Introduction to Automated Modeling using FEniCS

L. Ridgway Scott

University of Chicago

Release 0.1
DO NOT DISTRIBUTE

February 7, 2017

This page will contain copyright information in due course.

Contents

Chapter 1

Finite elements and FEniCS

1.1 The finite element method

Although the use of variational methods to solve PDEs can be traced earlier [60], the development of the finite element method started in earnest in the 1950's [56]. Finite element analysis is used by a large number of industries (Table 1.1) to solve many types of design problems (Table 1.2).

The gold standard software for five decades has been NASTRAN, a code developed in Fortran using software development techniques of the 1960's era. Based largely on this type of software alone, the industry has grown to produce multi-billion dollars of revenue each year. However, the simulation challenges of today and tomorrow, involving new materials with poorly understood fundamental properties, are stressing NASTRAN and its descendants to a breaking point. Moreover, as advanced technology becomes more highly refined and optimized, the need for simulation will continue to grow.

> *Soon the simulation software itself will become the majority of the value in product design. Thus the time is right to rethink technical simulation.*

In 2002, the FEniCS Project started at the University of Chicago devoted to changing the state of the art in simulation technology. It has many strengths that allow it to address a rapidly changing field of simulation challenges. FEniCS could become the basis for the next generation of simulation technology. The FEniCS Project originated in academic research, but now commercial support for FEniCS tools is available, and for the development of customer-driven applications based on the FEniCS structure.

We begin by reviewing the history of NASTRAN and FEniCS in some detail.

1.2 NASTRAN

NASTRAN (NASA STRucture ANalysis) is a finite element analysis (FEA) program that was originally developed for NASA in the late 1960s under NASA funding for the Aerospace industry. NASTRAN was released to the public in 1971 by NASA's Office of Technology Utilization.

Table 1.1: Industries using finite element software. Industries in black are taken from an MSC web page. Ones in red indicate industries that FEniCS can target as well.

Aerospace	Automotive	Consumer Products
Defense	Electronics	Energy
Heavy Equipment	Machinery	Materials
Medical	Motorsports	Packaging
Pharmaceutical	Rail	Shipbuilding
Sports equipment	Space systems	

Table 1.2: General applications for finite element software. Applications in black are taken from an MSC web page. Ones in red indicate industries that FEniCS can target as well.

Acoustics	Composites	Continuum Electrostatics
Crash & Safety	Density Funtional Theory	Design Optimization
Electromagnetics	Fatigue and Durability	Fluid flow
Fluid-structure interaction	Multibody Dynamics	Multidiscipline
Noise & Vibration	Nonlinear Analysis	Process Automation
Rheology	Rotor Dynamics	SPDM
Structural Analysis	Systems & Controls	Thermal Analysis

1.2.1 Software architecture

NASTRAN is written primarily in FORTRAN and contains over one million lines of code.

NASTRAN was designed using software techniques of the 1960's to consist of several modules. A module is a collection of FORTRAN subroutines designed to perform a specific task, such as processing model geometry, assembling matrices, applying constraints, solving matrix problems, calculating output quantities, conversing with the database, printing the solution, and so on.

1.2.2 Scope

NASTRAN is primarily a solver for finite element analysis. It does not have functionality that allows for graphically building a model or meshing. All input to and output from the program is in the form of text files.

1.3 FEniCS

The FEniCS Project started in an informal gathering in Hyde Park in 2002. It was loosely based on experiences gained in developing an earlier software system called Analysa, a commercial project funded largely by Ridgway Scott and Babak Bagheri [15], together with the experiences derived from a group based in Sweden led by Claes Johnson.

The FEniCS Project was publicly launched at the University of Chicago in 2003, and it has grown to an international collaboration involving primary developers at sites in England, Holland, Norway, and Sweden, as well as three sites in the US. It has many users world-wide.

It is estimated by `openhub.net` that the FEniCS Project represents 34 person-years of effort, with 25,547 software commits to a public repository, made by 87 contributors, and representing 134,932 lines of code. We estimate that there are about 50,000 downloads per year through a variety of sites. Google Analytics estimates that the FEniCS Project web page has about 10,000 monthly visitors.

The FEniCS project also recieves significant numbers of citations in the technical literature. This connotes more than just academic praise. It shows that the technical community feels that FEniCS tools are of significant value. For example, the FEniCS book [118] has about 150 citations per year and the DOLFIN paper [119] has about 50 citations per year. The citations to these two publications currently total 342 and 213, respectively, since they originally appeared.

FEniCS is a polymorphic acronym (e.g., Finite Elements nurtured in Computer Science, or For Everything new in Computational Science).

1.3.1 Leverage

The FEniCS Project leverages mathematical structure inherent in scientific models to automate the generation of simulation software. What makes FEniCS different from previous generations of software is that it uses compiler technology to generate software where possible, instead of just accumulating hand-coded software libraries. A major discovery was an optimized way to compute finite element matrices that makes finite element computation essentially as efficient as finite difference computation, while still retaining the geometric generality that is unique to finite element methods [101, 102, 103]. In 2012, a book was released [118] explaining both how FEniCS tools were developed as well as how they are used in a broad range of applications.

1.3.2 FEniCS advantages

FEniCS utilizes many modern software techniques. One of these is Just-in-Time (JIT) compilations, as did Analysa [15]. This allows a user to modify the physical models being used and quickly re-simulate. Similar tools allow for the order or type of finite element to be changed at will. FEniCS thus can go far beyond the standard concept of multi-physics. Instead of just mixing a few different physics models, FEniCS allows a continuum of models to be explored. This is one reason why such a diversity of phyiscal applications are presented in the book [118]. This is indicated to some extent by the items in red in Tables 1.2 and 1.1 where we see that FEniCS is able to target a broader range of applications and industries than were possible within the Nastran framework.

But there are many more examples of advantages. FEniCS is the first system to implement the full Periodic Table of Finite Elements [10]. This means that FEniCS users can utilize elements never before available for a variety of complicated models in finite element analysis. FEniCS also uses tensor representations [101, 102, 103, 104, 148] to make matrix generation more efficient. In addition, FEniCS interfaces to components that are state-of-the-art in different domains such as system solution and mesh generation. For example, FEniCS programs easily take advantage of PETSc (developed at Argonne National Laboratory), Trilios (developed at Sandia National Laboratory), and other linear and nonlinear system solvers. FEniCS also interfaces with CGAL, one of the leading mesh generation systems.

Chapter 2

Laplace-Poisson Equation

It is possible to define a function in a domain $\Omega \subset \mathbb{R}^d$ by specifying it on the boundary, $\partial\Omega$, together with its Laplacian (15.1) in the interior of the domain:

$$-\Delta u = f \text{ in } \Omega \tag{2.1}$$

together with boundary conditions

$$\begin{aligned} u &= 0 \text{ on } \Gamma \subset \partial\Omega \qquad \text{(Dirichlet)} \\ \frac{\partial u}{\partial n} &= 0 \text{ on } \partial\Omega\backslash\Gamma \qquad \text{(Neumann)} \end{aligned} \tag{2.2}$$

where $\frac{\partial u}{\partial n}$ denotes the derivative of u in the direction normal to the boundary, $\partial\Omega$.

To be precise, we assume that $\partial\Omega$ is Lipschitz continuous, we let $\mathbf{n}$ denote the outward unit normal vector to $\partial\Omega$, and we set $\frac{\partial u}{\partial n} = \mathbf{n} \cdot \nabla u$. The Laplace operator is defined by

$$\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}.$$

The equation (2.1) is known variously as Poisson's equation or Laplace's equation (especially when $f \equiv 0$). This equation forms the basis of a remarkable number of physical models. It serves as a basic equation for diffusion, elasticity, electrostatics, gravitation, and many more domains. In potential flow, the gradient of u is the velocity of incompressible, inviscid, irrotational fluid flow. The boundary $\partial\Omega$ is the surface of an obstacle moving in the flow, and one solves the equation on the exterior of Ω.

We will see (Section 15.1) that the right place to look for such the solution of such an equation is a Sobolev space denoted $H^1(\Omega)$ defined by

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) \ : \ \nabla v \in L^2(\Omega)^d \right\}, \tag{2.3}$$

where by $L^2(\Omega)$ we mean functions which are square integrable on Ω, and $L^2(\Omega)^d$ means d copies of $L^2(\Omega)$ (Cartesian product). There is a natural inner-product, and associated norm, on $L^2(\Omega)$ defined by

$$(v, w)_{L^2(\Omega)} = \int_\Omega v(\mathbf{x})\, w(\mathbf{x})\, d\mathbf{x}, \qquad \|v\|_{L^2(\Omega)} = \sqrt{(v, v)_{L^2(\Omega)}} = \Big(\int_\Omega v(\mathbf{x})^2\, d\mathbf{x} \Big)^{1/2}. \tag{2.4}$$

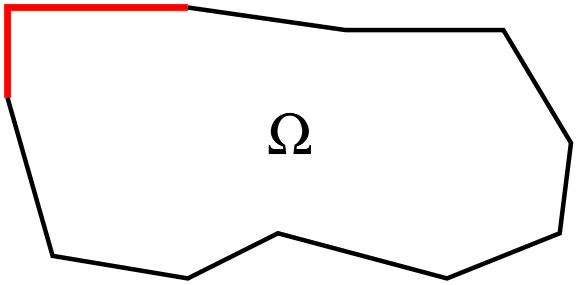

Figure 2.1: Domain Ω with Γ indicated in red.

Thus we can say that $v \in L^2(\Omega)$ if and only if $\|v\|_{L^2(\Omega)} < \infty$. For a vector-valued function $\mathbf{w}$, e.g., $\mathbf{w} = \nabla v$, we define

$$\|\mathbf{w}\|_{L^2(\Omega)} = \||\mathbf{w}|\|_{L^2(\Omega)} = \Big(\int_\Omega |\mathbf{w}(\mathbf{x})|^2 \, d\mathbf{x}\Big)^{1/2},$$

where $|\xi|$ denotes the Euclidean norm of the vector $\xi \in \mathbb{R}^d$.

2.1 Variational Formulation of Poisson's Equation

We now consider the equation (2.1) augmented with boundary conditions (2.2). To begin with, we assume that Γ has nonzero measure (that is, length or area, or even volume, depending on dimension). Later, we will return to the case when Γ is empty, the pure Neumann[1] case. A typical domain Ω is shown in Figure 2.1, with Γ shown in red.

To formulate the variational equivalent of (2.1) with boundary conditions (2.2), we define a variational space that incorporates the essential, i.e., Dirichlet, part of the boundary conditions in (2.2):

$$V := \big\{v \in H^1(\Omega) \; : \; v|_\Gamma = 0\big\}. \tag{2.5}$$

See Table 2.1 for an explanation of the various names used to describe different boundary conditions.

The appropriate bilinear form for the variational problem is determined by multiplying Poisson's equation by a suitably smooth function, integrating over Ω and then integrating by parts:

$$\begin{aligned}(f, v)_{L^2(\Omega)} &= \int_\Omega (-\Delta u) v \, d\mathbf{x} = \int_\Omega \nabla u \cdot \nabla v \, d\mathbf{x} - \oint_{\partial\Omega} v \, \frac{\partial u}{\partial n} \, ds \\ &= \int_\Omega \nabla u \cdot \nabla v \, d\mathbf{x} := a(u, v).\end{aligned} \tag{2.6}$$

[1] Carl Gottfried Neumann (1832—1925) was the son of Franz Ernst Neumann (1798–1895) who was a teacher of Kirchhoff. Carl (a.k.a. Karl) is also known for the Neumann series for matrices, and he was the thesis advisor of William Edward Story who at Clark University was the thesis advisor of Solomon Lefschetz.

generic name	example	honorific name
essential	$u = 0$	Dirichlet
natural	$\frac{\partial u}{\partial n} = 0$	Neumann

Table 2.1: Nomenclature for different types of boundary conditions.

The integration-by-parts formula derives from the divergence theorem

$$\int_\Omega \nabla\cdot \mathbf{w}(\mathbf{x})\, d\mathbf{x} = \oint_{\partial\Omega} \mathbf{w}(s)\cdot \mathbf{n}(s)\, ds \tag{2.7}$$

applied to $\mathbf{w} = v\nabla u$, together with the observations that $\frac{\partial u}{\partial n} = (\nabla u)\cdot \mathbf{n}$ and $\Delta u = \nabla\cdot (\nabla u)$. Here, $\mathbf{n}$ is the outward-directed normal to $\partial\Omega$. More precisely, we observe that

$$\nabla\cdot (v\nabla u) = \sum_{i=1}^d \big((v\nabla u)_i\big)_{,i} = \sum_{i=1}^d (v\, u_{,i})_{,i} = \sum_{i=1}^d v_{,i}u_{,i} + v\, u_{,ii} = \nabla v\cdot \nabla u + v\Delta u.$$

Thus the divergence theorem applied to $\mathbf{w} = v\nabla u$ gives

$$\oint_{\partial\Omega} v\nabla u(s)\cdot \mathbf{n}(s)\, ds = \int_\Omega \big(\nabla\cdot (v\nabla u)\big)(\mathbf{x})\, d\mathbf{x} = \int_\Omega (\nabla v\cdot \nabla u + v\Delta u)(\mathbf{x})\, d\mathbf{x},$$

which means that

$$\int_\Omega -v(\mathbf{x})\Delta u(\mathbf{x})\, d\mathbf{x} = \int_\Omega \nabla v(\mathbf{x})\cdot \nabla u(\mathbf{x})\, d\mathbf{x} - \oint_{\partial\Omega} v\,\frac{\partial u}{\partial n}\, ds. \tag{2.8}$$

The boundary term in (2.6) vanishes for $v \in V$ because either v or $\frac{\partial u}{\partial n}$ is zero on any part of the boundary. Thus, u can be characterized via

$$u \in V \ \text{ satisfying } \ a(u, v) = (f, v)_{L^2(\Omega)} \quad \forall v \in V. \tag{2.9}$$

The companion result that a solution to the variational problem in (2.9) solves Poisson's equation can also be proved [40], under suitable regularity conditions on u so that the relevant expressions in (2.1) and (2.2) are well defined. We will show how this is done in detail in the one-dimensional case in Section 6.2.3.

2.2 Formulation of the Pure Neumann Problem

In the previous section, we introduced the variational formulation for Poisson's equation with a combination of boundary conditions, and they all contained some essential (i.e., Dirichlet) component. The situation for the case of pure Neumann (or natural) boundary conditions

$$\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega \tag{2.10}$$

(i.e., when $\Gamma = \emptyset$) is a bit different, just as in the one-dimensional case (cf. Exercise 6.7). In particular, solutions are unique only up to an additive constant, and they can exist only if the right-hand side f in (2.1) satisfies

$$\int_\Omega f(\mathbf{x})\,d\mathbf{x} = \int_\Omega -\Delta u(\mathbf{x})\,d\mathbf{x} = \int_\Omega \nabla u(\mathbf{x}) \cdot \nabla 1\,d\mathbf{x} - \oint_{\partial\Omega} \frac{\partial u}{\partial n}\,ds = 0. \tag{2.11}$$

A variational space appropriate for the present case is

$$V = \left\{ v \in H^1(\Omega) \;:\; \int_\Omega v(\mathbf{x})\,d\mathbf{x} = 0 \right\}. \tag{2.12}$$

For any integrable function g, we define its *mean*, $\bar{g}$, as follows:

$$\bar{g} := \frac{1}{\mathrm{meas}(\Omega)} \int_\Omega g(\mathbf{x})\,d\mathbf{x}. \tag{2.13}$$

For any $v \in H^1(\Omega)$, note that $v - \bar{v} \in V$. Then $u - \bar{u}$ satisfies the variational formulation (2.9) with V defined as in (2.12). Conversely, if $u \in H^2(\Omega)$ solves the variational equation (2.9) with V defined as in (2.12), then u solves Poisson's equation (2.1) with a right-hand-side given by

$$\tilde{f}(\mathbf{x}) := f(\mathbf{x}) - \bar{f} \quad \forall \mathbf{x} \in \Omega \tag{2.14}$$

with boundary conditions (2.10).

2.3 Linear functionals as data

The expression $(f, v)_{L^2(\Omega)}$ on the right-hand side of (2.9) is an example of a **linear functional**. The right-hand side of (2.9) can be written succintly as

$$F(v) = (f, v)_{L^2(\Omega)} \qquad \forall v \in V. \tag{2.15}$$

The expression F is called a linear functional because (a) it is linear and (b) it has scalar values. By linear, we mean that $F(u + av) = F(u) + aF(v)$ for any scalar a and any $u, v \in V$.

The critical condition on a linear functional (a.k.a., **linear form**) for success in a variational formulation is that it be *bounded* or *continuous*. A **bounded linear functional** (equivalently a **continuous linear functional**) F on a normed space V must satisfy

$$|F(v)| \le C_F \|v\|_V \quad \forall v \in V. \tag{2.16}$$

A natural norm $\|\cdot\|_V$ for the space V defined in (6.6) is

$$\|v\|_a = \sqrt{a(v, v)}.$$

The smallest possible constant C_F for which this holds is called the **dual norm** of F and is defined by

$$\|F\|_{V'} := \sup_{0 \ne v \in V} \frac{|F(v)|}{\|v\|_V}. \tag{2.17}$$

We will see many bounded linear forms as right-hand sides in variational formulations. But there are many which are not bounded, such as

$$F(v) := v'(x_0) \tag{2.18}$$

for some $x_0 \in [0,1]$. This form is linear, but consider what it should do for the function $v \in H^1([0,1])$ given by

$$v(x) := |x - x_0|^{2/3} \tag{2.19}$$

(see Exercise 6.10).

2.4 Coercivity of the Variational Problem

The variational form $a(\cdot,\cdot)$ introduced in the previous two sections is **coercive** on the corresponding spaces V (see [40]): there is a constant C depending only on Ω and Γ such that

$$\|v\|^2_{H^1(\Omega)} \le c_0 a(v,v) \quad \forall v \in V. \tag{2.20}$$

We show in Section 6.4 how one can prove such a result in the one-dimensional case.

From (2.20), it follows that the problem (2.9) is well-posed. In particular, we easily see that the solution to the problem must be unique, for if f is identically zero then so is the solution. In the finite-dimensional case, this uniqueness also implies existence, and a similar result holds in the setting of infinite dimensional Hilbert spaces such as V. Moreover, the coercivity condition immediately implies a stability result, namely

$$\|u\|_{H^1(\Omega)} \le \frac{Ca(u,u)}{\|u\|_{H^1(\Omega)}} = C\frac{(f,u)_{L^2(\Omega)}}{\|u\|_{H^1(\Omega)}} \le C\|f\|_{V'}, \tag{2.21}$$

where $\|f\|_{V'}$ is defined in (2.17).

When coercivity holds, the **Lax-Milgram theorem** 2.1 guarantees that the variational problem (2.9) has a unique solution. There is an additional **continuity condition** that usually is straight-forward, namely that the form $a(\cdot,\cdot)$ is bounded on V, that is,

$$|a(u,v)| \le c_1 \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \quad \text{for all } u,v \in V. \tag{2.22}$$

In most cases, this condition is evident, but not in all as we describe in Section 5.2.1. What is often easy to see is that $a(v,v) \le C\|v\|^2_{H^1(\Omega)}$ for all $v \in V$. The connection between this condition and (2.22) is given by the Cauchy-Schwarz inequality (2.30). Thus we can consider the general variational formulation to find

$$u \in V \text{ satisfying } a(u,v) = F(v) \quad \forall v \in V. \tag{2.23}$$

Theorem 2.1 (Lax-Milgram) *Suppose that the variational form $a(\cdot,\cdot)$ is coercive (2.20) and continuous (2.22) (bounded) on $H^1(\Omega)$. Then the variational problem (2.23) has a unique solution u for every continuous (bounded) F defined on $H^1(\Omega)$. Moreover,*

$$\|u\|_{H^1(\Omega)} \le c_1 c_0 \sup_{v \in H^1(\Omega)} \frac{|F(v)|}{\|v\|_{H^1(\Omega)}}, \tag{2.24}$$

where c_0 is the constant in (2.20) and c_1 is the constant in (2.22).

The same type of bound as in (2.21) holds for discrete approximations as well under a very simple condition, indicated in (3.5). In this case, the bound corresponds to the **stability** of the numerical scheme.

2.5 Cauchy-Schwarz inequality

There is one small detail that we have let slip pass. The space V is defined using the requirement that $a(v,v) < \infty$, but what we need to know is that $a(u,v)$ is well defined for all $u, v \in V$. The latter is a consequence of the former, as follows.

Let $t \in \mathbb{R}$ be arbitrary, and expand $a(u-tv, u-tv)$ to get

$$a(u-tv,u-tv) = a(u-tv,u) - ta(u-tv,v) = a(u,u) - ta(v,u) - ta(u,v) + t^2 a(v,v). \quad (2.25)$$

The bilinear form $a(\cdot,\cdot)$ is symmetric: $a(v,w) = a(w,v)$, so (2.25) implies

$$a(u-tv,u-tv) = a(u,u) - 2ta(v,u) + t^2 a(v,v) = a(u,u) - 2ta(u,v) + t^2 a(v,v). \quad (2.26)$$

In particular, since $a(u-tv,u-tv) \geq 0$,

$$2ta(u,v) \leq a(u,u) + t^2 a(v,v). \quad (2.27)$$

For example, suppose that $a(v,v) = 0$. Choose the sign of t to be the sign of $a(u,v)$ and we conclude that

$$2|t|\,|a(u,v)| \leq a(u,u). \quad (2.28)$$

Since this holds for all $t \in \mathbb{R}$, we can let $|t| \to \infty$ to conclude that $a(u,v) = 0$. If $a(v,v) \neq 0$, define $t = \text{sign}(a(u,v))\|u\|_a/\|v\|_a$. If by chance $a(u,u) = 0$, then we reverse the previous argument to conclude that again $a(u,v) = 0$. If it is not zero, and thus $t \neq 0$, we can divide by $|t|$ in (2.27) to get

$$2|a(u,v)| \leq \frac{1}{|t|} a(u,u) + |t|\, a(v,v) = 2\|u\|_a \|v\|_a. \quad (2.29)$$

Thus we have proved the Cauchy-Schwarz inequality

$$|a(u,v)| \leq \|u\|_a \|v\|_a. \quad (2.30)$$

The Cauchy-Schwarz inequality is generally true for any non-negative, symmetric bilinear form. It is often stated as a property of an **inner-product**. Our bilinear form $a(\cdot,\cdot)$ is almost an inner-product except that it lacks one condition, non-degeneracy. In our case $a(v,v) = 0$ if v is constant, and for an inner-product, this is not allowed. One example of an inner-product is the bilinear form

$$(u,v)_{L^2(\Omega)} = \int_\Omega u(x)\, v(x)\, dx. \quad (2.31)$$

Here we see that $(v,v)_{L^2(\Omega)} = 0$ implies that $v \equiv 0$. But the Cauchy-Schwarz inequality does not require this additional property to be valid.

2.6 Method of manufactured solutions

We can test our technology by considering a problem with a known solution. One way to do this is to use the method of **manufactured solutions** [123]. Consider

$$\begin{aligned} -\Delta u &= 2\pi^2 \sin(\pi x)\sin(\pi y) \text{ in } \Omega = [0,1]\times[0,1] \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{2.32}$$

whose solution is $u(x,y) = \sin(\pi x)\sin(\pi y)$. Of course, we started with the solution $u(x,y) = \sin(\pi x)\sin(\pi y)$ and then computed its Laplacian to get $f = 2\pi^2 \sin(\pi x)\sin(\pi y)$. The implementation of this problem is given in Program 2.1.

```
 1 from dolfin import *
 2 
 3 # Create mesh and define function space
 4 mesh = UnitSquareMesh(32, 32)
 5 V = FunctionSpace(mesh, "Lagrange", 1)
 6 
 7 # Define boundary condition
 8 u0 = Constant(0.0)
 9 bc = DirichletBC(V, u0, "on_boundary")
10 
11 # Define variational problem
12 u = TrialFunction(V)
13 v = TestFunction(V)
14 you = Expression("(sin(3.141592*x[0]))*(sin(3.141592*x[1]))")
15 a = inner(grad(u), grad(v))*dx
16 L = (2*3.141592*3.141592)*you*v*dx
17 
18 # Compute solution
19 u = Function(V)
20 solve(a == L, u, bc)
21 plot(u, interactive=True)
```

Program 2.1: Code to implement the problem (2.32). The code in line 4 defines the domain (a square) and the boundary conditions are specified in lines 8 and 9. The code in lines 12 to 16 defines the ingredients of the variational formulation.

The code is a faithful representation of the variational formulation. The domain is represented by `mesh` which incapsulates both the definition of the domain and a subdivision of it. The form `a` in line 15 is just missing an integral sign and the domain of integration. The latter is implicit due to the link with the variational space `V` which encodes this information. The symbol `dx` stands for Lebesgue measure over the domain represented by `mesh`. It is necessary to specify this as other integrals will appear shortly.

technique	specification	example where used
keyword	`"on_boundary"`	Program 2.1
reserved function	`DomainBoundary()`	Program 3.1
defined function	`boundary`	Program 2.2, Program 4.1

Table 2.2: Different ways to specify the boundary of a domain.

In line 9 in Program 2.1, the third variable in `DirichletBC` specifies where Dirichlet boundary conditions are to be specified. Table 2.2 specifies two other ways of achieving the same thing. In Program 2.2, a logical function `boundary` is defined that is `True` if a point is outside the domain, and `False` otherwise. A small parameter `DOLFIN_EPS` (machine precision) is used to avoid floating point numbers being very close to zero. Thus for any points very close to the boundary of the square, the function `boundary` will evaluate to `True`.

```
def boundary(x):
    return x[0] < DOLFIN_EPS or         \
           x[0] > 1.0 - DOLFIN_EPS or  \
           x[1] < DOLFIN_EPS or         \
           x[1] > 1.0 - DOLFIN_EPS
```

Program 2.2: Code to define the boundary of a square explicitly.

2.7 Exercises

Exercise 2.1 *Modify Program 2.1 by using a different way to specify the boundary as described in Table 2.2. Make sure it gives a reasonable answer.*

Exercise 2.2 *Use the method of manufactured solutions to generate a polynomial solution on the square. Define $q(t) = t(1-t)$ and let $u(x,y) = q(x)q(y)$. Note that u satisfies Dirichlet boundary condtions on $\partial\Omega$. Find f such that $-\Delta u = f$. Modify Program 2.1 using this f and u.*

Exercise 2.3 *Use the method of manufactured solutions to generate a solution to the pure Neumann problem with boundary conditions (2.10) on the square. Start with $u(x,y) = (\cos\pi x)(\cos\pi y)$ and compute the corresponding f such that $-\Delta u = f$. Modify Program 2.1 using this f and u.*

Exercise 2.4 *Inhomogeneous Dirichlet boundary conditions $u = g$ on $\partial\Omega$ can be posed in two equivalent ways. Let us assume that $g \in H^1(\Omega)$ for simplicity. We can think of finding $u \in V + g$ such that*

$$a(u,v) = (f,v)_{L^2} \quad \forall v \in V.$$

Here $V = H^1_0(\Omega) = \{v \in H^1(\Omega) \,:\, v = 0 \text{ on } \partial\Omega\}$, and $V + g = \{v \in H^1(\Omega) \,:\, v - g \in V\}$. The other way that we can define u is by writing $u = u_0 + g$ with $u_0 \in V$ determined by

$$a(u_0, v) = (f, v)_{L^2} - a(g, v) \quad \forall v \in V.$$

Note that the linear functional $F(v) = (f, v)_{L^2} - a(g, v)$ is well defined and bounded for $v \in H^1(\Omega)$. But in either case, we need to justify the variational formulation. In the latter approach, u_0 is well defined since this is a standard variational formulation, but what we need to see is that it gives an unambiguous answer. For suppose that $u^i = u^i_0 + g^i$ for g^1 and g^2 that are the same on the boundary, that is, $g^1 - g^2 \in V$. Define

$$a(u^i_0, v) = (f, v)_{L^2} - a(g^i, v) \quad \forall v \in V.$$

Show that $u^1 = u^2$, that is, the solution depends only on the values of g on the boundary. (Hint: show that $u^1 - u^2 \in V$ and derive a variational expression for $u^1 - u^2$.)

Chapter 3

Variational approximation

3.1 Variational Approximation of Poisson's Equation

Let $\mathcal{T}_h$ denote a subdivision of Ω; typically this will be what is called a triangulation, made of triangles in two dimensions or tetrahedra in three dimensions. A triangulation of the domain in Figure 2.1 is shown in Figure 3.1(a). The main requirement for a triangulation is that no vertex of a triangle can be in the middle of an edge. However, more general subdivisions can be used that violate this property [18, 154, 163].

The main concept of the finite element method was to use each element of the subdivision as a separate domain in which to reason about the balance of forces or other concepts in the model. Mathematically, this corresponds to choosing a set of functions on each element to represent the variables used in the model. Often, the same set of functions is used on each element, although this is not necessary [154, 163, 59]. In this way, one constructs a finite dimensional space V_h which can be used in what is known as the **Galerkin method** to approximate the variational formulation (2.23), as follows:

$$\text{find } \; u_h \in V_h \; \text{ satisfying } \; a(u_h, v) = (f, v) \quad \forall v \in V_h. \tag{3.1}$$

Here we can think of h as designating the subdivision, or perhaps as a parameter that denotes the size of the elements of the subdivision.

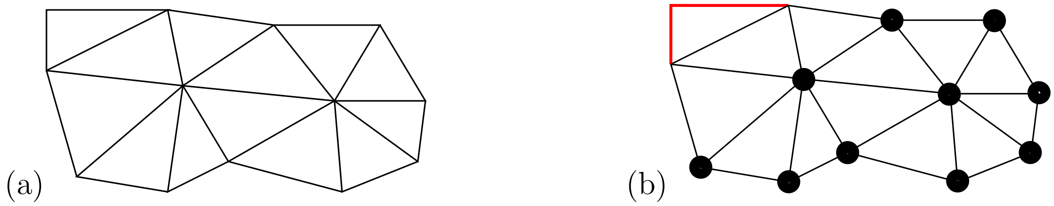

Figure 3.1: (a) Triangulation of the domain Ω. (b) Nodal positions for V_h are indicated by the black dots; note that vertices in Γ are not included, to respect the essential (Dirichlet) boundary condition.

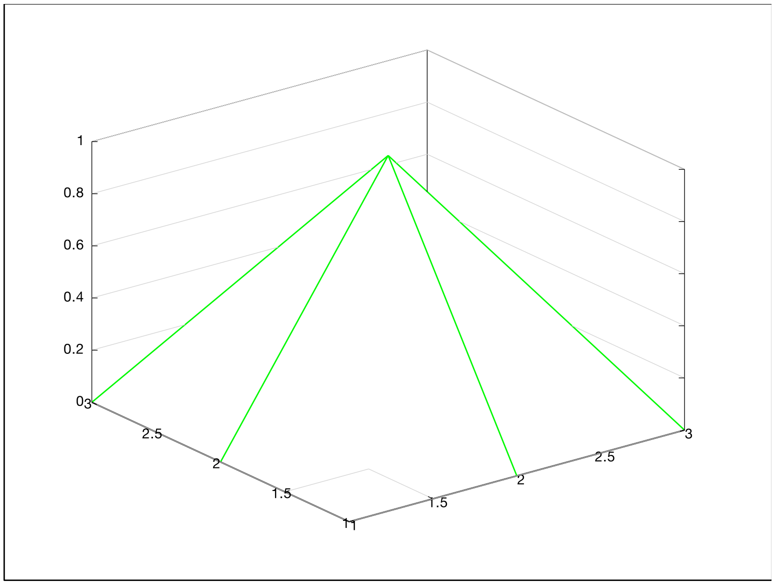

Figure 3.2: Typical basis function for continuous piecewise linear functions.

The coercivity condition implies stability for the discrete approximation, namely

$$\|u_h\|_{H^1(\Omega)} \le \frac{Ca(u_h,u_h)}{\|u_h\|_{H^1(\Omega)}} = C\frac{(f,u_h)}{\|u_h\|_{H^1(\Omega)}} \le C\|f\|_{V'}, \tag{3.2}$$

where we will explain the meaning of $\|f\|_{V'}$ in Section 6.2.1. In particular, if $f \equiv 0$, then $u_h \equiv 0$. Provided V_h is finite dimensional, this implies that (3.1) always has a unique solution. We can see this more clearly by choosing a basis $\{\phi_i \in V_h \; : \; i = 1, \ldots, N_h\}$. Write $u_h = \sum_i U_i\phi_i$. Using the linearity of the form $a(\cdot,\cdot)$ in each of its variables, we obtain the linear system $AU = F$ where

$$A_{ij} = a(\phi_i,\phi_j)\;\forall i,j = 1,\ldots,N_h, \qquad F_i = \int_\Omega f(\mathbf{x})\,\phi_i(\mathbf{x})\,d\mathbf{x}\;\forall i = 1,\ldots,N_h. \tag{3.3}$$

That is, since A is symmetric ($A_{ji} = a(\phi_j,\phi_i) = a(\phi_i,\phi_j) = A_{ij}$), we have

$$\begin{aligned} F_j &= \int_\Omega f(\mathbf{x})\,\phi_j(\mathbf{x})\,d\mathbf{x} = a(u_h,\phi_j) = a\big(\sum_i U_i\phi_i,\phi_j\big) \\ &= \sum_i U_i a(\phi_i,\phi_j) = \sum_i U_i A_{ij} = \sum_i A_{ji}U_i = \big(AU\big)_j \end{aligned} \tag{3.4}$$

for all $j = 1,\ldots,N_h$. We know from linear algebra that the solution to a linear system $AU = F$ exists uniquely if and only if the only solution for $F = \mathbf{0}$ is $U = \mathbf{0}$. The latter is guaranteed by the coercivity condition (2.21).

3.1.1 Piecewise linears

Given a triangulation, the simplest space V_h that we can construct is the set of continuous piecewise linear functions. This means that on each triangle (or tetrahedron), such functions

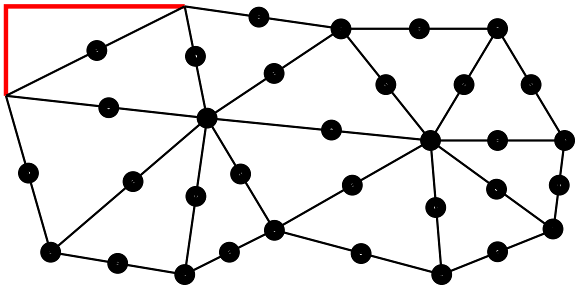

Figure 3.3: Nodes for quadratics: vertices and edge midpoints.

are linear, and moreover we contrive to make them continuous. A linear function is determined by its values at the vertices of a simplex. This is easy to see in one or two dimensions; the graph of the function is a line or plane going through the specified values at the vertices. If we demand that the values of $v \in V_h$ at vertices agree in all of the triangles meeting there, then it is not hard to see that the resulting function is continuous. In two dimensions, the values along edges are specified completely by the values at the vertices. Furthermore, we can define a basis for V_h in terms of functions that satisfy $\phi_i(\mathbf{x}_j) = \delta_{ij}$ (Kronecker δ). Such a function is depicted in Figure 3.2.

The vertices of a triangulation provide the **nodes** of the space V_h; these are shown as black dots in Figure 3.1. Note that we have indicated only the vertices where the nodal values are non-zero, respecting the boundary condition that $v = 0$ on Γ for $v \in V_h \subset V$.

In order to approximate the variational problem (2.23) with variational space (2.5), we need to insure that

$$V_h \subset V, \tag{3.5}$$

in order to apply Céa's Theorem [40, 2.8.1], which says the following.

Theorem 3.1 (Céa) *Suppose that $V_h \subset V$, that the variational form $a(\cdot,\cdot)$ is coercive (2.20) and continuous (2.22) (bounded) on $H^1(\Omega)$, and that F is well defined and bounded on $H^1(\Omega)$. Then*

$$\|u - u_h\|_{H^1(\Omega)} \leq c_1 c_0 \inf_{v \in V_h} \|u - v\|_{H^1(\Omega)}, \tag{3.6}$$

where c_0 is the constant in (2.20) and c_1 is the constant in (2.22).

3.1.2 Piecewise quadratic approximation

To obtain a more accurate solution in a cost effective way, it is often useful to use higher-order polynomials in each element in a subdivision. In Figure 3.3 we see the nodes for piecewise quadratic functions for the triangulation in Figure 3.1(a), respecting the essential boundary condition posed on Γ shown in red in Figure 2.1.

Again, we can define a basis for the space V_h of continuous piecewise quadratics in terms of functions that satisfy $\phi_i(\mathbf{x}_j) = \delta_{ij}$ (Kronecker δ), where the $\mathbf{x}_j$'s are the nodes in Figure 3.3. But now it is not so clear how we can be sure that this is a valid representation. What we

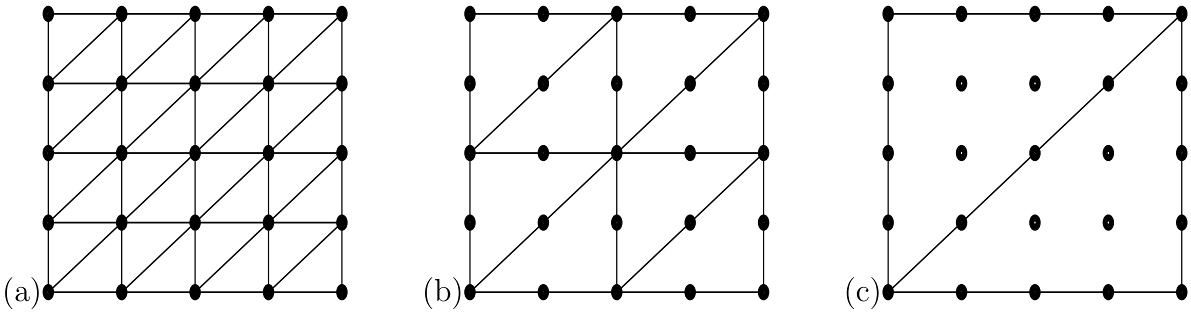

Figure 3.4: Varying mesh number M and polynomial degree k with the same number of nodes: (a) $M = 4$, $k = 1$ (linears), (b) $M = 2$, $k = 2$ (quadratics), (c) $M = 1$, $k = 4$ (quartics).

need to know is that this nodal representation is **unisolvent** on each triangle, meaning that on each triangle you can solve uniquely for a quadratic given the values at the specified nodes, the vertices and edge midpoints.

The way this is done is by degree reduction. On each edge, we have three distinct points that determine uniquely a quadratic, simply by invoking the fundamental theorem of algebra. In particular, we thus know that if all of the nodal values on one edge vanish, then the corresponding quadratic $q(x, y)$ must vanish on that edge. For simplicity, and without loss of generality, let us suppose that edge lies on the x-axis. Then $q(x, y) = y\ell(x, y)$ where ℓ is a linear polynomial in x and y. This can be verified by expanding q in powers of x and y (there are 6 terms) and invoking the fact that $q(x, y)$ vanishes on the edge lying on the x-axis. Now we use the fact that q also vanishes on the other two edges of the triangle, neither of which can lie on the x-axis, so that means that ℓ must also vanish on these edges. But this clearly implies that $\ell \equiv 0$, and thus $q \equiv 0$. By a simple result in linear algebra, we know that uniqueness of the representation implies existence of a representation, because we have made sure that we have exactly 6 nodal variables matching exactly the dimension (6) of the space of quadratic polynomials in two dimensions. Complete details are found in [40, Chapter 3].

3.1.3 Arbitrary degree polynomials

There is no limit on the degree of polynomials that can be used. The general family of elements is called the **Lagrange elements**. There is even some regularity to the pattern of nodes, as shown in Figure 3.4.

We can see the effect of varying the polynomial degree with a simple problem, using the method of manufactured solutions [123] (Section 2.6) using the problem (2.32), whose solution is $u(x, y) = \sin(\pi x)\sin(\pi y)$, which was first implemented in Program 2.1. A more sophisticated version is presented in Program 3.1. The main differences between Program 2.1 and Program 3.1 are found on lines 3–7 in Program 3.1. Line 3 imports the library `sys` to define the variables `sys.argv` and also imports an appropriate timer from the indicated library. Line 4 also imports an appropriate timer from the indicated library. The code in lines 6 and 7 are used to input data about the mesh size and polynomial degree from the command line. The importation of the library `math` in line 3 allows us to use $\pi =$ `math.pi` (lines 25 and 27) with full accuracy. In line 19, a construct is used to pass a parameter into the expression `f`. It is not allowed to include a variable, even a constant, inside the quotation marks in an `Expression` unless its value is defined in the `Expression` statement itself. The print command is standard Python, but it includes a `dolfin` command `errornorm` that we explain now.

One way to quatify the quality of a numerical solution is to compute the errors

$$\|u_h - u\|_{L^2(\Omega)} = \|u_h - (2\pi^2)^{-1} f\|_{L^2(\Omega)} \tag{3.7}$$

for different meshes (of the type shown in Figure 3.4) and polynomial degrees. Such errors, together with execution times for computing the numerical soltuion, are given in Table 3.1. The implementation of the error norm is given via the function `errornorm` in line 35 in Program 3.1. Recall that `f` is the exact solution and `u` is the finite element approximation. But `f` is an abstract expression, whereas `u` is in the finite element variational space (line 30). The function `errornorm` expects inputs of exactly this type as the first two entries. The third entry specifies which norm to compute. The final, optional entry deals with the issue of accuracy. Since the first entry is an exact expression, it can be evaluated to arbitrary accuracy, and the final entry specifies using a quadrature rule with accuracy three higher than would naturally be required just for computing the finite element (polynomial) degree accurately.

What we see in Table 3.1 is that the error can be reduced substantially by using higher-order polynomials. Increasing the mesh number for linear Lagrange elements does reduce the error, but the execution time grows commensurately with the error reduction. Using linears on a mesh of size 256 gives half the error of quadratics on a mesh of size 16, but the latter computation requires one-tenth of the time. For the same amount of time as this computation with quadratics, using quartics on a mesh of size 8 gives an error almost two orders of magnitude smaller.

Each mesh with double the number of mesh points was derived from the one with the smaller number of points by subdiving each triangle into four similar triangles. The cases with mesh number equal to 1, 2, and 4 are shown in Figure 3.5 in the case that crossed meshes are used.

To get the highest accuracy, the best strategy is to use higher polynomial order, up to a point. The most accurate computation occurs with polynomial degree 8 with a mesh number of 8. But the error quits decreasing at a certain point due to round-off error. We will discuss the effects of finite precision arithmetic is more detail in Section 6.7.3.

The times presented here should be viewed as approximate. There is significant variation due to system load from run to run. These computations were done on a MacBook Pro with 2.3 GHz Intel Core i7 and 16 GB 1600 MHz DDR3 memory. However, we do see order of magnitude variation depending on the mesh size and polynomial degree.

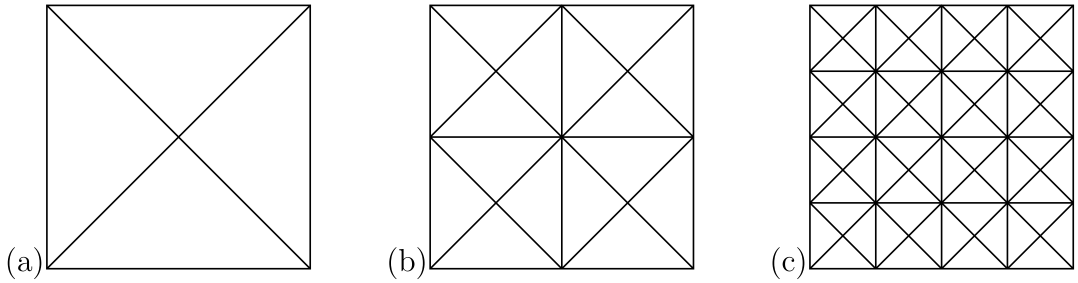

Figure 3.5: Crossed meshes with (a) mesh number 1, (b) mesh number 2, (c) mesh number 4.

degree	mesh number	L^2 error	time (s)
1	32	2.11e-03	0.09
2	8	5.65e-04	0.08
1	64	5.29e-04	0.13
1	128	1.32e-04	0.31
2	16	6.93e-05	0.08
1	256	3.31e-05	1.07
2	32	8.62e-06	0.11
2	64	1.08e-06	0.23
4	8	7.78e-07	0.08
8	2	7.29e-08	0.08
4	16	2.44e-08	0.11
16	1	1.61e-09	0.09
16	2	1.42e-09	0.12
4	32	7.64e-10	0.23
8	4	1.42e-10	0.09
4	64	2.39e-11	0.74
4	128	4.95e-12	3.0
8	8	3.98e-12	0.13
8	16	1.67e-11	0.33
8	32	6.78e-11	1.11
16	4	5.13e-09	0.25
16	8	2.14e-08	0.80

Table 3.1: Computational experiments with solving the problem (2.32). Degree refers to the polynomial degree, mesh number indicates the number of edges along each boundary side as indicated in Figure 3.5, L^2 error is the error measured in the $L^2([0,1]^2)$ norm, cf. (3.7), and time is in seconds. Meshes used are of the type shown in Figure 3.4. Results generated using Program 3.1.

3.1.4 Initial error estimates

It is easy to understand the basic error behavior for the finite element method. In the case of both piecewise linear and piecewise quadratics, we described the nodal basis functions ϕ_i which satisfy $\phi_i(x_j) = \delta_{ij}$ (Kronecker δ), where x_j denotes a typical node. For linears, the nodes are the vertices, and for quadratics the edge midpoints are added. For higher degree Lagrange elements, more edge nodes are involved, as well as interior nodes. For example, with cubics, the centroid of each triangle is a node.

Using such a nodal representation, we can define what is known as a **global interpolant** $\mathcal{I}_h$ defined on continuous functions, by

$$\mathcal{I}_h u = \sum_i u(\mathbf{x}_i)\phi_i. \tag{3.8}$$

Thus $\mathcal{I}_h$ maps continuous functions into the space V_h used in finite element computations.

Let $\mathcal{I}_h$ denote a global interpolant for a family of finite elements based on the components of $\mathcal{T}^h$. Let us suppose that $\mathcal{I}_h u$ is continuous, i.e., that the family of elements involved are C^0, as is true for the Lagrange family of elements. Further, suppose that the corresponding shape functions have an approximation order, m, that is

$$\|u - \mathcal{I}_h u\|_{H^1(\Omega)} \leq C h^{m-1} |u|_{H^m(\Omega)}. \tag{3.9}$$

In order to have good approximation, we need to have

$$\mathcal{I}_h \left(V \cap C^k(\Omega) \right) \subset V_h, \tag{3.10}$$

where k is the highest order of differentiation in the definition of $\mathcal{I}_h$, that is, $k = 0$ for Lagrange elements. However, we allow for the possibility that $k > 0$ since this holds for other element families.

If conditions (3.5) and (3.6) hold, then the unique solution, $u_h \in V_h$, to the variational problem

$$a(u_h, v) = (f, v) \quad \forall v \in V_h$$

satisfies

$$\|u - u_h\|_{H^1(\Omega)} \leq C \inf_{v \in V_h} \|u - v\|_{H^1(\Omega)}. \tag{3.11}$$

If conditions (3.9) and (3.10) hold, then

$$\|u - u_h\|_{H^1(\Omega)} \leq C h^{m-1} |u|_{H^m(\Omega)}.$$

The requirements (3.5) and (3.10) place a constraint on the subdivision in the case that Γ is neither empty nor all of the boundary. These requirements provide the **consistency** of the numerical approximation. In such a case, it is necessary to choose the mesh so that it aligns properly with the points where the boundary conditions change from Dirichlet to Neumann. For example, in two dimensions, if one uses Lagrange elements and insures that the points where the boundary conditions change are vertices in the triangulation, then defining

$$V_h := \mathcal{I}_h \left(V \cap C^0(\Omega) \right)$$

```
from dolfin import *
import sys,math
from timeit import default_timer as timer
startime=timer()
meshsize=int(sys.argv[1])
pdeg=int(sys.argv[2])
# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
V = FunctionSpace(mesh, "Lagrange", pdeg)
# Define boundary condition
u0 = Constant(0.0)
bc = DirichletBC(V, u0, DomainBoundary())
# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("(sin(mypi*x[0]))*(sin(mypi*x[1]))",mypi=math.pi)
a = inner(grad(u), grad(v))*dx
L = (2*math.pi*math.pi)*f*v*dx
# Compute solution
u = Function(V)
solve(a == L, u, bc)
aftersolveT=timer()
totime=aftersolveT-startime
print " ",pdeg," ",meshsize, \
      " %.2e"%errornorm(f,u,norm_type='l2', degree_rise=3)," %.3f"%totime
```

Program 3.1: Code to implement the problem (2.32) allowing variable mesh size and polynomial degree input from the command line.

is equivalent to defining V_h to be the space of piecewise polynomials that vanish on edges contained in Γ.

Since we have chosen the mesh so that the edges contained in Γ form a subdivision of the latter, it follows that (3.5) holds. On the other hand, if the set of edges where functions in V_h vanish is too small, we fail to obtain (3.5). If the set of edges where functions in V_h vanish is too big, (3.10) fails to hold. In the case of pure Dirichlet data, i.e., $\Gamma = \partial\Omega$, then V_h is just the set of piecewise polynomials that vanish on the entire boundary. In the case of pure Neumann data, i.e., $\Gamma = \emptyset$, V_h is the entire set of piecewise polynomials with no constraints at the boundary.

Even if we match the finite element space correctly with the set Γ where Dirichlet boundary conditions are imposed, there is an intrinsic singularity associated with changing boundary condition type along a straight boundary. This effect is explored in detail in Section 4.1.3.

3.1.5 Inhomogeneous Boundary Conditions

When boundary conditions are equal to zero, we often call them homogeneous, whereas we refer to nonzero boundary conditions as inhomogeneous. Inhomogeneous boundary conditions are easily treated. For example, suppose that we wish to solve (2.1) with boundary conditions

$$u = g_D \text{ on } \Gamma \subset \partial\Omega \qquad \text{and} \qquad \frac{\partial u}{\partial n} = g_N \text{ on } \partial\Omega\backslash\Gamma, \tag{3.12}$$

where g_D and g_N are given. For simplicity, let us assume that g_D is defined on all of Ω, with $g_D \in H^1(\Omega)$ and that $g_N \in L^2(\partial\Omega\backslash\Gamma)$. Define V to be the space (2.5). Then the variational formulation of (2.1) , (3.12) is as follows: find u such that $u - g_D \in V$ and such that

$$a(u,v) = (f,v)_{L^2(\Omega)} + \oint_{\partial\Omega\backslash\Gamma} g_N v\,ds \quad \forall v \in V. \tag{3.13}$$

This is well-posed since the linear form

$$F(v) := (f,v)_{L^2(\Omega)} + \oint_{\partial\Omega\backslash\Gamma} g_N v\,ds$$

is well defined (and continuous) for all $v \in V$. The equivalence of these formulations follows from (2.8): for any $v \in V$,

$$\int_\Omega (-\Delta u)v\,d\mathbf{x} = \int_\Omega \nabla u \cdot \nabla v\,d\mathbf{x} - \oint_{\partial\Omega} v\frac{\partial u}{\partial n}\,ds = a(u,v) - \oint_{\partial\Omega\backslash\Gamma} v\frac{\partial u}{\partial n}\,ds. \tag{3.14}$$

Thus, if u solves (2.1) with boundary conditions (3.12), then (3.13) follows as a consequence. Conversely, if u solves (3.13) then choosing v to vanish near $\partial\Omega$ shows that (2.1) holds, and thus

$$\oint_{\partial\Omega\backslash\Gamma} g_N v\,ds - \oint_{\partial\Omega\backslash\Gamma} v\frac{\partial u}{\partial n}\,ds = 0 \quad \forall v \in V.$$

Choosing v to be arbitrary proves (3.12) follows. Such arguments require some sophisticated tools in real analysis that are explained more fully in [40], and in the one-dimensional case, they are given in detail in Section 6.2.3.

The finite element approximation of (3.13) involves, typically, the use of an interpolant, $\mathcal{I}_h g_D$, of the Dirichlet data. We pick a subspace V_h of V just as before, and we seek u_h such that $u_h - \mathcal{I}_h g_D \in V_h$ and such that

$$a(u_h,v) = (f,v)_{L^2(\Omega)} + \oint_{\partial\Omega\backslash\Gamma} g_N v\,ds \quad \forall v \in V_h. \tag{3.15}$$

We can cast this in a more standard form as: find $\hat{u}_h = u_h - \mathcal{I}_h g_D \in V_h$ such that

$$a(\hat{u}_h,v) = (f,v)_{L^2(\Omega)} + \oint_{\partial\Omega\backslash\Gamma} g_N v\,ds + a(\mathcal{I}_h g_D, v) \quad \forall v \in V_h. \tag{3.16}$$

Then we can set $u_h = \hat{u}_h + \mathcal{I}_h g_D$. Fortunately, the `dolfin` built-in function `solve` automates all of this, so that the data g_D just needs to be specified.

3.2 Robin boundary conditions

It is frequently the case that more complex boundary conditions arise in physical models. The so-called Robin boundary conditions take the form

$$\alpha u + \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega\backslash\Gamma, \tag{3.17}$$

where α is a positive measurable function. (If α vanishes on some part of the boundary, then the boundary condition reduces to the standard Neumann condition there.) This will be coupled as before with a Dirichlet condition on Γ.

A variational formulation for this problem can be derived as follows. Let V be the space defined in (2.5) with the added proviso that $V = H^1(\Omega)$ in the case that $\Gamma = \emptyset$. From (2.8), we get

$$\begin{aligned}(f,v)_{L^2(\Omega)} &= \int_\Omega (-\Delta u(\mathbf{x}))v(\mathbf{x})\,d\mathbf{x} = \int_\Omega \nabla u(\mathbf{x})\cdot\nabla v(\mathbf{x})\,d\mathbf{x} - \oint_{\partial\Omega} v(s)\frac{\partial u}{\partial n}(s)\,ds \\ &= \int_\Omega \nabla u(\mathbf{x})\cdot\nabla v(\mathbf{x})\,d\mathbf{x} + \oint_{\partial\Omega} \alpha(s)\,v(s)\,u(s)\,ds,\end{aligned} \tag{3.18}$$

after substituting the boundary condition $\frac{\partial u}{\partial n} = -\alpha u$ on $\partial\Omega\backslash\Gamma$ and using the condition (2.5) that $v = 0$ on Γ. Thus we define a new variational form

$$a_{\text{Robin}}(u,v) := \int_\Omega \nabla u(\mathbf{x})\cdot\nabla v(\mathbf{x})\,d\mathbf{x} + \oint_{\partial\Omega} \alpha(s)\,v(s)\,u(s)\,ds. \tag{3.19}$$

The variational formulation for the equation (2.1) together with the Robin boundary condition (3.17) takes the usual form

$$u \in V \quad \text{satisfies} \quad a_{\text{Robin}}(u,v) = (f,v)_{L^2(\Omega)} \quad \forall v \in V. \tag{3.20}$$

The companion result that a solution to the variational problem in (3.20) solves both (2.1) and (3.17) can also be proved under suitable smoothness conditions.

Note that $a_{\text{Robin}}(\cdot,\cdot)$ is coercive on $H^1(\Omega)$, that is there is a constant $C < \infty$ such that

$$\|v\|^2_{H^1(\Omega)} \le C a_{\text{Robin}}(v,v) \quad \forall v \in H^1(\Omega), \tag{3.21}$$

provided that $\alpha > 0$. Thus the stability estimate (2.21) holds as well in this case.

A code implementing Robin boundary conditions (3.17) for the problem

$$-\Delta u = \sin(\pi x)\sin(\pi y) \text{ in } \Omega$$

is given in Program 3.2.

Exercise 3.1 *Repeat the experiments recorded in Table 3.1 but with the manufactured solution in Exercise 2.2. Explain why the error is so small for high-degree polynomial approximation even for a coarse mesh.*

```
 1 from dolfin import *
 2 
 3 # Create mesh and define function space
 4 mesh = UnitSquareMesh(32, 32)
 5 V = FunctionSpace(mesh, "Lagrange", 1)
 6 
 7 # Define variational problem
 8 u = TrialFunction(V)
 9 v = TestFunction(V)
10 f = Expression("(sin(3.141592*x[0]))*(sin(3.141592*x[1]))")
11 alfa = 1.0
12 a = inner(grad(u), grad(v))*dx + alfa*u*v*ds
13 L = (2*3.141592*3.141592)*f*v*dx
14 
15 # Compute solution
16 u = Function(V)
17 solve(a == L, u)
18 
19 # Plot solution
20 plot(u, interactive=True)
```

Program 3.2: Code to implement Robin boundary conditions. Note that the `solve` function in line 17 does not have a boundary condition function included in it.

Exercise 3.2 *Use Program 3.2 to explore the effect of the parameter α in Robin boundary conditions. Show that as $\alpha \to \infty$ that the solution tends to the solution of the Dirichlet problem. More precisely, compute the norm of the difference of the Robin solution from the known exact solution for the Dirichlet problem for large values of α. What happens when $\alpha \to 0$? Explain.*

Exercise 3.3 *Consider a regular mesh on $\Omega = [0,1] \times [0,1]$ which consists of $45°$ right triangles. Compute the "difference stencil" at the boundary points corresponding to using piecewise linear functions on this mesh in the variational approximation for the Robin boundary condition.*

Exercise 3.4 *Using an existing* `dolfin` *code for the standard boundary value problem for Laplace's equation, derive a code for Robin boundary conditions by implementing the form $a_{\text{Robin}}(\cdot,\cdot)$ using the standard form $a(\cdot,\cdot)$.*

Chapter 4

Singularities and the Laplace Equation

4.1 Geometry matters

The geometry of the domain boundary has a significant impact on the regularity of the solution. We begin by considering the problem

$$\begin{aligned} -\Delta u &= 0 \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega, \end{aligned} \tag{4.1}$$

where Ω is a polygonal domain in $\mathbb{R}^2$. We will see that the principal singularity of the solution can be identified, associated with what are often called re-entrant vertices.

4.1.1 L-shaped domain

The **L-shaped** domain Ω is depicted in Figure 4.1(a):

$$\Omega = \left\{(x,y) \in [-1,1]^2 \;:\; (x,y) = (r\cos\theta, r\sin\theta),\; 0 \le r \le 1,\; 0 < \theta < \tfrac{3}{2}\pi\right\}, \tag{4.2}$$

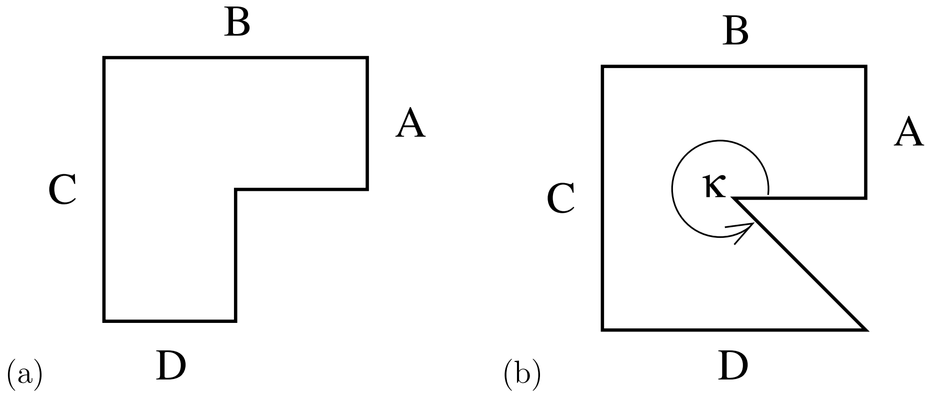

Figure 4.1: (a) L-shaped domain, (b) re-entrant corner of angle κ.

defined using polar coordinates $(x, y) = r(\cos\theta, \sin\theta)$. Again using polar coordinates, define

$$g(r(\cos\theta, \sin\theta)) = r^{2/3}\sin(\tfrac{2}{3}\theta). \tag{4.3}$$

We can think of $\partial\Omega$ consisting of two parts: the convex part $\Gamma_c = A \cup B \cup C \cup D$ where

$$\begin{aligned} A &= \{(1, y) \;:\; 0 \le y \le 1\}, & B &= \{(x, 1) \;:\; -1 \le x \le 1\}, \\ C &= \{(-1, y) \;:\; -1 \le y \le 1\}, & D &= \{(x, -1) \;:\; 0 \le x \le 1\}, \end{aligned} \tag{4.4}$$

(see Figure 4.1) and the re-entrant part

$$\Gamma_r = \{(0, y) \;:\; -1 \le y \le 0\} \cup \{(x, 0) \;:\; 0 \le x \le 1\}. \tag{4.5}$$

Then our data $g = 0$ on Γ_r. Moreover, it is not hard to see that g is **harmonic**, meaning $\Delta g = 0$, at least inside the open set Ω. This follows immediately from complex analysis, since g is the imaginary part of the complex analytic function $e^{(2/3)z}$. Deriving such a result is not easy using calculus, as we can indicate. First of all, using polar coordinates $(x, y) = r(\cos\theta, \sin\theta)$, we find the identities

$$\nabla r = \frac{(x, y)}{r} \qquad \text{and} \qquad \nabla\theta = \frac{(-y, x)}{r^2}.$$

This means that

$$\begin{aligned} \nabla g(x, y) &= \tfrac{2}{3}\big((\nabla r) r^{-1/3}\sin(\tfrac{2}{3}\theta) + (\nabla\theta) r^{2/3}\cos(\tfrac{2}{3}\theta)\big) \\ &= \tfrac{2}{3} r^{-4/3}\big((x, y)\sin(\tfrac{2}{3}\theta) + (-y, x)\cos(\tfrac{2}{3}\theta)\big) \\ &= \tfrac{2}{3} r^{-4/3}\big(x\sin(\tfrac{2}{3}\theta) - y\cos(\tfrac{2}{3}\theta), y\sin(\tfrac{2}{3}\theta) + x\cos(\tfrac{2}{3}\theta)\big) \\ &= \tfrac{2}{3} r^{-1/3}\big((\cos\theta)\sin(\tfrac{2}{3}\theta) - (\sin\theta)\cos(\tfrac{2}{3}\theta), (\sin\theta)\sin(\tfrac{2}{3}\theta) + (\cos\theta)\cos(\tfrac{2}{3}\theta)\big) \\ &= \tfrac{2}{3} r^{-1/3}\big(-\sin(\tfrac{1}{3}\theta), \cos(\tfrac{1}{3}\theta)\big), \end{aligned} \tag{4.6}$$

where we used the trigonometric identities that flow from the expressions ($\iota = \sqrt{-1}$)

$$\begin{aligned} \cos(\tfrac{1}{3}\theta) - \iota\sin(\tfrac{1}{3}\theta) &= \cos(-\tfrac{1}{3}\theta) + \iota\sin(-\tfrac{1}{3}\theta) = e^{-\iota(1/3)\theta} = e^{-\iota\theta}e^{\iota(2/3)\theta} \\ &= \big(\cos(-\theta) + \iota\sin(-\theta)\big)\big(\cos(\tfrac{2}{3}\theta) + \iota\sin(\tfrac{2}{3}\theta)\big) \\ &= \big(\cos\theta - \iota\sin\theta\big)\big(\cos(\tfrac{2}{3}\theta) + \iota\sin(\tfrac{2}{3}\theta)\big) \\ &= \big(\cos\theta\cos(\tfrac{2}{3}\theta) + \sin\theta\sin(\tfrac{2}{3}\theta)\big) + \iota\big(-\sin\theta\cos(\tfrac{2}{3}\theta) + \cos\theta\sin(\tfrac{2}{3}\theta)\big). \end{aligned} \tag{4.7}$$

The immediate result of the calculation (4.6) is that, for $0 < \theta < \frac{3}{2}\pi$, $|\nabla g(x, y)|$ blows up like $|(x, y)|^{-1/3}$, since

$$|\nabla g(x, y)| = |\nabla g(r\cos\theta, r\sin\theta)| = \tfrac{2}{3} r^{-1/3} = \tfrac{2}{3}|(x, y)|^{-1/3}.$$

Therefore $|\nabla g(x, y)|$ is square integrable, but it is obviously not bounded. Thus we see the benefit of working with Sobolev spaces, since this allows g to be considered a reasonable function even though it has an infinite gradient.

We can in principle use the vector calculus identity $\nabla\cdot(\phi\boldsymbol{\psi}) = \nabla\phi\cdot\boldsymbol{\psi} + \phi\nabla\cdot\boldsymbol{\psi}$ to compute $\Delta g = \nabla\cdot(\nabla g)$ to verify that $\Delta g = 0$, but the algebra is daunting. Instead, we can simply compute the solution via the standard variational problem (3.15) and see if we find $u = g$ throughout Ω. We leave this as Exercise 4.1. We also leave as Exercise 4.4 to verify that $\Delta g = 0$ by more classical analytical techniques.

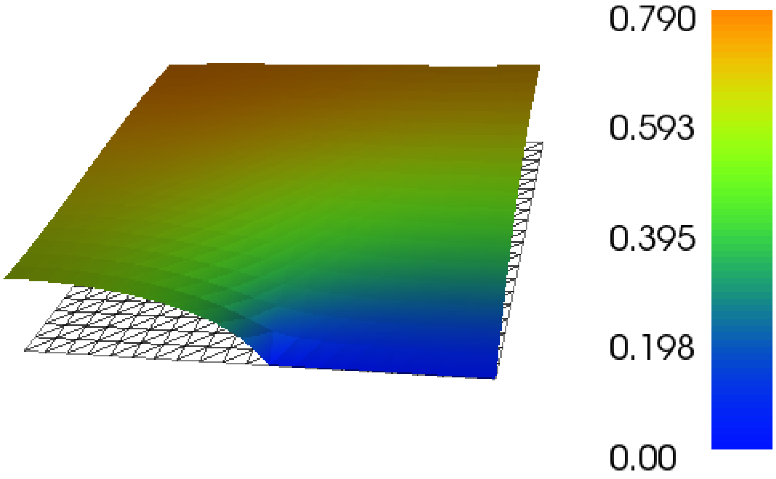

Figure 4.2: Illustration of the singularity that can occur when boundary condition types are changed, cf. (4.10), as well as a cut-away of the solution to the slit problem (4.9). Computed with piecewise linears on the indicated mesh.

4.1.2 General non-convex domains

The singularity for the L-shaped domain occurs for any domain with **non-convex vertex** as depicted in Figure 4.1(b), where the angle of the **re-entrant vertex** is κ. The L-shaped domain corresponds to $\kappa = \frac{3}{2}\pi$. The principle singularity for such a domain is of the form

$$g_\kappa(r(\cos\theta, \sin\theta)) = r^{\pi/\kappa}\sin((\pi/\kappa)\theta). \tag{4.8}$$

Note that when $\kappa < \pi$ (a convex vertex), the gradient of g_κ is bounded. We leave as Exercise 4.2 to explore this general case for various values of κ.

The largest that κ can be is 2π which corresponds to a **slit domain**. In this case, we have $g_{2\pi} = \sqrt{r}\sin(\frac{1}{2}\theta)$, which is still in $H^1(\Omega)$. The slit domain is often a model for **crack propagation**. An illustration of a problem on a slit domain is given by

$$\begin{gathered} -\Delta u = 1 \text{ in } [0,1]\times[-1,1] \\ u = 0 \text{ on } \Gamma, \quad \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega\backslash\Gamma, \end{gathered} \tag{4.9}$$

where $\Gamma = \big\{(x,0) \,:\, x \in [\frac{1}{2},1]\big\}$. The solution of (4.9) is depicted in Figure 4.2, where only the top half of the domain (that is, $[0,1]\times[0,1]$) is shown. The solution in the bottom half of the domain can be obtained by symmetric reflection across the x-axis.

The range of κ values for a realistic polygonal domain excludes a region around $\kappa = 0$ and $\kappa = \pi$. In particular, we see that $\kappa = \pi$ does not yield a singularity; the boundary is a straight line in this case, and $g_\pi(x,y) = r\sin\theta = y$, which is not singular. When $\kappa = 0$, there is no interior in the domain near this point. Thus for any polygonal domain with a finite number of vertices with angles κ_j, there is some $\epsilon > 0$ such that $\kappa_j \in [\epsilon, \pi - \epsilon] \cup [\pi + \epsilon, 2\pi]$ for all j.

In three dimensions, the set of possible singularities is much greater [58]. Edge singularities correspond to the vertex singularities in two dimensions, but in addition, vertex singularities appear [174]. The effect of smoothing singular boundaries is considered in [74].

4.1.3 Changing boundary condition type

The slit domain problem also allows us to assess the singularity that potentially occurs when boundary conditions change type along a straight line. Suppose that we have a domain $\Omega = \{(x,y) \in \mathbb{R}^2 \,:\, x \in [-1,1],\ y \in [0,1]\}$ and we impose homogeneous Dirichlet conditions on $\Gamma = \{(x,0) \in \mathbb{R}^2 \,:\, x \in [0,1]\}$ and Neumann conditions on $\Gamma^* = \partial\Omega\backslash\Gamma$. We can reflect the domain Ω around the line $y = 0$, and we get the domain $[-1,1]^2$ with a slit given by Γ. Therefore we see that $g_{2\pi} = \sqrt{r}\sin(\frac{1}{2}\theta)$ is harmonic in Ω, satisfies Dirichlet conditions on Γ and Neumann conditions on Γ^*.

We can expect such a singularity any time we switch from Dirichlet to Neumann boundary conditions along a straight boundary segment, even with homogeneous boundary condtions. We illustrate this with the following problem:

$$\begin{aligned} -\Delta u &= 0 \text{ in } [0,1]^2 \\ u = 0 \text{ on } \Gamma, \quad \frac{\partial u}{\partial n} &= 0 \text{ on } \partial\Omega\backslash\Gamma, \end{aligned} \tag{4.10}$$

where $\Gamma = \big\{(x,0) \,:\, x \in [\frac{1}{2}, 1]\big\}$, whose solution is depicted in Figure 4.2. The code for solving this problem is given in Program 4.1 We leave as Exercise 4.3 to explore this problem in more detail.

4.1.4 Optimal mesh refinements

When the form of a singularity is known, it is possible to predict what an optimal mesh refinement should be. We have seen that the gradient of the solution of the L-shaped problem blows up like $|(x,y)|^{-1/3}$ near the re-entrant corner. So suppose that, in general,

$$|\nabla^k u(\mathbf{r})| \approx C|\mathbf{r} - \mathbf{r}_0|^{-k+\gamma} \text{ for } \mathbf{r} \in \Omega, \tag{4.11}$$

where $\nabla^k u$ denotes the tensor of partial derivatives of order k of u, and $|\nabla^k u|$ denotes the Frobenius norm of that tensor, that is, the Euclidean norm of the tensor represented as a vector (the square root of the sum of squares of all entries). For the solution of the L-shaped problem, we have seen that this holds for $k = 1$ and $\gamma = 2/3$. It is possible to show that (4.11) holds for all $k \geq 1$ and $\gamma = \pi/\kappa$ for boundary vertices with angle κ. For simplicity, we assume that $\mathbf{r}_0 = \mathbf{0}$ from now on.

From (3.11) we have $\|u - u_h\|_{H^1(\Omega)} \leq C \inf_{v \in V_h} \|u - v\|_{H^1(\Omega)}$. For a non-uniform mesh, we need to use a more precise characterization of the interpolant (3.8) error:

$$\|u - \mathcal{I}_h u\|^2_{H^1(\Omega)} \leq C \sum_e \left(h_e^{m-1} \|u\|_{H^m(e)} \right)^2, \tag{4.12}$$

where the summation is over all of the elements e of the mesh subdivsion and h_e is the size of e.

Since we are assuming that the derivatives of the solution degrade in a radial fashion, let us also assume that the mesh is refined in a radial fashion. For each element e, let $\mathbf{r}_e$ denote its centroid $\mathbf{r}_e$. We assume that there is a monotonic mesh function μ such that $h_e \approx (1/n)\mu(|\mathbf{r}_e|)$, where n is a parameter that we can use to refine the mesh. For example, we will consider $\mu(r) = r^\beta$ for $\beta > 0$. Let $|e|$ denote the volume of an element e. With such a mesh and under the assumption (4.11), the error expression (4.12) takes the form

$$n^{2-2m} \sum_e \left(\mu(|\mathbf{r}_e|)^{m-1} |\mathbf{r}_e|^{-m+\gamma} \sqrt{|e|} \right)^2 \approx n^{2-2m} \int_\Omega \left(\mu(|\mathbf{r}|)^{m-1} |\mathbf{r}|^{-m+\gamma} \right)^2 d\mathbf{r}. \tag{4.13}$$

Taking $\mu(r) = r^\beta$, the integrand in (4.13) simplifies to $|\mathbf{r}|^p$ where $p = 2(\beta(m-1) - m + \gamma)$. Such an expression is integrable in d dimensions if and only if $p > -d$, that is, if

$$\beta > \frac{m - \gamma - d/2}{m - 1}.$$

For example, if $d = 2$ and $m = 2$ (piecewise linears in two dimensions), then the requirement is $\beta > 1 - \gamma$. For the L-shaped domain, this means $\beta > \frac{1}{3}$. However, for higher-order approximations, the appropriate mesh conditions will be different. In addition, the other corners of the L-shaped domain can also require mesh refinement. For these, $\gamma = 2$, and so using cubics ($m = 4$) also requires $\beta > \frac{1}{3}$ at these convex right angles. In this case, $\beta > 7/9$ is required at the re-entrant corner (for $m = 4$).

Recall that $\frac{1}{2} \leq \gamma < \infty$ in general ($\gamma = \pi/\kappa$). Thus when γ is sufficiently large (comparable with $m - d/2$), we can take $\beta \approx 0$, meaning a mesh of uniform size.

The mesh parameter n can be related to the number of degrees of freedom N, at least asymptotically, as follows. We can write

$$N \approx c_1 \sum_e |e|^0 \approx c_2 \sum_e h_e^{-d} |e|^1 \approx c_3 n^d \int_\Omega |r|^{-\beta d} \, d\mathbf{x} \approx c_4 n^d,$$

provided that $\beta < 1$, as we now assume.

4.2 An ill-posed problem?

It is tempting to idealize localized behavior in a physical system as occuring at a single point. For example, one might wonder what the shape of a drum head would be if one pushes down on it with a sharp pin. The Laplace equation models to a reasonable extent the deformation of the drum head (for small deformations), so one might consider

$$\begin{aligned} -\Delta u =& 0 \text{ in } \Omega \\ u(\mathbf{x}_0) =& u_0 \end{aligned} \tag{4.14}$$

where u_0 denotes the prescribed position of a knife edge. However, this problem is not well-posed. The difficulty is that one cannot constrain a function in $H^1(\Omega)$ at a single point. This is illustrated by the function

$$v(\mathbf{x}) = \log|\log|\mathbf{x}|\,| \tag{4.15}$$

which satsifies $v \in H^1(B)$ where $B = \left\{\mathbf{x} \in \mathbb{R}^2 \;:\; |\mathbf{x}| < \frac{1}{2}\right\}$ [40, Example 1.4.3]. This function does not have a well-defined point value at the origin. By shifting this function around, we realize that functions in H^1 may not have point values on a dense set of points. Thus setting a point value for a function in H^1 does not make sense.

It is possible to change to a Dirichlet problem

$$\begin{aligned} -\Delta u =& 0 \text{ in } \Omega \\ u =& u_0 \text{ on } \Gamma \end{aligned} \tag{4.16}$$

where Γ is a small curve representing the point of contact of the pencil with the drum head, and u_0 is some function defined on Γ. As long as Γ has positive length, this problem is well-posed. However, its behavior will degenerate as the length of Γ is decreased.

Another approach to modeling such phenomena is using the Dirac δ-function [40]:

$$\begin{aligned} -\Delta u =& \delta_{\mathbf{x}_0} \text{ in } \Omega \\ u =& 0 \text{ on } \partial\Omega, \end{aligned} \tag{4.17}$$

where $\delta_{\mathbf{x}_0}$ is the linear functional $\delta_{\mathbf{x}_0}(v) = v(\mathbf{x}_0)$. Again, there is an issue since this linear functional is not bounded on V, as the function v defined in (4.15) illustrates. On the other hand, the solution to (4.17) is known as the Green's function for the Laplacian on Ω (with Dirichlet conditions). It is possible to make sense of (4.17) using more sophisticated Sobolev spaces [40]. However, rather than taking that approach, we take one that effectively resolves the issue in conventional spaces. What we do is replace $\delta_{\mathbf{x}_0}$ by a smooth function $\delta^A_{\mathbf{x}_0}$ with the property that

$$\int_\Omega \delta^A_{\mathbf{x}_0}(\mathbf{x})\, v(\mathbf{x})\, d\mathbf{x} \to v(\mathbf{x}_0) \text{ as } A \to \infty \tag{4.18}$$

for sufficiently smooth v. We then consider the problem

$$\begin{aligned} \Delta u^A =& \delta^A_{\mathbf{x}_0} \text{ in } \Omega \\ u^A =& g \text{ on } \partial\Omega. \end{aligned} \tag{4.19}$$

Note that we can pick g to be the fundamental solution, and thus we have $u^A \to g$ as $A \to \infty$. For example, we can choose $\delta^A_{\mathbf{x}_0}$ to be Gaussian function of amplitude A and integral 1. In particular, in two dimensions,

$$\delta^A_{\mathbf{x}_0} = A\, e^{-\pi A|\mathbf{x}-\mathbf{x}_0|^2}. \tag{4.20}$$

degree	mesh number	amplitude	error	check-sum
1	128	10,000	5.27e-03	-2.34e-02
1	256	10,000	2.50e-03	-4.57e-09
1	512	10,000	1.47e-03	-2.22e-16
1	1024	10,000	1.08e-03	5.11e-15
4	256	10,000	9.73e-04	-1.02e-10
1	512	100,000	9.67e-04	-1.06e-03
1	1024	100,000	5.24e-04	-1.98e-14

Table 4.1: Data for the solution of (4.19). The amplitude is A, error is $\|u_h^A - g\|_{L^2(\Omega)}$, check-sum is the value $1-\int_\Omega \left(\delta^A_{\mathbf{x}_0}\right)_h d\mathbf{x}$ where $\left(\delta^A_{\mathbf{x}_0}\right)_h$ denotes the interpolant of $\delta^A_{\mathbf{x}_0}$ in V_h.

We check our requirement that the integral is 1 via the change of variables $\mathbf{y} = \sqrt{\pi A}\,\mathbf{x}$:

$$\int_{\mathbb{R}^2} \pi A\, e^{-\pi A|\mathbf{x}-\mathbf{x}_0|^2}\, d\mathbf{x} = \int_{\mathbb{R}^2} e^{-|\mathbf{y}-\mathbf{y}_0|^2}\, d\mathbf{y} = 2\pi \int_0^\infty e^{-r^2} r\, dr = \pi \int_0^\infty e^s\, ds = \pi.$$

In our experiments, $\mathbf{x}_0$ was chosen to be near the middle of the square $\Omega = [0,1]^2$, that is, $\mathbf{x}_0 = (0.50001, 0.50002)$ to avoid having the singularity at a grid point. The fundamental solution for the Laplace equation in two dimensions is

$$g(\mathbf{x}) = -\frac{1}{2\pi}\log|\mathbf{x}-\mathbf{x}_0|,$$

and so we took as boundary conditions $g(\mathbf{x}) = -\frac{1}{2\pi}\log|\mathbf{x}-\mathbf{x}_0|$ for $\mathbf{x} \in \partial\Omega$. Computational data for such computations with various meshes, polynomial degrees, and amplitudes A are shown in Table 4.1. We see that the approximation of the Green's function is only first order accurate, reflecting its limited regularity. Increasing the order of polynomials used is only of modest benefit. Increasing the amplitude of the approximate δ-function is useful up to a point, but making it larger is only of value if the mesh can be suitably refined.

Code to generate the data in Table 4.1 is given in Program 4.2.

4.3 Exercises

Exercise 4.1 *Compute the solution to the problem (4.1) with data g specified by (4.3). As solution metric, compute the norm of $u - g$. How does this depend on mesh size and polynomial order?*

Exercise 4.2 *Compute the solution to the problem (4.1) with data g_κ specified by (4.8). As solution metric, compute the norm of $u_\kappa - g_\kappa$. How does this depend on κ, mesh size and polynomial order?*

Exercise 4.3 *Let $\Omega = \{(x,y) \in \mathbb{R}^2 \,:\, x \in [0,1],\ y \in [0,1]\}$, $\Gamma = \left\{(x,0) \in \mathbb{R}^2 \,:\, x \in [\frac{1}{2},1]\right\}$. Solve $-\Delta u = 1$ in Ω with Dirichlet conditions on Γ and Neumann conditions on $\partial\Omega\backslash\Gamma$. See*

Figure 4.2 See if you can identify a constant c such that $u = cg_{2\pi} + v$ where v is smoother than u.

Exercise 4.4 *The method of manufactured solutions can benefit from many techniques of classical (analytical) applied mathematics. For example, the Laplace operator in polar coordinates is well known:*

$$\Delta f = f_{,rr} + r^{-1} f_{,r} + f_{,\theta\theta}.$$

Use this formula to verify that $\Delta g = 0$ for g specified by (4.3).

```
 1 from dolfin import *
 2 import sys,math
 3 
 4 parameters["form_compiler"]["quadrature_degree"] = 12
 5 
 6 meshsize=int(sys.argv[1])
 7 pdeg=int(sys.argv[2])
 8 
 9 # Create mesh and define function space
10 mesh = UnitSquareMesh(meshsize, meshsize)
11 V = FunctionSpace(mesh, "Lagrange", pdeg)
12 
13 # Define Dirichlet boundary ( 0.5 < x < 1 and y = 0 )
14 def gamma(x):
15   return x[0] > 0.5 and x[1] < DOLFIN_EPS
16 
17 # Define boundary condition
18 u0 = Constant(0.0)
19 bc = DirichletBC(V, u0, gamma)
20 
21 # Define variational problem
22 u = TrialFunction(V)
23 v = TestFunction(V)
24 f = Expression("1.0")
25 a = (inner(grad(u), grad(v)))*dx
26 L = f*v*dx
27 
28 # Compute solution
29 u = Function(V)
30 solve(a == L, u, bc)
31 
32 # Plot solution
33 plot(u, interactive=True)
```

Program 4.1: Code to implement the problem (4.10). In lines 14 and 15, we see code that defines the subset Γ of $\partial\Omega$ on which Dirichlet conditions are set.

```
from dolfin import *
import sys,math

meshsize=int(sys.argv[1])
pdeg=int(sys.argv[2])
amps=float(sys.argv[3])
ba=amps*math.pi
dfac=1/(4*math.pi)

# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
V = FunctionSpace(mesh, "Lagrange", pdeg)

# Define boundary condition
u0 = Expression("-d*log(pow(x[0]-0.50001,2)+pow(x[1]-0.50002,2))",d=dfac)
onec = Expression("1.0")
bc = DirichletBC(V, u0, DomainBoundary())

# Compute solution
u = Function(V)
solve(a == L, u, bc)
uone=project(onec,V)
fo=interpolate(f,V)
efo= 1-assemble(onec*fo*dx)
print " ",pdeg," ",meshsize," %.2e"%amps, \
      " %.2e"%errornorm(u0,u,norm_type=’l2’, degree_rise=0)," %.2e"%efo
```

Program 4.2: Code to implement the singularity problem (4.19).

Chapter 5

Laplace Plus Potential

We now augment the equation (2.1) with a potential Z, which is simply a function defined on Ω with real values. The PDE takes the form

$$-\Delta u + Zu = f \text{ in } \Omega \tag{5.1}$$

together with the boundary conditions (2.2). To formulate the variational equivalent of (2.1) with boundary conditions (2.2), we again use the variational space

$$V := \left\{ v \in H^1(\Omega) \ : \ v|_\Gamma = 0 \right\}. \tag{5.2}$$

Let Z denote a real valued function on Ω. The appropriate bilinear form for the variational problem is then

$$a_Z(u, v) = \int_\Omega \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) + Z(\mathbf{x})u(\mathbf{x})v(\mathbf{x})\, d\mathbf{x}. \tag{5.3}$$

In the case of homogeneous boundary conditions, we seek a solution $u \in V$ to

$$a_Z(u, v) = \int_\Omega f(\mathbf{x})v(\mathbf{x})\, d\mathbf{x} \quad \forall\, v \in V. \tag{5.4}$$

5.1 Bounded V

The simplest case is when Z is a constant, in which case (5.1) is often called the **Helmholtz equation**. This problem becomes interesting if Z is large, or equivalently, there is a small coefficient in front of Δ in (5.1). We propose Exercise 5.1 to explore this problem.

To understand coercivity in such problems, we first consider the eigenvalue problem

$$-\Delta u = \lambda u \text{ in } \Omega \tag{5.5}$$

together with the boundary conditions (2.2). Let us denote the solution of (5.5) by u_λ.

Let λ_0 be the lowest eigenvalue, and $u_{\lambda_0} \in V$ the corresponding eigenvector, for the eigenproblem problem (5.5), which we can write in variational form as

$$a_0(u_\lambda, v) = \lambda \int_\Omega u_\lambda(\mathbf{x})v(\mathbf{x})\, d\mathbf{x} \quad \forall\, v \in V, \tag{5.6}$$

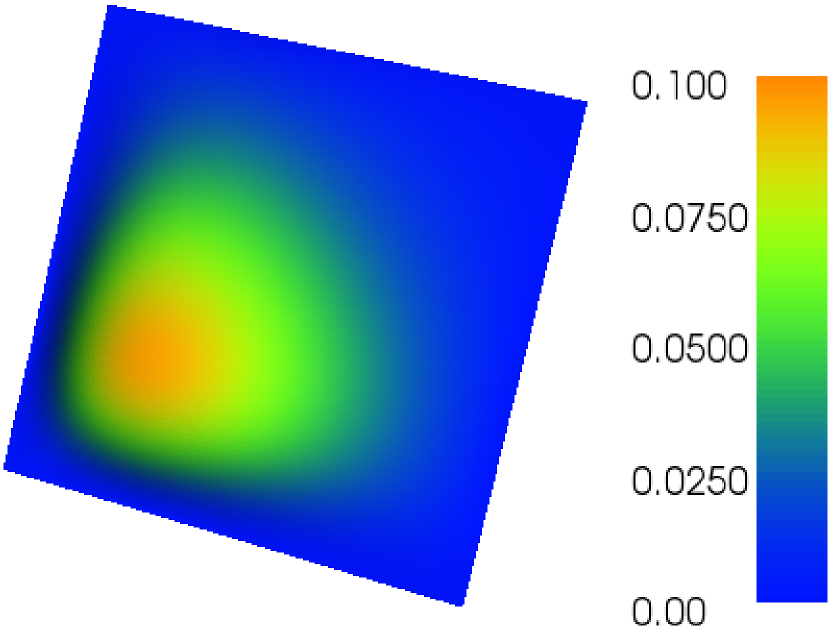

Figure 5.1: Asymptotic wavefunction perturbation computed with quartics on a mesh of size 100, with $L = 7$.

where $a_0(\cdot,\cdot)$ denotes the case $Z \equiv 0$, which is thus the same as the bilinear form $a(\cdot,\cdot)$ in (2.6). Coercivity (2.20) of the bilinear form $a_0(\cdot,\cdot)$ shows that $\lambda_0 > 0$. Moreover, if $Z(\mathbf{x}) > -\lambda_0$ for all $\mathbf{x} \in \Omega$, then the problem (5.4) is well-posed since it is still coercive.

5.2 Unbounded V

For certain unbounded potentials, it is still possible to show that (5.4) is well-posed. For example, if Z is either the Coulombic or graviational potential $Z(\mathbf{x}) = -|\mathbf{x}|^{-1}$, then the eigenvalue problem

$$a_Z(u_\lambda, v) = \lambda \int_\Omega u_\lambda(\mathbf{x}) v(\mathbf{x})\, d\mathbf{x} \quad \forall\, v \in V, \tag{5.7}$$

is well-posed, even in the case $\Omega = \mathbb{R}^3$. In this case, eigensolutions correspond to the wave functions of the hydrogen atom [140]. We propose Exercise 5.2 to explore this problem.

5.2.1 van der Waals interaction

The van der Waals interaction energy between two hydrogen atoms, separated by a distance R, is asymptotically of the form $-C_6 R^{-6}$ where the constant C_6 can be computed [48] by solving a two-dimensional PDE, as follows. Let $\Omega = [0,\infty]\times[0,\infty]$ and consider the PDE

$$-\frac{1}{2}\Delta u(r_1,r_2) + \left(\kappa(r_1)+\kappa(r_2)\right)u(r_1,r_2) = -\frac{1}{\pi}(r_1r_2)^2e^{-r_1-r_2} \text{ in } \Omega, \tag{5.8}$$

where the function κ is defined by $\kappa(r) = r^{-2} - r^{-1} + \frac{1}{2}$. The minimum of κ occurs at $r = 2$, and we have $\kappa(r) \geq \frac{1}{4}$. The problem (5.8) is well-posed in $H^1_0(\Omega)$, i.e., given Dirichlet conditions on the boundary of the quarter-plane Ω. The variational form for (5.8) is

$$a_\kappa(u,v) = \int_\Omega \tfrac{1}{2}\nabla u(r_1,r_2)\cdot\nabla v(r_1,r_2) + \big(\kappa(r_1)+\kappa(r_2)\big)u(r_1,r_2)\,v(r_1,r_2)\,dr_1dr_2, \tag{5.9}$$

defined for all $u,v \in H^1_0(\Omega)$, and it is coercive on $H^1_0(\Omega)$, since $\kappa(r_1)+\kappa(r_2) \geq \frac{1}{2}$. In particular,

$$a(v,v) \geq \frac{1}{2}\int_\Omega |\nabla v(r_1,r_2)|^2 + v(r_1,r_2)^2\,dr_1dr_2, \tag{5.10}$$

for all $v \in H^1_0(\Omega)$. The form $a(\cdot,\cdot)$ is continuous on $H^1_0(\Omega)$ because of the Hardy inequality

$$\int_0^\infty \big(u(r)/r\big)^2\,dr \leq 4\int_0^\infty \big(u'(r)\big)^2\,dr \tag{5.11}$$

for $u \in H^1_0(0,\infty)$. Note that it would not be continuous on all of $H^1(0,\infty)$; without the Dirichlet boundary condition, the form would be infinite for some functions in $H^1(0,\infty)$.

To be able to render this problem computationally feasible, we replace Ω by a square Ω_L of side L in length; $\Omega_L = [0,L]\times[0,L]$. Define $U(r_1,r_2) = u(Lr_1,Lr_2)$. Then $\Delta U(r_1,r_2) = L^2\Delta u(Lr_1,Lr_2)$. Thus

$$\begin{aligned}
-\frac{1}{2}L^{-2}\Delta U(r_1,r_2) = -\frac{1}{2}\Delta u(Lr_1,Lr_2) &= -\left(\kappa(Lr_1)+\kappa(Lr_2)\right)u(Lr_1,Lr_2)\\
&\quad -\frac{L^4}{\pi}(r_1r_2)^2e^{-Lr_1-Lr_2}\\
&= -\left(\hat\kappa_L(r_1)+\hat\kappa_L(r_2)\right)U(r_1,r_2) - \frac{L^4}{\pi}(r_1r_2)^2e^{-Lr_1-Lr_2},
\end{aligned} \tag{5.12}$$

where $\hat\kappa_L(r) = L^{-2}r^{-2} - L^{-1}r^{-1} + \frac{1}{2}$. Therefore U satisfies

$$-\tfrac{1}{2}L^{-2}\Delta U(r_1,r_2) + \left(\hat\kappa_L(r_1)+\hat\kappa_L(r_2)\right)U(r_1,r_2) = -\frac{L^4}{\pi}(r_1r_2)^2e^{-Lr_1-Lr_2}, \tag{5.13}$$

which we can pose with homogeneous Dirichlet boundary conditions ($u = 0$) on $\Omega_1 = [0,1]\times[0,1]$. Multiplying by $2L^2$, we obtain the equation

$$-\Delta U(r_1,r_2) + \left(\kappa_L(r_1)+\kappa_L(r_2)\right)U(r_1,r_2) = -\frac{2L^6}{\pi}(r_1r_2)^2e^{-Lr_1-Lr_2} = f(r_1,r_2), \tag{5.14}$$

where $\kappa_L(r) = 2r^{-2} - 2Lr^{-1} + L^2$ and

$$f(r_1,r_2) = -\frac{2L^6}{\pi}(r_1r_2)^2e^{-Lr_1-Lr_2}. \tag{5.15}$$

Thus we introduce the variational form

$$a_L(u,v)=\int_{[0,1]^2}\nabla u(r_1,r_2)\cdot\nabla v(r_1,r_2)+\big(\kappa_L(r_1)+\kappa_L(r_2)\big)u(r_1,r_2)\,v(r_1,r_2)\,dr_1dr_2 \tag{5.16}$$

Again, we have a variational problem in standard form: find $u_L\in V=H^1_0([0,1]^2)$ such that

$$a_L(u_L,v)=\int_{[0,1]^2}f(r_1,r_2)\,v(r_1,r_2)\,dr_1dr_2 \tag{5.17}$$

for all $v\in V$. The solution is shown in Figure 5.1 with $L=7$ computed on a mesh of size 100 with quartic Lagrange piecewise polynomials.

The code for solving this problem is given in Program 5.1.

The main quantity of interest [48, equation (3.25)] is

$$\begin{aligned}
C_6&=-\frac{32\pi}{3}\int_0^\infty\int_0^\infty r_1^2r_2^2e^{-(r_1+r_2)}u(r_1,r_2)\,dr_1dr_2\\
&\approx-\frac{32\pi}{3}\int_0^L\int_0^L r_1^2r_2^2e^{-(r_1+r_2)}u(r_1,r_2)\,dr_1dr_2\\
&\approx-\frac{32\pi}{3}\int_0^L\int_0^L r_1^2r_2^2e^{-(r_1+r_2)}U(r_1/L,r_2/L)\,dr_1dr_2\\
&=-\frac{32\pi}{3}\int_0^1\int_0^1 L^4R_1^2R_2^2e^{-(LR_1+LR_2)}U(R_1,R_2)\,L^2\,dR_1dR_2\\
&=-\frac{16\pi^2}{3}\frac{2L^6}{\pi}\int_0^1\int_0^1 R_1^2R_2^2e^{-(LR_1+LR_2)}U(R_1,R_2)\,dR_1dR_2\\
&=\frac{16\pi^2}{3}\int_0^1\int_0^1 f(R_1,R_2)\,U(R_1,R_2)\,dR_1dR_2,
\end{aligned}\tag{5.18}$$

where we made the substitution $r_i=LR_i$, $i=1,2$, and f is defined in (5.15).

To avoid singularities in the coefficients, we modified the potential to be

$$\kappa_L^\epsilon(r)=2(\epsilon+r)^{-2}-2L(\epsilon+r)^{-1}+L^2. \tag{5.19}$$

Computational results are shown in Table 5.1. The results were insensitive to ϵ for $\epsilon\le10^{-9}$.

5.2.2 Another formulation

The singularity $(r_i)^{-2}$ is difficult to deal with. But we can integrate by parts to soften its effect, as follows:

$$\begin{aligned}
\int_\Omega(r_i)^{-2}u(r_1,r_2)\,v(r_1,r_2)\,dr_1dr_2&=-\int_\Omega\Big(\frac{\partial}{\partial r_i}(r_i)^{-1}\Big)u(r_1,r_2)\,v(r_1,r_2)\,dr_1dr_2\\
&=\int_\Omega(r_i)^{-1}\frac{\partial}{\partial r_i}\big(u(r_1,r_2)\,v(r_1,r_2)\big)\,dr_1dr_2,
\end{aligned}\tag{5.20}$$

degree	quadrature	mesh number	C_6 error	ϵ	L	time
4	6	100	4.57e-07	1.00e-09	15.0	1.47
4	8	80	4.57e-07	1.00e-09	15.0	0.917
4	10	100	4.57e-07	1.00e-09	15.0	1.595
4	10	200	4.56e-07	1.00e-09	15.0	7.469
4	8	250	4.56e-07	1.00e-09	15.0	12.5
2	4	400	3.97e-07	1.00e-09	15.0	1.912
2	4	600	4.45e-07	1.00e-09	15.0	4.738
2	3	600	4.41e-07	1.00e-09	15.0	4.747
2	2	250	-1.22e-07	1.00e-09	15.0	0.786
2	3	250	-6.78e-08	1.00e-09	15.0	0.789
2	3	255	-2.74e-08	1.00e-09	15.0	0.786
2	3	260	9.14e-09	1.00e-09	15.0	0.792
2	3	265	4.23e-08	1.00e-09	15.0	0.837
2	4	250	5.41e-08	1.00e-09	15.0	0.788
2	4	240	-1.87e-08	1.00e-09	15.0	0.739

Table 5.1: Using finite element computation of $C_6 = 6.4990267054$ [48]. The potential was modified as in (5.19). Computations were done with 4 cores via MPI and a PETSc Krylov solver. Error values were the same for $\epsilon = 10^{-9}$ and $\epsilon = 10^{-12}$.

where for simplicity we define $\Omega = [0,1]^2$ here and for the remainder of this subsection. We have assumed that $u, v \in V = H_0^1(\Omega)$ in (5.20). Thus

$$\begin{aligned}
&\int_\Omega \big((r_1)^{-2} + (r_2)^{-2}\big) u(r_1,r_2)\, v(r_1,r_2)\, dr_1 dr_2 \\
&\quad = \int_\Omega \big((r_1)^{-1}, (r_2)^{-1}\big) \cdot \nabla\big(u(r_1,r_2)\, v(r_1,r_2)\big)\, dr_1 dr_2 \\
&\quad = \int_\Omega \big((r_1)^{-1}, (r_2)^{-1}\big) \cdot \Big(\big(\nabla u(r_1,r_2)\big)\, v(r_1,r_2) + \big(\nabla v(r_1,r_2)\big)\, u(r_1,r_2)\Big)\, dr_1 dr_2
\end{aligned} \tag{5.21}$$

Thus we introduce a new variational form (cf. (5.16))

$$\begin{aligned}
\hat{a}_L^\epsilon(u,v) = &\int_\Omega \nabla u(r_1,r_2) \cdot \nabla v(r_1,r_2) + \big(\hat{\kappa}_L^\epsilon(r_1) + \hat{\kappa}_L^\epsilon(r_2)\big) u(r_1,r_2)\, v(r_1,r_2)\, dr_1 dr_2 \\
+ 2 &\int_\Omega \widehat{\boldsymbol{\beta}}(r_1,r_2) \cdot \big(\big(\nabla u(r_1,r_2)\big)\, v(r_1,r_2) + u(r_1,r_2)\big(\nabla v(r_1,r_2)\big)\big)\, dr_1 dr_2
\end{aligned} \tag{5.22}$$

where

$$\widehat{\boldsymbol{\beta}}(r_1,r_2) = \big((r_1+\epsilon)^{-1}, (r_2+\epsilon)^{-1}\big), \qquad \hat{\kappa}_L^\epsilon(r) = -2L(r+\epsilon)^{-1} + 2L^2. \tag{5.23}$$

degree	mesh number	L^2 difference	time
1	256	6.86e-02	1.11
1	512	5.34e-02	5.1
1	1024	4.72e-02	29
2	512	4.54e-02	23
4	256	4.48e-02	18
8	128	4.47e-02	25
8	8	7.74e-02	23

Table 5.2: Boundary layer problem with $\epsilon = 10^{-6}$. Degree refers to the polynomial degree, mesh number indicates the number of edges along each boundary side as indicated in Figure 3.5, L^2 difference is $\|u - f\|_{L^2([0,1]^2)}$, and time is in seconds.

5.3 Exercises

Exercise 5.1 *Let $\epsilon > 0$. Consider the problem*

$$-\epsilon\Delta u_\epsilon + u_\epsilon = f \text{ in } \Omega = [0,1]^2,$$

together with boundary conditions (2.2), where f is held fixed independent of ϵ. This is known as a **singular perturbation** *problem. Give conditions for which you would expect $u_\epsilon \to f$ as $\epsilon \to 0$. Do a simple example in which f does not satisfy the boundary conditions (2.2), for example, take $f \equiv 1$ and homogenous Dirichlet conditions on all of $\partial\Omega$, and see what happens for small ϵ. Does $u_\epsilon \to f$ except in a small boundary layer? In what norm(s)? If the homogeneous boundary conditions hold on only a part Γ of the boundary, is there a boundary layer away from Γ? Compare your results with those in Table 5.2 which corresponds to the choices $f \equiv 1$ and $\epsilon = 10^{-6}$. Note that the best results require a large number of nodes to resolve the boundary layer, but among the different choices (linear, quadratics, and so forth), the results are about the same and take about the same time to compute. In particular, using a high-order polynomial does not provide particular benefit in this case.*

Exercise 5.2 *Consider the problem*

$$-\tfrac{1}{2}\Delta u(\mathbf{x}) - \frac{1}{|\mathbf{x}|}u(\mathbf{x}) = -\tfrac{1}{2}e^{-|\mathbf{x}|} \text{ for } x \in \mathbb{R}^3$$

and the condition at infinity that $u(\mathbf{x}) \to 0$ as $\mathbf{x} \to \infty$. First truncate the infinite domain to a box $[-L, L]^3$, and impose homogeneous Dirichlet conditions on the boundary of the box. Make a variational formulation for this problem and solve for $u = u_L$ for various values of L. Compare with the exact solution $u(\mathbf{x}) = ce^{-|\mathbf{x}|}$. Evaluate c by plotting $u_L(\mathbf{x})e^{+|\mathbf{x}|}$.

```
from dolfin import *
import sys,math
from timeit import default_timer as timer
parameters["form_compiler"]["quadrature_degree"] = 12
startime=timer()
meshsize=int(sys.argv[1])
pdeg=int(sys.argv[2])
ell=float(sys.argv[3])
myeps=float(sys.argv[4])
# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
V = FunctionSpace(mesh, "Lagrange", pdeg)
# Define boundary condition
u0 = Constant(0.0)
bc = DirichletBC(V, u0, DomainBoundary())
# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("-(2.0*pow(el,6)/mypi)*pow(x[0]*x[1],2)* \
               exp(-el*x[0]-el*x[1])",el=ell,mypi=math.pi)
kay = Expression("(2.0/(me+pow(x[0],2)))-2.0*(el/(me+x[0])) \
                 +(2.0/(me+pow(x[1],2)))-2.0*(el/(me+x[1])) \
                 +2.0*el*el",me=myeps,el=ell)
a = (inner(grad(u), grad(v))+kay*u*v)*dx
m = u*v*dx
L = f*v*dx
# Compute solution
u = Function(V)
solve(a == L, u, bc)
aftersolveT=timer()
mfu= (16.0*pow(math.pi,2)/3.0)*assemble(u*f*dx)
mer=mfu-6.49902670540
totime=aftersolveT-startime
print " ",pdeg," ",meshsize," %.2e"%mer," %.2e"%myeps, \
      " %.1f"%ell," %.3f"%totime
```

Program 5.1: Code to implement the problem (5.17).

Chapter 6

Variational Formulations in One Dimension

Here we develop all of the concepts of the variational formulation for differential equations. By considering the one-dimensional case, we are able to explain in detail many of the concepts.

6.1 Exact solution

Consider the two-point boundary value problem

$$\begin{aligned} -\frac{d^2u}{dx^2} &= f \text{ in } (0,1) \\ u(0) = g_0, \quad u'(1) &= g_1. \end{aligned} \tag{6.1}$$

The solution can be determined from f via two integrations. First of all, we can write

$$\frac{du}{dx}(t) = \int_t^1 f(s)\,ds + g_1 \tag{6.2}$$

using the boundary condition at $x = 1$. Integrating again shows that

$$u(x) = \int_0^x \int_t^1 f(s)\,ds\,dt + g_1 x + g_0 \tag{6.3}$$

using the boundary condition at $x = 0$. This shows that (6.1) is well-posed.

It will not be this easy to demonstrate the well-posedness of all the differential equations studied here. However, every investigation should (in principle) begin with this step.

The variational approach to differential equations is a powerful technique for studying their well-posedness and behavior as well as a critical step in generating a broad class of discretization schemes. It is often called the "weak" formulation as it allows the differential equation to be posed in a set of functions that allows a broader range of solutions. This generalization is not just a mathematical curiosity; rather it often allows the problems of most physical relevance to be addressed.

6.2 Weak Formulation of Boundary Value Problems

Suppose that u is the solution of (6.1) with $g_0 = 0$. Let v be any (sufficiently regular) function such that $v(0) = 0$. Then integration by parts yields

$$(f,v)_{L^2([0,1])} = \int_0^1 f(x)v(x)dx = \int_0^1 -u''(x)v(x)dx = \int_0^1 u'(x)v'(x)dx - g_1v(1). \tag{6.4}$$

Define

$$a(u,v) := \int_0^1 u'(x)v'(x)dx \tag{6.5}$$

and

$$V = \left\{ v \in L^2([0,1]) \ : \ a(v,v) < \infty \text{ and } v(0) = 0 \right\}. \tag{6.6}$$

Then we can say that the solution u to (6.1) is characterized by

$$u \in V \quad \text{such that} \quad a(u,v) = (f,v)_{L^2([0,1])} + g_1v(1) \qquad \forall v \in V, \tag{6.7}$$

which is called the **variational formulation** or **weak formulation** of (6.1).

The relationship (6.7) is called "variational" because the function v is allowed to vary arbitrarily. It has a natural interpretation in the setting of Hilbert spaces [40]. The Dirichlet boundary condition $u(0) = 0$ is called an **essential boundary condition** because it appears in the variational space. The Neumann boundary condition $u'(1) = 0$ is called an **natural boundary condition** because it does not appear in the variational space but rather is implied in the formulation.

Inhomogeneous Dirichlet boundary conditions are handled as follows in the variational formulation. Let u_0 be some function satisfying the inhomogeneous Dirichlet boundary conditions (but not necessarily the Neumann boundary conditions). Then

$$u - u_0 \in V \quad \text{such that} \quad a(u,v) = (f,v)_{L^2([0,1])} + g_1v(1) \qquad \forall v \in V. \tag{6.8}$$

Equivalently, this can be written as $u = w + u_0$ where

$$w \in V \quad \text{such that} \quad a(w,v) = (f,v)_{L^2([0,1])} + g_1v(1) - a(u_0,v) \qquad \forall v \in V. \tag{6.9}$$

Note that the general problem (6.8) can be written

$$w \in V \quad \text{such that} \quad a(w,v) = F(v) \qquad \forall v \in V \tag{6.10}$$

where F denotes a **linear functional** on the space V, i.e., a linear function defined for any $v \in V$ having a single real number as its value.

6.2.1 Linear functionals

The right-hand side of (6.9) can be written succintly as

$$F(v) = (f,v)_{L^2([0,1])} + g_1v(1) - a(u_0,v) \qquad \forall v \in V. \tag{6.11}$$

The expression F is called a linear functional because (a) it is linear and (b) it has scalar values. By linear, we mean that $F(u+av)=F(u)+aF(v)$ for any scalar a and any $u,v\in V$.

The critical condition on a linear functional for success in a variational formulation is that it be *bounded* or *continuous*. A **bounded linear functional** (equivalently a **continuous linear functional**) F on a normed space V must satisfy

$$|F(v)|\le C_F\|v\|_V \quad \forall v\in V. \tag{6.12}$$

A natural norm $\|\cdot\|_V$ for the space V defined in (6.6) is

$$\|v\|_a=\sqrt{a(v,v)}.$$

The smallest possible constant C_F for which this holds is called the **dual norm** of F and is defined by

$$\|F\|_{V'}:=\sup_{0\neq v\in V}\frac{|F(v)|}{\|v\|_V}. \tag{6.13}$$

The main point is that all the linear forms considered so far *are* bounded (see Exercise 6.11), in particular the Dirac δ-function, defined by $\delta(v)=v(1)$, as we show in Section 6.2.2. But it is also easy to think of others which are not, such as

$$F(v):=v'(x_0) \tag{6.14}$$

for some $x_0\in[0,1]$. This form is linear, but consider what it should do for the function $v\in V$ given by

$$v(x):=|x-x_0|^{2/3} \tag{6.15}$$

(see Exercise 6.10).

The general variational formulation (6.10) can be shown to be completely equivalent to the orignal differential equation (see [40, Theorem 0.1.4]). Moreover, it actually provides a framework that allows less regular data (arbitrary continuous linear functionals for F) as required by important physical applications. The expression $a(\cdot,\cdot)$ is called a **bilinear functional** on the space V, since it is a bilinear function defined on the Cartesian product $V\times V$ having a single real number as its value. If we fix one of the variables of a bilinear form, it yields a linear form in the remaining variable.

6.2.2 Sobolev's inequality

Consider the linear form $F(v)=v(x_0)$ for some $x_0\in[0,1]$. We want to prove that this is bounded on V. We write a function as the integral of its derivative and begin to estimate:

$$v(t)=\int_0^t v'(x)\,dx=\int_0^1 v'(x)w'(x)\,dx=a(v,w), \tag{6.16}$$

where the function $w\in V$ is defined by

$$w(x)=\begin{cases} x & 0\le x\le t\\ t & x\ge t\end{cases} \tag{6.17}$$

One benefit of our loosened notion of derivative is that such functions are indeed in V, even though the derivative of w is discontinuous. By the Cauchy-Schwarz inequality (2.30)

$$|v(t)| = |a(v,w)| \leq \|v\|_a \|w\|_a = \sqrt{t}\|v\|_a \leq \|v\|_a \tag{6.18}$$

for all $t \in [0,1]$. Inequality (6.18) is called Sobolev's inequality, and V is an example of a Sobolev space. What Sobolev's inequality inequality tells us is that, even though the functions in V are not smooth in the classical sense (derivatives can even be infinite at isolated points), they nevertheless have some type of classical regularity, namely continuity in this case.

Note that the first step in (6.16) uses the fact that for $v \in V$, $v(0) = 0$. This subtle point is nevertheless essential, since (6.18) is clearly false if this boundary condition is not available. In particular, if v is a constant function, then the right-hand-side of (6.18) is zero for this v whereas the left-hand-side is not (unless $v \equiv 0$). Sobolev's inequality holds in a more general setting, not requiring boundary conditions, but only when the bilinear form is augmented in some way that renders an inner-product.

6.2.3 Natural boundary conditions

We saw that the 'natural' boundary condition, e.g., $u'(1) = 0$ in (6.1) when $g_1 = 0$, disappears in the variational formulation (6.7). But if these are in some sense equivalent formulations (they are), then the natural boundary condition must be encoded in the variational formulation is some way. We can see this by reversing the process used to go from (6.1) to (6.7). So suppose that u satisfies (6.7), and also assume that it is smooth enough for us to integrate by parts:

$$\begin{aligned}(f,v)_{L^2([0,1])} &= \int_0^1 u'(x)v'(x)\,dx = \int_0^1 -u''(x)v(x)\,dx + \big(u'v\big)\Big|_0^1 \\ &= \int_0^1 -u''(x)v(x)\,dx + u'(1)v(1).\end{aligned} \tag{6.19}$$

Choosing first $v \in V$ that vanishes at $x = 1$, we conclude that

$$\int_0^1 \big(f + u''(x)\big)v(x)\,dx = 0$$

for all such v. From this, one can show that we necessarily have $-u'' = f$. Inserting this fact in (6.19), we conclude that $u'(1) = 0$ simply by taking a single v such that $v(1) \neq 0$, e.g., $v(x) = x$. Thus the natural boundary condition emerges from the variational formulation "naturally." And as an intermediate step, we see that u satisfies the first equation in (6.1), proving the equivalence of (6.1) and (6.7).

6.3 Galerkin Approximation

Let $V_h \subset V$ be any (finite dimensional) subspace. Let us consider (6.7) with V replaced by V_h, namely

$$u_h \in V_h \quad \text{such that} \quad a(u_h, v) = (f,v)_{L^2([0,1])} \qquad \forall v \in V_h. \tag{6.20}$$

Then (6.20) represents a square, finite system of equations for u_h which can easily be seen to be invertible [40]. Note how easily a discrete scheme for approximating (6.1) can be defined.

A matrix equation is derived by writing (6.20) in terms of a basis $\{\phi_i : 1 \le i \le n\}$ of V_h. Write u_h in terms of this basis, i.e.,

$$u_h = \sum_{j=1}^{n} U_j \phi_j$$

where the coefficients U_j are to be determined. Define

$$A_{ij} = a(\phi_j, \phi_i), \qquad F_i = (f, \phi_i) \quad \text{for} \quad i, j = 1, ..., n. \tag{6.21}$$

Set $\mathbf{U} = (U_j), \mathbf{A} = (A_{ij})$ and $\mathbf{F} = (F_i)$. Then (6.20) is equivalent to solving the (square) matrix equation

$$\mathbf{AU} = \mathbf{F}. \tag{6.22}$$

If we write $v = \sum V_j \phi_j$ then

$$a(u_h, v) = \mathbf{V}^t \mathbf{AU} \tag{6.23}$$

Therfore the symmetry and positivity of the form $a(\cdot, \cdot)$ is equivalent to the symmetry and positive-definiteness of $\mathbf{A}$. The invertibility of the system can be proved simply by checking that there are no nonzero $v \in V_h$ such that $0 = a(v, v)$. In the current case, this would imply that v is constant. Since $v \in V_h \subset V$ implies $v(0) = 0$, we must have $v \equiv 0$. Therefore, the solution u_h to (6.20) exists and is unique.

The matrix $\mathbf{A}$ is often referred to as the **stiffness matrix**, a name coming from corresponding matrices in the context of structural problems. Another important matrix is the **mass matrix**, namely

$$M_{ij} = (\phi_j, \phi_i)_{L^2([0,1])} \quad \text{for} \quad i, j = 1, ..., n. \tag{6.24}$$

If $f \in V$ with $f = \sum \tilde{F}_j \phi_j$ then (6.20) is equivalent to solving the matrix equation

$$\mathbf{AU} = \mathbf{M}\tilde{\mathbf{F}}. \tag{6.25}$$

6.3.1 Piecewise Polynomials – Finite Elements

Let $0 = x_0 < x_1 < ... < x_n = 1$ be a partition of $[0, 1]$, and let V_h be the linear space of functions v such that

- v is continuous everywhere
- $v|_{[x_{i-1}, x_i]}$ is a linear polynomial, $i = 1, ..., n$, and
- $v(0) = 0$.

The function space just defined can be described as the set of **continuous piecewise linear** functions with respect to the mesh (x_i).

For each $i = 1, .., n$ define ϕ_i by the requirement that $\phi_i(x_j) = \delta_{ij}$ the Kronecker delta. Then $\{\phi_i \ : \ 1 \leq i \leq n\}$ is called a **nodal basis** for V_h, and $\{v(x_i)\}$ are the **nodal values** of a function v. (The points $\{x_i\}$ are called the **nodes**.)

A function space consisting of **continuous piecewise quadratic** functions, with respect to the mesh (x_i), can be defined similarly. Let V_h be the linear space of functions v such that

- v is continuous everywhere
- $v|_{[x_{i-1},x_i]}$ is a quadratic polynomial, $i = 1, ..., n$, and
- $v(0) = 0$.

However, now there are additional nodes in the middle of each *element* $[x_{i-1}, x_i]$, i.e., at $(x_i + x_{i-1})/2$. Now the nodal numbering gets a bit complicated. Let $y_{2i} = x_i$ and let $y_{2i-1} = (x_i - x_{i-1})/2$ for $i = 1, \ldots, n$. Then the nodal basis is defined by $\phi_i(y_j) = \delta_{ij}$ for $i, j = 1, \ldots, 2n$

The Galerkin method using piecewise polynomials spaces described in terms of nodal values is called the finite-element method.

6.3.2 Relationship to Difference Methods

The stiffness matrix **A** as defined in (6.22), using the basis $\{\phi_i\}$ described in Section 6.3.1, can be interpreted as a difference operator. Let $h_i = x_i - x_{i-1}$. Then the matrix entries $A_{ij} = a(\phi_i, \phi_j)$ can be easily calculated to be

$$A_{ii} = h_i^{-1} + h_{i+1}^{-1}, A_{i,i+1} = A_{i+1,i} = -h_{i+1}^{-1} \qquad (i = 1, ..., n-1) \tag{6.26}$$

and $A_{nn} = h_n^{-1}$ with the rest of the entries of **A** being zero. Similarly, the entries of **F** can be approximated if f is sufficiently smooth:

$$(f, \phi_i) = \frac{1}{2}(h_i + h_{i+1})(f(x_i) + \mathcal{O}(h)) \tag{6.27}$$

where $h = \max h_i$. Thus, the $i - th$ equation of $\mathbf{AU} = \mathbf{F}$ (for $1 \leq i \leq n-1$) can be written as

$$\frac{-2}{h_i + h_{i+1}} \left[\frac{U_{i+1} - U_i}{h_{i+1}} - \frac{U_i - U_{i-1}}{h_i} \right] = \frac{2(f, \phi_i)}{h_i + h_{i+1}} = f(x_i) + \mathcal{O}(h). \tag{6.28}$$

The difference operator on the left side of this equation can also be seen to be an $\mathcal{O}(h)$ accurate approximation to the differential operator $-d^2/dx^2$. For a uniform mesh, the equations reduce to the familiar difference equations

$$-\frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} = f(x_i) + \mathcal{O}(h^2) \tag{6.29}$$

as we will see later in (6.50). Thus the finite difference and finite element discretization techniques can be seen to produce the same set of equations in many cases.

Even thought the difference method (6.28) is formally only first order accurate, one can show using the variational framework [40] that the resulting error is second order accurate:

$$e_h := \max_{1\leq i\leq 2n} |u(y_n) - u_n| \leq C_f h^2 \tag{6.30}$$

This shows that it may be useful to view a difference method as a variational method (if possible) for the purposes of analysis.

Note the slight difference between (6.53) and the last equation in (6.26). Because the variational form $a(\cdot,\cdot)$ is symmetric, the Galerkin method will always yield a symmetric matrix as is the case in (6.26). The last equation in (6.26) and (6.53) differ only by a simple factor of two, but this leads to a non-symmetric matrix. In applying boundary conditions with finite difference methods, care must be exercised to retain the symmetry of the original differential equation.

The system of equations obtained for the nodal variables (u_n) in the case of the Galerkin method using continuous piecewise quadratics does not look like a conventional finite difference method. The equations associated with the internal nodes are different from the ones associated with the subdivision points. On the other hand, they yield a more accurate method, satisfying

$$e_h := \max_{1\leq i\leq 2n} |u(y_n) - u_n| \leq C_f h^3. \tag{6.31}$$

6.4 Coercivity of the Variational Problem

The variational form $a(\cdot,\cdot)$ introduced in (6.5) is *coercive* on the corresponding spaces V (see [40]): there is a constant γ depending only on Ω and Γ such that

$$\|v\|^2_{H^1(\Omega)} \leq \gamma a(v,v) \quad \forall v \in V. \tag{6.32}$$

The proof of this is elementary. All we need to show is that

$$\|v\|^2_{L^2(\Omega)} \leq C a(v,v) \quad \forall v \in V, \tag{6.33}$$

from which (6.32) follows with constant $\gamma = C + 1$. To prove (6.33), we apply Sobolev's inequality (6.18). Thus

$$\int_0^1 v(t)^2\,dt \leq a(v,v)\int_0^1 t\,dt \leq \tfrac{1}{2}a(v,v) \tag{6.34}$$

which completes the proof of (6.33), with $C = 1/2$.

Note that our proof of Sobolev's inequality (6.18) uses the fact that for $v \in V$, $v(0) = 0$. This subtle point is nevertheless essential, since (6.32) is clearly false if this boundary condition is not available. In particular, if v is a constant function, then the right-hand-side of (6.32) is zero for this v whereas the left-hand-side is not (unless $v \equiv 0$).

From (6.32), it follows that the problem (6.10) is well-posed. In particular, we easily see that the solution to the problem must be unique, for if F is identically zero then so is the solution. In the finite-dimensional case, this uniqueness also implies existence, and a similar result holds in the setting of infinite dimensional Hilbert spaces such as V. Moreover, the coercivity condition immediately implies a stability result, namely

$$\|u\|_{H^1(\Omega)} \leq \frac{\gamma a(u,u)}{\|u\|_{H^1(\Omega)}} = \gamma \frac{F(u)}{\|u\|_{H^1(\Omega)}} \leq \gamma \|F\|_{H^{-1}(\Omega)}. \tag{6.35}$$

Here we are using the notation $\|F\|_{H^{-1}(\Omega)}$ for the dual norm of F in the dual space of $H^1(\Omega)$, i.e., $H^{-1}(\Omega) := (H_0^1(\Omega))'$ [40]. The same result holds for a discrete approximation as well.

As a byproduct, (2.21) proves continuity of the solution as a function of the data since the problem is linear. In particular, if F_i, $i = 1,2$, are two bounded linear forms, and u_i denotes the corresponding solutions to (6.10), then

$$\|u_1 - u_2\|_{H^1(\Omega)} \leq \gamma \|F_1 - F_2\|_{H^{-1}(\Omega)}. \tag{6.36}$$

6.5 More Variational Formulations

Consider the two-point boundary value problem

$$\begin{aligned} -\frac{d^2u}{dx^2} + \alpha(x)\frac{du}{dx} + \beta(x)u = f \text{ in } (0,1) \\ u(0) = g_0, \quad u'(1) = g_1. \end{aligned} \tag{6.37}$$

Then integration by parts can again be used to derive the variational formulation

$$a(u,v) = (f,v)_{L^2([0,1])} \quad \forall v \in V \tag{6.38}$$

where

$$a(u,v) := \int_0^1 u'(x)v'(x) + \alpha(x)u'(x)v(x) + \beta(x)u(x)v(x)\,dx. \tag{6.39}$$

This variational problem introduces a number of difficulties which will be addressed subsequently, such as how to integrate the expressions involving α and β. Typically this is done by numerical quadrature.

The question of coercivity of the form (6.39) can be addressed in at least simple cases. If $\beta \equiv 0$ and α is constant, then

$$a(v,v) = \int_0^1 v'(x)^2 + \tfrac{1}{2}\alpha(v^2)'(x)\,dx = \int_0^1 v'(x)^2\,dx + \tfrac{1}{2}\alpha v(1)^2 \quad \forall v \in V. \tag{6.40}$$

If $\alpha > 0$, then this is coercive. Regarding conditions needed on β to retain coercivity, see Exercise 6.14.

Nonlinear problems such as (6.57) can also be formulated variationally, as

$$a(u,v) + n(u,v) = (f,v)_{L^2([0,1])} \quad \forall v \in V \tag{6.41}$$

where $a(\cdot,\cdot)$ is as in (6.39) with $\alpha \equiv 0$ and $\beta(x) \equiv 4$. The nonlinearity has been separated for convenience in the form

$$n(u,v) = 6\int_0^1 u(x)^2 v(x)\,dx = 6\,(u^2,v)_{L^2([0,1])}. \tag{6.42}$$

A Galerkin method for a space with a basis $\{\phi_i\ :\ i = 1,\ldots,n\}$ can be written as a system of nonlinear equations

$$F_i(u) := a(u,\phi_i) + n(u,\phi_i) - (f,\phi_i)_{L^2([0,1])} = 0 \tag{6.43}$$

Writing $u = \sum_j U_j\phi_j$, Newton's method for this system of equations for (U_j) can be derived. However, it can also be cast in variational form as follows.

Instead of using a basis function, let us define a function F with coordinates parametrized by an arbitrary $v \in V$:

$$F_v(u) := a(u,v) + n(u,v) - (f,v)_{L^2([0,1])} \tag{6.44}$$

If $v = \phi_i$ then of course we have the previous function. Newton's method requires us to compute the derivative of F with respect to its "coordinates" which in this case correspond to elements of V. The derivative of F_v at u in the direction of $w \in V$ is, as always, a limit of a difference quotient,

$$\frac{F_v(u+\epsilon w) - F_v(u)}{\epsilon}, \tag{6.45}$$

as $\epsilon \to 0$. Expanding, we find that

$$\begin{aligned} F_v(u+\epsilon w) - F_v(u) &= \epsilon a(w,v) + 6\left((u+\epsilon w)^2 - u^2, v\right)_{L^2([0,1])} \\ &= \epsilon a(w,v) + 6\left(2\epsilon uw + \epsilon^2 w^2, v\right)_{L^2([0,1])}. \end{aligned} \tag{6.46}$$

Therefore

$$\lim_{\epsilon\to 0}\frac{F_v(u+\epsilon w) - F_v(u)}{\epsilon} = a(w,v) + 12\,(uw,v)_{L^2([0,1])} \tag{6.47}$$

for any $w \in V$. It is then easy to see (see Exercise 6.19) that Newton's method can be characterized by

$$\begin{gathered} u \leftarrow u - w \quad \text{where } w \text{ solves} \\ a(w,v) + 12\,(uw,v)_{L^2([0,1])} = a(u,v) + n(u,v) - (f,v)_{L^2([0,1])} \quad (= F_v(u)) \quad \forall v \in V \end{gathered} \tag{6.48}$$

6.6 Other Galerkin Methods

6.6.1 Spectral Elements – P Method

The definition of continuous piecewise polynomials of arbitrary degree P can be accomplished by continuing the pattern set by the linear and quadratic cases. There are $P-1$ nodal points in the interior of each interval in the subdivision, but otherwise the definition is the same. The use of high-order piecewise polynomials in Galerkin approximations goes by various names. Since the degree P can be used as the approximation parameter (that is, convergence is achieved by letting P increase), it is often called the 'P' method. It also goes by the name "spectral element" method because there are close similarities with the so-called spectral methods, yet there is the possibility of subdividing the domain using "elements."

6.6.2 Trigonometric Polynomials – Spectral Methods

Choosing spaces of trigonometric polynomials in the Galerkin approximation method leads to the class of discretizations which are popularly known as Spectral Methods.

6.7 Finite Difference Methods

The simplest way to approximate a differential equation often is to replace a differential operator with a difference operator. In this way we get the approximation for (6.1)

$$-u(x-h)+2u(x)-u(x+h)\approx h^2 f(x) \tag{6.49}$$

where $h>0$ is the mesh size to be used. Choosing $x=x_n:=nh$ for $n=0,1,\ldots,N$ where $N=1/h$, we get a system of linear equations

$$-u_{n-1}+2u_n-u_{n+1}=h^2 f(x_n) \tag{6.50}$$

where $u_n\approx u(x_n)$, which is the same as the finite element discretization which leads to (6.29).

One of the shortcomings of the finite-difference approach is that the boundary conditions have to be handled in an *ad hoc* way. The boundary condition at $x=0$ translates naturally into $u_0=g_0$. Thus, (6.50) for $n=1$ becomes

$$2u_1-u_2=h^2 f(x_1)+g_0. \tag{6.51}$$

However, the derivative boundary condition at $x=1$ must be approximated by a difference equation. A natural one to use is

$$u_{N+1}-u_{N-1}=2hg_1 \tag{6.52}$$

using a difference over an interval of length $2h$ centered at x_N. The second-difference (6.50) is itself a difference of first-differences over intervals of length h so there is some inconsistency, however both are centered differences (and therefore second-order accurate). See Exercise 6.10 for an example based on a non-centered difference approximation to the derivative boundary condition. Using (6.52), (6.50) for $n=N$ becomes

$$-2u_{N-1}+2u_N=h^2 f(x_N)+2hg_1. \tag{6.53}$$

Algebraically, we can express the finite difference method as

$$\mathbf{AU}=\mathbf{F} \tag{6.54}$$

where $\mathbf{U}$ is the vector with entries u_n and $\mathbf{F}$ is the vector with entries $h^2 f(x_n)$ appropriately modified at $n = 1$ and $n = N$ using the boundary data. The matrix $\mathbf{A}$ has all diagonal entries equal to 2. The first sub- and super-diagonal entries are all equal to -1, except the last sub-diagonal entry, which is -2.

6.7.1 `octave` Implementation

The creation of the matrix $\mathbf{A}$ in `octave` can be achieved by various techniques. However, to get reasonable performance for large N, you must restrict to sparse operations, as follows. For simplicity, we us consider the case where the differential equation to be solved is

$$\begin{aligned} -u''(x) &= \sin(x) \text{ for } x \in [0, \pi] \\ u(0) &= u(\pi) = 0. \end{aligned} \tag{6.55}$$

Then $u(x) = \sin(x)$ for $x \in [0, \pi]$. To create the difference operator A for this problem in a sparse format, you must specify only the non-zero entries of the matrix, that is, you give a list of triples: (i, j, A_{ij}). This is followed by an operation that amalgamates these triples into a sparse matrix structure; in `octave`, this operation is called `sparse`. The `octave` code for this is shown in Program 6.1.

```
dx=pi/(N+1);          % mesh size with N points
%                       define the diagonal matrix entries
i(1:N)=1:N;
j(1:N)=1:N;
v(1:N)= 2/(dx*dx);
%                       define the above-diagonal matrix entries
i(N+(1:(N-1)))=(1:(N-1));
j(N+(1:(N-1)))=1+(1:(N-1));
v(N+(1:(N-1)))= -1/(dx*dx);
%                       define the below-diagonal matrix entries
i((2*N-1)+(1:(N-1)))=1+(1:(N-1));
j((2*N-1)+(1:(N-1)))=(1:(N-1));
v((2*N-1)+(1:(N-1)))= -/(dx*dx);
%                       convert the entries into a sparse matrix format
A=sparse(i,j,v);
%                       define the right-hand side
F(1:N)=sin(dx*(1:N));
```

Program 6.1: `octave` code for solving (6.55): $-u''(x) = \sin(x)$ on $[0, \pi]$ with Dirichlet boundary conditions $u(0) = u(\pi) = 0$.

In `octave`, the solution to (6.54) can be achieved simply by writing

```
U=A\F ;
```

It is critical that one use vector constructs in `octave` to insure optimal performance. It executes much more rapidly, but the code is not shorter, more readible or less prone to error. See Exercise 6.5 for an example.

6.7.2 Reality Checks

The simplest way to determine whether an approximation to a differential equation is working or not is to attempt to solve a problem with a known solution. This is only one indication that all is working in one special set of conditions, not a proof of correctness. However, it is a very useful technique.

Using a graphical comparison of the approximation with the expected solution is the simplest way to find bugs, but a more demanding approach is often warrented. Using a *norm* to measure the difference between the approximation and the expected solution. This reduces a complex comparison to a single number. Moreover, there are extensive error estimates available which describe how various norms of the error should behave as a function of the maximum mesh size h.

For the approximation (u_n) (6.49) of the solution u of equation (6.49) (and ones similar to it) it can be shown [40] that

$$e_h := \max_{1\le n\le N} |u(x_n) - u_n| \le C_f h^2 \tag{6.56}$$

where C_f is a constant depending only on f. Thus one can do an experiment with a known u to see if the relationship (6.56) appears to hold as h is decreased. In particular, if the logarithm of e_h is plotted as a function of $\log h$, then the resulting plot should be linear, with a slope of two.

Consider the boundary value problem (6.55), that is, $-u'' = f$ on $[0,\pi]$ with $f(x) = \sin x$ and $u(0) = u(\pi) = 0$. Then the solution is $u(x) = \sin x$. The resulting error is depicted in Figure 6.1, where instead of the maximum error (6.56), the mean-squared (i.e., $L^2([0,\pi])$) error

$$\sqrt{h \sum_n (u(x_n) - u_n)^2}$$

has been plotted. The line $e_h = 0.1h^2$ has been added for clarity. Thus we see for $h \ge 10^{-4}$, the error diminishes quadratically. However, when the mesh size is much less than 10^{-4}, round-off error causes the accuracy to diminish, and the error even increases as the mesh size is further decreased.

6.7.3 Pitfall: Low Accuracy

All discretization methods can suffer from the effects of finite precision arithmetic. The easiest way to see this is that if h^2 is smaller than the smallest “machine ϵ” (defined to be the largest positive number whose addition to 1 in floating point returns 1) relative to the size of f and u, then (6.49) will effectively not distinguish between the real f and $f \equiv 0$.

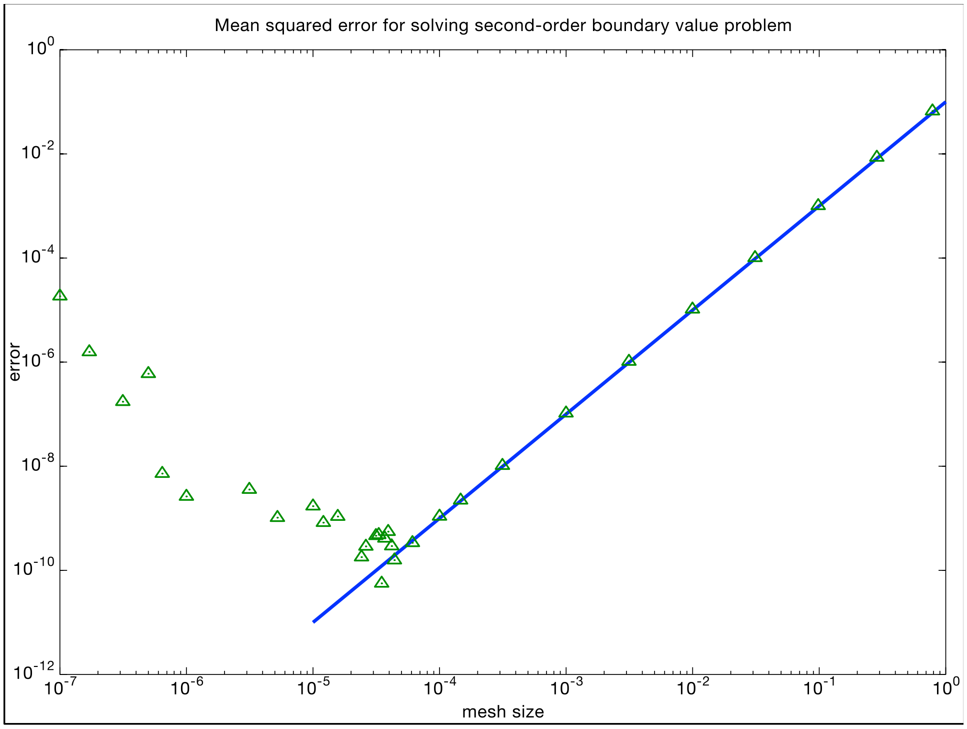

Figure 6.1: Error in $L^2([0,\pi])$ for the finite difference approximation of the boundary value problem for the differential equation $-u'' = \sin(x)$ on the interval $[0,\pi]$, with boundary conditions $u(0) = u(\pi) = 0$, as a function of the mesh size h. The solid line has a slope of 2 as a reference.

number of grid points	10	100	1000	10000
mesh size	2.86e-01	3.11e-02	3.14e-03	3.14e-04
condition number	4.84e+01	4.13e+03	4.06e+05	4.05e+07

Table 6.1: Condition number of Program 6.1 as a function of N. Recall that the mesh size $h = \pi/(N+1)$.

More subtle effects can also occur due to the increasing condition number of the linear system $\mathbf{A}$ as h tends to zero. We display the condition number for various mesh sizes in Table 6.1, and it appears that the condition number grows proportional to $N^2 \approx h^{-2}$. This effect can be amplified based on the method used to solve the linear system. We see in Figure 6.1 that the error decreases quadratically in the mesh size h up to a certain limit, and then it hits a wall. Accuracy behaves randomly and even decreases as the mesh size is further descreased. We see that the most accuracy we can achieve is closely related to the condition number multiplied by machine $\epsilon = 2.22 \times 10^{-16}$ in these compuations. Thus with $N = 10,000$, we cannot expect accuracy better than about 10^{-9}. Another way that finite-precision arithmetic can affect the accuracy of calculation is in the construction of the matrix $\mathbf{A}$ in more complex applications.

The simple way to avoid this difficulty is to avoid very small h values. The equivalent accuracy can be achieved by using a larger h and a more accurate discretization method. We postpone discussion of higher-order methods to later sections where Galerkin methods are discussed. These methods have the feature that it is often possible to define a family of methods of increasing degree of approximation. Indeed, the “spectral” family of methods relies entirely on increasing the degree of approximation on a fixed mesh.

6.7.4 Nonlinear Problems

Nonlinear differential equations can also be solved with often little more dificulty than linear ones. Consider the following problem:

$$\begin{gathered} -\frac{d^2u}{dx^2} + 4u + 6u^2 = f \text{ in } (0,\alpha) \\ u(0) = 0, \quad u'(\alpha) = 0. \end{gathered} \tag{6.57}$$

With $f = C$ (C = constant), this describes the profile of the radial component of fluid flow in a converging channel (a.k.a. Jeffrey-Hamel flow). In (6.57), differentiation is with respect to the polar angle ϕ and α is half of the angle (in radians) of convergence of the channel.

Given a solution u of (6.57), one can show that

$$\mathbf{u}(x,y) := \nu \, \frac{u(\operatorname{atan}(y/x))}{x^2+y^2} \, \mathbf{x}, \quad \mathbf{x} = (x,y) \in \Omega \tag{6.58}$$

solves the steady Navier-Stokes equations (14.1) with kinematic viscosity ν over a wedge domain Ω (cf. [112]).

The solution of (6.57) can be effected using a difference method for the differential operator as in Section 6.7.

$$-u_{n-1} + 2u_n - u_{n+1} + 4h^2 u_n + 6h^2 u_n^2 = h^2 C \tag{6.59}$$

where $u_n \approx u(x_n)$. Since this system of equations is nonlinear, we cannot solve it directly. A standard algorithm to use is Newton's method, which can be written as follows. First, we write the system of equations as $f(u_n) = 0$ where

$$f_n := -u_{n-1} + 2u_n - u_{n+1} + 4h^2 u_n + 6h^2 u_n^2 - h^2 C \tag{6.60}$$

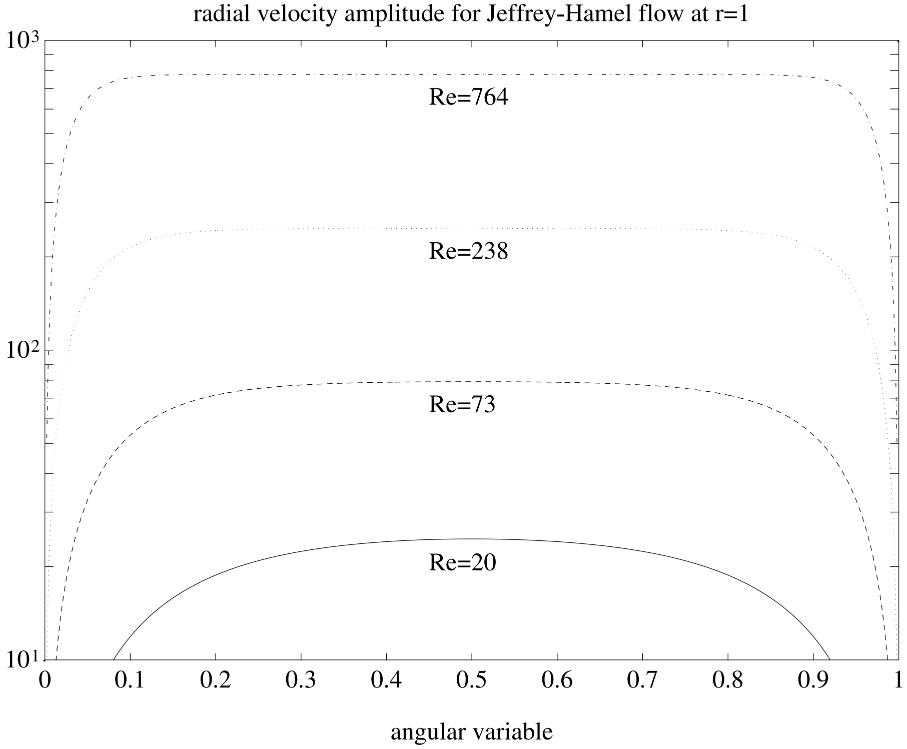

Figure 6.2: Solutions of the Jeffrey-Hamel equation with $C = 10^k$ for $k = 1, 2, 3, 4$.

Newton's iteration takes the form

$$u \leftarrow u - J_f^{-1} f(u) \tag{6.61}$$

where J_f denotes the Jacobian of the mapping f. This can be written in `octave` as follows.

Suppose that `A` is defined as in Section 6.7.1. Then `f` can be written

```
delta=((n+1)/alf)^2;
f = delta*A*uhf - 4*uhf - 6*ujh.*ujh + cvec;
```

where

```
cvec=C*ones(n,1);
```

The Jacobian `J` of `f` is

```
J = delta*A - 4*eye(n) - 12*diag(ujh,0);
```

Newton's method takes the following form in `octave`.

```
JA = delta*A - 4*eye(n);
ujh =- JA\cvec;
enorm = 1;
while (enorm >> .0000000000001)
    f = JA*ujh - 6*ujh.*ujh + cvec;
    J = JA - 12*diag(ujh,0);
    x = ujh - J\f;
    enorm = norm(ujh-x)/(norm(ujh)+norm(x));
    ujh=x;
end
```

6.8 Exercises

Exercise 6.1 *Use Taylor's theorem to derive (6.49).*

Exercise 6.2 *The solution to (6.1) with $f \equiv 1$ is $u(x) = x - \frac{x^2}{2}$. Use the method in Section 6.7 to approximate this problem. Implement these equations in* `octave` *on this test problem and determine the convergence rate. Where in the interval is the error largest?*

Exercise 6.3 *Replace (6.52) by*

$$u_{N+1} - u_N = 0 \tag{6.62}$$

Give the equation corresponding to (6.53) that results. How is it different? Implement these equations in `octave` *on the test problem in Section 6.7.2 and determine the convergence rate. Where in the interval is the error largest?*

Exercise 6.4 *Derive the finite difference approximation for the boundary value problem for the differential equation*

$$-u'' - u = f$$

with boundary conditions $u(0) = a$ and $u'(1) = b$. Write a `octave` *code for this and test it as in Section 6.7.2 for the exact solution $u(x) = \sin x$ with $f \equiv 0$, $g_0 = 0$ and $g_1 = \cos 1$.*

Exercise 6.5 *The matrix for the difference method in Section 6.7.1 can be implemented in* `octave` *as follows.*

```
    A=zeros(n);
for i=1:n
if i>1,A((i-1),i)=-1;  end
if i<n, A((i+1),i)=-1; end
A(i,i) = 2;
end
```

Compare this with the "vectorized" definition in Section 6.7.1 and determine the ratio of execution speed for the two methods of computing `A` *for* `n`*=100, 1000 and 10000. Can you identify any trends?*

Exercise 6.6 *Derive the matrix for the difference method Section 6.7.1 in the case of Dirichlet boundary conditions at both boundary points, that is, $u(0) = 0$ and $u(1) = 0$. Implement the matrix in* `octave` *with a "vectorized" definition.*

Exercise 6.7 *Consider the differential equation*

$$-\frac{d^2u}{dx^2} = f \text{ in } (0,1) \tag{6.63}$$

with Neumann boundary conditions at both boundary points, that is, $u'(0) = 0$ and $u'(1) = 0$. What function satisfies the differential equation with zero Neumann data? Show that solutions are unique only up to an additive constant, and they can exist only if the right-hand side f satisfies

$$\int_0^1 f(x)\,dx = 0. \tag{6.64}$$

Exercise 6.8 *Derive the matrix A for the difference method Section 6.7.1 in the case of Neumann boundary conditions at both boundary points, that is, $u'(0) = 0$ and $u'(1) = 0$. Implement the matrix in* `octave` *with a "vectorized" definition. Check to see if the matrix A is singular. What function satisfies the differential equation (see Exercise 6.7) with zero Neumann data? What is the associated null vector for the matrix A? Under what conditions does the equation $AX = F$ have a solution?*

Exercise 6.9 *Another method for solving nonlinear equations $f(u) = 0$ is the fixed-point iteration*

$$u \leftarrow u \pm \epsilon f(u) \tag{6.65}$$

for some parameter ϵ. Give an implementation of the Jeffrey-Hamel problem and compare it with Newton's method.

Exercise 6.10 *Consider the function v defined in (6.15). Show that it is in V (see Definition 6.6). Can one make sense of the linear form defined in (6.14) for this function?*

Exercise 6.11 *Give a separate, formal definition of each of the three different types of linear functionals given in the right-hand-side of (6.9). Can you say why the first and the last are bounded? For the middle one, consult (6.18).*

Exercise 6.12 *Implement the difference equations (6.28) in* `octave` *for a general mesh $0 = x_0 < x_1 < x_2 < \cdots < x_N = 1$. Use this method to approximate the problem in (6.1), with $f \equiv 1$ is $u(x) = x - \frac{x^2}{2}$. Implement these equations in* `octave` *on this test problem and determine the convergence rate as a function of $h := \max_i x_i - x_{i-1}$. Try different methods to generate random meshes. Where in the interval is the error largest?*

Exercise 6.13 *Derive the variational formulation for the boundary value problem for the differential equation*

$$-u'' - u = f$$

with boundary conditions $u(0) = g_0$ and $u'(1) = g_1$.

Exercise 6.14 *Consider the variational formulation in Exercise 6.13 for the boundary value problem for the differential equation*

$$-u'' - u = f$$

with boundary conditions $u(0) = 0$ and $u'(1) = 0$. Prove that this is coercive (hint: use (6.33)).

Exercise 6.15 *Derive the finite difference approximation corresponding to (6.28) for the boundary value problem for the differential equation*

$$-u'' - u = f$$

with boundary conditions $u(0) = g_0$ and $u'(1) = g_1$. Write a `octave` *code for this and test it as in Section 6.7.2 for the exact solution $u(x) = \sin x$ with $f \equiv 0$, $a = 0$ and $b = \cos 1$.*

Exercise 6.16 *Derive the matrix for the difference method (6.28) in the case of Dirichlet boundary conditions at both boundary points, that is, $u(0) = 0$ and $u(1) = 0$. Implement the matrix in* `octave` *with a "vectorized" definition.*

Exercise 6.17 *Consider the differential equation*

$$-u'' = f \text{ in } (0,1) \tag{6.66}$$

with Neumann boundary conditions at both boundary points, that is, $u'(0) = 0$ and $u'(1) = 0$ (see Exercise 6.7). Using the concept of coercivity, explain why this is not well-posed without some constraints on f and u.

Exercise 6.18 *Derive the matrix A for the difference method (6.28) in the case of Neumann boundary conditions at both boundary points, that is, $u'(0) = 0$ and $u'(1) = 0$. Implement the matrix in* `octave` *with a "vectorized" definition. Check to see if the matrix A is singular. What function satisfies the differential equation (see Exercise 6.7) with zero Neumann data? What is the associated null vector for the matrix A? Under what conditions does the equation $AX = F$ have a solution?*

Exercise 6.19 *Show that Newton's method for the system (6.43) (or equivalently (6.44)) can be characterized by the variational formulation (6.48).*

Exercise 6.20 *Show that the inhomogeneous Dirichlet boundary value problem (6.9), with V as in (6.6) and with $f \equiv 0$ and $g_1 = 0$ can be written in the form (6.10) with $u_0 := g_0(1-x)$ and*

$$F(v) = -g_0 a(1-x, v) = g_0 \int_0^1 v'\,dx = g_0 v(1) \quad \forall v \in V. \tag{6.67}$$

Investigate the choice $u_0 \equiv b_0$ (a constant function) and show that this leads to $F \equiv 0$. Why do these different variational formulations give equivalent results?

Exercise 6.21 *Consider the variational form $a(\cdot,\cdot)$ in (6.5) and define*

$$\tilde{a}(u, v) := a(u, v) + \gamma u(1)v(1). \tag{6.68}$$

Consider the variational problem (6.7) with $g_1 = 0$ and V as defined in (6.6). Show that this corresponds to having a boundary condition at 1 of Robin/Cauchy type: $u'(1) + \gamma u(1) = 0$.

Exercise 6.22 *Suppose α and β are smooth functions. Consider the variational form*

$$a(u, v) := \int_0^1 u'v' + \alpha u v' + (\beta - \alpha')uv\,dx \tag{6.69}$$

and the variational problem (6.7) with $g_1 = 0$ and V as defined in (6.6). Determine the corresponding differential equation and boundary conditions that the solution u must satisfy if it is known to be smooth.

Exercise 6.23 *Suppose α and β are smooth functions. Consider the variational form*

$$a(u, v) := \int_0^1 u'v' + \alpha u'v + \beta uv\,dx \tag{6.70}$$

and the variational problem (6.7) with $g_1 = 0$ and V as defined in (6.6). Determine the corresponding differential equation and boundary conditions that the solution u must satisfy if it is known to be smooth.

Exercise 6.24 *Examine the equations generated by the finite element method using piecewise linear functions for the problem discussed in Section 6.3.2. What are the equations that arise due to the boundary conditions? How does this compare with the finite difference approach?*

Chapter 7

The Heat Equation

Heat can be exchanged between two different bodies by diffusion, convection or radiation. The **heat equation** describes the diffusion of thermal energy in a medium [80]. In its simplest, one-dimensional form, it may be written

$$\begin{gathered}\frac{\partial u}{\partial t}(x,t)-\frac{\partial^2 u}{\partial x^2}(x,t)=f(x,t)\quad\forall x\in[0,1],\ t>0\\ u(x,0)=u_0(x)\quad\forall x\in[0,1]\end{gathered}\tag{7.1}$$

where $u(x,t)$ denotes the temperature of the medium at any given point x and time t. This equation is also known as the **diffusion equation**.

A simple example of heat diffusion in one spatial dimension is the transfer of heat across a window. The variable x denotes the distance from one face of the window pane to the other, in the direction perpendicular to the plane of the window. Near the outer edges of the window, three dimensional effects would be evident, but in the middle of the window, equation (7.1) would accurately describe the evolution of the temperature u inside the window.

The function f is included for completeness, but in many cases such a body source of heat would be zero. These equations must be supplemented by boundary contitions similar to the ones considered in Chapter 6. They could be of purely Dirichlet (or essential) type, viz.

$$u(0,t)=g_0(t),\quad u(1,t)=g_1(t)\quad\forall t>0,\tag{7.2}$$

or of purely Neumann (or natural) type, viz.

$$\frac{\partial u}{\partial x}(0,t)=g_0(t),\quad\frac{\partial u}{\partial x}(1,t)=g_1(t)\quad\forall t>0,\tag{7.3}$$

or a combination of the two:

$$u(0,t)=g_0(t),\quad\frac{\partial u}{\partial x}(1,t)=g_1(t)\quad\forall t>0.\tag{7.4}$$

Here g_i, $i=0,1$, are given functions of t. It is interesting to note that the pure Neumann condition (7.3) for the heat equation (7.1) does not suffer the same limitations on the data, or nonuniqueness of solutions, that the steady state counterpart does. However, there are compatibility conditions required to obtain smooth solutions.

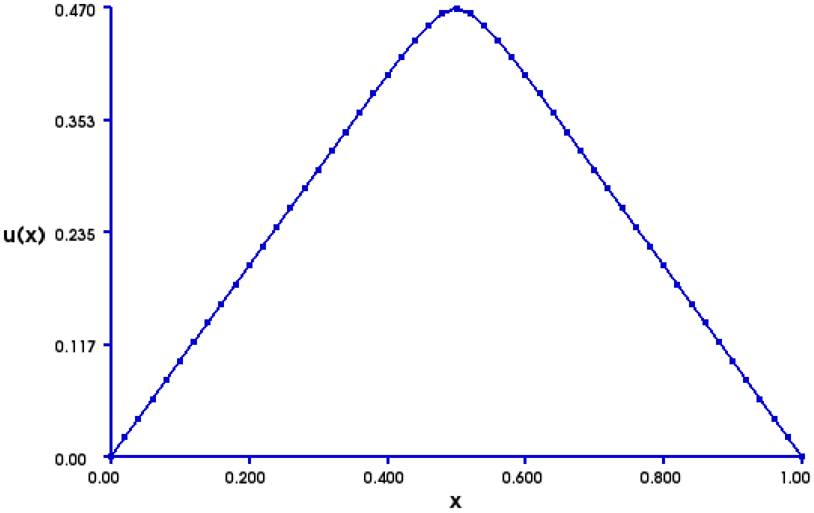

Figure 7.1: Solution of (7.1) with initial data (7.5) at time $t = 0.001$. Computed with piecewise linears with 50 mesh points (uniform mesh).

7.1 Basic behavior: smoothing

The main characteristic of the heat equation is that it smooths any roughness in the initial data. For example, in Figure 7.1 we show the solution at time $t = 0.001$ for the case

$$u_0(x) = \tfrac{1}{2} - |x - \tfrac{1}{2}|. \tag{7.5}$$

We see that the discontinuity of the derivative of u_0 at $x = \frac{1}{2}$ is instantly smoothed. This has a corollary for the backwards heat equation (Section 7.6) that we will explore subsequently. One type of nonsmooth behavior stems from a mismatch between boundary data and initial data. This is governed by compatibility conditions.

The code to generate Figure 7.1 is given in Program 7.1.

7.2 Compatibility Conditions

There is a **compatibility condition** for the boundary and initial data for the heat equation in order to have a smooth solution. This can be derived easily from the observation that the values of u on the spatial boundary have been specified twice at $t = 0$. Consider the case (7.4) of combined Dirichlet and Neumann boundary conditions. The first set of compatibility conditions is

$$u_0(0) = u(0,0) = g_0(0) \quad \text{and} \quad u_0'(1) = u_x(1,0) = g_1(0). \tag{7.6}$$

These are obtained by matching the two ways of specifying the solution at the boundary points $(x,t) = (0,0)$ and $(x,t) = (1,0)$. In the case of pure Dirichlet conditions (7.2) the compatibility conditions become

$$u_0(0) = u(0,0) = g_0(0) \quad \text{and} \quad u_0(1) = u(1,0) = g_1(0). \tag{7.7}$$

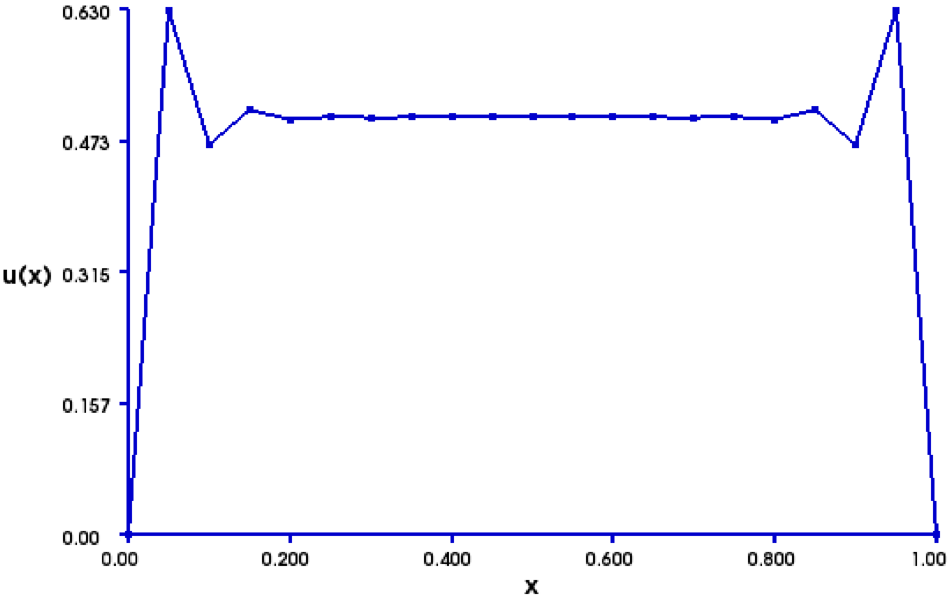

Figure 7.2: Heat equation with incompatible data after one time step with $\Delta t = 10^{-5}$; degree = 1, 20 mesh intervals. Initial values $u_0 = 0.5$.

In the case of pure Neumann conditions (7.3) the compatibility conditions become

$$u_0'(0) = u_x(0,0) = g_0(0) \quad \text{and} \quad u_0'(1) = u_x(1,0) = g_1(0). \tag{7.8}$$

The conditions involving derivatives (coming from the Neumann boundary conditions) are higher-order compatibility conditions than those for the function values (coming from the Dirichlet boundary conditions). They affect the boundedness of higher-order derivatives.

There are more conditions than this for real smoothness, since u satisfies a differential equation. In fact, for an abritrary order of smoothness, there are infinitely many such compatibility conditions, the first of these being one of the above, (7.6), (7.7) or (7.8), as appropriate depending on the boundary conditions in force. Again, consider the case (7.4) of combined Dirichlet and Neumann boundary conditions to start with. The second set of conditions arrises by using the differential equation $u_{xx} = u_t$ to trade spatial derivatives for temporal ones, then applying this at $t = 0$ and $x = 0$:

$$u_0''(0) = u_{xx}(0,0) = u_t(0,0) = g_0'(0) \quad \text{and} \quad u_0'''(1) = u_{xxx}(1,0) = u_{xt}(1,0) = g_1'(0). \tag{7.9}$$

We leave as exercises (Exercise 7.4 and Exercise 7.5) to give the corresponding second set of compatibility conditions for the pure Dirichlet and Neumann boundary conditions.

If these compatibilities are not satisfied by the data (or by the approximation scheme), wild oscillations (at least in some derivative, if not the solution itself) will result near $t = 0$ and $x = 0, 1$, as shown in Figure 7.2. Using higher resolution can eliminate the oscillations in

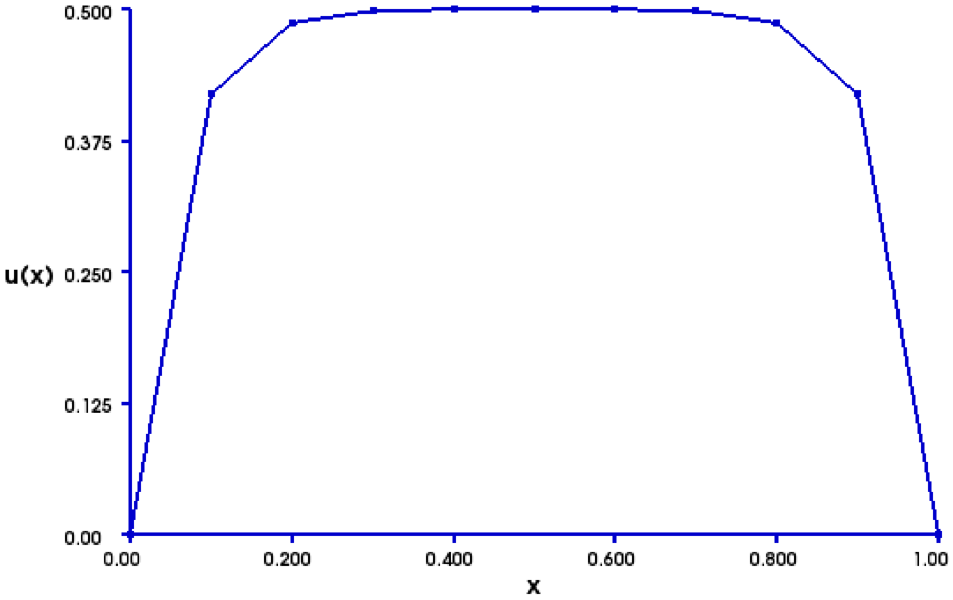

Figure 7.3: Heat equation with incompatible data after one time step with $\Delta t = 10^{-5}$; degree $= 2$, 10 mesh intervals. Initial values $u_0 = 0.5$.

some cases, as shown in Figure 7.3. In nonlinear problems, this can cause completely wrong results to occur.

The compatibility conditions, e.g. (7.6) and (7.9), do not have to be satisfied for the heat equation (7.1) for it to be well-posed in the usual sense. There is a unique solution in any case, but the physical model may be incorrect as a result if it is supposed to have a smooth solution. Compatibility conditions are a subtle form of constraint on model quality. In many problems they can be described in terms of local differential-algebraic constraints as in (7.6) and (7.9). However, in Section 14.3.2 we will see that such compatibility conditions can lead to global constraints that may be hard to verify or satisfy in practice.

7.3 Variational form of the heat equation

It is possible to derive a vatiational formulation involving integration over both x and t, but it is more common to use a variational formulation based on x alone. Recalling the notation of Chapter 6, we seek a function $\tilde{u}(t)$ of time with values in V such that $\tilde{u}(0) = u_0$

$$(\tilde{u}'(t), v)_{L^2(\Omega)} + a(\tilde{u}(t), v) = F(v) \quad \forall v \in V,\ t \geq 0, \tag{7.10}$$

where $\Omega = [0, 1]$ and $a(w, v) = \int_0^1 w'(x)v'(x)\,dx$. Since it is a bit awkward to work with a function of one variable (t) which is a function of another (x), we often write (7.10) in terms of $u(x,t)=\tilde{u}(t)(x)$. Using subscript notation for partial derivatives, it becomes

$$(u_t(\cdot,t),v)_{L^2(\Omega)}+a(u(\cdot,t),v)=F(v)\quad\forall v\in V. \tag{7.11}$$

for all t. If we remember the dependence on t, we can write this as

$$(u_t,v)_{L^2(\Omega)}+a(u,v)=F(v)\quad\forall v\in V. \tag{7.12}$$

A stability estimate follows immediately from the variational formulation. For simplicity, suppose that the right-hand-side form $F\equiv 0$ and that the boundary data vanishes as well (i.e., only the initial data is non-zero). Using $v=u$ (at any fixed t, i.e., $v=u(\cdot,t)$) in (7.12), we find

$$\frac{1}{2}\frac{\partial}{\partial t}\|u\|^2_{L^2(\Omega)}=(u_t,u)_{L^2(\Omega)}=-a(u,u)\le 0\quad\forall t\ge 0, \tag{7.13}$$

where Ω denotes the spatial interval $[0,1]$. From this, it follows by integrating in time that

$$\|u(\cdot,t)\|_{L^2(\Omega)}\le\|u(\cdot,0)\|_{L^2(\Omega)}=\|u_0\|_{L^2(\Omega)}\quad\forall t\ge 0. \tag{7.14}$$

This result is independent of any compatibility conditions. However, all it says is that the mean-square of the temperature u remains bounded by its initial value. If F is nonzero but bounded on V, i.e.,

$$|F(v)|\le\|F\|_{H^{-1}(\Omega)}\|v\|_{H^1(\Omega)}\quad\forall v\in V, \tag{7.15}$$

then we retain a bound on $\|u(\cdot,t)\|_{L^2(\Omega)}$:

$$\frac{1}{2}\frac{\partial}{\partial t}\|u\|^2_{L^2(\Omega)}=(u_t,u)_{L^2(\Omega)}=F(u)-a(u,u)\le\|F\|_{H^{-1}(\Omega)}\|v\|_{H^1(\Omega)}-a(u,u). \tag{7.16}$$

The form $a(\cdot,\cdot)$ always satisfies at least a weak type of coercivity of the form

$$\|v\|^2_{H^1(\Omega)}\le\gamma_1 a(v,v)+\gamma_2\|v\|^2_{L^2(\Omega)}\quad\forall v\in V, \tag{7.17}$$

known as **Gårding's inequality**. For example, this holds for the pure Neumann problem (7.3) with $V=H^1(\Omega)$ whereas the stronger form of coercivity (6.32) does not in this case. Applying (7.17) in (7.16) gives

$$\frac{\partial}{\partial t}\|u\|^2_{L^2(\Omega)}\le 2\|F\|_{H^{-1}(\Omega)}\|u\|_{H^1(\Omega)}-\frac{2}{\gamma_1}\|u\|^2_{H^1(\Omega)}+\frac{2\gamma_2}{\gamma_1}\|u\|^2_{L^2(\Omega)}. \tag{7.18}$$

Using the arithmetic-geometric mean inequality in the form

$$2rs\le\delta r^2+\frac{1}{\delta}s^2 \tag{7.19}$$

which holds for any $\delta>0$ and any real numbers r and s, we find

$$\frac{\partial}{\partial t}\|u\|^2_{L^2(\Omega)}\le\frac{\gamma_1}{2}\|F\|^2_{H^{-1}(\Omega)}+\frac{2\gamma_2}{\gamma_1}\|u\|^2_{L^2(\Omega)} \tag{7.20}$$

Gronwall's Lemma [170] implies

$$\|u(\cdot,t)\|_{L^2(\Omega)} \le \|u_0\|_{L^2(\Omega)} + e^{t(\gamma_2/\gamma_1)}\|F\|_{H^{-1}(\Omega)} \quad \forall t \ge 0. \tag{7.21}$$

Another stability result can be derived by using $v = u_t$ (assuming $F \equiv 0$ and the boundary data are zero) in (7.12), to find

$$\|u_t\|^2_{L^2(\Omega)} = -a(u,u_t) = -\tfrac{1}{2}\frac{\partial}{\partial t}a(u,u). \tag{7.22}$$

From this, it follows that

$$\frac{\partial}{\partial t}a(u,u) = -2\|u_t\|^2_{L^2(\Omega)} \le 0 \quad \forall t \ge 0. \tag{7.23}$$

Again integrating in time and using (7.14), we see that

$$\|u(\cdot,t)\|_{H^1(\Omega)} \le \|u(\cdot,0)\|_{H^1(\Omega)} = \|u_0\|_{H^1(\Omega)} \quad \forall t \ge 0. \tag{7.24}$$

This result is again independent of any compatibility conditions, and it says is that the mean-square of the gradient of the temperature u also remains bounded by its initial value. Of course, this presupposes that $u_0 \in V$, and this may not hold. Moreover, if the data F is not zero, this result will not hold. In particular, if the compatibility condition (7.6) does not hold, then $u_0 \notin V$ and $\|u(\cdot,t)\|_{H^1(\Omega)}$ will not remain bounded as $t \to 0$.

7.4 Discretiztion

The simplest discretization for the heat equation uses a spatial discretization method for ordinary differential equation in Section 6.7, for that part of the problem and a finite difference method for the temporal part. This technique of decomposing the problem into two parts is an effective technique to generate a numerical scheme, and it allows us to **reuse existing software** already developed for the o.d.e. problem. Many time dependent problems can be treated in the same manner. This technique goes by many names:

- (time) **splitting** since the time and space parts are separated and treated by independent methods
- the **method of lines** since the problem is solved on a sequence of lines (copies of the spatial domain), one for each time step.

7.4.1 Explicit Euler Time Discretization

The simplest time discretization method for the heat equation uses the forward (or explicit) Euler difference method. It takes the form

$$\begin{aligned} u^{n+1}(x) = u^n(x) + \Delta t\frac{\partial^2 u^n}{\partial x^2}(x,t) \quad \forall x \in [0,1], \\ u^0(x) = u_0(x) \quad \forall x \in [0,1] \\ u^n(0) = g_0(t) \quad \text{and} \quad u^n(1) = g_1(t) \quad \forall n > 0 \end{aligned} \tag{7.25}$$

where $u^n(x)$ denotes an approximation to $u(x, n\Delta t)$. Applying the finite difference or finite element approximation (6.50) to (7.25) yields a simple algorithm. The difficulty with this simple algorithm is that it is *unstable* unless `dt` is sufficiently small.

7.4.2 Implicit Euler Time Discretization

The simplest implicit time discretization method for the heat equation uses the backward (or implicit) Euler difference method. It takes the form

$$\begin{aligned} u^{n+1}(x) = u^n(x) + \Delta t \frac{\partial^2 u^{n+1}}{\partial x^2}(x,t) \quad \forall x \in [0,1], \\ u^0(x) = u_0(x) \quad \forall x \in [0,1] \\ u^n(0) = g_0(t) \quad \text{and} \quad u^n(1) = g_1(t) \quad \forall n > 0 \end{aligned} \tag{7.26}$$

where $u^n(x)$ again denotes an approximation to $u(x, n\Delta t)$. Applying the finite difference or finite element approximation (6.50) to (7.26) yields now a system of equations to be solved at each time step. This algorithm is *stable* for all `dt`, but now we have to solve a system of equations instead of just multiplying by a matrix. Note however that the system to be solved is just the same as in the ODE boundary value problems studied earlier, so the same family of techniques can be used.

7.4.3 Variational form of the time discretization

The explicit Euler time stepping method can be written in variational form as

$$(u^{n+1}, v)_{L^2(\Omega)} = (u^n, v)_{L^2(\Omega)} + \Delta t \left(F(v) - a(u^n, v)\right) \quad \forall v \in V. \tag{7.27}$$

Solving for u^{n+1} requires inverting the mass matrix (6.24).

The implicit Euler time stepping method can be written in variational form as

$$(u^{n+1}, v)_{L^2(\Omega)} + \Delta t \, a(u^{n+1}, v) = (u^n, v)_{L^2(\Omega)} + \Delta t \, F(v) \quad \forall v \in V. \tag{7.28}$$

Solving for u^{n+1} requires inverting linear combination of the stiffness matrix (6.21) and the mass matrix (6.24). This is now in the familiar form: find $u^{n+1} \in V$ such that

$$a_{\Delta t}(u^{n+1}, v) = F^n_{\Delta t}(v) \quad \forall v \in V,$$

where

$$a_{\Delta t}(v, w) = \int_\Omega vw + \Delta t v' w' \, d\mathbf{x}, \quad F^n_{\Delta t}(v) = (u^n, v)_{L^2(\Omega)} + \Delta t F(v) \quad \forall v, w \in V.$$

k	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	1	-1						
2	3/2	-2	1/2					
3	11/6	-3	3/2	$-1/3$				
4	25/12	-4	6/2	$-4/3$	1/4			
5	137/60	-5	10/2	$-10/3$	5/4	$-1/5$		
6	49/20	-6	15/2	$-20/3$	15/4	$-6/5$	1/6	
7	363/140	-7	21/2	$-35/3$	35/4	$-21/5$	7/6	$-1/7$

Table 7.1: Coefficients of the BDF schemes of degree k.

7.4.4 Mass lumping

It is disconcerting that the explicit Euler time-stepping scheme leads to a system of equations (involving the mass matrix) that has to be inverted at each time step. This can be avoided in many cases by replacing the exact integration in expressions like $(u^n, v)_{L^2(\Omega)}$ by appropriate quadrature. For example, with piecewise linear approximation in one spatial dimension, we could use trapezoidal rule for evaluating the expression in (6.24). In this case, instead of the matrix M, we get the identity matrix, and the algorithm (7.27) gets transformed to

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \left(\mathbf{F} - \mathbf{A}\mathbf{U}^n \right), \tag{7.29}$$

where we are using the vector notation preceeding (6.22).

7.5 Backwards differentiation formulæ

A popular way to achieve increased accuracy in time-dependent problems is to use a **backwards differentiation formula** (BDF)

$$\frac{du}{dt}(t_n) \approx \frac{1}{\Delta t} \sum_{i=0}^{k} a_n u_{n-i}, \tag{7.30}$$

where the coefficients $\{a_i \,:\, i = 0, \ldots k\}$ are given in Table 7.1. The BDF for $k = 1$ is the same as implicit Euler. The BDF formulæ satisfy [159]

$$\sum_{i=0}^{k} a_i u_{n-i} = \sum_{j=1}^{k} \frac{(-1)^j}{j} \Delta^j u_n, \tag{7.31}$$

where Δu_n is the sequence whose n-th entry is $u_n - u_{n-1}$. The higher powers are defined by induction: $\Delta^{j+1} u_n = \Delta(\Delta^j u_n)$. For example, $\Delta^2 u_n = u_n - 2u_{n-1} + u_{n-2}$, and in general Δ^j has coefficients given from Pascal's triangle. We thus see that $a_0 \neq 0$ for all $k \geq 1$; $a_0 = \sum_{i=1}^{k} 1/i$. Similarly, $a_1 = -k$, and for $j \geq 2$, ja_j is an integer conforming to Pascal's triangle.

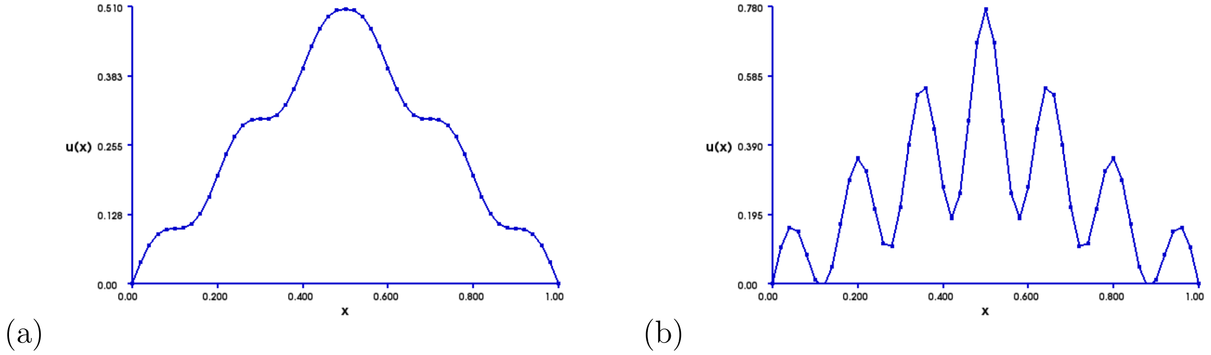

Figure 7.4: Solution of (7.32) with initial data (7.5) at time $t = 0.001$, computed with piecewise linears with 50 mesh points (uniform mesh). (a) One time step with $\Delta t = 0.001$. (a) Two time steps with $\Delta t = 0.0005$.

Given this simple definition of the general case of BDF, it is hard to imagine what could go wrong regarding stability. Unfortunately, the BDF method of order $k = 7$ is unconditionally unstable and hence cannot be used. We leave as exercises to explore the use of the BDF schemes.

7.6 The backwards heat equation

The heat equation is reversible with respect to time, in the sense that if we let time run backwards we get an equation that takes the final values to the initial values. More precisely, let $u(x, t)$ be the solution to (7.1) for $0 \leq t \leq T$. Let $v(x, t) := u(x, T - t)$. Then v solves the backwards heat equation

$$\begin{aligned} \frac{\partial v}{\partial t}(x,t) + \frac{\partial^2 v}{\partial x^2}(x,t) = 0 \quad \forall x \in [0,1],\ t > 0 \\ v(x,0) = v_0(x) = u(x,T) \quad \forall x \in [0,1] \\ v(0,t) = g_0(T-t), \quad v(1,t) = g_1(T-t) \quad \forall t > 0 \end{aligned} \tag{7.32}$$

and $v(x, T)$ will be the same as the initial data u_0 for (7.1).

Although (7.32) has a well-defined solution in many cases, it is not well-posed in the usual sense. It only has a solution starting from solutions of the heat equation. Moreover, such solutions may exist only for a short time, and then blow up. Thus great care must be used in attempting to solve the backwards heat equation.

One reason for interest in the backwards heat equation is in information recovery. If a process blurs information via a diffusion process, then running the backwards heat equation can potentially deblur the information. Thus this approach has been considered in image processing [44].

An example of the difficulty in reversing the heat equation is shown in Figure 7.4. What is depicted is an attempt to solve (7.32) with initial data (7.5) at time $t = 0.001$. What

formula	classification	example	formula
$x^2 + y^2 = 1$	elliptic	Laplace	$u_{,xx} + u_{,yy}$
$x^2 - y^2 = 1$	hyperbolic	wave	$u_{,xx} - u_{,yy}$
$y = x^2$	parabolic	diffusion/heat	$u_{,x} - u_{,yy}$

Table 7.2: Classification of linear PDEs. The column marked "formula" gives a typical formula for a conic section of whose name gives rise to the PDE classification name.

we see in Figure 7.4(a) is the result of using a uniform mesh in space with 50 points and piecewise linear approximation, using $\Delta t = 0.001$. What we see looks plausible, but when we check this by cutting the time step in half, and doubling the number of time steps, we see in Figure 7.4(b) that we get something radically different, with very many oscillations. If we continue this, by cutting the time step in half and doubling the number of time steps, we find even wilder oscillations.

What is going on? Remember that the heat equation always smooths the initial data as we move forward in time. Thus when we run time backwards, we must get a solution that is rougher than the initial data. Therefore there could not be a smooth solution for the backwards heat starting with the nonsmooth initial data (7.5). Thus the progression indicated in Figure 7.4 from panel (a) to panel (b) suggests an attempt to generate some very singular object. Recall that the the time in both panels is the same, and (b) represents using better time resolution. We leave as Exercise 7.15 to explore changing the spatial resolution by increasing both the number of mesh points and the polynomial degree of the finite element approximation.

7.7 Classification of PDEs

There are two major classifications of PDEs, one for linear PDEs and one for nonlinearities. The linear classification is simple: elliptic, parabolic, and hyperbolic. This is based on a simple algebraic dichotomy for second-order differential operators D in two dimensions.

But there are other equations of higher order that do not fit this classification, such as the dispersive Airy equation $u_t + u_{xxx} = 0$. On the other hand, using a more sophisticated classification [4], it is possible to see the Stokes equations as an elliptic system.

7.8 Exercises

Exercise 7.1 *Verify that the Gaussian* $u(x,t) := \frac{1}{(t+t_o)^{1/2}} e^{-x^2/(t+t_o)}$ *is an exact solution to (7.1). Use this to test the accuracy of a numerical code for some* $t_o > 0$*. What happens if you take* $t_o \to 0$*?*

Exercise 7.2 *Generalize the heat equation (7.1) to two spatial dimensions in which the spatial operator is the Laplace operator. Give a "method of lines" discretization for it. Test the code on the two-dimensional version of the exact solution in Exercise 7.1.*

Exercise 7.3 *Consider the generalized heat equation in two spatial dimensions (Exercise 7.2). Determine the first two compatibility conditions on the initial and boundary data to insure smooth solutions (see (7.6) and (7.6)). Give a demonstration of what happens if these are violated.*

Exercise 7.4 *Give the second set of compatibility conditions (7.9) in the case of pure Dirichlet conditions (7.2).*

Exercise 7.5 *Give the second set of compatibility conditions (7.9) in the case of pure Neumann boundary conditions (7.3).*

Exercise 7.6 *Derive the third set of compatibility conditions in the case of pure Dirichlet conditions (7.2), pure Neumann boundary conditions (7.3), and mixed Dirichlet and Neumann boundary conditions (7.4).*

Exercise 7.7 *Show that the inhomogeneous initial and boundary value problem (7.1)–(7.4) for the heat equation can be written in the form (7.12) with*

$$F(v) = (f, v) + g_1(t)v(1) - g_0(t)a(1 - x, v) \quad \forall v \in V. \tag{7.33}$$

Exercise 7.8 *Examine the numerical solution of (7.25) with incompatible data, e.g., where $u_0(x) = 1 - 2x$ and $g_0(t) = g_1(t) \equiv 0$. How does the error depend on x and t? Does it decrease as t increases? What is the rate of convergence in the L^2 norm for the solution $\|u(\cdot, t)\|_{L^2(\Omega)}$.*

Exercise 7.9 *Examine the stability limit of the explicit Euler scheme.*

Exercise 7.10 *Examine the stability limit of the implicit Euler scheme.*

Exercise 7.11 *Examine the stability limit of the second-order backwards difference scheme.*

Exercise 7.12 *Try solving the backwards heat equation (7.32) with the Gaussian $u(x,t) := \frac{1}{(t_0)^{1/2}} e^{-x^2/(t_0)}$ as initial data.*

Exercise 7.13 *Give a variational formulation of the problem*

$$\begin{aligned} \frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) + \beta \frac{\partial u}{\partial x}(x,t) + \gamma u(x,t) = f \quad \forall x \in [0,1],\ t > 0 \\ u(x,0) = u_0(x) \quad \forall x \in [0,1] \end{aligned} \tag{7.34}$$

with a simple implicit Euler time-stepping. Assume that β and γ are known functions of x.

Exercise 7.14 *Give a variational formulation of the problem*

$$\begin{aligned} \frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) + n(u(x,t)) = f \quad \forall x \in [0,1],\ t > 0 \\ u(x,0) = u_0(x) \quad \forall x \in [0,1] \end{aligned} \tag{7.35}$$

with a simple implicit Euler time-stepping, where n is the nonlinear function $n(u) = u^2$. Describe Newton's method to solve the nonlinear system at each time step.

Exercise 7.15 *Solve (7.32) with initial data (7.5) at time $t = 0.001$ using different spatial resolution by increasing both the number of mesh points and the polynomial degree of the finite element approximation. Summarize what you learn.*

```
from dolfin import *
import sys, math
import time

dt=float(sys.argv[1])
deg=int(sys.argv[2])
mno=int(sys.argv[3])

# Create mesh and define function space
mesh = UnitIntervalMesh(mno)
V = FunctionSpace(mesh, "Lagrange", deg)

# Define Dirichlet boundary (x = 0 or x = 1)
def boundary(x):
    return x[0] < DOLFIN_EPS or x[0] > 1.0 - DOLFIN_EPS

# Define boundary condition
g = Expression("0.5-std::abs(x[0]-0.5)")
u0 = Expression("0")

bc = DirichletBC(V, u0, boundary)

# Define variational problem
u = TrialFunction(V)
uold = Function(V)
v = TestFunction(V)
a = dt*inner(grad(u), grad(v))*dx + u*v*dx
F = uold*v*dx
u = Function(V)

uold.interpolate(g)
u.assign(uold)

# Compute one time step
solve(a == F, u, bc)

uold.assign(u)
plot(u, interactive=True)
```

Program 7.1: Code to implement the problem (7.1).

Chapter 8

Advection

Many models balance advection and diffusion. In the simplest case, this can be represented as follows. We use notation and concepts from Chapter 2 without reference. The basic advection-diffusion equation in a domain Ω is

$$-\epsilon\Delta u + \boldsymbol{\beta}\cdot\nabla u = f \text{ in } \Omega \tag{8.1}$$

where $\boldsymbol{\beta}$ is a vector-valued function indicating the advection direction. For simplicity, we again assume that we have boundary conditions

$$\begin{aligned} u = g \text{ on } \Gamma \subset \partial\Omega \qquad &\text{(Dirichlet)} \\ \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega\backslash\Gamma \qquad &\text{(Neumann)} \end{aligned} \tag{8.2}$$

where $\frac{\partial u}{\partial n}$ denotes the derivative of u in the direction normal to the boundary, $\partial\Omega$. This may or may not be a good model, but it is consistent with a behavior where the solution is not changing much near the out-flow boundary.

8.1 Posing Boundary Conditions

In an advection-diffusion model, the quantity u is being advected in the direction of $\boldsymbol{\beta}$. Thus there is often a sense of in-flow and out-flow parts of the boundary. Suppose that Γ represents the in-flow part. We will see that this is characterized by $\boldsymbol{\beta}\cdot\mathbf{n} < 0$ on Γ. Suppose that we specify the quantity u on Γ via

$$u = g_D \text{ on } \Gamma \subset \partial\Omega \tag{8.3}$$

The boundary condtion on the out-flow boundary requires some modeling. We will see what happens if we try to specify something inappropriate there subsequently, but for now let us describe the typical approach.

Since we usually do not know what u will look like near the out-flow part of the domain, it would be best to do something neutral. Suppose that we do nothing, in the sense that we take the approach that we just use the variational space V defined in (2.5) and then use (3.13) to define u, with $g_N = 0$ since we do not know how to specify g_N. This corresponds to the boundary condition

$$\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega\backslash\Gamma. \tag{8.4}$$

8.2 Variational Formulation of advection-diffusion

We again use the variational space V defined in (2.5). As before, using the three-step recipe, we define

$$\begin{aligned} a(u,v) &= \int_\Omega \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})\, d\mathbf{x} \\ b(u,v) &= \int_\Omega \big(\boldsymbol{\beta}(\mathbf{x}) \cdot \nabla u(\mathbf{x})\big) v(\mathbf{x})\, d\mathbf{x}. \end{aligned} \tag{8.5}$$

Here we see for the first time an alternative formulation: we could have integrated by parts in the advection term. To see what this means, we again invoke the divergence theorem (2.7) to yield

$$\oint_{\partial\Omega} u\, v\, \boldsymbol{\beta} \cdot \mathbf{n}\, ds = \int_\Omega \nabla\cdot \big(u\, v\, \boldsymbol{\beta}\big)\, d\mathbf{x} = \int_\Omega u\boldsymbol{\beta} \cdot \nabla v + v\boldsymbol{\beta} \cdot \nabla u + u\, v\, \nabla\cdot \boldsymbol{\beta}\, d\mathbf{x}. \tag{8.6}$$

In particular,

$$b(u,v) + b(v,u) = \oint_{\partial\Omega} u\, v\, \boldsymbol{\beta} \cdot \mathbf{n}\, ds - \int_\Omega u\, v\, \nabla\cdot \boldsymbol{\beta}\, d\mathbf{x}. \tag{8.7}$$

Before deriving an alternate formulation, let us first prove coercivity of the bilinear form $a_\beta(u,v) = a(u,v) + b(u,v)$.

8.2.1 Coercivity of the Variational Problem

Since we already know that $a(\cdot,\cdot)$ is coercive on

$$V = \left\{v \in H^1(\Omega) \ :\ v = 0 \text{ on } \Gamma\right\},$$

it suffices to determine conditions under which $b(v,v) \geq 0$ for all $v \in V$. From (8.7), we have

$$2b(v,v) = \oint_{\partial\Omega} v^2 \boldsymbol{\beta} \cdot \mathbf{n}\, ds - \int_\Omega v^2 \nabla\cdot \boldsymbol{\beta}\, d\mathbf{x}. \tag{8.8}$$

Define

$$\Gamma_0 = \{x \in \partial\Omega \ :\ \boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{n} = 0\}, \qquad \Gamma_\pm = \{x \in \partial\Omega \ :\ \pm\boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{n} > 0\}. \tag{8.9}$$

An important special case is when $\boldsymbol{\beta}$ is **incompressible**, meaning $\nabla\cdot \boldsymbol{\beta} = 0$. In this case, (8.8) simplifies to

$$2b(v,v) = \oint_{\Gamma_- \cup \Gamma_+} v^2 \boldsymbol{\beta} \cdot \mathbf{n}\, ds \geq \oint_{\Gamma_-} v^2 \boldsymbol{\beta} \cdot \mathbf{n}\, ds, \tag{8.10}$$

degree	mesh number	ϵ	$\epsilon^{-1}\|u_\epsilon - u_0\|_{L^2(\Omega)}$
4	8	1.0e+00	0.27270
4	8	1.0e-01	0.71315
4	8	1.0e-02	0.86153
4	8	1.0e-03	0.87976
4	8	1.0e-04	0.88172
4	8	1.0e-05	0.88190
4	8	1.0e-06	0.88191
4	8	1.0e-07	0.88192
4	8	1.0e-08	0.88192

Table 8.1: The diffusion advection problem (8.1)–(8.2) defines u_ϵ. u_0 is given in (8.15).

since, by definition,

$$\oint_{\Gamma_+} v^2 \boldsymbol{\beta} \cdot \mathbf{n} \, ds \geq 0.$$

Thus if we suppose that $\Gamma_- \subset \Gamma$, meaning that the part of the boundary where we impose Dirichlet boundary conditions includes all of Γ_-, then $b(v,v) \geq 0$ for all $v \in V$, and thus $a_\beta(\cdot,\cdot)$ is coercive on V. In this case, u can be characterized uniquely via

$$u \in V \text{ satisfies } a_\beta(u,v) = (f,v)_{L^2(\Omega)} \quad \forall v \in V. \tag{8.11}$$

In the case that $\nabla\cdot \boldsymbol{\beta} \neq 0$, but $\nabla\cdot \boldsymbol{\beta} \leq 0$, coercivity of $a_\beta(\cdot,\cdot)$ again follows provided $\Gamma_- \subset \Gamma$. However, for more general $\boldsymbol{\beta}$, no guarantees can be made.

The code to generate the data in Table 8.1 is given in Program 8.1.

8.2.2 An example

Let $\Omega = [0,1]^2$ and $\boldsymbol{\beta} = (1,0)$. Note that $\nabla\cdot \boldsymbol{\beta} = 0$. Let u_ϵ denote the solution of (8.1), and in the case that $\epsilon \to 0$, we denote the limiting solution by u_0 (if it exists). We can solve (8.1) with $\epsilon = 0$ formally for a possilbe limit u_0 via

$$u_0(x,y) = \int_0^x f(s,y)\, ds + u_0(0,y). \tag{8.12}$$

Note that

$$\Gamma_- = \{(0,y) \; : \; y \in [0,1]\}, \quad \Gamma_+ = \{(1,y) \; : \; y \in [0,1]\}, \quad \Gamma_0 = \{(x,\pm 1) \; : \; x \in [0,1]\}.$$

We know that the variational problem (8.11) is well-posed provided $\Gamma_- \subset \Gamma$ (and provided $\epsilon > 0$). Then (8.12) implies that the likely limit would be

$$u_0(x,y) = \int_0^x f(s,y)\, ds + g(0,y). \tag{8.13}$$

For example, if $f \equiv 1$, then $u_0(x,y) = x + g(0,y)$ for all $x, y \in [0,1]^2$. This solution persists for $\epsilon > 0$ if, for example, $g(x,y) = a + by$, since $\Delta u_0 = 0$. However, we need to pick the right boundary conditions if we want to get this solution. On the other hand, a Neumann condition $\frac{\partial u_0}{\partial x}(1,y) = 0$ holds if $f(1,y) = 0$, e.g., if $f(x,y) = 1 - x$. Then

$$u_0(x,y) = x - \tfrac{1}{2}x^2 + g(0,y)$$

when $\epsilon = 0$. If in addition, $\frac{\partial g}{\partial y}(0,0) = \frac{\partial g}{\partial y}(0,1) = 0$, then u_0 satisfies a Neumann condition on the top and bottom of $\Omega = [0,1]^2$. For example, we can take

$$g(x,y) = y^2\Big(1 - \frac{2}{3}y\Big). \tag{8.14}$$

In this case, we take $\Gamma = \Gamma_-$ and

$$u_0(x,y) = x - \tfrac{1}{2}x^2 + y^2\Big(1 - \frac{2}{3}y\Big). \tag{8.15}$$

When ϵ is small, u_ϵ should be a small perturbation of this. From Table 8.1, we see that indeed this is the case.

But if we also take $\Gamma_+ \subset \Gamma$ then we potentially obtain a constraint, in that (8.13) implies $g(1,y) = \int_0^1 f(s,y)\,ds + g(0,y)$, that is,

$$\int_0^1 f(s,y)\,ds = g(1,y) - g(0,y) \text{ for all } y \in [0,1]. \tag{8.16}$$

If the data does not satisfy the constraint (8.16), we might expect some sort of boundary layer for $\epsilon > 0$. In the case that g is given in (8.14) and $f(x,y) = 1 - x$, such a constraint holds, and we see in Figure 8.1(b) that there is a sharp boundary layer for $\epsilon = 0.001$. For $\epsilon = 0.1$, Figure 8.1(a) shows that the solution deviates from u_0 over a broader area. The middle ground, where $\epsilon = 0.01$, the boundary layer is still localized, as shown in Figure 8.2(a). If we attempt to resolve this problem with too few grid points, as shown in Figure 8.2(b), then we get spurious oscillations on the scale of the mesh.

8.2.3 Wrong boundary conditions

Let us now ask the question: what happens if $\Gamma_- \not\subset \Gamma$? Suppose that we take $\Gamma = \Gamma_+$, and we consider the problem (8.1)–(8.2) with g given in (8.14) and $f(x,y) = 1 - x$. Numerical solutions for the variational problem (8.11) are depicted in Figure 8.3. These look at first to be reasonable. At least the case $\epsilon = 1.0$ looks plausible, and reducing ϵ by a factor of 10 produces something like the boundary layer behavior we saw previously. However, look at the scale. The solution is now extremely large. And if we continue to reduce ϵ (see Exercise 8.2), the solution size becomes disturbingly large. Picking different orders of polynomials and values for ϵ tend to give random, clearly spurious results. Thus we conclude that the coercivity condition provides good guidance regarding how to proceed.

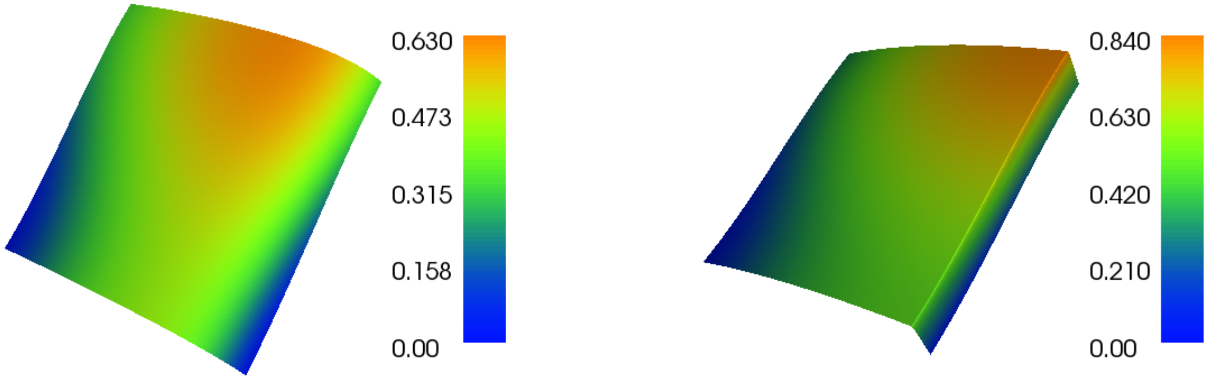

Figure 8.1: Diffusion-advection problem (8.1)–(8.2) with $\Gamma = \Gamma_- \cup \Gamma_+$ and g given in (8.14) and $f(x, y) = 1 - x$. Left: $\epsilon = 0.1$, u_ϵ computed using piecewise linears on a 100×100 mesh. Right: $\epsilon = 0.001$, u_ϵ computed using piecewise linears on a 1000×1000 mesh.

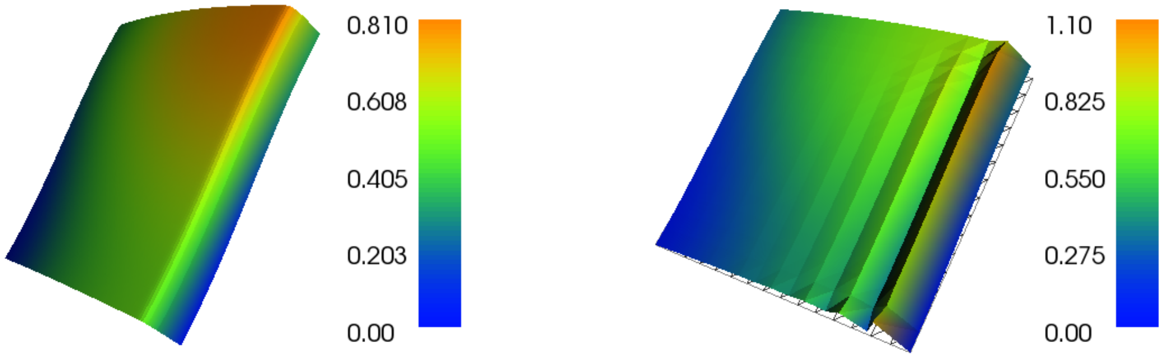

Figure 8.2: Diffusion-advection problem (8.1)–(8.2) with $\Gamma = \Gamma_- \cup \Gamma_+$ and g given in (8.14) and $f(x, y) = 1 - x$. Left: $\epsilon = 0.01$, u_ϵ computed using piecewise linears on a 100×100 mesh. Right: $\epsilon = 0.01$, u_ϵ computed using piecewise linears on a 15×15 mesh.

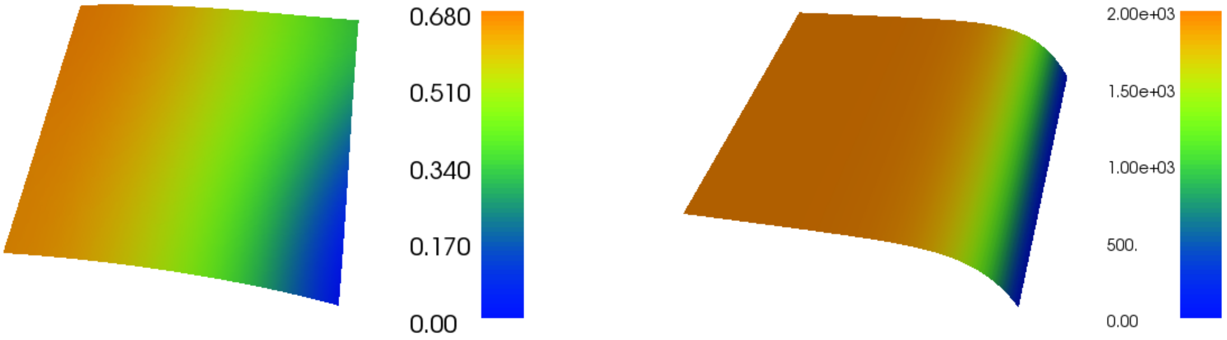

Figure 8.3: Diffusion-advection problem (8.1)–(8.2) with $\Gamma = \Gamma_+$ and g given in (8.14) and $f(x,y) = 1 - x$. The solution u_ϵ was computed using piecewise linears on a 100×100 mesh. Left: $\epsilon = 1.0$, Right: $\epsilon = 0.1$.

8.3 Transport equation

In some cases, there is no natural diffusion in a system, and we are left with pure advection. The resulting equation is often called a transport equation. Equations of this type play a major role in non-Newtonian fluid models as discussed in Chapter 17. As a model equation of this type, we consider

$$\tau u + \boldsymbol{\beta} \cdot \nabla u = f \text{ in } \Omega. \tag{8.17}$$

Without a diffusion term, it is not possible to pose Dirichlet boundary conditions arbitrarily. In the case where $\boldsymbol{\beta} \cdot \mathbf{n} = 0$ on $\partial\Omega$, the flow stays internal to Ω, and it has been shown [83, Proposition 3.7] that there is a unique solution $u \in L^2(\Omega)$ of (8.17) for any $f \in L^2(\Omega)$, provided that $\boldsymbol{\beta} \in H^1(\Omega)$. Such results are extended in [24, 25] to the general case in which boundary conditions are posed on Γ_-.

The variational formulation of (8.17) involves the bilinear form

$$a_\tau(u,v) = \int_\Omega \tau uv + (\boldsymbol{\beta} \cdot \nabla u) v \, d\mathbf{x}. \tag{8.18}$$

In this case, u can be characterized uniquely via

$$u \in V \text{ satisfies } a_\tau(u,v) = (f,v)_{L^2(\Omega)} \quad \forall v \in V. \tag{8.19}$$

In our simple example with $\boldsymbol{\beta} = (1,0)$, (8.17) can be written

$$\tau u(x,y) + u_{,x}(x,y) = f(x,y) \; \forall y \in [0,1].$$

Fix $y \in [0,1]$ and write $v(x) = e^{\tau x} u(x,y)$. Then

$$v'(x) = e^{\tau x} \big(\tau u(x,y) + u_{,x}(x,y) \big) = e^{\tau x} f(x,y),$$

so that

$$v(x) = v(0) + \int_0^x v'(s)\,ds = v(0) + \int_0^x e^{\tau s} f(s,y)\,ds.$$

Therefore

$$u(x,y) = e^{-\tau x} v(x) = e^{-\tau x}\Big(u(0,y) + \int_0^x e^{\tau s} f(s,y)\,ds\Big)\ \forall (x,y) \in [0,1] \times [0,1]. \tag{8.20}$$

For example, if we take $f(x,y) = e^{-\tau x}$, then $u(x,y) = g(y) + xe^{-\tau x}$, where g represents the Dirichlet data posed on $\Gamma = \Gamma_-$. We leave to Exercise 8.3 the development of a code for this problem.

The code to implement the transport problem (8.18) is given in Program 8.2.

8.4 Exercises

Exercise 8.1 *Consider the problem (8.5)–(8.6) implemented via the variational formulation (8.11). Explore the case $\Gamma = \Gamma_- \cup \Gamma_+$ with $\epsilon = 0.001$, data f and g as given in Figure 8.1, on a 100×100 mesh using piecewise linears. How large does the mesh need to be to eliminate the oscillations you see? What happens with higher degree approximations? How much smaller can you make the mesh? What gets the best balance of accuracy for time of computation?*

Exercise 8.2 *Consider the problem (8.5)–(8.6) implemented via the variational formulation (8.11). Explore the case $\Gamma = \Gamma_+$ for f and g as given in Figure 8.1. Solve with piecewise linears on a 100×100 mesh with $\epsilon = 0.01$. How big is the solution? Use quadratics and quartics and compare the size. But also is the solution negative in some cases?*

Exercise 8.3 *Consider the transport problem (8.17) implemented via the variational formulation (8.19) using the bilinear form in (8.18) where Ω and $\boldsymbol{\beta}$ are as in Section 8.2.2 and $\Gamma = \Gamma_-$. Take $f(x,y) = e^{-\tau x}$ and g is your choice. Modify your code for the diffusion-advection problem to implement this problem. Solve on various meshes and with various polynomial orders with $\epsilon = 10.0, 1.0, 0.1$ and other values of your choice. Use the exact solution (8.20) to test your code.*

```
from dolfin import *
import sys,math
from timeit import default_timer as timer

startime=timer()
pdeg=int(sys.argv[1])
meshsize=int(sys.argv[2])
acoef=float(sys.argv[3])

# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
V = FunctionSpace(mesh, "Lagrange", pdeg)

# Define Dirichlet boundary (x = 0)
def boundary(x):
  return x[0] < DOLFIN_EPS

# Define boundary condition
gee = Expression("x[1]*x[1]*(1.0-(2.0/3.0)*x[1])")
uex = Expression("(x[0]-(1.0/2.0)*x[0]*x[0])+ \
                  (x[1]*x[1]*(1.0-(2.0/3.0)*x[1]))")
bee = Constant((1.0,0.0))
bc = DirichletBC(V, gee, boundary)

# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("1.0-x[0]")
a = (acoef*inner(grad(u), grad(v))+inner(bee,grad(u))*v)*dx
L = f*v*dx

# Compute solution
u = Function(V)
solve(a == L, u, bc)
aftersolveT=timer()
totime=aftersolveT-startime
ue=interpolate(uex,V)
ge=interpolate(gee,V)
uerr=errornorm(ue,u,norm_type='l2', degree_rise=0)
print " ",pdeg," ",meshsize," %.1e"%acoef," %.1e"%uerr, \
      " %.5f"%(uerr/acoef)," %.3f"%totime
```

Program 8.1: Code to implement the advection problem (8.1)–(8.2).

```
from dolfin import *
import sys,math
from timeit import default_timer as timer

startime=timer()
pdeg=int(sys.argv[1])
meshsize=int(sys.argv[2])
acoef=float(sys.argv[3])

# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
V = FunctionSpace(mesh, "Lagrange", pdeg)

# Define Dirichlet boundary (x = 0)
def boundary(x):
  return x[0] < DOLFIN_EPS

# Define boundary condition
gee = Expression("x[1]*x[1]*(1.0-(2.0/3.0)*x[1])")
uex = Expression("(x[0]+(x[1]*x[1]*(1.0-(2.0/3.0)*x[1]))) \
                 *exp(-ac*x[0])",ac=acoef)
bee = Constant((1.0,0.0))
bc = DirichletBC(V, gee, boundary)

# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("exp(-ac*x[0])",ac=acoef)
a = (acoef*u*v+inner(bee,grad(u))*v)*dx
L = f*v*dx

# Compute solution
u = Function(V)
solve(a == L, u, bc)
aftersolveT=timer()
totime=aftersolveT-startime
uerr=errornorm(uex,u,norm_type=’l2’)
print " ",pdeg," ",meshsize," %.1e"%acoef," %.1e"%uerr," %.3f"%totime
# Plot solution
plot(u, interactive=True)
```

Program 8.2: Code to implement the transport problem (8.18).

Chapter 9

Mesh Adaptivity

In Section 4.1.4, refined meshes are shown to improve approximation substantially. Such meshes are derived analytically based on a priori information about the singularities of the solution. But it is possible to refine meshes automatically based on preliminary computations using techniques that estimate the size of the error for a given mesh. This fact is not at all obvious and requires some explanation.

One of the major advances of computational mathematics in the 20th century was the development of error indicators [40, Chapter 9]. Such indicators suggest where the computational error is largest and thus suggest regions where the mesh should be refined. This concept was pioneered mathematically by Ivo Babuška [13], primarily by focusing on the residual error for the equation. Again, it is not at all obvious that the residual could indicate where the error is large, but it is now well understood. The subject has developed significantly in the subsequent decades [5, 145, 172]. Other approaches to error estimation have also been widely used, especially the Zienkiewicz–Zhu error estimators [109, 131, 143].

9.1 Mesh terminology

There are various terms used to describe families of meshes. These become particularly relevant in the case of mesh adaptivity because several meshes are typically generated and we need to know what propterties hold uniformly for all of the meshes.

The simplest mesh restriction on a family of meshes $\{\mathcal{T}_h\}$ is that they be **nondegenerate**. We say that a mesh family is nondegenerate if there is a constant $C < \infty$ such that for each mesh, each element e in each mesh satisfies

$$\frac{\rho_{\max}(e)}{\rho_{\min}(e)} \leq C. \tag{9.1}$$

Here $\rho_{\min}(e)$ (respectively, $\rho_{\max}(e)$) is the radius of the largest ball contained in e (respectively, the radius of the smallest ball containing e).

A stronger notion of size restriction is that of **quasi-uniformity**. We make the assumption that for each mesh in the family, the parameter h satisfies

$$h = \max\left\{\rho_{\max}(e) \;:\; e \in \mathcal{T}_h\right\}. \tag{9.2}$$

That is, the mesh is labeled by its maximum element size. Then we say that a mesh family is **quasi-uniform** if there is a constant $C < \infty$ such that for each mesh, each element e in each mesh satisfies

$$\frac{h}{\rho_{\min}(e)} \leq C. \tag{9.3}$$

Note that any quasi-uniform family of meshes is necessarily nondegenerate.

There are important classes of meshes that **are** degenerate. One such class relates to problems with a large aspect ratio, or at least problems where there is a discrepancy between the approximation needs in one direction versus others [33, 34, 79, 108, 130]. Significant work regarding error estimators in such an anisotropic context has been done [7, 29, 62, 63, 64, 69, 107, 109, 110, 131, 143, 144, 162]. However, the subsequent material will be limited to the case of nondegenerate meshes.

9.2 Residual error estimators

Many successful error estimators are based on the **residual**. Consider the variational form

$$a_\alpha(v,w) = \int_\Omega \alpha(x)\nabla v(x)\cdot\nabla w(x)\,d\mathbf{x} \tag{9.4}$$

with α piecewise smooth, but not necessarily continuous. We will study the corresponding variational problem with Dirichlet boundary conditions on a polyhedral domain Ω in the n dimensions, so that $V = H_0^1(\Omega)$. For simplicity, we take the right-hand side for the variational problem to be a piecewise smooth function, f.

As usual, let V_h be piecewise polynomials of degree less than k on a mesh $\mathcal{T}_h$, and assume that the discontinuities of α and f fall on mesh faces (edges in two dimensions) in $\mathcal{T}_h$. That is, both α and f are smooth on each $T \in \mathcal{T}_h$. However, we will otherwise only assume that $\mathcal{T}_h$ is non-degenerate [40], since we will want to allow significant local mesh refinement.

Let u satisfy the usual variational formulation $a_\alpha(u,v) = (f,v)_{L^2(\Omega)}$ for all $v \in V$, and let $u_h \in V_h$ be the standard Galerkin approximation. The residual $R_h \in V'$ is defined by

$$R_h(v) = a_\alpha(u-u_h,v) \quad \forall v \in V. \tag{9.5}$$

Note that, by definition,

$$R_h(v) = 0 \quad \forall v \in V_h, \tag{9.6}$$

assuming $V_h \subset V$.

Let $\mathcal{A}$ denote the differential operator formally associated with the form (9.4), that is, $\mathcal{A}v := -\nabla\cdot(\alpha\nabla v)$. The residual can be computed by

$$\begin{aligned} R_h(v) &= \sum_T \int_T (f + \nabla\cdot(\alpha\cdot\nabla u_h))v\,dx + \sum_e \int_e [\alpha\mathbf{n}_e\cdot\nabla u_h]_{\mathbf{n}_e} v\,ds \\ &= \sum_T \int_T (f - \mathcal{A}u_h)v\,dx + \sum_e \int_e [\alpha\mathbf{n}_e\cdot\nabla u_h]_{\mathbf{n}_e} v\,ds \quad \forall v \in V, \end{aligned} \tag{9.7}$$

where $\mathbf{n}_e$ denotes a unit normal to e and $[\phi]_{\mathbf{n}}$ denotes the jump in ϕ across the face normal to $\mathbf{n}$:

$$[\phi]_{\mathbf{n}}(x) := \lim_{\epsilon\to 0} \phi(x+\epsilon\mathbf{n}) - \phi(x-\epsilon\mathbf{n}).$$

so that the expression in (9.7) is independent of the choice of normal $\mathbf{n}$ on each face.

There are two parts to the residual. One is the integrable function R_A defined on each element T by

$$R_A|_T := (f + \nabla\cdot(\alpha\nabla u_h))\,|_T = (f - \mathcal{A}u_h)\,|_T, \tag{9.8}$$

and the other is the "jump" term

$$R_J(v) := \sum_e \int_e [\alpha\mathbf{n}_e\cdot\nabla u_h]_{\mathbf{n}_e} v\, ds \quad \forall v\in V \tag{9.9}$$

The proof of (9.7) is derived by integrating by parts on each T, and the resulting boundary terms are collected in the term R_J. Assuming that $a_\alpha(\cdot,\cdot)$ is coercive on $H^1(\Omega)$, and inserting $v = e_h$ in (9.5), we see that

$$\frac{1}{c_0}|e_h|^2_{H^1(\Omega)} \le |a_\alpha(e_h,e_h)| = |R_h(e_h)|. \tag{9.10}$$

Therefore

$$\|e_h\|_{H^1(\Omega)} \le c_0 \sup_{v\in H^1_0(\Omega)} \frac{|R_h(v)|}{\|v\|_{H^1(\Omega)}}. \tag{9.11}$$

The right-hand side of (9.11) is the $H^{-1}(\Omega)$ norm of the residual. This norm is in principle computable in terms of the data of the problem (f and α) and u_h. But the typical approach is to estimate it using what is known as a Scott-Zhang [176] interpolant $\mathcal{I}_h$ which satisfies, for some constant γ_0,

$$\|v - \mathcal{I}_h v\|_{L^2(T)} \le \gamma_0 h_T |v|_{H^1(\widehat{T})} \tag{9.12}$$

for all $T\in\mathcal{T}_h$, where $\widehat{T}$ denotes the union of elements that contact T, and

$$\|v - \mathcal{I}_h v\|_{L^2(e)} \le \gamma_0 h_e^{1/2} |v|_{H^1(T_e)} \tag{9.13}$$

for all faces e in $\mathcal{T}_h$, where T_e denotes the union of elements that share the face e, where h_e (resp. h_T) is a measure of the size of e (resp. T). Dropping the subscript "e" when referring to a normal $\mathbf{n}$ to e, we get

$$\begin{aligned}|R_h(v)| &= |R_h(v-\mathcal{I}_h v)|\\ &= \Big|\sum_T \int_T R_A(v-\mathcal{I}_h v)\,dx + \sum_e \int_e [\alpha\mathbf{n}\cdot\nabla u_h]_{\mathbf{n}}(v-\mathcal{I}_h v)\,ds\Big|.\end{aligned} \tag{9.14}$$

Applying (9.12) and (9.13) to (9.14) we find

$$\begin{aligned}|R_h(v)| &= |R_h(v-\mathcal{I}_h v)|\\ &\leq \sum_T \|R_A\|_{L^2(T)}\|v-\mathcal{I}_h v\|_{L^2(T)} + \sum_e \|\,[\alpha\mathbf{n}\cdot\nabla u_h]_{\mathbf{n}}\,\|_{L^2(e)}\|v-\mathcal{I}_h v\|_{L^2(e)}\\ &\leq \sum_T \|R_A\|_{L^2(T)}\gamma_0 h_T |v|_{H^1(\widehat{T})} + \sum_e \|\,[\alpha\mathbf{n}\cdot\nabla u_h]_{\mathbf{n}}\,\|_{L^2(e)}\gamma_0 h_e^{1/2}|v|_{H^1(\widehat{T}_e)} \qquad (9.15)\\ &\leq \gamma\Big(\sum_T \|R_A\|^2_{L^2(T)}h_T^2 + \sum_e \|\,[\alpha\mathbf{n}\cdot\nabla u_h]_{\mathbf{n}}\,\|^2_{L^2(e)}h_e\Big)^{1/2}|v|_{H^1(\Omega)},\end{aligned}$$

where $\gamma = C\gamma_0$ for some constant C that depends only on the maximum number of elements in $\widehat{T}$ for each T. In view of (9.14), the **local error indicator** $\mathcal{E}_e$ is defined by

$$\mathcal{E}_e(u_h)^2 := \sum_{T\subset T_e} h_T^2\|f+\nabla\cdot(\alpha\nabla u_h)\|^2_{L^2(T)} + h_e\|\,[\alpha\mathbf{n}\cdot\nabla u_h]_{\mathbf{n}}\,\|^2_{L^2(e)}. \qquad (9.16)$$

With this definition, the previous inequalities can be summarized as

$$|R_h(v)| \leq \gamma\Big(\sum_e \mathcal{E}_e(u_h)^2\Big)^{1/2}|v|_{H^1(\Omega)}, \qquad (9.17)$$

which in view of (9.11) implies that

$$|e_h|_{H^1(\Omega)} \leq \gamma c_0\Big(\sum_e \mathcal{E}_e(u_h)^2\Big)^{1/2}, \qquad (9.18)$$

where γ is a constant only related to interpolation error.

9.3 Local error estimates and refinement

In the previous section, an upper bound for the global error $|u-u_h|_{H^1(\Omega)}$ was given in terms of locally defined error estimators (9.16). If the data f and α are themselves piecewise polynomials of some degree, there is a lower bound for the local error [40, Section 9.3]

$$|e_h|_{H^1(T_e)} \geq c\,\mathcal{E}_e(u_h) \qquad (9.19)$$

where $c>0$ depends only on the non-degeneracy constant for $\mathcal{T}_h$. One corollary of this estimate is the reverse inequality to (9.18),

$$|e_h|_{H^1(\Omega)} \geq \frac{c}{\sqrt{2}}\Big(\sum_{e\in\mathcal{T}_h}\mathcal{E}_e(u_h)^2\Big)^{1/2}.$$

A reverse inequality to (9.19) (a local *upper* bound) is not true in general. However, the local lower bound (9.19) suggests the philosophy that

the mesh should be refined wherever the local error indicator $\mathcal{E}_e(u_h)$ is big.

Unfortunately, we cannot be sure that where it is small that the error will necessarily be small. Distant effects may pollute the error and make it large even if the error indicator $\mathcal{E}_e(u_h)$ is small nearby.

9.4 Other norms

It is possible to have error estimators for other norms. For example, the pointwise error $u - u_h$ at $\mathbf{x}$ can be represented using the Green's function (Section 4.2) $G^{\mathbf{x}}$ via

$$(u - u_h)(\mathbf{x}) = a_\alpha(u - u_h, G^{\mathbf{x}}) = a_\alpha(u - u_h, G^{\mathbf{x}} - v) = R_h(G^{\mathbf{x}} - v) \quad \forall v \in V_h. \tag{9.20}$$

Thus choosing v as the Scott-Zhang interpolant of $G^{\mathbf{x}}$ [65, page 719] leads to an error indicator of the form

$$\mathcal{E}^\infty(u_h) := \max_{T\in\mathcal{T}_h} \Big(h_T^2 \|f + \nabla\cdot(\alpha\nabla u_h)\|_{L^\infty(T)} + h_e \| \left[\alpha\mathbf{n}\cdot\nabla u_h\right]_{\mathbf{n}} \|_{L^\infty(e)} \Big). \tag{9.21}$$

It can be proved [65] that there is a constant C such that

$$\|u - u_h\|_{L^\infty(\Omega)} \le C\mathcal{E}^\infty(u_h).$$

The error estimators in [65] apply as well to nonlinear problems and to singular perturbation problems as in Exercise 5.1. An extension to anisotropic meshes is given in [106] for two-dimensional problems and piecewise linear approximation.

9.5 Other goals

Instead of attempting to estimate norms of the error, we can estimate linear functionals of the error. For example, the quantity of interest (C_6) in Section 5.2.1 is an integral (5.18). The strategy of **goal oriented error estimation** [149] (a.k.a. **dual weighted residual method** [23]) is to solve a adjoint problem to find $z \in V$ such that

$$a^*(z, v) = \mathcal{M}(v) \quad \forall v \in V, \tag{9.22}$$

where $\mathcal{M}$ is the linear functional to be optimized. Here, the adjoint form $a^*(\cdot,\cdot)$ is defined for any bilinear form $a(\cdot,\cdot)$ via

$$a^*(v, w) = a(w, v) \quad \forall v, w \in V. \tag{9.23}$$

The concept of adjoint form can also be extended [149] to the case where $a(\cdot,\cdot)$ is defined on a pair of spaces $V \times W$ instead of $V \times V$ as is the usual case considered so far. It is also possible to extend the theory to allow the goal functional $\mathcal{M}$ to be nonlinear.

The code to generate Figure 9.1 is given in Program 9.1.

If $a(\cdot,\cdot)$ is symmetric, then $a^*(\cdot,\cdot)$ is the same as $a(\cdot,\cdot)$. But it is different for a form like $a_\beta(\cdot,\cdot)$ defined in Section 8.2:

$$a_\beta(v, w) = \int_\Omega \nabla v(\mathbf{x}) \cdot \nabla w(\mathbf{x}) + \big(\boldsymbol{\beta}(\mathbf{x}) \cdot \nabla v(\mathbf{x})\big) w(\mathbf{x})\, d\mathbf{x}.$$

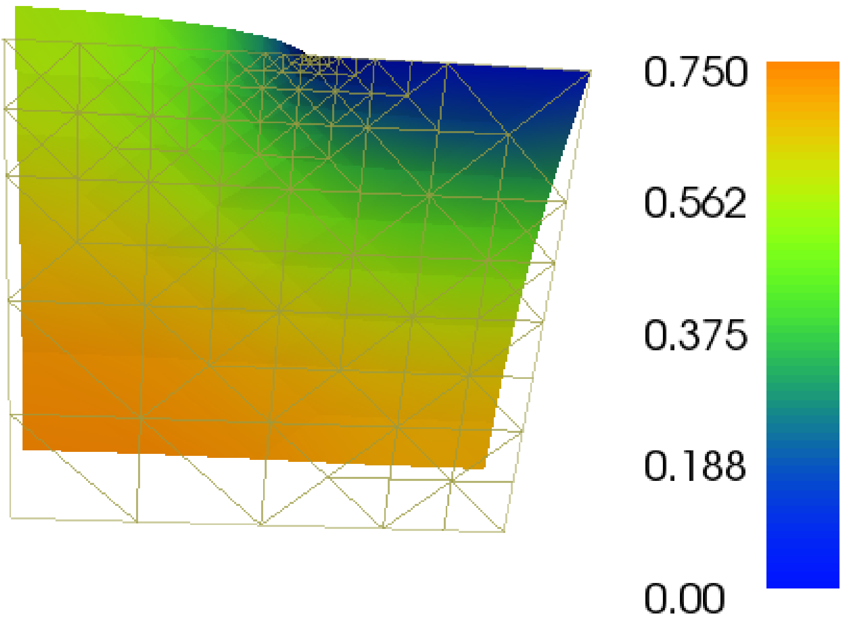

Figure 9.1: Adaptivity applied to the problem (4.10) using piecewise linears and an initial mesh of size 4 with a goal $\mathcal{M}(u)=\int_\Omega u^2\,d\mathbf{x}$. The initial, unrefined mesh is apparent in the lower-left corner of the domain.

We see from (8.7) that, if $\nabla\cdot\boldsymbol{\beta}=0$ and Dirichlet conditions are imposed on the boundary wherever $\boldsymbol{\beta}\cdot\mathbf{n}\neq 0$, then

$$\begin{aligned} a^*_\beta(v,w)=a_\beta(w,v)&=\int_\Omega \nabla v(\mathbf{x})\cdot\nabla w(\mathbf{x})-\big(\boldsymbol{\beta}(\mathbf{x})\cdot\nabla v(\mathbf{x})\big)w(\mathbf{x})\,d\mathbf{x}\\ &=a_{-\beta}(v,w)\quad\forall v,w\in V.\end{aligned}\tag{9.24}$$

Suppose as usual that $u\in V$ satisfies $a(u,v)=F(v)$ for all $v\in V$, that $u_h\in V_h$ satisfies $a(u_h,v)=F(v)$ for all $v\in V_h$, and that $V_h\subset V$. Then $a(u-u_h,v)=0$ for all $v\in V_h$, and

$$\begin{aligned}\mathcal{M}(u)-\mathcal{M}(u_h)=\mathcal{M}(u-u_h)&=a^*(z,u-u_h)=a(u-u_h,z)\\ &=a(u-u_h,z-v)=R_h(z-v)\quad\forall v\in V_h.\end{aligned}\tag{9.25}$$

Note the analogy with (9.20), where instead of the Green's function $G^{\mathbf{x}}$ we have z.

It is helpful to re-write (9.14) by re-balancing the jump terms via

$$|R_h(v)|=\Big|\sum_T\int_T R_A(v-\mathcal{I}_h v)\,dx+\sum_{e\subset\partial T}\int_e[\alpha\mathbf{n}\cdot\nabla u_h]^*_{\mathbf{n}}(v-\mathcal{I}_h v)\,ds\Big|,\tag{9.26}$$

where $[\phi]^*_{\mathbf{n}} = \frac{1}{2}[\phi]_{\mathbf{n}}$ for interior edges (or faces) and $[\phi]^*_{\mathbf{n}} = \phi$ for boundary edges (or faces). Thus the local error indication η_T is defined by

$$\eta_T(v) = \Big| \int_T R_A(v - \mathcal{I}_h v)\, dx + \sum_{e \subset \partial T} \int_e [\alpha \mathbf{n} \cdot \nabla u_h]^*_{\mathbf{n}} (v - \mathcal{I}_h v)\, ds \Big|.$$

Then

$$|\mathcal{M}(u - u_h)| \leq \sum_T \eta_T(z).$$

The strategy then is to refine the mesh where $\eta_T(z)$ is large.

The difficulty now is that we do not know z, and we need to compute it. Moreover, if we simply use the same approximation space V_h to compute z_h, then we get a false impression (take $v = z_h$ in (9.25)). Thus there is a need to have a higher-order approximation of z than would normally be provided via V_h. Different approaches to achieving this have been studied [23], including simply approximating via a globally higher-order method.

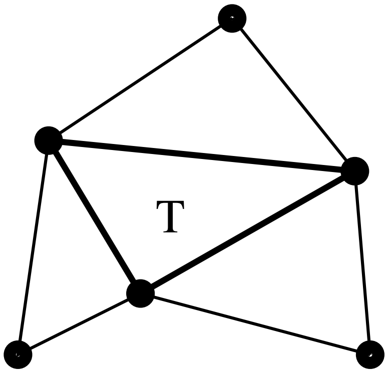

Figure 9.2: Nearby elements are used to construct a higher-order approximation to z.

What is done in Dolfin [149] (see also [23]) is to first compute z_h, then interpolate it on patches around a given element using a higher-degree approximation, as indicated in Figure 9.2, using the interpolant as an approximation to z. In Figure 9.2, we see what would be done for a piecewise linear approximation, and we see that including three neighboring triangles gives exactly the data needed to construct a quadratic. This approach is effective and relatively inexpensive, but there is a philosophical conundrum: if this approach does give higher-order accuracy, why not simply use it instead of using error estimators? One answer to this is that the error estimator is only trying to find where $z - \mathcal{I}_h z$ is large, that is where the derivatives of z are large. This is a more restricted problem than trying to get higher accuracy in general.

Another issue to consider is that there may be singularities in z that make its approximation poor. Fortunately, there is a duality between estimating the accuracy of u_h and the approximation of z [23] that can address this. However, there is a certain amount of art in goal-based error estimation that cannot be fully justified rigorously. In any case, it is clear that the approach works from Figure 9.1, where the problem (4.10) has been revisited using piecewise linears on an initial mesh of size 4 with a goal $\mathcal{M}(u) = \int_\Omega u^2 \, d\mathbf{x}$. The initial, unrefined mesh is apparent in the lower-left corner of the domain.

9.6 An example

Suppose that $a(\cdot,\cdot)$ is symmetric, so that $a^*(\cdot,\cdot) = a(\cdot,\cdot)$, and that $\mathcal{M}(v) = F(v)$. Then $z = u$. Such a situation occurs in Section 5.2.1, and it is explored further in Exercise 9.1.

9.7 Mesh generation

Even when properties of desired meshes are known, it is still a difficult task to generate meshes with these properties [152].

9.8 Exercises

Exercise 9.1 *Re-do the problem in Section 5.2.1 using adaptivity to compute C_6 via (5.18), with the goal*

$$\mathcal{M}(v) = \int_0^\infty \int_0^\infty r_1^2 r_2^2 e^{-(r_1+r_2)} v(r_1,r_2)\, dr_1 dr_2.$$

Truncate the integral in this definition appropriately to match the computational approach taken in Section 5.2.1.

Exercise 9.2 *Consider the variational form*

$$b_\beta(v,w) = \int_\Omega (\boldsymbol{\beta}\cdot\nabla v) w \, d\mathbf{x} + \oint_{\Gamma_-^\beta} u\, v\, \boldsymbol{\beta}\cdot\mathbf{n}\, ds,$$

where we define $\Gamma_-^\beta = \{\mathbf{x}\in\partial\Omega \,:\, \boldsymbol{\beta}(\mathbf{x})\cdot\mathbf{n}(\mathbf{x}) < 0\}$. Suppose that $\nabla\cdot\boldsymbol{\beta} = 0$. Use (8.7) to show that $b_\beta(v,w) = -b_{-\beta}(w,v)$. (Hint: show that $\Gamma_-^{-\beta} = \Gamma_+^\beta$.)

Exercise 9.3 *Consider the problem (4.10) using adaptivity with various goals. How course can the initial mesh be to obtain reasonable results?*

Exercise 9.4 *Define a goal function $\mathcal{M}$ via*

$$\mathcal{M}(v) = \int_\Omega \delta_{\mathbf{x}_0}^A \, v \, d\mathbf{x},$$

where $\delta_{\mathbf{x}_0}^A$ is defined in (4.20) and A is sufficiently large. Explain what this goal is trying to do in words (hint: look at Section 4.2). Write a code to use this goal in the problem (4.19) with $\mathbf{x}_0$ near $(\frac{1}{2},\frac{1}{2})$, and explain what happens. Compare this with having the goal $\mathcal{M}(v) = \int_\Omega v(x)^2 \, d\mathbf{x}$.

```
from dolfin import *
import sys,math

parameters["form_compiler"]["quadrature_degree"] = 12

meshsize=int(sys.argv[1])
pdeg=int(sys.argv[2])
qdeg=int(sys.argv[3])
mytol=float(sys.argv[4])

# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
V = FunctionSpace(mesh, "Lagrange", pdeg)

# Define Dirichlet boundary (x = 0 or x = 1 or y = 0 or y = 1)
def boundary(x):
  return x[0] > 0.5 and x[1] < DOLFIN_EPS

# Define boundary condition
u0 = Constant(0.0)
bc = DirichletBC(V, u0, boundary)

# Define variational problem
u = Function(V)
v = TestFunction(V)
f = Expression("1.0")
J = u*u*dx
F = (inner(grad(u), grad(v)))*dx - f*v*dx

# Compute solution
solve(F == 0, u, bc, tol=mytol, M=J)
# Plot solution
plot(u.leaf_node(), interactive=True)
```

Program 9.1: Code to generate Figure 9.1.

Chapter 10

Geometry approximation

If the boundary $\partial\Omega$ of a domain Ω is curved, it is often necessary to approximate it in some way. For simplicity, we will consider Laplace equation (2.1), viz.,

$$-\Delta u = f \text{ in } \Omega$$

together with homogeneous Diriclet boundary conditions on all of $\partial\Omega$:

$$u = 0 \text{ on } \partial\Omega.$$

Such boundary conditions are easy to satisfy on polygonal boundaries with piecewise polynomials, provided only that the mesh match the vertices of the boundary (in two dimensions, in three dimensions, edges must also be respected). However, for curved boundaries, exact satisfaction of boundary conditions is typically not possible. There are various ways in which this can be addressed:

- interpolate the boundary conditions [160] (a collocation approach)
- modify the polynomials via a change of coordinates (isoparametric elements)
- incorporate the boundary conditions into the variational form (Nitsche's method).

The name **isoparametric element** was coined by Bruce Irons[1] in part as a play on words, assuming that the audience was familiar with isoperimetric inequalities. The concept of employing a coordinate transformation to extend the applicability of finite elements was initiated by Ian Taig and later developed by Irons [2].

10.1 Nitsche's method

The code to generate Table 10.1 is given in Program 10.1.

[1] Bruce Irons (1924—1983) is known for many concepts in finite element analysis, including the Patch Test for nonconforming elements [57] and frontal solvers, among others.

poly. order	mesh no.	γ	L^2 error
1	32	1.00e+01	2.09e-03
2	8	1.00e+01	5.16e-04
4	8	1.00e+01	1.77e-06
4	128	1.00e+01	4.96e-12
8	16	1.00e+01	1.68e-11
16	8	1.00e+01	2.14e-08

Table 10.1: L^2 errors for Nitsche's method: effect of varying polynomial order and mesh size for fixed $\gamma = 10$. Here we take $h = N^{-1}$ where N is the mesh number.

The method of Nitsche[2] is useful for both curved and polygonal domains. It allows the use of functions that do not satisfy Dirichlet boundary conditions to approximate solutions which do satisfy Dirichlet boundary conditions. Define

$$a_\gamma(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\mathbf{x} + \gamma h^{-1} \oint_{\partial\Omega} uv \, dx - \oint_{\partial\Omega} \frac{\partial u}{\partial n} v \, dx - \oint_{\partial\Omega} \frac{\partial v}{\partial n} u \, dx, \tag{10.1}$$

where $\gamma > 0$ is a fixed parameter and h is the mesh size. We claim that the solution to Laplace's equation satisfies the variational problem

$$u \in V \text{ such that } a_\gamma(u,v) = (f,v)_{L^2} \quad \forall v \in V,$$

where $V = H_0^1(\Omega)$. The reason is that, for $u, v \in V$,

$$a_\gamma(u,v) = a(u,v),$$

where $a(\cdot,\cdot)$ is the usual bilinear form for the Laplace operator:

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\mathbf{x}.$$

But more generally, we know from (2.8) that, for all $v \in H^1(\Omega)$,

$$\int_\Omega f v \, d\mathbf{x} = \int_\Omega (-\Delta u) v \, d\mathbf{x} = a(u,v) - \oint_{\partial\Omega} v \frac{\partial u}{\partial n} \, ds = a_\gamma(u,v), \tag{10.2}$$

since $u = 0$ on $\partial\Omega$.

Now consider the discrete problem

$$\text{find } u_h \in V_h \text{ such that } a_\gamma(u_h, v) = (f,v)_{L^2} \quad \forall v \in V_h, \tag{10.3}$$

[2] Joachim A. Nitsche (1926—1996) made major contributions to the mathematical theory of the finite element method. In addition to his method for enforcing boundary conditions, his name is often invoked in referring to the duality technique for proving error estimates in lower-order norms, as Nitsche's Trick.

poly. order	mesh no.	γ	L^2 error
1	8	1.00e+02	3.23e-02
1	8	1.00e+01	3.10e-02
1	8	2.00e+00	2.76e-02
1	8	1.50e+00	3.93e-02
1	8	1.10e+00	6.00e-02
1	8	1.00e+00	1.80e-01
1	16	1.00e+00	2.11e-01
1	32	1.00e+00	1.81e-01
1	64	1.00e+00	1.40e-01
1	128	1.00e+00	1.03e-01
1	256	1.00e+00	7.43e-02

Table 10.2: L^2 errors for Nitsche's method: effect of varying γ.

where $V_h \subset H^1(\Omega)$ is not required to be a subset of $H^1_0(\Omega)$. As a consequence of (10.3) and (10.2), we have

$$a_\gamma(u-u_h, v) = 0 \quad \forall v \in V_h. \tag{10.4}$$

Nitsche's method is of interest when $V_h \not\subset V$, for otherwise we could use the usual formulation. The bilinear form $a_\gamma(v,v)$ is not defined for general $v \in H^1(\Omega)$. For example, take $\Omega = [0,1]^2$ and $v(x,y) = 1 + x^{2/3}$ for $(x,y) \in \Omega$. Then

$$v_{,x}(x,y) = \tfrac{2}{3}x^{-1/3} \text{ for all } x,y \in [0,1]^2.$$

Thus, $v_{,x}$ is square integrable on Ω, and since $v_{,y} = 0$, it is also square integrable on Ω. So $v \in H^1(\Omega)$. But, $v_{,x}(x,y) \to \infty$ as $x \to 0$ for all $y \in [0,1]$. In particular, this means that $\frac{\partial v}{\partial n} = \infty$ on the part of the boundary $\{(0,y) \,:\, y \in [0,1]\}$.

Nitsche's produces results of the same quality as are obtained with specifying Dirichlet conditions explicitly, as indicated in Table 10.1 (compare with Table 3.1). However, for particular values of γ, the behavior can be suboptimal, as indicated in Table 10.2.

Since $a_\gamma(\cdot,\cdot)$ is not continuous on $H^1(\Omega)$, we cannot use our standard approach to analyze the behavior of the Nitsche method (10.3). Or at least we cannot use the standard norms. Instead we define

$$|\!|\!|\, v \,|\!|\!| = \Big(a(v,v) + h\oint_{\partial\Omega} \Big|\frac{\partial v}{\partial n}\Big|^2 ds + h^{-1}\oint_{\partial\Omega} v^2\, ds\Big)^{1/2}. \tag{10.5}$$

The philosophy of this norm is that it penalizes departure from the Dirichlet boundary condition but it minimizes the impact of the normal derivative on the boundary. But it is also easy to see that this norm matches the different parts of the Nitsche bilinear form, so that

$$|a_\gamma(v,w)| \le C|\!|\!|\, v \,|\!|\!|\; |\!|\!|\, w \,|\!|\!| \quad \forall v,w \in H^1(\Omega), \tag{10.6}$$

with the proviso that the right-hand side of (10.6) might be infinite. One can show [170] that

$$|||\, v \,||| \leq Ch^{-1}\|v\|_{L^2(\Omega)} \quad \forall v \in V_h, \tag{10.7}$$

for any space V_h of piecewise polynomials on a reasonable mesh. Moreover, for such spaces, there exist $\gamma_0 > 0$ and $\alpha > 0$ such that, for $\gamma \geq \gamma_0$,

$$\alpha |||\, v \,|||^2 \leq a_\gamma(v,v) \quad \forall v \in V_h. \tag{10.8}$$

Although $|||\, v \,|||$ is not finite for all $v \in V$, we can assume that our solution u is sufficiently smooth that $|||\, u \,||| < \infty$. In this case, one can prove [170] that

$$|||\, u - u_h \,||| \leq \Big(1 + \frac{C}{\alpha}\Big) \inf_{v \in V_h} |||\, u - v \,|||_{L^2(\Omega)} \leq C' h^k \|u\|_{H^{k+1}(\Omega)} \tag{10.9}$$

using piecewise polynomials of degree k. In particular, this guarantees that

$$\|u_h\|_{L^2(\partial\Omega)} \leq Ch^{k+1/2}\|u\|_{H^{k+1}(\Omega)},$$

so that the Dirichlet conditions are closely approximated by Nitsche's method.

10.2 Exercises

Exercise 10.1 *Experiment with Nitsche's method for imposing homogeneous Dirichlet conditions on the unit square. Choose different meshes, polynomial degrees, and parameter γ. Can you take $\gamma = 0$? Can you take $\gamma < 0$?*

```
from dolfin import *
import sys, math

mypi=math.pi

meshsize=int(sys.argv[1])
pdeg=int(sys.argv[2])
gamma=float(sys.argv[3])
h=1.0/float(meshsize)

# Create mesh and define function space
mesh = UnitSquareMesh(meshsize, meshsize)
n = FacetNormal(mesh)
V = FunctionSpace(mesh, "Lagrange", pdeg)

# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("(sin(mypi*x[0]))*(sin(mypi*x[1]))",mypi=math.pi)
a = inner(grad(u), grad(v))*dx -u*(inner(n,grad(v)))*ds- \
          v*(inner(n,grad(u)))*ds+(gamma/h)*u*v*ds
L = (2*mypi*mypi)*f*v*dx

# Compute solution
u = Function(V)
solve(a == L, u )

print pdeg,meshsize," %.2e"%gamma," %.2e"%errornorm(f,u,norm_type='l2',\
      degree_rise=3)
```

Program 10.1: Code to implement Nitche's method for the problem (2.32).

Chapter 11

Scalar Elliptic Problems

The general scalar elliptic problem takes the form

$$-\sum_{i,j=1}^{d} \frac{\partial}{\partial x_j} \left(\alpha_{ij}(\mathbf{x}) \frac{\partial u}{\partial x_i}(\mathbf{x}) \right) = f(\mathbf{x}) \tag{11.1}$$

where the α_{ij} are given functions, together with suitable boundary conditions of the type considered previously.

To be elliptic, the functions $\alpha_{ij}(\mathbf{x})$ need to form a postive definite matrix at almost every point $\mathbf{x}$ in a domain Ω. Often, this is a symmetric matrix. More precisely, it is assumed that for some finite, positive constant C,

$$C^{-1} \leq |\xi|^{-2} \sum_{i,j=1}^{d} \alpha_{ij}(\mathbf{x})\, \xi_i\, \xi_j \leq C \quad \forall\, 0 \neq \xi \in \mathbb{R}^d, \text{ for almost all } \mathbf{x} \in \Omega. \tag{11.2}$$

Here the expression “for almost all” means the condition can be ignored on a set of measure zero, such as a lower-dimensional surface running through Ω. However, there is no need for the $\alpha_{ij}'s$ to be continuous, and in many important physical applications they are not.

An interpretation of the general scalar elliptic problem in classical terms is difficult when the α_{ij}’s are not differentiable. However, the variational formulation is quite simple. Define

$$a(u,v) := \int_\Omega \sum_{i,j=1}^{d} \alpha_{ij}(\mathbf{x}) \frac{\partial u}{\partial x_i}(\mathbf{x}) \frac{\partial v}{\partial x_j}\, d\mathbf{x}. \tag{11.3}$$

Using this bilinear form, the problem (11.1) can be posed as in (2.23) where the boundary conditions are incorporated in the space V as described in Section 2.1.

11.1 Discontinuous coefficients

It is frequently the case that the coefficients which arise in physical models vary so dramatically that it is appropriate to model them as discontinuous. These often arise due to a change in materials or material properties. Examples can be found in the modeling of nuclear reactors [71], porous media [70], semi-conductors [19], proteins in a solvent [14, 97] and on and on. However, the lack of continuity of the coefficients has less effect on the model than might be thought at first.

The critical factor is the *ellipticity* of the coefficients. We suppose that the coefficients form a positive definite matrix almost everywhere, that is, that

$$\sum_{i=1}^{d} \xi_i^2 \le c_0 \sum_{i,j=1}^{d} \alpha_{ij}(\mathbf{x})\xi_i\xi_j \quad \forall \xi \in \mathbb{R}^d,\ \mathbf{x} \in \Omega_0 \tag{11.4}$$

for some positive constant c_0 and some set Ω_0 where the complementary set $\Omega\backslash\Omega_0$ has *measure zero*, that is, contains no sets of positive volume. Examples of such sets are ones which consist of sets of lower-dimensional surfaces. On the set $\Omega\backslash\Omega_0$ the coefficients may jump from one value to another and so have no precise meaning, and we are allowed to ignore such sets.

We also assume that the coefficients are bounded almost everywhere:

$$\sum_{i,j=1}^{d} \alpha_{ij}(\mathbf{x})\xi_i\nu_j \le c_1\, |\xi|\, |\nu| \quad \forall \xi, \nu \in \mathbb{R}^d,\ \mathbf{x} \in \Omega_0, \tag{11.5}$$

for some finite constant c_1, where $|\xi|^2 = \sum_{i=1}^d \xi_i^2$.

The ellipticity constant ε is the ratio

$$\varepsilon := \frac{1}{c_0\, c_1}. \tag{11.6}$$

For elliptic coefficients, the coercivity condition (2.20) and the corresponding stability result (2.21) both hold, where $C = c_0\, c_1 = \varepsilon^{-1}$.

There is a subtle dependence of the regularity of the solution in the case of discontinuous coefficients [129]. It is not in general the case that the gradient of the solution is bounded. However, from the variational derivation, we see that the gradient of the solution is always square integrable. A bit more is true, that is, the p-th power of the solution is integrable for $2 \le p \le P_\varepsilon$ where P_ε is a number bigger than two depending only on the ellipticity constant ε in (11.6) (as ε tends to zero, P_ε tends to two).

Using the variational form (11.3) of the equation (11.1), it is easy to see that the *flux*

$$\sum_{i=1}^{d} \alpha_{ij}(\mathbf{x}) \frac{\partial u}{\partial x_i}(\mathbf{x}) n_j \tag{11.7}$$

is continuous across an interface normal to $\mathbf{n}$ even when the α_{ij}'s are discontinuous across the interface. This implies that the normal slope of the solution must have a jump (that is, the graph has a kink).

11.2 Dielectric models

One of the most important scalar elliptic equations with discontinuous coefficients is a model for the dielectric behavior of a protein in water. This takes the form

$$\begin{aligned} -\nabla\cdot(\epsilon\nabla u) &= \sum_{i=1}^{N} c_i\,\delta_{\mathbf{x}_i} \quad \text{in } \mathbb{R}^3 \\ u(\mathbf{x}) &\to 0 \quad \text{as } \mathbf{x}\to\infty, \end{aligned} \tag{11.8}$$

where the dielectric constant ϵ is small inside the protein, which we assume occupies the domain Ω, and large outside. Here, the point charges at $\mathbf{x}_i$ are modeled via Dirac δ-functions $\delta_{\mathbf{x}_i}$. The constant c_i corresponds to the charge at that point.

Confusion has arisen about the efficacy of error estimators due to the need for resolving point singularities $\mathbf{x}_i \in \Omega$ resulting from point charges [95]; this has limited the use of error estimators for such models. Error estimators necessarily indicate large errors anywhere there are fixed charges, thus throughout the protein, not primarily at the interface. Indeed, the singularity due to the point charges is more severe than that caused by the jump in the dielectric coefficient ϵ.

But we can introduce a splitting u=v+w where

$$v(\mathbf{x}) = \sum_{i=1}^{N} \frac{c_i}{|\mathbf{x}-\mathbf{x}_i|}. \tag{11.9}$$

Here we assume that units chosen so that fundamental solution of $-\epsilon_0\Delta u = \delta_0$ is $1/|\mathbf{x}|$, where ϵ_0 is the dielectric constant in Ω.

11.2.1 Equation for w

By definition, w is harmonic in both Ω and $\mathbb{R}^3\backslash\Omega$, and $w(\mathbf{x}) \to 0$ as $\mathbf{x}\to\infty$. But the jump in the normal derivative of w across the interface $B = \partial\Omega$ is not zero. Define

$$\left[\epsilon\frac{\partial w}{\partial n}\right]_B = \epsilon_0\frac{\partial w}{\partial n}\big|_{B-} - \epsilon_\infty\frac{\partial w}{\partial n}\big|_{B+},$$

where $B-$ denotes the inside of the interface, $B+$ denotes the outside of the interface, and $\mathbf{n}$ denotes the outward normal to Ω. The solution u of (11.8) satisfies $\left[\epsilon\frac{\partial u}{\partial n}\right]_B = 0$, so

$$\left[\epsilon\frac{\partial w}{\partial n}\right]_B = (\epsilon_\infty - \epsilon_0)\frac{\partial v}{\partial n}\big|_B\,.$$

From the intergration-by-parts formula (2.8), we have

$$a(w,\phi) = \oint_B \left[\epsilon\frac{\partial w}{\partial n}\right]_B \phi\,ds = (\epsilon_\infty - \epsilon_0)\oint_B \frac{\partial v}{\partial n}\phi\,ds$$

for all test functions ϕ. The linear functional F defined by

$$F(\phi) = (\epsilon_\infty - \epsilon_0) \oint_B \frac{\partial v}{\partial n} \phi \, ds \tag{11.10}$$

is clearly well defined for any test function, since v is smooth except at the singular points x_i, which we assume are in the interior of Ω, not on the boundary $B = \partial\Omega$. Thus w is defined by a standard variational formulation and can be computed accordingly, modulo the need to truncate the infinite domain at some distance from Ω. For example, we can define

$$B_R = \left\{ \mathbf{x} \in \mathbb{R}^3 \ : \ |\mathbf{x}| < R \right\},$$

and define

$$a_R(\phi, \psi) = \int_{B_R} \epsilon \nabla\phi \cdot \nabla\psi \, d\mathbf{x},$$

and solve for $w_R \in H^1_0(B_R)$ such that

$$a_R(w_R, \psi) = F(\psi) \quad \forall \psi \in H^1_0(B_R), \tag{11.11}$$

where F is defined in (11.10). Then $w_R \to w$ as $R \to \infty$.

11.2.2 Point-charge example

Let us consider a single point charge at the origin of a spherical domain

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^3 \ : \ |\mathbf{x}| < R \right\}$$

of radius $R > 0$. Let ϵ_0 denote the dielectric constant in Ω and ϵ_∞ denote the dielectric constant in $\mathbb{R}^3 \backslash \Omega$. Then the solution to (11.8) is

$$u(\mathbf{x}) = \begin{cases} \frac{1}{|\mathbf{x}|} - \frac{c}{R} & |\mathbf{x}| \le R \\ \frac{1-c}{|\mathbf{x}|} & |\mathbf{x}| \ge R, \end{cases} \tag{11.12}$$

where

$$c = 1 - \frac{\epsilon_0}{\epsilon_\infty}.$$

The verification is as follows. In Ω, we have $\Delta u = \delta_{\mathbf{0}}$. In $\mathbb{R}^3 \backslash \Omega$, we have $\Delta u = 0$. At the interface $B = \partial\Omega = \{\mathbf{x} \in \mathbb{R}^3 \ : \ |\mathbf{x}| = R\}$,

$$\frac{\partial u}{\partial n}\Big|_{B-} = \frac{\partial u}{\partial r}(R-) = \frac{-1}{R^2}, \qquad \frac{\partial u}{\partial n}\Big|_{B+} = \frac{\partial u}{\partial r}(R+) = \frac{-(1-c)}{R^2},$$

where $B-$ denotes the inside of the interface and $B+$ denotes the outside of the interface. Thus the jump is given by

$$\left[\epsilon \frac{\partial u}{\partial n}\right]_B = \epsilon_0 \frac{\partial u}{\partial n}\Big|_{B-} - \epsilon_\infty \frac{\partial u}{\partial n}\Big|_{B+} = \frac{-\epsilon_0 + (1-c)\epsilon_\infty}{R^2} = 0.$$

In this case, $v(\mathbf{x}) = 1/|\mathbf{x}|$, so

$$w(\mathbf{x}) = -c\begin{cases} \frac{1}{R} & |\mathbf{x}| \leq R \\ \frac{1}{|\mathbf{x}|} & |\mathbf{x}| \geq R. \end{cases} \tag{11.13}$$

Thus if we solve numerically for w, we have a much smoother problem. But as $R \to 0$, w becomes more singular.

11.2.3 Error estimators for electrostatic models

Error estimators used in models in which the numerical techniques must resolve the point singularities resulting from point charges [95] necessarily indicate large errors anywhere there are fixed charges, thus throughout the protein, not primarily at the interface. Indeed, the authors of [95, 16] considered a simplified algorithm to estimate errors due to the protein-solvent interface in the second paper. Further, in the subsequent paper [53], the authors considered the splitting advocated here and studied the corresponding improved error estimator.

When using the solution splitting that we advocate, one would expect that the primary numerical error would occur at the protein-water interface, due primarily to the jump in the dielectric coefficients at this interface. Specific studies of error estimators and adaptivity for elliptic problems with discontinuous coefficients have been an active area of research for over a decade [141, 142, 55, 164, 46, 173, 52, 54, 105].

If the error is dominated by the jump in the solution gradient at the interface, it is possible that the refinement necessary to represent the protein boundary already forces sufficient accuracy near the interface. In any case, it may well be that simple error indicators [16] are sufficient to obtain a good estimate of the errors. However, it seems prudent to examine this issue in detail as it has the potential to benefit a large class of codes that currently do not have a good way of determining appropriate resolution. Moreover, there is a reasonable chance that significant efficiencies can be obtained with a state-of-the-art approach to mesh refinement and coarsening [43].

11.3 Mixed Methods

The name "mixed method" is applied to a variety of finite element methods which have more than one approximation space. Typically one or more of the spaces play the role of Lagrange multipliers which enforce constraints. The name and many of the original concepts for such methods originated in solid mechanics [11] where it was desirable to have a more accurate approximation of certain derivatives of the displacement. However, for the Stokes equations which govern viscous fluid flow, the natural Galerkin approximation is a mixed method.

One characteristic of mixed methods is that not all choices of finite element spaces will lead to convergent approximations. Standard approximability alone is insufficient to guarantee success. Thus significant care is required in using them.

We will focus on mixed methods in which there are two bilinear forms and two approximation spaces. There are two key conditions that lead to the success of a mixed method.

Both are in some sense coercivity conditions for the bilinear forms. One of these will look like a standard coercivity condition, while the other, often called the *inf-sup* condition, takes a new form.

A model for fluid flow in a porous medium occupying a domain Ω takes the form

$$-\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\left(\alpha_{ij}(\mathbf{x})\frac{\partial p}{\partial x_j}(\mathbf{x})\right) = f(\mathbf{x}) \text{ in } \Omega, \tag{11.14}$$

where p is the pressure (we take an inhomogeneous right-hand-side for simplicity). Darcy's Law postulates that the fluid velocity $\mathbf{u}$ is related to the gradient of p by

$$u_i(\mathbf{x}) = -\sum_{j=1}^{d} \alpha_{ij}(\mathbf{x})\frac{\partial p}{\partial x_j}(\mathbf{x}) \quad \forall i = 1, \ldots, d. \tag{11.15}$$

The coefficients α_{ij}, which we assume form a symmetric, positive-definite matrix (almost everywhere–frequently the coefficients are discontinuous so there are submanifolds where they are ill defined), are related to the porosity of the medium. Of course, numerous other physical models also take the form (11.14), as in Section 11.2.

Combining Darcy's Law (11.15) and (11.14), we find $\nabla\cdot \mathbf{u} = f$ in Ω. A variational formulation for (11.14) can be derived by letting $\mathbf{A}(\mathbf{x})$ denote the (almost everywhere defined) inverse of the coefficient matrix (α_{ij}) and by writing $\nabla p = -\mathbf{A}\mathbf{u}$. Define

$$a(\mathbf{u},\mathbf{v}) := \sum_{i,j=1}^{d} \int_\Omega A_{ij}(\mathbf{x}) u_i(\mathbf{x}) v_j(\mathbf{x})\, d\mathbf{x}. \tag{11.16}$$

Then the solution to (11.14) solves

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) &= 0 \quad \forall \mathbf{v} \in \mathbf{V} \\ b(\mathbf{u},q) &= F(q) \quad \forall q \in \Pi\,, \end{aligned} \tag{11.17}$$

where $F(q) = -\int_\Omega f(\mathbf{x})\, q(\mathbf{x})\, d\mathbf{x}$, $\Pi = L^2(\Omega)$,

$$b(\mathbf{w},q) = \int_\Omega \mathbf{w}(\mathbf{x})\cdot\nabla q(\mathbf{x})\, d\mathbf{x} = -\int_\Omega \nabla\cdot \mathbf{w}(\mathbf{x})\, q(\mathbf{x})\, d\mathbf{x} + \oint_{\partial\Omega} q(\mathbf{x})\, \mathbf{w}(\mathbf{x})\cdot\mathbf{n}(\mathbf{x})\, d\mathbf{x}, \tag{11.18}$$

and we have a new space $\mathbf{V}$ defined by

$$\mathbf{V} := \left\{\mathbf{v} \in L^2(\Omega)^d \;:\; \nabla\cdot \mathbf{v} \in L^2(\Omega),\ \mathbf{v}\cdot\mathbf{n} = 0 \text{ on } \partial\Omega\right\}.$$

The space $\mathbf{V}$ is based on the space called $H(\mathrm{div};\Omega)$ [1] that has a natural norm given by

$$\|\mathbf{v}\|^2_{H(\mathrm{div};\Omega)} = \|\mathbf{v}\|^2_{L^2(\Omega)^d} + \|\nabla\cdot \mathbf{v}\|^2_{L^2(\Omega)}\,; \tag{11.19}$$

$H(\mathrm{div};\Omega)$ is a Hilbert space with inner-product given by

$$(\mathbf{u},\mathbf{v})_{H(\mathrm{div};\Omega)} = (\mathbf{u},\mathbf{v})_{L^2(\Omega)^d} + (\nabla\cdot \mathbf{u}, \nabla\cdot \mathbf{v})_{L^2(\Omega)}.$$

Thus we can write $\mathbf{V} := \{\mathbf{v} \in H(\mathrm{div};\Omega) \ : \ \mathbf{v}\cdot\mathbf{n} = 0 \text{ on } \partial\Omega\}$. The meaning of the boundary condition $\mathbf{v}\cdot\mathbf{n} = 0$ on $\partial\Omega$ can be made precise [1], but the tangential derivatives of a general function $\mathbf{v} \in H(\mathrm{div};\Omega)$ are not well defined.

The integration by parts in (11.18) follows from the divergence theorem. The bilinear form $a(\cdot,\cdot)$ is not coercive on all of $\mathbf{V}$, but it is coercive on the subspace $\mathbf{Z}$ of divergence-zero functions, since on this subspace the inner-product $(\cdot,\cdot)_{H(\mathrm{div};\Omega)}$ is the same as the $L^2(\Omega)$ inner-product. In particular, this proves uniqueness of solutions. Suppose that F is zero. Then $\mathbf{u} \in \mathbf{Z}$ and $a(\mathbf{u},\mathbf{u}) = 0$. Thus $\|\mathbf{u}\|_{L^2(\Omega)} = 0$, that is, $\mathbf{u} \equiv \mathbf{0}$. Existence and stability of solutions follows from the inf-sup condition: there is a finite, positive constant C such that for all $q \in L^2(\Omega)$

$$\begin{aligned} \|q\|_{L^2(\Omega)} &\le C \sup_{0\neq\mathbf{v}\in H^1(\Omega)} \frac{b(\mathbf{v},q)}{\|\mathbf{v}\|_{H^1(\Omega)}} \\ &\le C' \sup_{0\neq\mathbf{v}\in H(\mathrm{div};\Omega)} \frac{b(\mathbf{v},q)}{\|\mathbf{v}\|_{H(\mathrm{div};\Omega)}} \end{aligned} \tag{11.20}$$

where the first inequality is proved just like (12.17) (the only difference being the boundary conditions on $\mathbf{v}$) and the second follows from the inclusion $H^1(\Omega)^d \subset H(\mathrm{div};\Omega)$.

11.4 Discrete Mixed Formulation

Now let $V_h \subset V$ and $\Pi_h \subset \Pi$ and consider the variational problem to find $u_h \in V_h$ and $p_h \in \Pi_h$ such that

$$\begin{aligned} a(u_h,v) + b(v,p_h) &= F(v) \quad \forall v \in V_h\,, \\ b(u_h,q) &= 0 \quad \forall q \in \Pi_h\,. \end{aligned} \tag{11.21}$$

The case of an inhomogeneous right-hand-side in the second equation is considered in [40, Section 10.5] and in [156].

One family of spaces that can be used in certain mixed methods is the Taylor-Hood family consisting of the following spaces defined for $k \ge 2$. Let the space W_h^k denote the space of continuous piecewise polynomials of degree k (with no boundary conditions imposed). Let the space V_h be defined by

$$V_h = \left\{ v \in W_h^k \times W_h^k \ : \ v = 0 \text{ on } \partial\Omega \right\}. \tag{11.22}$$

and the space Π_h be defined by

$$\Pi_h = \left\{ q \in W_h^{k-1} \ : \ \int_\Omega q(\mathbf{x})\,d\mathbf{x} = 0 \right\}. \tag{11.23}$$

Then for $0 \le m \le k$

$$\|\mathbf{u} - \mathbf{u}_h\|_{H^1(\Omega)^2} \le Ch^m \left(\|\mathbf{u}\|_{H^{m+1}(\Omega)^2} + \|p\|_{H^m(\Omega)} \right)$$

To approximate the scalar elliptic problem (11.14) by a mixed method, we have to contend with the fact that the corresponding form $a(\cdot,\cdot)$ is not coercive on all of $\mathbf{V}$. It is clearly coercive on the space

$$Z = \{\mathbf{v} \in H(\mathrm{div};\Omega) \ : \ \nabla\cdot \mathbf{v} = 0\}$$

so that (11.17) is well-posed. However, some care is required to assure that it is well-posed as well on

$$Z_h = \{\mathbf{v} \in \mathbf{V}_h \ : \ b(\mathbf{v}, q) = 0 \quad \forall q \in \Pi_h\}\,.$$

One simple solution is to insure that $Z_h \subset Z$ and we will present one way this can be done.

Let $\mathbf{V}_h$ be as given in (11.22) and let $\Pi_h = \nabla\cdot \mathbf{V}_h$. Suppose that $k \geq 4$. Then under certain mild restrictions on the mesh [156] these spaces can be used. There are algorithms [156] that allow one to compute using $\Pi_h = \mathcal{D}V_h$ without having explicit information about the structure of Π_h as described in Section 12.6. For more information on mixed methods, see Section 12.3.

11.5 Numerical quadrature

The evaluation of forms having variable coefficients requires some sort of numerical quadrature. This presents an additional degree of approximation and potential cause for error.

11.6 Exercises

Exercise 11.1 *Use the variational formulation (11.11) to approximate the solution of (11.8) using the splitting $u = v + w$ where v is defined in (11.9). Use the example in Section 11.2.2 as a test case. Consider the impact of the approximation parameter R (see if the results stablize as $R \to \infty$), as well as the choice of mesh and approximation space, on the results. Choose an appropriate norm in which to measure the error.*

Exercise 11.2 *Use the variational formulation (11.11) to approximate the solution of (11.8) directly via the approximation (4.20) to the Dirac δ-function developed in Section 4.2. That is, define the linear functional F^A ($A > 0$) by*

$$F^A(v) = \sum_{i=1}^{N} c_i \int_\Omega \delta^A_{\mathbf{x}_i}(\mathbf{x})\, v(\mathbf{x})\, d\mathbf{x},$$

and solve for $u^{A,R} \in H^1_0(B_R)$ satisfying

$$a_R(u^{A,R}, v) = F^A(v) \quad \forall v \in H^1_0(B_R).$$

Use the example in Section 11.2.2 as a test case. Consider the impact of the approximation parameters A and R (see if the results stablize as $A, R \to \infty$), as well as the choice of mesh and approximation space, on the results. Choose an appropriate norm in which to measure the error. Compare with the approach in Exercise 11.1.

Exercise 11.3 *Carry out all of the details in the derivation of the variational formulation of the porous medium equation (11.14). In particular, answer the questions*

- *How does the first equation in (11.17) ensure that* $\nabla p = -\mathbf{A}\mathbf{u}$*?*
- *Why does the form* $b(\mathbf{w}, q) = \int_\Omega \mathbf{w}(\mathbf{x}) \cdot \nabla q(\mathbf{x})\, d\mathbf{x}$ *yield* $\nabla\cdot \mathbf{u} = f$ *via the second equation in (11.17)?*

Exercise 11.4 *Prove that for any domain* $\Omega \subset \mathbb{R}^d$ *and any* $\mathbf{v} \in H^1(\Omega)^d$*,* $d = 2$ *or* 3*,*

$$\|\mathbf{v}\|_{H(\mathrm{div};\Omega)} \leq \sqrt{d}\, \|\mathbf{v}\|_{H^1(\Omega)}.$$

Chapter 12

Stokes' Equations

We now see models in which the unknown functions are vector valued. This does not in itself present any major change to our variational formulation. However, for incompressible fluids, a significant computational new feature emerges. This forces attention on the numerical methods used to be sure of having valid simulations.

The model equations for all fluids take the form

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \nabla\cdot \mathbf{T} + \mathbf{f},$$

where $\mathbf{u}$ is the velocity of the fluid, p is the pressure, $\mathbf{T}$ is called the extra (or deviatoric) stress and $\mathbf{f}$ is externally given data. The models differ based on the way the stress $\mathbf{T}$ depends on the velocity $\mathbf{u}$. Time-independent models take the form

$$\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \nabla\cdot \mathbf{T} + \mathbf{f}. \tag{12.1}$$

For incompressible fluids, the equation (12.1) is accompanied by the condition

$$\nabla\cdot \mathbf{u} = 0, \tag{12.2}$$

which we will assume holds in the following discussion. For suitable expressions for $\mathbf{T}$ defined in terms of $\mathbf{u}$, the problem (12.1) and (12.2) can be shown to be well-posed, as we indicate in special cases.

The simplest expression for the stress is linear: $\mathbf{T} = \frac{1}{2}\eta\big(\nabla \mathbf{u} + \nabla \mathbf{u}^t\big)$, where η denotes the viscosity of the fluid. Such fluids are called Newtonian. Scaling by η, (12.1) becomes

$$\frac{1}{\eta}\mathbf{u} \cdot \nabla \mathbf{u} + \nabla \hat{p} - \Delta \mathbf{u} = \hat{\mathbf{f}}, \tag{12.3}$$

where $\hat{p} = (1/\eta)p$ and $\hat{\mathbf{f}} = (1/\eta)\mathbf{f}$. When η is large, the nonlinear term multiplied by η^{-1} is often dropped, resulting in a linear system called the **Stokes equations** when (12.2) is added. When the nonlinear equation is kept, equations (12.3) and (12.2) are called the **Navier-Stokes equations**, which we consider in Chapter 14.

The Stokes equations for the flow of a viscous, incompressible, Newtonian fluid can thus be written

$$\begin{aligned} -\Delta\mathbf{u} + \nabla p &= \mathbf{0} \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{12.4}$$

in a domain $\Omega \subset \mathbb{R}^d$, where $\mathbf{u}$ denotes the fluid velocity and p denotes the pressure [112]. These equations must be supplemented by appropriate boundary conditions, such as the Dirichlet boundary conditions, $\mathbf{u} = \boldsymbol{\gamma}$ on $\partial\Omega$. The key compatibility condition on the data comes from the divergence theorem:

$$\oint_{\partial\Omega} \boldsymbol{\gamma}\cdot\mathbf{n}\,ds = 0. \tag{12.5}$$

12.1 Stokes variational formulation

The variational formulation of (12.4) takes the form: Find $\mathbf{u}$ such that $\mathbf{u}-\boldsymbol{\gamma} \in \mathbf{V}$ and $p \in \Pi$ such that

$$\begin{aligned} a\left(\mathbf{u},\mathbf{v}\right) + b\left(\mathbf{v},p\right) &= 0 \quad \forall\mathbf{v}\in\mathbf{V}\,, \\ b(\mathbf{u},q) &= 0 \quad \forall q\in\Pi\,, \end{aligned} \tag{12.6}$$

where, e.g., $a(\cdot,\cdot) = a_\nabla(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are given by

$$a_\nabla(\mathbf{u},\mathbf{v}) := \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\,d\mathbf{x} = \int_\Omega \sum_{i,j=1}^d u_{i,j}v_{i,j}\,d\mathbf{x}, \tag{12.7}$$

$$b(\mathbf{v},q) := -\int_\Omega \sum_{i=1}^d v_{i,i}q\,d\mathbf{x}. \tag{12.8}$$

This is derived by multiplying (12.4) by $\mathbf{v}$ with a "dot" product, and integrating by parts as usual. Note that the second equation in (12.4) and (12.6) are related by multiplying the former by q and integrating, with no integration by parts.

The spaces $\mathbf{V}$ and Π are as follows. In the case of simple Dirichlet data on the entire boundary, $\mathbf{V}$ consists of the d-fold Cartesian product of the subset of $H^1(\Omega)$ of functions vanishing on the boundary. In this case, Π is the subset of $L^2(\Omega)$ of functions having mean zero. The latter constraint corresponds to fixing an ambient pressure.

Another variational formulation for (12.4) can be derived which is equivalent in some ways, but not identical to (12.6). Define

$$\epsilon(\mathbf{u})_{ij} = \tfrac{1}{2}\left(u_{i,j} + u_{j,i}\right) \tag{12.9}$$

and

$$a_\epsilon(\mathbf{u},\mathbf{v}) := 2\int_\Omega \sum_{i,j=1}^d \epsilon(\mathbf{u})_{ij}\epsilon(\mathbf{v})_{ij}\,d\mathbf{x}. \tag{12.10}$$

Then it can be shown that

$$a_\epsilon(\mathbf{u},\mathbf{v}) := a_\nabla(\mathbf{u},\mathbf{v}) \tag{12.11}$$

provided only that $\nabla\cdot\mathbf{u}=\mathbf{0}$ in Ω and $\mathbf{v}=\mathbf{0}$ on $\partial\Omega$ or $\nabla\cdot\mathbf{v}=0$ in Ω and $\mathbf{u}=\mathbf{0}$ on $\partial\Omega$. However, the natural boundary conditions associated with a_ϵ and a_∇ are quite different [90].

Inhomogeneous Dirichlet data can be incorporated in a standard variational problem much like the scalar case (Exercise 2.4 and Section 3.1.5) but with a twist. The variational formulation (12.6) can be written with $\mathbf{u}=\mathbf{u}_0+\boldsymbol{\gamma}$ where $\mathbf{u}_0\in\mathbf{V}$ and $p\in\Pi$ satisfy

$$\begin{aligned} a\left(\mathbf{u},\mathbf{v}\right)+b\left(\mathbf{v},p\right) &= -a(\boldsymbol{\gamma},\mathbf{v}) \quad \forall\mathbf{v}\in\mathbf{V}\,,\\ b(\mathbf{u}_0,q) &= -b(\boldsymbol{\gamma},q) \quad \forall q\in\Pi\,. \end{aligned} \tag{12.12}$$

The twist is that the second equation becomes inhomogeneous as well, unless by chance $\boldsymbol{\gamma}$ is divergence free.

12.2 Well-posedness of Stokes

We assume, as always, that the bilinear forms satisfy the continuity conditions

$$\begin{aligned} |a(\mathbf{v},\mathbf{w})| &\le C_a\|\mathbf{v}\|_{\mathbf{V}}\|\mathbf{w}\|_{\mathbf{V}} \quad \forall\mathbf{v},\mathbf{w}\in\mathbf{V}\\ |b(\mathbf{v},q)| &\le C_b\|\mathbf{v}\|_{\mathbf{V}}\|q\|_{\Pi} \quad \forall\mathbf{v}\in\mathbf{V},\, q\in\Pi \end{aligned} \tag{12.13}$$

for finite, positive constants C_a and C_b. This is easily proven for the Stokes problem. However, the Lax-Milgram theory does not suffice to establish the well-posedness of the Stokes equations due in part to the asymmetry of the equations for the variables $\mathbf{u}$ and p. Indeed, the second equation in (12.6) may be thought of as a simple constraint: $\nabla\cdot\mathbf{u}=0$. Thus it is natural to consider the space $\mathbf{Z}$ defined by

$$\mathbf{Z}=\{\mathbf{v}\in\mathbf{V}\;:\; b(\mathbf{v},q)=0 \quad \forall q\in\Pi\}=\{\mathbf{v}\in\mathbf{V}\;:\;\nabla\cdot\mathbf{v}=0\}\,. \tag{12.14}$$

We may simplify the variational formulation (12.6) to: find $\mathbf{u}\in\mathbf{Z}$ such that

$$a(\mathbf{u},\mathbf{v})=F(\mathbf{v}) \quad \forall\mathbf{v}\in\mathbf{Z}, \tag{12.15}$$

which is a standard variational formulation. Thus the problem for $\mathbf{u}$ is wellposed if we assume that $a(\cdot,\cdot)$ is coercive on $\mathbf{Z}$:

$$\|\mathbf{v}\|_{H^1(\Omega)}^2 \le c_0\, a(\mathbf{v},\mathbf{v}) \quad \forall\mathbf{v}\in\mathbf{Z} \tag{12.16}$$

for some finite, positive constant c_0. We will see that this framework is useful for other, so-called **mixed formulations** as well.

The bound (12.16) can be proved for both of the forms a_∇ and a_ϵ with suitable boundary conditions, for all of $\mathbf{V}$ as well. The proof is familar in the case of a_∇, since the only functions for which $a_\nabla(\mathbf{v},\mathbf{v})=0$ are constants. The coercivity of a_ϵ is called Korn's inequality [40].

Well-posedness for the pressure follows from the inf-sup condition

$$\|q\|_{L^2(\Omega)} \le C \sup_{0\neq \mathbf{v}\in \mathbf{V}} \frac{b(\mathbf{v},q)}{\|\mathbf{v}\|_{H^1(\Omega)}}, \quad \forall q\in\Pi, \tag{12.17}$$

which is proved [9] by solving $\nabla\cdot\mathbf{v}=q$ with $\mathbf{v}\in\mathbf{V}$ and $\|\mathbf{v}\|_{H^1}\le C\|q\|_{L^2}$. The term inf-sup encapsulates the fact that we assume that (12.17) holds for all $q\in\Pi$, so that

$$\frac{1}{C}\le \inf_{q\in\Pi} \sup_{0\neq\mathbf{v}\in H^1(\Omega)} \frac{b(\mathbf{v},q)}{\|\mathbf{v}\|_{H^1(\Omega)}}.$$

Suppose that $F=0$. Then (12.15) implies that $\mathbf{u}=0$, and so from (12.6) we have

$$b(\mathbf{v},p)=0 \quad \forall\mathbf{v}\in\mathbf{V}.$$

It follows from (12.17) that $p=0$. Thus coercivity of $a(\cdot,\cdot)$ and the inf-sup condition (12.17) guarantee uniqueness of the solution of (12.6). It can similarly be shown that they imply existence and stability [40].

12.3 Mixed Method Formulation

The general formulation of the discretization (12.6) is of the form

$$\begin{aligned} a(u_h,v)+b(v,p_h) &= F(v) \quad \forall v\in V_h\\ b(u_h,q) &= G(q) \quad \forall q\in\Pi_h, \end{aligned} \tag{12.18}$$

where $F\in V'$ and $G\in\Pi'$ (the "primes" indicate dual spaces [40]). It is called a "mixed method" since the variables v and q are mixed together. For the Stokes problem (12.6), the natural variational formulation is already a mixed method, whereas it is an optional formulation in other settings (see Section 11.3).

In the discrete mixed method, V and Π are two Hilbert spaces with subspaces $V_h\subset V$ and $\Pi_h\subset\Pi$, respectively. The main twist in the variational formulation of mixed methods with inhomogeneous boundary conditions is that the term G is not zero [156].

We will assume there is a continuous operator $\mathcal{D}:V\to\Pi$ such that

$$b(v,p)=(\mathcal{D}v,p)_\Pi \quad \forall p. \tag{12.19}$$

In the Stokes problem, $\mathcal{D}=\nabla\cdot$. Let P_Π denote the Riesz representation of G in Π_h, that is

$$(P_\Pi\, G,q)_\Pi = G(q) \quad \forall q\in\Pi_h\,. \tag{12.20}$$

Note that the second equation in (12.18) says that

$$P_\Pi\,\mathcal{D}u_h = P_\Pi\, G \tag{12.21}$$

where we also use $P_\Pi\, g$ to denote the Π-projection of $g\in\Pi$ onto Π_h.

We assume that the bilinear forms satisfy the continuity conditions (12.13) and the coercivity conditions

$$\begin{aligned} \alpha\|v\|_V^2 &\le a(v,v) \quad \forall v \in Z \cup Z_h \\ \beta\|p\|_\Pi &\le \sup_{v\in V_h} \frac{b(v,p)}{\|v\|_V} \quad \forall p \in \Pi_h. \end{aligned} \tag{12.22}$$

Here Z and Z_h are defined by

$$Z = \{v \in V \ : \ b(v,q) = 0 \quad \forall q \in \Pi\} \tag{12.23}$$

and

$$Z_h = \{v \in V_h \ : \ b(v,q) = 0 \quad \forall q \in \Pi_h\} \tag{12.24}$$

respectively.

The mixed formulation can be posed in the canonical variational form (6.10) by writing

$$\begin{aligned} \mathcal{A}((u,p),(v,q)) :=& a(u,v) + b(v,p) + b(u,q) \\ \mathcal{F}((v,q)) :=& G(q) + F(v) \end{aligned} \tag{12.25}$$

for all $(v,q) \in \mathcal{V} := V \times \Pi$. This can be solved by direct methods (Gaussian elimination). However, other algorithms can be used in special cases, as we discuss in Section 12.6.

12.4 Taylor-Hood method

The first widely-used spaces used for the Stokes equations (12.6) were the so-called Taylor-Hood spaces, as follows. Let V_h^k denote C^0 piecewise polynomials of degree k on a non-degenerate triangulation of a polygonal domain $\Omega \subset \mathbb{R}^d$ of maximum element diameter h. Let

$$\mathbf{V}_h = \left\{ \mathbf{v} \in \left(V_h^k\right)^d \ : \ \mathbf{v} = 0 \text{ on } \partial\Omega \right\} \tag{12.26}$$

and let

$$\Pi_h = \left\{ q \in V_h^{k-1} \ : \ \int_\Omega q(x)\,d\mathbf{x} = 0 \right\}. \tag{12.27}$$

It can be proved that (12.22) holds in both two and three dimensions under very mild restrictions on the mesh [28].

Note that $Z_h \not\subset Z$ for the Taylor-Hood method. The one drawback of Taylor-Hood is that the divergence free condition can be substantially violated, leading to a loss of mass conservation [113]. This can be avoided if we force the divergence constraint to be satisfied by a penalty method. Another difficulty with the Taylor-Hood method is that it is necessary to construct the constrained space Π_h in (12.27). Many systems, including `dolfin`, do not provide a simple way to construct such a subspace of a space like V_h^{k-1}. Thus special linear algebra must be performed. Finally, we will see that it is possible to avoid Π_h completely, leading to smaller systems (see Figure 12.1) [132].

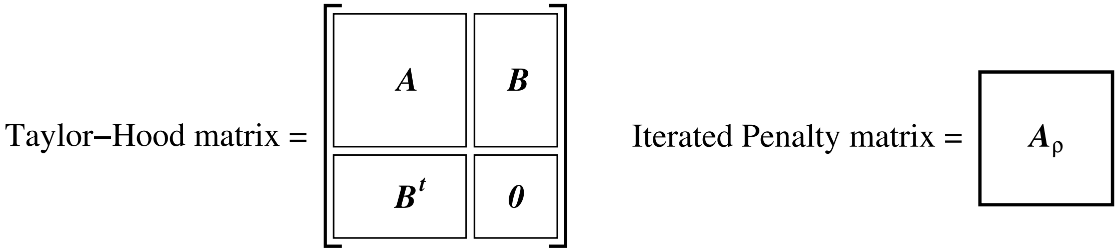

Figure 12.1: Comparison of the matrix sizes for Taylor-Hood and the Iterated Penalty Method. The full matrix for Taylor-Hood includes blocks $\mathbf{B}$ and $\mathbf{B}^t$ corresponding to the variational form $b(v, q)$ for $v \in V_h$ and $q \in \Pi_h$. The iterated penalty method involves a matrix just the size of $\mathbf{A}$, the matrix corresponding to the variational form $a(v, w)$ for $v, w \in V_h$.

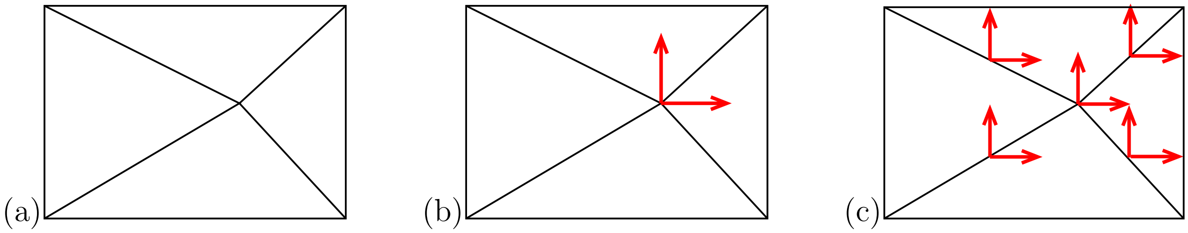

Figure 12.2: (a) A triangulation with only one interior vertex; (b) degrees of freedom for piecewise linear vector functions on this mesh that vanish on the boundary; (c) degrees of freedom for piecewise quadratic vector functions on this mesh that vanish on the boundary.

12.5 Constraint counting

The discrete version of the Stokes equations can be written as

$$a(\mathbf{u}_h, \mathbf{v}) = F(\mathbf{v}) \quad \forall \mathbf{v} \in Z_h.$$

From Céa's theorem, the error $\mathbf{u} - \mathbf{u}_h$ is bounded by the best approximation from Z_h. So it might appear that the pressure and its approximation play no significant role, with the only issue being to see that the space Z_h is not over-constrained. But in fact it can be. When we choose the pressure space, we are choosing something to constrain the divergence of the velocity space. This is the main issue with the linkage between the pressure approximation space and the velocity approximation space.

To understand this, take V_k to be vector Lagrange elements of degree k in two dimensions. The divergence of such (vector) elements form a subspace of discontinuous finite elements of degree $k-1$, which we denote by W_{k-1}. So we can begin by understanding how these two spaces, $\nabla\cdot V_k$ and W_{k-1}, are related. Let us simply count the number of degrees of freedom for the mesh depicted in Figure 12.2.

There are only two degrees of freedom, as indicated in Figure 12.2(b), for piecewise linear vector functions on this mesh that vanish on the boundary. That is, $\dim V_h = 2$. If we take Π_h to be piecewise constants on this mesh which have mean zero, then $\dim \Pi_h = 3$ (one for each triangle, minus 1 for the mean-zero constraint.) Thus we see that the inf-sup condition

(12.22) cannot hold; V_h is only two-dimensional, so the three-dimensional space Π_h must have a p orthogonal to $\nabla\cdot \mathbf{v}$ for all $\mathbf{v} \in V_h$. Moreover, it is not hard to see that $Z_h = \{\mathbf{0}\}$, the space with only the zero function.

For V_h being piecewise quadratic vector functions on the mesh in Figure 12.2(a), the dimension of V_h is 10, as shown in Figure 12.2(c); there are 4 edge nodes and 1 vertex node, and 2 degrees of freedom for each. If we take Π_h to be discontinuous piecewise linears on this mesh which have mean zero, then $\dim \Pi_h = 11$ (three for each triangle, minus 1 for the mean-zero constraint.) Thus we see that V_h is still too small to match Π_h, just by dimensional analysis.

One way to resolve this dilemma is to reduce the number of constraints implied by $b(\mathbf{v}, q) = 0$. This could be done by making the pressure space smaller, or (equivalently as it turns out) reducing the accuracy of integration in computing $b(\mathbf{v}, q)$. Such reduced or selective integration has been extensively studied [122].

12.5.1 Higher-degree approximation

Another way to eliminate the limiting behavior is to go to higher degree polynomials. For V_h being piecewise quartic vector functions on the mesh in Figure 12.2(a), $\dim V_h = 50$ (there are 3 nodes per edge, 3 nodes per triangle and one vertex node). Correspondingly, if Π_h consists of discontinuous piecewise cubics with mean zero, then $\dim \Pi_h = 39$ (10 degrees of freedom per triangle and one mean-zero constraint). Thus we do have $\dim V_h >> \dim \Pi_h$ in this case. However, the counting of constraints has to be more careful. The divergence operator maps V_h to Π_h, with its image being W_h, and its kernel is Z_h. So $\dim V_h = \dim W_h + \dim Z_h$. We hope that $W_h = \Pi_h$, which is true if $\dim W_h = \dim \Pi_h = 39$. Thus we need to show that $\dim Z_h = 11$ in this case.

We can write $Z_h = \operatorname{curl} S_h$ where S_h is the space of C^1 (scalar) piecewise quintic functions on the mesh in Figure 12.2(a). One can check that, indeed, this space is specified uniquely by 11 parameters [133]. Moreover, under some mesh restrictions, it can be shown [158, 157] that the inf-sup condition (12.22) holds with $\beta > 0$ independent of the mesh size.

When $\Pi_h = \nabla\cdot V_h$, the resulting algorithm is often called the Scott-Vogelius method [45, 50, 114, 115, 157, 158]. In Section 12.6, we consider an algorithm for solving for the velocity and pressure without dealing explicitly with the pressure space Π_h, facilitating the choice $\Pi_h = \nabla\cdot V_h$.

12.5.2 Malkus crossed-quadrilaterals

David Malkus realized that there was something special about triangulations involving crossed-quadrilaterals, that is quadrilaterals containing four triangles obtained by drawing the two diagonals of the quadrilateral, as shown in Figure 12.3(a). He referred to such elements as crossed triangles [121]. The vertices where the diagonals cross came to be known as **singular vertices** [133, 134]. At such vertices, the characterization of $\nabla\cdot V_k$ changes. In addition to the fact that the average of $p \in \nabla\cdot V_k$ must be zero over the four elements, now there is a local condition constraining p at the singular vertex. Since p is discontinuous, let

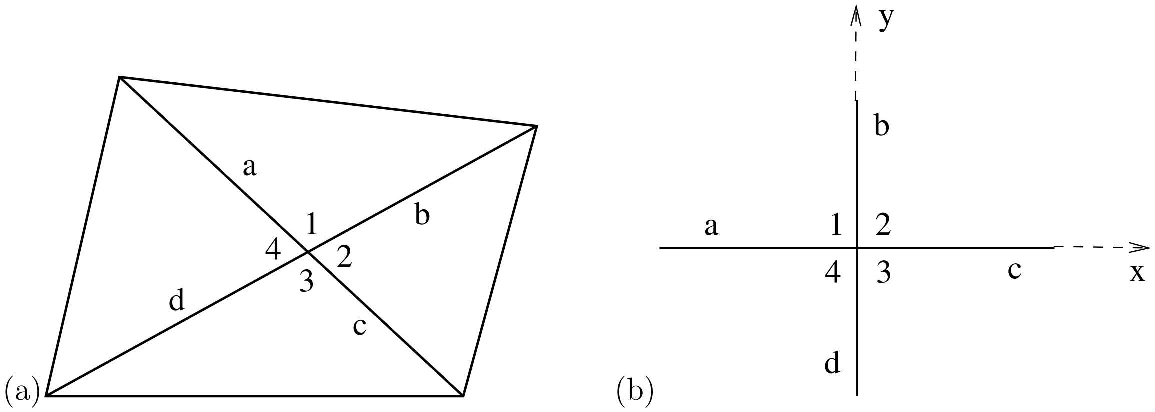

Figure 12.3: (a) A triangulation based on the diagonals of a quadrilateral in which opposite edges are parallel: $a\|c$, $b\|d$. (b) Notation for understanding local constraint at a singular vertex.

us denote the four values of p at the singular vertex σ by

$$\pi^1 = p_{ab}(\sigma), \quad \pi^2 = p_{bc}(\sigma), \quad \pi^3 = p_{cd}(\sigma), \quad \pi^4 = p_{da}(\sigma), \tag{12.28}$$

where π_{ij} denotes the restriction of p in the triangle with edges i,j at the central vertex. Then the singular vertex condition is that

$$\pi^1 - \pi^2 + \pi^3 - \pi^4 = 0. \tag{12.29}$$

To see why (12.29) must hold, it helps to consider a simple example in which the two diagonals are orthogonal, as indicated in Figure 12.3(b). In this case, we can assume without loss of generality that the edges lie on the x and y axes, as indicated in Figure 12.3(b). Let the velocity vector be denoted by (u,v). Then u is continuous on the edge a, and so $u_{,x}$ is also continuous on the edge a. Using notation similar to (12.28), we can say that $u_{ab,x}(\sigma) - u_{ad,x}(\sigma) = 0$. Similarly, $u_{bc,x}(\sigma) - u_{cd,x}(\sigma) = 0$. Subtracting the second expression from the one before gives

$$u_{ab,x}(\sigma) - u_{bc,x}(\sigma) + u_{cd,x}(\sigma) - u_{ad,x}(\sigma) = 0.$$

A similar argument can be applied to $v_{,y}$, and adding the two expressions yields (12.29). We leave an Exercise 12.3 to prove (12.29) in the general case when the diagonals are not perpendicular.

The import of the condition (12.29) is a reduction in the size of the space $\nabla\cdot V_k$. Moreover, it means that there are only two degrees of freedom in the target discontinuous piecewise constant subspace of W_0 in the case $k=1$. Thus the two degrees of freedom indicated in Figure 12.2(b) are just enough to match the corresponding pressure variables, leading to a well-posed numerical method. This method is thus another method where we can take $\Pi = \nabla\cdot V$. Moreover, an explicit basis of Π is available. In terms of the π variables, a basis consists of

$$(1,0,-1,0) \quad \text{and} \quad (1,1,-1,-1),$$

in each quadrilateral.

12.6 Iterated Penalty Method

Whenever one has a system of equations that can be written in terms of one equation holding on a constrained set of variables, an iterative technique can be used to solve them that may be quite efficient. The key is to turn the constraint into an operator that gets applied with some penalty in an iterative fashion until the constraint is satisfied to some tolerance.

Consider a general mixed method of the form (12.18). Let $\rho' \in \mathbb{R}$ and $\rho > 0$. The iterated penalty method defines $\mathbf{u}^n \in V_h$ and p^n by

$$\begin{aligned} a(\mathbf{u}^n,\mathbf{v}) + \rho'\,(\mathcal{D}\mathbf{u}^n,\mathcal{D}\mathbf{v})_\Pi &= F(\mathbf{v}) - b(\mathbf{v},p^n) + \rho' G(\mathcal{D}\mathbf{v}) \quad \forall \mathbf{v}\in V_h \\ p^{n+1} &= p^n + \rho\,(\mathcal{D}\mathbf{u}^n - P_\Pi\, G) \end{aligned} \tag{12.30}$$

where $P_\Pi\, G$ is defined in (12.20). Recall also that (12.21) says that $P_\Pi\,\mathcal{D}u_h = P_\Pi\, G$.

The algorithm does not require $P_\Pi\, G$ to be computed, only

$$b(\mathbf{v},P_\Pi\, G) = (\mathcal{D}\mathbf{v},P_\Pi\, G)_\Pi = G(P_\Pi\,\mathcal{D}\mathbf{v}) \tag{12.31}$$

for $\mathbf{v}\in V_h$. Suppose that $G(q) := -b(\boldsymbol{\gamma},q)$. Then

$$G(P_\Pi\,\mathcal{D}\mathbf{v}) = -b(\boldsymbol{\gamma},P_\Pi\,\mathcal{D}\mathbf{v}) = -(\mathcal{D}\boldsymbol{\gamma},P_\Pi\,\mathcal{D}\mathbf{v})_\Pi. \tag{12.32}$$

If further $\Pi_h = \mathcal{D}V_h$, then

$$b(\mathbf{v},P_\Pi\, G) = G(P_\Pi\,\mathcal{D}\mathbf{v}) = -(\mathcal{D}\boldsymbol{\gamma},\mathcal{D}\mathbf{v})_\Pi \tag{12.33}$$

for $\mathbf{v}\in V_h$.

One key point of the iterated penalty method is that the system of equations represented by the first equation in (12.30) for $\mathbf{u}^n$, namely

$$a(\mathbf{u}^n,\mathbf{v}) + \rho'\,(\mathcal{D}\mathbf{u}^n,\mathcal{D}\mathbf{v})_\Pi = F(\mathbf{v}) - b(\mathbf{v},p^n) + \rho'\, G(\mathcal{D}\mathbf{v}) \quad \forall \mathbf{v}\in V_h, \tag{12.34}$$

will be symmetric if is $a(\cdot,\cdot)$ is symmetric, and it will be positive definite if $a(\cdot,\cdot)$ is coercive and $\rho' > 0$.

Suppose that that $\Pi_h = \mathcal{D}V_h$. Then since $G(q) := -b(\boldsymbol{\gamma},q)$,

$$(P_\Pi\, G,q)_\Pi = G(q) = -b(\boldsymbol{\gamma},q) = -(\mathcal{D}\boldsymbol{\gamma},q)_\Pi = -(P_\Pi\,\mathcal{D}\boldsymbol{\gamma},q)_\Pi \quad \forall q\in\Pi_h\,, \tag{12.35}$$

that is, $P_\Pi\, G = -P_\Pi\,\mathcal{D}\boldsymbol{\gamma}$. Then

$$p^{n+1} = p^n + \rho P_\Pi\,\mathcal{D}\,(\mathbf{u}^n + \boldsymbol{\gamma}) \tag{12.36}$$

since $P_\Pi\,\mathcal{D}\mathbf{u}^n = \mathcal{D}\mathbf{u}^n$.

If we begin with $p^0 = 0$ then, for all $n > 0$,

$$p^n = \rho P_\Pi\,\mathcal{D}\sum_{i=0}^{n}\left(\mathbf{u}^i + \boldsymbol{\gamma}\right) = P_\Pi\,\mathcal{D}\mathbf{w}^n. \tag{12.37}$$

where

$$\mathbf{w}^n := \rho \sum_{i=0}^{n} \left(\mathbf{u}^i + \boldsymbol{\gamma}\right). \tag{12.38}$$

Note that

$$b(\mathbf{v}, p^n) = (\mathcal{D}\mathbf{v}, p^n)_\Pi = (\mathcal{D}\mathbf{v}, P_\Pi\, \mathcal{D}\mathbf{w}^n)_\Pi = (\mathcal{D}\mathbf{v}, \mathcal{D}\mathbf{w}^n)_\Pi \tag{12.39}$$

since $P_\Pi\, \mathcal{D}\mathbf{v} = \mathcal{D}\mathbf{v}$.

Thus, the iterated penalty method implicitly becomes

$$\begin{aligned} a(\mathbf{u}^n, \mathbf{v}) + \rho'\, (\mathcal{D}\mathbf{u}^n, \mathcal{D}\mathbf{v})_\Pi &= F(\mathbf{v}) - (\mathcal{D}\mathbf{v}, \mathcal{D}\mathbf{w}^n)_\Pi - \rho'(\mathcal{D}\mathbf{v}, \mathcal{D}\boldsymbol{\gamma})_\Pi \quad \forall \mathbf{v} \in V_h \\ \mathbf{w}^{n+1} &= \mathbf{w}^n + \rho\, (\mathbf{u}^n + \boldsymbol{\gamma}). \end{aligned} \tag{12.40}$$

In the case $\rho' = \rho$ this simplifies to

$$\begin{aligned} a(\mathbf{u}^n, \mathbf{v}) + \rho\, (\mathcal{D}\mathbf{u}^n, \mathcal{D}\mathbf{v})_\Pi &= F(\mathbf{v}) - (\mathcal{D}\mathbf{v}, \mathcal{D}(\mathbf{w}^n + \rho\boldsymbol{\gamma}))_\Pi \quad \forall \mathbf{v} \in V_h \\ \mathbf{w}^{n+1} &= \mathbf{w}^n + \rho\, (\mathbf{u}^n + \boldsymbol{\gamma}). \end{aligned} \tag{12.41}$$

Thus we see that the introduction of inhomogeneous boundary conditions does not lead to a dramatic change to the formulation of the iterated penalty method. The main difference is that the "pressure" term

$$p_h = P_\Pi\, \mathcal{D}\mathbf{w}_h, \tag{12.42}$$

where $\mathbf{w}_h := \mathbf{w}^n$ for the value of n at which the iteration is terminated, cannot be computed directly since $\mathbf{w}_h \notin V_h$. That is, $\mathcal{D}\mathbf{w}_h \notin \Pi_h$. On the other hand, p_h is not needed at all for the computation of u_h and can be computed only as needed. For example, in a time-stepping scheme it need not be computed every time step.

The "pressure" term can be calculated in various ways. It satisfies the system

$$(p_h, \mathcal{D}\mathbf{v})_\Pi = (\mathcal{D}\mathbf{w}_h, \mathcal{D}\mathbf{v})_\Pi \quad \forall \mathbf{v} \in V_h. \tag{12.43}$$

We can write $p_h = \mathcal{D}\mathbf{z}_h$ for various $\mathbf{z}_h \in V_h$, with $\mathbf{z}_h$ satisfying the under-determined system

$$(\mathcal{D}\mathbf{z}_h, \mathcal{D}\mathbf{v})_\Pi = (\mathcal{D}\mathbf{w}_h, \mathcal{D}\mathbf{v})_\Pi \quad \forall \mathbf{v} \in V_h. \tag{12.44}$$

Various techniques could be used to specify $\mathbf{z}_h$, but one simple one is to use the system (12.18), namely, to find $\mathbf{z}_h \in V_h$ and $q_h \in \Pi_h$ such that

$$\begin{aligned} a(\mathbf{z}_h, \mathbf{v}) + b(\mathbf{v}, q_h) &= 0 \quad \forall \mathbf{v} \in V_h \\ b(\mathbf{z}_h, q) &= (\mathcal{D}\mathbf{w}_h, q)_\Pi \quad \forall q \in \Pi_h \end{aligned} \tag{12.45}$$

which is (12.18) with $F \equiv 0$ and $G(q) := (\mathcal{D}\mathbf{w}_h, q)_\Pi$.

The iterated penalty method can be used to solve (12.45), yielding an algorithm of the form

$$\begin{aligned} a(\mathbf{z}^n, \mathbf{v}) + \rho\, (\mathcal{D}\mathbf{z}^n, \mathcal{D}\mathbf{v})_\Pi &= -\,(\mathcal{D}\mathbf{v}, \mathcal{D}(\boldsymbol{\zeta}^n - \mathbf{w}_h))_\Pi \quad \forall \mathbf{v} \in V_h \\ \boldsymbol{\zeta}^{n+1} &= \boldsymbol{\zeta}^n + \rho\, (\mathbf{z}^n - \mathbf{w}_h) \end{aligned} \tag{12.46}$$

in the case $\rho' = \rho$. This involves inverting the same algebraic system as in (12.41), so very little extra work or storage is involved.

12.7 Application to Stokes

The iterated penalty method (12.40) (with $\rho' = \rho$) for (12.4) takes the form

$$\begin{aligned} \tilde{a}\left(\mathbf{u}^{\ell,n},\mathbf{v}\right) + \rho\left(\nabla\cdot\mathbf{u}^{\ell,n},\nabla\cdot\mathbf{v}\right)_{L^2} &= -a\left(\boldsymbol{\gamma},\mathbf{v}\right) - \rho\left(\nabla\cdot\left(\mathbf{w}^{\ell,n}+\boldsymbol{\gamma}\right),\nabla\cdot\mathbf{v}\right)_{L^2} \quad \forall\mathbf{v}\in\mathbf{V} \\ \mathbf{w}^{\ell,n+1} &= \mathbf{w}^{\ell,n} - \rho\left(\mathbf{u}^{\ell,n}+\boldsymbol{\gamma}\right) \end{aligned} \tag{12.47}$$

where either $p^{\ell,0} = 0$ or (i.e. $\mathbf{w}^{\ell,0} = 0$) or $\mathbf{w}^{\ell,0} = \mathbf{w}^{\ell-1,N}$ where N is the final value of n at time-step $\ell - 1$. If for some reason $p^{\ell} = p^{\ell,n} = P_{\Pi_h}\nabla\cdot\mathbf{w}^{\ell,n}$ were desired, it could be computed separately. Note that the second equation in (12.47) is not a variational equation, just an ordinary one. This will occur frequently in this section.

For example, algorithm (12.41) could be used to compute

$$\begin{aligned} \tilde{a}\left(\mathbf{z}^n,\mathbf{v}\right) + \rho\left(\nabla\cdot\mathbf{z}^n,\nabla\cdot\mathbf{v}\right)_{L^2} &= -\rho\left(\nabla\cdot\left(\boldsymbol{\zeta}^n-\mathbf{w}^{\ell,n}\right),\nabla\cdot\mathbf{v}\right)_{L^2} \quad \forall\mathbf{v}\in\mathbf{V} \\ \boldsymbol{\zeta}^{n+1} &= \boldsymbol{\zeta}^n - \rho\left(\mathbf{z}^n-\mathbf{w}^{\ell,n}\right) \end{aligned} \tag{12.48}$$

starting with, say, $\boldsymbol{\zeta}^0 \equiv 0$. Then $\mathbf{z}^n$ will converge to $\mathbf{z}\in\mathbf{V}_h$ satisfying $\nabla\cdot\mathbf{z} = P_{\Pi_h}\nabla\cdot\mathbf{w}^{\ell,n} = p^{\ell,n}$. Note that (12.48) requires the same system to be solved as for computing $\mathbf{u}^{\ell,n}$ in (12.47), so very little extra code or data-storage is required.

The potential difficulty caused by having inhomogeneous boundary data can be seen for high-order finite elements. For simplicity, consider the two-dimensional case. Let W_h^k denote piecewise polynomials of degree k on a triangular mesh, and let V_h^k denote the subspace of W_h^k consisting of functions that vanish on the boundary. Let $\mathbf{V}_h = V_h^k \times V_h^k$, and let $\Pi_h = \nabla\cdot\mathbf{V}_h$. Then each $q\in\Pi_h$ is continuous at all boundary singular [40, Section 12.6] vertices, σ_i, in the mesh. On the other hand, inhomogeneous boundary conditions will require the introduction of some $\boldsymbol{\gamma}\in W_h^k\times W_h^k$. It is known [158] that $\nabla\cdot\left(W_h^k\times W_h^k\right)$ consists of all piecewise polynomials of degree $k-1$, a larger space than Π_h if boundary singular vertices are present in the mesh. On the other hand, if there are no boundary singular vertices, there is no need to form the projection since $\Pi_h = \left\{q\in\nabla\cdot\left(W_h^k\times W_h^k\right) \;:\; \int_\Omega q(x)\,dx = 0\right\}$ in this case.

12.8 Convergence and stopping criteria

The convergence properties of (12.30) follow from [40, Chapter 13].

Theorem 12.1 *Suppose that the forms (12.18) satisfy (12.13) and (12.22). for V_h and $\Pi_h = \mathcal{D}V_h$. Then the algorithm (12.30) converges for any $0 < \rho < 2\rho'$ for ρ' sufficiently large. For the choice $\rho = \rho'$, (12.30) converges geometrically with a rate given by*

$$C_a\left(\frac{1}{\beta}+\frac{C_a}{\alpha\beta}\right)^2 \Big/ \rho' .$$

The following stopping criterion follows from [40, Chapter 13].

Theorem 12.2 *Suppose that the forms (12.18) satisfy (12.13) and (12.22). for V_h and $\Pi_h = \mathcal{D}V_h$. Then the errors in algorithm (12.30) can be estimated by*

$$\|\mathbf{u}^n - \mathbf{u}_h\|_V \le \left(\frac{1}{\beta} + \frac{C_a}{\alpha\beta}\right) \|\mathcal{D}\mathbf{u}^n - P_\Pi\, G\|_\Pi$$

and

$$\|p^n - p_h\|_\Pi \le \left(\frac{C_a}{\beta} + \frac{C_a^2}{\alpha\beta} + \rho' C_b\right) \|\mathcal{D}\mathbf{u}^n - P_\Pi\, G\|_\Pi.$$

When $G(q) = -b(\boldsymbol{\gamma}, q)$, then $P_\Pi\, G = -P_\Pi\, \mathcal{D}\boldsymbol{\gamma}$ and since $\mathcal{D}\mathbf{u}^n \in \Pi_h$,

$$\begin{aligned} \|\mathcal{D}\mathbf{u}^n - P_\Pi\, G\|_\Pi &= \|P_\Pi\, \mathcal{D}\,(\mathbf{u}^n + \boldsymbol{\gamma})\|_\Pi \\ &\le \|\mathcal{D}\,(\mathbf{u}^n + \boldsymbol{\gamma})\|_\Pi, \end{aligned} \tag{12.49}$$

and the latter norm is easier to compute, avoiding the need to compute $P_\Pi\, G$. We formalize this observation in the following result.

Corollary 12.1 *Under the conditions of Theorem (12.2) the errors in algorithm (12.30) can be estimated by*

$$\|\mathbf{u}^n - \mathbf{u}_h\|_V \le \left(\frac{1}{\beta} + \frac{C_a}{\alpha\beta}\right) \|\mathcal{D}\,(\mathbf{u}^n + \boldsymbol{\gamma})\|_\Pi$$

and

$$\|p^n - p_h\|_\Pi \le \left(\frac{C_a}{\beta} + \frac{C_a^2}{\alpha\beta} + \rho' C_b\right) \|\mathcal{D}\,(\mathbf{u}^n + \boldsymbol{\gamma})\|_\Pi.$$

A code implementing the Iterated Penalty Method is given in Program 12.1.

12.9 Exercises

Exercise 12.1 *Prove that (12.11) holds provided only that $\nabla\cdot\mathbf{u} = 0$ in Ω and $\mathbf{v} = \mathbf{0}$ on $\partial\Omega$ or $\nabla\cdot\mathbf{v} = 0$ in Ω and $\mathbf{u} = \mathbf{0}$ on $\partial\Omega$. (Hint: first verify that*

$$\nabla\mathbf{u} : \nabla\mathbf{v} = 2\epsilon(\mathbf{u}) : \epsilon(\mathbf{v}) - \sum_{i,j=1}^{d} u_{i,j} v_{j,i} \tag{12.50}$$

and then integrate by parts.)

Exercise 12.2 *Define the linear functional F in (12.15) that corresponds to the nonhomogeneous Dirichlet problem in the variational formulation (12.6).*

Exercise 12.3 *Prove (12.29) in the general case when the diagonals are not perpendicular.*

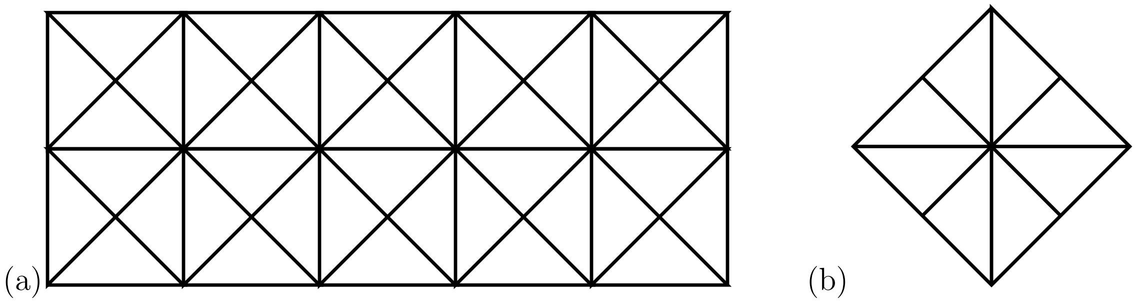

Figure 12.4: (a) A triangulation consisting of crossed quadrilaterals. (b) Support of additonal velocity test functions.

Exercise 12.4 *Consider a mesh consisting of crossed quadrilaterals, as shown in Figure 12.4(a). Let V_h denote piecewise linears on this mesh that vanish on the boundary. Let $\Pi_h = \nabla\cdot V_h$. Prove that Π_h consists of piecewise constants satisfying the constraint (12.29) in each quadrilateral together with the global mean-zero constraint. Show that each element of Π_h can be represented in each quadrilateral in the form of a sum of three basis functions*

$$P_1 = (1, 0, -1, 0) \quad \textit{and} \quad P_2 = (1, 1, -1, -1) \; \textit{and} \quad P_3 = (1, 1, 1, 1),$$

where the individual values of the four-vector are the values of the basis function in each triangle in a given quadrilateral. Note that the global mean-zero contstraint just involves the coefficients of the basis functions P_3 in each quadrilateral, namely that the sum of the coefficients is zero. Prove the $\inf - \sup$ *condition for this mesh for the pair V_h and Π_h. (Hint: In Section 12.5.2, we indicated how to construct $\mathbf{v} \in V_h$ such that $\mathbf{v}$ is supported in a given triangle and $\nabla\cdot \mathbf{v} = c_1 P_1 + c_2 P_2$. Thus we need only construct $\mathbf{v}$ to match the pressures that are piecewise constant on each quadrilateral. Consider piecewise linear velocities supported on the mesh indicated in Figure 12.4(b). Construct two different velocity functions whose divergence is as indicated in Figure 12.5. Use these functions to control a discrete form of the gradient of the pressure and modify the argument in [40, Section 12.6].)*

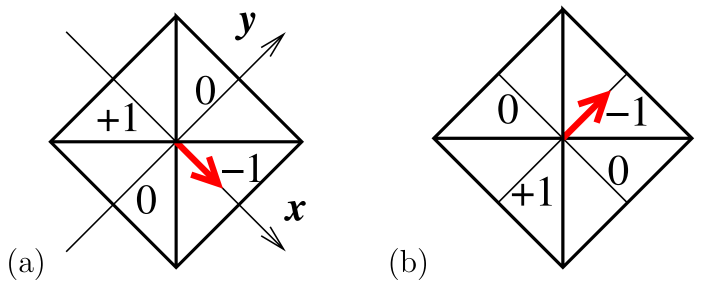

Figure 12.5: Values of divergence of piecewise linear vectors on the mesh indicated in Figure 12.4(a) for two different velocity test functions.

Exercise 12.5 *Consider the spaces introduced in Exercise 12.4, that is, let* V_h *be piecewise linear functions on a crossed mesh that vanish on the boundary, and Let* $\Pi_h = \nabla\cdot V_h$. *Use the iterated penalty method introduced in Section 12.6 to solve the Stokes equations with these spaces.*

Exercise 12.6 *The term* **spurious modes** *refers to pressure variables that are not properly controlled by the divergence of the velocity test functions. You can have good approximation in the velocity space and yet have spurious pressure modes. Consider the crossed meshes in Figure 12.4(a) and let* V_h *be piecewise linear functions on such a mesh that vanish on the boundary. Let* Π_h *be all piecewise constants that have global mean zero on this mesh. Prove that* Z_h *has good approximation properties. Also show that the spurious modes can be characterized as having the quartet of values given by the checkerboard pattern* $P_4 = (+1, -1, +1, -1)$. *(Hint: note that* Z_h *is the same as if we took* $\Pi_h = \nabla\cdot V_h$ *and apply the stability result of Exercise 12.4.)*

Exercise 12.7 *Consider the spaces introduced in Exercise 12.6, that is, let* V_h *be piecewise linear functions on a crossed mesh that vanish on the boundary. Let* Π_h *be all piecewise constants that have global mean zero on this mesh. Compute the solution of a smooth problem with these spaces. What happens?*

Exercise 12.8 *The Darcy-Stokes-Brinkman models [175] involve an operator of the form*

$$-\eta\Delta\mathbf{u} + \boldsymbol{\gamma}\mathbf{u} + \nabla p = f \tag{12.51}$$

together with the usual divergence constraint $\nabla\cdot\mathbf{u} = 0$ *and boundary conditions. Here* $\boldsymbol{\gamma}$ *can be a matrix function corresponding to the porosity in Darcy flow. When* η *is small, we get Darcy flow. When* $\boldsymbol{\gamma}$ *is small, we get Stokes flow. In a time stepping scheme for Stokes,* $\boldsymbol{\gamma} = (\Delta t)^{-1} I$, *where* I *is the identity matrix. The name Brinkman is associated with the general model. Experiment with this model using the Scott-Vogelius element, using the iterated penalty method to solve the linear system.*

```
from dolfin import *

meshsize = 16
mesh = UnitSquareMesh(meshsize, meshsize, "crossed")
k=4
V = VectorFunctionSpace(mesh, "Lagrange", k)
# define boundary condition
gee = Expression(("sin(4*pi*x[0])*cos(4*pi*x[1])", \
                          "-cos(4*pi*x[0])*sin(4*pi*x[1])"))
bc = DirichletBC(V, gee, "on_boundary")
# set the parameters
f = Expression(("28*pow(pi, 2)*sin(4*pi*x[0])*cos(4*pi*x[1])", \
                "-36*pow(pi, 2)*cos(4*pi*x[0])*sin(4*pi*x[1])"))
r = 1.0e3
# define test and trial functions, and function that is updated
u = TrialFunction(V)
v = TestFunction(V)
w = Function(V)
# set the variational problem
a = inner(grad(u), grad(v))*dx + r*div(u)*div(v)*dx
b = -div(w)*div(v)*dx
F = inner(f, v)*dx
u = Function(V)
pde = LinearVariationalProblem(a, F - b, u, bc)
solver = LinearVariationalSolver(pde)
# Scott-Vogelius iterated penalty method
iters = 0; max_iters = 10; div_u_norm = 1
while iters < max_iters and div_u_norm > 1e-10:
    # solve and update w
    solver.solve()
    w.vector().axpy(-r, u.vector())
    # find the L^2 norm of div(u) to check stopping condition
    div_u_norm = sqrt(assemble(div(u)*div(u)*dx(mesh)))
    print "norm(div u)=%.2e"%div_u_norm
    iters += 1
print k,meshsize," %.2e"%errornorm(gee,u,norm_type=’l2’, degree_rise=3)
```

Program 12.1: Code to implement the Iterated Penalty Method.

Chapter 13

Solid mechanics

There is a strong analogy between models for fluids and solids. The model equations for all solids take the form

$$\rho \mathbf{u}_{tt} = \nabla\cdot \mathbf{T} + \mathbf{f},$$

where $\mathbf{u}$ is the displacement of the solid, $\mathbf{T}$ is called the Cauchy stress and $\mathbf{f}$ is externally given data. The models differ based on the way the stress $\mathbf{T}$ depends on the displacement $\mathbf{u}$. Time-independent models take the form

$$-\nabla\cdot \mathbf{T} = \mathbf{f}. \tag{13.1}$$

13.1 Elasticity

The simplest expression for the stress is linear: $\mathbf{T} = \mathbf{C} : \boldsymbol{\epsilon}$, where $\mathbf{C}$ is a material tensor, the **constituitive matrix**, and $\boldsymbol{\epsilon} = \frac{1}{2}\big(\nabla\mathbf{u} + \nabla\mathbf{u}^t\big)$. Such solids are called elastic. For isotropic models,

$$C_{ijkl} = K\delta_{ij}\delta_{kl} + \mu\big(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \tfrac{2}{3}\delta_{ij}\delta_{kl}\big), \text{ for } i,j,k,l = 1,2,3, \tag{13.2}$$

where δ_{ij} is the Kronecker-δ, K is the **bulk modulus** (or incompressibility) of the material, and μ is the **shear modulus**. Carrying out the tensor contraction, we have

$$\begin{aligned} T_{ij} &= K\delta_{ij}\epsilon_{kk} + 2\mu\big(\epsilon_{ij} - \tfrac{1}{3}\delta_{ij}\epsilon_{kk}\big) = \lambda\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij} \\ &= \lambda\delta_{ij}\nabla\cdot\mathbf{u} + \mu\big(\nabla\mathbf{u} + \nabla\mathbf{u}^t\big)_{ij} = \lambda\delta_{ij}\nabla\cdot\mathbf{u} + \mu\big(u_{i,j} + u_{j,i}\big), \end{aligned} \tag{13.3}$$

where $\lambda(= K - \frac{2}{3}\mu)$ and μ are known as the **Lamé parameters**, and the Einstein summation convention was used ($\epsilon_{kk} = \sum_{k=1}^{3}\epsilon_{kk} = \nabla\cdot\mathbf{u}$).

13.2 Elasticity variational formulation

The variational formulation of (13.1) takes the form: Find $\mathbf{u}$ such that $\mathbf{u} - \boldsymbol{\gamma} \in V$ such that

$$a_C\left(\mathbf{u},\mathbf{v}\right) = F(\mathbf{v}) \quad \forall \mathbf{v} \in V\,, \tag{13.4}$$

where $a(\cdot,\cdot) = a_{\nabla}(\cdot,\cdot)$ and $F(\cdot)$ are given by

$$a_C(\mathbf{u},\mathbf{v}) := \int_\Omega \mathbf{T} : \nabla \mathbf{v}\, d\mathbf{x} = \lambda \int_\Omega (\nabla\cdot \mathbf{u})(\nabla\cdot \mathbf{v})\, d\mathbf{x} + \mu \int_\Omega \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right) : \nabla \mathbf{v}\, d\mathbf{x}, \tag{13.5}$$

and

$$F(\mathbf{v}) := \int_\Omega \mathbf{f} \cdot \mathbf{v}\, d\mathbf{x}. \tag{13.6}$$

This is derived by multiplying (13.3) by $\mathbf{v}$ with a "dot" product, and integrating by parts as usual.

The space V consists of the d-fold Cartesian product of the subset of $H^1(\Omega)$ of functions vanishing on the boundary.

13.3 Anti-plane strain

In anti-plane strain [22], the component of strain normal to a particular plane (we will take it to be the (x_1, x_2) plane) is the only non-zero displacement, that is, $u_1 = u_2 = 0$, and thus $\mathbf{u} = (0, 0, w)$. This is an idealized state when the dimension of Ω is large in the x_3-direction, and the applied force is in that direction only, that is, $\mathbf{f} = (0, 0, f)$. It is assumed that the displacement $w = u_3$ is independent of x_3, although it does depend on (x_1, x_2). In particular, $\nabla\cdot \mathbf{u} = 0$. Thus

$$\nabla \mathbf{u} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ w_{,1} & w_{,2} & 0 \end{pmatrix} \text{ and } \boldsymbol{\epsilon} = \tfrac{1}{2} \begin{pmatrix} 0 & 0 & w_{,1} \\ 0 & 0 & w_{,2} \\ w_{,1} & w_{,2} & 0 \end{pmatrix}.$$

Therefore

$$\mathbf{T} = \mu \begin{pmatrix} 0 & 0 & w_{,1} \\ 0 & 0 & w_{,2} \\ w_{,1} & w_{,2} & 0 \end{pmatrix}.$$

and so

$$\nabla\cdot \mathbf{T} = \mu \begin{pmatrix} 0 \\ 0 \\ w_{,11} + w_{,22} \end{pmatrix} = \mu \begin{pmatrix} 0 \\ 0 \\ \Delta w \end{pmatrix}.$$

Thus the problem of anti-plane strain reduces to the familiar Laplace equation, $-\mu \Delta w = f$.

13.4 Plane strain

In plane strain [22], the component of strain normal to a particular plane (we will take it to be the (x, y) plane) is zero. This is the idealized state when the dimension of Ω is large in the z-direction, and the applied forces in that direction are zero. Thus $\mathbf{u} = (u, v, 0)$ and

$$\nabla \mathbf{u} = \begin{pmatrix} u_{,x} & u_{,y} & 0 \\ v_{,x} & v_{,y} & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } \boldsymbol{\epsilon} = \begin{pmatrix} u_{,x} & \frac{1}{2}(u_{,y} + v_{,x}) & 0 \\ \frac{1}{2}(u_{,y} + v_{,x}) & v_{,y} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

and thus the variational problem (13.4) applies where we identify the integration in (13.5) and (13.6) as being just over a two-dimensional domain.

13.5 The Plate-Bending Biharmonic Problem

Plates are thin structures, planar in the simplest case. Let us suppose that $\widehat{\Omega}$ is some domain in the x, y plane, and $\Omega = \Big\{(x, y, z)\ :\ (x, y) \in \widehat{\Omega},\ z \in [-\tau, \tau]\Big\}$, where we assume that τ is small with respect to the dimensions of $\widehat{\Omega}$. Using the **Kirchhoff hypothesis**,[1] the displacement $\mathbf{u} = (u, v, w)$ satisfies

$$u \approx -zw_{,x}, \quad v \approx -zw_{,y}.$$

See Figure 13.1.

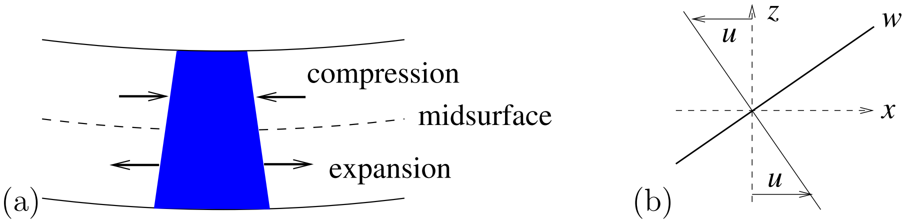

Figure 13.1: Relation between out-of-plane dispalcement and in-plane displacement leading to the Kirchhoff hypothesis. (a) Bending causes a combination of compression and expansion (or extension). (b) The slope of w causes a deformation in the (x, y) plane, that is, a change in (u, v), that depends on z. This is shown in the (x, z) plane.

Thus the equations of elasticity can be written as an equation for just the deflection w normal to the plane of the plate:

$$\nabla\mathbf{u} = \begin{pmatrix} -zw_{,xx} & -zw_{,xy} & -w_{,x} \\ -zw_{,xy} & -zw_{,yy} & -w_{,y} \\ w_{,x} & w_{,y} & 0 \end{pmatrix}, \qquad \boldsymbol{\epsilon} = \begin{pmatrix} -zw_{,xx} & -zw_{,xy} & 0 \\ -zw_{,xy} & -zw_{,yy} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\text{and } \mathbf{T} = -z\lambda\Delta w \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \mu \begin{pmatrix} -zw_{,xx} & -zw_{,xy} & 0 \\ -zw_{,xy} & -zw_{,yy} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Multiplying by $\mathbf{v} = -z\nabla\phi$ where ϕ depends only on x, y, integrating over Ω and integrating by parts, we find

$$\lambda \int_\Omega z^2 \Delta w \Delta\phi \, d\mathbf{x} + \mu \int_\Omega z^2 \begin{pmatrix} w_{,xx} & w_{,xy} \\ w_{,xy} & w_{,yy} \end{pmatrix} : \begin{pmatrix} \phi_{,xx} & \phi_{,xy} \\ \phi_{,xy} & \phi_{,yy} \end{pmatrix} d\mathbf{x} = \int_\Omega \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}.$$

[1] Gustav Robert Kirchhoff (1824—1887) was a mathematical physicist, made famous by work done on electrical circuits as a graduate student. Kirchhoff coined the term "black body radiation" and, with Bunsen, discovered caesium and rubidium, among many other achievements.

For simplicity, we consider the case where the body force $\mathbf{f} = \mathbf{0}$. We can expand to get

$$\begin{aligned}\begin{pmatrix} w_{,xx} & w_{,xy} \\ w_{,xy} & w_{,yy} \end{pmatrix} : \begin{pmatrix} \phi_{,xx} & \phi_{,xy} \\ \phi_{,xy} & \phi_{,yy} \end{pmatrix} &= w_{,xx}\phi_{,xx} + 2w_{,xy}\phi_{,xy} + w_{,yy}\phi_{,yy} \\ &= \Delta w \Delta \phi - w_{,xx}\phi_{,yy} - w_{,yy}\phi_{,xx} + 2w_{,xy}\phi_{,xy}.\end{aligned} \tag{13.7}$$

Note that the integral of z^2 with respect to z over the plate thickness is equal to $\frac{2}{3}\tau^3$. Then the above becomes

$$\tfrac{2}{3}\tau^3 \int_{\widehat{\Omega}} (\lambda + \mu)\Delta w \Delta \phi + \mu(-w_{,xx}\phi_{,yy} - w_{,yy}\phi_{,xx} + 2w_{,xy}\phi_{,xy})\, dxdy = 0.$$

Let $a_P(\cdot,\cdot)$ be the bilinear form defined on $H^2(\Omega)$ given by

$$a_P(u,v) := \int_\Omega \Delta u\, \Delta v - (1-\nu)\left(2u_{xx}v_{yy} + 2u_{yy}v_{xx} - 4u_{xy}v_{xy}\right)\, dxdy \tag{13.8}$$

where ν is a physical constant known as Poisson's ratio, and $2(1-\nu) = \mu/(\lambda+\mu)$. In the model for the bending of plates, ν is restricted to the range $[0, \frac{1}{2}]$, although negative values of ν correspond to auxetic materials [6]. However, $a_P(\cdot,\cdot)$ is known [3] to satisfy a Gårding-type inequality,

$$a_P(v,v) + K\|v\|^2_{L^2(\Omega)} \geq \alpha \|v\|^2_{H^2(\Omega)} \quad \forall v \in H^2(\Omega)\,, \tag{13.9}$$

where $\alpha > 0$ and $K < \infty$, for all $-3 < \nu < 1$. Note that for $\nu = 1$, such an inequality cannot hold as $a_P(v,v)$ vanishes in that case for all harmonic functions, v.

A simple coercivity estimate can be derived for $0 < \nu < 1$, as follows. Write

$$\begin{aligned} a_P(v,v) &= \int_\Omega \nu\left(v_{xx} + v_{yy}\right)^2 + (1-\nu)\left(\left(v_{xx} - v_{yy}\right)^2 + 4v_{xy}^2\right)\, dxdy \\ &\geq \min\{\nu, 1-\nu\} \int_\Omega \left(v_{xx} + v_{yy}\right)^2 + \left(v_{xx} - v_{yy}\right)^2 + 4v_{xy}^2\, dxdy \\ &= 2\min\{\nu, 1-\nu\} \int_\Omega v_{xx}^2 + v_{yy}^2 + 2v_{xy}^2\, dxdy \\ &= 2\min\{\nu, 1-\nu\} |v|^2_{H^2(\Omega)}. \end{aligned} \tag{13.10}$$

From (13.10), it follows that $a_P(\cdot,\cdot)$ is coercive over any closed subspace, $V \subset H^2(\Omega)$, such that $V \cap \mathcal{P}_1 = \emptyset$ (see [40, Section 5.9]). Thus, there is a constant $\alpha > 0$ such that

$$a_P(v,v) \geq \alpha \|v\|^2_{H^2(\Omega)} \quad \forall v \in V. \tag{13.11}$$

For $F \in H^2(\Omega)'$ and $V \subset H^2(\Omega)$, we consider the problem: find $u \in V$ such that

$$a_P(u,v) = F(v) \quad \forall v \in V. \tag{13.12}$$

As a consequence of the Riesz Representation Theorem (cf. 2.5.6), we have the following.

Theorem 13.1 *If $V \subset H^2(\Omega)$ is a closed subspace such that $V \cap \mathcal{P}_1 = \emptyset$ and* (13.10) *holds, then* (13.12) *has a unique solution.*

If u is sufficiently smooth, then integration by parts can be carried out, say with $v \in C_0^\infty(\Omega)$, to determine the differential equation satisfied. For example, if

$$F(v) := \int_\Omega f(x,y)v(x,y)\,dxdy,$$

where $f \in L^2(\Omega)$, then under suitable conditions on Ω [27] we have $u \in H^4(\Omega)$. Note that when integrating by parts, all of the terms multiplied by $1-\nu$ cancel, as they all yield various versions of the cross derivative u_{xxyy}. Thus, we find that $\Delta^2 u = f$ holds in the L^2 sense, independent of the choice of ν.

Various boundary conditions are of physical interest. Let $V^{\rm ss}$ denote the subset of $H^2(\Omega)$ consisting of functions which vanish (to first-order only) on $\partial\Omega$, i.e.,

$$V^{\rm ss} = \left\{ v \in H^2(\Omega) \ : \ v = 0 \text{ on } \partial\Omega \right\}.$$

With this choice for V in (13.12), the resulting model is called the “simply-supported” plate model, since the displacement, u, is held fixed (at a height of zero), yet the plate is free to rotate at the boundary. The “clamped” plate model consists of choosing $V^{\rm c} = \overset{\circ}{H}{}^2(\Omega)$, the subset of $H^2(\Omega)$ consisting of functions which vanish to second order on $\partial\Omega$:

$$V^{\rm c} = \left\{ v \in H^2(\Omega) \ : \ v = \frac{\partial v}{\partial \nu} = 0 \text{ on } \partial\Omega \right\}.$$

Here, the rotation of the plate is also prescribed at the boundary.

In the simply-supported case ($V = V^{\rm ss}$), there is another, *natural* boundary condition that holds. In this sense, this problem has a mixture of Dirichlet and Neumann boundary conditions, but they hold on all of $\partial\Omega$. The natural boundary condition is found using integration by parts, but with v having an arbitrary, nonzero normal derivative on $\partial\Omega$. One finds (Bergman and Schiffer 1953) that the “bending moment” $\Delta u + (1-\nu)u_{tt}$ must vanish on $\partial\Omega$, where u_{tt} denotes the second directional derivative in the tangential direction. These results are summarized in the following.

Theorem 13.2 *Suppose that V is any closed subspace satisfying $\overset{\circ}{H}{}^2(\Omega) \subset V \subset H^2(\Omega)$. If $f \in L^2(\Omega)$, and if $u \in H^4(\Omega)$ satisfies* (13.12) *with $F(v) = (f,v)$, then u satisfies*

$$\Delta^2 u = f$$

in the $L^2(\Omega)$ sense. For $V = V^{\rm c}$, u satisfies

$$u = \frac{\partial u}{\partial \nu} = 0 \text{ on } \partial\Omega$$

and for $V = V^{\rm ss}$, u satisfies

$$u = \Delta u + (1-\nu)u_{tt} = 0 \text{ on } \partial\Omega.$$

To approximate (13.12), we need a subspace V_h of $H^2(\Omega)$. For example, we could take a space based on the Argyris elements [40, Examples 3.2.10 and 3.2.11]. With either choice of V as above, if we choose V_h to satisfy the corresponding boundary conditions, we obtain the following.

Theorem 13.3 *If $V_h \subset V$ is based on Argyris elements of order $k \geq 5$ then there is a unique $u_h \in V_h$ such that*

$$a_P(u_h, v) = F(v) \quad \forall v \in V_h.$$

Moreover,

$$\begin{aligned} \|u - u_h\|_{H^2(\Omega)} &\leq C \inf_{v \in V_h} \|u - v\|_{H^2(\Omega)} \\ &\leq Ch^{k-1} \|u\|_{H^{k+1}(\Omega)}. \end{aligned} \tag{13.13}$$

Since the Argyris interpolant is defined for $u \in H^4(\Omega)$, the above result also holds for k replaced by any integer s in the range $3 \leq s \leq k$. Provided sufficient regularity holds (Blum & Rannacher 1980) for the solution of (13.12), namely

$$\|u\|_{H^{s+1}(\Omega)} \leq C \|f\|_{H^{s-3}(\Omega)} ,$$

then we also have the following negative-norm estimate (cf. [40, Section 5.9]).

Theorem 13.4 *Assuming* (13.5) *holds for $s = m + 3$, with $0 \leq m \leq k - 3$, we have*

$$\|u - u_h\|_{H^{-m}(\Omega)} \leq Ch^{m+2} \|u - u_h\|_{H^2(\Omega)}.$$

It is interesting to note that we do not immediately get an estimate in the H^1 norm for the error $u - u_h$. An estimate of the form

$$\|u - u_h\|_{H^2(\Omega)} \leq Ch^{s-1} \|u\|_{H^{s+1}(\Omega)}$$

can be proved for any $1 \leq s \leq k$ [40, Section 5.9]. For more details regarding the biharmonic equation model for plate bending, see the survey [161].

Several mixed methods reducing the biharmonic problem to a system of second-order problems have been developed [76, 84, 77].

13.6 The Babuška Paradox

The Babuška Paradox relates to the limit of polygonal approximations to a smooth boundary. For example, let Ω be the unit disc, and let Ω_n denote regular polygons inscribed in Ω with n sides. Then the Paradox is that the solutions of the simply supported plate problems on Ω_n converge to the solution of the clamped plate problem on Ω as $n \to \infty$ [128, Chapter 18, Volume II].

The reason for the paradox is that, at each vertex of Ω_n , the gradient of w must vanish for any sufficiently smooth function that vanishes on $\partial\Omega_n$. This is illustrated in Figure 13.2.

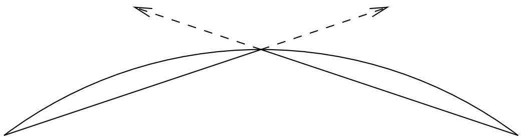

Figure 13.2: Polygonal approximation in the Babuška Paradox. At a vertex, the gradient of the simply supported solution must vanish as its tangential derivatives are zero in two independent directions.

Thus in the limit, $\nabla w = 0$ on the boundary, where $w = \lim_{n\to\infty} w_n$ and w_n denotes the solution of simply supported plate problem on Ω_n.

A corollary of this paradox is the following. Suppose we approximate a smooth domain Ω by polygons Ω_n and form the finite element approximation $w_{n,h}$ of the simply supported plate problem, say with $h = 1/n$. Then as $n \to \infty$ (equivalently, $h \to 0$), we expect that $w_{n,h}$ will converge to the solution w of the clamped plate problem on Ω, not the simply supported problem. This type of numerical error is the most insidious possible, in that the convergence is likely to be quite stable. There will be no red flags to indicate that something is wrong.

The Babuška Paradox has been widely studied [12, 127, 135, 146], and it is now generally known as the Babuška-Sapondzhyan Paradox [51, 126, 147]. Since the biharmonic equation arises in other contexts, including the Stokes equations, this paradox is of broader interest [171, 166, 68, 49].

13.7 Exercises

Exercise 13.1 *Let $\widehat{\Omega}$ be the unit square, and let*

$$\Omega = \Big\{(x, y, z) \ : \ (x, y) \in \widehat{\Omega}, \ z \in [-\tau, \tau]\Big\}.$$

Compare the solution of the plate bending problem on $\widehat{\Omega}$ to the solution of the full elasticity problem on Ω for various values of τ.

Chapter 14

The Navier-Stokes Equations

In Chapter 12, we derived the Navier-Stokes equations under the assumption that the stress depends linearly upon the gradient of the fluid velocity. Here we develop the variational theory for these equations and present some computational algorithms for solving them.

14.1 The Navier-Stokes Equations

The Navier-Stokes equations for the flow of a viscous, incompressible, Newtonian fluid can be written

$$\begin{aligned} -\Delta\mathbf{u} + \nabla p &= -R\left(\mathbf{u}\cdot\nabla\mathbf{u} + \mathbf{u}_t\right) \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{14.1}$$

in $\Omega \subset \mathbb{R}^d$, where $\mathbf{u}$ denotes the fluid velocity, p denotes the pressure, and R denotes the Reynolds number [112]. In our context, $R = 1/\eta$ where η denotes the fluid (kinematic) viscosity.

These equations must be supplemented by appropriate boundary conditions, such as the Dirichlet boundary conditions, $\mathbf{u} = \boldsymbol{\gamma}$ on $\partial\Omega$.

A complete variational formulation of (14.1) takes the form: Find $\mathbf{u}$ such that $\mathbf{u} - \boldsymbol{\gamma} \in V$ and $p \in \Pi$ such that

$$\begin{aligned} a\left(\mathbf{u},\mathbf{v}\right) + b\left(\mathbf{v},p\right) + &R\Big(c\left(\mathbf{u},\mathbf{u},\mathbf{v}\right) + \left(\mathbf{u}_t,\mathbf{v}\right)_{\underset{\sim}{L}^2}\Big) = 0 \quad \forall \mathbf{v} \in V\,, \\ &b(\mathbf{u},q) = 0 \quad \forall q \in \Pi\,, \end{aligned} \tag{14.2}$$

where e.g. $a(\cdot,\cdot)$, $b(\cdot,\cdot)$ and $c(\cdot,\cdot,\cdot)$ are given by

$$a(\mathbf{u},\mathbf{v}) := \int_\Omega \sum_{i,j=1}^n u_{i,j} v_{i,j}\, d\mathbf{x}, \tag{14.3}$$

$$b(\mathbf{v},q) := -\int_\Omega \sum_{i=1}^n v_{i,i}\, q\, d\mathbf{x}, \tag{14.4}$$

$$c(\mathbf{u},\mathbf{v},\mathbf{w}) := \int_\Omega (\mathbf{u}\cdot\nabla\mathbf{v})\cdot\mathbf{w}\,d\mathbf{x}, \tag{14.5}$$

and $(\cdot,\cdot)_{\underset{\sim}{L}^2}$ denotes the $L^2(\Omega)^d$-inner-product. The spaces V and Π are the same as for the Stokes equations (Chapter 12), as are the forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$. As noted in Chapter 12, the $a(\cdot,\cdot)$ form can be either the gradient form (12.7) or the "epsilon" form (12.10).

14.1.1 Properties of the nonlinear term

The trilinear form (14.5) has some special properties that reflect important physical properties of fluid dynamics. To see these, we need to derive some calculus identities. For any vector-valued function $\mathbf{u}$ and scalar-valued function v, the product rule for derivatives gives (Exercise 14.2)

$$\nabla\cdot(\mathbf{u}\,v) = (\nabla\cdot\mathbf{u})v + \mathbf{u}\cdot\nabla v. \tag{14.6}$$

For any vector-valued function $\mathbf{u}$ and scalar-valued functions v and w, applying the product rule for derivatives again gives

$$\nabla\cdot(\mathbf{u}\,v\,w) = (\nabla\cdot\mathbf{u})v\,w + (\mathbf{u}\cdot\nabla v)w + (\mathbf{u}\cdot\nabla w)v.$$

Thus if we apply the divergence theorem we get

$$0 = \int_\Omega (\nabla\cdot\mathbf{u})v\,w\,d\mathbf{x} + \int_\Omega (\mathbf{u}\cdot\nabla v)w\,d\mathbf{x} + \int_\Omega (\mathbf{u}\cdot\nabla w)v\,d\mathbf{x},$$

provided that the product $\mathbf{u}\,v\,w$ vanishes on $\partial\Omega$. Thus if $\mathbf{u}$ satisfies the divergence constraint $\nabla\cdot\mathbf{u} = 0$ in (14.1) and the product $\mathbf{u}\,v\,w$ vanishes on $\partial\Omega$, we have

$$\int_\Omega (\mathbf{u}\cdot\nabla v)w\,d\mathbf{x} = -\int_\Omega (\mathbf{u}\cdot\nabla w)v\,d\mathbf{x}. \tag{14.7}$$

Suppose now that $\mathbf{v}$ and $\mathbf{w}$ are vector-valued functions, and $\mathbf{u}$ or $\mathbf{v}$ or $\mathbf{w}$ vanishes at each point on $\partial\Omega$. Applying (14.7), we find (Exercise 14.3)

$$\int_\Omega (\mathbf{u}\cdot\nabla\mathbf{v})\cdot\mathbf{w}\,d\mathbf{x} = -\int_\Omega (\mathbf{u}\cdot\nabla\mathbf{w})\cdot\mathbf{v}\,d\mathbf{x}. \tag{14.8}$$

Thus we have shown the following.

Lemma 14.1 *Suppose that* $\mathbf{u}$ *satisfies the divergence constraint* $\nabla\cdot\mathbf{u} = 0$ *in (14.1) and that* $\mathbf{u}$ *or* $\mathbf{v}$ *or* $\mathbf{w}$ *vanishes at each point on* $\partial\Omega$*. Then the trilinear form (14.5) is antisymmetric in the last two arguments:*

$$c(\mathbf{u},\mathbf{v},\mathbf{w}) = -c(\mathbf{u},\mathbf{w},\mathbf{v}). \tag{14.9}$$

In particular, if $\mathbf{v}$ *satisfies the divergence constraint* $\nabla\cdot\mathbf{v} = 0$ *and vanishes on* $\partial\Omega$*, then* $c(\mathbf{v},\mathbf{v},\mathbf{v}) = 0$.

14.1.2 Time-stepping schemes

One of the simplest time-stepping schemes for the Navier-Stokes equations (14.1) is implicit with respect to the linear terms and explicit with respect to the nonlinear terms. Expressed in variational form, it is

$$\begin{aligned} a\left(\mathbf{u}^{\ell},\mathbf{v}\right)+b\left(\mathbf{v},p^{\ell}\right)+R\,c\left(\mathbf{u}^{\ell-1},\mathbf{u}^{\ell-1},\mathbf{v}\right)+\frac{R}{\Delta t}\left(\mathbf{u}^{\ell}-\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}&=0,\\ b\left(\mathbf{u}^{\ell},q\right)&=0, \end{aligned} \tag{14.10}$$

where, here and below, $\mathbf{v}$ varies over all V (or V_h) and q varies over all Π (or Π_h) and Δt denotes the time-step size.

In practice, we often use more efficient time-stepping schemes, but the main issues related to solving the resulting linear equations remain the same. At each time step, one has a problem to solve of the form (12.18) for $(\mathbf{u}^{\ell},p^{\ell})$ but with the form $a(\cdot,\cdot)$ more general, namely

$$\tilde{a}(\mathbf{u},\mathbf{v}):=a(\mathbf{u},\mathbf{v})+\tau\left(\mathbf{u},\mathbf{v}\right)_{\underset{\sim}{L}^2}, \tag{14.11}$$

where the constant $\tau=R/\Delta t$. Numerical experiments will be presented subsequently for such a problem. Note that the linear algebraic problem to be solved at each time step is the same. This improves the efficiency of the solution process substantially.

Equation (14.10) may now be written as a problem for $\left(\mathbf{u}^{\ell},p^{\ell}\right)$ which is nearly of the form (12.18):

$$\begin{aligned} \tilde{a}\left(\mathbf{u}^{\ell},\mathbf{v}\right)+b\left(\mathbf{v},p^{\ell}\right)&=-R\,c\left(\mathbf{u}^{\ell-1},\mathbf{u}^{\ell-1},\mathbf{v}\right)+\tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2},\\ b\left(\mathbf{u}^{\ell},q\right)&=0. \end{aligned} \tag{14.12}$$

Note that we have $\mathbf{u}^{\ell}=\boldsymbol{\gamma}$ on $\partial\Omega$, that is, $\mathbf{u}^{\ell}=\mathbf{u}_0^{\ell}+\boldsymbol{\gamma}$ where $\mathbf{u}_0^{\ell}\in V$. The variational problem for $\mathbf{u}^{\ell}$ can be written: Find $\mathbf{u}_0^{\ell}\in V$ and $p\in\Pi$ such that

$$\begin{aligned} \tilde{a}\left(\mathbf{u}_0^{\ell},\mathbf{v}\right)+b\left(\mathbf{v},p\right)&=-\tilde{a}\left(\boldsymbol{\gamma},\mathbf{v}\right)\ -R\,c\left(\mathbf{u}^{\ell-1},\mathbf{u}^{\ell-1},\mathbf{v}\right)+\tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}\quad\forall\mathbf{v}\in V\\ b\left(\mathbf{u}_0^{\ell},q\right)&=-\,b\left(\boldsymbol{\gamma},q\right)\quad\forall q\in\Pi. \end{aligned} \tag{14.13}$$

This is of the form (12.18) with

$$\begin{aligned} F(\mathbf{v})&=-\tilde{a}\left(\boldsymbol{\gamma},\mathbf{v}\right)-R\,c\left(\mathbf{u}^{\ell-1},\mathbf{u}^{\ell-1},\mathbf{v}\right)+\tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}\quad\forall\mathbf{v}\in V\\ G\left(q\right)&=-\,b\left(\boldsymbol{\gamma},q\right)\quad\forall q\in\Pi. \end{aligned} \tag{14.14}$$

Note that the inhomogeneous boundary data $\boldsymbol{\gamma}$ appears in both right-hand sides, F and G. Thus we are forced to deal with a nonzero G, but we will see that this can be handled quite naturally.

14.1.3 Justifying the choice

We need to have some way to see that our choices in the definition of the time-stepping scheme are reasonable. To do so, we develop some bounds that characterize the evolution of the error from one step to the next. Subtracting two consecutive versions of (14.12), we find the following formula for $\mathbf{e}^\ell := \mathbf{u}^\ell - \mathbf{u}^{\ell-1}$:

$$\begin{aligned}
\tilde{a}\left(\mathbf{e}^\ell, \mathbf{e}^\ell\right) &= R\Big(c\left(\mathbf{u}^{\ell-1}, \mathbf{u}^{\ell-1}, \mathbf{e}^\ell\right) - c\left(\mathbf{u}^{\ell-2}, \mathbf{u}^{\ell-2}, \mathbf{e}^\ell\right)\Big) + \tau\left(\mathbf{e}^{\ell-1}, \mathbf{e}^\ell\right)_{\underset{\sim}{L}^2}, \\
&= R\Big(c\left(\mathbf{u}^{\ell-1}, \mathbf{u}^{\ell-1}, \mathbf{e}^\ell\right) - c\left(\mathbf{u}^{\ell-2}, \mathbf{u}^{\ell-1}, \mathbf{e}^\ell\right) \\
&\qquad + c\left(\mathbf{u}^{\ell-2}, \mathbf{u}^{\ell-1}, \mathbf{e}^\ell\right) - c\left(\mathbf{u}^{\ell-2}, \mathbf{u}^{\ell-2}, \mathbf{e}^\ell\right)\Big) + \tau\left(\mathbf{e}^{\ell-1}, \mathbf{e}^\ell\right)_{\underset{\sim}{L}^2}, \\
&= R\Big(c\left(\mathbf{e}^{\ell-1}, \mathbf{u}^{\ell-1}, \mathbf{e}^\ell\right) + c\left(\mathbf{u}^{\ell-2}, \mathbf{e}^{\ell-1}, \mathbf{e}^\ell\right)\Big) + \tau\left(\mathbf{e}^{\ell-1}, \mathbf{e}^\ell\right)_{\underset{\sim}{L}^2}.
\end{aligned} \tag{14.15}$$

From the Cauchy-Schwarz inequality (see Exercise 14.1), we find

$$|c\left(\mathbf{u}^{\ell-2}, \mathbf{e}^{\ell-1}, \mathbf{e}^\ell\right)| \le \|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)} \left(a\left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right)\right)^{1/2} \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right)^{1/2}.$$

From (14.9) and the Cauchy-Schwarz inequality, we find

$$|c\left(\mathbf{e}^{\ell-1}, \mathbf{u}^{\ell-1}, \mathbf{e}^\ell\right)| = |c\left(\mathbf{e}^{\ell-1}, \mathbf{e}^\ell, \mathbf{u}^{\ell-1}\right)| \le \|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)} \left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right)^{1/2} \left(a(\mathbf{e}^\ell, \mathbf{e}^\ell)\right)^{1/2}.$$

Estimating the terms in (14.15) with the "c" form in them and applying the Cauchy-Schwarz inequality, we find

$$\begin{aligned}
\tilde{a}(\mathbf{e}^\ell, \mathbf{e}^\ell) &\le R\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)} \left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right)^{1/2} \left(a(\mathbf{e}^\ell, \mathbf{e}^\ell)\right)^{1/2} \\
&\quad + R\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)} \left(a(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1})\right)^{1/2} \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right)^{1/2} + \tau\left(\mathbf{e}^{\ell-1}, \mathbf{e}^\ell\right)_{\underset{\sim}{L}^2}, \\
&\le R\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)} \left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right)^{1/2} \left(a\left(\mathbf{e}^\ell, \mathbf{e}^\ell\right)\right)^{1/2} \\
&\quad + R\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)} \left(a(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1})\right)^{1/2} \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right)^{1/2} \\
&\quad + \tau\left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right)^{1/2} \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right)^{1/2}.
\end{aligned}$$

Using the simple fact that $2xy \le x^2 + y^2$ for all real numbers x, y, we get

$$\begin{aligned}
\tilde{a}(\mathbf{e}^\ell, \mathbf{e}^\ell) &\le \tfrac{1}{2}\left(R\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}\right)^2 \left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right) + \tfrac{1}{2}a\left(\mathbf{e}^\ell, \mathbf{e}^\ell\right) \\
&\quad + \tfrac{1}{2}\left(R\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)}\right)^2 \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right) + \tfrac{1}{2}a\left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right) \\
&\quad + \tfrac{1}{2}\tau\left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right) + \tfrac{1}{2}\tau\left(\mathbf{e}^\ell, \mathbf{e}^\ell\right) \\
&= \tfrac{1}{2}\left(R\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}\right)^2 \left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right) + \tfrac{1}{2}\left(R\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)}\right)^2 \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right) \\
&\quad + \tfrac{1}{2}\tilde{a}\left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right) + \tfrac{1}{2}\tilde{a}\left(\mathbf{e}^\ell, \mathbf{e}^\ell\right).
\end{aligned}$$

Subtracting $\frac{1}{2}\tilde{a}\left(\mathbf{e}^\ell, \mathbf{e}^\ell\right)$ from both sides of this inequality and then multiplying by 2 we get

$$\tilde{a}(\mathbf{e}^\ell, \mathbf{e}^\ell) \le \left(R\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}\right)^2 \left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right) + \left(R\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)}\right)^2 \left(\mathbf{e}^\ell, \mathbf{e}^\ell\right) + \tilde{a}\left(\mathbf{e}^{\ell-1}, \mathbf{e}^{\ell-1}\right).$$

Recalling that $\tau = R/\Delta t$, we have $R^2 = \tau R\Delta t$ and thus find

$$\begin{aligned}\tilde{a}\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right) \leq\, & \tilde{a}\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right) \\ & + R\Delta t\left(\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}\right)^2 \tau\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right) + R\Delta t\left(\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)}\right)^2 \tau\left(\mathbf{e}^{\ell},\mathbf{e}^{\ell}\right) \\ \leq\, & \tilde{a}\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)\left(1 + R\Delta t\left(\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}\right)^2\right) \\ & + R\Delta t\left(\|\mathbf{u}^{\ell-2}\|_{L^\infty(\Omega)}\right)^2 \tilde{a}\left(\mathbf{e}^{\ell},\mathbf{e}^{\ell}\right).\end{aligned}$$

Thus

$$\tilde{a}\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right) \leq \tilde{a}\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)\frac{1 + R\Delta t\|\mathbf{u}^{\ell-1}\|^2_{L^\infty(\Omega)}}{1 - R\Delta t\|\mathbf{u}^{\ell-2}\|^2_{L^\infty(\Omega)}}. \tag{14.16}$$

We will show that (14.16) implies that the time-stepping scheme is stable provided Δt is small enough and $\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}$ stays bounded. However, $\|\mathbf{u}^{\ell}\|_{L^\infty(\Omega)}$ can not be bounded *a priori* in terms of $\tilde{a}\left(\mathbf{u}^\ell,\mathbf{e}^\ell\right)$, so we must monitor the size of this term during the computation to guarantee the validity of our estimates.

To see why (14.16) is a useful bound, we need to develop some estimates. First of all, we suppose that there is a constant c such that

$$b_\ell := R\max_{i=0,\ldots,\ell-1}\left(\|\mathbf{u}^i\|_{L^\infty(\Omega)}\right)^2 \leq c \leq \frac{1}{3\Delta t}. \tag{14.17}$$

The quantity b_ℓ is something that we can monitor as the solution progresses, and if we find $b_\ell > 1/(3\Delta t)$ at some stage, then, in principle, we should decrease the time step. In view of (14.17), (14.16) becomes

$$\tilde{a}\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right) \leq \tilde{a}\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)\frac{1+\epsilon}{1-\epsilon},$$

where $\epsilon = c\Delta t$. It is easy to show (Exercise 14.4) that

$$\frac{1+\epsilon}{1-\epsilon} \leq 1 + 3\epsilon \quad \forall\epsilon \leq \frac{1}{3}. \tag{14.18}$$

Thus we must only consider a simple estimate for a sequence of positive real numbers e_k of the form $e_k \leq (1 + c\Delta t)e_{k-1}$. By induction, we conclude that $e_k \leq (1 + c\Delta t)^k e_0$. Suppose that $t = k\Delta t$. Then (Exercise 14.5)

$$e_k \leq (1 + ct/k)^k e_0 \leq e^{ct} e_0 \quad \forall k. \tag{14.19}$$

Combining the previous estimates, we find that, provided (14.17) holds, then

$$\tilde{a}(\mathbf{e}^\ell,\mathbf{e}^\ell) \leq e^{3c\,\ell\Delta t}\,\tilde{a}(\mathbf{e}^0,\mathbf{e}^0). \tag{14.20}$$

Thus we see that (14.16) implies that $\tilde{a}\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right)$ can grow at most exponentially in time, uniformly as $\Delta t \to 0$, provided only that $\|\mathbf{u}^{\ell-1}\|_{L^\infty(\Omega)}$ stays bounded, something we can monitor as the computation evolves.

14.2 Bounds for the solution of Navier-Stokes

The existence of smooth solutions to the Navier-Stokes equations is not known to hold in three dimensions for all time. This is one of the major open problems of modern mathematics, so it is interesting to understand what is and is not known. Uniqueness for the time-dependent equations (14.1) can be demonstrated for sufficiently smooth solutions. But the real question to know is what class is appropriate that contains solutions.

14.2.1 A priori bounds for the solution of Navier-Stokes

If you put $\mathbf{v}=\mathbf{u}$ in (14.3), a simple bound appears [116] that shows that the L^2 norm of $\mathbf{u}$ is always decreasing, say for $\boldsymbol{\gamma}=0$ as we now assume, and it implies that the H^1 norm of $\mathbf{u}$ is square-integrable in time. However, this alone is not sufficient smoothness to demonstrate uniqueness of solutions except in the two-dimensional case [116]. If we instead multiply (14.1) by $-\Delta\mathbf{u}$, integrate over Ω and integrate by parts, we find (since $\Delta\mathbf{u}$ also has divergence zero)

$$\|\Delta\mathbf{u}\|^2_{L^2(\Omega)}+R\left(c\left(\mathbf{u},\mathbf{u},-\Delta\mathbf{u}\right)+\tfrac{1}{2}\tfrac{d}{dt}a\left(\mathbf{u},\mathbf{u}\right)\right)=0\,, \tag{14.21}$$

Using the arithmetic-geometric mean inequality in the form

$$2rs\le\delta r^2+\frac{1}{\delta}s^2 \tag{14.22}$$

which holds for any $\delta>0$ and any real numbers r and s, we find that

$$\left|c\left(\mathbf{u},\mathbf{u},-\Delta\mathbf{u}\right)\right|\le\tfrac{1}{2}CR\int_\Omega|\mathbf{u}\cdot\nabla\mathbf{u}|^2\,d\mathbf{x}+\frac{1}{2R}\|\Delta\mathbf{u}\|^2_{L^2(\Omega)}\,. \tag{14.23}$$

Integrating (14.21) in time and applying (14.23) we find

$$\begin{aligned}a\left(\mathbf{u}(t),\mathbf{u}(t)\right)+\int_0^t\|\Delta\mathbf{u}\|^2_{L^2(\Omega)}\,ds\le\\ a\left(\mathbf{u}_0,\mathbf{u}_0\right)+CR^2\int_0^t\int_\Omega|\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x}\,ds\,,\end{aligned} \tag{14.24}$$

where we think of $\mathbf{u}(s)$ as a function of space (on Ω) for each s and $\mathbf{u}_0=\mathbf{u}(0)$.

Lemma 14.2 *Suppose that Ω is a three-dimensional domain. For any $r>3$, there is a $C<\infty$ such that*

$$\int_\Omega|\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x}\le C\|\mathbf{u}\|^2_{L^r(\Omega)}\|\mathbf{u}\|^{\epsilon_r}_{L^2(\Omega)}|\mathbf{u}|^{2-\epsilon_r}_{H^2(\Omega)}\,. \tag{14.25}$$

for all $\mathbf{u}\in H^2(\Omega)^3$, where $\epsilon_r=1-\frac{3}{r}$.

Let $p' = r/2$ ($> 3/2$) and $p = \frac{p'}{p'-1}$ (< 3). Then Hölder's inequality implies

$$\int_\Omega |\mathbf{u}(s) \cdot \nabla \mathbf{u}(s)|^2 \, d\mathbf{x} \le C \|\mathbf{u}\|_{L^r(\Omega)}^2 |\mathbf{u}|_{W_{2p}^1(\Omega)}^2 \, . \tag{14.26}$$

Sobolev's imbedding implies that $L^{2p}(\Omega) \subset H^1(\Omega)$, and

$$|\mathbf{u}|_{W_{2p}^1(\Omega)} \le C \|\mathbf{u}\|_{L^2(\Omega)}^{\epsilon/2} \, |\mathbf{u}|_{H^2(\Omega)}^{1-\epsilon/2} \, , \tag{14.27}$$

where ϵ must satisfy (by scaling) the relation

$$-1 + \frac{3}{2p} = \frac{3}{2}\frac{\epsilon}{2} - \frac{1}{2}\left(1 - \frac{\epsilon}{2}\right) = \epsilon - \frac{1}{2} \tag{14.28}$$

which implies

$$\epsilon = \frac{3}{2p} - \frac{1}{2} = \frac{1}{2}\left(\frac{3}{p} - 1\right) = \frac{1}{2}\left(2 - \frac{3}{p'}\right) = 1 - \frac{3}{r} \tag{14.29}$$

and completes the proof.

Now we need the arithmetic-geometric mean inequality in the form

$$rs \le \delta r^{1+\epsilon} + \delta^{-1/\epsilon} s^{1+1/\epsilon} \tag{14.30}$$

which holds for any $\delta > 0$, $\epsilon > 0$ and any non-negative real numbers r and s. This can be proved by a simple case analysis. If $rs \le \delta r^{1+\epsilon}$, we are done. If not, then $rs > \delta r^{1+\epsilon}$, and so $s > \delta r^\epsilon$. This implies $s^{1/\epsilon} > \delta^{1/\epsilon} r$, and so $rs < \delta^{-1/\epsilon} s^{1+1/\epsilon}$.

Combining (14.30) (with $\epsilon = (1 - \epsilon_r/2)^{-1} - 1$ and $\delta = 1/2CR^2$), elliptic regularity ($|\mathbf{u}|_{H^2(\Omega)}^2 \le C \|\Delta \mathbf{u}\|_{L^2(\Omega)}^2$) and (14.25), we find

$$CR^2 \int_\Omega |\mathbf{u}(s) \cdot \nabla \mathbf{u}(s)|^2 \, d\mathbf{x} \le C' R^{-2+4/\epsilon_r} \|\mathbf{u}\|_{L^r(\Omega)}^{2+4/\epsilon_r} + \tfrac{1}{2} |\Delta \mathbf{u}|_{L^2(\Omega)}^2 \tag{14.31}$$

since $\|\mathbf{u}\|_{L^2(\Omega)} \le C \|\mathbf{u}\|_{L^r(\Omega)}$, and

$$1/\epsilon = \frac{1}{(1 - \epsilon_r/2)^{-1} - 1} = \frac{1 - \epsilon_r/2}{1 - (1 - \epsilon_r/2)} = \frac{1 - \epsilon_r/2}{\epsilon_r/2} = \frac{2}{\epsilon_r} - 1 \, , \tag{14.32}$$

and thus $1 + 1/\epsilon = 2/\epsilon_r$, which implies that

$$(2 + \epsilon_r)(1 + 1/\epsilon) = (2 + \epsilon_r)\frac{2}{\epsilon_r} = \frac{4}{\epsilon_r} + 2 \, . \tag{14.33}$$

Finally combining (14.31) with (14.24), we find

$$a(\mathbf{u}(t), \mathbf{u}(t)) + \tfrac{1}{2} \int_0^t \|\Delta \mathbf{u}\|_{L^2(\Omega)}^2 \, ds \le a(\mathbf{u}_0, \mathbf{u}_0) + C' R^{-2+4/\epsilon_r} \int_0^t \|\mathbf{u}\|_{L^r(\Omega)}^{2+4/\epsilon_r} \, ds \, , \tag{14.34}$$

where C is a constant depending on r but independent of R and $\mathbf{u}$. Recall that Ω is assumed to be a three-dimensional domain. In the two-dimensional case, better bounds are obtained. Estimate (14.34) implies that $\mathbf{u}$ remains smooth as long as $\|\mathbf{u}\|_{L^r(\Omega)}$ remains bounded, for any $r > 3$. However, there is (so far) no way to prove a priori that $\|\mathbf{u}\|_{L^r(\Omega)}$ remains bounded.

14.2.2 A posteriori bounds for the solution of Navier-Stokes

Here we review results presented in [30]. A more sophisticated approach, with better estimates, is presented in [94].

In Section 14.2.1, we saw that if we only knew that $\|\mathbf{u}\|_{L^r(\Omega)}$ (for $r > 3$) were bounded in time, then so would be $\|\mathbf{u}\|_{H^1(\Omega)}$, and $\|\mathbf{u}\|_{H^2(\Omega)}$ would be square-integrable as a function of time. This is sufficient to prove uniqueness of the solutions [116].

A finite element approximation of (14.1), including approximating the time derivative appropriately, can lead to an estimate of the form

$$\|\mathbf{u}(t) - \mathbf{u}_h(t)\|_{L^2(\Omega)} \leq C(t)h\|\mathbf{u}\|_{L^\infty(0,t;H^1(\Omega))} \tag{14.35}$$

The coefficient $C(t)$ may be growing exponentially in t, but it is independent of h and $\mathbf{u}$. Using simple inverse estimates for the finite element spaces, a bound of the form

$$\|\mathbf{u}(t) - \mathbf{u}_h(t)\|_{L^r(\Omega)} \leq C(t)h^{1-d/2+d/r}\|\mathbf{u}\|_{L^\infty(0,t;H^1(\Omega))} \tag{14.36}$$

holds for $2 \leq r \leq 6$. We will mainly be interested in the case of $d = 3$ dimensions, the two-dimensional case being qualitatively easier. When $r = 6$, the exponent of h is zero ($d = 3$), and there is no convergence rate. If $r = 4$, the convergence rate is $h^{1/4}$.

Define quantities

$$\begin{aligned} \sigma(t) =& \sigma^r(t) := \sup_{0\leq s\leq t} \|\mathbf{u}(s)\|_{L^r(\Omega)} \quad \text{and} \\ \sigma_h(t) =& \sigma_h^r(t) := \sup_{0\leq s\leq t} \|\mathbf{u}_h(s)\|_{L^r(\Omega)} \,. \end{aligned} \tag{14.37}$$

Then both σ and σ_h are continuous, non-decreasing functions of t. From now on, r will be fixed in the interval $3 < r < 6$, and the dependence of everything on r will be suppressed. Then the triangle inequality implies

$$\sigma(t) \leq \sigma_h(t) + C(t)h^{1-d/2+d/r}\|\mathbf{u}\|_{L^\infty(0,t;H^1(\Omega))} \tag{14.38}$$

On the other hand, (14.34) implies that

$$\|\mathbf{u}\|^2_{L^\infty(0,t;H^1(\Omega))} \leq \|\mathbf{u}_0\|^2_{H^1(\Omega)} + C(R)t\sigma(t)^\gamma \tag{14.39}$$

where $\gamma = 2 + 4/\epsilon_r$. Combining these two inequalities, we have

$$\sigma(t) \leq \sigma_h(t) + C(t)h^{1-d/2+d/r}\left(\|\mathbf{u}_0\|^2_{H^1(\Omega)} + C(R)t\sigma(t)^\gamma\right)^{1/2} \tag{14.40}$$

where $\gamma = 2 + 4/\epsilon_r$. Define a function ϕ by

$$\phi(\sigma, t) := \left(\|\mathbf{u}_0\|^2_{H^1(\Omega)} + C(R)t\sigma^\gamma\right)^{1/2}, \tag{14.41}$$

where we view $\mathbf{u}_0$ and R as fixed parameters for the moment. Then we have the circular-looking bound

$$\sigma(t) \leq \sigma_h(t) + C(t)h^{1-d/2+d/r}\phi(\sigma(t), t)\,. \tag{14.42}$$

We have the following result [30].

Theorem 14.1 *Suppose that* $\mathbf{u}_0$ *is not zero and that* h *is sufficiently small so that* $\sigma(0) < 2\sigma_h(0)$. *Further assume that*

$$C(t)h^{1-d/2+d/r}\phi(2\sigma_h(t),t) \le \tfrac{1}{2}\sigma_h(t) \tag{14.43}$$

for all $0 \le t \le T$. *Then*

$$\sigma(t) < 2\sigma_h(t) \tag{14.44}$$

for all $0 \le t \le T$. *Thus a smooth solution to the Navier-Stokes equations exists on the interval* $[0,T]$ *as long as (14.43) holds for a given* h.

To prove this, it suffices to do a type of induction on T. It follows for $t = 0$ by our assumption. By the continuity of σ and σ_h, the inequality must persist for some small interval $0 \le t \le \delta$. Suppose that δ is the maximum value for which (14.44) holds on $0 \le t \le \delta$. But then the continuity of σ again, (14.42), and the fact that ϕ is strictly increasing in σ imply that

$$\begin{aligned}\sigma(\delta) = \sup_{0\le t<\delta}\sigma(t) &\le \sup_{0\le t<\delta}\sigma_h(t) + C(t)h^{1-d/2+d/r}\phi(\sigma(t),t)\\ &\le \sup_{0\le t<\delta}\sigma_h(t) + C(t)h^{1-d/2+d/r}\phi(2\sigma_h(t),t)\\ &\le \sup_{0\le t<\delta}\sigma_h(t) + \tfrac{1}{2}\sigma_h(t) = \tfrac{3}{2}\sigma_h(\delta) < 2\sigma_h(\delta)\end{aligned} \tag{14.45}$$

Thus the strict inequality holds up to $t = \delta$, and continuity again shows that it must persist a bit further, unless $\delta = T$. This concludes the proof.

The estimate (14.43) is computable for any given boundary value problem. If it fails to hold at some point, it would be necessary to reduce h and recompute. It is always possible, for any σ, to pick h so that, say

$$3C(t)h^{1-d/2+d/r}\phi(2\sigma,t) \le \sigma \tag{14.46}$$

as the next value of h to try.

More precisely, we continue to march along in time until we have

$$C(t)h^{1-d/2+d/r}\phi(2\sigma_h(t),t) = \tfrac{1}{2}\sigma_h(t) \tag{14.47}$$

for some value of $t = t_0$ but (14.43) fails for larger $t > t_0$. Now we reduce h and re-compute. One would expect that (14.43) would still hold out to t_0 for the smaller h, but there is no guarantee.

In the typical case, (14.43) would hold further, to some point $t_1 > t_0$. In this case, we have extended the interval of validity of the Navier-Stokes equations further to t_1, and we can continue as we like indefinitely in this way. Of course, it might be that the situation gets worse and (14.43) only holds out to $t_1 < t_0$, and refining h again makes it even worse. This is what would be expected if the solution did not exist for such large values of t.

14.3 Compatibility Conditions

For the Navier-Stokes equations, there are compatibility conditions like those found for the heat equation in Section 7.2. Here we refer to these as "local" compatibility conditions since they can be determined by purely local considerations. We begin with a description of these. However, there are also non-local compatibility conditions for the Navier-Stokes equations, and we describe them subsequently.

14.3.1 Local Compatibility Conditions

There are local *compatibility conditions* for the boundary and initial data for the Navier-Stokes equations similar to the ones for the heat equation in order to have a smooth solution. These can be derived again from the observation that that the values of u on the spatial boundary have been specified twice at $t = 0$. The first condition is simply

$$\mathbf{u_0}(\mathbf{x}) = \boldsymbol{\gamma}(\mathbf{x}) \quad \forall \mathbf{x} \in \partial\Omega. \tag{14.48}$$

Additional conditions arrise by using the differential equation at $t = 0$ and for $x \in \partial\Omega$, but we post-pone temporarily deriving one of these. However, we find a new type of condition, namely,

$$\nabla\cdot \mathbf{u_0} = 0. \tag{14.49}$$

Although this condition is quite obvious, it does pose a significant constraint on the initial data.

If either of these compatibilities are not satisfied by the data (or by the approximation scheme), wild oscillations will result near $t = 0$ and $x \in \partial\Omega$. In such a nonlinear problem, this can cause completely wrong results to occur.

Another condition that arises due to the incompressibility (or divergence-free) condition is the following:

$$\oint_{\partial\Omega} \mathbf{n}\cdot\boldsymbol{\gamma} = 0. \tag{14.50}$$

This arises simply from (2.8), and it says that the amount of fluid coming into the domain must balance the amount of fluid going out of the domain: the net mass flux into the domain is zero. If this compatibility condition is violated, then the solution can clearly not have divergence zero.

The compatibility conditions (14.48) and (14.49) do not have to be satisfied for the Navier-Stokes equations (14.1) to be well-posed in the usual sense. There is a unique solution in any case, but the physical model may be incorrect as a result if it is supposed to have a smooth solution. Compatibility conditions are a very subtle form of constraint on model quality.

The compatibility conditions (14.48) and (14.49) are described in terms of local differential-algebraic constraints. However, in Section 14.3.2 we will see that such compatibility conditions can lead to global constraints that may be extremely hard to verify or satisfy in practice.

14.3.2 A nonlocal compatibility condition [93]

Simply applying the first equation in (14.1) on $\partial\Omega$ at $t = 0$ we find

$$-\Delta \mathbf{u_0} + \nabla p_0 = -R \left(\boldsymbol{\gamma} \cdot \nabla \mathbf{u_0} + \boldsymbol{\gamma}'\right) \text{ on } \partial\Omega \tag{14.51}$$

where p_0 denotes the initial pressure. Since p_0 is not part of the data, it might be reasonable to assume that the pressure initially would just adjust to insure smoothness of the system, i.e., satisfaction of (14.51), which we can re-write as

$$\nabla p_0 = \Delta \mathbf{u_0} - R \left(\boldsymbol{\gamma} \cdot \nabla \mathbf{u_0} + \boldsymbol{\gamma}'\right) \text{ on } \partial\Omega \tag{14.52}$$

However, taking the divergence of the first equation in (14.1) at $t = 0$ we find

$$\Delta p_0 = -R \nabla\cdot \left(\mathbf{u_0} \cdot \nabla \mathbf{u_0}\right) \text{ in } \Omega. \tag{14.53}$$

Thus p_0 must satisfy a Laplace equation with the full gradient specified on the bounary by (14.52). This is an over-specified system (one too many boundary conditions, see Section 15.1), so not all $\mathbf{u_0}$ will satisfy it. Note that

$$\nabla\cdot \left(\mathbf{v} \cdot \nabla \mathbf{v}\right) = \sum_{i,j} v_{i,j} v_{j,i} \text{ in } \Omega \tag{14.54}$$

if $\nabla\cdot \mathbf{v} = \mathbf{0}$.

The only simple way to satisfy both (14.52) and (14.53) is to have $\mathbf{u_0} \equiv \mathbf{0}$ and $\boldsymbol{\gamma} = \boldsymbol{\gamma}' = \mathbf{0}$, that is to start with the fluid at rest. For $R > 0$, it is easy to see that the system given by (14.52) and (14.53) is over-specified since $\boldsymbol{\gamma}'$ can be chosen arbitrarily.

14.4 Iterated Penalty Method

The itereated penalty method can be used to enforce the incompressiblity constraint as was done for the Stokes system.

14.4.1 Iterated Penalty Method for Navier-Stokes

The iterated penalty method (12.40) (with $\rho' = \rho$) for (14.10) takes the form

$$\begin{aligned}\tilde{a}\left(\mathbf{u}^{\ell,n}, \mathbf{v}\right) + \rho \left(\nabla\cdot \mathbf{u}^{\ell,n}, \nabla\cdot \mathbf{v}\right)_{L^2} &= -a\left(\boldsymbol{\gamma}, \mathbf{v}\right) - R\, c\left(\mathbf{u}^{\ell-1}, \mathbf{u}^{\ell-1}, \mathbf{v}\right) \\ &\quad + \tau \left(\mathbf{u}^{\ell-1}, \mathbf{v}\right)_{\underset{\sim}{L^2}} - \rho \left(\nabla\cdot \left(\mathbf{w}^{\ell,n} + \boldsymbol{\gamma}\right), \nabla\cdot \mathbf{v}\right)_{L^2} \quad \forall \mathbf{v} \in \mathbf{V} \\ \mathbf{w}^{\ell,n+1} &= \mathbf{w}^{\ell,n} - \rho \left(\mathbf{u}^{\ell,n} + \boldsymbol{\gamma}\right)\end{aligned} \tag{14.55}$$

where either $p^{\ell,0} = 0$ or (i.e. $\mathbf{w}^{\ell,0} = 0$) or $\mathbf{w}^{\ell,0} = \mathbf{w}^{\ell-1,N}$ where N is the final value of n at time-step $\ell - 1$. If for some reason $p^{\ell} = p^{\ell,n} = P_{\Pi_h} \nabla\cdot \mathbf{w}^{\ell,n}$ were desired, it could be computed separately.

For example, algorithm (12.41) could be used to compute

$$\begin{aligned}\tilde{a}\left(\mathbf{z}^n,\mathbf{v}\right)+\rho\left(\nabla\cdot\mathbf{z}^n,\nabla\cdot\mathbf{v}\right)_{L^2}=&-\rho\left(\nabla\cdot\left(\boldsymbol{\zeta}^n-\mathbf{w}^{\ell,\mathbf{n}}\right),\nabla\cdot\mathbf{v}\right)_{L^2}\quad\forall\mathbf{v}\in\mathbf{V}\\ \boldsymbol{\zeta}^{n+1}=&\boldsymbol{\zeta}^n-\rho\left(\mathbf{z}^n-\mathbf{w}^{\ell,n}\right)\end{aligned}\tag{14.56}$$

starting with, say, $\boldsymbol{\zeta}^0\equiv 0$. Then $\mathbf{z}^n$ will converge to $\mathbf{z}\in\mathbf{V}_h$ satisfying $\nabla\cdot\mathbf{z}=P_{\Pi_h}\nabla\cdot\mathbf{w}^{\ell,n}=p^{\ell,n}$. Note that (14.56) requires the same system to be solved as for computing $\mathbf{u}^{\ell,n}$ in (14.55), so very little extra code or data-storage is required.

The potential difficulty caused by having inhomogeneous boundary data can be seen for high-order finite elements. For simplicity, consider the two-dimensional case. Let W_h^k denote piecewise polynomials of degree k on a triangular mesh, and let V_h^k denote the subspace of W_h^k consisting of functions that vanish on the boundary. Let $\mathbf{V}_h=V_h^k\times V_h^k$, and let $\Pi_h=\nabla\cdot\mathbf{V}_h$. Then each $q\in\Pi_h$ is continuous at all boundary singular [40, Section 12.6] vertices, σ_i, in the mesh. On the other hand, inhomogeneous boundary conditions will require the introduction of some $\boldsymbol{\gamma}\in W_h^k\times W_h^k$. It is known [158] that $\nabla\cdot\left(W_h^k\times W_h^k\right)$ consists of all piecewise polynomials of degree $k-1$, a larger space than Π_h if boundary singular vertices are present in the mesh. On the other hand, if there are no boundary singular vertices, there is no need to form the projection since $\Pi_h=\left\{q\in\nabla\cdot\left(W_h^k\times W_h^k\right)\ :\ \int_\Omega q(x)\,dx=0\right\}$ in this case.

A more complex time-stepping scheme could be based on the variational equations

$$\begin{aligned}a\left(\mathbf{u}^\ell,\mathbf{v}\right)+b\left(\mathbf{v},p^\ell\right)+R\,c\left(\mathbf{u}^{\ell-1},\mathbf{u}^\ell,\mathbf{v}\right)+\frac{R}{\Delta t}\left(\mathbf{u}^\ell-\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}&=0,\\ b\left(\mathbf{u}^\ell,q\right)&=0,\end{aligned}\tag{14.57}$$

in which the nonlinear term has been approximated in such a way that the linear algebraic problem changes at each time step. It takes the form (12.18) with a form $\tilde{a}(\cdot,\cdot)$ given by

$$\tilde{a}\left(\mathbf{u},\mathbf{v};\mathbf{U}\right)=a\left(\mathbf{u},\mathbf{v}\right)+\int_\Omega\left(\tau\mathbf{u}\cdot\mathbf{v}+\mathbf{U}\cdot\nabla\mathbf{u}\cdot\mathbf{v}\right)\,d\mathbf{x}\tag{14.58}$$

where $\mathbf{U}=R\mathbf{u}^n$ arises from linearizing the nonlinear term.

Even though the addition of the $\mathbf{U}$ term makes it non-symmetric, $\tilde{a}(\cdot,\cdot)$ will be coercive for τ sufficiently large (i.e., for $\Delta t/R$ sufficiently small). In fact, when $\nabla\cdot\mathbf{U}\equiv 0$ then integrating by parts yields

$$\begin{gathered}\int_\Omega\mathbf{U}\cdot\nabla\mathbf{v}\cdot\mathbf{v}\,d\mathbf{x}=\int_\Omega\sum_{i,j=1}^d U_iv_{j,i}v_j\,d\mathbf{x}=\\ \int_\Omega\sum_{i,j=1}^d U_i\frac{1}{2}\left(v_j^2\right)_{,i}\,d\mathbf{x}=-\frac{1}{2}\int_\Omega\sum_{i,j=1}^d U_{i,i}v_j^2\,d\mathbf{x}=0\end{gathered}\tag{14.59}$$

for all $v\in V$ so that

$$\alpha\|\mathbf{v}\|_V^2\le\tilde{a}(\mathbf{v},\mathbf{v})\quad\forall\mathbf{v}\in V\tag{14.60}$$

for the same choice of $\alpha>0$ as before. Of course, $\tilde{a}(\cdot,\cdot)$ is continuous:

$$\tilde{a}(\mathbf{v},\mathbf{w})\le C_a\|\mathbf{v}\|_V\|\mathbf{w}\|_V\quad\forall\mathbf{v},\mathbf{w}\in V\tag{14.61}$$

but now C_a depends on both τ and $\mathbf{U}$.

14.4.2 Convergence and stopping criteria

The convergence properties of (12.30) follow from [40, Chapter 13].

Theorem 14.2 *Suppose that the forms (12.18) satisfy (12.13) and (12.22). for V_h and $\Pi_h = \mathcal{D}V_h$. Then the algorithm (12.30) converges for any $0 < \rho < 2\rho'$ for ρ' sufficiently large. For the choice $\rho = \rho'$, (12.30) converges geometrically with a rate given by*

$$C_a\left(\frac{1}{\beta}+\frac{C_a}{\alpha\beta}\right)^2 \Big/ \rho' .$$

The following stopping criterion follows from [40, Chapter 13].

Theorem 14.3 *Suppose that the forms (12.18) satisfy (12.13) and (12.22). for V_h and $\Pi_h = \mathcal{D}V_h$. Then the errors in algorithm (12.30) can be estimated by*

$$\|\mathbf{u}^n - \mathbf{u}_h\|_V \le \left(\frac{1}{\beta}+\frac{C_a}{\alpha\beta}\right)\|\mathcal{D}\mathbf{u}^n - P_\Pi\, G\|_\Pi$$

and

$$\|p^n - p_h\|_\Pi \le \left(\frac{C_a}{\beta}+\frac{C_a^2}{\alpha\beta}+\rho' C_b\right)\|\mathcal{D}\mathbf{u}^n - P_\Pi\, G\|_\Pi.$$

When $G(q) = -b(\boldsymbol{\gamma}, q)$, then $P_\Pi\, G = -P_\Pi\, \mathcal{D}\boldsymbol{\gamma}$ and since $\mathcal{D}\mathbf{u}^n \in \Pi_h$,

$$\begin{aligned}\|\mathcal{D}\mathbf{u}^n - P_\Pi\, G\|_\Pi &= \|P_\Pi\, \mathcal{D}\left(\mathbf{u}^n + \boldsymbol{\gamma}\right)\|_\Pi \\ &\le \|\mathcal{D}\left(\mathbf{u}^n + \boldsymbol{\gamma}\right)\|_\Pi,\end{aligned} \tag{14.62}$$

and the latter norm is easier to compute, avoiding the need to compute $P_\Pi\, G$. We formalize this observation in the following result.

Corollary 14.1 *Under the conditions of Theorem (14.3) the errors in algorithm (12.30) can be estimated by*

$$\|\mathbf{u}^n - \mathbf{u}_h\|_V \le \left(\frac{1}{\beta}+\frac{C_a}{\alpha\beta}\right)\|\mathcal{D}\left(\mathbf{u}^n + \boldsymbol{\gamma}\right)\|_\Pi$$

and

$$\|p^n - p_h\|_\Pi \le \left(\frac{C_a}{\beta}+\frac{C_a^2}{\alpha\beta}+\rho' C_b\right)\|\mathcal{D}\left(\mathbf{u}^n + \boldsymbol{\gamma}\right)\|_\Pi.$$

14.5 Implicit time-stepping

A fully implicit time-stepping scheme for the Navier-Stokes equations can be defined as follows. Expressed in variational form, it is

$$\begin{aligned}a\left(\mathbf{u}^\ell, \mathbf{v}\right) + b\left(\mathbf{v}, p^\ell\right) + R\, c\left(\mathbf{u}^\ell, \mathbf{u}^\ell, \mathbf{v}\right) + \frac{R}{\Delta t}\left(\mathbf{u}^\ell - \mathbf{u}^{\ell-1}, \mathbf{v}\right)_{\underset{\sim}{L}^2} &= 0, \\ b\left(\mathbf{u}^\ell, q\right) &= 0,\end{aligned} \tag{14.63}$$

where $\mathbf{v}$ varies over all V (or V_h) and q varies over all Π (or Π_h) and Δt denotes the time-step size. More efficient time-stepping schemes will take a similar form, such as the backwards differentiation schemes. In particular, (14.63) is the first-order backwards differentiation scheme.

At each time step, one has a problem to solve for $(\mathbf{u}^\ell, p^\ell)$ with the form $\tilde{a}(\cdot,\cdot)$ as in (14.11) where the constant $\tau = R/\Delta t$. It takes the form

$$\begin{aligned} \tilde{a}\left(\mathbf{u}^\ell,\mathbf{v}\right) + b\left(\mathbf{v},p^\ell\right) + R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{v}\right) &= \tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}, \\ b\left(\mathbf{u}^\ell,q\right) &= 0. \end{aligned} \tag{14.64}$$

However, (14.64) is nonlinear, so an algorithm must be chosen to linearize it. One of the simplest is the fixed-point iteration, which takes the form:

$$\begin{aligned} \tilde{a}\left(\mathbf{u}^{\ell,k},\mathbf{v}\right) + b\left(\mathbf{v},p^{\ell,k}\right) &= -R\,c\left(\mathbf{u}^{\ell,k-1},\mathbf{u}^{\ell,k-1},\mathbf{v}\right) + \tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}, \\ b\left(\mathbf{u}^{\ell,k},q\right) &= 0. \end{aligned} \tag{14.65}$$

This iteration can be started with an extrapolated value, e.g., $\mathbf{u}^{\ell,0} := 2\mathbf{u}^{\ell-1} - \mathbf{u}^{\ell-2}$, and once convergence is achieved, we set $\mathbf{u}^\ell = \mathbf{u}^{\ell,k}$.

Note that we have $\mathbf{u}^{\ell,k} = \boldsymbol{\gamma}$ on $\partial\Omega$, that is, $\mathbf{u}^{\ell,k} = \mathbf{u}_0^{\ell,k} + \boldsymbol{\gamma}$ where $\mathbf{u}_0^{\ell,k} \in V$. The variational problem for $\mathbf{u}^{\ell,k}$ can be written: Find $\mathbf{u}_0^{\ell,k} \in V$ and $p \in \Pi$ such that

$$\begin{aligned} \tilde{a}\left(\mathbf{u}_0^{\ell,k},\mathbf{v}\right) + b\left(\mathbf{v},p\right) &= -\tilde{a}\left(\boldsymbol{\gamma},\mathbf{v}\right) \; - R\,c\left(\mathbf{u}^{\ell,k-1},\mathbf{u}^{\ell,k-1},\mathbf{v}\right) + \tau\left(\mathbf{u}^{\ell},\mathbf{v}\right)_{\underset{\sim}{L}^2} \quad \forall \mathbf{v} \in V \\ b\left(\mathbf{u}_0^{\ell,k},q\right) &= -\,b\left(\boldsymbol{\gamma},q\right) \quad \forall q \in \Pi. \end{aligned} \tag{14.66}$$

This is of the form (12.18) with

$$\begin{aligned} F(\mathbf{v}) &= -\tilde{a}\left(\boldsymbol{\gamma},\mathbf{v}\right) - R\,c\left(\mathbf{u}^{\ell,k-1},\mathbf{u}^{\ell,k-1},\mathbf{v}\right) + \tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2} \quad \forall \mathbf{v} \in V \\ G\left(q\right) &= -\,b\left(\boldsymbol{\gamma},q\right) \quad \forall q \in \Pi. \end{aligned} \tag{14.67}$$

Again, the inhomogeneous boundary data $\boldsymbol{\gamma}$ appears in both right-hand sides, F and G.

14.5.1 Stability of the exact solution

The nonlinear problem (14.64) has excellent stability properties. To see why, let us assume for simplicity that the boundary data $\boldsymbol{\gamma}$ has been extended to Ω so that $\nabla\cdot\boldsymbol{\gamma} = 0$. Then we let $\mathbf{v} = \mathbf{u}^\ell - \boldsymbol{\gamma}$ in (14.64), and we find

$$\tilde{a}\left(\mathbf{u}^\ell,\mathbf{u}^\ell - \boldsymbol{\gamma}\right) = -R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{u}^\ell - \boldsymbol{\gamma}\right) + \tau\left(\mathbf{u}^{\ell-1},\mathbf{u}^\ell - \boldsymbol{\gamma}\right)_{\underset{\sim}{L}^2}. \tag{14.68}$$

Let us write $\mathbf{u}_0^\ell=\mathbf{u}^\ell-\boldsymbol{\gamma}$, etc., and simplifying a bit, we have

$$\begin{aligned}\tilde a\left(\mathbf{u}^\ell,\mathbf{u}^\ell\right)&=\tilde a\left(\mathbf{u}^\ell,\boldsymbol{\gamma}\right)-R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{u}_0^\ell\right)+\tau\left(\mathbf{u}^{\ell-1},\mathbf{u}_0^\ell\right)_{\underset{\sim}{L}^2}\\&=\tilde a\left(\mathbf{u}^\ell,\boldsymbol{\gamma}\right)-R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{u}_0^\ell\right)+\tau\left(\mathbf{u}^{\ell-1},\mathbf{u}^\ell\right)_{\underset{\sim}{L}^2}-\tau\left(\mathbf{u}^{\ell-1},\boldsymbol{\gamma}\right)_{\underset{\sim}{L}^2}\\&=\tilde a\left(\mathbf{u}^\ell,\boldsymbol{\gamma}\right)-R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{u}^\ell\right)+R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\boldsymbol{\gamma}\right)+\tau\left(\mathbf{u}^{\ell-1},\mathbf{u}^\ell\right)_{\underset{\sim}{L}^2}-\tau\left(\mathbf{u}^{\ell-1},\boldsymbol{\gamma}\right)_{\underset{\sim}{L}^2}\\&=L\left(\boldsymbol{\gamma}\right)-R\oint_{\partial\Omega}\boldsymbol{\gamma}\cdot\mathbf{n}\left|\boldsymbol{\gamma}\right|^2\,dx+\tau\left(\mathbf{u}^{\ell-1},\mathbf{u}^\ell\right)_{\underset{\sim}{L}^2}\end{aligned}\tag{14.69}$$

where

$$L\left(\boldsymbol{\gamma}\right):=\tilde a\left(\mathbf{u}^\ell,\boldsymbol{\gamma}\right)+R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\boldsymbol{\gamma}\right)-\tau\left(\mathbf{u}^{\ell-1},\boldsymbol{\gamma}\right)_{\underset{\sim}{L}^2}\tag{14.70}$$

since $c\left(\mathbf{u}^\ell,\mathbf{u}_0^\ell,\mathbf{u}_0^\ell\right)=0$. Note that

$$L\left(\mathbf{v}\right)=\tilde a\left(\mathbf{u}^\ell,\mathbf{v}\right)+R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{v}\right)-\tau\left(\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}=0\tag{14.71}$$

for all $\mathbf{v}\in V$. Thus the value of $L\left(\boldsymbol{\gamma}\right)$ does not depend on the representation of $\boldsymbol{\gamma}$ in the interior. Therefore, the value of $L\left(\boldsymbol{\gamma}\right)$ must depend only on the $\boldsymbol{\gamma}$ on the boundary in some way. Integrating by parts, we find that

$$L\left(\boldsymbol{\gamma}\right)=\oint_{\partial\Omega}\left(\boldsymbol{\gamma}\cdot\mathbf{n}\right)\sigma_{\mathbf{n}}\,dx\tag{14.72}$$

where $\sigma_{\mathbf{n}}$ is essentially the normal stress associated with the problem (14.64). We can think of $\sigma_{\mathbf{n}}=\mathcal{N}\boldsymbol{\gamma}$ where $\mathcal{N}=\mathcal{N}(\tau,\mathbf{u}^{\ell-1})$ represents a type of (non-linear) Dirichlet-to-Neumann map.

14.5.2 Convergence of fixed-point iteration

The convergence of the iterative scheme (14.65) can be analyzed as follows. Subtracting two consecutive versions of (14.65), we find the following formula for $\mathbf{e}^k:=\mathbf{u}^{\ell,k}-\mathbf{u}^{\ell,k-1}$:

$$\begin{aligned}\tilde a\left(\mathbf{e}^k,\mathbf{e}^k\right)&=R\Big(c\left(\mathbf{u}^{\ell,k-1},\mathbf{u}^{\ell,k-1},\mathbf{e}^k\right)-c\left(\mathbf{u}^{\ell,k-2},\mathbf{u}^{\ell,k-2},\mathbf{e}^k\right)\Big)\\&=R\Big(c\left(\mathbf{u}^{\ell,k-1},\mathbf{u}^{\ell,k-1},\mathbf{e}^k\right)-c\left(\mathbf{u}^{\ell,k-2},\mathbf{u}^{\ell,k-1},\mathbf{e}^k\right)+\\&\qquad c\left(\mathbf{u}^{\ell,k-2},\mathbf{u}^{\ell,k-1},\mathbf{e}^k\right)-c\left(\mathbf{u}^{\ell,k-2},\mathbf{u}^{\ell,k-2},\mathbf{e}^k\right)\Big)\\&=R\Big(c\left(\mathbf{e}^{k-1},\mathbf{u}^{\ell,k-1},\mathbf{e}^k\right)+c\left(\mathbf{u}^{\ell,k-2},\mathbf{e}^{k-1},\mathbf{e}^k\right)\Big).\end{aligned}\tag{14.73}$$

Estimating the terms with the "c" form in them, we find (for suitable $p>2$)

$$\begin{aligned}\tilde a\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right) \le& CR\|\mathbf{u}^{\ell,k-1}\|_{W^1_p(\Omega)}\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)^{1/2}\left(a\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right)\right)^{1/2}\\ &+CR\|\mathbf{u}^{\ell,k-2}\|_{L^\infty(\Omega)}\left(a\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)\right)^{1/2}\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right)^{1/2}\\ \le&\tfrac12\left(CR\|\mathbf{u}^{\ell,k-1}\|_{W^1_p(\Omega)}\right)^2\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)+\tfrac12 a((\mathbf{e}^\ell,\mathbf{e}^\ell)\\ &+\frac{1}{2\tau}\left(CR\|\mathbf{u}^{\ell,k-2}\|_{L^\infty(\Omega)}\right)^2 a\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)+\frac{\tau}{2}\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right)\\ \le&\frac{CR^2}{\tau}\left(\|\mathbf{u}^{\ell,k-1}\|^2_{W^1_p(\Omega)}+\|\mathbf{u}^{\ell,k-2}\|^2_{L^\infty(\Omega)}\right)\tilde a\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)+\tfrac12\tilde a\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right)\end{aligned}\tag{14.74}$$

where "C" denotes a generic constant. Therefore

$$\tilde a\left(\mathbf{e}^\ell,\mathbf{e}^\ell\right)\le\frac{CR^2}{\tau}\left(\|\mathbf{u}^{\ell,k-1}\|^2_{W^1_p(\Omega)}+\|\mathbf{u}^{\ell,k-2}\|^2_{L^\infty(\Omega)}\right)\tilde a\left(\mathbf{e}^{\ell-1},\mathbf{e}^{\ell-1}\right)\tag{14.75}$$

Therefore the fixed-point iteration for solving the nonlinear equations (14.64) for the time-stepping is convergent provided τ is large enough and $\|\mathbf{u}^{\ell,k-2}\|_{L^\infty(\Omega)}$ and $\|\mathbf{u}^{\ell,k-1}\|_{W^1_p(\Omega)}$ stay bounded. Note that the parameter $\frac{R^2}{\tau}$ appearing in (14.75) is equal to $R\Delta t$.

14.5.3 Stability of fixed-point iteration

In practice, the convergence criterion

$$CR^2\left(\|\mathbf{u}^{\ell,k-1}\|^2_{W^1_p(\Omega)}+\|\mathbf{u}^{\ell,k-2}\|^2_{L^\infty(\Omega)}\right)<\tau\tag{14.76}$$

for convergence of fixed point iteration given in (14.75) can be too restrictive. It may not be necessary to solve exactly to achieve the stability promised in Section 14.5.1. To simplify notation, let $\mathbf{v}_0^k=\mathbf{u}_0^{\ell,k}$ and $\mathbf{v}^k=\mathbf{v}_0^k+\gamma=\mathbf{u}^{\ell,k}$. Then setting $\mathbf{v}=\mathbf{v}_0^k$ in (14.66) yields

$$\begin{aligned}\tilde a\left(\mathbf{v}_0^k,\mathbf{v}_0^k\right)&=-\tilde a\left(\boldsymbol{\gamma},\mathbf{v}_0^k\right)\ -R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\mathbf{v}_0^k\right)+\tau\left(\mathbf{u}^\ell,\mathbf{v}_0^k\right)_{\underset{\sim}{L}^2}\\ &=-\tilde a\left(\boldsymbol{\gamma},\mathbf{v}_0^k\right)\ -R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\mathbf{v}_0^k-\mathbf{v}^{k-1}\right)\\ &\qquad+R\oint_{\partial\Omega}\boldsymbol{\gamma}\cdot\mathbf{n}\,|\boldsymbol{\gamma}|^2\,dx+\tau\left(\mathbf{u}^\ell,\mathbf{v}_0^k\right)_{\underset{\sim}{L}^2}\\ &=-\tilde a\left(\boldsymbol{\gamma},\mathbf{v}_0^k\right)\ -R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\mathbf{v}_0^k-\mathbf{v}_0^{k-1}\right)\\ -R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\boldsymbol{\gamma}\right)&+R\oint_{\partial\Omega}\boldsymbol{\gamma}\cdot\mathbf{n}\,|\boldsymbol{\gamma}|^2\,dx+\tau\left(\mathbf{u}^\ell,\mathbf{v}_0^k\right)_{\underset{\sim}{L}^2}\end{aligned}\tag{14.77}$$

Suppose that the following condition holds:

$$\left|c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\mathbf{v}_0^k-\mathbf{v}_0^{k-1}\right)\right|\le a(\mathbf{v}_0^k,\mathbf{v}_0^k)/R\,.\tag{14.78}$$

Then

$$\begin{aligned}\tau\left(\mathbf{v}_0^k,\mathbf{v}_0^k\right) =& -\tilde{a}\left(\boldsymbol{\gamma},\mathbf{v}_0^k\right) \; - R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\mathbf{v}_0^k-\mathbf{v}_0^{k-1}\right) - a\left(\mathbf{v}_0^k,\mathbf{v}_0^k\right)\\ & - R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\boldsymbol{\gamma}\right) + R\oint_{\partial\Omega}\boldsymbol{\gamma}\cdot\mathbf{n}\,|\boldsymbol{\gamma}|^2\,dx + \tau\left(\mathbf{u}^\ell,\mathbf{v}_0^k\right)_{\underset{\sim}{L}^2}\\ \le& -\tilde{a}\left(\boldsymbol{\gamma},\mathbf{v}_0^k\right) \; - R\,c\left(\mathbf{v}^{k-1},\mathbf{v}^{k-1},\boldsymbol{\gamma}\right)\\ & + R\oint_{\partial\Omega}\boldsymbol{\gamma}\cdot\mathbf{n}\,|\boldsymbol{\gamma}|^2\,dx + \tau\left(\mathbf{u}^\ell,\mathbf{v}_0^k\right)_{\underset{\sim}{L}^2}\end{aligned} \tag{14.79}$$

14.6 The slot problem

Let us now consider an example with an industrial application. For simplicity, we consider a two-dimensional version of this problem [117]. An instrument for measuring rheometic properties of non-Newtonian fluids is based on flow in a channel over a slot. It is based on the assumption that the flow is steady, i.e., that the Reynolds number is not too large. The normal-stress difference between the top of the channel over the slot and the bottom of the slot is related to a certain parameter that characterizes non-Newtonian fluids. The normal-stress difference is most useful if it corresponds to a small Reynolds number (i.e., slow flow), but then the instrument must be run more slowly. Running it faster involves a higher Reynolds number, and a correction must be made in this case. The correction involves knowing the normal-stress difference for a Newtonian fluid [117]. Here we show how this might be accurately calculated.

The normal stress involves derivative information, and this is typically less accurately calculated than, say, the velocity itself. Moreover, evaluating any quantity at a point, or on a surface, is less accurate than a globally defined quantity. Often stresses or fluxes can be calculated by a global integral with the appropriate test function. For example, the normal stress on a surface Γ is equal to

$$N(\mathbf{v}) := a_\epsilon(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) \tag{14.80}$$

where a_ϵ was defined in (12.10), provided that $\mathbf{v}=\mathbf{n}$ on Γ, and $\mathbf{v}=\mathbf{0}$ on $\partial\Omega\backslash\Gamma$. To see this, one simply integrates by parts. This result holds only for the exact solution $(\mathbf{u},p)$. However, if we apply the same idea for the discrete solution, i.e.,

$$N_h(\mathbf{v}) := a_\epsilon(\mathbf{u}_h,\mathbf{v}) + b(\mathbf{v},p_h) \tag{14.81}$$

then the error

$$N(\mathbf{v}) - N_h(\mathbf{v}) := a_\epsilon(\mathbf{u}-\mathbf{u}_h,\mathbf{v}) + b(\mathbf{v},p-p_h) \tag{14.82}$$

can be bounded by the global error in the approximation.

It may not seem so easy to define a function that has normal component equal to one on part of the boundary and jumps to zero abruptly. Indeed such a function must not be in $H^1(\Omega)$, but it can be in $W_p^1(\Omega)$ for any $p<2$. For example, suppose that $\partial\Omega$ is the x-axis

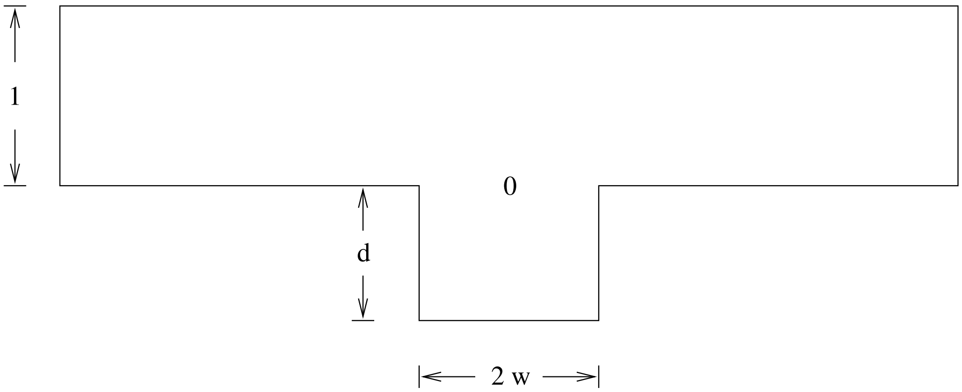

Figure 14.1: Domain configuration and notation for the slot problem.

and Γ is the positive half line. Then

$$\mathbf{v}(x,y) := \begin{cases} (0, -\cos\theta) & \text{for} \quad x \geq 0 \\ (0,0) & \text{for} \quad x \leq 0 \end{cases} \tag{14.83}$$

has this property, where (r,θ) are polar coordinates. Since

$$\cos\theta = \frac{x}{r} = \frac{x}{\sqrt{x^2+y^2}} = \frac{\operatorname{sign}(x)}{\sqrt{1+(y/x)^2}}$$

we can use this idea to construct a general function with the desired properties. Note that the value of $N(\mathbf{v})$ or $N_h(\mathbf{v})$ is independent of the choice of $\mathbf{v}$. Given two functions $\mathbf{v}_1$ and $\mathbf{v}_2$, then $N(\mathbf{v}_1) - N(\mathbf{v}_2) = N(\mathbf{v}_1 - \mathbf{v}_2) = 0$ since $\mathbf{v}_1 - \mathbf{v}_2$ vanishes on the boundary. (A similar result holds for N_h as well.)

For the geometry of the slot problem (Figure 14.1), define a vector function whose x-component is zero and whose y-component, denoted v, is given by the following expression:

$$\begin{aligned} v(x,y) := \frac{y+d}{1+d}\left(1+\left(\frac{1-y}{x+w}\right)^2\right)^{-1/2}\left(1+\left(\frac{1-y}{x-w}\right)^2\right)^{-1/2} \\ +\frac{1-y}{1+d}\left(1+\left(\frac{y+d}{x+w}\right)^2\right)^{-1/2}\left(1+\left(\frac{y+d}{x-w}\right)^2\right)^{-1/2} \end{aligned} \tag{14.84}$$

for $|x| < w$ and $-d < y < 1$, and zero elsewhere. Then $\mathbf{v}(x,y) := (0, v(x,y))$ has the desired property that

$$N(\mathbf{v}) := a_\epsilon(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) \tag{14.85}$$

is the normal stress difference between the two segments $\{(x,1) \ : \ |x| \leq w\}$ on the top and $\{(x,-d) \ : \ |x| \leq w\}$ on the bottom.

14.7 Free-boundary between two fluids

We consider a problem in which there are two immiscible fluids, such as air and water, for which the boundary between the two fluids is unknown. We assume that we have $\mathbf{u} = \mathbf{g}$ on $\partial\Omega$, that is, $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_\mathbf{g}$ where $\mathbf{u}_0 \in V$. The variational problem for $\mathbf{u}$ can be written: Find $\mathbf{u}_0 \in V$ and $p \in \Pi$ such that

$$\begin{aligned} a\left(\mathbf{u}_0, \mathbf{v}\right) + b\left(\mathbf{v}, p\right) &= -a\left(\mathbf{u}_\mathbf{g}, \mathbf{v}\right) - R\, c\left(\mathbf{u}_0 + \mathbf{u}_\mathbf{g}, \mathbf{u}_0 + \mathbf{u}_\mathbf{g}, \mathbf{v}\right) \\ &\qquad + \sigma \oint_\gamma \kappa \mathbf{v} \cdot \mathbf{n}\, ds \quad \forall \mathbf{v} \in V \\ b\left(\mathbf{u}_0, q\right) &= -b\left(\mathbf{u}_\mathbf{g}, q\right) \quad \forall q \in \Pi. \end{aligned} \tag{14.86}$$

Here γ denotes the interface between the two fluids, κ denotes its curvature, and $\mathbf{n}$ the outward normal.

We will consider level-set methods in which there is a function ϕ whose zero set is γ. For simplicity, we will assume that $\nabla\phi$ is the unit normal to γ, that is, we suppose that $\mathbf{n} = \nabla\phi$. Then the coupling term involving the curvature κ may be written

$$\begin{aligned} \oint_\gamma \kappa \mathbf{v} \cdot \mathbf{n}\, ds &= \oint_\gamma -(\nabla\cdot \mathbf{n})\mathbf{v} \cdot \mathbf{n}\, ds = \oint_\gamma -((\nabla\cdot \mathbf{n})\mathbf{v}) \cdot \mathbf{n}\, ds \\ &= \int_\Omega -H(\phi)\nabla\cdot \left(\mathbf{v}\nabla\cdot \mathbf{n}\right) dx \end{aligned} \tag{14.87}$$

where $H(t)$ is a Heavyside function that is $-\frac{1}{2}$ for negative t and $+\frac{1}{2}$ for positive t. We will also use the notation H to denote a regularized approximation to the Heavyside function, so that

$$\begin{aligned} \oint_\gamma \kappa \mathbf{v} \cdot \mathbf{n}\, ds &\approx \int_\Omega -H(\phi)\nabla\cdot \left(\mathbf{v}\nabla\cdot \mathbf{n}\right) dx = \int_\Omega (\nabla H(\phi)) \cdot \left(\mathbf{v}\nabla\cdot \mathbf{n}\right) dx \\ &= \int_\Omega H'(\phi)\nabla\phi \cdot \left(\mathbf{v}\nabla\cdot \mathbf{n}\right) dx = \int_\Omega H'(\phi)\mathbf{n} \cdot \left(\mathbf{v}\nabla\cdot \mathbf{n}\right) dx \\ &= \int_\Omega \delta(\phi)\mathbf{n} \cdot \left(\mathbf{v}\nabla\cdot \mathbf{n}\right) dx = \int_\Omega \delta(\phi)(\mathbf{n}\cdot\mathbf{v})\nabla\cdot \mathbf{n}\, dx \end{aligned} \tag{14.88}$$

where of course $\delta = H'$ is a regularized Dirac delta function. Writing $\mathbf{f} = \delta(\phi)\mathbf{v}$ we have

$$\begin{aligned} \oint_\gamma \kappa \mathbf{v} \cdot \mathbf{n}\, ds &= \int_\Omega \mathbf{f} \cdot \mathbf{n}\nabla\cdot \mathbf{n}\, dx = -\int_\Omega \nabla(\mathbf{f}\cdot\mathbf{n}) \cdot \mathbf{n}\, dx \\ &= -\int_\Omega \left((\nabla\mathbf{f})\mathbf{n} + (\nabla\mathbf{n})\mathbf{f}\right) \cdot \mathbf{n}\, dx \end{aligned} \tag{14.89}$$

where $\nabla\mathbf{f}$ denotes the matrix $(\nabla\mathbf{f})_{ij} := f_{j,i}$.

Since we supposed that $\mathbf{n} = \nabla\phi$ we have $n_{i,j} = \phi_{,ij} = \phi_{,ji} = n_{j,i}$. Therefore the term

$$\begin{aligned} ((\nabla\mathbf{n})\mathbf{f}) \cdot \mathbf{n} &= \sum_{i,j} f_j n_{j,i} n_i = \sum_{i,j} f_j n_{i,j} n_i = \tfrac{1}{2} \sum_{i,j} f_j (n_i^2)_{,j} \\ &= \tfrac{1}{2} \sum_j f_j (|\mathbf{n}|^2)_{,j} = \tfrac{1}{2}\mathbf{f} \cdot \nabla|\mathbf{n}|^2 = 0, \end{aligned} \tag{14.90}$$

since $\mathbf{n}$ is a unit vector. As a result, we have

$$\oint_\gamma \kappa\mathbf{v}\cdot\mathbf{n}\,ds = -\int_\Omega ((\nabla\mathbf{f})\mathbf{n})\cdot\mathbf{n}\,dx. \tag{14.91}$$

Recall that $\mathbf{f}=\delta(\phi)\mathbf{v}$. Therefore

$$f_{i,j} = \delta'(\phi)\phi_{,j}v_i + \delta(\phi)v_{i,j} = \delta'(\phi)n_jv_i + \delta(\phi)v_{i,j} \tag{14.92}$$

and

$$\nabla\cdot\mathbf{f} = \delta'(\phi)\nabla\phi\cdot\mathbf{v} + \delta(\phi)\nabla\cdot\mathbf{v} = \delta'(\phi)\mathbf{n}\cdot\mathbf{v} + \delta(\phi)\nabla\cdot\mathbf{v}. \tag{14.93}$$

Thus

$$\begin{aligned}\mathbf{n}^t(\nabla\mathbf{f})\mathbf{n} &= \sum_{i,j} n_i f_{i,j} n_j = \sum_{i,j} n_i\left(\delta'(\phi)n_jv_i + \delta(\phi)v_{i,j}\right)n_j\\ &= \delta'(\phi)\left(\sum_{i,j} n_i n_j^2 v_i\right) + \delta(\phi)\left(\sum_{i,j} n_i v_{i,j} n_j\right)\\ &= \delta'(\phi)\left(\mathbf{n}\cdot\mathbf{v}\right) + \delta(\phi)\left(\sum_{i,j} n_i v_{i,j} n_j\right)\\ &= \nabla\cdot\mathbf{f} + \delta(\phi)\left(-\nabla\cdot\mathbf{v} + \sum_{i,j} n_i v_{i,j} n_j\right) = \nabla\cdot\mathbf{f} + \delta(\phi)\mathbf{n}^tD(\mathbf{v})\mathbf{n},\end{aligned} \tag{14.94}$$

where $D\mathbf{v} := \nabla\mathbf{v} - (\nabla\cdot\mathbf{v})I$ and I denotes the identity matrix. Thus we find

$$\oint_\gamma \kappa\mathbf{v}\cdot\mathbf{n}\,ds = \int_\Omega \delta(\phi)\mathbf{n}^tD(\mathbf{v})\mathbf{n}\,dx \tag{14.95}$$

since $\int_\Omega \nabla\cdot\mathbf{f}\,dx = 0$.

An alternative derivation is available in two dimensions. We may write $\kappa\mathbf{n} = \frac{\partial\mathbf{t}}{\partial s}$, where $\mathbf{t}$ denotes the tangent vector to γ, and we have

$$\begin{aligned}\oint_\gamma \kappa\mathbf{v}\cdot\mathbf{n}\,ds &= \oint_\gamma \frac{\partial\mathbf{t}}{\partial s}\cdot\mathbf{v}\,ds = -\oint_\gamma \mathbf{t}\cdot\frac{\partial\mathbf{v}}{\partial s}\,ds\\ &= -\oint_\gamma \mathbf{t}\cdot\nabla\mathbf{v}\cdot\mathbf{t}\,ds \approx -\int_\Omega \delta(\phi)\mathbf{t}\cdot\nabla\mathbf{v}\cdot\mathbf{t}\,dx.\end{aligned} \tag{14.96}$$

The relation between (14.95) and (14.96) is given by the fact that $\mathbf{n}=J\mathbf{t}$ where J denotes the rotation matrix for a ninety-degree angle, and the observation that

$$D(\mathbf{v}) = -J\nabla\mathbf{v}J \tag{14.97}$$

in two dimensions.

Thus the variational problem (14.86) becomes: Find $\mathbf{u}_0\in V$ and $p\in\Pi$ such that

$$\begin{aligned}a\left(\mathbf{u}_0,\mathbf{v}\right) + b\left(\mathbf{v},p\right) &= -a\left(\mathbf{u_g},\mathbf{v}\right) - R\,c\left(\mathbf{u}_0+\mathbf{u_g},\mathbf{u}_0+\mathbf{u_g},\mathbf{v}\right)\\ &\quad + \sigma\int_\Omega \delta(\phi)\mathbf{n}^tD(\mathbf{v})\mathbf{n}\,dx \quad \forall\mathbf{v}\in V\\ b\left(\mathbf{u}_0,q\right) &= -b\left(\mathbf{u_g},q\right) \quad \forall q\in\Pi.\end{aligned} \tag{14.98}$$

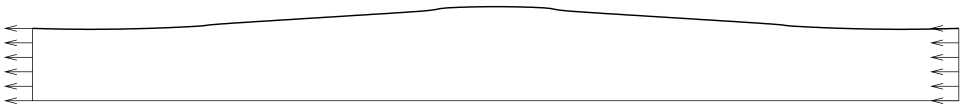

Figure 14.2: Tsunami in moving coordinates.

14.8 Simulating tsunamis

Tsunamis are long surface waves on an ocean surface that can cause remarkable devastation when they reach coastlines. In deep water, they can be imperceptable due to their relatively small amplitude. However, they travel extremely rapidly, and they can propagate with little change in shape for thousands of miles.

Scientific research on surface waves related to tsunamis began with observations by Scott-Russell in 1845. Mathmatical research on such waves can be traced to Boussinesq [35, p. 360], and it became popularized via the model of Korteweg-de Vries (KdV) for unidirectional waves [153]. These early models involved one space dimension, and the central unknown is the wave height. Recent Boussinesq models involve two space dimensions [120, 72] which allows modelling of more complex wave patterns.

The underlying fluid mechanical model is based on the Navier-Stokes equations with a free surface in d dimensions. When $d = 2$, we assume that the flow is constant in the third variable. The $d - 1$-dimensional models (KdV, Boussinesq) attempt to relate the height of the free surface to the internal flow by certain simplifications. A key assumption is that the internal flow is irrotational.

A more precise model is based on the Euler equations in the full d-dimensional spatial variables [61]. When the flow is irrotational, these equations can be simplified in a way that is very useful for computation [138, 137, 136].

Our objective here is to assess the need to make such approximations by considering direct numerical solution of the full Navier-Stokes equations in settings where one obtains the appropriate long-wave solutions. In some geometries, the Navier-Stokes flows will be turbulent, and this requires a different strategy to obtain reasonable results [41].

There have been extensive measurements of tsunamis in nature [86]. Similarly, there have been extensive comparisons between laboratory experiments and mathematical models [92, 31].

There are theoretical results showing a close relationship between the reduced-dimensional models and the Euler system [32, 99, 153]. Since the Euler equations are the limit of the Navier-Stokes system as the viscosity tends to zero, we expect that there is a solitary wave solution to the Navier-Stokes equations, at least for small enough amplitude.

14.8.1 Navier-Stokes in moving coordinates

$$\begin{aligned} \hat{\mathbf{u}}_t - \nu\Delta\hat{\mathbf{u}} + \hat{\mathbf{u}}\cdot\nabla\hat{\mathbf{u}} + \nabla p &= -G\mathbf{k} \text{ in } \Omega \\ \nabla\cdot\hat{\mathbf{u}} &= 0 \text{ in } \Omega, \end{aligned} \tag{14.99}$$

where $\mathbf{k}$ is the unit vector in the vertical direction and G is a constant related to gravity.

Define $\mathbf{v}$ by the ansazt $\hat{\mathbf{u}}(\mathbf{x},t)=\mathbf{v}(\mathbf{x}-ct\mathbf{i})$, where $\mathbf{i}$ is the unit vector in the direction of the propagation of the tsunami. Then $\mathbf{u}_t(\mathbf{x},t)=-c\mathbf{i}\cdot\nabla\mathbf{v}(\mathbf{x}-ct\mathbf{i})$. Thus

$$\begin{aligned}-c\mathbf{i}\cdot\nabla\mathbf{v}-\nu\Delta\mathbf{v}+\mathbf{v}\cdot\nabla\mathbf{v}+\nabla p&=-G\mathbf{k}\text{ in }\Omega\\ \nabla\cdot\mathbf{v}&=0\text{ in }\Omega.\end{aligned}\tag{14.100}$$

Define $\mathbf{u}(\mathbf{x},t)=\mathbf{v}-c\mathbf{i}$. Then

$$\begin{aligned}-\nu\Delta\mathbf{u}+\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p&=-G\mathbf{k}\text{ in }\Omega\\ \nabla\cdot\mathbf{u}&=0\text{ in }\Omega.\end{aligned}\tag{14.101}$$

Thus the appropriate boundary conditions would be $\mathbf{u}=\mathbf{f}=-c\mathbf{i}$ at the ends of the domain and along the bottom, as shown in Figure 14.2. Writing $\tilde{\mathbf{u}}=\mathbf{u}-\mathbf{f}$, we find

$$\begin{aligned}-\nu\Delta\tilde{\mathbf{u}}+\tilde{\mathbf{u}}\cdot\nabla\tilde{\mathbf{u}}-\mathbf{f}\cdot\nabla\tilde{\mathbf{u}}+\nabla p&=-G\mathbf{k}\text{ in }\Omega\\ \nabla\cdot\tilde{\mathbf{u}}&=0\text{ in }\Omega,\end{aligned}\tag{14.102}$$

where $\tilde{\mathbf{u}}=0$ on the non-free part of the boundary.

The relevant boundary conditions on the free surface are $\mathbf{u}\cdot\mathbf{n}=0$ and

$$\mathbf{n}^t(\nabla\mathbf{u}+\nabla\mathbf{u}^t)\mathbf{t}_i=0,\ i=1,\dots,d-1,\tag{14.103}$$

where $\{\mathbf{t}_i\}$ is a basis for the tangent space of the free surface.

14.8.2 Numerical methods for Navier-Stokes

We will be interested in a geometry as shown in Figure 14.2.

The divergence constraint $\nabla\cdot\mathbf{u}=0$ becomes the dominant term in the equation. Thus a nature finite element approach to consider is one that satisfies the divergence constraint exactly [157, 158, 169, 45, 50, 115, 81, 177]. For details regarding solvers for these elements see [114].

14.9 Exercises

Exercise 14.1 *Prove that*

$$\Big|\int_\Omega(\mathbf{u}\cdot\nabla\mathbf{v})\cdot\mathbf{w}\,d\mathbf{x}\Big|\le\|\mathbf{u}\|_{L^\infty(\Omega)}\int_\Omega\|\nabla\mathbf{v}(x)\|_F|\mathbf{w}(x)|\,d\mathbf{x},$$

where $\|M\|_F:=\sqrt{M:M}$ denotes the Frobenius norm of any matrix M and $|\mathbf{w}(x)|$ denotes the Euclidean norm of $\mathbf{w}(x)$. (Hint: for any matrix M and vector V, write $\sum_i(MV)_i^2=\sum_i\big(\sum_j M_{ij}V_j\big)^2$ and apply the Cauchy-Schwarz inequality for vectors to show that $|MV|\le\|M\|_F|V|$.)

Exercise 14.2 *Prove (14.6). (Hint: just write*

$$\nabla\cdot(\mathbf{u}v)=\sum_i \frac{\partial(u_i v)}{\partial x_i}$$

and apply the product rule to each term, using the fact that $\mathbf{u}\cdot\nabla v=\sum_i u_i\frac{\partial v}{\partial x_i}$*.)*

Exercise 14.3 *Prove (14.8). (Hint: write out the dot products and apply (14.7).)*

Exercise 14.4 *Prove (14.18). (Hint: multiply both sides by* $1-\epsilon$*.)*

Exercise 14.5 *Prove that*

$$(1+ct/k)^k\le e^{ct}\quad\forall k\ge 1,$$

and show that in addition that

$$(1+ct/k)^k\to e^{ct}\ \ \text{as}\ \ k\to\infty.$$

(Hint: use the identity $x^k=e^{k\log x}$ *and prove the bound* $\log(1+y)\le y$ *for* $y>0$*.)*

Exercise 14.6 *A more stable time-stepping scheme is based on the variational equations*

$$\begin{aligned} a\left(\mathbf{u}^\ell,\mathbf{v}\right)+b\left(\mathbf{v},p^\ell\right)+R\,c\left(\mathbf{u}^\ell,\mathbf{u}^\ell,\mathbf{v}\right)+\frac{R}{\Delta t}\left(\mathbf{u}^\ell-\mathbf{u}^{\ell-1},\mathbf{v}\right)_{\underset{\sim}{L}^2}&=0,\\ b\left(\mathbf{u}^\ell,q\right)&=0,\end{aligned}\tag{14.104}$$

in which the nonlinear term has been evaluated at the current time step. This leaves a nonlinear problem to be solved. Formulate Newton's method for solving the nonlinear problem (14.104) at each time step. How does the resulting linear algebraic problem change at each time step? How does it compare with (14.58)?

Exercise 14.7 *Consider the time-stepping scheme (14.104) in which the nonlinear term has been evaluated at the current time step. Formulate a fixed point iteration for solving the nonlinear problem (14.104) at each time step such that the resulting linear algebraic problem does not change at each time step.*

Exercise 14.8 *Identify the differential equations corresponging to the following variant of (14.2): Find* $\mathbf{u}$ *such that* $\mathbf{u}-\boldsymbol{\gamma}\in V$ *and* $p\in\Pi$ *such that*

$$\begin{aligned} a(\mathbf{u},\mathbf{v})+b(\mathbf{v},p)+R\big(c(\mathbf{v},\mathbf{u},\mathbf{u})+(\mathbf{u}_t,\mathbf{v})_{\underset{\sim}{L}^2}\big)&=0\quad\forall\mathbf{v}\in V\,,\\ b(\mathbf{u},q)&=0\quad\forall q\in\Pi\,,\end{aligned}\tag{14.105}$$

where we have switched the order of $\mathbf{u}$ *and* $\mathbf{v}$ *in the "c" form. How does it compare with (14.1)?*

Exercise 14.9 *Identify the differential equations corresponging to the following variant of (14.2): Find* $\mathbf{u}$ *such that* $\mathbf{u}-\boldsymbol{\gamma}\in V$ *and* $p\in\Pi$ *such that*

$$\begin{aligned} a\left(\mathbf{u},\mathbf{v}\right)+b\left(\mathbf{v},p\right)+&R\Big(c\left(\mathbf{u},\mathbf{v},\mathbf{u}\right)+\left(\mathbf{u}_t,\mathbf{v}\right)_{\underset{\sim}{L}^2}\Big)=0\quad\forall\mathbf{v}\in V\,,\\ b(\mathbf{u},q)&=0\quad\forall q\in\Pi\,,\end{aligned}\tag{14.106}$$

where we have switched the order of $\mathbf{u}$ *and* $\mathbf{v}$ *in the "c" form. How does it compare with (14.1)?*

Exercise 14.10 *Consider the variant of (14.2) in which we switch the order of* $\mathbf{u}$ *and* $\mathbf{v}$ *in the "a" form. Does it change the equations from what appears in (14.1)? If so, to what? If not, why not?*

Exercise 14.11 *Replace* $\cos\theta$ *by* $\cos^2\theta$ *in (14.83) and show that it has the same desired properties. Use this to derive another function* $\mathbf{v}:=(0,v)$ *where*

$$\begin{aligned} v(x,y):=\frac{y+d}{1+d}&\left(1+\left(\frac{1-y}{x+w}\right)^2\right)^{-1}\left(1+\left(\frac{1-y}{x-w}\right)^2\right)^{-1}\\ +\frac{1-y}{1+d}&\left(1+\left(\frac{y+d}{x+w}\right)^2\right)^{-1}\left(1+\left(\frac{y+d}{x-w}\right)^2\right)^{-1}\end{aligned}$$

for $|x|<w$ *and* $-d<y<1$*, and zero elsewhere, that satisfies (14.85).*

Exercise 14.12 *Prove that Sobolev's imbedding implies (in three dimensions)*

$$|\mathbf{u}|_{W^1_{2p}(\Omega)}\le C\|\mathbf{u}\|^{\epsilon/2}_{L^r(\Omega)}\,|\mathbf{u}|^{1-\epsilon/2}_{H^2(\Omega)}\,,\tag{14.107}$$

where $\epsilon=2\frac{r-3}{r+3}$.

Exercise 14.13 *Prove that (14.31) can be improved in the three-dimensional case to*

$$CR^2\int_\Omega|\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x}\le CR^{-2+4/\epsilon_r}\|\mathbf{u}\|^2_{L^2(\Omega)}\|\mathbf{u}\|^{4/\epsilon_r}_{L^r(\Omega)}+\tfrac{1}{2}|\mathbf{u}|^2_{H^2(\Omega)}\tag{14.108}$$

Exercise 14.14 *Show, in the three-dimensional case, that if*

$$\int_\Omega|\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x}\le C\|\mathbf{u}\|^{2+\epsilon}_{L^2(\Omega)}|\mathbf{u}|^{2-\epsilon}_{H^2(\Omega)}\tag{14.109}$$

for all $\mathbf{u}\in H^2(\Omega)^3$*, then* ϵ *must satisfy (by scaling) the relation*

$$1=\frac{3}{2}\left(2+\epsilon\right)-\frac{1}{2}\left(2-\epsilon\right)=2+2\epsilon\tag{14.110}$$

which implies $\epsilon=-1/2$.

Exercise 14.15 *Show that in the three-dimensional case,*

$$\int_{\Omega} |\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x} \le C\|\mathbf{u}\|_{L^4(\Omega)}^{2+\epsilon}|\mathbf{u}|_{H^2(\Omega)}^{2-\epsilon} \tag{14.111}$$

for all $\mathbf{u}\in H^2(\Omega)^3$*, where* ϵ *satisfies* $1=\frac{3}{4}(2+\epsilon)-\frac{1}{2}(2-\epsilon)=\frac{1}{2}+\frac{5}{4}\epsilon$ *which implies* $\epsilon=2/5$*. Compare this with (14.31).*

Exercise 14.16 *Show that the scaling arguments applied to the proposed inequality (in 3-D)*

$$\int_{\Omega} |\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x} \le C\|\mathbf{u}\|_{L^3(\Omega)}^{2+\epsilon}|\mathbf{u}|_{H^2(\Omega)}^{2-\epsilon} \tag{14.112}$$

would imply that ϵ *satisfies* $1=(2+\epsilon)-\frac{1}{2}(2-\epsilon)=1+\frac{3}{2}\epsilon$ *or that* $\epsilon=0$*. Is the estimate (14.112) true with* $\epsilon=0$*? What would this mean in terms of the arguments of Section 14.2?*

Exercise 14.17 *Show, in the two-dimensional case, that if* $r>2$ *then*

$$\int_{\Omega} |\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x} \le C\|\mathbf{u}\|_{L^r(\Omega)}^{2+\epsilon}|\mathbf{u}|_{H^2(\Omega)}^{2-\epsilon} \tag{14.113}$$

for all $\mathbf{u}\in H^2(\Omega)^2$*, where* $\epsilon>0$*. (Hint:* ϵ *satisfies the relation*

$$0=\frac{2}{r}(2+\epsilon)-(2-\epsilon)=2\left(\frac{2}{r}-1\right)+\left(\frac{2}{r}+1\right)\epsilon \tag{14.114}$$

and $2/r<1$*.)*

Exercise 14.18 *Show, in the three-dimensional case, that*

$$\int_{\Omega} |\mathbf{u}(s)\cdot\nabla\mathbf{u}(s)|^2\,d\mathbf{x} \le C\|\mathbf{u}\|_{L^2(\Omega)}^{2+\epsilon}|\mathbf{u}|_{H^3(\Omega)}^{2-\epsilon} \tag{14.115}$$

for all $\mathbf{u}\in H^2(\Omega)^3$*, where* ϵ *satisfies*

$$1=\frac{3}{2}(2+\epsilon)-\frac{3}{2}(2-\epsilon)=3\epsilon \tag{14.116}$$

which implies $\epsilon=1/3$*.*

Chapter 15

Pitfalls in Modeling using PDEs

Systems of partial differential equations provide the basis for some of the most powerful models of physical and social phenomena. The formalism of such models allows a remarkable level of automation of the process of simulating complex systems. Leveraging this potential for automation has been developed to the greatest extent by the FEniCS Project. However, using partial differential equation models involves numerous *pitfalls*. We highlight many of these pitfalls and discuss ways to circumvent them.

It is simply not possible to provide a solution to all systems of differential equations. Any given differential equation may be *ill-posed*, meaning that it does not make sense to talk about the solution for one reason or another. At the moment, there is no simple criterion to determine if a system of differential equations is well-posed. As a result, it is not possible to provide software that solves all systems of differential equations automagically. The first step, then, is to determine if a system being studied is *well-posed*.

There are several ways in which a differential equation can fail to be well-posed. The most serious potential pitfall for a differential equation is the lack of a solution regardless of any boundary conditions or initial conditions. In Section 15.4, we give an example of such an equation with this extreme behavior. Although it may be quite rare, such a pitfall does exist if one tries to solve arbitrary systems of partial differential equations.

If a physical system is supposed to have a unique state given suitable determining conditions, then a mathematical model having multiple solutions is seriously flawed. The typical cause of a system of partial differential equations to have too many solutions is a lack of boundary conditions. It is not at all trivial to determine what the right number of boundary conditions might be for an arbitrary system of partial differential equations, and getting it wrong could lead to either too many solutions or too few! In section Section 15.1 we present a case where both of these can be seen.

Equally damaging, but often more subtle to detect, is the lack of continuous dependence of the solution on the data of a mathematical model, at least when the physical problem should have this property. Continuous dependence of the solution on the data will be verified for many systems of partial differential equations using various techniques subsequently. However, it is not always required that a physical problem have this property. One such "ill-posed" problem is considered in Section 7.6.

All of these shortcomings of models can be summarized as a lack of *well-posedness.* We will discuss some techniques to determine if a particular differential equation is well-posed, but this is generally beyond the scope of the book. Rather, it is assumed that the reader is trying to solved a well-posed problem.

Equally difficult to insure, even for differential equations that are well-posed, is the stability and consistency (and equivalently, convergence) of the numerical approximations. Even for well-posed systems of differential equations, there is no automatic way to define a discrete approximation scheme that will always converge to the solution of the differential equation as the approximation is refined. We discuss various pitfalls and examine in depth many of the most critical. However, we make no attempt to be exhaustive. We are mainly concerned with the implementation of convergent algorithms using structured programming techniques. It is assumed that the reader is trying to solved a well-posed problem with a stable and consistent numerical approximation.

A basic language we frequently employ is that of the variational formulation of differential equations. This is a powerful formulation that allows a simple proof of well-posedness in many cases. Moreover, it leads to stable and consistent numerical schemes through the use of appropriate approximation spaces of functions. In this way, finite element methods, spectral methods, spectral element methods, boundary element methods, collocation methods, variational difference methods and other discretization methods can be derived and analyzed with respect to stability and consistency.

15.1 Right-sizing BCs

Differential equations typically have too many solutions of the equations themselves to specify a solution in any reasonable sense. A unique solution, required by most physical models, is typically determined by boundary conditions and, for time-dependent problems, initial conditions. We will use the following notation of partial differential equations for the "Laplacian" operator

$$\Delta := \sum_{i=1}^{d} \frac{\partial^2}{\partial {x_i}^2}. \tag{15.1}$$

Consider the Laplace equation

$$-\Delta u = f \tag{15.2}$$

Then for $f \equiv 0$ the solutions are *harmonic* functions, and the real part of any complex analytic function in the plane (in two space dimensions) is harmonic. For any solution to (15.2), we can get another by adding any harmonic function. Thus there are way too many solutions to (15.2) without any further restrictions.

We will see in Chapter 2 that specifying the value of u on the boundary of some open set Ω makes the solution of (15.2) unique in Ω. That is, the system of equations

$$\begin{aligned} -\Delta u =& f \text{ in } \Omega \\ u =& g \text{ on } \partial\Omega \end{aligned} \tag{15.3}$$

has a unique solution, under suitable smoothness conditions on f, g and the boundary $\partial\Omega$.

There is no unique type of boundary condition that is appropriate for a given system of differential equations. For example, the system

$$\begin{aligned} -\Delta u =& f \text{ in } \Omega \\ \frac{\partial u}{\partial n} =& g \text{ on } \partial\Omega \end{aligned} \tag{15.4}$$

also has a solution, under suitable smoothness conditions on f, g and the boundary $\partial\Omega$, provided in addition that a simple compatibility condition exists between f and g, namely,

$$\int_\Omega f(\mathbf{x})\, d\mathbf{x} + \oint_{\partial\Omega} g(s)\, ds = 0. \tag{15.5}$$

This compatibility condition is a consequence of the divergence theorem (2.7) together with the observations that $\frac{\partial u}{\partial n} = (\nabla u)\cdot\mathbf{n}$ and $\Delta u = \nabla\cdot(\nabla u)$. Here, $\mathbf{n}$ is the outward-directed normal to $\partial\Omega$.

For any solution u to (15.4), $u+c$ is also a solution for any constant c. Thus there is a certain degree of non-uniqueness here, but it can be seen (Section 2.2) that this is all there is. That is, solutions to (15.4) exist and are unique up to an additive constant, provided the single compatibility contion (15.5) holds.

If some is good, then one might think more is better. However, it is easy to see that the system of equations

$$\begin{aligned} -\Delta u =& f \text{ in } \Omega \\ u =& g_0 \text{ on } \partial\Omega \\ \frac{\partial u}{\partial n} =& g_1 \text{ on } \partial\Omega \end{aligned} \tag{15.6}$$

has too many boundary conditions. Since the condition $u = g_0$ on $\partial\Omega$ already uniquely determines the solution u, it will only be a miracle that $\frac{\partial u}{\partial n} = g_1$ also holds on $\partial\Omega$. More precisely, there is a linear mapping A defined on functions on $\partial\Omega$ such that (15.6) has a solution if and only if $g_1 = Ag_0$ (see Exercise 15.4). Similarly, the system

$$\begin{aligned} -\Delta u =& f \text{ in } \Omega \\ \nabla u =& \mathbf{g} \text{ on } \partial\Omega \end{aligned} \tag{15.7}$$

is over-specified. It is closely related to (15.6) if we observe that the second equation says that the tangential derivative of u is equal to that of g_0. The over-determined boundary value problem (15.7) appears in a non-local compatibility condition for the Navier-Stokes equations (Section 14.3.2).

15.2 Numerical Stability

The simplest differential equation to solve is an ordinary differential equation

$$\frac{du}{dt} = f(u,t) \tag{15.8}$$

with initial value

$$u(0) = u_0 \tag{15.9}$$

where we are interested in solving on some interval $[0, T]$.

The definition of the derivative as a limit of difference quotients suggests a method of discretization:

$$\frac{du}{dt}(t) \approx \frac{u(t + \Delta t) - u(t)}{\Delta t} \tag{15.10}$$

where Δt is a small positive parameter. This suggests an algorithm for generating a sequence of values $u_n \approx u(n\Delta t)$ given by (for example)

$$u_n = u_{n-1} + \Delta t f(u_n, t_n) \tag{15.11}$$

where $t_n = n\Delta t$.

The algorithm (15.11) is called the **implicit Euler** method, and it can be shown that it generates a sequence with the property that

$$|u(t_n) - u_n| \leq C_{f,T}\Delta t \quad \forall t_n \leq T \tag{15.12}$$

provided that we solve the implicit equation (15.11) for u_n exactly and we compute with exact arithmetic. The issue of solving the nonlinear equation at each step is important but not a show-stopper. However, the requirement of using finite-precision arithmetic means that the best error behavior we could expect is

$$|u(t_n) - u_n| \leq C_{f,T}\Delta t + n\epsilon \quad \forall t_n \leq T \tag{15.13}$$

where ϵ measures the precision error that occurs at each step in (15.11). It is useful to re-write (15.13) using the fact that $n = t_n/\Delta t$ as

$$|u(t_n) - u_n| \leq C_{f,T}\Delta t + \frac{t_n\epsilon}{\Delta t} \tag{15.14}$$

which shows that the error reaches a minimum and cannot be reduced by reducing Δt.

One way to increase the accuracy in (15.13) is to use a more accurate approximation of the derivative than (15.10), such as given by the backwards differentiaion formulæ (BDF) defined in Section 7.5

$$\frac{du}{dt}(t) \approx \frac{1}{\Delta t}\sum_{i=0}^{k} a_i u_{n-i} \tag{15.15}$$

where the coefficients $\{a_i \,:\, i = 0, \dots k\}$ are chosen so that (15.15) is exact for polynomials of degree k. The BDF for $k = 1$ is the same as implicit Euler. Using the approximation (15.15), we get an algorithm of the form

$$\sum_{i=0}^{k} a_i u_{n-i} = \Delta t f(u_n, t_n) \tag{15.16}$$

which can be solved for u_n provided $a_0 \neq 0$. In this case, the final error estimate would be

$$|u(t_n) - u_n| \leq C_{f,T,k}\Delta t^k + \frac{t_n\epsilon}{\Delta t}. \tag{15.17}$$

Ultimate accuracy is still limited, but smaller absolute errors (with larger Δt) can be achieved with higher values of k. For example, suppose that

- $\epsilon = 10^{-6}$ (which corresponds to single precision on a 32-bit machine)
- $T = 1$ and
- (for the sake of argument) $C_{f,T,k} = 1$.

Then with implicit Euler ($k = 1$) the smallest error we can get is 10^{-3} with $\Delta t = 10^{-3}$. On the other hand, with $k = 2$ we get an error of size 10^{-4} with $\Delta t = 10^{-2}$. Not only is this a smaller error but less work needs to be done to achieve it. In practice, the constant $C_{f,T,k}$ would depend on k and the exact error behavior would likely be different in detail, but the general conclusion that a higher-order scheme may be better still holds. The BDF methods for $k = 2$ and 3 are extremely popular schemes.

We see that higher-order schemes can lead to more managible errors and potentially less work for the same level of accuracy. Thus it seems natural to ask whether there are limits to choosing the order to be arbitrarily high. Unfortunately, not all of the BDF schemes are viable. Beyond degree six, they become **unconditionally unstable**. Let us examine the question of stability via a simple experiment. Suppose that, after some time T_0, it happens that $f(u,t) = 0$ for $t \geq T_0$. Then the solution u remains constant after T_0, since $\frac{du}{dt} \equiv 0$. What happens in the algorithm (15.16) is that we have

$$\sum_{i=0}^{k} a_i u_{n-i} = 0 \tag{15.18}$$

for $n \geq T_0/\Delta t$. However, this does not necessarily imply that u_n would tend to a constant. Let us examine what the solutions of (15.18) could look like.

Consider the sequence $u_n := \xi^{-n}$ for some number ξ. Plugging into (15.18) we find

$$0 = \sum_{i=0}^{k} a_i \xi^{-n+i} = \xi^{-n} \sum_{i=0}^{k} a_i \xi^i \tag{15.19}$$

If we define the polynomial p_k by

$$p_k(\xi) = \sum_{i=0}^{k} a_i \xi^i \tag{15.20}$$

we see that we have a null solution to (15.18) if and only if ξ is a root of p_k. If there is a root ξ of p_k where $|\xi| < 1$ then we get solutions to (15.18) which grow like

$$u_n = \xi^{-n} = \left(\frac{1}{\xi}\right)^{t_n/\Delta t}. \tag{15.21}$$

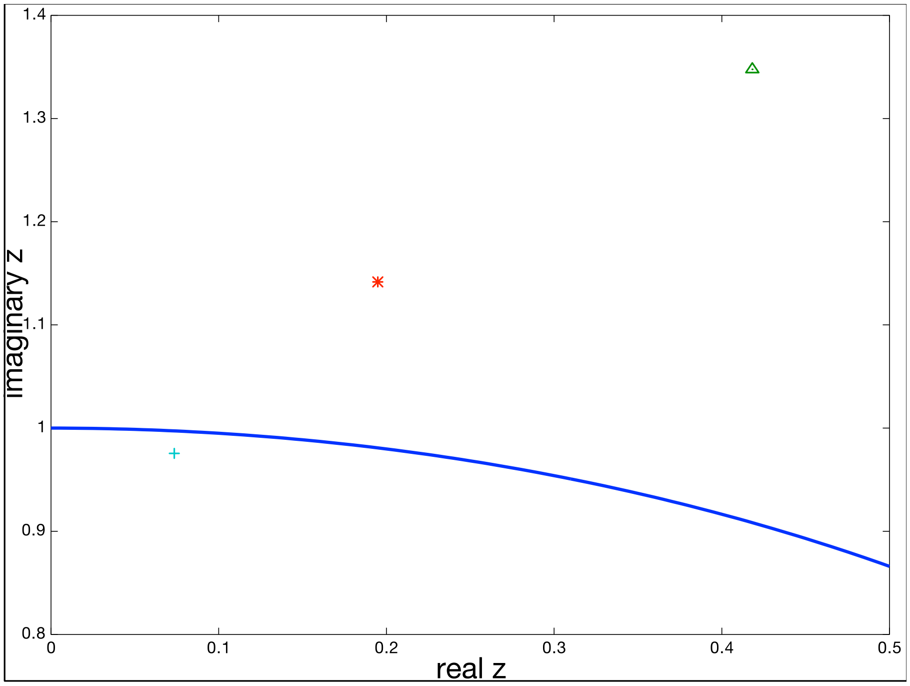

Figure 15.1: Roots of polynomials (15.20) with the smallest modulus are plotted for degrees $k = 5$ (triangle), $k = 6$ (asterisk), and $k = 7$ (plus). The solid line indicates the unit circle in the complex plane.

Not only does this blow up exponentially, the exponential rate goes to infinity as $\Delta t \to 0$. This clearly spells disaster. On the other hand, if $|\xi| > 1$, then the solution (15.21) goes rapidly to zero, and more rapidly as $\Delta t \to 0$. For roots ξ with $|\xi| = 1$ the situation is more complicated, and $\xi = 1$ is always a root because the sum of the coefficients a_i is always zero. Instability occurs if there is a multiple root on the unit circle $|\xi| = 1$. In general, one must consider all complex (as well as real) roots ξ.

Given this simple definition of the general case of BDF, it is hard to imagine what could go wrong regarding stability. Unfortunately, the condition that $|\xi| \geq 1$ for roots of $p_k(\xi) = 0$ restricts k to be six or less for the BDF formulæ. In Figure 15.1, complex roots with the smallest modulus are plotted for degrees $k = 5$ (triangle), $k = 6$ (asterisk), and $k = 7$ (plus). The solid line indicates the unit circle in the complex plane. Unfortunately, for $k = 7$ there is a pair of complex conjugate roots $z_{\pm} = 0.0735 \pm 0.9755\iota$ with (the same) complex modulus less than 1: $|z_{\pm}| \approx 0.97827$.

In Chapter 12, the inf-sup condtion for mixed methods encapsulates another type of numerical stability that effects the choice of finite element spaces suitable for fluid flow problems.

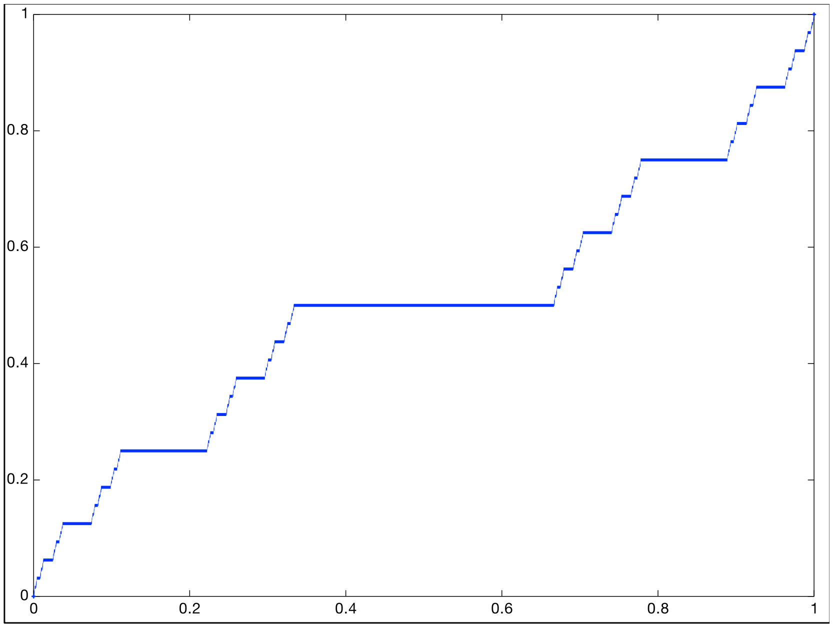

Figure 15.2: Cantor function (15.22).

15.3 The right box

The variational formulation of PDEs is not only a convenient way to describe a model, it also provides a way to ensure that a model is well-posed. One of the critical ingredients in the variational formulation is the space of functions in which one seeks the solution. We can see that this is just the right size to fit the needs of most modeling problems.

We can think of the space of functions in the variational formulation of PDEs as a box in which we look for a solution. If the box were too big, we might get spurious solutions. If it is too small, then we may get no solutions.

A box too small is easy to describe: it would be natural to think we could just work with the space C^m of functions whose derivatives up through order m are continuous. If the number of derivatives in the PDE is less than or equal to m, then all derivatives are classically defined, and the equation itself makes sense as a equation among numbers at all points in the model domain. However, many problems naturally arise that do not fit into this box. For example, when the geometry of the boundary of a domain in two-dimensions has an interior angle that is greater than π (and hence the domain is not convex), a singularity arises even for the Laplace equation (Chapter 2).

A box too big can be described already in one-dimension. The famous Cantor "middle-thirds" function is defined as follows on the unit interval $[0,1]$:

$$C(x) := \begin{cases} \frac{1}{2} & \frac{1}{3} \leq x < \frac{2}{3} \\ \frac{1}{2}C(3x) & 0 \leq x < \frac{1}{3} \\ \frac{1}{2}(1 + C(3x-2)) & \frac{2}{3} \leq x < 1 \end{cases} \tag{15.22}$$

The recursive nature of this definition means that it can easily be computed. Any programming language allowing recursion is suitable for implementation. By definition, the derivative of C is zero at almost every point (a notion made precise by Lebesgue measure theory [38].)

This is problematic, since we expect solutions of linear PDEs to be unique in most cases. But the simplest equation $u' = f$ would have $u + C$ as a solution for any given solution u. However, the derivative of C does not qualify as a Lebesgue integrable function.

Thus the choice of spaces V of functions whose derivatives are Lebesgue integrable functions provides just the right size box in the variational formulation. These are called Sobolev spaces, and they are reviewed in Chapter 19.

15.4 Local Solvability

When we state an equation such as $\Delta u = f$, we presume that it is possible to specify u, at least locally, by giving a combination of its derivatives $(u_{,11} + u_{,22} + \cdots)$. What is it that makes this possible? That is, should we always assume that an arbitrary combination of derivatives can be specified without any internal consistency required?

It is easy to see one kind of partial differential equation that would make little sense:

$$\frac{\partial^2 u}{\partial x\,\partial y} = -\frac{\partial^2 u}{\partial y\,\partial x}, \tag{15.23}$$

since we know that for smooth functions, the order of cross derivatives does not matter. Thus (15.23) corresponds to an equation of the form $t = -t$ and has only the zero solution.

There are some differential equations that simply have no solution even locally, independent of any boundary conditions. A famous example is due to Hans Lewy. Consider the equation

$$\frac{\partial u}{\partial x_1} - \iota\frac{\partial u}{\partial x_2} + 2(\iota x_1 - x_2)\frac{\partial u}{\partial x_3} = f, \tag{15.24}$$

where ι is the imaginary unit, $\iota = \sqrt{-1}$. Then for most infinitely differentiable functions f there is no solution of this equation in *any* open set in three-space. Note that this has nothing to do with boundary conditions, just with satisfying the differential equation. This equation is a complex equation ($\iota = \sqrt{-1}$) but it can easily be written as a system of two real equations for the real and imaginary parts of u respectively (see Exercise 15.3).

There is a general condition that must be satisfied in order that linear partial differential equations have a local solution. This condition is known as the **local solvability condition**. To explain the condition, we need to introduce some notation. Let $D = -\iota\left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_j}, \ldots, \frac{\partial}{\partial x_d}\right)$ stand for the vector of complex partial differentials, and let $\alpha = (\alpha_1, \ldots, \alpha_j, \ldots, \alpha_d)$ be a **multi-index** (i.e., a vector of non-negative integers), so that

$$D^\alpha u := (-\iota)^{|\alpha|}\frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_j}}{\partial x_j^{\alpha_j}} \cdots \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}} u\,, \tag{15.25}$$

where $|\alpha| := \alpha_1 + \cdots \alpha_j + \cdots \alpha_d$. For any d-dimensional variable ξ, we can form the monomial

$$\xi^\alpha := {\xi_1}^{\alpha_1} \cdots {\xi_j}^{\alpha_j} \cdots {\xi_d}^{\alpha_d} \tag{15.26}$$

so that D^α is the same as ξ^α with the substitution $\xi_j = -i\partial/\partial x_j$. In this notation, the Lewy equation (15.24) becomes

$$-\iota D_1 u + D_2 u - 2(x_1 + \iota x_2)D_3 u = f \tag{15.27}$$

The reason for the factor $-\iota$ in the definition of D is so that the Fourier transform of D works out nicely; if $\hat{u}$ denotes the Fourier transform of u, then $\widehat{D^\alpha u}(\xi) = \xi^\alpha \hat{u}(\xi)$.

Suppose that the differential operator in question takes the form

$$P(\mathbf{x}, D) = \sum_{|\alpha|\le m} a_\alpha(\mathbf{x}) D^\alpha\,, \tag{15.28}$$

that is, suppose that we want to consider linear partial differential equations of the form

$$P(\mathbf{x}, D)u = \sum_{|\alpha|\le m} a_\alpha(\mathbf{x}) D^\alpha u = f \tag{15.29}$$

for some f. We can form the corresponding **symbol** $P(\mathbf{x}, \xi)$ of the linear partial differential operator

$$P(\mathbf{x}, \xi) = \sum_{|\alpha|\le m} a_\alpha(\mathbf{x}) \xi^\alpha\,, \tag{15.30}$$

Define the **principal part** of the symbol, P_m, by

$$P_m(\mathbf{x}, \xi) = \sum_{|\alpha|= m} a_\alpha(\mathbf{x}) \xi^\alpha\,, \tag{15.31}$$

and correspondingly the complex conjugate of the principal part of the symbol, $\overline{P}_m$, by

$$\overline{P}_m(\mathbf{x}, \xi) = \sum_{|\alpha|= m} \overline{a_\alpha(\mathbf{x})} \xi^\alpha\,. \tag{15.32}$$

Also define the following partial derivatives of the principal part of the symbol:

$$P_m^{(j)}(\mathbf{x}, \xi) := \frac{\partial P_m}{\partial \xi_j}(\mathbf{x}, \xi)\,, \quad P_{m,j}(\mathbf{x}, \xi) := \frac{\partial P_m}{\partial x_j}(\mathbf{x}, \xi) \tag{15.33}$$

and define their complex conjugates analogously. Finally, define the **commutator** $C_{2m-1}(\mathbf{x}, \xi)$ of the principal part of the symbol via

$$C_{2m-1}(\mathbf{x}, \xi) := \iota \sum_{j=1}^{d} \left(P_m^{(j)}(\mathbf{x}, \xi)\overline{P}_{m,j}(\mathbf{x}, \xi) - \overline{P}_m^{(j)}(\mathbf{x}, \xi) P_{m,j}(\mathbf{x}, \xi)\right) \tag{15.34}$$

which is a polynomial of degree $2m-1$ in ξ with real coefficients.

Theorem 15.1 *If the differential equation (15.29) has a solution in a set Ω for every smooth f that vanishes near the boundary of Ω, then*

$$C_{2m-1}(\mathbf{x}, \xi) = 0 \tag{15.35}$$

for all ξ and all $x \in \Omega$ such that $P_m(\mathbf{x}, \xi) = 0$.

Here, the notion of "solution" is very weak; it need not be a smooth solution. Thus the result provides a very stringent condition on the symbol in order to expect any sort of solution at all. For a complete description of this and other examples see [96]; see [36] for more recent results and references. The most striking feature of the local solvability condition (15.35) is that it is a "closed" condition. Otherwise said, "non-solvability" is an open condition: if $C_{2m-1}(\mathbf{x},\xi) \neq 0$ then small perturbations would not be expected to make it vanish. Moreover, even if (15.35) holds for one set of coefficients a_α, it may fail to hold for a small perturbation. Finally, we will be interested in nonlinear partial differential equations; if these have a solution, then the solution can be viewed as solutions to linear partial differential equations with appropriate coefficients (which depend on the particular solution).

Despite the pessimism implied by the local solvability condition (15.35), we will see that there are indeed broad classes of nonlinear partial differential equations which can be proved to have solutions. But this should not be taken for granted in general. In proposing a new model for a new phenomenon, the first question to ask is whether it makes sense at this most fundamental level.

15.5 PDE's in classical spaces

We have emphasized working with PDEs in Sobolev spaces but without indicating why this is necessary. It turns out that the usual C^k spaces of functions with k-th order continuous derivatives are not appropriate for characterizing PDEs. We will just give an extended example to explain this and leave it to references [75, 82] for more details. Consider the Laplace equation

$$-\Delta u = f, \tag{15.36}$$

for the moment in all of $\mathbb{R}^d$ without regard for boundary conditions. Then a solution can be generated by convolution with the fundamental solution G:

$$u = G * f, \tag{15.37}$$

where the convolution operator is defined by $v * w(x) = \int_{\mathbb{R}^d} v(x-y)w(y)\,dy$ and

$$G(x) = \begin{cases} c_2 \log|x| & d=2 \\ c_d|x|^{2-d} & d \geq 3. \end{cases} \tag{15.38}$$

To have a classical solution to (15.36), we would demand that $u \in C^2$, and we might expect this for $f \in C^0$. We will see that this is not the case.

We can differentiate the expression (15.38) to represent derivatives of u in terms of f via

$$D^\alpha u = (D^\alpha G) * f, \tag{15.39}$$

using a well-known property of convolution, where D^α is defined in (15.25). It is not hard to show that

$$D^\alpha G(x) = A_\alpha(x)|x|^{-d} \text{ for } |\alpha| = 2, \tag{15.40}$$

where A_α is homogeneous of degree zero. Thus $D^\alpha G$ is what is known as a Calderón-Zygmund singular kernel, and the mapping $f \to (D^\alpha G) * f$ is a singular integral operator. Such operators are known [82, Section 9.4] to be bounded on L^p for $1 < p < \infty$, but not for the extreme values of p, as we will see shortly. The key property of singular kernels that supports this is that A_α has mean zero. This cancellation property allows the singular integral to be controlled in a natural way, but only in a mean sense. If we take

$$f(x) = \frac{A_\alpha(x)}{|\log|x||}\chi(|x|), \quad \chi(r) = \begin{cases} 1 & r \le \frac{1}{4} \\ 2 - 4r & \frac{1}{4} \le r \le \frac{1}{2} \\ 0 & r \ge \frac{1}{2} \end{cases}, \tag{15.41}$$

we get a divergent integral for $D^\alpha u = (D^\alpha G) * f$,

$$D^\alpha u(0) = \int_{|x| \le \frac{1}{2}} \frac{A_\alpha(x)^2 \chi(|x|)}{|x|^d \,|\log|x||} dx = C_{\alpha,d} \int_0^{\frac{1}{2}} \frac{\chi(r)\, dr}{r|\log r|} = \infty, \tag{15.42}$$

indicating that $D^\alpha u(x) \to \infty$ as $x \to 0$. Thus we see that bounded f can lead to solutions u of (15.36) whose second derivatives are not bounded. We leave as Exercise 15.6 to explore this.

Our example shows why the Calderón-Zygmund theorem fails for $p = \infty$, since f is bounded and the second derivatives of u are not bounded. But it also does more, as the function f is both bounded and continuous, since $f(x) \to 0$ as $x \to 0$. Our example shows that, not only is $u \notin C^2$, its second derivatives are not even bounded. In [82, exercise 4.9(a)], a different example is given in which f is continuous but $u \notin C^2$. The root cause of this behavior is the fact that the solution operator for the Laplace equation involves an aggregation of values. Thus if f has bad behavior over an extended region, this can add up to yield unexpected behavior of the solution.

15.6 Exercises

Exercise 15.1 *Consider the example following (15.17). How small can the error in (15.17) be made using a 5-th order ($k = 5$) BDF formula? What value of Δt yields the smallest error?*

Exercise 15.2 *Compute the solution of an ordinary differential equation using the 7-th order ($k = 7$) BDF formula. What happens after a few times steps?*

Exercise 15.3 *Write (15.24) as a system of two real equations for the real and imaginary parts of u respectively.*

Exercise 15.4 *Give a precise definition of the "Dirichlet to Neumann" map A such that (15.6) has a solution if and only if $g_1 = Ag_0$.*

Exercise 15.5 *Prove that the local solvability condition (15.35) does not hold for the equation (15.24).*

Exercise 15.6 *Compute $A_{(2,0)}$ in (15.40) for $d = 2$, that is,*

$$A_{(2,0)}(x,y) = \frac{\partial^2}{\partial x^2} \log|(x,y)|.$$

Take f as defined in (15.41) and solve $-\Delta u = f$ in a domain containing the origin, with your choice of boundary conditions. Compute the second derivatives of u near the origin. Do they blow up as the mesh is made finer?

Chapter 16

Solvers

In large computations, the rate limiting step is often the solution of a linear system. In some cases, it is sufficient to use Gaussian elimination or its variants (such as Cholesky factorization). Such algorithms are called **direct methods**. However, for large, three-dimensional simulations, **iterative methods** are frequently more efficient.

16.1 Stationary iterative methods

There are three important classes of iterative methods. The first of these are known equivalently as **stationary iterative methods** and **relaxation methods** [87, 98]. Examples include Jacobi, Gauss-Seidel, SOR, SSOR, etc.; these basic techniques are still used in certain contexts, and many of the concepts behind them are frequently used in more complicated solvers. In particular, relaxation methods are frequently used as **smoothers** for multi-grid methods. Typically, the simpler the iterative method, the easier it is to implement a parallel version.

Suppose that we are solving a linear system $AX = F$. The general form of a stationary iterative scheme is

$$Nx^{n+1} = PX^n + F,$$

where $A = N - P$ and N is chosen to be an easily invertible matrix, e.g., diagonal (Jacobi) or triangular (Gauss-Seidel, SOR). The error $E^n = X - X^n$ satisfies

$$x^{n+1} = MX^n,$$

algorithm	sufficient conditions on A for convergence
Jacobi	generalized diagonally dominant
Gauss-Seidel	symmetric, positive definite
SOR	symmetric, positive definite

Table 16.1: Stationary iterative methods for solving $AX = F$ and conditions on A that guarantee convergence.

algorithm	matrices for which the method applies
CG	symmetric, positive definite
MINRES	symmetric
GMRES	invertible

Table 16.2: Krylov subspace based methods for solving $AX = F$ and conditions on A that guarantee convergence.

where $M = N^{-1}P$. Thus convergence is equivalent to $\rho(M) < 1$ where ρ is the spectral radius. It is known [159] that Jacobi is convergent for generalized diagonally dominant matrices, and Gauss-Seidel and SOR are convergent for symmetric, positive definite matrices.

16.2 Krylov methods

Krylov[1] methods are a class of techniques based on projecting the solution onto an increasing subspace of vectors that is efficient to create.

Suppose that we are solving a linear system $AX = F$. Then the Krylov subspace of order k is the linear space spanned by

$$F, AF, \ldots, A^k F.$$

Such vectors are easily created iteratively via $A^i F = A(A^{i-1}F)$, where $A^0 F = F$.

The first of the Krylov methods is called **conjugate gradients** (a.k.a. **CG**) and was developed by Hestenes[2] and Stiefel[3]. Conjugate gradients converges for symmetric positive definite matrices, and it has an optimality property [159] that makes it extremely attractive.

The algorithm **MINRES** is applicable to symmetric but indefinite matrices [139]. Although CG and MINRES utilize the same Krylov space of vectors, they minimize different quantities. CG minimizes $\|X - X^k\|_A$, where $\|y\|_A = \sqrt{y^t A y}$, whereas MINRES minimizes $\|F - AX^k\|_{\ell^2}$, where $\|y\|_{\ell^2} = \sqrt{y^t y}$. It is easy to see that CG requires A to be positive definite, since $\|\cdot\|_A$ is not a norm otherwise. For symmetric, positive definite matrices, MINRES can outperform CG in some cases [78].

The algorithm **GMRES** [151, 88, 47] can be used for general matrices. The **Arnoldi** algorithm [42, 89] is closely related to GMRES.

[1] Alexei Nikolaevich Krylov (1863–1945) was very active in the theory and practice of shipbuilding and is commemorated by the Krylov Shipbuilding Research Institute.

[2] Magnus Rudolph Hestenes (1906–1991) obtained a Ph.D. at the University of Chicago with Gilbert Bliss in 1932.

[3] Eduard L. Stiefel (1909–1978) is known both as a pure mathematician (for his work on the Stiefel-Whitney characteristic classes) and as a computational mathematician (he was also an early user and developer of computers [167]). Stiefel was the advisor of Peter Henrici as well as 63 other students over a period of 37 years. Henrici was the advisor of Gilbert Strang, one of the early pioneers of the mathematical theory of the finite element method.

16.3 Multi-grid

Multi-grid methods apply to problems posed on grids that are sufficiently structured to talk about coarse grids and fine grids. In a variational setting, this occurs if we have a sequence of subspaces $V^i \subset V$ with the property that $V^i \subset V^{i+1}$. The solutions to the variational problems

$$\text{Find } u^i \in V^i \text{ such that } a(u^i, v) = F(v) \quad \forall v \in V^i \tag{16.1}$$

are then increasingly accurate approximations to the solution u of the variational problem

$$\text{Find } u \in V \text{ such that } a(u, v) = F(v) \quad \forall v \in V. \tag{16.2}$$

But more importantly, u^i can be used as an initial guess for an iterative scheme for solving for u^{i+1}. However, the real power of multi-grid is deeper than this. Suppose that we are trying to solve for $u^i \in V^i$ satisfying (16.1) and we have an approximate solution $w^i \in V^i$. Then the residual error $r^i = u^i - w^i$ satisfies

$$a(r^i, v) = a(u^i, v) - a(w^i, v) = F(v) - a(w^i, v) \quad \forall v \in V, \tag{16.3}$$

which does not involve knowing u^i. The magic of multi-grid is to approximate (16.3) on a coarser space (V^{i-1}). Of course, we need to know that w^i is smooth enough that this is an effective strategy, but this can be achieved by using a variety of iterative methods, such as the stationary iterative methods, to remove high frequencies. For details, see [40].

It is not strictly required to have $V^i \subset V^{i+1}$. There are two ways in which $V^i \not\subset V^{i+1}$ occurs. One is when discontinuous finite element spaces are used [39]. Another is when the underlying grids are not nested [114, 178, 43, 85].

Multi-grid methods were initiated by Bakhvalov[4] [17] in 1966. Achi Brandt [37] began popularizing and developing the method in 1977. Bank and Dupont [20, 21] gave one of the first proofs of convergence of the method, in research that initiated at the University of Chicago.

16.4 Preconditioners

The convergence rate of many iterative methods depends on the condition number of the linear system. For a symmetric, positive definite linear system, we can take the condition number to be defined as the ratio of the largest eigenvalue divided by the smallest eigenvalue. Linear systems associated with partial differential operators often have a condition numbers that grow inversely with the mesh resolution. This is because the PDE often has eigenvalues of unbounded size, and the finer the mesh the larger the eigenvalues that can be resolved. In particular, eigenfunctions often oscillate with a frequency roughly proportional to the eigenvalue. Thus the finer meshes resolve higher frequencies. Therefore iterative methods introduce a limit on our ability to resolve solutions based on mesh refinement.

[4] Nikolai Sergeevich Bakhvalov (1934—2005) studied with both Sobolev and Kolmogorov.

We have seen that round-off is strongly affected by the condition number of a linear system. One approach to this dilemma is to use higher-order approximations, but this is limited by the fact that the higher-order approximation resolves higher frequency eigenfunctions, so the condition number is not reduced. A better approach is to scale the linear system appropriately to control the size of the eigenvalues.

The simplest scaling is diagonal preconditioning. For a given matrix A, define $\operatorname{diag}(A)$ to be the diagonal matrix with the same diagonal entries as A. Then the diagonal preconditioning of A is the matrix $\operatorname{diag}(A)^{-1}A$. Fortunately, it is simple to compute the inverse of a diagonal matrix $P = \operatorname{diag}(A)^{-1}$. A more complex version of this idea is to produce a preconditioner by using an incomplete factorization of the system [91, 26] in which a sparsity condition is enforced autocratically. That is, Gaussian elimination (or other direct method) is performed in a way that only creates certain nonzero elements in the resulting factors. If for example we restrict the factors to be only diagonal, then the algorithm is equivalent to diagonal preconditioning. On the other hand, we make no restriction on sparsity, and allow arbitrary fill-in, then the algorithm produces the exact inverse (via forward and backward solution algorithms). The general case falls in between.

One common choice is to limit the sparsity of the factors to be the same as the sparsity pattern of the original matrix. Such a factorization is obtained by following an elimination algorithms (e.g., Gaussian elimination, Cholesky factorization, etc.), but when the algorithm calls for fill-in to occur, these additions to the original sparse structure of A are ignored. This yields a matrix P with the same sparsity pattern as A and yet P is in some sense an approximation to A^{-1}.

The benefit of preconditioning is that the iterative method performs as if the matrix condition number is that of the preconditioned system PA [73]. Thus significant benefit can occur. The main objective is to choose P to be as close to A^{-1} as possible.

The discrete Green's function provides a way to solve a system [155], and this can be used to create an efficient, parallel algorithm to implement a preconditioner [155].

A general understanding of preconditioners for linear systems arising in solving PDEs is given in [124]. They describe how to extend the concept of preconditioner to the PDE itself, and this in turn, when discretized, provides an effective preconditioner that can be robust with respect to mesh size and parameters in the PDE. In (16.4), we see two ways to construct preconditioners, and [124] advocates the bottom-left of the diagram.

$$\begin{tikzcd}
\text{PDE operator } A \arrow[r, "\text{discretization}"] \arrow[d, "\substack{\text{approx.}\\ \text{inverse}}"] & A_h \arrow[d, "\substack{\text{approx.}\\ \text{inverse}}"] \\
\text{preconditioner } P \text{ for PDE} \arrow[r, "\text{discretization}"] & P_h\,.
\end{tikzcd} \tag{16.4}$$

Chapter 17

Rheological flows

Rheology is a general term referring to flow of any matter. Here we will restrict this to mean non-Newtonian fluids. Newtonian fluids satisfy the Navier-Stokes equations, and we will assume the reader is familiar with the material in Chapter 12 and Chapter 14 as a basis for the current chapter.

There are many models of non-Newtonian fluid flow. We cover just a few of the major ones.

Cystic fibrosis is a genetic disease that is caused in part by a change in rheology of the mucus in the lungs [100, 8, 111]. According to an industry website, “there are two layers of intrabronchial mucus: a low viscosity, high elasticity periciliary liquid (PCL), and a higher, more viscous airway surface fluid (ASF). The elasticity of mucus appears to change with the application of stress, and may be important to the rate of beating of bronchial epithelial cilia. Furthermore, in CF, the PCL liquid layer is thinner [165], and the thicker ASF tends to impede the movement of the cilia. In healthy humans, the concentration of mucins in the mucus is about 1In CF patients, the mucin content increases to as much as 3-4This results is increased viscosity, reduced elasticity and increased adhesiveness, which hinders movement of the mucus(3). The absolute viscosity of normal mucus is generally 1 kg/m-sec, but can be as much as 500 kg/m-sec in CF patients(4).”

The cause of the change in rheology at the molecular level has been studied [168].

17.1 Grade-two fluid model

17.2 The convected Maxwell models

Chapter 18

Poisson-Boltzman equation

Continuum models for the electrostatics of solvents around proteins are often used in drug design []. Recently, non-local models of the dielectric have been developed that are of the same order of computational cost as the local model [66, 67].

18.1 The Poisson–Boltzmann equation

18.2 The nonlocal Poisson–Boltzmann model

18.3 Exercises

Chapter 19

Tutorial on Sobolev spaces

The coercivity result (2.20) is the key to well-posedness of boundary value problems for elliptic problems. It implicitly encodes both a statement of stability and a description of suitable boundary conditions. Here, we offer a tutorial on why this is true, as well as some other results in approximation theory that are closely connected. It is intriguing that the concepts of approximability and coercivity can be so closely linked. For simplicity, let us assume that Ω is convex, so that for any two points in Ω, all points on the line joining them is also in Ω. Most of the results derived in this simple setting can be proved for more general domains [40].

19.1 An integral representation

We begin with a very simple representation using integral calculus. Suppose that x and y are two points in some domain Ω in d-dimensional space, and observe that we can write

$$u(\mathbf{y}) - u(\mathbf{x}) = \int_0^1 (\mathbf{y}-\mathbf{x})\cdot\nabla u(\mathbf{x}+s(\mathbf{y}-\mathbf{x}))\,ds\,. \tag{19.1}$$

This is just the multi-dimensional version of the calculus theorem

$$f(1) - f(0) = \int_0^1 f'(s)\,ds\,. \tag{19.2}$$

We can obtain (19.1) from (19.2) by defining $f(s) := u(\mathbf{x}+s(\mathbf{y}-\mathbf{x}))$.

Let us now integrate (19.1) with respect to y over Ω, to get

$$|\Omega|(\overline{u} - u(\mathbf{x})) = \int_\Omega\int_0^1 (\mathbf{y}-\mathbf{x})\cdot\nabla u(\mathbf{x}+s(\mathbf{y}-\mathbf{x}))\,ds\,d\mathbf{y}\,, \tag{19.3}$$

where $|\Omega|$ is the measure of Ω and $\overline{u}$ denotes the mean of u:

$$\overline{u} := \frac{1}{|\Omega|}\int_\Omega u(\mathbf{y})\,d\mathbf{y}\,. \tag{19.4}$$

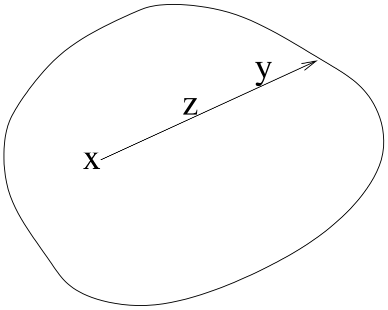

Figure 19.1: Notation for integral represenation.

Make the change of variables $\mathbf{z} = \mathbf{x} + s(\mathbf{y} - \mathbf{x})$, so that $\mathbf{y} = \mathbf{x} + s^{-1}(\mathbf{z} - \mathbf{x})$ and $d\mathbf{y} = s^{-d} d\mathbf{y}$; see Figure 19.1. With this representation, the variable $\mathbf{z}$ also ranges over all of Ω, but not independently of s. The range of values of s is restricted by the fact that $\mathbf{y} = \mathbf{x} + s^{-1}(\mathbf{z} - \mathbf{x})$ must remain in Ω; and for each $\mathbf{z} \in \Omega$, there is a $\sigma(\mathbf{z})$ such that $x + \sigma(\mathbf{z})^{-1}(\mathbf{z} - \mathbf{x}) \in \partial\Omega$. In fact, $\sigma(\mathbf{z})$ is just $|\mathbf{z} - \mathbf{x}|$ divided by the distance from $\mathbf{x}$ to $\partial\Omega$ along a line passing through $\mathbf{z}$. Since $\mathbf{z} \in \Omega$, this description of $\sigma(\mathbf{z})$ shows that it is always less than one. More importantly, $\sigma(\mathbf{z})^{-1} \leq \operatorname{diam}(\Omega)|\mathbf{z} - \mathbf{x}|^{-1}$, where $\operatorname{diam}(\Omega)$ denotes the diameter of Ω (the largest distance between any two points in Ω).

Plugging and chugging, we get via Fubini's Theorem that

$$\begin{aligned} u(\mathbf{x}) =& \overline{u} - \frac{1}{|\Omega|} \int_\Omega \int_{\sigma(\mathbf{z})}^1 (\mathbf{z} - \mathbf{x}) \cdot \nabla u(\mathbf{z}) s^{-1-d} \, ds \, d\mathbf{z} \,, \\ =& \overline{u} - \frac{1}{|\Omega|} \int_\Omega \frac{1}{d} \left(\sigma(\mathbf{z})^{-d} - 1 \right) (\mathbf{z} - \mathbf{x}) \cdot \nabla u(\mathbf{z}) \, d\mathbf{z} \,, \\ =& \overline{u} + \int_\Omega k(\mathbf{x}, \mathbf{z}) \cdot \nabla u(\mathbf{z}) \, d\mathbf{z} \,, \end{aligned} \tag{19.5}$$

where $k(\mathbf{x}, \mathbf{z}) := \frac{1}{|\Omega| d} \left(\sigma(\mathbf{z})^{-d} - 1 \right) (\mathbf{x} - \mathbf{z})$. Note that

$$|k(\mathbf{x}, \mathbf{z})| \leq \frac{\operatorname{diam}(\Omega)^d}{|\Omega| d} |\mathbf{x} - \mathbf{z}|^{1-d} \tag{19.6}$$

is integrable in $\mathbf{z}$ for $\mathbf{x}$ fixed.

19.2 Sobolev inclusions and Riesz potentials

The kernel in (19.6) is bounded by what are called Riesz potentials which have simple properties. In this section, we derive various bounds for such potentials. Our main results will include an inclusion relation of the form

$$W_p^m(\Omega) \subset W_q^k(\Omega) \tag{19.7}$$

for suitable indices m, k, p, q. We can anticipate (and remember) the appropriate relationship among these indices by considering units of the semi-norms

$$|\cdot|_{W_p^m(\Omega)} \quad \text{and} \quad |\cdot|_{W_q^k(\Omega)}. \tag{19.8}$$

These units can be determined either by a dimensional analysis, or by scaling the spatial variable by a dilation. In either case, it is easy to see that the dimensions are

$$L^{-m+d/p} \quad \text{and} \quad L^{-k+d/q} \tag{19.9}$$

respectively, since each differentiation contributes a factor of L^{-1}, and the Lebesgue measure contributes a factor of $L^{n/p}$ (due to the fact that the $1/p$-th root is taken: note that the derivatives are raised to the p-th power first). Comparing these exponents suggests that we should have

$$m - \frac{d}{p} \geq k - \frac{d}{q} \tag{19.10}$$

in order for (19.7) to hold. That is, the way to count Sobolev derivatives is to take the number of L^p derivatives and subtract d/p from that. It is clear that if we can prove (19.7) for $m = 1$ and $k = 0$ then the general result obtains by iterating on the indices, as follows. It immediately follows for $k = m - 1$ by applying the first case to derivatives of order $m - 1$. Applying this result together with the corresponding result for $m \leftarrow m - 1$ and $k \leftarrow m - 2$ yields the desired result when $m - k = 2$, with appropriate "p" indices. The general result is obtained by continuing in this way.

Now let us show how (19.7) can be verified, and explain the fine print on exactly when it applies.

Lemma 19.1 *If $f \in L^p(\Omega)$ for $1 < p < \infty$ and $m > d/p$, then*

$$g(\mathbf{x}) := \int_\Omega |\mathbf{x} - \mathbf{z}|^{-n+m} |f(\mathbf{z})| \, d\mathbf{z} \leq C \|f\|_{L^p(\Omega)} \quad \forall \, \mathbf{x} \in \Omega. \tag{19.11}$$

This inequality also holds for $p = 1$ if $m \geq d$.

Proof. First assume that $1 < p < \infty$ and $m > d/p$. Let $1/p + 1/q = 1$. By Hölder's inequality, we have

$$\begin{aligned} \int_\Omega |\mathbf{x} - \mathbf{z}|^{-n+m} |f(\mathbf{z})| \, d\mathbf{z} &\leq \left(\int_\Omega |\mathbf{x} - \mathbf{z}|^{(-n+m)q} \, d\mathbf{z} \right)^{1/q} \|f\|_{L^p(\Omega)} \\ &\leq C \left(\int_0^{\text{diam}(\Omega)} r^{(-n+m)q+d-1} \, dr \right)^{1/q} \|f\|_{L^p(\Omega)} \\ &= C \, \|f\|_{L^p(\Omega)} \, . \end{aligned} \tag{19.12}$$

If $m \geq d$, then $|\mathbf{x} - \mathbf{z}|^{-n+m}$ is bounded, and

$$\int_\Omega |\mathbf{x} - \mathbf{z}|^{-n+m} |f(\mathbf{z})| \, d\mathbf{z} \leq C \|f\|_{L^1(\Omega)}. \tag{19.13}$$

Corollary 19.1 *For $u \in W_p^1(\Omega)$,*

$$\|u - \overline{u}\|_{L^\infty(\Omega)} \le C|u|_{W_p^1(\Omega)},$$

provided that $d < p \le \infty$ or $d = 1$ and $p \ge 1$.

Lemma 19.2 (Sobolev's Inequality) *Suppose Ω is bounded and convex. If u is in $W_p^m(\Omega)$ where either (i) $1 < p < \infty$ and $m > d/p$ or (ii) $p = 1$ and $d = 1$, then u is continuous on Ω and*

$$\|u\|_{L^\infty(\Omega)} \le C\|u\|_{W_p^m(\Omega)}.$$

Proof. The inequality holds because

$$\begin{aligned}\|u\|_{L^\infty(\Omega)} &\le \|u - \overline{u}\|_{L^\infty(\Omega)} + \|\overline{u}\|_{L^\infty(\Omega)}\\ &\le |u|_{W_p^1(\Omega)} + \|u\|_{L^1(\Omega)}\\ &\le C\|u\|_{W_p^1(\Omega)}.\end{aligned} \tag{19.14}$$

The proof that u is continuous on Ω follows by a density argument.

We can think of the Riesz potentials as arising just by convolution. That is, we can define g in (19.11) as $g = K * |f|$ where we extend f by zero outside Ω and

$$K(\mathbf{x}) = |\mathbf{x}|^{m-n} \tag{19.15}$$

on a ball of radius diam(Ω) and zero outside. Strictly speaking, the definitions (19.11) and (19.15) of g agree only on Ω. Thus by Young's inequality we have

$$\begin{aligned}\|g\|_{L^r(\Omega)} &\le \|K\|_{L^q(\Omega)}\|f\|_{L^p(\Omega)}\\ &\le C\|f\|_{L^p(\Omega)}\end{aligned} \tag{19.16}$$

(provided that $K \in L^q(\Omega)$) where

$$\frac{1}{r} = \frac{1}{p} + \frac{1}{q} - 1 = \frac{1}{p} - \frac{1}{q'} \tag{19.17}$$

and $1/q + 1/q' = 1$. Then it is easy to see that $K \in L^q(\Omega)$ provided $q(m-n) + n - 1 > -1$. This is equivalent to $m > n/q'$, or $-1/q' > -m/n$. Thus we have $g \in L^r(\Omega)$ provided

$$\frac{1}{r} = \frac{1}{p} - \frac{1}{q'} > \frac{1}{p} - \frac{m}{n} \tag{19.18}$$

By more precise arguments, it is in fact possible to prove [125] that the Riesz potential (19.15) maps $L^p(\Omega)$ to $L^r(\Omega)$ precisely for $\frac{1}{r} = \frac{1}{p} - \frac{m}{n}$. Thus the following holds.

Theorem 19.1 *For $u \in W_p^1(\Omega)$,*

$$\|u - \overline{u}\|_{L^r(\Omega)} \le C|u|_{W_p^1(\Omega)}, \tag{19.19}$$

provided that

$$\frac{1}{r} \ge \frac{1}{p} - \frac{1}{n} \tag{19.20}$$

19.3 Compactness in Sobolev spaces

In Section 19.2 we saw that Sobolev spaces with one set of indices are naturally included in another, eg. (19.7). We now want to show that this inclusion is a compact mapping, that is, a bounded set in the stronger norm is compact in the weaker norm.

Our main ingredient is Theorem 19.1. We apply this on a triangulation $\mathcal{T}_h$ of Ω of size h to get an approximation result for piecewise constant approximation.

Lemma 19.3 *For $u \in W^1_p(\Omega)$, let u_h denote the piecewise constant approximation of u on a triangulation $\mathcal{T}_h$ of Ω of size h. Then*

$$\|u - u_h\|_{L^r(\Omega)} \leq C h^{1-n/p+n/r} |u|_{W^1_p(\Omega)}, \tag{19.21}$$

provided that

$$\frac{1}{r} \geq \frac{1}{p} - \frac{1}{n} \tag{19.22}$$

Let K be a bounded subset of $W^1_p(\Omega)$; that is, assume that $|u|_{W^1_p(\Omega)} \leq \gamma$ for all $u \in K$. Let $\epsilon > 0$. For any $h > 0$, the image of K via the projection $u \to u_h$ is a bounded set in a finite dimensional space. In fact, we can cover the set $K_h := \{u_h \ : \ u \in K\}$ by a finite union of balls in $L^r(\Omega)$ of radius $\epsilon/2$ centered at a simple lattice centered on piecewise constant functions v^j_h. Then for any $u \in K$ we have u_h in such a ball around some particular v^j_h for some j. Therefore

$$\begin{aligned} \|u - v^j_h\|_{L^r(\Omega)} &\leq \|u - u_h\|_{L^r(\Omega)} + \|u_h - v^j_h\|_{L^r(\Omega)} \\ &\leq C h^{1-n/p+n/r} \gamma + \epsilon/2 \\ &\leq \epsilon \end{aligned} \tag{19.23}$$

provided that we choose h small enough, if the exponent $1 - n/p + n/r$ is positive. Thus we have shown that, for any $\epsilon > 0$, we can cover K by a finite number of balls of radius ϵ in $L^r(\Omega)$. Thus K is compact in $L^r(\Omega)$ [150]. Thus we have proved the following result for $m = 1$ and $k = 0$. The general result follows by iterating on the indices.

Theorem 19.2 *Let K be a bounded subset of $W^m_p(\Omega)$. Then K is a compact subset of $W^k_r(\Omega)$ if*

$$k - \frac{n}{r} < m - \frac{n}{p}\,. \tag{19.24}$$

19.4 Polynomial approximation in Sobolev spaces

We can think of Theorem 19.1 as an approximation result, which applied to a set of elements of a subdivision of size h yields a result like Lemma 19.3 for piecewise constant approximation. It is useful to ask about higher-order approximation, and we can use Theorem 19.1 to generate such a result.

Suppose we apply Theorem 19.1 to a derivative $u^{(\alpha)}$. Then we have

$$\|u^{(\alpha)} - \overline{u^{(\alpha)}}\|_{L^r(\Omega)} \le C|u|_{W_p^{|\alpha|+1}(\Omega)}, \tag{19.25}$$

provided that

$$\frac{1}{r} \ge \frac{1}{p} - \frac{1}{n} \tag{19.26}$$

as we now assume. Let $c_\alpha = \overline{u^{(\alpha)}}/\alpha!$. Note that

$$\partial^\alpha \mathbf{x}^\beta = \delta_{\alpha,\beta}\alpha!\,, \tag{19.27}$$

where $\delta_{\alpha,\beta}$ is the Kronecker δ: equal to one if $\alpha = \beta$ and zero otherwise. Thus $\partial^\alpha c_\alpha \mathbf{x}^\alpha = \overline{u^{(\alpha)}}$. Therefore

$$\|\partial^\alpha(u - c_\alpha \mathbf{x}^\alpha)\|_{L^r(\Omega)} \le C|u|_{W_p^{|\alpha|+1}(\Omega)}\,. \tag{19.28}$$

Now apply this for all α such that $m = |\alpha|$ and set

$$q_m(\mathbf{x}) = \sum_{|\alpha|=m} c_\alpha \mathbf{x}^\alpha \tag{19.29}$$

Observe that (19.27) implies that

$$|u - q_m|_{W_r^m(\Omega)} \le C|u|_{W_p^{m+1}(\Omega)}\,. \tag{19.30}$$

Write $q_m = Q^m u$ where Q^m is the (bounded linear) operator which takes u to q_m. Then we have

$$|u - Q^m u|_{W_r^m(\Omega)} \le C|u|_{W_p^{m+1}(\Omega)}\,. \tag{19.31}$$

Iterating (19.31), we find

$$\begin{aligned} |(u - Q^m u) - Q^{m-1}(u - Q^m u)|_{W_{r'}^{m-1}(\Omega)} &\le C|u - Q^m u|_{W_r^m(\Omega)} \\ &\le C|u|_{W_p^{m+1}(\Omega)} \end{aligned} \tag{19.32}$$

Define $Q_1^m u = Q^m u + Q^{m-1}(u - Q^m u)$ and $Q_0^m = Q^m$ for completeness. Note that

$$\begin{aligned} |u - Q_1^m u|_{W_{r'}^m(\Omega)} &= |(u - Q^m u) - Q^{m-1}(u - Q^m u)|_{W_r^m(\Omega)} \\ &= |u - Q^m u|_{W_{r'}^m(\Omega)} \\ &\le C|u|_{W_p^{m+1}(\Omega)} \end{aligned} \tag{19.33}$$

since the derivatives of order m vanish on Q^{m-1}. Thus we have

$$|u - Q_1^m u|_{W_r^m(\Omega)} + |u - Q_1^m u|_{W_r^{m-1}(\Omega)} \le C|u|_{W_p^{m+1}(\Omega)}\,. \tag{19.34}$$

Iterating, we can show that there exists a mapping Q_m^m onto polynomials of degree m such that

$$\|u - Q_m^m u\|_{W_r^m(\Omega)} \le C|u|_{W_p^{m+1}(\Omega)}\,. \tag{19.35}$$

19.5 Exercises

Bibliography

[1] *The Space H(div;Ω)*, pages 99–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] Cédric Adam, T. J. R. Hughes, Salim Bouabdallah, Malek Zarroug, and Habibou Maitournam. Selective and reduced numerical integrations for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:732–761, 2015.

[3] Shmuel Agmon. *Lectures on Elliptic Boundary Value Problems. Prepared for Publication by B. Frank Jones, with the Assistance of George W. Batten.* 1965.

[4] Shmuel Agmon, Avron Douglis, and Louis Nirenberg. Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions II. *Communications on pure and applied mathematics*, 17(1):35–92, 1964.

[5] Mark Ainsworth and J. Tinsley Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1):1–88, 1997.

[6] Alderson Alderson and KL Alderson. Auxetic materials. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 221(4):565–575, 2007.

[7] Thomas Apel, Sergei Grosman, Peter K. Jimack, and Arnd Meyer. A new methodology for anisotropic mesh refinement based upon error gradients. *Applied Numerical Mathematics*, 50(3):329–341, 2004.

[8] Ernst M. App, Rita Kieselmann, Dietrich Reinhardt, Hermann Lindemann, Bonnie Dasgupta, Malcolm King, and Peter Brand. Sputum rheology changes in cystic fibrosis lung disease following two different types of physiotherapy: flutter vs. autogenic drainage. *CHEST Journal*, 114(1):171–177, 1998.

[9] D. N. Arnold, M. Vogelius, and L. R. Scott. Regular inversion of the divergence operator with Dirichlet boundary conditions on a polygon. *Ann. Scuola Norm. Sup. Pisa Cl. Sci.–Serie IV*, XV:169–192, 1988.

[10] Douglas Arnold and Anders Logg. Periodic table of the finite elements. *SIAM News, November*, 2014.

[11] Douglas N. Arnold. Mixed finite element methods for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 82(1-3):281–300, 1990.

[12] Ivo Babuška and Juhani Pitkäranta. The plate paradox for hard and soft simple support. *SIAM Journal on Mathematical Analysis*, 21(3):551–576, 1990.

[13] Ivo Babuška and Werner C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978.

[14] B. Bagheri, A. Ilin, and L. R. Scott. Parallel 3-D MOSFET simulation. In *Proceedings of the* 27th *Annual Hawaii International Conference on System Sciences*, volume 1, pages 46–54, 1994.

[15] Babak Bagheri and L. R. Scott. About Analysa. Research Report UC/CS TR-2004-09, Dept. Comp. Sci., Univ. Chicago, 2004.

[16] Nathan Baker, Michael Holst, and Feng Wang. Adaptive multilevel finite element solution of the Poisson–Boltzmann equation II. Refinement at solvent-accessible surfaces in biomolecular systems. *Journal of computational chemistry*, 21(15):1343–1352, 2000.

[17] Nikolai Sergeevich Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Computational Mathematics and Mathematical Physics*, 6(5):101–135, 1966.

[18] Wolfgang Bangerth and Rolf Rannacher. *Adaptive finite element methods for differential equations*. Birkhäuser, 2013.

[19] R. E. Bank. *Computational Aspects of VLSI Design with an Emphasis on Semiconductor Device Simulation*. Amer. Math. Soc., 1990.

[20] R. E. Bank and T. Dupont. Analysis of a two-level scheme for solving finite element equations. Research Report CNA-159, Center for Numerical Analysis, Univ. Texas–Austin, 1980.

[21] Randolph E. Bank and Todd Dupont. An optimal order process for solving finite element equations. *Mathematics of Computation*, 36(153):35–51, 1981.

[22] J. R. Barber. Linear elasto-statics. In Jose Merodio and Giuseppe Saccomandi, editors, *Continuum Mechanics-Volume I*, page 344. Encyclopedia of Life Support Systems (EOLSS), Oxford, UK, 2008.

[23] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica 2001*, 10:1–102, 2001.

[24] J.-M. Bernard. Steady transport equation in the case where the normal component of the velocity does not vanish on the boundary. *SIAM Journal on Mathematical Analysis*, 44(2):993–1018, 2012.

[25] J.-M. Bernard. Solutions in H^1 of the steady transport equation in a bounded polygon with a full non-homogeneous velocity. *JMPA, submitted*, 2016.

[26] H. Berryman, J. Saltz, W. Gropp, and R. Mirchandaney. Krylov methods preconditioned with incompletely factored matrices on the CM-2. *Journal of Parallel and Distributed Computing*, 8:186–190, 1990.

[27] H. Blum and R. Rannacher. On the boundary value problem of the biharmonic operator on domains with angular corners. *Math. Meth. Appl. Sci.*, 2:556–581, 1980.

[28] D. Boffi. Three-dimensional finite element methods for the Stokes problem. *SIAM J. Num. Anal.*, 34:664 – 670, 1997.

[29] Richard Bois, Michel Fortin, and André Fortin. A fully optimal anisotropic mesh adaptation method based on a hierarchical error estimator. *Computer Methods in Applied Mechanics and Engineering*, 209:12–27, 2012.

[30] J. L. Bona, W. G. Pritchard, and L. R. Scott. A posteriori error estimates for exact and approximate solutions of time–dependent problems. In *Seminar on Numerical Analysis and Its Applications to Continuum Physics, Colecao ATAS 12, Sociedade Brasileira de Matematica*, pages 102–111, 1980.

[31] J. L. Bona, W. G. Pritchard, and L. R. Scott. An evaluation of a model equation for water waves. *Philos. Trans. Roy. Soc. London Ser. A 302*, pages 457–510, 1981.

[32] Jerry L. Bona, Thierry Colin, and David Lannes. Long wave approximations for water waves. *Archive for Rational Mechanics and Analysis*, 178(3):373–410, 2005.

[33] Y. Bourgault, M. Picasso, F. Alauzet, and A. Loseille. On the use of anisotropic a posteriori error estimators for the adaptative solution of 3D inviscid compressible flows. *International journal for numerical methods in fluids*, 59(1):47–74, 2009.

[34] Yves Bourgault and Marco Picasso. Anisotropic error estimates and space adaptivity for a semidiscrete finite element approximation of the transient transport equation. *SIAM Journal on Scientific Computing*, 35(2):A1192–A1211, 2013.

[35] J. Boussinesq. *Essai sur la théorie des eaux courantes.* Imprimerie nationale, 1877.

[36] Antonio Bove and Tatsuo Nishitani. Necessary conditions for local solvability for a class of differential systems. *Communications in Partial Differential Equations*, 27:1301 – 1336, 2002.

[37] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.

[38] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, third edition, 2008.

[39] Susanne C. Brenner. A nonconforming multigrid method for the stationary stokes equations. *Mathematics of Computation*, pages 411–437, 1990.

[40] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, third edition, 2008.

[41] M. Breuer, N. Peller, Ch. Rapp, and M. Manhart. Flow over periodic hills–numerical and experimental study in a wide range of Reynolds numbers. *Computers & Fluids*, 38(2):433–457, 2009.

[42] Peter N. Brown. A theoretical comparison of the Arnoldi and GMRES algorithms. *SIAM Journal on Scientific and Statistical Computing*, 12(1):58–78, 1991.

[43] Peter R. Brune, Matthew G. Knepley, and L. Ridgway Scott. Unstructured geometric multigrid in two and three dimensions on complex and graded meshes. *SIAM J. Sci. Computing*, 35(1):A173–A191, 2013.

[44] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Image enhancement by non-local reverse heat equation. *Preprint CMLA*, 22:2006, 2006.

[45] Erik Burman and Alexander Linke. Stabilized finite element schemes for incompressible flow using Scott–Vogelius elements. *Applied Numerical Mathematics*, 58(11):1704–1719, 2008.

[46] Zhiqiang Cai and Shun Zhang. Recovery-based error estimator for interface problems: Conforming linear elements. *SIAM Journal on Numerical Analysis*, 47(3):2132–2156, 2009.

[47] Stephen L. Campbell, Ilse C.F. Ipsen, C. Tim Kelley, and Carl D. Meyer. Gmres and the minimal polynomial. *BIT Numerical Mathematics*, 36(4):664–675, 1996.

[48] Eric Cancès and L. Ridgway Scott. van der Waals interactions between two hydrogen atoms: The Slater-Kirkwood method revisited. *submitted to SIMA*, TBD, 2016.

[49] Eduardo Casas, Andreas Günther, and Mariano Mateos. A paradox in the approximation of dirichlet control problems in curved domains. *SIAM Journal on Control and Optimization*, 49(5):1998–2007, 2011.

[50] Michael A. Case, Vincent J. Ervin, Alexander Linke, and Leo G. Rebholz. A connection between Scott-Vogelius and grad-div stabilized Taylor-Hood FE approximations of the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 49(4):1461–1481, 2011.

[51] Gregory A. Chechkin, Dag Lukkassen, and Annette Meidell. On the sapondzhyan–babuška paradox. *Applicable Analysis*, 87(12):1443–1460, 2008.

[52] Huangxin Chen, Xuejun Xu, and Weiying Zheng. Local multilevel methods for second order elliptic problems with highly discontinuous coefficients. *J. Camp. Math*, 30:223–248, 2012.

[53] Long Chen, Michael J. Holst, and Jinchao Xu. The finite element approximation of the nonlinear Poisson-Boltzmann equation. *SIAM journal on numerical analysis*, 45(6):2298–2320, 2007.

[54] Long Chen, Junping Wang, and Xiu Ye. A posteriori error estimates for weak galerkin finite element methods for second order elliptic problems. *Journal of Scientific Computing*, 59(2):496–511, 2014.

[55] Zhiming Chen and Shibin Dai. On the efficiency of adaptive finite element methods for elliptic problems with discontinuous coefficients. *SIAM Journal on Scientific Computing*, 24(2):443–462, 2002.

[56] Ray W. Clough. Original formulation of the finite element method. *Finite elements in analysis and design*, 7(2):89–101, 1990.

[57] Ivan Cormeau. Bruce Irons: A non-conforming engineering scientist to be remembered and rediscovered. *International Journal for Numerical Methods in Engineering*, 22(1):1–10, 1986.

[58] Martin Costabel and Monique Dauge. Singularities of electromagnetic fields in polyhedral domains. *Archive for Rational Mechanics and Analysis*, 151(3):221–276, 2000.

[59] Martin Costabel, Monique Dauge, and Christoph Schwab. Exponential convergence of hp-FEM for Maxwell equations with weighted regularization in polygonal domains. *Mathematical Models and Methods in Applied Sciences*, 15(04):575–622, 2005.

[60] Richard Courant et al. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc*, 49(1):1–23, 1943.

[61] W. Craig and C. Sulem. Numerical simulation of gravity waves. *Journal of Computational Physics*, 108(1):73–83, 1993.

[62] Franco Dassi, Simona Perotto, and Luca Formaggia. A priori anisotropic mesh adaptation on implicitly defined surfaces. *SIAM Journal on Scientific Computing*, 37(6):A2758–A2782, 2015.

[63] Franco Dassi, Hang Si, Simona Perotto, and Timo Streckenbach. Anisotropic finite element mesh adaptation via higher dimensional embedding. *Procedia Engineering*, 124:265–277, 2015.

[64] Luca Dedè, Stefano Micheletti, and Simona Perotto. Anisotropic error control for environmental applications. *Applied Numerical Mathematics*, 58(9):1320–1339, 2008.

[65] Alan Demlow and Natalia Kopteva. Maximum-norm a posteriori error estimates for singularly perturbed elliptic reaction-diffusion problems. *Numerische Mathematik*, 133:707742, 2016.

[66] Xie Dexuan, Yi Jiang, Peter Brune, and L. Ridgway Scott. A fast solver for a nonlocal dielectric continuum model. *SISC*, 34(2):B107–B126, 2012.

[67] Xie Dexuan, Yi Jiang, and L. Ridgway Scott. Efficient algorithms for solving a nonlocal dielectric model for protein in ionic solvent. *SISC*, 35(6):B1267B1284, 2014.

[68] Ibrahima Dione, Cristian Tibirna, and José Urquiza. Stokes equations with penalised slip boundary conditions. *International Journal of Computational Fluid Dynamics*, 27(6-7):283–296, 2013.

[69] Manfred Dobrowolski, Steffen Gräf, and Christoph Pflaum. On a posteriori error estimators in the finite element method on anisotropic meshes. *Electronic Transactions on Numerical Analysis*, 8:36–45, 1999.

[70] J. Douglas, Jr. and U. Hornung. *Flow in Porous Media: Proceedings of the Oberwolfach Conference, June 21-27, 1992*. Birkhauser, 1993.

[71] James J. Duderstadt and William R. Martin. *Transport Theory*. Wiley, 1979.

[72] D. Dutykh. *Modélisation mathématique des tsunamis*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2007.

[73] S. C. Eisenstat. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM J. Scientific and Stat. Comput.*, 2:1–4, 1981.

[74] Charles L. Epstein and Michael O'Neil. Smoothed corners and scattered waves. *arXiv preprint arXiv:1506.08449*, 2015.

[75] Lawrence C. Evans. *Partial differential equations*. Providence, Rhode Land: American Mathematical Society, 1998.

[76] Richard S. Falk. Approximation of the biharmonic equation by a mixed finite element method. *SIAM Journal on Numerical Analysis*, 15(3):556–567, 1978.

[77] Richard S. Falk and John E. Osborn. Error estimates for mixed methods. *RAIRO-Analyse numérique*, 14(3):249–277, 1980.

[78] David Chin-Lung Fong and Michael A. Saunders. CG versus MINRES: An empirical comparison. *SQU Journal for Science*, 17(1):44–62, 2012.

[79] Luca Formaggia, Stefano Micheletti, and Simona Perotto. Anisotropic mesh adaptation in computational fluid dynamics: application to the advection–diffusion–reaction and the stokes problems. *Applied Numerical Mathematics*, 51(4):511–533, 2004.

[80] Jean Baptiste Joseph Fourier. *Théorie analytique de la chaleur*. Firmin Didot, Paris, 1822.

[81] Jürgen Fuhrmann, Alexander Linke, Hartmut Langmach, and Helmut Baltruschat. Numerical calculation of the limiting current for a cylindrical thin layer flow cell. *Electrochimica Acta*, 55(2):430 – 438, 2009.

[82] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order*. Springer, second edition, 2001.

[83] V. Girault and L. Ridgway Scott. Analysis of a two-dimensional grade-two fluid model with a tangential boundary condition. *J. Math. Pures Appl.*, 78:981–1011, 1999.

[84] Roland Glowinski and Olivier Pironneau. Numerical methods for the first biharmonic equation and for the two-dimensional Stokes problem. *SIAM Review*, 21(2):167–212, 1979.

[85] Lars Grasedyck, Lu Wang, and Jinchao Xu. A nearly optimal multigrid method for general unstructured grids. *Numerische Mathematik*, 134(3):637–666, 2016.

[86] R. Green. The sweep of long water waves across the Pacific Ocean. *Australian journal of physics*, 14(1):120–128, 1961.

[87] Anne Greenbaum. *Iterative methods for solving linear systems*, volume 17. Siam, 1997.

[88] Anne Greenbaum, Vlastimil Pták, and Zdenvěk Strakoš. Any nonincreasing convergence curve is possible for GMRES. *Siam journal on matrix analysis and applications*, 17(3):465–469, 1996.

[89] Anne Greenbaum and Lloyd N. Trefethen. GMRES/CR and Arnoldi/Lanczos as matrix approximation problems. *SIAM Journal on Scientific Computing*, 15(2):359–368, 1994.

[90] Max D. Gunzburger. *Finite element methods for viscous incompressible flows.* Academic Press, 1989.

[91] I. A. Gustafson. Modified incomplete cholesky (MIC) methods. In *Preconditioning Methods: Analysis and Applications*, pages 265–294. Gordon and Breach Science Publishers, 1983.

[92] J.L. Hammack and H. Segur. The Korteweg-de Vries equation and water waves. part 2. comparison with experiments. *Journal of Fluid mechanics*, 65(02):289–314, 1974.

[93] John G. Heywood and Rolf Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem. I. Regularity of solutions and second-order error estimates for spatial discretization. *SIAM Journal on Numerical Analysis*, 19(2):275–311, 1982.

[94] John G. Heywood and Rolf Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem, part II: Stability of solutions and error estimates uniform in time. *SIAM journal on numerical analysis*, 23(4):750–777, 1986.

[95] Michael Holst, Nathan Baker, and Feng Wang. Adaptive multilevel finite element solution of the Poisson–Boltzmann equation I. Algorithms and examples. *Journal of computational chemistry*, 21(15):1319–1342, 2000.

[96] Lars Hörmander. *Linear Partial Differential Operators*. Springer-Verlag, 1969.

[97] A. Ilin, B. Bagheri, L.R.Scott, J.M.Briggs, and J.A.McCammon. Parallelization of Poisson-Boltzmann and Brownian Dynamics calculation. In *Parallel Computing in Computational Chemistry*, Washington D.C., to appear in 1995. ACS Books.

[98] C. T. Kelley. *Iterative methods for optimization*, volume 18. Siam, 1999.

[99] Youen Kervella, Denys Dutykh, and Frédéric Dias. Comparison between three-dimensional linear and nonlinear tsunami generation models. *Theoretical and computational fluid dynamics*, 21(4):245–269, 2007.

[100] Malcolm King, Bonnie Dasgupta, Robert P. Tomkiewicz, and Neil E. Brown. Rheology of cystic fibrosis sputum after in vitro treatment with hypertonic saline alone and in combination with recombinant human deoxyribonuclease i. *American journal of respiratory and critical care medicine*, 156(1):173–177, 1997.

[101] R. C. Kirby, M. Knepley, A. Logg, and L. R. Scott. Optimizing the evaluation of finite element matrices. *SIAM J. Sci. Computing*, 27:741–758, 2005.

[102] R. C. Kirby, A. Logg, L. R. Scott, and A. R. Terrel. Topological optimization of the evaluation of finite element matrices. *SIAM J. Sci. Computing*, 28:224–240, 2006.

[103] R. C. Kirby and L. R. Scott. Geometric optimization of the evaluation of finite element matrices. *SIAM J. Sci. Computing*, 29:827–841, 2007.

[104] Robert C. Kirby and Anders Logg. Efficient compilation of a class of variational forms. *ACM Transactions on Mathematical Software (TOMS)*, 33(3):17, 2007.

[105] Tzanio V. Kolev, Jinchao Xu, and Yunrong Zhu. Multilevel preconditioners for reaction-diffusion problems with discontinuous coefficients. *arXiv preprint arXiv:1411.7092*, 2014.

[106] Natalia Kopteva. Maximum-norm a posteriori error estimates for singularly perturbed reaction-diffusion problems on anisotropic meshes. *SIAM Journal on Numerical Analysis*, 53(6):2519–2544, 2015.

[107] Gerd Kunert. An a posteriori residual error estimator for the finite element method on anisotropic tetrahedral meshes. *Numerische Mathematik*, 86(3):471–490, 2000.

[108] Gerd Kunert. Robust a posteriori error estimation for a singularly perturbed reaction–diffusion equation on anisotropic tetrahedral meshes. *Advances in Computational Mathematics*, 15(1-4):237–259, 2001.

[109] Gerd Kunert and Serge Nicaise. Zienkiewicz–Zhu error estimators on anisotropic tetrahedral and triangular finite element meshes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(6):1013–1043, 2003.

[110] Gerd Kunert and Rüdiger Verfürth. Edge residuals dominate a posteriori error estimates for linear finite element methods on anisotropic triangular and tetrahedral meshes. *Numerische Mathematik*, 86(2):283–303, 2000.

[111] Samuel K. Lai, Ying-Ying Wang, Denis Wirtz, and Justin Hanes. Micro- and macrorheology of mucus. *Advanced drug delivery reviews*, 61(2):86–100, 2009.

[112] L.D. Landau and E.M. Lifshitz. *Fluid Dynamics*. Oxford: Pergammon, 1959.

[113] Alexander Linke. Collision in a cross-shaped domain – a steady 2d NavierStokes example demonstrating the importance of mass conservation in CFD. *Computer Methods in Applied Mechanics and Engineering*, 198(4144):3278 – 3286, 2009.

[114] Alexander Linke, Gunar Matthies, and Lutz Tobiska. Non-nested multi-grid solvers for mixed divergence-free Scott–Vogelius discretizations. *Computing*, 83(2-3):87–107, 2008.

[115] Alexander Linke, Leo G. Rebholz, and Nicholas E. Wilson. On the convergence rate of grad-div stabilized Taylor–Hood to Scott–Vogelius solutions for incompressible flow problems. *Journal of Mathematical Analysis and Applications*, 381(2):612–626, 2011.

[116] J. L. Lions. *Quelques méthodes de résolution des problèmes aux limites non linéaires.* Paris: Dunod, 1969.

[117] A. S. Lodge, W. G. Pritchard, and L. R. Scott. The hole–pressure problem. *IMA J. Applied Math.*, 46:39–66, 1991.

[118] A. Logg, K.A. Mardal, and G. Wells. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer-Verlag New York Inc, 2012.

[119] Anders Logg and Garth N Wells. Dolfin: Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)*, 37(2):20, 2010.

[120] N.D. Lopes, P.J.S. Pereira, and L. Trabucho. Improved Boussinesq equations for surface water waves. In A. Logg, K.A. Mardal, and G. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method: T he FEniCS Book*, pages 471–504. Springer-Verlag New York Inc, 2012.

[121] D. S. Malkus and E. T. Olsen. Linear crossed triangles for incompressible media. *North-Holland Mathematics Studies*, 94:235–248, 1984.

[122] David S. Malkus and Thomas J. R. Hughes. Mixed finite element methods–reduced and selective integration techniques: a unification of concepts. *Computer Methods in Applied Mechanics and Engineering*, 15(1):63–81, 1978.

[123] Renier Gustav Marchand. *The method of manufactured solutions for the verification of computational electromagnetic codes.* PhD thesis, Stellenbosch University, 2013.

[124] Kent-Andre Mardal and Ragnar Winther. Preconditioning discretizations of systems of partial differential equations. *Numerical Linear Algebra with Applications*, 18(1):1–40, 2011.

[125] V. G. Mazya. *Sobolev spaces.* New York : Springer-Verlag, 1985.

[126] Vladimir Gilelevich Maz'ya and Sergei Aleksandrovich Nazarov. About the sapondzhyn-babuška paradox in the plate theory. *Dokl. Akad. Nauk. Arm. Rep*, 78:127–130, 1984.

[127] Vladimir Gilelevich Maz'ya and Sergei Aleksandrovich Nazarov. Paradoxes of limit passage in solutions of boundary value problems involving the approximation of smooth domains by polygonal domains. *Izvestiya: Mathematics*, 29(3):511–533, 1987.

[128] Vladimir Mazya, Serguei Nazarov, and Boris A Plamenevskij, editors. *Asymptotic Theory of Elliptic Boundary Value Problems in Singularly Perturbed Domains.* Springer, 2000.

[129] N. G. Meyers. An L^p-estimate for the gradient of solutions of second order elliptic divergence equations. *Annali della Scuola Normale Superiore di Pisa. Ser. III.*, XVII:189–206, 1963.

[130] S Micheletti and S Perotto. Anisotropic recovery-based a posteriori error estimators for advection-diffusion-reaction problems. In Andrea Cangiani, Ruslan L. Davidchack, Emmanuil Georgoulis, Alexander N. Gorban, Jeremy Levesley, and Michael V. Tretyakov, editors, *Numerical Mathematics and Advanced Applications 2011*, pages 43–51. Springer, 2013.

[131] Stefano Micheletti and Simona Perotto. Reliability and efficiency of an anisotropic Zienkiewicz–Zhu error estimator. *Computer methods in applied mechanics and engineering*, 195(9):799–835, 2006.

[132] Hannah Morgan and L. Ridgway Scott. Towards the ultimate finite element method for the stokes equations. *SISC*, submitted, 2017.

[133] J. Morgan and L. R. Scott. A nodal basis for C^1 piecewise polynomials of degree $n \geq 5$. *Math. Comp.*, 29:736–740, 1975.

[134] John Morgan and L. R. Scott. The dimension of the space of C^1 piecewise–polynomials. Research Report UH/MD 78, Dept. Math., Univ. Houston, 1990.

[135] Sergei Aleksandrovich Nazarov and Maksim Viktorovich Olyushin. Approximation of smooth contours by polygonal ones. paradoxes in problems for the Lamé system. *Izvestiya: Mathematics*, 61(3):619, 1997.

[136] David P. Nicholls. Traveling water waves: Spectral continuation methods with parallel implementation. *Journal of Computational Physics*, 143(1):224 – 240, 1998.

[137] David P. Nicholls. Boundary perturbation methods for water waves. *GAMM-Mitteilungen*, 30(1):44–74, 2007.

[138] David P. Nicholls and Fernando Reitich. Stable, high-order computation of traveling water waves in three dimensions. *European Journal of Mechanics - B/Fluids*, 25(4):406 – 424, 2006.

[139] Christopher C. Paige and Michael A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12(4):617–629, 1975.

[140] Linus Pauling and E. Bright Wilson. *Introduction to Quantum Mechanics with Applications to Chemistry*. Dover, 1985.

[141] Martin Petzoldt. *Regularity and error estimators for elliptic problems with discontinuous coefficients*. PhD thesis, Freie Universität Berlin, Germany, 2001.

[142] Martin Petzoldt. A posteriori error estimators for elliptic equations with discontinuous coefficients. *Advances in Computational Mathematics*, 16(1):47–75, 2002.

[143] Marco Picasso. An anisotropic error indicator based on Zienkiewicz–Zhu error estimator: Application to elliptic and parabolic problems. *SIAM Journal on Scientific Computing*, 24(4):1328–1355, 2003.

[144] Marco Picasso. Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives. *Computer Methods in Applied Mechanics and Engineering*, 196(1):14–23, 2006.

[145] Ekkehard Ramm, E. Rank, R. Rannacher, K. Schweizerhof, E. Stein, W. Wendland, G. Wittum, Peter Wriggers, and Walter Wunderlich. *Error-controlled adaptive finite elements in solid mechanics*. John Wiley & Sons, 2003.

[146] Rolf Rannacher. Finite-element approximation of simply supported plates and the babuska paradox. *ZEITSCHRIFT FUR ANGEWANDTE MATHEMATIK UND MECHANIK*, 59(3):T73–T76, 1979.

[147] G Rieder. On the plate paradox of sapondzhyan and babuška. *Mechanics Research Communications*, 1(1):51–53, 1974.

[148] Marie E. Rognes, Robert C. Kirby, and Anders Logg. Efficient assembly of h(div) and h(curl) conforming finite elements. *SIAM Journal on Scientific Computing*, 31(6):4130–4151, 2009.

[149] Marie E. Rognes and Anders Logg. Automated goal-oriented error control i: Stationary variational problems. *SIAM Journal on Scientific Computing*, 35(3):C173–C193, 2013.

[150] Walter Rudin. *Functional analysis.* New York: McGraw-Hill, 1991. Second Edition.

[151] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[152] Stephan Schmidt. A two stage CVT/eikonal convection mesh deformation approach for large nodal deformations. *arXiv preprint arXiv:1411.7663*, 2014.

[153] G. Schneider and C. E. Wayne. The long-wave limit for the water wave problem I. the case of zero surface tension. *Communications on Pure and Applied Mathematics*, 53(12):1475–1535, 2000.

[154] Dominik Schötzau, Christoph Schwab, and Rolf Stenberg. Mixed hp-FEM on anisotropic meshes II: Hanging nodes and tensor products of boundary layer meshes. *Numerische Mathematik*, 83(4):667–697, 1999.

[155] L. R. Scott. Elliptic preconditioners using fast summation techniques. In *Domain Decomposition Methods in Science and Engineering (Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University), D. E. Keyes & J. Xu, eds.*, volume Contemproary Mathematics 180, pages 311–323. American Mathematical Society, 1995.

[156] L. R. Scott, A. Ilin, R. Metcalfe, and B. Bagheri. Fast algorithms for solving high-order finite element equations for incompressible flow. In *Proceedings of the International Conference on Spectral and High Order Methods*, pages 221–232, Houston, TX, 1995. University of Houston.

[157] L. R. Scott and M. Vogelius. Conforming finite element methods for incompressible and nearly incompressible continua. In *Large Scale Computations in Fluid Mechanics, B. E. Engquist, et al., eds.*, volume 22 (Part 2), pages 221–244. Providence: AMS, 1985.

[158] L. R. Scott and M. Vogelius. Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials. *M^2AN (formerly R.A.I.R.O. Analyse Numérique)*, 19:111–143, 1985.

[159] L. Ridgway Scott. *Numerical Analysis.* Princeton Univ. Press, 2011.

[160] R. Scott. Interpolated boundary conditions in the finite element method. *SIAM J. Numer. Anal.*, 12:404–427, 1975.

[161] R. Scott. A survey of displacement methods for the plate bending problem. In *Formulations and Computational Algorithms in Finite Element Analysis, K.–J. Bathe, J. T. Oden, and W. Wunderlich, eds.*, pages 855–876. Cambridge: M. I. T. Press, 1977.

[162] Kunibert G. Siebert. An a posteriori error estimator for anisotropic refinement. *Numerische Mathematik*, 73(3):373–398, 1996.

[163] Pavel Šolín, Jakub Červenỳ, and Ivo Doležel. Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM. *Mathematics and Computers in Simulation*, 77(1):117–132, 2008.

[164] Rob Stevenson. An optimal adaptive finite element method. *SIAM journal on numerical analysis*, 42(5):2188–2217, 2005.

[165] Jung Soo Suk, Samuel K. Lai, Ying-Ying Wang, Laura M. Ensign, Pamela L. Zeitlin, Michael P. Boyle, and Justin Hanes. The penetration of fresh undiluted sputum expectorated by cystic fibrosis patients by non-adhesive polymer nanoparticles. *Biomaterials*, 30(13):2591–2597, 2009.

[166] Guido Sweers. A survey on boundary conditions for the biharmonic. *Complex Variables and Elliptic Equations*, 54(2):79–93, 2009.

[167] V. Szebehely, D. Saari, J. Waldvogel, and U. Kirchgraber. Eduard L. Stiefel (1909–1978). *Celestial Mechanics and Dynamical Astronomy*, 21:2–4, Jan. 1980. 10.1007/BF01230237.

[168] Robert Tarran, Brian Button, Maryse Picher, Anthony M. Paradiso, Carla M. Ribeiro, Eduardo R. Lazarowski, Liqun Zhang, Peter L. Collins, Raymond J. Pickles, Jeffrey J. Fredberg, et al. Normal and cystic fibrosis airway surface liquid homeostasis: The effects of phasic shear stress and viral infections. *Journal of Biological Chemistry*, 280(42):35751–35759, 2005.

[169] Andy R. Terrel, L. Ridgway Scott, Matthew Gregg Knepley, Robert C. Kirby, and Garth N. Wells. Finite elements for incompressible fluids. In A. Logg, K.A. Mardal, and G. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, pages 381–394. Springer-Verlag New York Inc, 2012.

[170] V. Thomee. *Galerkin Finite Element Methods for Parabolic Problems*. Springer Verlag, 1997.

[171] R Verfürth. Finite element approximation of steady Navier-Stokes equations with mixed boundary conditions. *RAIRO-Modélisation mathématique et analyse numérique*, 19(3):461–475, 1985.

[172] Rüdiger Verfürth. *A posteriori error estimation techniques for finite element methods*. Oxford University Press, 2013.

[173] Martin Vohralík. Guaranteed and fully robust a posteriori error estimates for conforming discretizations of diffusion problems with discontinuous coefficients. *Journal of Scientific Computing*, 46(3):397–438, 2011.

[174] M. N. Vu, S. Geniaut, P. Massin, and J. J. Marigo. Numerical investigation on corner singularities in cracked plates using the G-theta method with an adapted θ field. *Theoretical and Applied Fracture Mechanics*, 77:59 – 68, 2015.

[175] Xiaoping Xie, Jinchao Xu, and Guangri Xue. Uniformly-stable finite element methods for darcy-stokes-brinkman models. *Journal of Computational Mathematics*, pages 437–455, 2008.

[176] S. Zhang and L. R. Scott. Finite element interpolation of non–smooth functions satisfying boundary conditions. *Math. Comp.*, 54:483–493, 1990.

[177] Shangyou Zhang. Divergence-free finite elements on tetrahedral grids for $k \geq 6$. *Math. Comp.*, 80:669695, 2011.

[178] Shangyou Zhang and L. R. Scott. Higher dimensional non-nested multigrid methods. *Math. Comp.*, 58:457–466, 1992.