as we recall, depends only on features of the problem, namely the free boundary conditions, defined by \mathbf{G} and \mathbf{b} , derivatives of the base solution, $\tilde{\mu}_0$, which depend again on \mathbf{G} as well as on global flux conditions, and on the operator acting on \mathbf{u} , the Laplacian in this case, contained in \mathbf{K} and hence in \mathbf{w} . Requiring that the numerator never equal zero is the same as condition (2.17), a satisfying consistency in our analyses.

The denominator of λ depends on the choice made for the velocity. Since the sign of $\lambda(|\alpha|)$ indicates the stability of the method with the chosen velocity, equation (2.29) can be used to evaluate the effects of our velocity choice independently from the underlying problem. For a well-behaved denominator, one that does not change sign, we must choose a good scheme, which amounts to choosing a good velocity. Moreover, \mathbf{v} affects the order of the denominator and, hence, the order of growth of λ . We now discuss the effects of our velocity choice.

2.4 Good velocity choices

We are now in a position to present general rules for choosing good velocities, that is, those that are single-signed giving $\lambda = \mathcal{O}(1)$ in $|\alpha|$. Since the numerator of λ is fixed by the underlying problem, we reduce our discussion to asking when the denominator of equation (2.29) is of highest order. At the very least, this minimises the growth rate of $\lambda(|\alpha|)$. We can accomplish this by determining the elements of \mathbf{w} and choosing \mathbf{v} to select its highest order elements.

Consider the calculation of \mathbf{w} . Since the boundary conditions are linearly independent, we can row-reduce \mathbf{G} to be upper triangular without affecting the problem. The bottom row of such row-reduced \mathbf{G} is a pure Dirichlet condition. We write this row-reduced \mathbf{G} as a sum of its Neumann and Dirichlet components,

$$\begin{aligned}\mathbf{G} &= [\mathbf{N}, \mathbf{D}] = \mathbf{G}_N + \mathbf{G}_D, \\ \mathbf{G}_N &= [\mathbf{N}, \mathbf{0}], \mathbf{G}_D = [\mathbf{0}, \mathbf{D}],\end{aligned}\tag{2.32}$$

and do likewise for \mathbf{K} :

$$\begin{aligned}\mathbf{K} &= \mathbf{K}_N + \mathbf{K}_D, \\ \mathbf{K}_N &= \begin{bmatrix} \pm|\alpha|\mathbf{I} \\ \mathbf{0} \end{bmatrix}, \mathbf{K}_D = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix},\end{aligned}\tag{2.33}$$

where \mathbf{I} is the m -by- m identity matrix. (We have abused notation somewhat here, in that elements in the upper half of \mathbf{K} may be variously $+|\alpha|$ or $-|\alpha|$.) With these definitions, we observe that $\mathbf{G}_N\mathbf{K}_D = \mathbf{G}_D\mathbf{K}_N = 0$, so

$$\mathbf{G}\mathbf{K} = \mathbf{G}_N\mathbf{K}_N + \mathbf{G}_D\mathbf{K}_D = \pm|\alpha|\mathbf{N} + \mathbf{D}\tag{2.34}$$

(again with an abuse of notation for $\pm|\alpha|$), where not only is \mathbf{N} upper triangular, but, because \mathbf{G} is row-reduced, the bottom row of \mathbf{N} is zero. With this in mind, we see that $\mathbf{G}\mathbf{K}$ has at most m elements of $\mathcal{O}(|\alpha|)$, none of these elements appearing in its bottom row. Thus, its left nullspace, \mathbf{w} , which by definition is orthogonal to its columnspace, has elements of at most $\mathcal{O}(|\alpha|)$ in all but its bottom row, where it is $\mathcal{O}(|\alpha|^q)$, $1 \leq q \leq m$; q is the number of different $\mathcal{O}(|\alpha|)$ terms in $\mathbf{G}\mathbf{K}$. We see that the numerator of λ is in general $\mathcal{O}(|\alpha|^q)$, and its denominator is also $\mathcal{O}(|\alpha|^q)$ as long as $\mathbf{v} = [0, 0, 1]^T$. We revisit our two

examples, the ice-water problem, and the feedback problem, to illustrate this calculation.

Recall that for the ice-water problem,

$$\mathbf{G} = \begin{bmatrix} 1 & -a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

and $\mathbf{w} = [1, |\alpha|, a|\alpha|]^T$. We see the highest-order elements of \mathbf{w} in the positions corresponding to the rows of \mathbf{G} representing pure Dirichlet conditions. A choice for \mathbf{v} that maximises the denominator of λ will be non-zero in at least one of its last two elements. Note that ice-water problem is somewhat unique among problems of two dependent variables in that it has not one, but two pure Dirichlet conditions at the free boundary. This means that there are two highest-order elements of \mathbf{G} , and hence two basis directions for \mathbf{v} that maximise the order of the denominator. Although one consequence of this is that the denominator can be at most $\mathcal{O}(|\alpha|)$ rather than $\mathcal{O}(|\alpha|^2)$, as is the case for the general $m = 2$ problem, so, too, is the numerator limited to $\mathcal{O}(|\alpha|)$, the order of the highest-order elements of \mathbf{w} .

For example, if we calculate the stiffness of the trial method using the Neumann and Dirichlet velocities introduced on page 11, we find that

$$\lambda = 2|\alpha| \quad (2.35)$$

for the Neumann velocity, where $\mathbf{v} = [1, 0, 0]^T$, and

$$\lambda = \frac{2\sqrt{2}}{1+a} \quad (2.36)$$

for the Dirichlet velocity, where (normalised) $\mathbf{v} = [0, 1/\sqrt{2}, 1/\sqrt{2}]^T$. Indeed, the Neumann velocity is $\mathcal{O}(|\alpha|)$ and the Dirichlet velocity is $\mathcal{O}(1)$. Again, the effects of the wave-number dependence of the stiffness for each of these \mathbf{v} will be seen in the computer experiments in Chapter 4.

For a second example, consider the unconditionally well-posed (negative) feedback problem, where $\delta^+ = -1$ and $\delta^- = 1$, so that

$$\mathbf{G} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & a & 0 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad (2.37)$$

and

$$\mathbf{w} = \begin{bmatrix} a|\alpha| + 1 \\ -(|\alpha| + 1) \\ -(|\alpha| + 1)(a|\alpha| + 1) \end{bmatrix}. \quad (2.38)$$

The highest order element of \mathbf{w} again corresponds to the problem's Dirichlet condition in the third row of \mathbf{G} , and choosing a velocity such that \mathbf{v} is non-zero in its third element is enough to ensure that the denominator is the same order as the numerator in $|\alpha|$ and, hence, that $\lambda(|\alpha|) = \mathcal{O}(1)$ for large $|\alpha|$.

Thus, regardless of the possibility that there may be more than one pure Dirichlet condition represented in \mathbf{G} , the preceding argument and example indicate that using Dirichlet conditions from the row-reduced \mathbf{G} as the velocity gives a λ that is $\mathcal{O}(1)$ in $|\alpha|$. This is a general rule:

In order to guarantee that λ be $\mathcal{O}(1)$, a Dirichlet condition must be included in the velocity.

A similar analysis shows that choosing pure Neumann conditions for the velocity gives an $\mathcal{O}(|\alpha|^{q-1})$ denominator, $1 \leq q \leq m$ as defined following (2.34), whereupon $\lambda = \mathcal{O}(|\alpha|)$. For a fixed step size, the method becomes unstable for some sufficiently large wave-number, or, equivalently, for some sufficiently fine discretisation. Indeed, the effect velocity choice has on stability manifests itself in previous work by Kärkkäinen and Tiihonen, where one of their (shape optimisation) algorithms fails to converge for certain fine discretisations [18]. Moreover, this behaviour is consistent with the well-known gain of regularity incurred in deducing Dirichlet boundary data from Neumann data. The good behaviour of Dirichlet velocities agrees with the converse result, that the Neumann-to-Dirichlet boundary map tends to improve data regularity. Regularity relationships between boundary data are well-known in the inverse problems community, cf. [29], a direct consequence of the trace theorem for functions in Sobolev spaces, cf. [2, ch 7].

Garabedian's approach to the Riabouchinsky flow problem (1.3a-1.3d) is to use $\partial_n u + \kappa u = 1$ to solve the fixed-boundary problem. If we replace the differential equation (1.3a) with $\Delta\psi(x, r) = 0$, and linearise around $r = R = 1$ (this changes the LHS of (1.3d) to 1), we have a paradigm problem similar to that treated by Garabedian that fits our analysis. In Garabedian's approach, κ is the curvature (not the stiffness), and can be considered constant to first order in our linearised treatment. (Garabedian makes this choice because this is a boundary condition that does not change under small normal perturbations of Γ .) We set the error in u (actually, $-u$ is the correct direction) as the velocity. The problem is one-sided ($m = 1$) so we solve for u in the half-plane $r \geq 1$ and take $\mathbf{K} = [-|\alpha|, 1]^T$. \mathbf{G} is the identity matrix, $\mathbf{b} = [-1, 0]^T$ (the negative sign is a result of our choice of \hat{n} as the upward, not outward, normal). Note that $m = 1$ is a special case for which \mathbf{G} has no nullspace. Instead, we must have nonzero \mathbf{b} so that there is a nonzero particular solution, $\tilde{\mathbf{r}}_P$. In this example, $\tilde{\mathbf{r}} = \tilde{\mathbf{r}}_P = \mathbf{b}$, giving $\tilde{\mu}_0 = [0, 1]^T$. Garabedian's approach amounts to

$$\mathbf{M} = \begin{bmatrix} 0 & -1 \\ 1 & \kappa \end{bmatrix}. \quad (2.39)$$

Using equation (2.31), we have

$$\lambda = \frac{-|\alpha|}{\kappa + |\alpha|}, \quad (2.40)$$

and in Garabedian's method, λ is bounded for large wave-numbers. Instead, if we consider the original approach, where $u = 0$ is applied in the fixed boundary

solve and the error in u_n is used to adjust the boundary,

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.41)$$

and

$$\lambda = -|\alpha|. \quad (2.42)$$

λ is unbounded with $|\alpha|$ for this standard velocity choice. If we use the Garabedian velocity on our linearised problem, the curvature of the base boundary is zero, so $\lambda = -1$ for all α . Referring back to our convergence discussion on page 14, we see that this gives a growth rate $G = 0$ with an optimal $k = 1$. This is exactly the value of k that Garabedian successfully uses in his method.

One point remains: while the numerator of λ will not change sign with $|\alpha|$ as long as the well-posedness condition in equation (2.17) is satisfied, it is possible that choices of \mathbf{v} give λ that change sign for $|\alpha| > 0$. We must be wary of this possibility in choosing \mathbf{v} . A random choice for \mathbf{v} , even one that maximises the denominator of λ , can destabilise the solve of an otherwise well-posed problem. It is possible to choose \mathbf{v} giving $\text{Re}(\lambda) > 0$ for all $|\alpha|$. However, these \mathbf{v} are straightforward to correct by choosing $-\mathbf{v}$ for the problem. Our concern, then, is limited to those \mathbf{v} that cause a change sign in $\mathbf{w}^T \mathbf{v}$, regardless whether the sign change is from negative to positive or positive to negative. We define \mathbf{v} to be *workable* if $\mathbf{w}^T \mathbf{v}$ does not change sign with $|\alpha|$.

With this caveat, we can construct plots of the behaviour of λ for each choice of \mathbf{v} . For example, if our problem has two dependent variables ($m = 2$), \mathbf{v} is a 3-vector, and, normalised, sits on the unit sphere in \mathbb{R}^3 . Figures 2.2 and 2.3 plot the effect of velocity choice on the workability and stability of solutions to the ice-water system and unconditionally well-posed feedback problem, respectively. The axes for these plots represent the radial and azimuthal angles of \mathbf{v} and the area shading of the plots indicates the effect of \mathbf{v} on each problem. The velocity choice plots show that in the case of the ice-water problem, some combination of the pure Dirichlet velocities, $\mathbf{v} = a[0, 0, \pm 1]^T + b[0, \pm 1, 0]^T$, gives a method that is $\mathcal{O}(1)$ and workable. In fact, the only workable $\mathcal{O}(|\alpha|)$ velocity is the classical Stefan velocity! In the case of the negative feedback problem, the method using the pure Dirichlet velocity, $\mathbf{v} = [0, 0, \pm 1]^T$, again gives an $\mathcal{O}(1)$ method that is workable, as do combinations of the Dirichlet and Neumann conditions. While there are relatively few unworkable velocities for the negative feedback problem, identifying which velocities to avoid would be difficult without the linear perturbation theory. In Garabedian's example, represented by equation (2.40), it happens that the curvature κ is always positive, so his method is workable.

In all three cases presented here, the ice-water problem, the negative feedback problem, and Garabedian's example, the pure Dirichlet velocity gives a workable method having $\mathcal{O}(1)$ stiffness. This suggests the conjecture that for well-posed problems, choosing the pure Dirichlet condition for the velocity *always* gives a workable $\mathcal{O}(1)$ trial method. Such a result would be especially useful for cases where $m > 2$, and the construction of two-dimensional stability maps such as figures 2.2 and 2.3 is not possible. Unfortunately, this conjecture

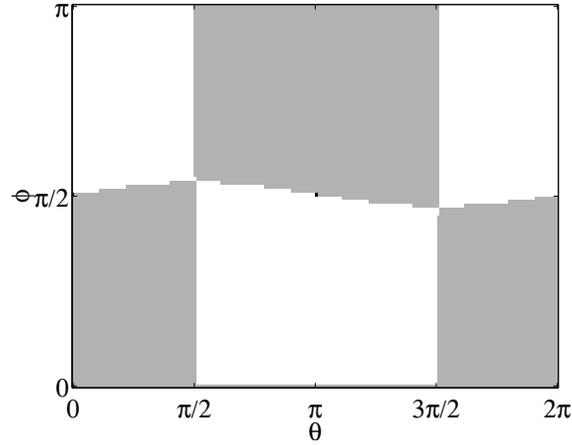


Figure 2.2: Effect of \mathbf{v} on the ice-water problem. $a = 10$. White areas represent unworkable \mathbf{v} , grey areas represent workable \mathbf{v} where $\lambda = \mathcal{O}(1)$, and the velocities $\mathbf{v} = [\pm 1, 0, 0]^T$, not distinguishable at this plot resolution, are workable \mathbf{v} where $\lambda = \mathcal{O}(|\alpha|)$.

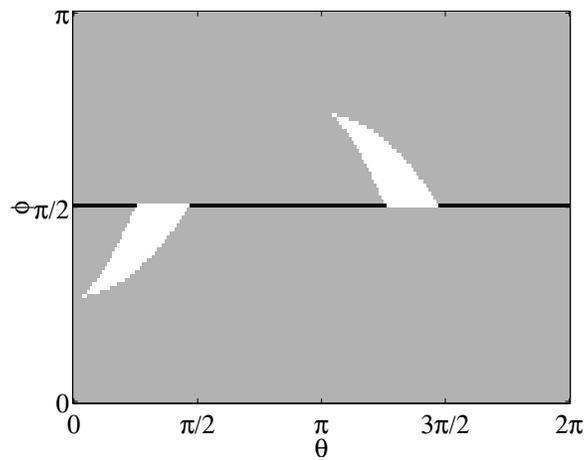


Figure 2.3: Effect of \mathbf{v} on the unconditionally well-posed negative feedback problem. $a = 10$. White areas represent unworkable \mathbf{v} , grey areas represent workable \mathbf{v} where $\lambda = \mathcal{O}(1)$, and black areas represent workable \mathbf{v} where $\lambda = \mathcal{O}(|\alpha|)$.

is at this time unproven, and with the results presented here, we can only test given velocity choices for their workability and effect on stiffness.

The treatment of the free boundary problem thus far has assumed infinite domains where Γ may support an infinite collection of wave-modes. In a practical numerical solve, however, Γ will be represented by a set of points on a finite domain. While our analysis of stiffness and stability is indicative of the behaviour of a practical scheme, before proceeding to the computer examples, we must analyse stability in the context of the discretised problem. Fortunately, with some adaptation, our work on the stability of the infinite linearised problem applies.

Chapter 3

Analysis of the Implementation

Although velocity choice is our primary point of discussion, an implementation of the explicit scheme requires further treatment of the boundary in the solve. Γ is represented by discrete points, with a spline fit to represent the continuous boundary. This lends some independence between the numerics of the free boundary adjustment and the numerics of the PDE solves in D^+ and D^- . In this way, we analyse a semi-discrete scheme in which we consider the PDE to be solved exactly. In addition, we smooth the data used in computing the local velocity of Γ at each iteration by averaging the PDE solutions (and their derivatives normal to Γ) by appropriate windowing functions (also called mollifying functions) along the arc length of the spline. Both of these operations, spline fitting and smoothing by windowing, can be treated as signal filtering processes. Windowing, especially, is an integral part of signal analysis, and is treated in the engineering literature as low-pass filtering (low-frequency signal selection). In light of existing tools that treat signals in the frequency domain, we make extensive use of Fourier transform space in the succeeding analysis, consistent with our frequency-dependent treatment of stiffness. This section both i) verifies that the spline fit and smoothing operations do not destabilise schemes deemed otherwise stable by the preceding linear stability analysis; and ii) shows that the stiffness predictions from the idealised linear analysis extend to a practical implementation having a discretised boundary. Our domain is now 2π -periodic in x .

3.1 Fourier transform of the cubic spline

Weighing the costs of implementation complexity and the effective benefits, we choose a piecewise cubic polynomial spline fit to approximate Γ . Since our linear stability criterion (2.29) assumes an infinite spectrum of modes carried by Γ , we first seek a relationship between the necessarily finite modes carried by the discretisation of the free boundary, and the modes carried by the spline fit of the discretisation. We continue to assume that Γ may be represented as a discretisation of a function $y = \eta(x)$. However, $\eta(x)$ is now a 2π -periodic function in x .

We discretise $y = \eta(x)$ by N points (x_j, y_j) evenly spaced by $\Delta x = x_{j+1} - x_j$

so that its 2π -periodic cubic spline interpolation is

$$y(x) = \sum_{j=0}^{N-1} y_j(x) \mathcal{X}_{[x_j, x_{j+1})}(x), \quad (3.1)$$

$$y_j(x) = A_j y_{j+1} + B_j y_j + C_j y_{j+1}'' + D_j y_j'',$$

where $\mathcal{X}_{[x_j, x_{j+1})}(x)$ is the characteristic function for the interval $[x_j, x_{j+1})$, and

$$A_j = \frac{x-x_j}{\Delta x} \quad B_j = 1 - A_j = \frac{x_{j+1}-x}{\Delta x}, \quad (3.2)$$

$$C_j = A_j^3 - A_j, \quad D_j = B_j^3 - B_j.$$

This representation of the cubic spline fit is taken from [22]. For a fixed discretisation, $[x_j]$, $y(x)$ depends linearly on the parameters $[y_j]$ and $[y_j'']$. However, $[y_j'']$ is the solution to the linear system $\mathbf{E}[y_j''] = \mathbf{F}[y_j]$, where, for periodic Γ ,

$$\mathbf{E} = \begin{bmatrix} 4 & 1 & & 1 \\ 1 & 4 & 1 & \\ & & \ddots & \\ 1 & & 1 & 4 \end{bmatrix}, \quad (3.3)$$

$$\mathbf{F} = \begin{bmatrix} -2 & 1 & & 1 \\ 1 & -2 & 1 & \\ & & \ddots & \\ 1 & & 1 & -2 \end{bmatrix}, \quad (3.4)$$

are N -by- N matrices, so $y(x)$ depends linearly on the data $[y_j]$. We wish to compute the Fourier transform of $y(x)$. Since this transform is also linear, we represent the data as a sum of its discrete Fourier modes,

$$y_j = \sum_{n=0}^{N-1} \tilde{y}(n) W_N^{nj}, \quad (3.5)$$

where $W_N^{nj} := \exp(i\frac{2\pi}{N}nj)$ is the kernel of the discrete Fourier transform. We compute the Fourier transform of $y(x)$ with the specific data $s_j = W_N^{nj}$ and sum the results over all modes, n . Now, since s_j are also eigenvectors of \mathbf{E} , bF , we can assume $y_j'' = b W_N^{nj}$, where b is determined from the linear solve,

$$\begin{aligned} E[b W_N^{nj}] &= F[W_N^{nj}], \\ (W_N^{-n} + 4 + W_N^n)[b W_N^{nj}] &= (W_N^{-n} - 2 + W_N^n)[W_N^{nj}], \\ b &= \frac{\cos(2\pi n/N) - 1}{\cos(2\pi n/N) + 2}. \end{aligned} \quad (3.6)$$

We wish to compute the Fourier transform of the 2π -periodic spline fit $y(x)$. From the variety of definitions for the Fourier transform, we choose

$$\hat{y}(n) = \frac{1}{2\pi} \int_0^{2\pi} y(x) e^{-inx} dx. \quad (3.7)$$

The interpolation (3.1) is C^2 in its 2π -periodic domain, so we can integrate by parts, observing that $y(0) = y(2\pi)$, $y'(0) = y'(2\pi)$, and $y''(0) = y''(2\pi)$. Now,

$$\hat{y}(n) = \frac{1}{2\pi} \frac{1}{(in)^3} \int_0^{2\pi} y'''(x) e^{-inx} dx. \quad (3.8)$$

Furthermore, $y(x)$ is C^∞ a.e., that is, except at the finitely many data points. So with convergence of the Fourier series for $y(x)$, we compute its continuous Fourier transform piecewise:

$$\hat{y}(n) = \frac{1}{2\pi} \frac{1}{(in)^3} \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} y_j'''(x) e^{-inx} dx, \quad (3.9)$$

where from the definition of the cubic spline,

$$y_j'''(x) = \frac{6}{(\Delta x)^3} (y_{j+1}'' - y_j''). \quad (3.10)$$

With our pure Fourier data, substitute $y_j'' = b W_N^{nj}$, so

$$y_j'''(x) = \frac{6b}{(\Delta x)^3} (W_N^n - 1) W_N^{nj} = \text{const.} \quad (3.11)$$

Evaluation of the integrals is now trivial, and, after some manipulation, the Fourier transform of the cubic spline fit of a pure Fourier mode is

$$s_N(n) = s(n/N) = \frac{3}{\cos(2\pi n/N) + 2} \frac{\sin^4(\pi n/N)}{(\pi n/N)^4}. \quad (3.12)$$

Details of the calculation of the Fourier integral can be found in Appendix A.3.

Given the Fourier representation of the data as in (3.5), we can write the Fourier transform of the cubic spline fit of general data as

$$\hat{y}(n) = \frac{3}{\cos(2\pi n/N) + 2} \frac{\sin^4(\pi n/N)}{(\pi n/N)^4} \tilde{y}(n) = s_N(n) \tilde{y}(n), \quad (3.13)$$

and the behaviour of $s_N(n)$ indicates how high-order modes not present in the discrete data appear in the continuous cubic spline fit. This relationship assumes that n takes all integer values, yet $\tilde{y}(n)$ is defined only for $0 \leq n \leq N-1$. Thus, we make sense of (3.13) by noting that W_N^{jn} is N -periodic, so we define $\tilde{y}(n - rN) = \tilde{y}(n)$ for integer r .

Equation (3.13) shows that stability depends on the response of our choice of generalised Stefan velocity to all modes in $y(x)$, not just the low-order modes in the discretisation $[y_j]$. Three qualities of $s_N(n)$ are immediately apparent: i) for $n \rightarrow \infty$, $s_N(n) \sim \mathcal{O}(n^{-4})$; ii) $s_N(n)$ is non-negative; and iii) the argument of $s_N(n)$ is the ratio of the mode number to the number of points in the discretisation, and does not depend on n explicitly. The first quality is a consequence of the degree of smoothness of the piecewise cubic fit, and serves as a check of the calculation; the remaining two features are important for the stability analysis that follows. Figure 3.1 gives a sketch of $s_N(n)$.

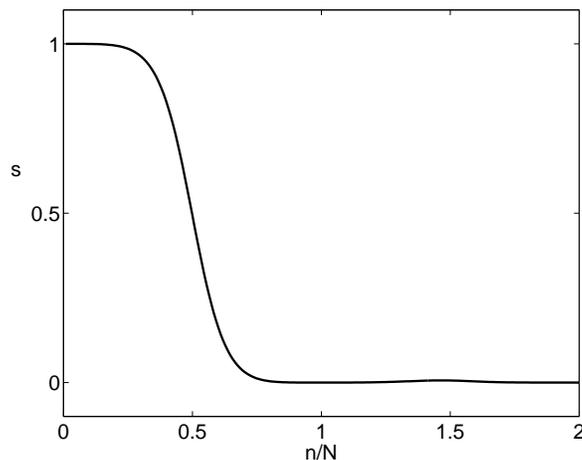


Figure 3.1: A sketch of $s_N(n) = s(n/N)$, showing its non-negativity and rapid decay. $s_N(n)$ is even, so only $n/N \geq 0$ is shown.

3.2 The cubic spline in generalised Stefan schemes

Using the Fourier transform of the cubic spline fit, we identify the continuous spectrum in $|\alpha|$ from Chapter 2 with our discrete spectrum in n , and seek a decay rate analogous to $\lambda(|\alpha|)$ that indicates the stability of the scheme. At each iteration in the computation, we determine the error in the residual boundary condition at each x_j , and adjust y_j proportional to this error. As determined by the linear stability analysis, the adjustment to each point of Γ is proportional to $\lambda(|\alpha|)$ for each pure wave-mode, so on our 2π -periodic domain, the total adjustment is the sum over integer modes,

$$\dot{y}(x) = \sum_{n=-\infty}^{\infty} \lambda(n) \hat{y}(n) e^{inx}. \quad (3.14)$$

The notation $\dot{y}(x)$ is suggestive of the velocity of the boundary. $\hat{y}(n)$ is again the continuous Fourier transform of the spline fit. As we are interested only in the adjustments of the points (x_j, y_j) , we recognise $x_j = \frac{2\pi}{N}j$ and rewrite (3.14) so

$$\dot{y}(x_j) = \sum_{r=-\infty}^{\infty} \sum_{n=0}^{N-1} \lambda(n - rN) \hat{y}(n - rN) W_N^{j(n-rN)}. \quad (3.15)$$

Next, using (3.13), we substitute for the Fourier transform of the spline fit. Recalling that at worst, $\lambda(|\alpha|) = \mathcal{O}(|\alpha|)$, and $s_N = \mathcal{O}(n^{-4})$, we are guaranteed

uniform convergence, and can exchange the sums. $W_N^{j(n-rN)}$ is N -periodic, so

$$\dot{y}(x_j) = \sum_{n=0}^{N-1} \left(\sum_{r=-\infty}^{\infty} \lambda(n-rN) s_N(n-rN) \tilde{y}(n-rN) \right) W_N^{nj}. \quad (3.16)$$

We see that the expression in brackets is the DFT of $[\dot{y}(x_j)]$. By our definition, $\tilde{y}(n)$ is N -periodic, hence

$$\dot{\tilde{y}}(n) = \gamma_N(n) \tilde{y}(n), \quad (3.17)$$

where

$$\gamma_N(n) = \sum_{r=-\infty}^{\infty} \lambda(n-rN) s(n/N-r), \quad (3.18)$$

reverting to notation for s that reflects its single argument, n/N . Just as $\lambda(|\alpha|)$ does for continuous spectrum Γ , $\gamma_N(n)$ indicates the stability of the solve with spline fit data having N points in its discretisation.

We can use equation (3.18) to assess the stability of our scheme. If $\lambda \sim \mathcal{O}(|\alpha|^q)$, then

$$\lambda(n-rN) \sim A N^q |n/N-r|^q, \quad (3.19)$$

for some constant, A , and

$$\gamma_N(n) \sim A N^q \sum_{r=-\infty}^{\infty} |n/N-r|^q s(n/N-r). \quad (3.20)$$

Here we observe the benefits of the spline fit in $s(n/N)$. $s \geq 0$ for all n , so if our Stefan velocity gives $\lambda < 0$, $\gamma < 0$ also. Since $s(n/N)$ decays as $\mathcal{O}(n^{-4})$, the sum only converges as long as $q < 3$. However, for our ice-water example, $q \leq 1$, so $\gamma_N(n) \sim \mathcal{O}(N^q)$, just as $\lambda(|\alpha|) \sim \mathcal{O}(|\alpha|^q)$. The behaviour of the spline fit implementation coincides with that of the problem in the idealised setting.

3.3 Smoothing Stefan velocity data

Although we now have rules for generalised Stefan schemes that will work in practise, we have neglected an important element of the method: the PDE solve of the domains D^\pm that occurs at each time step introduces error to the velocity computation. The velocity computed directly from the PDE solution (and its derivatives normal to Γ) has an error according to the PDE solver used. One way of attenuating this error is to smooth solution data along the boundary by convolution of that data with a windowing function. (This technique is also called *mollification*.) With the framework of the spline fit implementation, we are in a position to incorporate smoothing into our analysis.

Convolution in real space becomes multiplication in Fourier space. Thus, with the appropriate modification, the velocity in (3.14) is

$$\dot{y}(x) = 2\pi \sum_{n=-\infty}^{\infty} \lambda(n) \hat{f}(n) \hat{y}(n) e^{inx}. \quad (3.21)$$

\hat{f} is the Fourier transform of the windowing function, and the factor of 2π comes from our definition of the Fourier transform (3.7) applied to convolution. With this modification,

$$\gamma_N(n) = 2\pi \sum_{r=-\infty}^{\infty} \lambda(n - rN) \hat{f}(n - rN) s(n/N - r). \quad (3.22)$$

We begin discussion of appropriate windowing functions with an example.

Although not computationally efficient, an example windowing function is the 2π -periodic Gaussian,

$$f(x) = \sum_{r=-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - 2\pi r)^2}{2\sigma^2}\right), \quad (3.23)$$

normalised such that the area under $f(x)$ on $[0, 2\pi]$ is 1. σ controls the degree of smoothing by $f(x)$. To see the effect of varying σ , we compute the Fourier transform \hat{f} according to our convention (3.7):

$$\hat{f}(n) = \frac{1}{2\pi} \exp\left(-\frac{n^2\sigma^2}{2}\right), \quad (3.24)$$

whereupon we see the usual spreading of data in the frequency domain as $f(x)$ narrows in real space. Details of this calculation, including a general result for computing the Fourier transform of periodic functions, can be found in Appendix A.4. We may consider the velocity data provided by the PDE solver to be the sum of an exact solution and noise. The spectrum of the noise depends on the PDE solver used, and the amplitude of the noise depends on the order error of the solver. While we cannot eliminate this error altogether, we see that the effect of the windowing function is to attenuate the high-frequency error. Of course, the smoothing function also attenuates high-frequency components of the solution; an ideal smoothing function attenuates only modes having error and solution values of the same magnitude.

We control the degree of smoothing by varying σ . In real space, σ represents the width in x over which the window averages data; in Fourier space, σ controls the largest frequencies retained by the scheme. Thus, we expect that low-frequency modes will remain unaffected by the smoothing. If we fix σ , in turn smoothing over a fixed x -interval, as we increase N , an ever smaller fraction of modes, n with respect to N , are left unaffected by smoothing, and we see dissipation of the fine detail we may wish to capture by a fine discretisation. Instead, we can set $\sigma = a \Delta x = a \frac{2\pi}{N}$, affecting smoothing over approximately a data points. Now,

$$\hat{f}(n - rN) = \frac{1}{2\pi} \exp\left(-2\pi^2 a^2 (n/N - r)^2\right), \quad (3.25)$$

and a constant fraction of low-frequency modes are relieved of dissipation, such that we can recover the fine boundary detail we desire. Here we explicitly see the trade-off between consistency and stability endemic to smoothing techniques.

With this windowing function, we have

$$\gamma_N(n) = \sum_{r=-\infty}^{\infty} \lambda(n - rN) \exp(-2\pi^2 a^2 (n/N - r)^2) s(n/N - r). \quad (3.26)$$

The incorporation of this rapidly-decaying smoothing function does more than damp error from the PDE solve; it also guarantees convergence of the sum. As previously mentioned, for the ice-water example, λ does not grow quickly enough relative to the cubic spline decay for convergence of the sum to be an issue. However, for higher-order problems (problems with many dependent variables, where $m > 2$) or lower-order splines (ie piecewise linear) applying smoothing can force convergence of $\gamma_N(n)$ and aid stability.

In addition to being easy to work with analytically, this example windowing function has several good practical features. Most importantly, its Fourier transform is everywhere positive. Hence, smoothing by windowing here cannot destabilise the solve by introducing sign changes in $\gamma_N(n)$. $f(x)$ has unit mass, so

$$\lim_{\sigma \rightarrow 0} \|v - f * v\|_{L_p} = 0, \quad v \in L_p, \quad (3.27)$$

where $*$ denotes convolution. $\hat{f}(n)$ is everywhere real, indicative of the even symmetry of $f(x)$: when averaging about a point, data to the left of that point should weigh equally with data to the right. $f(x)$ is periodic, consistent with the problem layout. Finally, the degree of smoothing, hence the error introduced, is easy to control.

However, our choice for $f(x)$ falls short in that it does not have compact support as is often desirable in computation. Furthermore, it may be desirable to choose smoothing functions tailored to the spectrum of the error produced by the specific PDE solver employed. In addition, we have chosen to smooth over x rather than boundary arc length. Because the perturbation is small in our analysis, the two metrics are nearly identical, but this will not be true in general and may be important for highly irregular boundaries. All the same, we ignore these shortcomings and use the Gaussian window to smooth over x in the example computations. For other windowing functions, the reader is invited to consult [16] and like texts on signal analysis. Theoretical treatments of windowing (mollifying) functions appear in many numerical texts, cf. [20].

With our practical scheme established, we proceed to the computer examples.

Chapter 4

Computer Examples

We now demonstrate the effect of velocity choice on the ice-water model system of Figure 1.7. For this problem, we use the free boundary conditions (2.2) for \mathbf{G} , the differential operator matrix \mathbf{K} representing the Laplacian, and adjust the boundary at each iteration by two different choices of velocities, \mathbf{v} . In particular, we wish to compare the velocities presented on page 11, a Neumann velocity (the classical Stefan velocity, in this case) and a velocity containing Dirichlet components. That is, we compare $\mathbf{v} = [1, 0, 0]^T$, which we refer to as the *Neumann* method, and $\mathbf{v} = [0, 1/\sqrt{2}, 1/\sqrt{2}]^T$, which we refer to as the *Dirichlet* method. As we have already investigated, there exist choices for \mathbf{v} that give unworkable methods. However, looking again at the map in Figure 2.2, we note that our velocity choices do not make for unworkable schemes.

Our domain is the rectangle $-\pi \leq x < \pi$, $-1 \leq y \leq 1$, with periodic boundary conditions at $x = \pm\pi$. Our boundary conditions are $u(x, -1) = -1$ on the bottom boundary, and

$$u(x, 1) = \begin{cases} \frac{3}{2} & -\frac{\pi}{2} \leq x \leq \frac{\pi}{2} \\ \frac{1}{2} & \text{elsewhere} \end{cases} . \quad (4.1)$$

We set $a = 10$ and our initial Γ as $y = 0$, discretised and cubic spline fit by N points $[(x_j, y_j)]$.

To solve the fixed boundary problem at each iteration, we use the two remaining free boundary conditions orthogonal to \mathbf{v} . For the Neumann method, we impose $u^+ = u^- = 0$, and for the Dirichlet method, we impose $u^+ = u^-$ and $u_n^+ - au_n^- = 0$. With these conditions imposed, at each iteration, we solve $\Delta u = 0$ in each subdomain using the commercial finite element method (FEM) package FEMLAB, version 2.2 produced by Comsol Inc [17]. The FEM solve uses first-order (piecewise linear) Lagrange elements on mesh elements having diameter $D \approx 0.3$ near $y = \pm 1$ and $D \approx 2\pi/3N$ near Γ . This ensures that to within the resolution of the boundary, our trial method uses a good approximation to the exact solution of $\Delta u^\pm = 0$ near Γ . We again emphasise the simplicity of this implementation: the use of this particular FEM package to solve the fixed-boundary problem is independent of our method of adjusting the boundary. Very little original code had to be written in the implementation of this scheme.

Following each FEM solve, we compute the residual $v(x)$ in the appropriate third boundary condition along the boundary, the residual in $u_n^+ - au_n^-$ for the Neumann method, and the residual in $(1/2)(u^+ + u^-)$ for the Dirichlet method. We smooth the residual by convolution of $v(x)$ with respect to x by

the normalised 2π -periodic Gaussian function of our example, equation (3.23). We choose $\sigma = \pi/N$, so that this convolution computes the smoothed residual $v_j(x)$ at each boundary point (x_j, y_j) by smoothing over approximately $b = 2$ data points to either side of each point in question. The algorithm moves each boundary point normally to Γ , so

$$(x_j^{n+1}, y_j^{n+1}) = (x_j^n, y_j^n) + kv_j \hat{n}, \quad (4.2)$$

where k is the time step size, and returns a cubic spline fit of this revised Γ to the succeeding FEM solve. Note that our implementation deviates from our analysis on this point, as the boundary is moved only in the y -direction in the analysis. The implementation is consistent with the more general case where Γ is not nearly flat.

With the boundary discretised and spline fit by $N = 50$ points, the largest stable time step for the Neumann method is $k = 0.01$, and for the Dirichlet method is $k = 0.2$. The smaller time steps required by the Neumann method corresponds to the stiffness predictions for the ice-water problem on page 27. Figures 4.1 and 4.2 for the Neumann and Dirichlet methods, respectively, illustrate the progress of these solves showing u in each subdomain, as well as the development of the free boundary, Γ . Note that each method solves a different problem, but converges to the same steady-state boundary. Furthermore, the initial boundary for each case is $y = 0$, a poor approximation to the solution, yet both methods converge.

The effects of the stiffness of each method can be seen as N is increased, which we observe by comparing our numerical solution with the exact solution. (Indeed, with this problem, we are in the position of having the exact solution to the free boundary problem at hand. Essentially, by setting $v(x, -1) = au(x, -1)$ and leaving $v(x, 1) = u(x, 1)$, we solve $\Delta v = 0$ in the entire domain. This change of variables in the lower subdomain ensures that the flux is continuous everywhere in the domain, and, thus, the boundary is represented by the level set $v = 0$. A closed-form solution exists for this problem for v , and $v = 0$ can be written as an implicit relationship between x and y . Details of this calculation appear in Appendix A.5.)

Figure 4.3 compares the Neumann and Dirichlet methods for a variety of N . The plots show the mean square error in the boundary points,

$$\text{error} = \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (y_j - y_{\text{exact}}(x_j))^2}, \quad (4.3)$$

at each iteration over a range of N taken from the set

$$N \in \{20, 50, 100, 200, 500\}. \quad (4.4)$$

Time steps k from the set

$$k \in \{1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, \dots\} \quad (4.5)$$

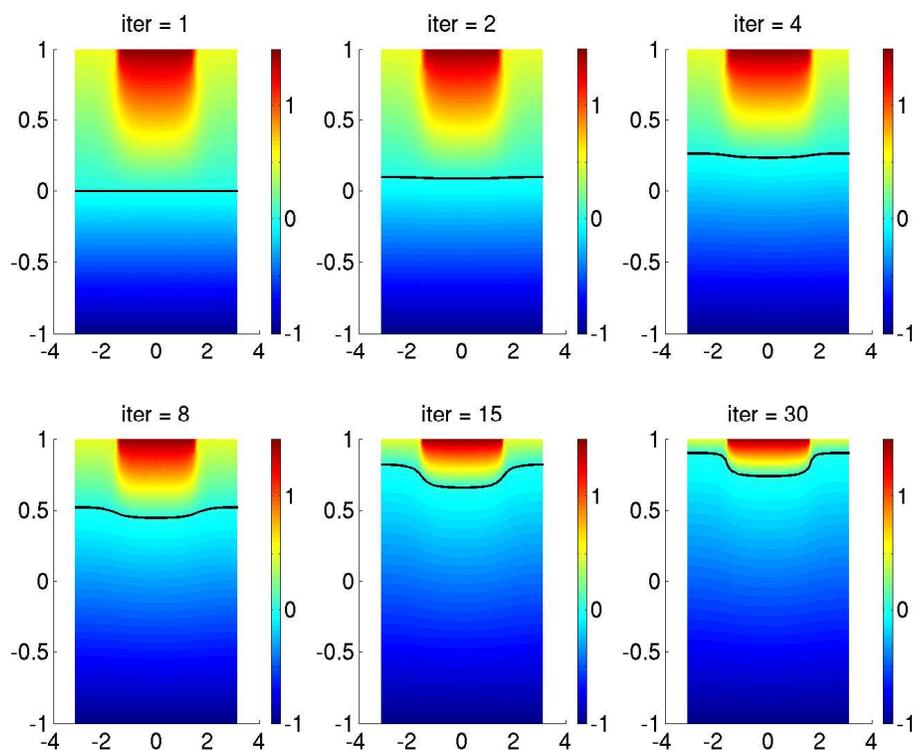


Figure 4.1: Evolution of the ice-water boundary by the Neumann method. The black line represents Γ while the colour data represents u . The initial boundary is $y = 0$.

were tested for each N and the error data for the stable method having the largest k for each given N is reported in the figure. Using the Dirichlet method, the same k can be used for each choice of N , whereas with the Neumann method, to maintain stability k must be decreased as N is increased, consistent with our analysis. As a result of this need to increase k , the number of iterations required to converge to the solution increases also. The maximum stable step sizes for each N are given in Table 4.1.

We note here that even at (mean) steady state, there is oscillation present in the Neumann method not present in the Dirichlet method. Figure 4.4 shows this oscillation in detail for $N = 10, 20, 50$. We see a reason for this oscillation by looking at the linearised velocity for our generalised Stefan schemes. Using equation (2.28), we have

$$v_n = -|\alpha|[1, a]\mathbf{c} \quad (4.6)$$

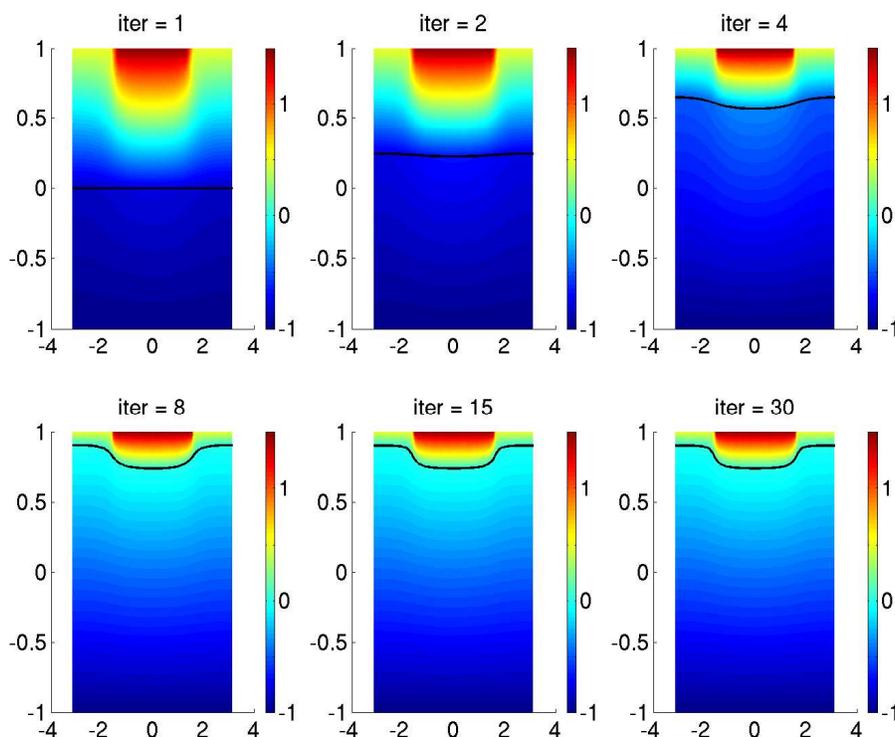


Figure 4.2: Evolution of the ice-water boundary by the Dirichlet method. Again, the black line represents Γ while the colour data represents u .

for the Neumann method, versus

$$v_n = -\frac{1}{\sqrt{2}}[1, 1]\mathbf{c} + \left(1 + \frac{1}{a}\right)h \quad (4.7)$$

for the Dirichlet method. The finite element method incurs high-order wave-modes in the solution, and although the magnitude of these wave-modes is small, the linear dependence on $|\alpha|$ of the Neumann method puts this method close to the limit of stability, resulting in oscillations. The Dirichlet method does not have linear $|\alpha|$ -dependence and hence does not suffer these oscillations.

Finally, we observe that although the error tends to zero as $N \rightarrow \infty$, we are ultimately limited by the accuracy of the FEM solve. The mean square error decays as $\mathcal{O}(N^{-1/2})$, supposedly consistent with the accuracy predictions for our finite element method. These estimates are future work. Accuracy as a function of N is shown in Figure 4.5. The accuracies of the two methods are comparable. However, for $N = 500$, while the Dirichlet method took approximately 3 hours to solve for the free boundary, the Neumann method took approximately 6 days

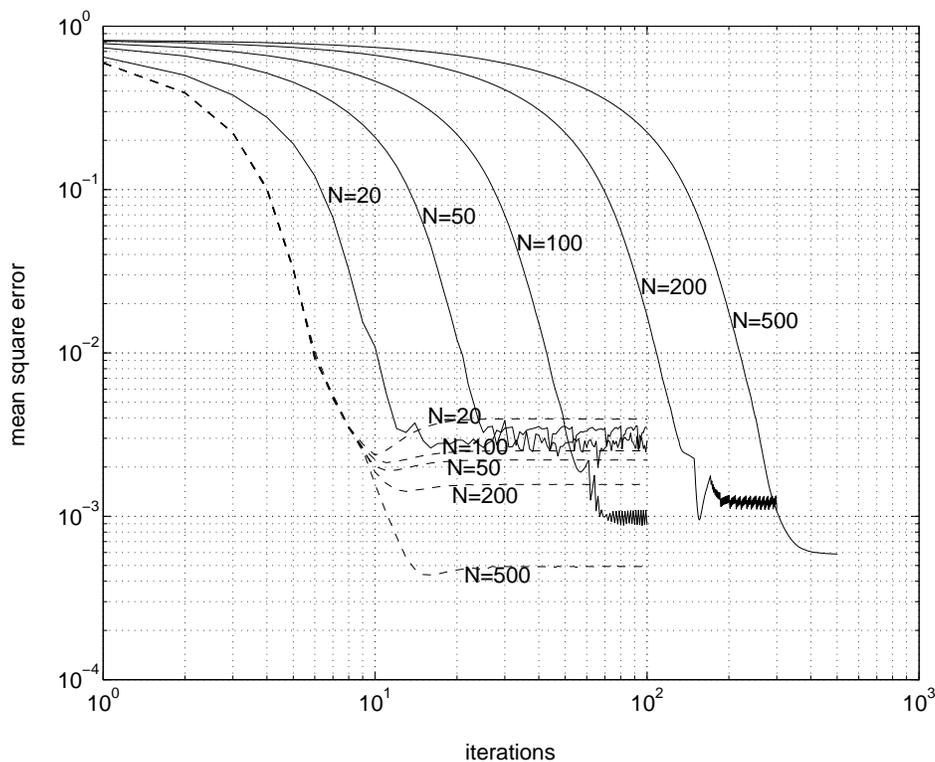


Figure 4.3: Plots of mean square error versus iteration number, showing the effect of N on the maximum size of k for each method. Solid lines represent the Neumann method; dashed lines represent the Dirichlet method. The oscillations for the Neumann method are discussed in the text.

N	maximum k	
	Neumann method	Dirichlet method
20	0.02	0.2
50	0.01	0.2
100	0.005	0.2
200	0.002	0.2
500	0.001	0.2

Table 4.1: Comparison of the Neumann and Dirichlet methods by the maximum stable step sizes k for each discretisation N .

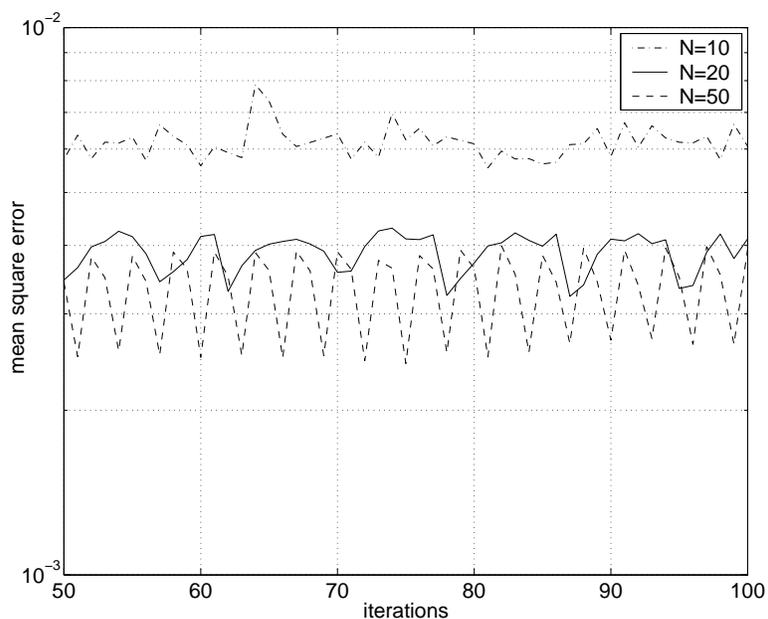


Figure 4.4: Close view of the oscillations in the solution by the Neumann method. Unlike Figure 4.3, the horizontal axis is linear.

running on the same machine. This observation is encouraging justification for using generalised Stefan velocities for solves where the governing PDEs might be more complicated.

The work culminating in these computer examples constitutes our main body of study. The central idea, that making good velocity choices can lead to superior numerical methods, has been verified in this example, and the way is now paved to apply these ideas in more general settings. This will be the course of the next and final chapter.

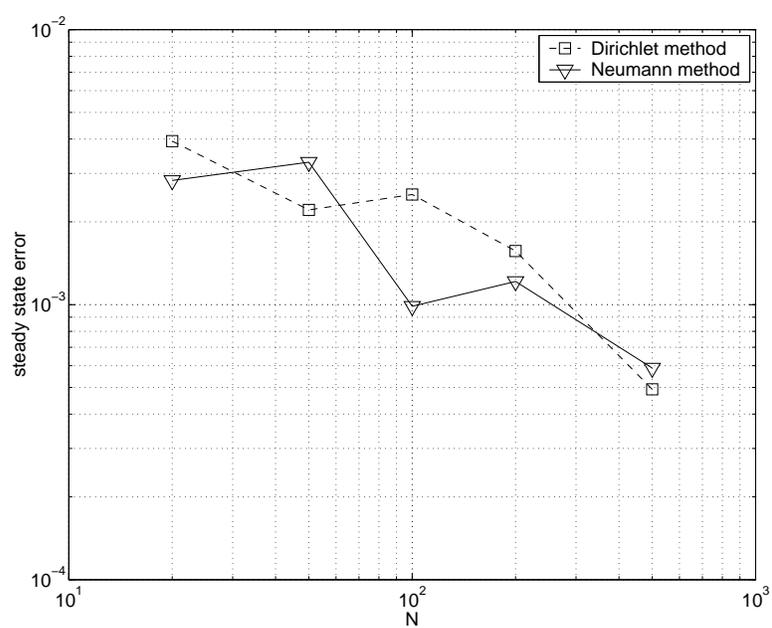


Figure 4.5: Comparison of the accuracy of the Neumann and Dirichlet methods for increasing N with the k in Table 4.1.

Chapter 5

Continuing Work

While one immediate goal of this project is to apply a generalised Stefan scheme to the porous media phase change problem presented in the introductory chapter, other work continues. The first part of this chapter discusses a level set method for free boundary adjustment that retains the good stiffness characteristics of the generalised Stefan velocity method. Level set methods have an advantage over the front tracking method employed thus far in that they do not require *a priori* knowledge of the free boundary topology. In Section 5.1, we treat a level set method with the flat boundary used in Chapter 2, then in Section 5.2, we consider both the front tracking and level set methods applied to more general boundaries. Section 5.3 concerns the well-posedness of boundary data near points where the free boundary contacts the physical boundary of the problem, an issue we have thus far avoided through the use of infinite and periodic domains. Finally, Section 5.4 summarises this work. Work presented in this chapter is motivated by the well-posedness of generalised Stefan problems and the workability of their numerical solutions.

5.1 Generalised Stefan velocities for a level set method

Our analysis has thus far assumed that one coordinate of our free boundary can be represented as a function of another coordinate, that is, for example, that in \mathbb{R}^2 , the boundary location is described by the points $(x, \eta(x))$. This will not be true in general, such as in computations on bubble formation and Hele-Shaw flow, cf. [19, ch 7], where the free boundaries (surfaces, in \mathbb{R}^3) enclose a region of the domain. Computing such free boundaries can be challenging, primarily because the topology of the enclosed regions can change, for example, when two bubbles merge, or a single bubble divides. In the case of steady-state solves, we must be concerned with making accurate initial guesses for the steady-state topology of the free boundary. In three (or more) dimensions, the front tracking technique used thus far for our trial method can be particularly difficult to implement stably, and so we present an alternative.

A successful strategy for dealing with free boundary problems, particularly those in which the free boundary topology changes over the course of the solve, is the level set method of Osher and Sethian. Their method was first presented in [21] for addressing curvature-driven boundary evolution, yet has been further developed, whereupon Sethian's text [23] treats the extension of the level set

idea to other techniques and applications. Instead of tracking the free boundary, the level set method solves for a function, $\phi(\mathbf{x}, t)$, that typically represents the signed distance from any point \mathbf{x} in the domain to the boundary. Hence, the level set $\phi = 0$ represents the boundary location. ϕ obeys an evolution equation that depends on the velocity of the boundary. Our goal in this section is to show that the evolution equation for ϕ inherits the desirable stiffness properties obtained by making good velocity choices for the corresponding front tracking method.

Osher and Sethian treat the time-dependent problem of front propagation in \mathbb{R}^n as an initial value problem for a function $\phi : [0, \infty) \times \mathbb{R}^n \rightarrow \mathbb{R}$. Again, the zero level set of ϕ represents the (moving) free boundary. We develop an evolution equation for ϕ beginning with

$$\phi(\mathbf{x}(t), t) = 0, \quad (5.1)$$

the set $\mathbf{x}(t)$ representing the free boundary. The t -dependence of \mathbf{x} represents the general tendency of the free boundary to change location with time. By the chain rule,

$$\phi_t + \nabla\phi(\mathbf{x}(t), t) \cdot \mathbf{x}'(t) = 0, \quad \mathbf{x} \in \Gamma \quad (5.2)$$

Now, the normal velocity of the boundary, $v_n = \mathbf{x}'(t) \cdot \hat{n}$. The gradient of a function is normal to its level sets, so $\hat{n} = \nabla\phi/|\nabla\phi|$. Thus, ϕ obeys

$$\begin{cases} \phi_t + v_n |\nabla\phi| = 0, & \mathbf{x} \in D \\ \phi(\mathbf{x}, t = 0) = f(\mathbf{x}), & \mathbf{x} \in D \end{cases} \quad (5.3)$$

Note that this level set method is a capturing method, and we have extended its evolution equation to the entire domain D , and thus removed the explicit dependence of \mathbf{x} on t to reflect this. v_n may depend on any combination of local, global and intrinsic properties of the system: for our development, we take v_n to be a generalised Stefan velocity of our trial method. This means that time t in (5.3) represents artificial iteration time, just as throughout our development of the front tracking trial method. Since ϕ must be solved for all \mathbf{x} , not just for those \mathbf{x} on Γ , and that v_n must also be known for all \mathbf{x} . Extending v_n to the entire domain from Γ may not be trivial, and we shall revisit this point at the end of this section. ϕ is not unique for a particular Γ : there are many functions that share the same zero level set while satisfying the evolution equation (5.3). This means that any number of initial $f(\mathbf{x})$ are permissible for the same free boundary problem. It is often desirable, however, to specify ϕ as a signed distance function, where ϕ is $+1$ times the distance from the free boundary on one side of Γ and -1 times the distance from the free boundary on the other side.

With this framework, we suggest an algorithm for using the level set method with generalised Stefan velocities to solve the steady state free boundary problem.

1. Choose an initial level set function. Its zero level set is the initial approximation to the free boundary.

2. Solve the resulting fixed boundary PDE using all but one of the free boundary conditions.
3. The remaining condition will not be satisfied unless the position of the free boundary is accurate: the residual in the remaining condition is the boundary velocity.
4. Extend the boundary velocity to the entire domain, and step the level set equation (5.3) forward. If the adjustment of the level set function is small, stop; otherwise, adopt the new zero level set as the boundary and return to step 2.

The level set method is touted for its ability to evolve a free boundary without knowledge of its location at every step. It may seem as though we nullify this particular advantage in that we must know the boundary location at each iteration in order to solve the fixed boundary PDE compute v_n . However, unlike front tracking, we do not need to follow the individual boundary points from one iteration to the next. This algorithm suggests extra work in everywhere computing a function and its velocities for which we desire only a small subset of its values (ie near $\phi = 0$), but we needn't pay too much attention to the accuracy of the solve for ϕ far from its zero level set. Furthermore, as previously mentioned, there may be great advantages higher dimensional applications, where the boundary parametrisation required for explicit front tracking of changing topologies is not trivial.

In order to ascertain the stiffness of the hybrid method, as an example, let us return to the linearised system considered in Chapter 2, where a base boundary separates upper and lower regions, D^+ and D^- , respectively. The Laplacian applies to the dependent variables above and below the free boundary, Γ . Instead of perturbing the base boundary, we choose a level set function describing the base boundary and perturb that base level set function. We shall reuse much of the notation from Chapter 2.

Let

$$\phi(x, y) = \phi_0(x, y) + \epsilon\phi_1(x, y), \quad (5.4)$$

where

$$\phi_0(x, y) = y, \quad (5.5a)$$

$$\phi_1(x, y) = h(t)g(y)e^{i\alpha x}. \quad (5.5b)$$

We have suppressed the t -dependence of ϕ for clarity. Although we shall discover that the perturbation amplitude of the eigenfunctions associated with Γ do not depend on y , we retain this freedom for the time being. We next determine how the perturbation in ϕ affects the boundary location. Assume again that

$$\eta \sim \eta_0 + \epsilon\eta_1 + \dots \quad (2.3)$$

Now we can compute the boundary location, $y = \eta(x)$, where $\phi(x, \eta(x)) = 0$. Again, we suppress the t -dependence of η for clarity. Computing term by term,

$$\phi_0(x, \eta(x)) = \phi_0(x, \eta_0(x)) + \epsilon\eta_1(x)\phi_{0y}(x, \eta_0(x)) + \mathcal{O}(\epsilon^2), \quad (5.6a)$$

$$\phi_1(x, \eta(x)) = \phi_1(x, \eta_1(x)) + \mathcal{O}(\epsilon). \quad (5.6b)$$

So,

$$\begin{aligned} 0 &= \phi(x, \eta(x)) = \phi_0(x, \eta(x)) + \epsilon\phi_1(x, \eta(x)) \\ &= \phi_0(x, \eta_0(x)) + \epsilon[\phi_1(x, \eta_0(x)) + \eta_1(x)\phi_{0y}(x, \eta_0(x))] + \mathcal{O}(\epsilon^2). \end{aligned} \quad (5.7)$$

Considering our choices for ϕ_0 and ϕ_1 , and matching terms, we have

$$\eta_0 = 0, \quad (5.8a)$$

$$\eta_1 = -h(t)g(0)e^{i\alpha x}. \quad (5.8b)$$

To within a change in sign, we have the same boundary perturbation we applied in the linear stability analysis of Chapter 2. Other choices for ϕ_0 which set $y = 0$ to as the base boundary are possible (for example, $\phi_0 = ay, a = \text{const}$), but this choice makes ϕ_0 a signed distance function. Thus, the boundary velocity is given by the eigenvalues for

$$\mathbf{v}^T \mathbf{G} \mathbf{K} \mathbf{c} - h(t)g(0)\mathbf{v}^T \mathbf{G} \tilde{\boldsymbol{\mu}}_0 = g(0) \frac{dh}{dt} = \lambda h(t)g(0), \quad (5.9)$$

where the sign of h on the LHS has been reversed from that in equation (2.28) according to the change in $\eta_1(x)$ by the level set setting. We can repeat exactly the same calculation as that in Chapter 2 to obtain

$$\lambda = -\frac{\mathbf{w}^T \mathbf{G} \tilde{\boldsymbol{\mu}}_0}{\mathbf{w}^T \mathbf{M}^{-1} \mathbf{e}_1}. \quad (5.10)$$

The decay for $h(t)g(0)$ differs only from that of the original linear analysis by a change in sign. Just as in Chapter 2, features of the problem, the linear operator, the global fluxes and the boundary conditions, appear in the numerator, and the affect of velocity choice appears in the denominator. Even in this new setting, λ remains dependent on our generalised Stefan velocity choice.

We require v_n in order to return from this result to the level set evolution equation (5.3). Recalling that

$$v_n = \frac{d\eta}{dt}, \quad (2.27)$$

and $d\eta/dt = \epsilon g(0)e^{i\alpha x} dh/dt$, we have $v_n = -\epsilon\lambda(\alpha)h(t)g(0)e^{i\alpha x}$, so ϕ evolves according to

$$\phi_t - \epsilon\lambda(\alpha)h(t)g(0)e^{i\alpha x} |\nabla\phi| = 0 \quad (5.11)$$

near $y = 0$. Numerically, (5.11) will be solved by a method explicit in time, and must obey a Courant-Friedrichs-Lewy (CFL) condition for stability of its PDE scheme. This condition depends directly on λ , contained in the leading term to the spacial derivative, and so we see that level set equation does indeed inherit stiffness properties according to our velocity choice, \mathbf{v} .

If we substitute $\phi = \phi_0 + \epsilon\phi_1$ into (5.11), and collect the $\mathcal{O}(\epsilon)$ terms (the $\mathcal{O}(1)$ terms are zero), we find that $g(y) = g(0)$ is the natural way to extend the

velocity away from Γ for the linearised problem. However, this analysis assumes that the amplitude, $h(t)$, decays uniformly throughout the domain. If, instead, we replace $h(t)g(y)$ with the single amplitude $h(t, y)$, we see that more generally, the stiffness of the level set solve is related to how the velocity is extended from the Γ to the entire domain. Such freedom may be more convenient numerically, especially in more complicated domains, and we expect the method of velocity extension to have an effect on the problem stiffness. Velocity extension schemes are presented both in the original paper [1], and in the text [23]. It is not known if such schemes will cause the stiffness of the level set equation to deviate significantly from the predictions for the associated velocity choice. For the time being, such study remains future work.

5.2 Extensions to general operators and base boundaries

Before leaving the level set framework, we carry the analysis of Chapter 2 one step further, and consider the behaviour of generalised Stefan schemes applied to a larger set of boundaries and (elliptic) operators. Such extension is motivated by the ability of level set functions to describe boundaries that may not be representable as functions $y = \eta(x)$.

Before incorporating the level set method, we return to the basic front tracking trial method, and consider the stability of perturbations to more general boundaries. We consider the base boundary location \mathbf{x}_0 parametrised by \mathbf{s} . (Our boundaries all have co-dimension 1, so $\dim(\mathbf{s}) = \dim(\mathbf{x}) - 1$.) Perturb the free boundary,

$$\mathbf{x} = \mathbf{x}_0(\mathbf{s}) + \epsilon \mathbf{x}_1(\mathbf{s}), \quad (5.12)$$

Just as in Chapter 2, we seek perturbation solutions,

$$\mathbf{u} \sim \mathbf{u}_0 + \epsilon \mathbf{u}_1 + \dots \quad (5.13)$$

Again, we use Taylor series to approximate conditions at the perturbed boundary by conditions at the base boundary, so

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \mathbf{u}(\mathbf{x}_0 + \epsilon \mathbf{x}_1), \\ &= \mathbf{u}(\mathbf{x}_0) + \epsilon \mathbf{x}_1 \cdot \nabla \mathbf{u}(\mathbf{x}_0), \\ &= \mathbf{u}_0(\mathbf{x}_0) + \epsilon [\mathbf{u}_1(\mathbf{x}_0) + \mathbf{x}_1 \cdot \nabla \mathbf{u}_0(\mathbf{x}_0)] + \mathcal{O}(\epsilon^2), \\ &= \mathbf{u}_0(\mathbf{s}) + \epsilon [\mathbf{u}_1(\mathbf{s}) + \mathbf{x}_1(\mathbf{s}) \cdot \nabla \mathbf{u}_0(\mathbf{s})] + \mathcal{O}(\epsilon^2). \end{aligned} \quad (5.14)$$

The last equality follows because all boundary information can be parametrised by \mathbf{s} . Using the matrix representation, \mathbf{G} , for the free boundary conditions, we assume that the base solution is satisfied, so $\mathbf{G}\mathbf{u}_0(\mathbf{x}_0) = \mathbf{b}$. To satisfy the $\mathcal{O}(\epsilon)$ condition, we need the normal derivative,

$$\mathbf{u}_n(\mathbf{x}) = \mathbf{u}_n(\mathbf{x}_0 + \epsilon \mathbf{x}_1),$$

$$\begin{aligned}
&= \mathbf{u}_{0n}(\mathbf{x}_0) + \epsilon[\mathbf{u}_1(\mathbf{x}_0) + \mathbf{x}_1 \cdot \nabla \mathbf{u}_0(\mathbf{x}_0)]_n + \mathcal{O}(\epsilon^2), \\
&= \nabla \mathbf{u}_0(\mathbf{x}_0) \cdot \hat{n} + \epsilon[\nabla \mathbf{u}_1(\mathbf{x}_0) \cdot \hat{n} + \mathbf{x}_1 \cdot \hat{n} \Delta \mathbf{u}_0(\mathbf{x}_0)] + \mathcal{O}(\epsilon^2), \\
&= \nabla \mathbf{u}_0(\mathbf{s}) \cdot \hat{n} + \epsilon[\nabla \mathbf{u}_1(\mathbf{s}) \cdot \hat{n} + \mathbf{x}_1(\mathbf{s}) \cdot \hat{n} \Delta \mathbf{u}_0(\mathbf{s})] + \mathcal{O}(\epsilon^2). \quad (5.15)
\end{aligned}$$

Next, we generalise the elliptic operator. In each region D^\pm sharing the free boundary Γ , we solve

$$\begin{cases} L\mathbf{u}_1 = 0 & \text{in } D^\pm \\ \mathbf{u}_1 = \psi_\alpha(\mathbf{s}) & \text{on } \Gamma \end{cases}. \quad (5.16)$$

We expect far-field conditions, such as global flux conditions, on the problem to be satisfied by the base solution, so $\mathbf{u}_1 \rightarrow 0$ far from Γ . As long as L is linear and separable¹ in the boundary coordinates, \mathbf{s} , this reduces the solution to a linear combination of solutions whose values and normal derivatives at the boundary are represented by

$$\tilde{\mathbf{u}}_1 = \psi_\alpha(\mathbf{s})\mathbf{K}\mathbf{c}. \quad (5.17)$$

\mathbf{K} again gives us the relationship between the solutions near Γ , given by ψ_α and $\psi_{\alpha n}$. In the case of the Laplacian with base boundary $\eta(x) = 0$, we had

$$\mathbf{K} = \begin{bmatrix} -|\alpha| & 0 \\ 0 & |\alpha| \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.18)$$

but in general, we will have

$$\mathbf{K} = \begin{bmatrix} \varphi_1(\alpha) & & & & \\ & \varphi_2(\alpha) & & & \\ & & \ddots & & \\ & & & \varphi_m(\alpha) & \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & & 1 \end{bmatrix}. \quad (5.18)$$

$\{\varphi_i(\alpha)\}$ represent the ratios of the normal derivatives the solutions $\mathbf{u}_{i\alpha}(\mathbf{s})$ to their values at the base boundary. $\{\varphi_i(\alpha)\}$ depend on the operator L and the base boundary location, and in effect describe the smoothness of the Dirichlet-to-Neumann map at Γ . Note that in this more general setting, $\{\varphi_i(\alpha)\}$ may be complex, yielding complex λ . However, the stability criterion illustrated by Figure 1.8 still applies. The appropriate perturbation,

$$\mathbf{x}_1 = h(t)g(\mathbf{s})\psi_\alpha(\mathbf{s})\hat{n}(\mathbf{s}), \quad (5.19)$$

¹Separability as we saw for the Laplacian near the base boundary $\eta(x) = 0$ is not guaranteed by linearity. The behaviour of general L near Γ has not been addressed here, and may indeed be important for the nonlinear, singular examples of operators presented in Chapter 1.

is now apparent. Sturm-Liouville theory states that the set $\{\psi_\alpha\}$ will be complete, so we are again performing a spectral stability analysis. This means that the $\mathcal{O}(\epsilon)$ -terms of (5.14) and (5.15) become, respectively,

$$\mathbf{u}_1(\mathbf{s}) + h(t)g(\mathbf{s})\psi_\alpha(\mathbf{s})\hat{n}(\mathbf{s}) \cdot \nabla \mathbf{u}_0(\mathbf{s}), \quad (5.20)$$

and

$$\nabla \mathbf{u}_1(\mathbf{s}) \cdot \hat{n} + h(t)\psi_\alpha(\mathbf{s})\Delta \mathbf{u}_0(\mathbf{s}). \quad (5.21)$$

We can now compute λ as in Chapter 2, obtaining

$$\lambda = \frac{\mathbf{w}^T \mathbf{G} \tilde{\boldsymbol{\mu}}_0}{\mathbf{w}^T \mathbf{M}^{-1} \mathbf{e}_1}, \quad (2.31)$$

where, according to (5.20) and (5.21), we now have

$$\tilde{\boldsymbol{\mu}}_0 = \begin{bmatrix} \Delta \mathbf{u}_0(\mathbf{s}) \\ \nabla \mathbf{u}_0(\mathbf{s}) \cdot \hat{n}(\mathbf{s}) \end{bmatrix}. \quad (5.22)$$

As long as each $\varphi_i(\alpha) > \mathcal{O}(1)$ in α , the derivative terms of \mathbf{K} give a similar contribution to the stiffness as in the ice-water example.

In Chapter 2, we applied our linear analysis to a variation on the Riabouchinsky flow problem addressed by Garabedian. We made a simplifying modification to the differential equation, solving the Laplacian rather than the complete flow equation

$$u_{xx} + u_{rr} - \frac{1}{r}u_r = 0. \quad (1.3a)$$

We are now in a position to consider the linearised free boundary dynamics of this more general linear equation. Let us assume, as in Chapter 2, that the base boundary is at $y = 1$, that we are solving in the upper half-plane, and that we can linearise the boundary conditions about this point.

Equation (1.3a) is separable. We set $u(x, r) = X(x)R(r)$, and choose α for the eigenvalue. The separated equations are

$$X'' + \alpha^2 X = 0, \quad (5.23a)$$

$$r^2 R'' - rR' - \alpha^2 r^2 R = 0. \quad (5.23b)$$

The first equation gives us the familiar orthonormal set of functions parallel to the boundary, $\{e^{i\alpha x}\}$. The second equation gives us the solution behaviour perpendicular to the boundary: solutions to this equation determine $\varphi(\alpha)$. $m = 1$ for this problem, so there is only one such Dirichlet-to-Neumann function.

The solutions to (5.23b) are $yI_1(|\alpha|y)$ and $yK_1(|\alpha|y)$ for $y > 0$, where $I_1(\xi)$ and $K_1(\xi)$ are modified Bessel functions of order 1. We use the second solution, which decays for $y \rightarrow \infty$. Using the series expansion for $K_1(\xi)$, cf. [4, ch 11], we compute the asymptotic expansion for

$$\varphi = \frac{[yK_1(|\alpha|y)]'}{yK_1(|\alpha|y)} \quad (5.24)$$

near $y = 1$, finding that $\varphi(\alpha) \sim -|\alpha|$, just as for the case when the operator L is the Laplacian. Therefore, $\mathbf{K} \sim [-|\alpha|, 1]^T$. This explains why our Chapter 2 analysis of the problem using the Laplacian, so $\mathbf{K} = [-|\alpha|, 1]^T$, so accurately explains the success of Garabedian's method. Results regarding the regularity lost in the Dirichlet-to-Neumann map [29], as well as this example, suggest that in general, when L is a second-order elliptic differential operator, $\varphi_i(\alpha) \sim |\alpha|$. However, the analysis needed to prove this conjecture is beyond the scope of this work.

To apply this analysis to the level set method of Osher and Sethian, instead of perturbing the boundary, $\mathbf{x}(\mathbf{s})$, we again perturb the level set function, seeking how

$$\phi = \phi_0 + \epsilon\phi_1 = 0 \quad (5.25)$$

affects the boundary perturbation, \mathbf{x}_1 , above. Again, we expand

$$\begin{aligned} 0 &= \phi(\mathbf{x}_0 + \epsilon\mathbf{x}_1 + \mathcal{O}(\epsilon^2)), \\ &= \phi(\mathbf{x}_0) + \epsilon\mathbf{x}_1 \cdot \nabla\phi(\mathbf{x}_0) + \mathcal{O}(\epsilon^2), \\ &= \phi_0(\mathbf{x}_0) + \epsilon[\phi_1(\mathbf{x}_0) + \mathbf{x}_1 \cdot \nabla\phi_0(\mathbf{x}_0)] + \mathcal{O}(\epsilon^2), \\ &= \phi_0(\mathbf{s}) + \epsilon[\phi_1(\mathbf{s}) + \mathbf{x}_1(\mathbf{s}) \cdot \nabla\phi_0(\mathbf{s})] + \mathcal{O}(\epsilon^2). \end{aligned} \quad (5.26)$$

where we have again suppressed the t -dependence of ϕ for clarity. ϕ_0 is a base solution, so $\phi_0(\mathbf{s}) = 0$, and, to first order,

$$\phi_1(\mathbf{s}) = -\mathbf{x}_1(\mathbf{s}) \cdot \nabla\phi_0(\mathbf{s}). \quad (5.27)$$

To isolate \mathbf{x}_1 , we note that we wish to find \mathbf{x}_1 normal to the base free boundary, so that $\mathbf{x}_1 = x_1\hat{n}$. Recalling that $\hat{n} = \nabla\phi_0/|\nabla\phi_0|$,

$$\phi_1(\mathbf{s}) = -x_1 \frac{\nabla\phi_0}{|\nabla\phi_0|} \cdot \nabla\phi_0 = -x_1 |\nabla\phi_0(\mathbf{s})|, \quad (5.28)$$

so as a result of our perturbation to ϕ_0 ,

$$\mathbf{x}_1 = -\frac{\phi_1(\mathbf{s})}{|\nabla\phi_0(\mathbf{s})|^2} \nabla\phi_0(\mathbf{s}). \quad (5.29)$$

Having identified how a perturbation in ϕ affects \mathbf{x}_1 , we use equations (5.14) and (5.15) to see that the $\mathcal{O}(\epsilon)$ value and normal derivative conditions at \mathbf{x}_0 become, respectively,

$$\mathbf{u}_1(\mathbf{s}) - \phi_1(\mathbf{s}) \frac{\nabla\phi_0 \cdot \nabla\mathbf{u}_0}{\nabla\phi_0 \cdot \nabla\phi_0}, \quad (5.30)$$

and

$$\nabla\mathbf{u}_1(\mathbf{s}) \cdot \hat{n} - \phi_1(\mathbf{s}) \frac{\Delta\mathbf{u}_0}{|\nabla\phi_0|}. \quad (5.31)$$

Computing the condition on the normal derivative is, admittedly, some work, and we omit the details. We again write $\phi_1(\mathbf{x}) = h(t)g(\mathbf{r})\psi_\alpha(\mathbf{x})$, where ψ_α is from a set of orthonormal functions, this time taken over the entire domain.

With \mathbf{r} defined to be the coordinates orthogonal to \mathbf{s} , $g(\mathbf{r})$ will again represent how the velocity varies away from Γ . We can again identify the problem stiffness as in equation (2.29), with the modification,

$$\tilde{\mu}_0 = \left[\begin{array}{c} \frac{\Delta \mathbf{u}_0}{-\frac{|\nabla \phi_0|}{\nabla \phi_0 \cdot \nabla \mathbf{u}_0}} \\ -\frac{\nabla \phi_0 \cdot \nabla \mathbf{u}_0}{\nabla \phi_0 \cdot \nabla \phi_0} \end{array} \right]. \quad (5.32)$$

Applying this to the development in the previous section, where $\phi_0 = y$ and $\phi_1 = h(t)g(0)e^{i\alpha x}$, we use equation (5.19) to compute \mathbf{x}_1 ,

$$\mathbf{x}_1 = -h(t)g(0)e^{i\alpha x}\hat{n}(\mathbf{s}). \quad (5.33)$$

In this case, $\hat{n}(\mathbf{s}) = \hat{n}(x, 0) = (0, 1)$, so this exactly represents the perturbation $y = \epsilon\eta_1(x) = -\epsilon h(t, y)e^{i\alpha x}$. $\nabla\phi_0(\mathbf{x}_0) = \nabla\phi_0(x, 0) = 1$, so since $\mathbf{u}_0 = \mathbf{u}_0(y)$, we have $\tilde{\mu} = -[\mathbf{u}_{0yy}, \mathbf{u}_{0y}]^T$. This gives the same λ as that in the linear analysis, with the change in sign computed in the previous section.

Just as with the argument in Chapter 2, to complete this discussion, we would have to verify the existence of base solutions, not only for \mathbf{u}_0 , but also for ϕ_0 , the corresponding level set function. The conclusion, however, is the same, that there exist generalised Stefan velocities that make the solve i) workable, and ii) $\mathcal{O}(1)$ in the mode number α . This reiterates the idea of the previous section, that regardless of the topology at hand, the level set method of Osher and Sethian can be used in combination with generalised Stefan velocities to form a numerical method for steady state free boundary problems with good stability properties.

5.3 Boundary contact points

In our exposition, we have chosen to make predictions and run simulations on periodic domains, such that the free boundary never contacts the outer boundary of the problem. This is by no means physically realistic, and near points of boundary contact, we must supply some additional information to the problem to determine the locations and angles of contact between the free boundary and the outer boundary of the domain. In his discussion of the dam seepage problem, Crank mentions the issue of boundary contact, but does not address it in general [8]. Determining contact angles for elliptic solves with our general class of boundary conditions does not appear to have been studied in the literature. We focus here on the well-posedness of free boundary problems near contact points, rather than on numerical schemes.

Consider the half-plane of Figure 5.1 divided into two wedge-shaped regions by a free boundary, a straight line through the origin. This is a linearised view near a point of contact, O , between the free boundary, Γ , and the physical boundary, $\partial D^+ \cup \partial D^-$, of the problem. Here, the point of contact of Γ is fixed, but the contact angle, ϕ , in the figure, is not. This suggests that our boundary

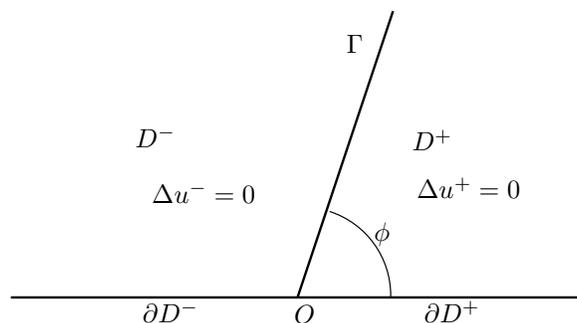


Figure 5.1: Free boundary near a point of contact with the physical boundary.

data explicitly fixes O , and we expect the behaviour of the solutions u^\pm near O to determine ϕ . We will use the polar coordinates (r, θ) near O , with the convention at all surfaces that $\hat{n} = \hat{\theta}$, so $d/dn = (1/r)d/d\theta$.

As an example, we use the free boundary conditions for the ice-water problem, for which $u^+ = u^- = 0$ and $\partial_n u^+ - a\partial_n u^- = 0$ on Γ at steady state. Suppose we have Dirichlet conditions along the physical boundary. We know that at O , $u = 0$, so we choose the conditions $u^+(r, 0) = pr$ on ∂D^+ and $u^-(r, \pi) = -qr$ on ∂D^- , p, q constants. In polar coordinates, the general solution for $\Delta u = 0$ is

$$u(r, \theta) = \sum_{n=0}^{\infty} r^n (A_n \cos n\theta + B_n \sin n\theta). \quad (5.34)$$

However, for our linearised treatment, we take only the $n = 1$ term (the $n = 0$ term is zero because $u = 0$ at O) so to solve the ice-water problem in these wedge-regions, we have

$$u^\pm = r(A^\pm \cos \theta + B^\pm \sin \theta), \quad (5.35)$$

and we seek the four coefficients A^\pm, B^\pm .

The conditions on ∂D^+ and ∂D^- , where $\theta = 0$ and $\theta = \pi$, respectively, give $A^+ = p$ and $A^- = q$. At Γ , we assume some $0 < \phi < \pi$, whereupon the conditions $u^+ = 0$ and $u^- = 0$ give $B^+ = -p \cot \phi$ and $B^- = -q \cot \phi$. With A^\pm, B^\pm determined, the final free boundary condition should determine the unknown contact angle, ϕ . $u_n^+ - au_n^- = 0$ gives

$$-A^+ \sin \phi + B^+ \cos \phi + aA^- \sin \phi - aB^- \cos \phi = 0, \quad (5.36)$$

which simplifies to

$$\frac{p - aq}{\sin \phi} = 0. \quad (5.37)$$

This is a surprising result, for it does not specify ϕ (except to say that $\sin \phi \neq 0$), but, rather, specifies the condition $p = aq$ on the boundary data along

$\partial D^+ \cup \partial D^-$, such that all three conditions along Γ , wherever it may be, can be satisfied. Strictly speaking, even with $p = aq$, the problem remains ill-posed, for the contact angle of the free boundary remains unknown.

As a solution to this difficulty, we suggest applying the mixed boundary data

$$\begin{aligned} u^+ - u_n^+ &= pr & \text{on } \partial D^+, \\ u^- + au_n^- &= -qr & \text{on } \partial D^-. \end{aligned} \quad (5.38)$$

These are radiation, or negative feedback, conditions, the same as those considered on Γ in Chapter 2. We use the same ansatz (5.35) for the solutions for u^\pm in terms of A^\pm, B^\pm , and applying the revised boundary conditions at $\partial D^+, \partial D^-$, we have

$$\begin{aligned} A^+r - B^+ &= pr, \\ -A^-r - aB^- &= -qr, \end{aligned} \quad (5.39)$$

while the conditions at Γ again give

$$\begin{aligned} r(A^+ \cos \phi + B^+ \sin \phi) &= 0, \\ r(A^- \cos \phi + B^- \sin \phi) &= 0, \\ -A^+ \sin \phi + B^+ \cos \phi + aA^- \sin \phi - aB^- \cos \phi &= 0. \end{aligned} \quad (5.40)$$

Comparing powers of r in (5.39) gives $A^+ = p, B^+ = 0, A^- = q, B^- = 0$ and with conditions (5.40), the problem seems overdetermined. However, we see that the third condition on Γ recovers the requirement $p = aq$, and the first two conditions agree that $\cos \phi = 0$, and thus determine the contact angle, $\phi = \pi/2$.

Choosing radiation conditions at the outer boundary seems physically reasonable, but also turns out to be necessary to give a well-posed problem. However, it is not clear that these are the only conditions on ∂D^\pm giving well-posed problems. A more general treatment of this topic with different conditions on Γ is yet to be done, and remains outside the scope of this work. Yet, this initial investigation indicates that boundary conditions near free boundary contact points cannot be chosen arbitrarily, and a system for setting such boundary conditions must be considered in order to develop algorithms for solving more general steady state free boundary problems.

5.4 Summary

This work considers the convergence of free boundary solves by the trial method. In a system of m dependent variables sharing a free boundary, the trial method assumes a fixed approximation to the boundary, solves the problem using m of the boundary conditions, and uses the residual in the remaining boundary condition to adjust the free boundary for the succeeding fixed-boundary solve. This procedure iterates until the residual is sufficiently small. For its motivation from a change-of-phase model, the residual condition, the choice of which is somewhat arbitrary, is called a generalised Stefan velocity. As solving the fixed-boundary problem is expensive, we explore a simple strategy for maximising the convergence rate while maintaining stability.

We consider the general class of free boundary problems, having second-order elliptic operators in each domain, with mixed conditions at the free boundary. As a model for this class of problems, we study systems having boundary conditions linear in Dirichlet and Neumann data at the free boundary, and satisfying $\Delta u = 0$ for each of m dependent variables sharing the free boundary.

A linear analysis of such systems reveals that given certain restrictions on global conditions of the problem, as well as local well-posedness, maximising convergence of the trial method without forsaking stability is possible. In choosing the residual condition, or velocity, using a boundary condition containing Dirichlet terms yields a numerical method that has a convergence rate, hence stability properties, independent of the discretisation used for the free boundary. This is our main result.

This result is confirmed in a computer experiment for a model of a water-ice change-of-phase system, which computes the boundary between the two phases. The test program uses a commercial finite element code to solve the fixed-boundary problem at each iteration. Hence, little original code is required to implement the generalised Stefan method.

We close this study by considering a mild extension to elliptic systems with more general operators. We also comment on a hybrid scheme that uses generalised velocities to capture the free boundary using a level set method. This future work can allow the computation of free boundaries with unknown topologies, while retaining the good convergence characteristics of our generalised Stefan velocity choices.

Bibliography

- [1] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1999.
- [2] Robert A. Adams. *Sobolev Spaces*. Academic Press, New York, 1975.
- [3] H. W. Alt and L. A. Caffarelli. Existence and regularity for a minimum problem with free boundary. *Journal für die Reine und angewandte Mathematik*, 325:105–144, 1981.
- [4] George B. Arfken and Hans J. Weber. *Mathematical Methods for Physicists*. Harcourt/Academic Press, San Diego, 5th edition, 2001.
- [5] William E. Boyce and Richard C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. Wiley & Sons, Toronto, 5th edition, 1992.
- [6] L. Bridge, R. Bredean, M. J. Ward, and B. R. Wetton. The analysis of a two-phase zone with condensation in a porous medium. *Journal of Engineering Mathematics*, 45:247–268, 2003.
- [7] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.
- [8] John Crank. *Free and Moving Boundary Problems*. Clarendon, Oxford, 1984.
- [9] C. Cuvelier and R. M. S. M. Schulkes. Some numerical methods for the computation of capillary free boundaries governed by the Navier-Stokes equations. *SIAM Review*, 32:355–423, 1990.
- [10] John R. Dormand. *Numerical Methods for Differential Equations*. CRC, New York, 1996.
- [11] M. Flucher and M. Rumpf. Bernoulli’s free-boundary problem, qualitative theory and numerical approximation. *Journal für die Reine und angewandte Mathematik*, 486:165–204, 1997.
- [12] Andrew Fowler. *Mathematical Models in the Applied Sciences*. Cambridge University Press, Cambridge, 1997.

-
- [13] I. A. Frigaard and C. Nouar. Nonlinear stability of poiseuille flow of a Bingham fluid: theoretical results and comparison with phenomenological criteria. *Journal of Non-Newtonian Fluid Mechanics*, 100:127–149, 2001.
- [14] P. R. Garabedian. The mathematical theory of three-dimensional cavities and jets. *Bulletin of the American Mathematical Society*, 62:219–235, 1956.
- [15] B. Gustafson, H.-O. Kreiss, and J. Olinger. *Time Dependent Problems and Difference Methods*. Wiley & Sons, New York, 1995.
- [16] Richard W. Hamming. *Numerical Methods for Scientists and Engineers*. Mc-Graw Hill, New York, 1973.
- [17] Comsol Inc. <http://www.comsol.com>. Accessed June 2003.
- [18] Kari T. Kärkkäinen and Timo Tiihonen. Free surfaces: shape sensitivity analysis and numerical methods. *International Journal for Numerical Methods in Engineering*, 44:1079–1098, 1999.
- [19] J. Ockendon, S. Howison, A. Lacey, and A. Movchan. *Applied Partial Differential Equations*. Oxford, New York, 1999.
- [20] J. T. Oden and J. N. Reddy. *An Introduction to the Mathematical Theory of Finite Elements*. Wiley & Sons, Toronto, 1976.
- [21] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 100:209–228, 1988.
- [22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge, New York, 2nd edition, 1992.
- [23] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge, New York, 2nd edition, 1999.
- [24] J. A. Sethian and J. Strain. Crystal growth and dendritic solidification. *Journal of Computational Physics*, 98:231–253, 1992.
- [25] N. Shamsundar. Comparison of numerical methods for diffusion problems with moving boundaries. In *Moving Boundary Problems (Gatlinburg, 1977)*, pages 165–185, New York, 1978. Academic Press.
- [26] J. Simon. Differentiation with respect to the domain in boundary value problems. *Numerical Functional Analysis and Optimization*, 2:649–687, 1980.
- [27] J. Sokolowski and J. P. Zolensio. *Introduction to Shape Optimization, Shape Sensitivity Analysis*. Springer-Verlag, New York, 1992.
- [28] Gilbert Strang. *Linear Algebra and Its Applications*. Harcourt, Brace, Jovanovich, San Diego, 3rd edition, 1988.

-
- [29] John Sylvester and Gunther Uhlmann. The Dirichlet to Neumann map and applications. In *Inverse Problems in Partial Differential Equations, (Artaca, CA, Jul 28-Aug 4, 1989)*, Philadelphia, 1990. SIAM.
- [30] Roger Temam. *Navier-Stokes Equations*. North-Holland, Amsterdam, 1977.
- [31] Timo Tiihonen. Shape optimization and trial methods for free boundary problems. *RAIRO, Mathematical Modelling and Numerical Analysis*, 31:805–825, 1997.
- [32] Timo Tiihonen. Fixed point methods for internal free boundary problems. *Numerical Functional Analysis and Optimization*, 19:399–413, 1998.
- [33] Timo Tiihonen and Jari Järvinen. On fixed point (trial) methods for free boundary problems. In *Free Boundary Problems in Continuum Mechanics (Novosibirsk, 1991)*, International Series of Numerical Mathematics, 106, pages 339–350, Basel, 1992. Birkhäuser.
- [34] K. S. Udell. Heat transfer in porous media heated from above with evaporation, condensation, and capillary effects. *Journal of Heat Transfer*, 105:485–492, August 1983.

Appendix A

Calculation Details

A.1 Ellipticity of the porous media flow problem

We consider the derivation of the problem of two-phase flow in a porous medium introduced in the examples in Chapter 1. By studying the physics of the problem, our aim is to provide enough detail to indicate that the problem is elliptic.

In the vapour only zone, we can describe the flow using the two variables pressure $P_v(x, t)$ and temperature $T(x, t)$. The molar density C will be determined by the ideal gas law

$$C = \frac{P_v}{RT} \quad (\text{A.1})$$

where R is the ideal gas constant. Conservation of heat and mass are given below:

$$\Delta T = 0 \quad (\text{A.2a})$$

$$\nabla \cdot (C\mathbf{u}) = 0, \quad (\text{A.2b})$$

where we assume here in (A.2a) the heat is transported primarily through conduction in the solid phase. The average vapour velocity \mathbf{u} is given by Darcy's law:

$$\mathbf{u} = -K_v(\nabla P_v + (0, \rho g)) \quad (\text{A.3})$$

where K_v is a constant containing vapour viscosity and the permeability of the porous medium, g is gravitational acceleration, and ρ is the mass density. $\rho = MC$, where $M = 18$ is the molar mass of water vapour. Combining (A.2b, A.3, A.1) we obtain

$$\nabla \cdot \left\{ \frac{K_v P_v}{RT} \left(\nabla P + \left(0, \frac{MP_v}{RT} \right) \right) \right\} = 0. \quad (\text{A.4})$$

Since the highest order term above is

$$\frac{K_v P_v}{RT} \Delta P_v \quad (\text{A.5})$$

we can recognise (A.4, A.2a) as a (nonlinear) elliptic system for P_v and T .

In the two-phase zone we make a saturation assumption, that is that the vapour pressure P_v is a given function $P_{sat}(T)$ that can be experimentally determined. Fitted P_{sat} approximations can be found in [6] for example. The

saturation pressure is an increasing function. In the two phase zone, the variables of interest are the temperature $T(x, y)$ and the saturation $s(x, y)$, the fraction of pore space occupied by liquid water. Gradients in vapour pressure drive a flow of vapour and gradients in the liquid pressure drive a flow of liquid:

$$u_l = -s^3 K_l (\nabla P_l + (0, \rho_l g)) \quad (\text{A.6})$$

$$u_v = -(1-s)^3 K_v (\nabla P_v + (0, \rho_v g)) \quad (\text{A.7})$$

where u_l is the liquid velocity, K_l is a constant, P_l is the liquid pressure, ρ_l is the liquid density (taken constant), u_v is the vapour velocity, $P_v = P_{sat}(T)$ is the vapour pressure, and ρ_v is the vapour density that can be determined from T and $P_{sat}(T)$ as above. The terms s^3 in (A.6) and $(1-s)^3$ in (A.7) are empirical relative permeability terms: it is easier to move liquid water and harder to move vapour when s is large.

The liquid pressure and the vapour pressure are related as follows:

$$P_l = P_v - P_c \quad (\text{A.8})$$

where $P_c(s)$ is the capillary pressure, taken to be a known (decreasing) function of s . We can write the conservation of liquid, vapour and thermal energy as follows:

$$\nabla(\rho_l u_l) = \gamma \quad (\text{A.9a})$$

$$\nabla(\rho_v u_v) = -\gamma \quad (\text{A.9b})$$

$$\Delta T = -K_T \gamma \quad (\text{A.10})$$

where γ is the local condensation rate and K_T is a constant, the ratio of the heat of condensation to the conductivity of the sand pack. Equation (A.10) can be seen to be elliptic in T using (A.9b) for γ by following back through (A.7) and recalling that $P_v = P_{sat}(T)$ is increasing with T . Equations (A.9a, A.9b) can be combined to eliminate γ giving

$$\nabla(\rho_l u_l + \rho_v u_v) = 0. \quad (\text{A.11})$$

Considering u_l from (A.6) using the capillary relationship for P_l (A.8) with $P'_c(s) < 0$ it is possible to interpret (A.11) as an elliptic equation for s , although one that is degenerate when $s = 0$.

Thus we have an elliptic system for T and P_v above the free surface Γ and an elliptic system for T and s below the free surface. We will show below that at the interface, five mixed Dirichlet-Neumann conditions are specified, thus that this problem also fits into our general framework. On the free boundary Γ we have the following conditions:

$$s = 0 \quad (\text{A.12a})$$

$$[T] = 0 \quad (\text{A.12b})$$

$$[P_v] = 0 \quad (\text{A.12c})$$

$$(\rho_v u_v)_+ \cdot \hat{n} = (\rho_v u_v + \rho_l u_l)_- \cdot \hat{n} \quad (\text{A.12d})$$

$$\left[\frac{\partial T}{\partial n} \right] = K_T (\rho_l u_l)_- \quad (\text{A.12e})$$

where subscripts $+$ denote limiting values from the vapour region and $-$ from the two-phase region, and $[\cdot]$ denotes the difference in quantities from one side of Γ to the other. Here, (A.12b) and (A.12c) denote continuity of temperature and pressure, respectively, (A.12d) is the conservation of total water, and (A.12e) is a thermal balance, allowing the presence of a condensation or evaporation front at Γ . Note that condition (A.12a) forces the system in the two-phase zone to be degenerate at the free surface.

At physical boundaries we specify two conditions: a given temperature and a no-flux mass condition. Finally, the total water content in the closed system is specified to make this a well-posed problem.

A.2 Calculation of λ

Recall from page 25 that we wish to isolate λ in the expression for the velocity,

$$\mathbf{v}^T \mathbf{G} \mathbf{K} \mathbf{c} + h \mathbf{v}^T \mathbf{G} \tilde{\mu}_0 = \frac{dh}{dt} = \lambda h. \quad (2.28)$$

Define \mathbf{A} to be the $m+1$ -by- m matrix with unit vector columns independent of \mathbf{v} . \mathbf{A} can be interpreted as the linear combinations of interface conditions used in the PDE solve at each iteration. Thus,

$$\mathbf{A}^T \mathbf{G} \mathbf{K} \mathbf{c} + h \mathbf{A}^T \mathbf{G} \tilde{\mu}_0 = 0. \quad (A.13)$$

From \mathbf{A} and \mathbf{v} construct the matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{v}^T \\ \mathbf{A}^T \end{bmatrix}. \quad (2.30)$$

With this construction, we can combine equations (2.28) and (A.13) to obtain

$$\mathbf{M} \mathbf{G} \mathbf{K} \mathbf{c} + h \mathbf{M} \mathbf{G} \tilde{\mu}_0 = [\lambda h, 0, 0]^T = \lambda h \mathbf{e}_1, \quad (A.14)$$

where $\mathbf{e}_1 = [1, 0, \dots, 0]^T$. \mathbf{M} is full-rank, so its inverse exists, and

$$\mathbf{G} \mathbf{K} \mathbf{c} + h \mathbf{G} \tilde{\mu}_0 = \lambda h \mathbf{M}^{-1} \mathbf{e}_1. \quad (A.15)$$

Finally, we multiply by the left nullspace of $\mathbf{G} \mathbf{K} = \mathbf{w}^T$, so

$$h \mathbf{w}^T \mathbf{G} \tilde{\mu}_0 = \lambda h \mathbf{w}^T \mathbf{v}. \quad (A.16)$$

$h \neq 0$, so we divide through by h and rearrange to give

$$\lambda = \frac{\mathbf{w}^T \mathbf{G} \tilde{\mu}_0}{\mathbf{w}^T \mathbf{M}^{-1} \mathbf{e}_1}. \quad (2.31)$$

Expression (2.31) can be simplified further if we can assume something about \mathbf{A} , the method used to solve the fixed-boundary PDE. If the columns of \mathbf{A} are

orthonormal to \mathbf{v} , we can row-reduce \mathbf{M} so that it is an orthonormal matrix, \mathbf{Q} . That is, $\mathbf{Q} = \mathbf{E}\mathbf{M}$, where \mathbf{E} is a row-echelon matrix. The first row of \mathbf{Q} is \mathbf{v}^T and row reduction only affects \mathbf{M} by making its rows \mathbf{A}^T orthonormal to each other. Therefore,

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \mathbf{X} & \\ 0 & & & \end{bmatrix}, \quad (\text{A.17})$$

where \mathbf{X} depends on how \mathbf{A} is row-reduced, so

$$\mathbf{M}^{-1}\mathbf{e}_1 = \mathbf{Q}^T\mathbf{E}\mathbf{e}_1 = \mathbf{v}. \quad (\text{A.18})$$

This gives

$$\lambda = \frac{\mathbf{w}^T\mathbf{G}\tilde{\mu}_0}{\mathbf{w}^T\mathbf{v}}, \quad (2.29)$$

the decay rate representing the stiffness of the problem presented in Chapter 2.

A.3 The Fourier integral of the cubic spline

Recall from page 33 that we wish to calculate

$$\hat{y}(n) = \frac{1}{2\pi} \int_0^{2\pi} y(x)e^{-inx} dx, \quad (3.7)$$

where we recall that

$$\hat{y}(n) = \frac{1}{2\pi} \frac{1}{(in)^3} \frac{6b}{(\Delta x)^3} (W_N^n - 1) \sum_{j=0}^{N-1} W_N^{nj} \int_{x_j}^{x_{j+1}} e^{-inx} dx, \quad (\text{A.19})$$

since $y'''(x)$ is constant on (x_j, x_{j+1}) . We compute the integral, as well as substituting $\Delta x = 2\pi/N$ and $x_j = j2\pi/N$:

$$\begin{aligned} \hat{y}(n) &= \frac{1}{2\pi} \left(\frac{1}{in}\right)^3 \frac{6b}{(\Delta x)^3} (W_N^n - 1) \sum_{j=0}^{N-1} W_N^{nj} \left[\frac{e^{-inx}}{-in} \right]_{x_j}^{x_{j+1}}, \\ &= -\frac{3b}{8\pi n} \left(\frac{N}{\pi n}\right)^3 (W_N^n - 1) \sum_{j=0}^{N-1} W_N^{nj} e^{-in\frac{2\pi}{N}j} (e^{-in\frac{2\pi}{N}} - 1), \\ &= -\frac{3b}{8\pi n} \left(\frac{N}{\pi n}\right)^3 (W_N^n - 1)(W_N^{-n} - 1) \sum_{j=0}^{N-1} W_N^{nj} W_N^{-nj}. \end{aligned} \quad (\text{A.20})$$

we recall that

$$b = \frac{\cos(2\pi n/N) - 1}{\cos(2\pi n/N) + 2}, \quad (3.6)$$

and compute

$$(W_N^n - 1)(W_N^{-n} - 1) = 2 \left(1 - \cos \left(\frac{2\pi n}{N} \right) \right). \quad (\text{A.21})$$

Since $W_N^{nj} W_N^{-nj} = 1$, the sum is N , and we can simplify the trigonometric expressions using $1 - \cos(2\theta) = 2 \sin^2(\theta)$ to obtain

$$\hat{y}(n) = \frac{3}{\cos(2\pi n/N) + 2} \frac{\sin^4(\pi n/N)}{(\pi n/N)^4}. \quad (\text{A.22})$$

This is the Fourier transform of a single discretised mode, W_N^{nj} . If, more generally, we have

$$y_j = \sum_{n=0}^{N-1} \tilde{y}(n) W_N^{nj}, \quad (\text{3.5})$$

where $[y_j]$ is arbitrary data, then the Fourier transform of the cubic spline fit to these points is the sum of the spline fits of each of the modes $\tilde{y}(n)$, so for general $[y_j]$,

$$\hat{y}(n) = \frac{3}{\cos(2\pi n/N) + 2} \frac{\sin^4(\pi n/N)}{(\pi n/N)^4} \tilde{y}(n), \quad (\text{3.13})$$

as stated in the main text.

A.4 The Fourier transform of periodic functions

In Chapter 3, page 35, we discuss smoothing by a 2π -periodic Gaussian window, and as such use the Fourier transform of this window. Let us compute the Fourier transform of more general functions, those of the form

$$f_{2\pi}(x) = \sum_{r=-\infty}^{\infty} f(x - 2\pi r). \quad (\text{A.23})$$

Our Gaussian window is, of course, a special case of this form.

If $f(x - 2\pi r) \in L^2[0, 2\pi]$ for all $r \in \mathbb{Z}$, and the sum (A.23) converges uniformly for $x \in [0, 2\pi]$, then

$$\begin{aligned} \hat{f}_{2\pi}(n) &= \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_{r=-\infty}^{\infty} f(x - 2\pi r) \right) e^{-inx} dx, \\ &= \frac{1}{2\pi} \sum_{r=-\infty}^{\infty} \int_0^{2\pi} f(x - 2\pi r) e^{-inx} dx. \end{aligned} \quad (\text{A.24})$$

Substitute $u = x - 2\pi r$, and recognise that $e^{in2\pi r} = 1$ for $r, n \in \mathbb{Z}$ so

$$\begin{aligned}\hat{f}_{2\pi}(n) &= \frac{1}{2\pi} \sum_{r=-\infty}^{\infty} \int_{2\pi(1-r)}^{-2\pi r} f(u) e^{-inu} e^{in2\pi r} du, \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(u) e^{-inu} du, \\ &= \hat{f}(n),\end{aligned}\tag{A.25}$$

the Fourier transform of $f(x)$ over all space. However, unlike the Fourier transform of $f(x)$, $x \in (-\infty, \infty)$, \hat{f} is to be evaluated only for integer arguments. The Fourier transform of our 2π -periodic Gaussian window follows directly.

A.5 Series solution to model ice-water problem

The model ice-water problem computed numerically in the examples can be solved analytically by scaling $u^-(x, y)$ by a , the ratio of thermal conductivities. We set $v(x, y) = u^+(x, y)$ in D^+ and $v(x, y) = au^-(x, y)$ in D^- . This guarantees the continuity of the flux $-\nabla v$ everywhere in the domain, and determining the location of Γ amounts to solving for the level set $v(x, y) = 0$.

With this scaling, we have the problem

$$\begin{cases} \Delta v = 0 & \text{in } D \\ v(x, 1) = u^+(x, 1) & \text{on } y = 1 \\ v(x, -1) = au^-(x, -1) & \text{on } y = -1 \end{cases},\tag{A.26}$$

where $D = D^+ \cup D^-$ is our domain $-\pi \leq x \leq \pi, -1 \leq y \leq 1$, 2π -periodic in x , and we recall that for our specific problem, $a = 10$, $u^-(x, -1) = -1$, and

$$u^+(x, 1) = \begin{cases} \frac{3}{2} & -\frac{\pi}{2} \leq x \leq \frac{\pi}{2} \\ \frac{1}{2} & \text{elsewhere} \end{cases}.\tag{4.1}$$

We can solve (A.26) by separation of variables, or by inspection, to obtain the two solutions $\exp(\pm ny) \exp(inx)$, $n \in \mathbb{Z}$ for the homogeneous problem, and $ay + b$, $a, b = \text{const}$ for the non-homogeneous problem. Thus, $v(x, y)$ is given by the series solution

$$v(x, y) = ay + b + \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} (a_n e^{ny} + b_n e^{-ny}) e^{inx}.\tag{A.27}$$

Solving the problem for $v(x, y)$, then, amounts to determining the coefficients $a, b, \{a_n\}, \{b_n\}$.

We solve the zeroth-order problem (that for a, b) using the boundary data $v_0 = -10$ at $y = -1$ and the minimum of v_1 at $y = 1$, $\check{v}_1 = 1/2$. Intuition alone suggests this will work *a priori*, but the $v(x, y)$ we calculate does turn out

to solve the problem. With this boundary data, the inhomogeneous solution is $a = 11/2, b = -9/2$.

Looking now at the homogeneous terms, we note that the factor $(a_n e^{ny} + b_n e^{-ny})$ in the sum is the Fourier transform of $v(x, y)$ with respect to x at location y . Thus, comparing this factor to the Fourier transform of the homogeneous data at $y = \pm 1$ will recover $\{a_n\}, \{b_n\}$. The homogeneous data at $y = -1$ is 0, as is its Fourier transform. The homogeneous data at $y = 1$ is 1 for $-\pi/2 < x \leq \pi/2$ and 0 elsewhere, so its Fourier transform is

$$\frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} e^{-inx} dx = \frac{1}{\pi n} \sin\left(\frac{n\pi}{2}\right). \quad (\text{A.28})$$

So, matching data at $y = \pm 1$,

$$a_n e^{-n} + b_n e^n = 0, \quad (\text{A.29a})$$

$$a_n e^n + b_n e^{-n} = \frac{1}{\pi n} \sin\left(\frac{n\pi}{2}\right). \quad (\text{A.29b})$$

Solving, we have

$$a_n = \frac{1}{2 \sinh(n)} \frac{1}{n\pi} \sin\left(\frac{n\pi}{2}\right), \quad (\text{A.30a})$$

$$b_n = -\frac{e^{-2n}}{2 \sinh(n)} \frac{1}{n\pi} \sin\left(\frac{n\pi}{2}\right), \quad (\text{A.30b})$$

and

$$a_n e^{ny} + b_n e^{-ny} = \frac{1}{n\pi} \sin\left(\frac{n\pi}{2}\right) \frac{\sinh[n(y+1)]}{\sinh(2n)}. \quad (\text{A.31})$$

Note that this factor is even in n , as should be expected since the boundary data is even. We can use this property to represent the solution as a sum of cosines,

$$v(x, y) = \frac{11}{2}y - \frac{9}{2} + \sum_{n=1}^{\infty} \frac{2}{n\pi} \sin\left(\frac{n\pi}{2}\right) \frac{\sinh[n(y+1)]}{\sinh(2n)} \cos(nx). \quad (\text{A.32})$$

Observe that this solution does indeed satisfy the boundary conditions, and our choice for the zeroth-order boundary conditions is reasonable. Using this solution, and the derivative $v_y(x, y)$, we can use Newton's method to numerically solve for the y -location of the boundary at each fixed x by finding the root of $v(x, y) = 0$. This solution is an infinite series, yet we can approximate it numerically using only a finite number of terms. Fortunately, the free boundary location is in $y < 1$, where the size of the summand decays as $\mathcal{O}(e^{-n}/n)$. When comparing this exact solution to that obtained by a boundary discretised by N points, N terms of this series are more than enough to approximate the exact solution such that the error of the numerical trial method solve far exceeds that of our truncated series approximation of the exact solution. Moreover, the Newton's method solve would generally be fraught with instability due to

poor initial root approximations. However, it is straightforward to implement here, as we know the approximate boundary shape from the numerical interface solves, from which Newton's method applied to our exact solution converges rapidly and stably.