Shape Optimisation and Robust Solvers for Incompressible Flow



Florian Wechsung

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

Contents

1 Introduction **5**

1.1 Mathematical formulations of optimisation over shapes 5

1.2 Structure and contributions of this thesis 7

1.2.1 Shape optimisation 7

1.2.2 Fast solvers for the incompressible Navier–Stokes equations . . 8

1.2.3 Shape optimisation with multigrid solvers 9

1.3 Software 10

I Shape optimisation **11**

2 Shape optimisation and shape calculus **12**

2.1 General spaces of deformations 13

2.2 Discussion of the necessary regularity of deformations 19

2.3 Reduction of PDE constraints via adjoint solutions 23

2.4 Structure of Shape Derivatives 27

2.5 Numerical shape optimisation 30

2.5.1 Optimisation 30

2.5.2 Discretisation 31

2.5.3 Software 32

3 Automated calculation of shape derivatives **35**

3.1 Shape differentiation on the reference element 36

3.2 Examples 39

3.3 Code validation 44

3.4 Shape optimisation of a pipe 45

4 Nearly conformal mesh deformation methods for shape optimisation in two dimensions 50
4.1 Discussion of deformations and mesh quality 51
4.2 Conformal mappings in two dimensions 53
4.3 Nearly conformal shape gradients 55
4.3.1 CR(α) + $\mathring{H}^1$-shape gradients 56
4.3.2 Nearly conformal shape gradients with mixed boundary conditions 61
4.3.3 CR(α) + H(sym)-shape gradients 62
4.4 Numerical experiments 63
4.4.1 Levelset example 63
4.4.2 A negative example: Annulus deformation 68
4.4.3 Energy minimisation in slow flow 69
4.5 Discussion of the three dimensional case 75

II Large scale solvers for the Navier–Stokes equations 77

5 Augmented Lagrangian preconditioning for the stationary incompressible Navier–Stokes equations 78
5.1 Preconditioning Strategies 80
5.2 Augmented Lagrangian preconditioning 82
5.3 Solving the top-left block 84
5.4 Subspace correction methods for nearly singular problems 86

6 Augmented Lagrangian preconditioning for a low order discretisation 95
6.1 Robust smoothing and prolongation in two dimensions 96
6.1.1 Prolongation 96
6.1.2 Relaxation 98
6.2 Robust smoothing and prolongation in three dimensions 103
6.2.1 Prolongation 103
6.2.2 Smoothing 106
6.3 Including the advection term 107
6.4 Numerical Results 109
6.4.1 Algorithm details 109
6.4.2 Solving the top-left block only: nearly incompressible elasticity 110
6.4.3 Solver verification with the method of manufactured solutions 114

6.4.4 Two-dimensional experiments 116
6.4.5 Three-dimensional experiments 120
6.4.6 Computational performance 121

7 Augmented Lagrangian preconditioning for divergence-free finite element pairs 125
7.1 Stability of the Scott-Vogelius element 126
7.2 Solving the top-left block 128
7.2.1 Robust prolongation 128
7.2.2 Robust smoothing 130
7.3 Stabilisation for the advection terms 136
7.4 Numerical results 137
7.4.1 Solving the top-left block only: nearly incompressible elasticity 137
7.4.2 Verification and pressure robustness 139
7.4.3 Algorithmic performance in two and three dimensions 140

III Combination and conclusions 145

8 Shape optimisation with geometric multigrid solvers 146
8.1 Nestedness preserving high-order deformations 146
8.2 Regularisation 150
8.3 Drag minimisation of an immersed object 152
8.4 Optimisation of a pipe 155

9 Conclusions and outlook 161
9.1 Shape Optimisation 161
9.2 Robust large-scale solvers for the incompressible Navier–Stokes equations 162

Acknowledgements

I would like to thank the following people who made this thesis possible:

My supervisor Patrick Farrell for his excellent guidance, supervision, and advice. His curiosity and enthusiasm for solving problems have been a great source of inspiration and motivation throughout these years.

Lawrence Mitchell, Alberto Paganini, and Kevin Sturm, who were fantastic collaborators and excellent sparring partners in many discussions on both analysis and implementation.

Mark Taylor, for introducing me to the engineering side of computational fluid dynamics and giving me a glimpse into the world of race cars.

Chris Breward and Colin Please, for a great CDT and their support throughout four years of InFoMM.

The students of InFoMM cohort 2 and the students in the numerical analysis group for making even long working days enjoyable. In particular Abi, Bogdan, Matt, Michael, Thomas, and Tino for the fantastic time spent together in Oxford and abroad.

The wonderful Caoimhe, for her continuous love and support throughout both good and stressful times.

My parents and brother, who have always supported and encouraged me in all endeavours. This would not have been possible without you.

Abstract

When designing a new car or a plane, engineers need to solve the Navier–Stokes equations to understand how air flows around the object. Based on experience and intuition, they modify the design slightly, then solve the equations again, and inspect the changes. This process is iterated many times until a final design that minimises or maximises some quantity of interest, such as drag or lift, is found. The goal of shape optimisation is to automate this type of process.

In this thesis we address several issues related to shape optimisation. Focussing on the case when the shape is discretised using a mesh and when PDE constraints are solved using the finite element method, we describe a reformulation of the shape derivative as the derivative of the pushforward from the reference element. This viewpoint allows for automated calculation of shape derivatives in finite element software. When shape optimisation is performed by deforming an initial mesh, the choice of deformation is important. We propose a new Hilbert space structure on the space of deformations that results in high mesh quality of the deformed domains.

We then focus on the solution of a particular PDE constraint given by the steady, incompressible Navier–Stokes equations that govern laminar flow. The solution of these equations becomes challenging for large Reynolds number. We develop augmented Lagrangian based preconditioners that exhibit robust performance as the Reynolds number is increased. The effectiveness and scalability of the developed solvers is demonstrated for a range of test problems.

Chapter 1

Introduction

The design of optimal shapes is a task faced by engineers in a wide range of disciplines. Often this problem can be expressed as minimisation of an objective functional J over a set of admissible shapes $\mathcal{U}_{\text{ad}}$, i.e. it is of the form

$$\underset{\Omega \in \mathcal{U}_{\text{ad}}}{\text{minimise}}\ J(\Omega). \tag{1.1}$$

Classical objectives are the drag or lift of a plane, the downforce of a race car, the strength of a structure, or the efficiency of an acoustic device.

1.1 Mathematical formulations of optimisation over shapes

There is a range of difficulties associated with finding an optimal shape. The first is that there is no obvious structure on the set of shapes. There are two popular approaches for reformulating the optimisation in a tractable manner. In topology optimisation, one considers a function $\varphi : \mathbb{R}^d \to \{0, +1\}$ and then defines the domain as $\Omega := \{\varphi = 1\}$. Optimisation over shapes is thus translated to optimisation over all such functions φ. However, this problem is, in general, ill-posed and does not admit optimal solutions as often designs can be improved by adding finer and finer microstructures [SP98]. In practice this means that solutions are mesh dependent and that often the solution that was found cannot be manufactured.

A common approach to counter this is to relax the problem to a search for functions $\varphi : \mathbb{R}^d \to [0, 1]$ and then to penalise intermediate values of φ; this leads to a diffuse but narrow boundary of Ω. In order to avoid fine microstructures, one adds a regularisation term, e.g. in the form of the gradient of φ; this is the diffuse analogue to penalising the perimeter of the resulting shape. The SIMP (solid isotropic material with penalisation) [Ben89] and the phase field approach [BC03] are of this type.

Another problem arises when considering problems with a constraint given by a partial differential equation (PDE): since the domain obtained using a regularisation has a diffuse interface, it is not clear where to solve the PDE. The usual solution is to consider a so-called *Ersatzmaterial* approach. In the case of aerodynamic optimisation this means that one considers the car or wing as a porous medium with extremely low porosity [Gar+16] and solves for the flow in the entire space. However, this can lead to both inaccuracies and ill-conditioning of the PDE problem to be solved.

The approach of shape optimisation is different. Here, one assumes that one is given a reasonable initial guess for the shape and then deforms this initial shape in order to minimise the functional [Mic72; MS76; Sim80; DZ11; HP15].

In parametric shape optimisation, this is done by parameterising either the shape itself or deformations of the shape and then optimising for the coefficients of the parameterisation. This has the advantage that the optimisation then reduces to an optimisation problem in $\mathbb{R}^n$, for which a vast amount of theory and algorithms exist [NW06]. However, only a small, finite-dimensional space of shapes can be represented by such parameterisations and hence the desired optimal shape might not be included in the search space. For these reasons, we follow the alternative, nonparametric approach and consider deformations that lie in an infinite-dimensional space $\mathcal{D}$ of vector fields:

$$\underset{\mathbf{T} \in \mathcal{D}}{\text{minimise}} \; J(\mathbf{T}(\Omega)). \tag{1.2}$$

For many applications, the objective depends not only on the shape, but also on the solution to some PDE defined on the shape. Given a function space $\mathcal{X}(\Omega)$ on Ω we denote the PDE constraint on Ω by $e(\Omega, \cdot) : \mathcal{X}(\Omega) \to \mathcal{X}(\Omega)^*$. Then the optimisation problem to be solved is given by

$$\begin{aligned} \underset{\mathbf{T} \in \mathcal{D},\; u \in \mathcal{X}(\mathbf{T}(\Omega))}{\text{minimise}} \quad & J(\mathbf{T}(\Omega), u) \\ \text{subject to} \quad & e(\mathbf{T}(\Omega), u) = 0. \end{aligned} \tag{1.3}$$

Shape optimisation thus combines functional analysis and geometry with two areas of numerical analysis. First, it requires the development of efficient optimisation algorithms on infinite-dimensional function spaces. These can often be borrowed from classical PDE constrained optimisation but usually need to be modified in order to take the nonstandard nature of the control into account. Secondly, since the functional needs to be evaluated for many different shapes, it requires the ability to solve the constraining PDE in an efficient way. This twofold nature of shape optimisation is reflected in the structure of this thesis. Part I discusses several aspects of shape optimisation, whereas Part II is focussed on the fast solution of a particular PDE

constraint, namely the incompressible Navier–Stokes equations. Lastly, in Part III we address some issues that arise when using geometric multigrid solvers for shape optimisation problems.

1.2 Structure and contributions of this thesis

1.2.1 Shape optimisation

In order to perform efficient optimisation over a large control space, the calculation of derivatives is crucial. In the case of shape optimisation, this itself poses a challenge, in particular when the objective is subject to a PDE constraint, as both the PDE and its solution change as the domain deforms. Applying the rules of shape calculus is often tedious and prone to error: even for simple objectives and constraints these calculations regularly span many pages in papers and dissertations on shape optimisation.

To express partial differential equations in weak form, Alnæs et al. developed the Unified Form Language (UFL) [Aln+14]. This language has become the foundation for the increasingly popular FEniCS [LMW12] and Firedrake [Rat+16] finite element libraries. Furthermore, the dolfin-adjoint [Far+13] project has demonstrated that this language can be used to automatically derive adjoints for PDE constrained optimisation problems. We will describe an extension to UFL for automatically and symbolically deriving shape derivatives of a large class of integrals. The key insight of this implementation is that a classical Gâteaux derivative with respect to the pull-back performed when assembling integrals over finite element functions is equivalent to shape differentiation. The approach is generic, extends to higher order derivatives, and makes shape optimisation easily accessible to users of FEniCS and Firedrake. In particular, it enables rapid development of solvers for shape optimisation problems that are constrained by complicated, nonlinear PDEs such as the Navier–Stokes equations. This work is presented in Chapter 3 and has been published in [Ham+19].

Once the shape derivative has been calculated, it can be used to obtain deformations of the domain that lead to a reduction of the objective. These deformations are often obtained by calculating the Riesz representative of the shape derivative in a particular Hilbert space of deformations. When the domain is represented by a mesh, changing the shape corresponds to moving the vertices of the mesh. However, care is required in order to avoid tangling, stretching, or kinks when deforming the mesh. This is crucial, as highly stretched elements can reduce the accuracy of the finite element solution or even lead to failure of convergence.

One way to avoid such problems is to consider conformal mappings for the mesh deformations. These mappings preserve angles and hence avoid stretching of the elements in the mesh. In two dimensions they can be characterised by the Cauchy–Riemann equations. In Chapter 4 we will describe a procedure that augments a given inner product with a least-squares formulation of the Cauchy–Riemann equations to change the geometry of the function space in a way that penalises deformations that are not conformal. We prove that in a particular limit, this yields perfectly conformal deformations and demonstrate high mesh quality in several numerical examples. This work has been published in [ISW18].

1.2.2 Fast solvers for the incompressible Navier–Stokes equations

In the case of aerodynamic and hydrodynamic shape optimisation, the PDE constraint is often given by some variant of the incompressible Navier–Stokes equations. A good solver for the linear systems arising in Newton's method applied to these equations should have two properties: first, that its computational complexity scales well with the number of degrees of freedom (dofs); and secondly, that its convergence does not degrade as parameters such as the Reynolds number are varied. It has proven very difficult to develop solvers that exhibit both properties; matrix factorisations are robust to Reynolds number but scale badly with dof count, whereas Schur complement based algorithms such as pressure convection-diffusion (PCD) and least-squares commutator (LSC) scale linearly in the dof count but their performance decreases as the Reynolds number is increased [KLW02; Elm+06; ESW14].

In 2006, Benzi & Olshanskii [BO06] proposed a preconditioner of augmented Lagrangian type for the incompressible Navier–Stokes equations. The key idea is to augment the momentum block with a term penalising the divergence of the velocity. Since we require incompressibility, this does not change the discrete solution of the system but it turns out that it simplifies the approximation of the Schur complement significantly. However, since this added term has a large kernel consisting of solenoidal vector fields, solving the momentum block becomes difficult. Benzi & Olshanskii proposed a multigrid scheme for this block in two dimensions that builds on previous work of Schöberl [Sch99a; Sch99b]. Before focussing on particular discretisations, we describe the general strategy of augmented Lagrangian preconditioning and then outline the framework for robust multigrid schemes developed by Schöberl in Chapter 5.

The work of Benzi & Olshanskii focussed on a particular low order discretisation in two dimensions. In Chapter 6 we extend their work and present a Reynolds-robust preconditioner for the stationary, incompressible Navier–Stokes equations in three dimensions. While the same general strategy applies as in two dimensions, the prolongation operator in three dimensions requires modifications in order to obtain a robust scheme. We give an efficient, fully parallel implementation on unstructured grids and demonstrate only very mild growth of iteration counts up to Reynolds number as large as 10 000 in two dimensions and 5 000 in three dimensions. This work has been published in [FMW19].

A drawback of the approach of Benzi & Olshanskii is that it requires piecewise constant pressures and is hence limited to first order convergence. In addition, the utilised finite elements only enforce the divergence constraint weakly and as a consequence the discretisation is not pressure robust: errors in the pressure solution affect the velocity. As discussed in the recent survey [Joh+17], this lack of pressure robustness particularly affects the solution at high Reynolds number: the velocity error has a term that scales with the Reynolds number multiplied by the pressure error. As a result, discretisations which do satisfy pressure robustness are becoming increasingly popular. In Chapter 7 we will demonstrate how a similar scheme can be developed for the pressure robust Scott-Vogelius finite element [SV85]. However, since this element is known not to be inf-sup stable on arbitrary meshes, we consider barycentrically refined meshes instead. These meshes result in a nonnested hierarchy for which we will develop a suitable prolongation operator. In addition, motivated by recent results in finite element exterior calculus, we will construct a tailored relaxation operator to obtain an effective multigrid scheme. This combination of finite element and preconditioner yields a solver with performance and approximation quality that are both robust with respect to the Reynolds number. A manuscript on this work is in preparation.

1.2.3 Shape optimisation with multigrid solvers

The multigrid solvers that we develop in Part II rely heavily on mesh hierarchies with a (partially) nested structure. In Chapter 8 we will describe a shape optimisation strategy that is based on high order deformations of the coarse grid and preserves the nestedness of the hierarchy. We will combine the automated calculation of shape derivatives from Chapter 3, the Cauchy–Riemann based inner product from Chapter 4, and the solver developed in Chapter 6 to perform shape optimisation for two classical problems in two and three dimensions.

Finally, in Chapter 9 we summarise the work and outline possible extensions and directions for future study.

1.3 Software

The methods and algorithms developed as part of this thesis have been implemented using the Firedrake [Rat+16] and PETSc [Bal+18] libraries. Links to the code implementing the numerical examples are given at the end of each chapter.

Part I

Shape optimisation

Chapter 2

Shape optimisation and shape calculus

A shape functional is a function that maps domains to real numbers.

Definition 2.1.

Let $\mathcal{A} \subset \mathcal{P}(\mathbb{R}^d) := \left\{\Omega : \Omega \subset \mathbb{R}^d\right\}$ *be a set of admissible shapes. A* shape functional *is a map*

$$J : \mathcal{A} \to \mathbb{R}. \tag{2.1}$$

Classically, shape optimisation has been studied by considering deformations of an initial domain Ω, i.e. one considers vector fields $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ and then calculates $J(\mathbf{T}(\Omega))$ [MS76; Sim80; DZ11; HP15]. These deformations are typically contained in the space of differentiable or Lipschitz continuous functions. Though we assume the reader to be familiar with these spaces, we recall their definition and elementary properties in order to set the notation.

For an open set $\Omega \subset \mathbb{R}^d$ we denote the space of *k-times continuously differentiable functions* from Ω to $\mathbb{R}^m$ by $C^k(\Omega; \mathbb{R}^m)$. In order to equip these functions with a Banach space structure, we need to require that they are well-behaved at the boundary: we denote the space of functions $f \in C^k(\Omega; \mathbb{R}^m)$ so that f and its derivatives can continuously be extended to $\overline{\Omega}$ by $C^k(\overline{\Omega}; \mathbb{R}^m)$. For $0 \leq l \leq k$ we can define the seminorms

$$|f|_{C^l(\overline{\Omega};\mathbb{R}^m)} := \begin{cases} \sup_{x\in\overline{\Omega}} \left|\left|\left| f(x) \right|\right|\right|_2 & \text{if } l = 0, \\ \sup_{x\in\overline{\Omega}} \left|\left|\left| \mathrm{D}^l f(x) \right|\right|\right|_2 & \text{if } l \geq 1, \end{cases} \tag{2.2}$$

where $\left|\left|\left|\cdot\right|\right|\right|_2$ is the usual 2-norm for (multi-)linear operators. Using these we can define the norms

$$\|f\|_{C^k(\overline{\Omega};\mathbb{R}^m)} := \sum_{l=0}^{k} |f|_{C^l(\overline{\Omega};\mathbb{R}^m)}\,, \tag{2.3}$$

and indeed $(C^k(\overline{\Omega}; \mathbb{R}^m), \|\cdot\|_{C^k(\overline{\Omega};\mathbb{R}^m)})$ is a Banach space.

Sometimes, enforcing differentiability of the deformations is too strict a requirement; Lipschitz deformations are an alternative that are well-behaved almost everywhere but allow for kinks. A function $f : \Omega \to \mathbb{R}^m$ is said to be *Lipschitz continuous* if

$$\exists c > 0, \forall x, y \in \Omega, \ \|f(y) - f(x)\|_2 \leq c \|x - y\|_2 . \tag{2.4}$$

The smallest c for which the above inequality holds is denoted by $\mathrm{Lip}(f)$. We denote the spaces of Lipschitz continuous functions by $C^{0,1}(\Omega; \mathbb{R}^m)$ and the space of Lipschitz continuously k-times differentiable functions by $C^{k,1}(\Omega; \mathbb{R}^m)$. As before we define the spaces $C^{k,1}(\overline{\Omega}; \mathbb{R}^m)$ and these are Banach spaces with the norm

$$\|f\|_{C^{k,1}(\overline{\Omega};\mathbb{R}^m)} := \max \left\{ \|f\|_{C^k(\overline{\Omega};\mathbb{R}^m)} , \max_{0 \leq |\alpha| \leq k} \sup_{x,y \in \Omega, \ x \neq y} \frac{\|\partial^\alpha f(y) - \partial^\alpha f(x)\|_2}{\|x - y\|_2} \right\}. \tag{2.5}$$

From now on we will use boldface to denote vector fields. Furthermore, capital letters will denote vector fields that are used to deform the domain whereas lower case letters will be used in the context of PDE constraints on the domain.

2.1 General spaces of deformations

For a given space $\mathcal{D}$ of functions from $\mathbb{R}^d$ to $\mathbb{R}^d$, we denote perturbations of the identity by

$$\mathbf{T}_\mathbf{X} := \mathbf{Id} + \mathbf{X}, \quad \mathbf{X} \in \mathcal{D}. \tag{2.6}$$

In order to avoid overlapping, compressing or tearing of the initial shape Ω when deforming it, we cannot consider arbitrary $\mathbf{X}$ but only those so that $\mathbf{T}_\mathbf{X}$ is bijective and $(\mathbf{T}_\mathbf{X})^{-1}$ is itself a perturbation of the identity, i.e. $(\mathbf{T}_\mathbf{X})^{-1} = \mathbf{Id} + \mathbf{Y}$ for some $\mathbf{Y} \in \mathcal{D}$.

This idea was introduced by Micheletti in [Mic72] for spaces of differentiable functions and subsequently extended to more general function spaces. We follow the exposition in [DZ11].

Definition 2.2.
Let $\mathcal{D}$ be a real vector space of mappings from $\mathbb{R}^d$ to $\mathbb{R}^d$. We define the Micheletti space of deformations $\mathcal{F}(\mathcal{D})$ *via*

$$\mathcal{F}(\mathcal{D}) := \left\{ \mathbf{T}_\mathbf{X} = \mathbf{Id} + \mathbf{X} : \mathbf{X} \in \mathcal{D}, \mathbf{T}_\mathbf{X} \text{ bijective and } \mathbf{T}_\mathbf{X}^{-1} - \mathbf{Id} \in \mathcal{D} \right\}. \tag{2.7}$$

We will use this notation for the rest of this thesis: for $\mathbf{X} \in \mathcal{D}$, $\mathbf{T}_\mathbf{X} = \mathbf{Id} + \mathbf{X}$ denotes the perturbation of the identity by $\mathbf{X}$ and $\Omega_\mathbf{X} = \mathbf{T}_\mathbf{X}(\Omega)$ denotes the perturbed domain.

One can show that if $\mathcal{D}$ satisfies a closedness property, then $\mathcal{F}(\mathcal{D})$ is a group for the composition operator.

Theorem 2.3.

Let $\mathcal{D}$ be a real vector space of mappings from $\mathbb{R}^d$ to $\mathbb{R}^d$. Then $\mathcal{F}(\mathcal{D})$ is a group for the composition $\circ$ if and only if

$$\forall\ \mathbf{Y} \in \mathcal{D},\ \forall\ \mathbf{Id} + \mathbf{X} \in \mathcal{F}(\mathcal{D}), \quad \mathbf{Y} \circ (\mathbf{Id} + \mathbf{X}) \in \mathcal{D}. \tag{2.8}$$

In particular $(\mathbf{Id} + \mathbf{X})^{-1} - \mathbf{Id} = -\mathbf{X} \circ (\mathbf{Id} + \mathbf{X})^{-1} \in \mathcal{D}$.

Proof. See [DZ11, Chapter 3, Theorem 2.1]. □

This assumption is satisfied for $\mathcal{D} = C^{k,1}(\overline{\mathbb{R}^d}; \mathbb{R}^d)$ and for $\mathcal{D} = C^k(\overline{\mathbb{R}^d}; \mathbb{R}^d)$ for $k \geq 0$. If $\mathcal{D}$ is a normed space, then one can define a metric d on $\mathcal{F}(\mathcal{D})$ via

$$d(\mathbf{Id}, \mathbf{T}) := \inf_{\substack{n \in \mathbb{N}_{>0},\ \mathbf{T}_i \in \mathcal{F}(\mathcal{D}), \\ \mathbf{T} = \mathbf{T}_1 \circ \cdots \circ \mathbf{T}_n}} \sum_{i=1}^{n} \|\mathbf{T}_i - \mathbf{Id}\|_{\mathcal{D}} + \left\|\mathbf{T}_i^{-1} - \mathbf{Id}\right\|_{\mathcal{D}} \tag{2.9}$$

and

$$d(\mathbf{T}, \mathbf{S}) := d(\mathbf{Id}, \mathbf{S} \circ \mathbf{T}^{-1}). \tag{2.10}$$

Under some assumptions that are again satisfied for the case $\mathcal{D} = C^{k,1}(\overline{\mathbb{R}^d}; \mathbb{R}^d)$ and for $\mathcal{D} = C^k(\overline{\mathbb{R}^d}; \mathbb{R}^d)$, d is indeed a metric on $\mathcal{F}(\mathcal{D})$.

Theorem 2.4.

Let $\mathcal{D}$ be equal to $C^k(\overline{\mathbb{R}^d}; \mathbb{R}^d)$ or $C^{k,1}(\overline{\mathbb{R}^d}; \mathbb{R}^d)$, $k \geq 0$. Then $(\mathcal{F}(\mathcal{D}), d)$ is a complete metric space.

Proof. See [DZ11, Chapter 3, Theorem 2.6]. □

The definition of the Micheletti spaces motivates to consider the following functional

$$\begin{aligned} J_\Omega : \mathcal{D} &\to \mathbb{R} \\ \mathbf{X} &\mapsto J(\mathbf{T}_{\mathbf{X}}(\Omega)), \end{aligned} \tag{2.11}$$

and to minimise it over all $\mathbf{X} \in \mathcal{D}$ with $\mathbf{T}_{\mathbf{X}} \in \mathcal{F}(\mathcal{D})$. For this functional we can now consider its Gâteaux and Fréchet derivatives.

Definition 2.5.

Let X and Y be Banach spaces, let $U \subset X$ be an open subset and let $F : U \to Y$.

- *F is said to have a* Gâteaux semiderivative *at x in the direction h if the following limit exists and is finite:*

$$\mathrm{d}F(x; h) := \lim_{s \downarrow 0} \frac{F(x + sh) - F(x)}{s}. \tag{2.12}$$

- *F is said to be* Gâteaux differentiable *at x if it has a Gâteaux semiderivative at x in all directions $h \in X$ and the map*

$$h \mapsto \mathrm{d}F(x;h) : X \to \mathbb{R} \tag{2.13}$$

is linear and continuous.

- *F is said to be* Fréchet differentiable *at x if F is Gâteaux differentiable at x and*

$$\lim_{\|h\|_X \to 0} \frac{\|F(x+h) - F(x) - \mathrm{d}F(x;h)\|_Y}{\|h\|_X} = 0. \tag{2.14}$$

If F is Gâteaux differentiable at x then $\mathrm{d}F(x;\cdot)$ is an element of X^* and we will sometimes write the derivative as the dual pairing

$$\langle \mathrm{d}F(x), h \rangle_{X^*,X} \equiv \mathrm{d}F(x;h). \tag{2.15}$$

Given a shape functional J, we say that J *is shape differentiable* in Ω if J_Ω is Fréchet differentiable at 0,

$$\mathrm{d}J_\Omega(0;\mathbf{V}) = \lim_{s \to 0} \frac{J_\Omega(s\mathbf{V}) - J_\Omega(0)}{s} \tag{2.16}$$

Note that this definition is equivalent to the classical expression

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \lim_{s \to 0} \frac{J((\mathbf{Id} + s\mathbf{V})(\Omega)) - J(\Omega)}{s} \tag{2.17}$$

and in fact we will often write $\mathrm{d}J(\Omega)[\mathbf{V}]$ to refer to the shape derivative.

Example 2.6. *Let $g \in C^1(\mathbb{R}^d;\mathbb{R})$ and define*

$$J(\Omega) = \int_\Omega g \,\mathrm{d}x. \tag{2.18}$$

Then

$$J_\Omega(\mathbf{X}) = \int_{\Omega_\mathbf{X}} g \,\mathrm{d}x = \int_\Omega g \circ \mathbf{T}_\mathbf{X} \det(\mathrm{D}\mathbf{T}_\mathbf{X}) \,\mathrm{d}x \tag{2.19}$$

and hence using $\det(\mathrm{D}\mathbf{T}_\mathbf{X}) = 1 + \mathrm{div}(\mathbf{X}) + o(\|\mathbf{X}\|)$, we obtain that J is shape differentiable with shape derivative given by

$$\mathrm{d}J_\Omega(0;\mathbf{V}) = \int_\Omega \nabla g \cdot \mathbf{V} + g\,\mathrm{div}(\mathbf{V}) \,\mathrm{d}x = \int_\Omega \mathrm{div}(g\mathbf{V}) \,\mathrm{d}x. \tag{2.20}$$

At a domain $\Omega_\mathbf{X}$, there are now two kinds of derivatives: the shape derivative, e.g. the derivative of $J_{\Omega_\mathbf{X}}$ at 0, and the classical Fréchet derivative of J_Ω at $\mathbf{X}$. These two are related via the following Theorem.

Theorem 2.7 ([DZ11, Theorem 9.3.4])**.**
Let J be a shape functional, $\mathcal{D}$ be a normed vector subspace of $C^{0,1}(\overline{\mathbb{R}^d}, \mathbb{R}^d)$, and $\mathbf{X} \in \mathcal{D}$ with $\mathbf{T}_\mathbf{X} \in \mathcal{F}(\mathcal{D})$.

1. *J_Ω has a Gâteaux semiderivative at $\mathbf{X}$ in the direction $\mathbf{V} \in \mathcal{D}$ if and only if $J_{\Omega_\mathbf{X}}$ has a Gâteaux semiderivative at 0 in direction $\mathbf{V} \circ \mathbf{T}_\mathbf{X}^{-1}$ and it holds*

$$\mathrm{d}J_\Omega(\mathbf{X}; \mathbf{V}) = \mathrm{d}J_{\Omega_\mathbf{X}}(0; \mathbf{V} \circ \mathbf{T}_\mathbf{X}^{-1}). \tag{2.21}$$

2. *J_Ω is Gâteaux differentiable at $\mathbf{X}$ if and only if $J_{\Omega_\mathbf{X}}$ is Gâteaux differentiable at 0.*

3. *J_Ω is Fréchet differentiable at $\mathbf{X}$ if and only if $J_{\Omega_\mathbf{X}}$ is Fréchet differentiable at 0.*

Proof. We observe that, for $s > 0$,

$$\mathbf{Id} + \mathbf{X} + s\mathbf{V} = [\mathbf{Id} + s(\mathbf{V} \circ (\mathbf{Id} + \mathbf{X})^{-1})] \circ (\mathbf{Id} + \mathbf{X}) \tag{2.22}$$

and hence

$$\begin{aligned} &\frac{J([\mathbf{Id} + \mathbf{X} + s\mathbf{V}](\Omega)) - J([\mathbf{Id} + \mathbf{X}](\Omega))}{s} \\ =&\frac{J([\mathbf{Id} + s(\mathbf{V} \circ (\mathbf{Id} + \mathbf{X})^{-1})](\Omega_\mathbf{X})) - J(\Omega_\mathbf{X})}{s} \\ =&\frac{J([\mathbf{Id} + s(\mathbf{V} \circ \mathbf{T}_\mathbf{X}^{-1})](\Omega_\mathbf{X})) - J(\Omega_\mathbf{X})}{s}, \end{aligned} \tag{2.23}$$

i.e. the two difference quotients in the definition of the derivatives $\mathrm{d}J_\Omega(\mathbf{X}; \mathbf{V})$ and $\mathrm{d}J_{\Omega_\mathbf{X}}(0; \mathbf{V} \circ \mathbf{T}_\mathbf{X}^{-1})$ are identical. This implies all three statements. □

This property is crucial for the implementation of shape optimisation algorithms and will be used repeatedly in later chapters.

The optimisation problem that we would like to solve is thus

$$\begin{aligned} \underset{\mathbf{X} \in \mathcal{D}}{\text{minimise}} \quad & J_\Omega(\mathbf{X}) \\ \text{subject to} \quad & \mathbf{T}_\mathbf{X} \in \mathcal{F}(\mathcal{D}). \end{aligned} \tag{2.24}$$

However, in practice we do not want to consider deformations of the entire space $\mathbb{R}^d$, but are only interested in the deformation of Ω itself. The natural consequence is then to minimise $J_\Omega(\mathbf{X})$ over a class of functions defined on Ω that are diffeomorphisms onto their image. This leads to a new problem: while for functions that are defined on the entire space, it is essentially enough to ensure that they are locally diffeomorphisms, as Hadamard's global inverse function theorem then guarantees that

they are global diffeomorphisms; this is not the case for functions defined on subsets only. In consequence one often makes the explicit or tacit assumption that the initial shape Ω is *close* in some sense to the optimal shape and only a small perturbation of the initial shape is necessary. However, when this assumption is violated and the optimisation algorithm chooses deformations $\mathbf{T}_\mathbf{X}$ that are not diffeomorphisms this will lead to a collapsed or overlapping domain, see Figure 2.1. We will now develop

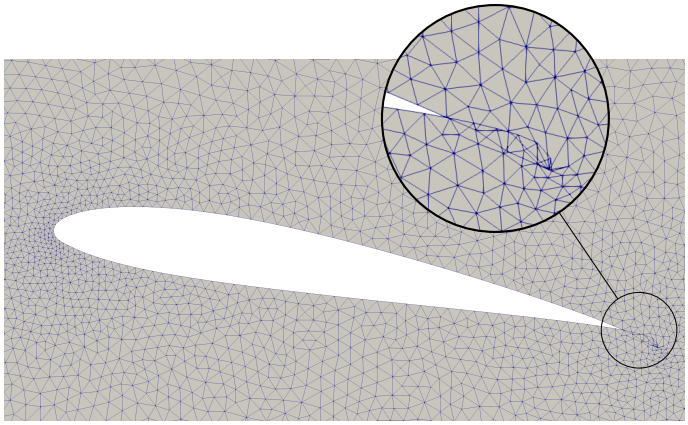


Figure 2.1: The channel around an airfoil is highly nonconvex; this means that deformations can quickly lead to overlapping of the mesh.

a sufficient condition on $\mathbf{X}$ that guarantees that $\mathbf{T}_\mathbf{X}$ is indeed a diffeomorphism; a crucial ingredient for this condition is the bi-Lipschitz property.

Definition 2.8.
Let $\Omega \subset \mathbb{R}^d$ be an open set and let $\mathbf{X} : \Omega \to \mathbb{R}^d$ be a Lipschitz continuous function. We say that $\mathbf{X}$ is bi-Lipschitz if there exists an $\varepsilon > 0$ such that

$$\varepsilon \|x - y\| \leq \|\mathbf{X}(x) - \mathbf{X}(y)\| \leq \mathrm{Lip}(\mathbf{X}) \|x - y\| \qquad \text{for all } x, y \in \Omega. \tag{2.25}$$

We denote the largest ε that satisfies the above inequality by $\underline{\mathrm{Lip}}(\mathbf{X})$.

The lower bound immediately implies that $\mathbf{X}$ is injective, hence bijective onto its image $O = \mathbf{X}(\Omega)$. Now let $x', y' \in O$ with $\mathbf{X}(x) = x'$ and $\mathbf{X}(y) = y'$ for some $x, y \in \Omega$. Then

$$\begin{aligned} \left\|\mathbf{X}^{-1}(x') - \mathbf{X}^{-1}(y')\right\| &= \|x - y\| \\ &\leq \frac{1}{\underline{\mathrm{Lip}}(\mathbf{X})} \|\mathbf{X}(x) - \mathbf{X}(y)\| = \frac{1}{\underline{\mathrm{Lip}}(\mathbf{X})} \|x' - y'\| \end{aligned} \tag{2.26}$$

and similarly

$$\begin{aligned} \left\|\mathbf{X}^{-1}(x') - \mathbf{X}^{-1}(y')\right\| &= \|x - y\| \\ &\geq \frac{1}{\mathrm{Lip}(\mathbf{X})} \|\mathbf{X}(x) - \mathbf{X}(y)\| = \frac{1}{\mathrm{Lip}(\mathbf{X})} \|x' - y'\| . \end{aligned} \tag{2.27}$$

From this we can deduce that

$$\mathrm{Lip}(\mathbf{X}^{-1}) = \frac{1}{\underline{\mathrm{Lip}}(\mathbf{X})} \quad \text{and} \quad \underline{\mathrm{Lip}}(\mathbf{X}^{-1}) = \frac{1}{\mathrm{Lip}(\mathbf{X})}. \tag{2.28}$$

We can show that small enough Lipschitz continuous perturbations of the identity are actually bi-Lipschitz.

Proposition 2.9 ([DZ11, §3, Theorem 2.15])**.**
Let $\Omega \subset \mathbb{R}^d$ be an open set and let $\mathbf{X} : \Omega \to \mathbb{R}^d$ be a Lipschitz continuous function with $\mathrm{Lip}(\mathbf{X}) < 1$. Then $\mathbf{T}_\mathbf{X} = \mathbf{Id} + \mathbf{X}$ is bi-Lipschitz.

Proof. Let $x, y \in \Omega$, then

$$\|\mathbf{T}_\mathbf{X}(x) - \mathbf{T}_\mathbf{X}(y)\| = \|x - y + \mathbf{X}(x) - \mathbf{X}(y)\| \leq (1 + \mathrm{Lip}(\mathbf{X})) \|x - y\|. \tag{2.29}$$

Similarly,

$$\|\mathbf{T}_\mathbf{X}(x) - \mathbf{T}_\mathbf{X}(y)\| \geq \|x - y\| - \|\mathbf{X}(x) - \mathbf{X}(x)\| \geq (1 - \mathrm{Lip}(\mathbf{X})) \|x - y\|. \tag{2.30}$$

□

Due to its global nature it is not clear how one would enforce a condition on the Lipschitz constant in an optimisation algorithm. However, for a certain class of domains, the Lipschitz constant of a function can be estimated by the supremum of the gradient of the function. This gives us a local constraint on $\mathbf{X}$ that we can enforce in the optimisation algorithm.

Definition 2.10.
Let $\Omega \subset \mathbb{R}^d$ be an open set. We say that Ω is quasiconvex if there exists $\zeta(\Omega)$ such that for all $x, y \in \Omega$

$$\inf \left\{ \int_0^1 \|\gamma'(s)\| \, \mathrm{d}s : \gamma \in C^1([0,1]; \Omega), \gamma(0) = x, \gamma(1) = y \right\} \leq \frac{1}{\zeta(\Omega)} \|x - y\|. \tag{2.31}$$

The value $\zeta(\Omega)$ is a measure for how nonconvex a domain is; for convex domains any two points can be connected by a straight line and it holds that $\zeta(\Omega) = 1$. The channel around the airfoil in Figure 2.1 is an example of a quasiconvex domain.

Now let $\mathbf{X}$ be continuously differentiable on Ω and let γ be a path between x, $y \in \Omega$. Then it holds that

$$\|\mathbf{X}(x) - \mathbf{X}(y)\| = \left\| \int_0^1 (\mathbf{X} \circ \gamma)'(s) \, \mathrm{d}s \right\| \leq \int_0^1 |||(\mathrm{D}\mathbf{X})(\gamma(s))|||_2 \, \|\gamma'(s)\|_2 \, \mathrm{d}s. \tag{2.32}$$

Taking the infimum over all paths between x and y yields

$$\mathrm{Lip}(\mathbf{X}) \leq |\mathbf{X}|_{C^1(\overline{\Omega}; \mathbb{R}^d)} \frac{1}{\zeta(\Omega)}, \tag{2.33}$$

and hence we can ensure that $\mathbf{T}_\mathbf{X} = \mathbf{Id} + \mathbf{X}$ is bi-Lipschitz if $|\mathbf{X}|_{C^1(\overline{\Omega}; \mathbb{R}^d)} < \zeta(\Omega)$.

Proposition 2.11.
Let $\Omega \subset \mathbb{R}^d$ be a quasiconvex, open set and let $k \geq 1$. If $\mathbf{X} \in C^k(\overline{\Omega}; \mathbb{R}^d)$ satisfies $|\mathbf{X}|_{C^1(\overline{\Omega};\mathbb{R}^d)} \equiv \sup \|\|\mathrm{D}\mathbf{X}\|\|_2 < \zeta(\Omega)$ then $\mathbf{T}_\mathbf{X}$ is a C^k-diffeomorphism.

Proof. Since $|\mathbf{X}|_{C^1(\overline{\Omega};\mathbb{R}^d)} < \zeta(\Omega) \leq 1$ it follows that $\mathbf{T}_\mathbf{X}$ has nonzero derivative everywhere and hence $\mathbf{T}_\mathbf{X}$ is locally a C^k-diffeomorphism by the inverse function theorem. From the calculation above we deduced that $\mathbf{T}_\mathbf{X}$ is bi-Lipschitz and hence globally bijective. Thus $\mathbf{T}_\mathbf{X}$ is globally a C^k-diffeomorphism. □

On quasiconvex domains the space of Lipschitz functions can be identified with the Sobolev space $W^{1,\infty}$ and the Lipschitz constant can be bounded by the $W^{1,\infty}$ norm.
Theorem 2.12 ([HKT08, Theorem 7])**.**
Let $\Omega \subset \mathbb{R}^d$ be a quasiconvex, open set. Then $C^{0,1}(\overline{\Omega}; \mathbb{R}^m) = W^{1,\infty}(\Omega; \mathbb{R}^m)$ in the sense that every Lipschitz function is in $W^{1,\infty}(\Omega; \mathbb{R}^m)$ and every function in $W^{1,\infty}(\Omega; \mathbb{R}^m)$ has a representative in $C^{0,1}(\overline{\Omega}; \mathbb{R}^m)$. Furthermore $f \in W^{1,\infty}(\Omega; \mathbb{R}^m)$ is differentiable almost everywhere, its strong derivative coincides almost everywhere with its weak derivative and it holds that

$$\mathrm{Lip}(f) \leq \zeta(\Omega) \|\mathrm{D}f\|_\infty . \tag{2.34}$$

Hence we can formulate a version of Proposition 2.11 for Lipschitz continuous functions.
Corollary 2.13.
Let $\Omega \subset \mathbb{R}^d$ be a quasiconvex, open set. If $\mathbf{X} \in W^{1,\infty}(\Omega; \mathbb{R}^d)$ satisfies $\operatorname{ess\,sup} \|\|\mathrm{D}\mathbf{X}\|\|_2 < \zeta(\Omega)$, then $\mathbf{T}_\mathbf{X}$ is a bi-Lipschitz map.

To summarise, given a shape functional J, initial shape Ω and a class of deformations $\mathcal{D}$ that is either $C^{0,1}(\overline{\Omega}; \mathbb{R}^d)$, $W^{1,\infty}(\Omega; \mathbb{R}^d)$ or $C^k(\overline{\Omega}; \mathbb{R}^d)$, the optimisation problem that we consider is

$$\begin{aligned} \underset{\mathbf{X} \in \mathcal{D}}{\text{minimise}} \quad & J_\Omega(\mathbf{X}) \\ \text{subject to} \quad & \|\|\mathrm{D}\mathbf{X}\|\|_2 \leq \zeta(\Omega) - \varepsilon \text{ a.e.} \end{aligned} \tag{2.35}$$

Here we have introduced a fixed $\varepsilon > 0$ in order to minimise over a closed set.

2.2 Discussion of the necessary regularity of deformations

When deciding what regularity to choose for the deformations one has to balance different demands on the space of deformations.

1. The deformations need to satisfy a minimum regularity in order to guarantee existence of a solution to a PDE constraint on the deformed domain.

2. The deformation space should be large enough to contain all deformations of interest to the engineer.

3. The functional under consideration should be Fréchet differentiable in the deformation space.

4. The deformation space should have the properties necessary to define optimisation algorithms and prove their convergence.

We will see that satisfying all four properties is often impossible. In order to address the first and the second point, we need to understand regularity concepts for domains.

Definition 2.14 (Regularity of domains)**.**
Let $\Omega \subset \mathbb{R}^d$ be an open set with $\partial\Omega \neq \emptyset$, let $k \geq 0$, $l \in [0,1]$.

1. *Ω is* locally the epigraph of a $C^{k,l}$ function *if for each point $x \in \partial\Omega$ there exists $r > 0$, and a $C^{k,l}$ function $\gamma : \mathbb{R}^{d-1} \to \mathbb{R}$ such that, upon relabeling and reorienting the coordinate axes if necessary, we have*

$$\Omega \cap B(x,r) = \{x \in B(x,r) : x_n > \gamma(x_1, \ldots, x_{d-1})\}. \tag{2.36}$$

   We also say that Ω is strongly $C^{k,l}$ *and in the case $k = 0$, $l = 1$, we say that Ω is* strongly Lipschitz.

2. *Ω is said to be* weakly $C^{k,l}$ regular*, if for all $x \in \partial\Omega$ there exists an $r > 0$ and a function $\mathbf{G} : B(x,r) \to \mathbb{R}^d$ such that*

   - *$\mathbf{G}$ is injective,*
   - *$\mathbf{G}$ and $\mathbf{G}^{-1}$ (defined on its image) are $C^{k,l}$*
   - *$\Omega \cap B(x,r) = \{y \in B(x,r) : G_d(y) < 0\}$ where G_d is the d-th component of $\mathbf{G}$.*

   Again, in the case $k = 0$, $l = 1$, we say that Ω is weakly Lipschitz regular*.*

Remark 2.15. *One can easily check, that strong Lipschitz regularity implies weak Lipschitz regularity, by defining $\mathbf{G}(x) = (x_1, \ldots, x_{d-1}, \gamma(x_1, \ldots, x_{d-1}) - x_d)$. In fact, for domains that are at least continuously differentiable, strong and weak regularity are equivalent. However, the proof for this uses the implicit function theorem and hence does not extend to the Lipschitz case [Gri11, Theorem 1.2.1.5].*

In many applications, kinks in the domain should explicitly be allowed and the deformation should be able to create new kinks as well as flatten existing ones, hence Lipschitz deformations should be admissible and deformations cannot be required to be C^1. Lipschitz continuity is also the regularity that is necessary for the change of variables formula to hold — a key ingredient in the calculation of shape derivatives, as it relates the initial and the deformed domain.

Theorem 2.16 (Change of variables)**.**
Let $\Omega \subset \mathbb{R}^d$ be an open set, let $\mathbf{T} : \Omega \to \mathbb{R}^d$ be a bi-Lipschitz function. Then, for every integrable function $g : \mathbf{T}(\Omega) \to \mathbb{R}^d$,

$$\int_\Omega g(\mathbf{T}) \, |\det(\mathrm{D}\mathbf{T})| \, \mathrm{d}x = \int_{\mathbf{T}(\Omega)} g \, \mathrm{d}x. \tag{2.37}$$

Proof. See [EG15, Ch. 3.3, Theorem 2]. □

The shape optimisation problems that we want to study will usually contain a PDE constraint. However, for most equations of interest, existence and uniqueness of a solution to the PDE requires the domain to be strongly Lipschitz regular. The following example shows that applying a Lipschitz continuous deformation to a strongly Lipschitz domain, does not imply that the deformed domain is again strongly Lipschitz, but only implies weak Lipschitz regularity [Gri11, p. 8]. This is only guaranteed when the deformation is at least C^1.

Lemma 2.17.
Let $\Omega \subset \mathbb{R}^d$ be strongly Lipschitz and let $\mathbf{T} : \Omega \to \mathbb{R}^d$ be a C^1-diffeomorphism onto its image. Then $\mathbf{T}(\Omega)$ is again strongly Lipschitz.

Proof. See [HMT07, Theorem 4.1]. □

Lemma 2.18.
Let $\Omega \subset \mathbb{R}^d$ be weakly Lipschitz and let $\mathbf{T} : \Omega \to \mathbb{R}^d$ be bi-Lipschitz. Then $\mathbf{T}(\Omega)$ is again weakly Lipschitz.

Proof. Let $\mathbf{G}$ be the map for Ω satisfying the assumptions of Definition 2.14, then $\mathbf{T}^{-1} \circ \mathbf{G}$ also satisfies these assumptions and hence $\mathbf{T}(\mathbf{G})$ is weakly Lipschitz. □

However, for two reasons we can still justify using Lipschitz deformations. First, one can show that small Lipschitz deformations do indeed preserve strong Lipschitz regularity, and secondly, it is possible to extend certain existence and uniqueness results to deformations of strong Lipschitz domains, even if they are potentially just weakly Lipschitz.

Lemma 2.19.
Let $\Omega \subset \mathbb{R}^d$ be strongly Lipschitz. Then there exists $c(\Omega)$ with $0 < c(\Omega) < 1$ such that

$$\forall \mathbf{X} \in C^{0,1}(\mathbb{R}^d;\mathbb{R}^d) \text{ s.t. } \operatorname{Lip}(\mathbf{X}) \leq c(\Omega), \quad (\mathbf{Id}+\mathbf{X})(\Omega) \text{ is strongly Lipschitz.} \tag{2.38}$$

Proof. See [Bel+97, Lemma 3]. □

Existence and uniqueness results usually require strong Lipschitz regularity of the domain. However, for Lipschitz deformations of such domains one can often use a pull-back argument to extend these results. We illustrate this for the Poisson problem.

Proposition 2.20 ([DZ11, § 10.5.5])**.**
Let $\Omega \subset \mathbb{R}^d$ be a strongly Lipschitz domain and let $\mathbf{T} : \Omega \to \mathbb{R}^d$ be a bi-Lipschitz function. Then for any $g \in L^2(T(\Omega))$ there exists a unique weak solution to the Poisson problem with homogeneous boundary conditions, i.e.

$$\exists! u \in H^1_0(\mathbf{T}(\Omega)) \text{ s.t. } \int_{\mathbf{T}(\Omega)} \nabla u \cdot \nabla v \,\mathrm{d}x = \int_{\mathbf{T}(\Omega)} gv \,\mathrm{d}x \quad \text{for all } v \in H^1_0(\mathbf{T}(\Omega)). \tag{2.39}$$

Remark 2.21. *In the same way as [DZ11], we employ two notations for derivatives. We use Euler notation for deformations of domains and vector calculus notation for PDE constraints.*

Proof of Proposition 2.20. Using an approximation argument we obtain that $v \mapsto \tilde{v} := v \circ \mathbf{T}$ is a bijective mapping from $H^1_0(\mathbf{T}(\Omega))$ to $H^1_0(\Omega)$. By the change of variables formula it holds that

$$\begin{aligned}
&\int_{\mathbf{T}(\Omega)} \nabla u \cdot \nabla v \,\mathrm{d}x - \int_{\mathbf{T}(\Omega)} gv \,\mathrm{d}x \\
&= \int_\Omega ((\nabla u) \circ \mathbf{T}) \cdot ((\nabla v) \circ \mathbf{T}) \,|\det(\mathrm{D}\mathbf{T})| - (g \circ \mathbf{T})(v \circ \mathbf{T}) \,|\det \mathrm{D}\mathbf{T}| \,\mathrm{d}x \\
&= \int_\Omega (\mathrm{D}\mathbf{T}^{-\top} \nabla (u \circ \mathbf{T})) \cdot (\mathrm{D}\mathbf{T}^{-\top} \nabla (v \circ \mathbf{T})) \,|\det(\mathrm{D}\mathbf{T})| \\
&\qquad\qquad - (g \circ \mathbf{T})(v \circ \mathbf{T}) \,|\det(\mathrm{D}\mathbf{T})| \,\mathrm{d}x \\
&= \int_\Omega (\nabla \tilde{u})^\top (\mathrm{D}\mathbf{T}^{-1} \,\mathrm{D}\mathbf{T}^{-\top}) \nabla \tilde{v} \,|\det(\mathrm{D}\mathbf{T})| - \tilde{g}\tilde{v} \,|\det(\mathrm{D}\mathbf{T})| \,\mathrm{d}x.
\end{aligned} \tag{2.40}$$

Using Lemma 2.22 we can invoke the Lax-Milgram Theorem to guarantee existence of a unique solution $\tilde{u} \in H^1_0(\Omega)$ to

$$\int_\Omega (\nabla \tilde{u})^\top (\mathrm{D}\mathbf{T}^{-1} \,\mathrm{D}\mathbf{T}^{-\top}) \nabla \tilde{v} \,|\det(\mathrm{D}\mathbf{T})| - \tilde{g}\tilde{v} \,|\det(\mathrm{D}\mathbf{T})| \,\mathrm{d}x = 0 \tag{2.41}$$

for all $\tilde{v} \in H^1_0(\Omega)$. If we then define $u = \tilde{u} \circ \mathbf{T}^{-1}$, we obtain a solution to the problem on $\mathbf{T}(\Omega)$. To see that this solution is unique, assume that there exists a second

solution w. Then $\tilde{w} = w \circ \mathbf{T}$ is a distinct solution to the problem on Ω, which is a contradiction. □

Lemma 2.22.
Let $\Omega \subset \mathbb{R}^d$ be an open set and let $\mathbf{T} : \Omega \to \mathbb{R}^d$ be a bi-Lipschitz continuous function. Then for $\mathbf{a},\ \mathbf{b} \in \mathbb{R}^d$ it holds that

$$\begin{aligned} \left|\mathbf{b}^\top \mathrm{D}\mathbf{T}^{-1}\, \mathrm{D}\mathbf{T}^{-\top}\mathbf{a}\right| &\leq \underline{\mathrm{Lip}}(\mathbf{T})^{-2} \left\|\mathbf{b}\right\| \left\|\mathbf{a}\right\|, \\ \mathbf{a}^\top \mathrm{D}\mathbf{T}^{-1}\, \mathrm{D}\mathbf{T}^{-\top}\mathbf{a} &\geq \mathrm{Lip}(\mathbf{T})^{-2} \left\|\mathbf{a}\right\|^2, \\ \left|\det(\mathrm{D}\mathbf{T})\right| &\geq \underline{\mathrm{Lip}}(\mathbf{T})^{d} \ \textit{a.e.} \end{aligned} \tag{2.42}$$

Proof. Consider the singular value decomposition $\mathrm{D}\mathbf{T} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, with $\sigma_1 \geq \ldots \geq \sigma_d$. Then it holds that $\sigma_1 \leq \mathrm{Lip}(\mathbf{T})$ and $\sigma_d \geq \underline{\mathrm{Lip}}(\mathbf{T})$. Now observe that

$$\mathrm{D}\mathbf{T}^{-1}\, \mathrm{D}\mathbf{T}^{-\top} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^\top\mathbf{U}\boldsymbol{\Sigma}^{-1}\mathbf{V}^\top = \mathbf{V}\boldsymbol{\Sigma}^{-2}\mathbf{V}^\top \tag{2.43}$$

and hence

$$\begin{aligned} \left|\mathbf{b}^\top \mathrm{D}\mathbf{T}^{-1}\, \mathrm{D}\mathbf{T}^{-\top}\mathbf{a}\right| &\leq \sigma_d^{-2} \left\|\mathbf{b}\right\| \left\|\mathbf{a}\right\| \\ \mathbf{a}^\top \mathrm{D}\mathbf{T}^{-1}\, \mathrm{D}\mathbf{T}^{-\top}\mathbf{a} &\geq \sigma_1^{-2} \left\|\mathbf{a}\right\|^2. \end{aligned} \tag{2.44}$$

The last claim follows from the fact that the absolute value of the determinant of a matrix is equal to the product of its singular values. □

In the same fashion, existence and uniqueness of solutions on domains that are Lipschitz deformations of strong Lipschitz domains can be proven for other PDEs of interest. The third requirement, differentiability of the shape functional, is usually satisfied in $W^{1,\infty}$. Unfortunately, neither $W^{1,\infty}(\Omega;\mathbb{R}^d)$ nor the spaces of continuously differentiable functions are reflexive, making it difficult to prove existence of optimal deformations. In fact, existence and uniqueness of optimal shapes is only known for a small number of problems.

2.3 Reduction of PDE constraints via adjoint solutions

Many shape optimisation problems are subject to an additional PDE constraint. This complicates the problem, as the PDE and its solution will depend on the shape.

Let $\Omega \subset \mathbb{R}^d$ be a given initial domain and denote its deformations by $\Omega_{\mathbf{X}} = \mathbf{T}_{\mathbf{X}}(\Omega)$. Assume the PDE constraint on $\Omega_{\mathbf{X}}$ is given by a (possibly nonlinear) function

$$\begin{aligned} e(\Omega_{\mathbf{X}}, \cdot) : \mathcal{X}(\Omega_{\mathbf{X}}) &\to \mathcal{X}(\Omega_{\mathbf{X}})^* \\ u &\mapsto \big(v \mapsto e(\Omega_{\mathbf{X}}, u; v)\big) \end{aligned} \tag{2.45}$$

that is required to be zero, $e(\Omega_{\mathbf{X}}, u) = 0$. Here $\mathcal{X}(\Omega_{\mathbf{X}})$ is some Sobolev space in which we search for a solution, e.g. $\mathcal{X}(\Omega_{\mathbf{X}}) = H_0^1(\Omega_{\mathbf{X}}; \mathbb{R})$. If the mapping

$$u \mapsto u \circ \mathbf{T}_{\mathbf{X}}^{-1} : \mathcal{X}(\Omega) \to \mathcal{X}(\Omega_{\mathbf{X}}) \tag{2.46}$$

is bijective, then we can employ a pull-back in order to express the PDE constraint for all perturbations $\Omega_{\mathbf{X}}$ as a constraint defined on Ω. For ease of notation we will denote the space on the initial domain simply by $\mathcal{X} \equiv \mathcal{X}(\Omega)$. Define

$$e_\Omega : \mathcal{D} \times \mathcal{X} \to \mathcal{X}^* \tag{2.47}$$

$$(\mathbf{X}, u) \mapsto \big(v \mapsto e_\Omega(\mathbf{X}, u; v) := e(\Omega_{\mathbf{X}}, u \circ \mathbf{T}_{\mathbf{X}}^{-1}; v \circ \mathbf{T}_{\mathbf{X}}^{-1})\big), \tag{2.48}$$

where $\mathbf{T}_{\mathbf{X}} = \mathbf{Id} + \mathbf{X}$. We arrive at the formulation

$$\begin{aligned} \underset{\mathbf{X}\in\mathcal{D},\ u\in\mathcal{X}}{\text{minimise}} \quad & J_\Omega(\mathbf{X}, u) \\ \text{subject to} \quad & e_\Omega(\mathbf{X}, u) = 0, \\ & \|\mathrm{D}\mathbf{X}\|_2 \leq \zeta(\Omega) - \varepsilon \text{ a.e.} \end{aligned} \tag{2.49}$$

There are two approaches for solving such constrained problems. The first is to solve for $\mathbf{X}$ and u simultaneously; this is known as the all-at-once approach. Alternatively, if there is a well-defined solution operator $\mathbf{X} \mapsto S(\mathbf{X})$ such that $e_\Omega(\mathbf{X}, S(\mathbf{X})) = 0$, then one can reduce the optimisation problem to

$$\begin{aligned} \underset{\mathbf{X}\in\mathcal{D}}{\text{minimise}} \quad & \mathcal{J}_\Omega(\mathbf{X}) \\ \text{subject to} \quad & \|\mathrm{D}\mathbf{X}\|_2 \leq \zeta(\Omega) - \varepsilon \text{ a.e.}, \end{aligned} \tag{2.50}$$

where $\mathcal{J}_\Omega(\mathbf{X}) = J_\Omega(\mathbf{X}, S(\mathbf{X}))$. To perform efficient optimisation, the derivative of $\mathcal{J}_\Omega$ is needed. Recalling Theorem 2.7 we observe that we only need to know how to calculate the derivative at $\mathbf{X} = 0$. To calculate the derivative at a general $\mathbf{X}$, we can then make use of the identity

$$\mathrm{d}\mathcal{J}_\Omega(\mathbf{X}; \mathbf{V}) = \mathrm{d}\mathcal{J}_{\Omega_{\mathbf{X}}}(0; \mathbf{V} \circ \mathbf{T}_{\mathbf{X}}^{-1}). \tag{2.51}$$

However, in order to calculate the derivative of $\mathcal{J}_\Omega$, the solution operator S needs to be differentiated as well.

Assume that J and e are continuously Fréchet differentiable and furthermore assume that $\frac{\partial e_\Omega}{\partial u}(0, S(0))$ is an invertible map from $\mathcal{X}$ to $\mathcal{X}^*$. Then, by the implicit function theorem, the solution operator S is continuously differentiable and furthermore it holds that

$$\mathrm{d}\mathcal{J}_\Omega(0; \mathbf{V}) = \left\langle \frac{\partial J_\Omega}{\partial \mathbf{X}}(0, S(0)), \mathbf{V} \right\rangle + \left\langle \frac{\partial J_\Omega}{\partial \mathbf{X}}, \mathrm{d}S(0; \mathbf{V}) \right\rangle \quad \text{for all } \mathbf{V} \in \mathcal{D}. \tag{2.52}$$

However, in order to avoid having to differentiate the solution operator S, we now illustrate the adjoint approach to calculating the derivative under PDE constraints [Hin+09]. We begin by defining the Lagrangian

$$\mathcal{L}(\mathbf{X}, u, v) := J_\Omega(\mathbf{X}, u) + e_\Omega(\mathbf{X}, u; v) \tag{2.53}$$

and observe that

$$\mathcal{J}_\Omega(\mathbf{X}) = \mathcal{L}(\mathbf{X}, S(\mathbf{X}), v) \quad \text{for all } v \in \mathcal{X}. \tag{2.54}$$

Then it holds that for any $v \in \mathcal{X}$ and $\mathbf{V} \in \mathcal{D}$

$$\begin{aligned} \mathrm{d}\mathcal{J}_\Omega(0; \mathbf{V}) &\equiv \left\langle \frac{\mathrm{d}\mathcal{J}_\Omega(0)}{\mathrm{d}\mathbf{X}}, \mathbf{V} \right\rangle \\ &= \left\langle \frac{\partial \mathcal{L}}{\partial \mathbf{X}}(0, S(0), v), \mathbf{V} \right\rangle + \left\langle \frac{\partial \mathcal{L}}{\partial u}(0, S(0), v), \langle S'(0), \mathbf{V} \rangle \right\rangle. \end{aligned} \tag{2.55}$$

Hence, if we pick v such that

$$\frac{\partial \mathcal{L}}{\partial u}(0, S(0), v) = 0 \quad \text{in } \mathcal{X}^* \tag{2.56}$$

then we obtain that

$$\mathrm{d}\mathcal{J}_\Omega(0; \mathbf{V}) = \left\langle \frac{\partial \mathcal{L}}{\partial \mathbf{X}}(0, S(0), v), \mathbf{V} \right\rangle \quad \text{for all } \mathbf{V} \in \mathcal{D}. \tag{2.57}$$

Equation (2.56) can equivalently be written as

$$\frac{\partial e_\Omega}{\partial u}(0, S(0); v) - \frac{\partial J_\Omega}{\partial u}(0, S(0)) = 0 \quad \text{in } \mathcal{X}^* \tag{2.58}$$

and is known as the *adjoint equation* and v is the *adjoint* to the *state* $u = S(0)$.

Example 2.23. *To illustrate the above, we consider the example of energy minimisation in Stokes flow. We consider a channel $A \subset \mathbb{R}^d$ containing an obstacle $B \subset\subset A$ and define the domain $\Omega = A \setminus \overline{B}$. The (reduced) objective that we consider is given by*

$$\mathcal{J}(\Omega) = \int_\Omega \frac{1}{2} \|\nabla \mathbf{u}\|^2 \, \mathrm{d}x \tag{2.59}$$

where $\mathbf{u}$ satisfies the Stokes equations. In strong form these are given by: find $(\mathbf{u}, p)$ such that

$$\begin{aligned} -\Delta \mathbf{u} + \nabla p &= 0 \quad && \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \quad && \text{in } \Omega, \\ \mathbf{u} &= 0 \quad && \text{on } \partial B, \\ \mathbf{u} &= \mathbf{u}_\infty \quad && \text{on } \partial A, \end{aligned} \tag{2.60}$$

for some constant far-field velocity $\mathbf{u}_\infty$*. The weak form is given by: find* $\mathbf{u} \in H^1(\Omega;\mathbb{R}^d)$ *and* $p \in L^2(\Omega)$ *such that* $\mathbf{u} = \mathbf{u}_\infty$ *on* ∂A*,* $\mathbf{u} = 0$ *on* ∂B*, and*

$$\begin{aligned}\langle e(\Omega,(\mathbf{u},p)),(\mathbf{v},q)\rangle := \int_\Omega \nabla\mathbf{u} \colon \nabla\mathbf{v}\,\mathrm{d}x \\ - \int_\Omega (\nabla\cdot\mathbf{v})p\,\mathrm{d}x - \int_\Omega (\nabla\cdot\mathbf{u})q\,\mathrm{d}x \stackrel{!}{=} 0\end{aligned} \tag{2.61}$$

for all $\mathbf{v} \in H^1_0(\Omega;\mathbb{R}^d)$*,* $q \in L^2_0(\Omega)$*. We use the subscript* $\mathbf{X}$ *to denote the composition with* $\mathbf{T}_\mathbf{X}^{-1}$*, e.g.* $\mathbf{v}_\mathbf{X} = \mathbf{v}\circ\mathbf{T}_\mathbf{X}^{-1}$*. The Lagrangian is then given by*

$$\begin{aligned}\mathcal{L}_\Omega(\mathbf{X},(\mathbf{u},p);(\mathbf{v},q)) = &\int_{\Omega_\mathbf{X}} \frac{1}{2}\|\nabla\mathbf{u}_\mathbf{X}\|^2\,\mathrm{d}x + \int_{\Omega_\mathbf{X}} \nabla\mathbf{u}_\mathbf{X}\colon\nabla\mathbf{v}_\mathbf{X}\,\mathrm{d}x \\ &- \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\nabla\cdot\mathbf{v}_\mathbf{X})\,\mathrm{d}x - \int_{\Omega_\mathbf{X}} q_\mathbf{X}(\nabla\cdot\mathbf{u}_\mathbf{X})\,\mathrm{d}x \\ = &\int_\Omega \frac{1}{2}\|\nabla\mathbf{u}_\mathbf{X}\|^2\circ\mathbf{T}_\mathbf{X}\det(\mathrm{D}\mathbf{T}_\mathbf{X})\,\mathrm{d}x \\ &+ \int_\Omega (\nabla\mathbf{u}_\mathbf{X})\circ\mathbf{T}_\mathbf{X}\colon(\nabla\mathbf{v}_\mathbf{X}\circ)\mathbf{T}_\mathbf{X}\det(\mathrm{D}\mathbf{T}_\mathbf{X})\,\mathrm{d}x \\ &- \int_\Omega (\nabla\cdot\mathbf{v}_\mathbf{X})\circ\mathbf{T}_\mathbf{X} p\det(\mathrm{D}\mathbf{T}_\mathbf{X})\,\mathrm{d}x \\ &- \int_\Omega (\nabla\cdot\mathbf{u}_\mathbf{X})\circ\mathbf{T}_\mathbf{X} q\det(\mathrm{D}\mathbf{T}_\mathbf{X})\,\mathrm{d}x.\end{aligned} \tag{2.62}$$

Note that

$$(\nabla\mathbf{v}_\mathbf{X})\circ\mathbf{T}_\mathbf{X} = \mathrm{D}\mathbf{X}^{-\top}\nabla\mathbf{v} \tag{2.63}$$

and

$$(\nabla\cdot\mathbf{v}_\mathbf{X})\circ\mathbf{T}_\mathbf{X} = \operatorname{tr}(\mathrm{D}\mathbf{X}^{-\top}\nabla\mathbf{v}). \tag{2.64}$$

Using the identities

$$\begin{aligned}\langle\partial_\mathbf{X}\det(\mathrm{D}\mathbf{T}_\mathbf{X})|_{\mathbf{X}=0},\mathbf{V}\rangle &= \operatorname{div}(\mathbf{V}),\\ \langle\partial_\mathbf{X}\,\mathrm{D}\mathbf{T}_\mathbf{X}^{-1}|_{\mathbf{X}=0},\mathbf{V}\rangle &= -\mathrm{D}\mathbf{V},\end{aligned} \tag{2.65}$$

we obtain

$$\begin{aligned}&\langle\partial_\mathbf{X}\mathcal{L}_\Omega(0,(\mathbf{u},p),(\mathbf{v},q));\mathbf{V}\rangle \\ =&\int_\Omega -\nabla\mathbf{u}\colon(\mathrm{D}\mathbf{V}^\top\nabla\mathbf{u}) + \frac{1}{2}\|\nabla\mathbf{u}\|^2\operatorname{div}(\mathbf{V})\,\mathrm{d}x \\ &+\int_\Omega -[\mathrm{D}\mathbf{V}^\top\nabla\mathbf{u}]\colon\nabla\mathbf{v} - \nabla\mathbf{u}\colon[\mathrm{D}\mathbf{V}^\top\nabla\mathbf{v}] + \nabla\mathbf{u}\colon\nabla\mathbf{v}\operatorname{div}(\mathbf{V})\,\mathrm{d}x \\ &+\int_\Omega \operatorname{tr}(\mathrm{D}\mathbf{V}^\top\nabla\mathbf{v})p - (\nabla\cdot\mathbf{v})p\operatorname{div}(\mathbf{V})\,\mathrm{d}x \\ &+\int_\Omega \operatorname{tr}(\mathrm{D}\mathbf{V}^\top\nabla\mathbf{u})q - (\nabla\cdot\mathbf{u})q\operatorname{div}(\mathbf{V})\,\mathrm{d}x.\end{aligned} \tag{2.66}$$

The adjoint problem is obtained by taking the derivative of the Lagrangian with respect to the state: find $(\mathbf{v}, q)$ *such that*

$$\begin{aligned} 0 &\overset{!}{=} \langle \partial_{(\mathbf{u},p)} \mathcal{L}_\Omega(0, (\mathbf{u}, p); (\mathbf{v}, q)), (\hat{\mathbf{u}}, \hat{p}) \rangle \\ &= \int_\Omega \nabla \hat{\mathbf{u}} \colon \nabla \mathbf{u} \, \mathrm{d}x + \int_\Omega \nabla \hat{\mathbf{u}} \colon \nabla \mathbf{v} \, \mathrm{d}x - \int_\Omega (\nabla \cdot \mathbf{v}) \hat{p} \, \mathrm{d}x - \int_\Omega (\nabla \cdot \hat{\mathbf{w}}) q \, \mathrm{d}x \end{aligned} \tag{2.67}$$

for all $\hat{\mathbf{u}} \in H_0^1(\Omega; \mathbb{R}^d)$, $\hat{p} \in L_0^2(\Omega)$. *For this particular problem, we can see that the solution to the adjoint equation is given by* $(\mathbf{v}, q) = (0, p)$. *Thus by* (2.57) *we conclude that the shape derivative is given by*

$$\begin{aligned} \mathrm{d}\mathcal{J}_\Omega(0; \mathbf{V}) &= \langle \partial_{\mathbf{X}} \mathcal{L}_\Omega(0, (\mathbf{u}, p); (0, p)), \mathbf{V} \rangle \\ &= \int_\Omega -\nabla \mathbf{u} \colon (\mathrm{D}\mathbf{V}^\top \nabla \mathbf{u}) + \frac{1}{2} \|\nabla \mathbf{u}\|^2 \operatorname{div}(\mathbf{V}) \, \mathrm{d}x \\ &\quad + \int_\Omega \operatorname{tr}(\mathrm{D}\mathbf{V}^\top \nabla \mathbf{u}) p - (\nabla \cdot \mathbf{u}) p \operatorname{div}(\mathbf{V}) \, \mathrm{d}x. \end{aligned} \tag{2.68}$$

Differentiability for the Navier–Stokes case was proven in [Bel+97]. We remark that having a closed form expression for the adjoint equation is not typical.

2.4 Structure of Shape Derivatives

We have formulated shape optimisation problems as a search for optimal deformations of an initial shape. However, since there are many mappings that map an initial domain to the same deformed domain, we cannot hope for uniqueness of optimal deformations. Furthermore, this nonuniqueness has a significant impact on the structure of shape derivatives. Before stating the general structure theorem, we illustrate this using a simple example.

Example 2.24. *In Example 2.6 we studied the shape objective*

$$J(\Omega) = \int_\Omega g \, \mathrm{d}x \tag{2.69}$$

for a function $g \in C^1(\mathbb{R}^d, \mathbb{R})$ *and showed that*

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \int_\Omega \operatorname{div}(g\mathbf{V}) \, \mathrm{d}x. \tag{2.70}$$

Assuming Lipschitz regularity of $\partial\Omega$ *we can apply integration by parts and we obtain that*

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \int_{\partial\Omega} g\mathbf{V} \cdot \mathbf{n} \, \mathrm{d}x. \tag{2.71}$$

This means that, to first order, the shape objective is invariant with respect to deformations in the interior of Ω *or tangential deformation of* $\partial\Omega$.

The expression in (2.70) for the derivative involving the integral over Ω is known as the *volume form*, and the expression in (2.71) involving the integral over the boundary is known as the *surface form of the derivative*. Indeed, for a domain with sufficiently smooth boundary $\partial\Omega$, $\mathrm{d}J_\Omega$ can always be expressed as a distribution on the boundary acting only on normal vector fields. A similar statement holds for the second derivative.

Theorem 2.25 (Structure Theorem)**.**
Assume that J is differentiable at Ω.

1. *Assume Ω has C^1 regularity, then there exists a continuous linear functional $\tilde{l}_1 : C^1(\partial\Omega;\mathbb{R}^d) \to \mathbb{R}$ such that*

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \tilde{l}_1(\mathbf{V}) \quad \text{for all } \mathbf{V} \in C^1(\overline{\Omega};\mathbb{R}^d). \tag{2.72}$$

2. *Assume Ω has C^2 regularity, then there exists a continuous linear functional $l_1 : C^1(\partial\Omega) \to \mathbb{R}$ such that*

$$\mathrm{d}J(\Omega)[\mathbf{V}] = l_1(\mathbf{V}\cdot\mathbf{n}) \quad \text{for all } \mathbf{V} \in C^1(\overline{\Omega};\mathbb{R}^d). \tag{2.73}$$

3. *Assume Ω has C^3 regularity and J is twice differentiable, then there exists a continuous, bilinear $l_2 : C^1(\partial\Omega)\times C^1(\partial\Omega) \to \mathbb{R}$ such that*

$$\begin{aligned}\mathrm{d}^2J(\Omega)[\mathbf{V},\mathbf{W}] = {} & l_2(\mathbf{V}\cdot\mathbf{n},\mathbf{W}\cdot\mathbf{n}) \\ & - l_1(\mathbf{V}^\tau\cdot\mathrm{D}^\tau\mathbf{n}\mathbf{W} + \mathbf{n}\cdot\mathrm{D}^\tau\mathbf{V}\mathbf{W} + \mathbf{n}\cdot\mathrm{D}^\tau\mathbf{W}\mathbf{V})\end{aligned} \tag{2.74}$$

   for all $\mathbf{V},\mathbf{W} \in C^2(\overline{\Omega};\mathbb{R}^d)$.

Proof. See [DZ11, §9, Theorem 3.6 & Corollary 1], [NP02, Theorem 2.1], [Stu16, Theorem 2.6 & 2.10]. □

The structure theorem implies that for sufficiently regular domains and deformations,

- the first shape derivative only depends on deformations on the boundary and in normal direction,
- the second shape derivative only depends on deformations on the boundary (in normal or tangential direction),
- at a stationary point, the second shape derivative only depends on deformations on the boundary and in normal direction.

The structure theorem motivates the formulation of optimisation problems as a search over normal deformations of the boundary and to use the surface form of the derivative. This has the advantage that the derivative then does not have the large kernel consisting of deformations that have zero normal component on the boundary, but has several disadvantages. First, finite element approximations and polyhedral meshes usually do not satisfy the requirements for the structure theorem and hence simply discretising the surface form of the shape derivative obtained for the continuous problem introduces an error. It was shown in [HPS15] that the volume formulation of the derivative admits higher accuracy when computed numerically, in fact it usually corresponds to the exact derivative of the discrete problem. Secondly, if the boundary of a domain is C^{k+1} regular, then the normal is only C^k regular and hence with each update in normal direction one loses regularity of the domain. For these reasons, using the volume form of the shape derivative has been more popular in recent works on shape optimisation. However, the invariance of the shape objective with respect to interior and (to first order) tangential deformations introduces other problems. Close to the minimiser those components of the derivative that arise from the discretisation can dominate; often this is reflected in deformations that only move interior nodes and do not change the domain but have an effect on the value of the discretised objective. When these spurious modes dominate, this can lead to degeneration of the mesh. It also poses problems for the development of second order methods: since the Hessian has a large nullspace a standard Newton method cannot be applied. One way of dealing with this issue is to add a regularisation term to the objective or to the Hessian [KU14; Sch18]. Alternatively, one can introduce a smaller control space $\mathcal{E}$ (e.g. one that lives on the boundary), an extension operator $E : \mathcal{E} \to \mathcal{D}$ and then minimise the objective $\tilde{J}_\Omega(e) = J_\Omega(E(e))$. This still allows for the use of the volume formula of the shape derivative and if $\mathcal{E}$ is chosen appropriately then the first and second derivatives of $\tilde{J}_\Omega$ do not have the same nullspace [Bra+09; Bra+12; KU14].

A new approach was recently suggested in [Etl+18], where the gradient is projected onto a subspace $\tilde{\mathcal{D}} \subset \mathcal{D}$. The space $\tilde{\mathcal{D}}$ is chosen so that the projection is given by the identity for smooth shapes, but removes spurious deformations in the interior and in tangential direction on the boundary for the discretised problem.

The contributions of the next two chapters, namely the automated calculation of shape derivatives and an inner product that creates nearly conformal deformations, are orthogonal to this issue and hence we do not choose a particular form of regularisation but perform the optimisation directly on the deformation space $\mathcal{D}$. Finally, for the

problems we study in Chapter 8 we will combine Tikhonov regularisation with a projection approach similar to [Etl+18], though for a different choice of subspace.

2.5 Numerical shape optimisation

2.5.1 Optimisation

In the previous section we formulated shape optimisation as an optimisation problem looking for an optimal deformation $\mathbf{X} \in \mathcal{D}$. In order to perform gradient based optimisation, we assume that the shape derivative satisfies the higher regularity $\mathrm{d}J \in \mathcal{H}^*$ for some Hilbert space $(\mathcal{H}, (\cdot,\cdot)_{\mathcal{H}})$. Given a shape derivative $\mathrm{d}J_\Omega(\mathbf{X};\cdot) \in \mathcal{H}^*$, the shape gradient $\nabla J_\Omega(\mathbf{X}) \in \mathcal{H}$ is then given by the solution to

$$\text{find } \mathbf{U} \in \mathcal{H} \text{ s.t. } (\mathbf{U},\mathbf{V})_{\mathcal{H}} = \mathrm{d}J_\Omega(\mathbf{X};\mathbf{V}) \text{ for all } \mathbf{V} \in \mathcal{H}. \tag{2.75}$$

We emphasise the distinction between the shape derivative as an element in the dual space and the shape gradient as an element in the primal space and that the gradient is dependent on the inner product defined on the space of deformations. Later in Chapter 4 we will use this dependence to define an inner product that results in gradients that have particularly good properties. Note that obtaining the gradient from the derivative involves the solution of a PDE; this is in contrast to classical optimisation in $\mathbb{R}^n$ equipped with the l^2 inner product, where the gradient is simply the transpose of the derivative.

In Theorem 2.7 it was shown that

$$\mathrm{d}J_\Omega(\mathbf{X};\mathbf{V}) = \mathrm{d}J_{\Omega_\mathbf{X}}(0;\mathbf{V}\circ\mathbf{T}_\mathbf{X}^{-1}) = \mathrm{d}J(\Omega_\mathbf{X})[\mathbf{V}\circ\mathbf{T}_\mathbf{X}^{-1}], \tag{2.76}$$

relating the derivative from the reference domain $\mathrm{d}J_\Omega(\mathbf{X};\cdot)$ to the classical shape derivative $\mathrm{d}J(\Omega_\mathbf{X})[\cdot]$. This is crucial because it means that the derivative as an element of the dual space on the *reference domain* Ω can be calculated on the *deformed domain* $\Omega_\mathbf{X}$.

A simple steepest descent algorithm then reads as follows:

Algorithm 1 Steepest descent shape optimisation algorithm

$\mathbf{X} \leftarrow 0$
$k \leftarrow 0$
while not converged **do**
    Set $\Omega_{\mathbf{X}} = \{x + \mathbf{X}(x) : x \in \Omega\}$.
    Calculate the shape derivative $\mathrm{d}J(\Omega_{\mathbf{X}}) \in [\mathcal{H}(\Omega_{\mathbf{X}})]^*$.
    Calculate the shape gradient $\nabla J(\mathbf{X}) \in [\mathcal{H}(\Omega)]$ satisfying

$$(\nabla J(\mathbf{X}), \mathbf{V})_{\mathcal{H}} = \mathrm{d}J(\Omega_{\mathbf{X}})[\mathbf{V} \circ \mathbf{T}_{\mathbf{X}}^{-1}] \text{ for all } \mathbf{V} \in \mathcal{H}. \tag{2.77}$$

    Set $\mathbf{X} = \mathbf{X} - \alpha_k \nabla J(\mathbf{X})$ for some step size $\alpha_k > 0$.
    $k \leftarrow k + 1$
end while

Here the step size α_k is usually chosen to satisfy the (strong) Wolfe conditions, ensuring that both the objective and the gradient are reduced sufficiently [NW06, §3.1].

First order methods can perform poorly for optimisation problems with very anisotropic Hessians. To improve performance, quasi Newton-methods approximate the Hessian of the objective by inspecting the change of the gradient throughout the optimisation. This adds little extra computational cost but can improve performance significantly [NW06, §7.2, Theorem 3.7]. We will use the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) [Noc80], [NW06, §7.2].

2.5.2 Discretisation

We use the finite element method to solve the governing PDEs and hence we assume that the initial domain Ω is given as a triangulation $\mathcal{M}_h$ of Ω into closed triangles/tetrahedra that satisfy $\overline{\Omega} = \cup_{K \in \mathcal{M}_h} K$ and $K_1^\circ \cap K_2^\circ = \emptyset$ for all $K_1 \neq K_2$ in $\mathcal{M}_h$. Such a triangulation can be used to define a set of global basis function $b_i : \Omega \to \mathbb{R}^d$, $1 \leq i \leq n$, spanning the finite element space. The finite element space $b_{\Omega,i}$ is called *Lagrangian of order p*, if

1. the functions $b_{\Omega,i}$ are continuous in $\overline{\Omega}$,
2. $b_{\Omega,i}|_K$ is a polynomial of order p for all $K \in \mathcal{M}_h$, and
3. the degrees of freedom are given by point evaluations.

The global basis functions are related to *local basis functions* in the following way: there exists a *reference element* $\hat{K}$, local basis functions $\hat{b}_j$ and bijective, affine functions

$\left\{\mathbf{G}_K : K \to \hat{K} \mid K \in \mathcal{M}_h(\Omega)\right\}$ such that

$$b_{\Omega,i}|_K = \hat{b}_j \circ \mathbf{G}_K \quad \text{for some } j \text{ for all } i \text{ and } K \in \mathcal{M}_h(\Omega). \tag{2.78}$$

These scalar functions induce a natural basis $\{\mathbf{b}_{\Omega,i}\}_{i \leq d \cdot M}$ for the space of vector fields.

Often one considers piecewise affine deformations of the initial domain Ω, i.e. $\mathbf{T}_\mathbf{X} = \mathbf{Id} + \mathbf{X}$ where $\mathbf{X} = \sum_i \alpha_i \mathbf{b}_{\Omega,i}$. The basis functions on the deformed domain $\Omega_\mathbf{X} = \mathbf{T}_\mathbf{X}(\Omega)$ are then given by

$$\tilde{b}_{\Omega_\mathbf{X},i}|_{K_\mathbf{X}} = \hat{b}_i \circ \mathbf{G}_{K_\mathbf{X}}, \tag{2.79}$$

where $\mathbf{G}_{K_\mathbf{X}} = \mathbf{G}_K \circ \mathbf{T}_\mathbf{X}|_K{}^{-1}$ and $K_\mathbf{X} = \mathbf{T}_\mathbf{X}(K)$. Note that $\mathbf{G}_{K_\mathbf{X}}$ is again an affine transformation and that the relationship between the basis functions on the deformed and on the initial domain is given by $b_{\Omega_\mathbf{X},i} = b_{\Omega,i} \circ T_\mathbf{X}^{-1}$. It turns out that this relation is crucial for the calculation of the shape derivative. Revisiting (2.75) in the discretised case, we need to find $\mathbf{U} = \sum_{j=0}^{d \cdot M} u_j \mathbf{b}_{\Omega,j}$ such that

$$\sum_j \alpha_j \underbrace{(\mathbf{b}_{j,\Omega}, \mathbf{b}_{i,\Omega})_\mathcal{H}}_{=A_{ij}} = \mathrm{d}J_\Omega(\mathbf{X}; \mathbf{b}_{i,\Omega}) \quad \text{for all } i. \tag{2.80}$$

The matrix $A = \{A_{ij}\}_{ij}$ is assembled once on the reference domain; since it is used repeatedly we will usually factorise it as the cost of this factorisation is amortized throughout the optimisation procedure. The right-hand side can be rewritten as

$$\mathrm{d}J_\Omega(\mathbf{X}; \mathbf{b}_{i,\Omega}) \overset{(2.76)}{=} \mathrm{d}J_{\Omega_\mathbf{X}}(0; \mathbf{b}_{i,\Omega} \circ \mathbf{T}_\mathbf{X}^{-1}) = \mathrm{d}J_{\Omega_\mathbf{X}}(0; \mathbf{b}_{i,\Omega_\mathbf{X}}) \quad \text{for all } i. \tag{2.81}$$

This means that the initially awkward looking relationship between the shape derivative on the reference domain Ω and the derivative on $\Omega_\mathbf{X}$ exactly translates to the relationship between finite element basis functions on the initial and the deformed domain. In particular, this means that the right-hand side in (2.80) can be assembled on the deformed domain $\Omega_\mathbf{X}$.

2.5.3 Software

The numerical experiments in this thesis are all performed using the Firedrake library [Rat+16]. Firedrake is a library for the automated solution of PDEs using the finite element method. Problems are expressed in the Unified Form Language (UFL) [Aln+14] which was developed as part of the FEniCS project [LMW12]. We show an example solving the Poisson problem on the unit square in Listing 1. The symbolic manner in which PDEs are specified in UFL (see lines 10 and 12) enables

many automated calculations, such as the derivation of Jacobians, adjoints and shape derivatives.

Listing 1 Sample code using Firedrake and UFL to solve the Poisson problem.

```
from firedrake import *

# Load a mesh and create a CG1 function space
mesh = UnitSquareMesh(10, 10)
V = FunctionSpace(mesh, "Lagrange", 1)

# Define the trial and the test function
u = Function(V)
v = TestFunction(V)

# Define a right hand side
x, y = SpatialCoordinate(mesh)
rhs = (1 + 8*pi*pi)*cos(2*pi*x)*cos(2*pi*y)

# Define the PDE in weak form and specify boundary conditions
F = inner(grad(u), grad(v)) * dx - rhs * v * dx
bc = DirichletBC(V, 0, "on_boundary")

# Specify the Krylov method (Conjugate Gradient)
# and the preconditioner (AMG) to be used
sp = {'ksp_type': 'cg', 'pc_type': 'hypre'}
solve(F==0, u, bcs=bc, solver_parameters=sp)

# Output the solution in VTK format
File("u.pvd").write(u)
```

For efficient assembly, Firedrake generates and compiles C code kernels at runtime. The arising linear systems are solved using the PETSc library [Bal+97; Bal+16; Bal+18].

To facilitate the efficient formulation of shape optimisation problems in Firedrake, a lightweight library called Fireshape was developed with Alberto Paganini. The library provides efficient ways of specifying deformation spaces to be used. This ranges from the common choice of piecewise affine finite elements (corresponding to simply moving the vertices of the mesh) to high-order spline spaces that can enforce arbitrary regularity on the deformations. The spline approach is described in detail in [PWF18] but not included in this thesis as it is mostly Paganini's work.

UFL already provides methods to automatically calculate derivatives of expressions with respect to functions, which has been used in the dolfin-adjoint project to automatically derive adjoints for linear, nonlinear, steady or transient PDEs [Far+13; MFD19]. In Chapter 3 we will extend UFL to calculate shape derivatives.

While there exists a vast number of software libraries that implement gradient based optimisation algorithms, many assume that the control space is given by $\mathbb{R}^n$ and that the inner product is the l^2 inner product. The Rapid Optimization Library (ROL) of the Trilinos software package is a C++ implementation of a large class of first and second order optimisation algorithms for constrained and unconstrained optimisation. ROL was developed with PDE constrained optimisation in mind [RK14] and allows for the user to provide custom data structures, inner products and Riesz-maps to facilitate optimisation in function spaces. A Python wrapper for ROL written in collaboration with Chris Richardson is available at [WR].

Combining the capabilities of UFL, Firedrake, dolfin-adjoint, and ROL, we are able to perform shape optimisation with Fireshape by simply writing a solver for the constraint equation and an objective to be minimised: the required adjoints and shape derivatives are then all calculated automatically and ROL is called to perform the optimisation.

Chapter 3

Automated calculation of shape derivatives

While only relying on basic transformation rules, the calculation of shape derivatives is often a lengthy and error-prone exercise. The reason for this is that even simple, linear PDEs are nonlinear with respect to the domain. We illustrate this by considering the partial derivative of the weak form of the Laplace equation, that is

$$e(\Omega, u; v) := \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x. \tag{3.1}$$

We can rewrite this equation as described in Section 2.3,

$$e(\mathbf{X}, u; v) = \int_{\Omega_\mathbf{X}} \nabla u_\mathbf{X} \cdot \nabla v_\mathbf{X} \,\mathrm{d}x, \tag{3.2}$$

where $u_\mathbf{X} = u \circ \mathbf{T}_\mathbf{X}^{-1}$ and $v_\mathbf{X} = v \circ \mathbf{T}_\mathbf{X}^{-1}$. Performing a pullback to Ω, we obtain

$$\begin{aligned} e(\mathbf{X}, u; v) &= \int_\Omega (\nabla u_\mathbf{X}) \circ \mathbf{T}_\mathbf{X} \cdot (\nabla v_\mathbf{X}) \circ \mathbf{T}_\mathbf{X} \det(\mathrm{D}\mathbf{T}_\mathbf{X}) \,\mathrm{d}x \\ &= \int_\Omega (\mathrm{D}\mathbf{T}_\mathbf{X})^{-\top} \nabla u \cdot (\mathrm{D}\mathbf{T}_\mathbf{X}{}^{-\top} \nabla v) \det(\mathrm{D}\mathbf{T}_\mathbf{X}) \,\mathrm{d}x. \end{aligned} \tag{3.3}$$

Differentiating with respect to $\mathbf{X}$, we obtain

$$\left\langle \frac{\partial e(0, u; v)}{\partial \mathbf{X}}, \mathbf{V} \right\rangle = -\int_\Omega (\mathrm{D}\mathbf{V})^\top \nabla u \cdot \nabla v + \nabla u \cdot (\mathrm{D}\mathbf{V})^\top \nabla v - \nabla u \cdot \nabla v \,\mathrm{div}(\mathbf{V}) \,\mathrm{d}x. \tag{3.4}$$

For the second derivative, the number of terms increases to 12. One can easily imagine how lengthy such calculations become once more complicated PDEs such as the Navier–Stokes equations are considered.

To simplify this process, Schmidt [Sch18] introduces the open-source library FEMorph: an automatic shape differentiation toolbox for the Unified Form Language.

FEMorph is based on refactoring UFL expressions and applying shape calculus differentiation rules recursively. It can compute first- and second-order shape derivatives (both in so-called weak and strong form), and it has been successfully employed to solve shape optimisation problems [SSW18]. The disadvantage of this approach is that it requires the manual implementation of individual shape calculus rules as well as intricate restructuring and pattern matching on UFL expressions. In addition, the tool assumes that an affinely mapped finite element space is used.

This chapter presents an alternative approach to automated shape differentiation. The key idea is to rely solely on pullbacks and standard Gâteaux derivatives. The approach is generic and robust, as it circumvents the need to implement individual shape calculus rules. We only require a minor addition to UFL, as UFL already supports Gâteaux derivatives with respect to functions and is aware of the different types of pullbacks required for a large number of finite element spaces. As a result of the work presented, UFL is now capable of automatically shape differentiating almost any surface or volume integral that can be expressed in it.

3.1 Shape differentiation on the reference element

We recall the definition of the shape derivative from (2.17): the shape derivative of J at Ω in the direction $\mathbf{V} \in W^{1,\infty}(\Omega, \mathbb{R}^d)$ is given by

$$\mathrm{d}J(\Omega)[\mathbf{V}] \coloneqq \lim_{s \to 0} \frac{J((\mathbf{Id} + s\mathbf{V})(\Omega)) - J(\Omega)}{s}. \tag{3.5}$$

To simplify notation, for the rest of this chapter we fix a deformation direction $\mathbf{V}$ and denote

$$\mathbf{T}_s = \mathbf{Id} + s\mathbf{V} \quad \text{and} \quad \Omega_s = \mathbf{T}_s(\Omega). \tag{3.6}$$

To illustrate the shape differentiation of a shape functional, we consider the prototypical example

$$J(\Omega_s) = \int_{\Omega_s} u_s \,\mathrm{d}x. \tag{3.7}$$

where u_s is some scalar function. The subscript s highlights the possible dependence of u_s on the domain Ω_s; we write $u = u_0$.

The standard procedure to compute $\mathrm{d}J$ is to transform the integral and rewrite

$$J(\Omega_s) = \int_{\Omega} (u_s \circ \mathbf{T}_s) \det(\mathrm{D}\mathbf{T}_s) \,\mathrm{d}x. \tag{3.8}$$

Note that $\det(\mathrm{D}\mathbf{T}_s) > 0$ for s sufficiently small. Then, by linearity of the integral, the shape derivative $\mathrm{d}J$ is given by

$$\begin{aligned}\mathrm{d}J(\Omega)[\mathbf{V}] &= \int_\Omega \mathrm{d_s}((u_s \circ \mathbf{T}_s) \det \mathrm{D}\mathbf{T}_s)\,\mathrm{d}x \\ &= \int_\Omega \mathrm{d_s}(u_s \circ \mathbf{T}_s) + u\,\mathrm{div}(\mathbf{V})\,\mathrm{d}x. \end{aligned} \tag{3.9}$$

where $\mathrm{d_s}(\,\cdot\,)$ denotes the derivative with respect to s at $s = 0$. The term $\mathrm{d_s}(u_s \circ \mathbf{T}_s)$ is often called the *material derivative* [Ber10]. Its explicit formula depends on whether the function u_s does or does not dependent on the domain Ω_s (see Section 3.2).

We now focus on the case of a single triangle K given as the pushforward of a reference element $\hat{K}$, i.e. $K = \mathbf{F}(\hat{K})$. We observe that if K is deformed by a vector field $\mathbf{T}_s$, then the deformed element satisfies

$$\mathbf{T}_s(K) = (\mathbf{F} + s\mathbf{F} \circ \mathbf{V})(\hat{K}). \tag{3.10}$$

As mentioned above, the integrand of a shape objective may change as the domain is deformed. Viewing the objective from the reference element, we denote this dependence with the subscript $u_\mathbf{F}$ and write our shape objective as

$$J(K) \equiv J(\mathbf{F}(\hat{K})) = \int_{\mathbf{F}(\hat{K})} u_\mathbf{F}\,\mathrm{d}x. \tag{3.11}$$

To calculate the shape derivative in the direction $\mathbf{V}$, we note that $(\mathbf{Id} + s\mathbf{V}) \circ \mathbf{F} = \mathbf{F} + s\mathbf{V} \circ \mathbf{F}$ and hence

$$\begin{aligned} J((\mathbf{Id} + s\mathbf{V})(K)) &= \int_{(\mathbf{F}+s\mathbf{V}\circ\mathbf{F})(\hat{K})} u_{\mathbf{F}+s\mathbf{V}\circ\mathbf{F}}\,\mathrm{d}x \\ &= \int_{\hat{K}} u_{\mathbf{F}+s\mathbf{V}\circ\mathbf{F}} \circ (\mathbf{F} + s\mathbf{V} \circ \mathbf{F}) \det(\mathrm{D}(\mathbf{F} + s\mathbf{V} \circ \mathbf{F}))\,\mathrm{d}\hat{x}, \end{aligned} \tag{3.12}$$

where we use $\mathrm{d}\hat{x}$ to refer to an integral over the reference element. The shape derivative is given by taking the derivative at $s = 0$, and hence we observe that

$$\mathrm{d}J(K)[\mathbf{V}] = \int_{\hat{K}} \langle \mathrm{d_F}[u_\mathbf{F} \circ \mathbf{F} \det(\mathrm{D}\mathbf{F})], \mathbf{V} \circ \mathbf{F}\rangle\,\mathrm{d}\hat{x}. \tag{3.13}$$

Usually the direction $\mathbf{V}$ is a Lagrange finite element function and we denote $\hat{\mathbf{V}} = \mathbf{V} \circ \mathbf{F}$. We thus obtain

$$\mathrm{d}J(K)[\mathbf{V}] = \int_{\hat{K}} \big(\langle \mathrm{d_F}[u_\mathbf{F} \circ \mathbf{F}], \hat{\mathbf{V}}\rangle + u_\mathbf{F} \circ \mathbf{F}\,\mathrm{tr}(\mathrm{D}\hat{\mathbf{V}}\,\mathrm{D}\mathbf{F}^{-1})\big) \det(\mathrm{D}\mathbf{F})\,\mathrm{d}\hat{x}. \tag{3.14}$$

Now let Ω be a mesh given by the union of elements $\{K_i\}_i$. Adding subscripts to the deformation, the integrand and the mapping from the reference element to denote the restriction to the element K_i, we then obtain

$$\begin{aligned} &\mathrm{d}J(\Omega)[\mathbf{V}] \\ &= \sum_i \int_{\hat{K}} \left(\langle \mathrm{d}_{\mathbf{F}_i}[u_{i,\mathbf{F}_i} \circ \mathbf{F}_i], \hat{\mathbf{V}}_i \rangle + u_{i,\mathbf{F}_i} \circ \mathbf{F}_i \operatorname{tr}(\mathrm{D}\hat{\mathbf{V}}_i \, \mathrm{D}\mathbf{F}_i^{-1}) \right) \det(\mathrm{D}\mathbf{F}_i) \, \mathrm{d}\hat{x}. \end{aligned} \tag{3.15}$$

Equation (3.15) gives an alternative expression for the shape derivative (3.9): this new expression is assembled on the reference element instead of on the physical domain. More importantly, the derivation above also implies that it is possible to evaluate $\mathrm{d}J(\Omega)[\mathbf{V}]$ by performing the steps necessary for the assembly of $J(\Omega)$ and, at the appropriate time, inserting a standard Gâteaux directional derivative with respect to the pushforward. Simply adding this step to the existing pipeline in UFL enables completely automated shape differentiation (see Figure 3.1). Note that this also enables the computation of higher-order shape derivatives by simply taking higher-order Gâteaux derivatives in (3.15).

Remark 3.1. *The integral in* (3.15) *is approximated by quadrature. For polynomial expressions, UFL automatically chooses a degree so that the integration is exact. When this is not possible, an estimate for a suitable degree is made. UFL also allows the user to specify the quadrature degree. In that case it is crucial that the same quadrature degree is specified for the assembly of the objective and the shape derivative.*

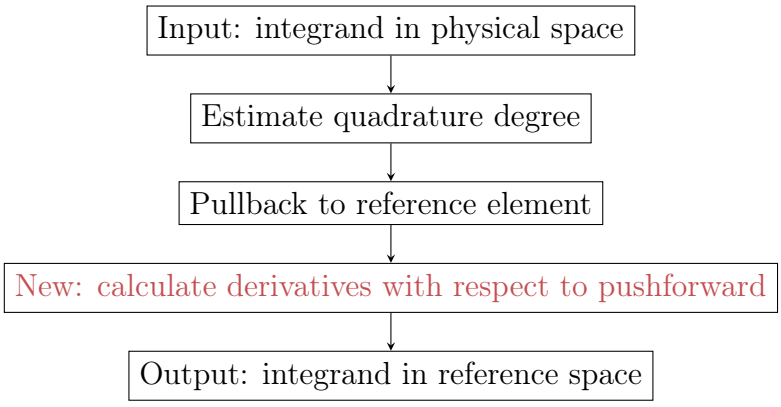


Figure 3.1: Symbolic workflow in UFL to transform integrals from physical to reference space.

3.2 Examples

We now study several examples that cover most standard applications. We begin with the simple case where the integrand is a globally defined function without dependence on the domain.

Example 3.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ and consider*

$$J(\Omega) = \int_\Omega f \,\mathrm{d}x. \tag{3.16}$$

In the notation of (3.9) *this corresponds to $u_s = f$ and we obtain the shape derivative in physical space:*

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \int_\Omega \mathrm{d}_\mathrm{s}(f \circ \mathbf{T}_s) + f \operatorname{div}(\mathbf{V}) \,\mathrm{d}x = \int_\Omega \nabla f \cdot \mathbf{V} + f \operatorname{div}(\mathbf{V}) \,\mathrm{d}x. \tag{3.17}$$

Similarly, in the notation of (3.15) *we have $u_{i,\mathbf{F}_i} = f$ and we obtain the equivalent expression on the reference element:*

$$\mathrm{d}J(K)[\mathbf{V}] = \sum_i \int_{\hat{K}} \left(\hat{\mathbf{V}}_i \cdot (\nabla f \circ \mathbf{F}_i) + f \circ \mathbf{F}_i \operatorname{tr}(\mathrm{D}\hat{\mathbf{V}}_i \,\mathrm{D}\mathbf{F}_i^{-1})\right) \det(\mathrm{D}\mathbf{F}_i) \,\mathrm{d}\hat{x}. \tag{3.18}$$

This is now done automatically in UFL. Example code is shown in Listing 2 for the case $f(x, y) = \sin(x)\cos(y)$.

Listing 2 Firedrake code to compute $\mathrm{d}J$ for Example 3.2.

```
from firedrake import *
mesh = UnitSquareMesh(10, 10)
x, y = X = SpatialCoordinate(mesh)
f = sin(x) * cos(y)
J = f * dx
dJ = assemble(derivative(J, X))
```

We now study objectives with integrands given by finite element functions.

Example 3.3. *Consider a family $\{V_h(\Omega_s)\}_s$ of affinely mapped, scalar finite element spaces on the deformed domains $\{\Omega_s\}_s$. Given $v \in V_h(\Omega)$, we are interested in the sensitivity of the integral of v when the domain is deformed but the coefficient vector corresponding to v is kept constant. Since the element is affinely mapped, we obtain that the relation between the function on the deformed domain and on the initial domain is given by*

$$v_s = v \circ \mathbf{T}_s^{-1}, \tag{3.19}$$

and the objective is

$$J(\Omega_s) = \int_{\Omega_s} v_s \,\mathrm{d}x. \tag{3.20}$$

The expression for the shape derivative on Ω can then be obtained using (3.9)*:*

$$\mathrm{d}J(\Omega)[\mathbf{V}_s] = \int_\Omega \mathrm{d}_\mathrm{s}(v_{\mathbf{T}_s} \circ \mathbf{T}_s) + v \operatorname{div}(\mathbf{V}) \,\mathrm{d}x = \int_\Omega v \operatorname{div}(\mathbf{V}) \,\mathrm{d}x. \tag{3.21}$$

On the other hand, on a single element K_i, we observe that $v|_{K_i} = \hat{v}_i \circ \mathbf{F}_i^{-1}$ for some function $\hat{v}_i$ on the reference element. Assume now the element is deformed with a deformation $\mathbf{V}$, then we note that $v_s|_{\mathbf{T}_s(K)} = \hat{v}_i \circ (\mathbf{F}_i + s\mathbf{V} \circ \mathbf{F}_i)^{-1}$. Let $\mathbf{F} = \mathbf{F}_i + s\mathbf{V} \circ \mathbf{F}_i$ then we conclude

$$v_s|_{\mathbf{T}_s(K)} = v_{i,\mathbf{F}} := \hat{v}_i \circ \mathbf{F}^{-1} \tag{3.22}$$

We thus obtain

$$\begin{aligned} &\mathrm{d}J(\Omega)[\mathbf{V}] \\ &= \sum_i \int_{\hat{K}} \Big(\underbrace{\langle \mathrm{d}_{\mathbf{F}_i}[v_{i,\mathbf{F}_i} \circ \mathbf{F}_i], \hat{\mathbf{V}}_i \rangle}_{=0} + v_{i,\mathbf{F}_i} \circ \mathbf{F}_i \operatorname{tr}(\mathrm{D}\hat{\mathbf{V}}_i \,\mathrm{D}\mathbf{F}_i^{-1}) \Big) \det(\mathrm{D}\mathbf{F}_i) \,\mathrm{d}\hat{x} \\ &= \sum_i \int_{\hat{K}} \left(\hat{v}_i \operatorname{tr}(\mathrm{D}\hat{\mathbf{V}}_i \,\mathrm{D}\mathbf{F}_i^{-1}) \right) \det(\mathrm{D}\mathbf{F}_i) \,\mathrm{d}\hat{x}. \end{aligned} \tag{3.23}$$

The new feature in UFL automatically performs the pullback and calculates the derivative, as shown below.

Listing 3 Firedrake code to compute $\mathrm{d}J$ for Example 3.3.

```
from firedrake import *
mesh = UnitSquareMesh(10, 10)
V = FunctionSpace(mesh, "CG", 1)
x, y = X = SpatialCoordinate(mesh)
v = interpolate(sin(x) * cos(y), V)
J = v * dx
dJ = assemble(derivative(J, X))
```

Example 3.4. *We continue the previous example but now study the case when the shape objective is given by*

$$J(\Omega_s) = \int_{\Omega_s} \|\nabla v_s\|^2 \,\mathrm{d}x. \tag{3.24}$$

By $v_s = v \circ \mathbf{T}_s^{-1}$ we obtain

$$\nabla v_s = \mathrm{D}\mathbf{T}_s^{-\top} \nabla v \tag{3.25}$$

and hence the shape derivative in physical space is given by

$$\begin{aligned} \mathrm{d}J(\Omega)[\mathbf{V}_s] &= \int_\Omega \mathrm{d}_\mathrm{s}(v_{\mathbf{T}_s} \circ \mathbf{T}_s) + \|\nabla v\|^2 \operatorname{div}(\mathbf{V}) \,\mathrm{d}x \\ &= \int_\Omega -2\nabla v \cdot (\mathrm{D}\mathbf{V}^\top \nabla v) + \|\nabla v\|^2 \operatorname{div}(\mathbf{V}) \,\mathrm{d}x \end{aligned} \tag{3.26}$$

To obtain the expression on the reference element, we observe that the integrand on a single element K_i satisfies

$$u_{i,\mathbf{F}} = \| \mathrm{D}\mathbf{F}^{-\top} \nabla \hat{v}_i \|^2 \circ \mathbf{F}^{-1} \tag{3.27}$$

and hence

$$\begin{aligned} &\mathrm{d}J(\Omega)[\mathbf{V}] \\ &= \sum_i \int_{\hat{K}} \left(\langle \mathrm{d}_{\mathbf{F}_i}[u_{i,\mathbf{F}_i} \circ \mathbf{F}_i], \hat{\mathbf{V}}_i \rangle + \| \mathrm{D}\mathbf{F}^{-\top} \hat{v}_i \|^2 \operatorname{tr}(\mathrm{D}\hat{\mathbf{V}}_i \, \mathrm{D}\mathbf{F}_i^{-1}) \right) \det(\mathrm{D}\mathbf{F}_i) \, \mathrm{d}\hat{x} \\ &= \sum_i \int_{\hat{K}} \big(-2(\mathrm{D}\mathbf{F}_i^{-\top} \nabla \hat{v}_i) \cdot (\mathrm{D}\mathbf{F}_i^{-\top} \, \mathrm{D}\hat{\mathbf{V}}_i^{\top} \, \mathrm{D}\mathbf{F}_i{}^{-\top} \nabla \hat{v}_i) + \hat{v}_i \operatorname{tr}(\mathrm{D}\hat{\mathbf{V}}_i \, \mathrm{D}\mathbf{F}_i^{-1}) \big) \\ &\qquad\qquad \det(\mathrm{D}\mathbf{F}_i) \, \mathrm{d}\hat{x}. \end{aligned} \tag{3.28}$$

Listing 4 Firedrake code to compute $\mathrm{d}J$ for Example 3.4.

```
from firedrake import *
mesh = UnitSquareMesh(10, 10)
V = FunctionSpace(mesh, "CG", 1)
x, y = X = SpatialCoordinate(mesh)
v = interpolate(sin(x) * cos(y), V)
J = inner(grad(v), grad(v)) * dx
dJ = assemble(derivative(J, X))
```

So far we have only studied affinely mapped elements. We now highlight the effect of different choices of finite elements on the shape derivative and demonstrate that the implementation in UFL handles these cases automatically.

Example 3.5. *We consider the same objective as in Example 3.3 again, but this time we study the case of an $H(\mathrm{div})$ conforming element. In that case a finite element vector field $\mathbf{v}$ is mapped from the reference domain using the contravariant Piola transform, i.e.*

$$\mathbf{v} \circ \mathbf{F}_i|_{K_i} = \frac{1}{\det(\mathrm{D}\mathbf{F}_i)} \mathrm{D}\mathbf{F}_i \hat{\mathbf{v}}_i. \tag{3.29}$$

As a consequence, the relationship between a function on the original domain and on the deformed domain is given by

$$\mathbf{v}_s = \frac{1}{\det(\mathrm{D}\mathbf{T}_s)} \mathrm{D}\mathbf{T}_s \mathbf{v} \circ \mathbf{T}_s^{-1}. \tag{3.30}$$

To obtain a scalar integrand we take the inner product with some constant vector $\mathbf{a}$,

and then calculate the shape derivative of $J(\Omega) = \int_\Omega \mathbf{v} \cdot \mathbf{a} \, \mathrm{d}x$*:*

$$\begin{aligned} \mathrm{d}J(\Omega)[\mathbf{V}] &= \int_\Omega \big(\mathrm{d_s}(\mathbf{v}_{\mathbf{T}_s} \circ \mathbf{T}_s) + \operatorname{div}(\mathbf{V})\big) \cdot \mathbf{a} \, \mathrm{d}x \\ &= \int_\Omega \big([-\operatorname{div}(\mathbf{V}) + \mathrm{D}\mathbf{V}]\mathbf{v} + \mathbf{v} \operatorname{div}(\mathbf{V})\big) \cdot \mathbf{a} \, \mathrm{d}x \\ &= \int_\Omega \big(\mathrm{D}\mathbf{V}\mathbf{v}\big) \cdot \mathbf{a} \, \mathrm{d}x. \end{aligned} \tag{3.31}$$

For an $H(\mathrm{curl})$ *conforming element the relationships are*

$$\mathbf{v} \circ \mathbf{F}_i|_{K_i} = (\mathrm{D}\mathbf{F}_i)^{-\top} \hat{\mathbf{v}}_i. \tag{3.32}$$

and hence

$$\mathbf{v}_s = (\mathrm{D}\mathbf{T}_s)^{-\top} \mathbf{v} \circ \mathbf{T}_s^{-1}, \tag{3.33}$$

and hence the shape derivative for the same objective is given by

$$\begin{aligned} \mathrm{d}J(\Omega)[\mathbf{V}] &= \int_\Omega \big(\mathrm{d_s}(\mathbf{v}_{\mathbf{T}_s} \circ \mathbf{T}_s) + \operatorname{div}(\mathbf{V})\big) \cdot \mathbf{a} \, \mathrm{d}x \\ &= \int_\Omega \big(\mathrm{D}\mathbf{V}^\top \mathbf{v} + \mathbf{v} \operatorname{div}(\mathbf{V})\big) \cdot \mathbf{a} \, \mathrm{d}x. \end{aligned} \tag{3.34}$$

Despite the objective being seemingly the same as in Example 3.3, the type of finite element function and its mapping to the reference element directly affects the shape derivative. Since UFL *is aware of the mapping that is applied for finite element spaces such as the Nédélec spaces, it automatically performs the correct pullback. We do not give the expression on the reference domain, but show code in Listing 5 that automatically derives the equivalent expressions on the reference element to* (3.31) *and* (3.34)*. We highlight that the user does not have to specify the mapping manually and that the correctness is checked by asserting that the automatically derived shape derivative is equal to (up to machine accuracy) the manually implemented shape derivative.*

Listing 5 Firedrake code to compute the shape derivative for objectives involving nonaffinely mapped elements.

```
from firedrake import *
import numpy as np
mesh = UnitSquareMesh(10, 10)
defo = TestFunction(mesh.coordinates.function_space())
a = Constant((1, 1))
x, y = X = SpatialCoordinate(mesh)

V = FunctionSpace(mesh, "Nedelec 1st kind H(div)", 1)
v = project(as_vector([sin(x), cos(y)]), V)
J = inner(v, a) * dx
dJ = assemble(derivative(J, X, defo))
dJmanual = assemble(inner(grad(defo)*v, a) * dx)
# compare manual implementation and automatically derived shape derivative
assert np.allclose(dJ.vector()[:], dJmanual.vector()[:])

V = FunctionSpace(mesh, "Nedelec 1st kind H(curl)", 1)
v = project(as_vector([sin(x), cos(y)]), V)
J = inner(v, a) * dx
dJ = assemble(derivative(J, X, defo))
dJmanual = assemble(inner(-transpose(grad(defo))*v + v *div(defo), a)*dx)
# compare manual implementation and automatically derived shape derivative
assert np.allclose(dJ.vector()[:], dJmanual.vector()[:])
```

Remark 3.6. *The current implementation in UFL does not work for elements such as the Hermite element that require different pullbacks for point evaluation and derivative degrees of freedom, as for these elements the pullback is not performed in UFL.*

Lastly we consider an objective constrained by a boundary value problem.

Example 3.7. *Consider*

$$J(\Omega_s) = \int_{\Omega_s} u_s \,\mathrm{d}x \tag{3.35}$$

where u_s is the Lagrange finite element solution to the boundary value problem

$$\begin{aligned} -\Delta u_s + u_s &= f \quad \text{in } \Omega_s, \\ \partial u_s/\partial \mathbf{n} &= 0 \quad \text{on } \partial\Omega_s. \end{aligned} \tag{3.36}$$

As described in Section 2.3, we introduce the Lagrangian

$$\mathcal{L}_s(u_s, v_s) \coloneqq J(\Omega_s) + e_s(u_s, v_s), \tag{3.37}$$

where

$$e_s(u_s, v_s) \coloneqq \int_{\Omega_s} \nabla u_s \cdot \nabla v_s + u_s v_s - f v_s \,\mathrm{d}x = 0 \quad \text{for all } v_s \in \mathbf{V}_h(\Omega_s) \tag{3.38}$$

stems from the weak formulation of the PDE constraint (3.36)*. The shape derivative* $\mathrm{d}J$ *in direction* $\mathbf{V}$ *is then equal to the derivative of* $\mathrm{L}_s(u \circ \mathbf{T}_s^{-1}, p \circ \mathbf{T}_s^{-1})$ *with respect*

to s, where u is the solution to (3.38) for $s = 0$ and $p \in V_h(\Omega)$ is the solution to the adjoint boundary value problem. The shape derivative of $\mathrm{L}_s(u \circ \mathbf{T}_s^{-1}, p \circ \mathbf{T}_s^{-1})$ can be computed as in Examples 3.3 and 3.4. The result is

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \int_\Omega (u + \nabla u \cdot \nabla p + up - fp)\operatorname{div}(\mathbf{V}) - p\nabla f \cdot \mathbf{V} - \nabla u \cdot (\mathrm{D}\mathbf{V} + \mathrm{D}\mathbf{V}^T)\nabla p \,\mathrm{d}x. \quad (3.39)$$

Again we omit the equivalent formula on the reference element because of its length. However, as Listing 6 shows, UFL removes the need to derive the shape derivative by hand, and we can easily compute $\mathrm{d}J$ with a single line of code.

Listing 6 Firedrake code to compute $\mathrm{d}J$ from Example 3.7 when $f(x, y) = xy$ in (3.38).

```
from firedrake import *
mesh = UnitSquareMesh(10, 10)
V = FunctionSpace(mesh, "CG", 1)
x, y = X = SpatialCoordinate(mesh)
u, p, v = Function(V), Function(V), TestFunction(V)
e = inner(grad(u), grad(v))*dx + u*v*dx - x*y*v*dx
J = u * dx
solve(e == 0, u) # solve the state equation
L = replace(e, {v: p}) + J # define the Lagrangian
solve(derivative(L, u) == 0, p) # solve the adjoint equation
dJ = assemble(derivative(L, X))
```

3.3 Code validation

We validate our implementation by testing that the Taylor expansions truncated to first and second order satisfy the asymptotic conditions

$$\Delta_1(J, s) = O(s^2) \quad \text{and} \quad \Delta_2(J, s) = O(s^3). \quad (3.40)$$

where

$$\Delta_1(J, s) \coloneqq |J((\mathbf{Id} + s\mathbf{V})(\Omega)) - J(\Omega) - s\,\mathrm{d}J(\Omega)[\mathbf{V}]| \quad (3.41)$$

and

$$\Delta_2(J, s) \coloneqq |J((\mathbf{Id} + s\mathbf{V})(\Omega)) - J(\Omega) - s\,\mathrm{d}J(\Omega)[\mathbf{V}] - \frac{1}{2}s^2\mathrm{d}^2J(\Omega)[\mathbf{V}, \mathbf{V}]|. \quad (3.42)$$

In Figure 3.2, we plot the values of Δ_1 and Δ_2 for $s = 2^{-1}, 2^{-2}, \ldots, 2^{-10}$, and J as in Examples 3.2 and 3.4 from the previous section (we denote these functionals J_1 and J_2 respectively). The vector field $\mathbf{V}$ is chosen randomly. This experiment clearly displays the asymptotic rates predicted by (3.40). The same convergence rates

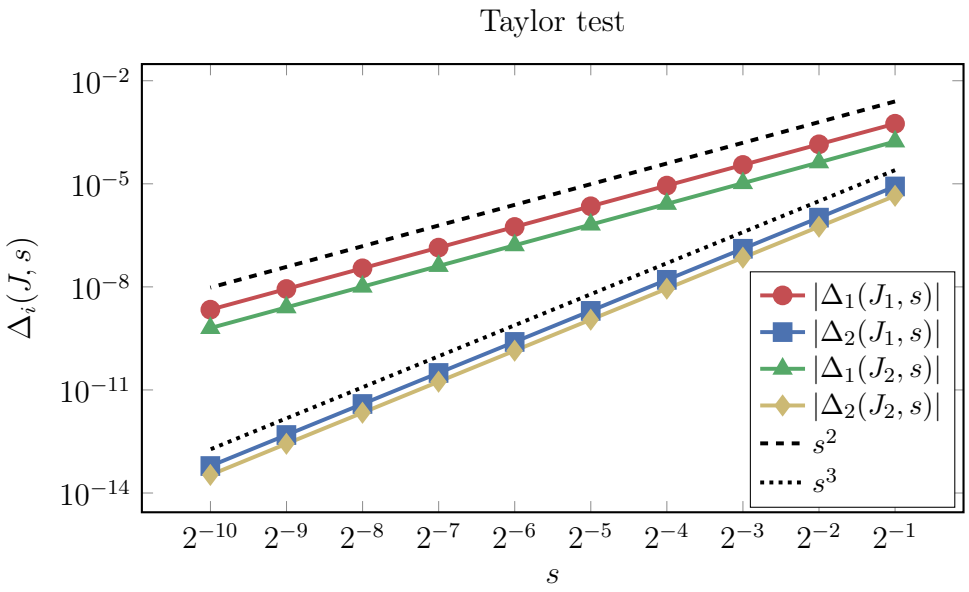


Figure 3.2: Taylor test for Examples 3.2 and 3.7. The convergence rates match the expected convergence.

can be observed for other test cases, including functionals that are not linear in u and involve integration both in Ω and on $\partial\Omega$, and that are constrained to linear and nonlinear boundary value problems with nonconstant right-hand sides and nonconstant Neumann and Dirichlet boundary conditions.

3.4 Shape optimisation of a pipe

In the classical paper by Sigmund [Sig01] *A 99 line topology optimization code written in MATLAB* was presented. The code was developed for educational purposes to illustrate the basic features of a typical topology optimisation routine in a concise but readable fashion. It was later shortened even further in [And+11]. In this section we show that a similarly terse implementation of a basic shape optimisation routine is possible in Firedrake and UFL.

As a test case we consider the optimisation of a pipe to minimise the dissipation of kinetic energy of the fluid into heat. This example is taken from [Sch10, §6.2.3]. To simplify the exposition, we use a very simple optimisation strategy. At the end of the section, we will comment on possible improvements.

A pipe contains viscous fluid, which flows in from the left and is modelled using the incompressible Navier–Stokes equations. To be precise, let Ω be the shape of the pipe, let $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{wall}} \cup \Gamma_{\text{out}}$, and let $\mathbf{u}$ and p be the velocity and the pressure of

the fluid, respectively. Then, $\mathbf{u}$ and p satisfy

$$\begin{aligned} -2\nu\nabla\cdot(\mathrm{E}\mathbf{u}) + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = 0 \quad & \text{in } \Omega, \\ \nabla\cdot\mathbf{u} = 0 \quad & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} \quad & \text{on } \Gamma_{\text{in}}, \\ \mathbf{u} = 0 \quad & \text{on } \Gamma_{\text{wall}}, \\ p\mathbf{n} - 2\nu\mathrm{E}\mathbf{u}\cdot\mathbf{n} = 0 \quad & \text{on } \Gamma_{\text{out}}, \end{aligned}$$

where $\mathrm{E}\mathbf{u} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^\top)$ is the symmetric part of the gradient, $\mathbf{g}$ is given as Poiseuille flow, and as in [Sch10] we pick $\nu = 1/400$.

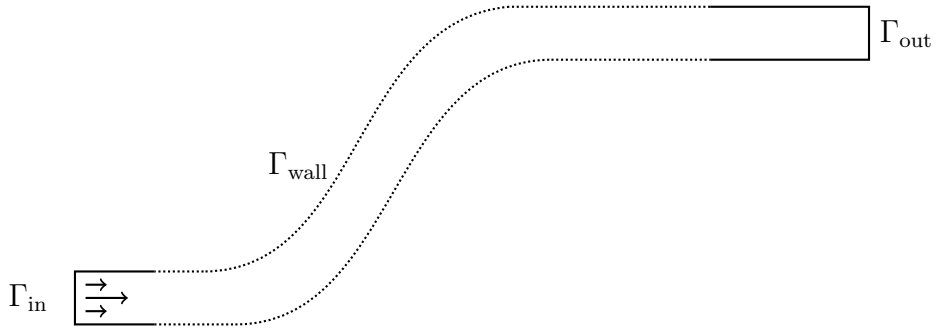


Figure 3.3: Computational domain for the Navier–Stokes pipe optimisation problem. The dotted part of the boundary is free to be deformed.

The goal is to modify the central region of the pipe so that the shape functional

$$J(\Omega) = \int_\Omega \nu\mathrm{E}\mathbf{u} : \mathrm{E}\mathbf{u}\,\mathrm{d}x$$

is minimised. To solve this shape optimisation problem, we parametrise the initial design with a polygonal mesh and update the node coordinates using the steepest descent algorithm with fixed step size. As descent directions, we use Riesz representatives of the shape gradient with respect to the standard H^1_0 inner product. To avoid degenerate results, we penalise changes of the pipe volume. The whole algorithm, including the solution of state and adjoint equations, and the calculation of shape derivatives, is shown in Listing 7.

In lines 2–4 we load the finite element mesh `pipe.msh` and extract the vertex coordinates. The mesh is generated with Gmsh [GR09]. Lines 5–8 define the Gramian matrix of the inner product employed to compute descent directions. In lines 9–13, we define the space of $[\mathbb{P}_2]^2$–$\mathbb{P}_1$ Taylor-Hood finite elements, which we use to discretise the weak formulation of the Navier–Stokes equations and set up the functions containing

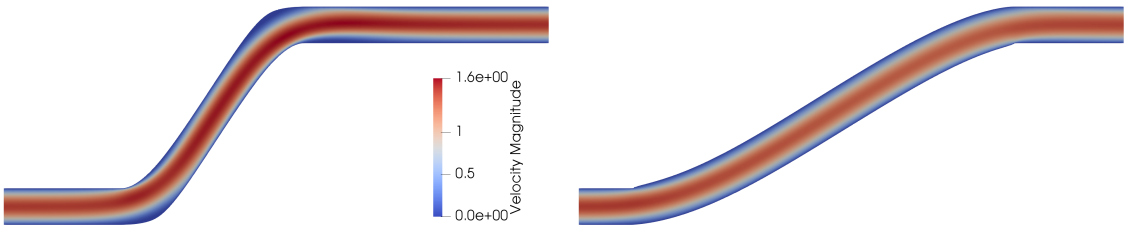


Figure 3.4: Initial (left) and optimised (right) shape of a pipe connecting a given inflow and outflow.

the solutions to the state and adjoint equation as well as the test functions for the weak form. In lines 14–19 we define the weak formulation of the Navier–Stokes equations as well as some parameters to prescribe the use of the MUMPS direct solver [Ame+00] to solve the linearised equations. In lines 20–26, we define the shape functional J, the functional describing the volume of the shape, as well as the Lagrangian and its shape derivative.

In lines 27–31 we set up a function that updates the solution to the state and the adjoint equations. Note that, whenever the function `solve_state_and_adjoint` is called, the new values of the velocity $\mathbf{u}$ are stored in the file `u.pvd` (which can be visualised using Paraview [AGL05]). Finally, lines 33–46 contain the optimisation algorithm: for 100 iterations we compute the shape derivative (lines 34–35), compute the descent direction (line 36), update the domain (line 38), and update the state and adjoint solutions (line 39). The optimised shape is displayed in Figure 3.4, the convergence is shown in Figure 3.5.

Note that Listing 7 can be readily used for a 3D problem by simply passing a 3D mesh and changing the inflow boundary condition.

While this example demonstrates how even a fairly complicated shape optimisation problem can be expressed and solved very concisely in Firedrake and UFL, it also demonstrates that typically shape optimisation problems suffer from several difficulties that go beyond the solution of the PDE and the calculation of derivatives. In Section 2.4 we discussed that the search for deformations is poorly conditioned since the functional is nearly invariant with respect to deformations with support inside the domain. This problem manifests itself here in the form of significant slower convergence of the optimisation scheme after ~ 100 iterations. In addition, we observe kinks in the shape where the domain changes from being fixed to being free to move. This is likely due to the singularity that arises when solving the Poisson problem in order to obtain the shape gradient, since at these points the boundary condition changes from a Dirichlet to a natural boundary condition.

Listing 7 Complete code to run a simple steepest descent algorithm to optimise the shape of the pipe. Lines 23 and 25 use the newly developed automatic shape differentiation.

```
1  from firedrake import *

2  mesh = Mesh("pipe.msh")
3  coords = mesh.coordinates.vector()
4  X = SpatialCoordinate(mesh)

5  W = mesh.coordinates.function_space()
6  gradJ = Function(W)
7  phi, psi = TrialFunction(W), TestFunction(W)
8  A_riesz = assemble(inner(grad(phi), grad(psi)) * dx, bcs=DirichletBC(W, 0, [1, 2, 3]))

9  Z = VectorFunctionSpace(mesh, "CG", 2) * FunctionSpace(mesh, "CG", 1)
10 z, z_adj = Function(Z), Function(Z)
11 u, p = split(z)
12 test = TestFunction(Z)
13 v, q = split(test)

14 nu = 1./400.
15 e = 2*nu*inner(sym(grad(u)), sym(grad(v)))*dx - p*div(v)*dx \
16     + inner(dot(grad(u), u), v)*dx + div(u)*q*dx
17 uin = 6 * as_vector([(1-X[1])*X[1], 0])
18 bcs = [DirichletBC(Z.sub(0), 0., [3, 4]), DirichletBC(Z.sub(0), uin, 1)]
19 sp = {"mat_type": "aij", "pc_type": "lu", "pc_factor_mat_solver_type": "mumps"}

20 J = nu * inner(sym(grad(u)), sym(grad(u))) * dx
21 volume = Constant(1.) * dx(domain=mesh)
22 target_volume = assemble(volume)
23 dvol = derivative(volume, X)
24 L = replace(e, {test: z_adj}) + J
25 dL = derivative(L, X)
26 c = 1/20 # penalty parameter for volume constraint

27 out = File("u.pvd")
28 def solve_state_and_adjoint():
29     solve(e==0, z, bcs=bcs, solver_parameters=sp)
30     solve(derivative(L, z)==0, z_adj, bcs=homogenize(bcs), solver_parameters=sp)
31     out.write(z.split()[0])

32 solve_state_and_adjoint()
33 for i in range(200):
34     dJ = assemble(dL).vector() + assemble(dvol).vector() * c * 2 \
35         * (assemble(volume)-target_volume)
36     solve(A_riesz, gradJ, dJ)
37     print("i = %3i; J = %.6f; ||dJ|| = %.6f" % (i, assemble(J), norm(grad(gradJ))))
38     coords -= gradJ.vector()
39     solve_state_and_adjoint()
```

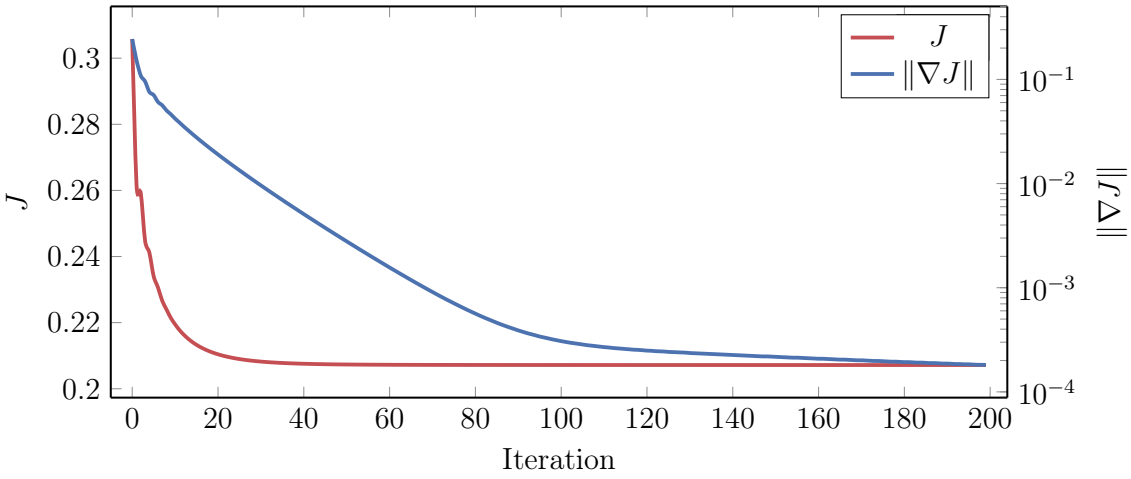


Figure 3.5: Convergence for the pipe optimisation problem: the value of the objective is reduced by approximately a third, the H^1-norm of the gradient is reduced by three orders of magnitude. Convergence slows down after approximately 100 iterations.

Finally, we note that the convergence could be improved significantly by using a quasi Newton method and a linesearch. We refrain from using either here to keep the example terse.

Code availability

The automated calculation of shape derivatives has been merged into UFL, Firedrake and FEniCS. The code for the numerical experiments from this chapter is available at `https://github.com/florianwechsung/ThesisNumerics`. For reproducibility, all major Firedrake components as well as the code used to run these examples have been archived on Zenodo [Fir19; Wec19c].

Chapter 4

Nearly conformal mesh deformation methods for shape optimisation in two dimensions

In Chapter 2 we formulated shape optimisation problems as a search for optimal deformations $\mathbf{T} = \mathbf{Id} + \mathbf{X}$ of an initial domain Ω. Many shape optimisation problems are subject to a PDE constraint and often this constraint is solved using the finite element method. In such cases the domain Ω is replaced by an approximation Ω^h that consists of a regular triangulation. Then the optimisation process successively deforms the initial domain Ω^h by applying transformations to the grid points. As mentioned in Section 2.4, there are many deformations $\mathbf{T}$ of an initial domain that all map to the same deformed domain.

As we will discuss in Section 4.1, in order to obtain an accurate solution to the PDE constraint, the geometry of the mesh is crucial; in many cases a mesh consisting of triangles / tetrahedra that are as close to unilateral as possible is desirable. However, in general there is no guarantee that the deformed mesh $\mathbf{T}(\Omega_0^h)$ is still regular, i.e. that cells have not become highly stretched or even overlap. For this reason, a common strategy is to either remesh the domain at every step in the optimisation or every few steps when the mesh quality has degraded past a threshold [Mor+12; Sch18; EH18; HL19]. However, frequent remeshing is undesirable for several reasons: first, it can be an expensive process for complicated geometries and may even require manual intervention. Secondly, changing the mesh introduces noise into the objective, as two different meshes of the same geometry usually result in slightly different solutions to the PDE constraint and different values of the objective.

Therefore, when remeshing is impractical, not possible, or expensive, it is important to find 'tame' deformation fields $\mathbf{T}$ that deform the mesh 'as well as possible' while

decreasing the cost function J sufficiently well.

In [Sch14] a two stage approach is introduced: a 'good' deformation is calculated on the boundary first and then extended into the volume by solving a convection-diffusion equation. In several papers the good properties of the linear elasticity equations have been observed for mesh deformation [Dwi09; SS16]. Here the shape derivative is essentially viewed as a force acting on an elastic medium. Lastly, we mention free-form deformation boxes, initially developed to model simple deformations in computer graphics but now frequently employed in aerodynamic design optimisation to parametrise shape deformations [Sam04; PEA15].

The approach presented in this work is based on conformal mappings in two dimensions; this is the family of mappings that preserve angles. These mapping are attractive as mesh properties such as equilateral elements remain constant under a conformal mapping and have been used for mesh generation extensively [Bro81; Riz81; Ive82]. We consider the two-dimensional case as then conformality can be achieved by enforcing that the deformation satisfies the Cauchy–Riemann equations. However, as the Cauchy–Riemann equations do not admit solutions for any given boundary data, we consider a relaxed formulation instead. Based on this idea, we will develop an inner product that penalises nonconformal deformations. A gradient based method using this inner product will then at each step pick a 'good' deformation that preserves the mesh quality. We will briefly outline a possible extension to three dimensions at the end of the chapter.

4.1 Discussion of deformations and mesh quality

There are several reasons why the choice of deformation is important. As discussed in the previous chapter, the primary criterion for a good deformation is that it should be a diffeomorphism: this is crucial as this is required by the transformation rule relating integrals over $\mathbf{T}(\Omega)$ and Ω, which is used in the calculation of shape derivatives.

In addition, when representing the domain with a mesh and using finite elements to solve for a PDE constraint on the domain, the deformation becomes even more important as it directly influences the underlying mesh. On a convex domain an overlap of the mesh implies a negative determinant of the deformation, meaning that mass and stiffness matrices become indefinite and iterative solvers such as conjugate gradients will fail.

However, merely being a diffeomorphism is not enough for a good deformation. Deformations that strongly stretch or compress a mesh resulting in long and thin

elements can have consequences for both the approximation error and the performance of linear solvers. We henceforth assume that $\Omega \subset \mathbb{R}^2$ is a bounded domain with a polygonal boundary $\partial\Omega$. Let $\mathcal{M} = \{K\}$ be a simplicial triangulation of Ω consisting of triangles K such that

$$\overline{\Omega} = \bigcup_{K \in \mathcal{M}} K. \tag{4.1}$$

For every element $K \in \mathcal{M}$, $h(K)$ denotes the diameter of the smallest ball containing K and $\rho(K)$ is the diameter of the largest ball contained in K. A measure for the quality of an element is given by the ratio

$$\eta(K) := \frac{h(K)}{2\rho(K)} \in [1, \infty). \tag{4.2}$$

The value of $\eta(K)$ is 1 for equilateral triangles and becomes large for long-and-thin triangles. Convergence results for finite element solutions are often based on the fundamental classic interpolation estimate (see e.g. [Cia02, Theorem 3.1.5]), stating that

$$|v - \Pi_K^k v|_{H^m(K)} \leq C \frac{h(K)^{k+1}}{\rho(K)^m} |v|_{H^{k+1}(K)} \quad \text{for } v \in H^{k+1}(K) \tag{4.3}$$

where $m \in \{0, 1\}$ and k is the polynomial degree and Π_K^k is the interpolation into the space of polynomials of degree k on the element K. If we choose $m = 1$ and $k = 1$, we see that the approximation error for the gradient when using piecewise affine elements depends both on the mesh size $h(K)$ but also on the quality indicator $\eta(K)$.

These interpolation estimates have been significantly refined in subsequent work and it has been understood that stretched elements do not reduce the approximation quality if they are aligned with the Hessian of the function [FP01]. A classical example for this are boundary layer meshes at the wall in the context of fluid dynamics: the solution changes rapidly in the direction away from the wall but only slowly tangentially along the wall. Hence, elements that are thin in normal direction but long in tangential direction are often used. However, for cases with less a priori knowledge about the structure of the solution, an isotropic mesh is typically desired. We refer to [Knu07] for a more detailed discussion of mesh quality measures and their use.

Another reason that uniform meshes are often desirable is the effect of stretched meshes on the performance of linear solvers. This is closely related to anisotropy in the differential operators. To see this, consider the case of a typical finite difference

stencil for the Laplacian on a structured grid in two dimensions:

$$\begin{aligned}\Delta u(x,y) = \partial_{xx}u(x,y) + \partial_{yy}u(x,y) \approx &\frac{u(x-h_x,y) + u(x+h_x,y) - 2u(x,y)}{h_x^2} \\ &+ \frac{u(x,y-h_y) + u(x,y+h_y) - 2u(x,y)}{h_y^2}.\end{aligned} \tag{4.4}$$

If we set $h_y = h_x/\sqrt{\varepsilon}$, i.e. the grid spacing is large in y direction, then the resulting stencil is equivalent to that of the operator $\partial_{xx}u(x,y) + \varepsilon\partial_{yy}u(x,y)$ on a uniform grid. This operator has a near-nullspace of functions that are independent of y and hence becomes poorly conditioned as $\varepsilon \to 0$. This problem is studied in detail for example in the multigrid community where methods such as plane-smoothing or semi-coarsening have been developed to capture this nullspace [Bra82, Section 3.3], [Wes04, p. 124], [PG97]. However, these methods are more expensive and add complexity to the implementation, and for isotropic problems a regular grid will mean that they can be avoided.

For the reasons discussed, we are interested in finite element meshes consisting of elements that are nearly equilateral. If we assume that the initial mesh is regular, then a way of preserving this quality is by considering angle-preserving mappings.

4.2 Conformal mappings in two dimensions

A function $\mathbf{X} : \Omega \to \mathbb{R}^2$ is called *conformal* if it preserves angles between curves. Identifying the vector field $\mathbf{X}$ in the canonical way with a complex function, it is well known that a complex differentiable function with nonvanishing complex derivative is conformal [Gam01, p. 59]. Conversely, a conformal C^1 vector field with nonvanishing real derivatives is in fact complex differentiable [Gam01, p. 126].

We recall that complex differentiability can be characterised by the Cauchy–Riemann equations: any C^1 vector field $\mathbf{X} = (X_1, X_2)$ with partial derivatives that satisfy

$$\begin{aligned}\partial_x X_1 - \partial_y X_2 &= 0, \\ \partial_y X_1 + \partial_x X_2 &= 0,\end{aligned} \tag{4.5}$$

is complex differentiable. There exist similar results requiring weaker assumptions on $\mathbf{X}$; we refer to [GM78]. Again, the converse result holds. A function that is complex differentiable everywhere is called *holomorphic*.

Denoting

$$\mathrm{B} := \begin{pmatrix} -\partial_x & \partial_y \\ \partial_y & \partial_x \end{pmatrix} \tag{4.6}$$

we are looking for deformations $\mathbf{X}$ that satisfy $\mathrm{B}\mathbf{X} = 0$. In an abuse of notation, the partial derivatives in (4.6) will refer to strong or weak derivatives, depending on the regularity of $\mathbf{X}$. The operator B gives rise to two natural boundary value problems:

Problem 4.1 (Cauchy–Riemann Dirichlet BVP)**.**
Given $\mathbf{h} \in [C(\partial\Omega)]^2$ *the Cauchy–Riemann Dirichlet boundary value problem reads: find* $\mathbf{X} \in [C^1(\Omega)]^2 \cap [C(\overline{\Omega})]^2$ *such that*

$$\mathrm{B}\mathbf{X} = 0 \quad \text{in } \Omega, \tag{4.7a}$$

$$\mathbf{X} = \mathbf{h} \quad \text{on } \partial\Omega. \tag{4.7b}$$

Problem 4.2 (Cauchy–Riemann Neumann BVP)**.**
Given $\mathbf{g} \in [C(\partial\Omega)]^2$ *the Cauchy–Riemann Neumann boundary value problem reads: find* $\mathbf{X} \in [C^1(\overline{\Omega})]^2$ *such that*

$$\mathrm{B}\mathbf{X} = 0 \quad \text{in } \Omega, \tag{4.8a}$$

$$\mathrm{D}\mathbf{X}\mathbf{n} = \mathbf{g} \quad \text{on } \partial\Omega, \tag{4.8b}$$

where $\mathbf{n}$ *denotes the outward-pointing normal vector field along* $\partial\Omega$*.*

Observe that the twofold application of the Cauchy–Riemann operator yields the vectorial Laplace operator $\mathrm{B}^2 := \mathrm{B}\circ\mathrm{B}{=}\mathbf{I}_2\Delta$, where $\mathbf{I}_2 \in \mathbb{R}^{2\times 2}$ denotes the identity matrix in $\mathbb{R}^2$. Therefore, every holomorphic mapping is also harmonic, that is, the Laplacian of the function vanishes. The converse does not hold (take e.g. $\mathbf{X}(x, y) = (ax, y)$ for $a \neq 1$), and in fact the boundary value problems (4.7) and (4.8) do not have solutions in general. However, when the data $\mathbf{h}$ in the Dirichlet case and $\mathbf{g}$ in the Neumann case satisfy certain compatibility conditions then (4.7) and (4.8) admit solutions, respectively. In [Beg05, Theorem 7 and 8] a closed formula for the solution to (4.7) and (4.8) on the unit disc is given and necessary and sufficient conditions are provided.

The set of holomorphic mappings might seem to be small, however, the Riemann mapping theorem guarantees that every simply-connected domain $\Omega \subset \mathbb{R}^2$ can be mapped to the unit disc $\mathbf{D} \subset \mathbb{R}^2$ via a bi-holomorphic map. The precise statement of the Riemann mapping theorem is as follows; see [Con95, Thm. 4.2, p. 160].

Definition 4.3.
Let $\Omega \subset \mathbb{R}^2$ *be open. We call* $\mathbf{X} : \Omega \to \mathbf{X}(\Omega)$ *a bi-holomorphic mapping if* $\mathbf{X}$ *is holomorphic and injective with holomorphic inverse.*

Theorem 4.4.
Let $\Omega \subsetneq \mathbb{R}^2$ *be a simply-connected domain. Then there exists a bi-holomorphic map* $\mathbf{X}$ *from* Ω *onto the unit disc* $\mathbf{D} \subset \mathbb{R}^2$*, such that* $\mathbf{X}(\mathbf{a}) = 0$ *and* $\mathbf{X}'(\mathbf{a})$ *is positive definite.*

This theorem motivates us to perform shape optimisation in two dimensions using conformal mappings.

4.3 Nearly conformal shape gradients

While usually the shape derivative $\mathrm{d}J(\Omega)[\mathbf{V}]$ in (2.17) is defined with at least $\mathbf{V} \in C^{0,1}(\overline{\Omega}; \mathbb{R}^d)$ regularity in mind, for optimisation purposes it is commonly assumed that $\mathrm{d}J(\Omega)$ is a well-defined and continuous functional on $H^1(\Omega; \mathbb{R}^d)$. This is necessary to obtain a Hilbert-space structure to perform gradient based optimisation. In practice, the function space $H^1(\Omega; \mathbb{R}^d)$ is often discretised using Lagrange finite element functions, which in turn have $C^{0,1}(\overline{\Omega}; \mathbb{R}^d)$ regularity again.

Let $(\mathcal{H}, (\cdot,\cdot)_{\mathcal{H}})$ be a Hilbert space that is continuously embedded into $H^1(\Omega; \mathbb{R}^2)$. The *$\mathcal{H}$-shape gradient* of J at Ω with respect to $(\mathcal{H}, (\cdot,\cdot)_{\mathcal{H}})$ is defined as the unique element $\nabla J(\Omega) \in \mathcal{H}$ that satisfies

$$(\nabla J(\Omega), \mathbf{V})_{\mathcal{H}} = \mathrm{d}J(\Omega)[\mathbf{V}] \quad \text{for all } \mathbf{V} \in \mathcal{H}. \tag{4.9}$$

There is a close relation between the $\mathcal{H}$-shape gradient and steepest descent directions in $\mathcal{H}$. Indeed, it can be shown (see [ES18, Lemma 2.2]) that the negative normalised $\mathcal{H}$-shape gradient $-\nabla J(\Omega)/\|\nabla J(\Omega)\|_{\mathcal{H}}$ solves the minimisation problem

$$\min_{\substack{\mathbf{U}\in\mathcal{H},\\ \|\mathbf{U}\|_{\mathcal{H}}=1}} \mathrm{d}J(\Omega)[\mathbf{U}] \tag{4.10}$$

and hence the negative $\mathcal{H}$-shape gradient is in fact the steepest descent direction in $\mathcal{H}$. In addition, we observe that the $\mathcal{H}$-shape gradient is the minimiser of

$$\min_{\mathbf{U}\in\mathcal{H}} \frac{1}{2}\|\mathbf{U}\|_{\mathcal{H}}^2 - \mathrm{d}J(\Omega)[\mathbf{U}]. \tag{4.11}$$

In order to retain mesh quality, we propose to use $\mathcal{H}$-shape gradients that satisfy the Cauchy–Riemann equations as well as possible; one way to achieve this is by replacing (4.11) by the following optimisation problem

$$\min_{\substack{\mathbf{U}\in\mathcal{H},\\ \mathrm{B}\mathbf{U}=0}} \frac{1}{2}\|\mathbf{U}\|_{\mathcal{H}}^2 - \mathrm{d}J(\Omega)[\mathbf{U}], \tag{4.12}$$

where B corresponds to the Cauchy–Riemann equations using weak partial derivatives. We will see later that the minimisation problem (4.12) is related to the Cauchy–Riemann Neumann boundary value problem. However, we already know that the Neumann boundary value problem does not always have solutions because the boundary

data has to satisfy compatibility conditions. Therefore, even when Ω is not a stationary point, it might happen that the only solution to (4.12) is given by $\mathbf{U} \equiv 0$.

In order to enlarge the shape deformation space, but still retain some conformality we propose to enforce the conformality constraint weakly by adding a penalty term. We will see later that in certain cases this approach yields strictly conformal mappings. Thus, we consider the following regularised version of (4.12),

$$\min_{\mathbf{U}\in\mathcal{H}} \frac{1}{2}\left(\frac{1}{\alpha}\|\mathrm{B}\mathbf{U}\|_{L^2}^2 + \|\mathbf{U}\|_{\mathcal{H}}^2\right) - \mathrm{d}J(\Omega)[\mathbf{U}], \quad \alpha > 0. \tag{4.13}$$

The existence of a unique minimiser follows by standard arguments (see proof of Lemma 4.6). In addition, this minimiser is then the $\mathcal{H}$-shape gradient of J at Ω with respect to $\mathcal{H}$ equipped with the inner product

$$(\mathbf{U},\mathbf{V})_{\mathrm{CR}(\alpha)+\mathcal{H}} := \frac{1}{\alpha}(\mathrm{B}\mathbf{U},\mathrm{B}\mathbf{V})_{L^2} + (\mathbf{U},\mathbf{V})_{\mathcal{H}}. \tag{4.14}$$

Lastly, we remark that the solution in the limit $\alpha \to 0$ is also known as a *viscosity solution*, see [Att96].

4.3.1 $\mathrm{CR}(\alpha) + \mathring{H}^1$-shape gradients

We begin by studying the space $\mathcal{H} := \mathring{H}^1(\Omega;\mathbb{R}^2)$ consisting of all functions $\mathbf{U} \in H^1(\Omega;\mathbb{R}^2)$ with mean zero, that is, $\int_\Omega \mathbf{U}\,\mathrm{d}x = 0$. This space becomes a Hilbert space when equipped with the inner product

$$(\mathbf{U},\mathbf{V})_{\mathring{H}^1} := (\mathrm{D}\mathbf{U},\mathrm{D}\mathbf{V})_{[L^2]^{2\times 2}} := \int_\Omega \mathrm{D}\mathbf{U} : \mathrm{D}\mathbf{V}\,\mathrm{d}x. \tag{4.15}$$

The shape gradient is then given by the solution to the following optimisation problem.

Problem 4.5.
We study the following relaxation of (4.12)*: find a minimiser* $\mathbf{U}_\alpha \in \mathring{H}^1(\Omega;\mathbb{R}^2)$ *of*

$$\min_{\mathbf{U}\in\mathring{H}^1(\Omega;\mathbb{R}^2)} \frac{1}{2}\left(\frac{1}{\alpha}\|\mathrm{B}\mathbf{U}\|_{[L^2]^2}^2 + \|\,\mathrm{D}\mathbf{U}\|_{[L^2]^{2\times 2}}^2\right) - \mathrm{d}J(\Omega)[\mathbf{U}], \quad \alpha > 0, \tag{4.16}$$

where $\|\,\mathrm{D}\mathbf{U}\|_{[L^2]^{2\times 2}}^2 := \int_\Omega \mathrm{D}\mathbf{U} : \mathrm{D}\mathbf{U}\,\mathrm{d}x$.

To simplify the exposition, we assume that the derivative can be expressed as the integral over a function $\mathbf{g}$ on the boundary (cf. Section 2.4), i.e.

$$\mathrm{d}J(\Omega)[\mathbf{V}] = \int_{\partial\Omega} \mathbf{g}\cdot\mathbf{V}\,\mathrm{d}s \quad \text{for all } \mathbf{V}\in\mathcal{H}. \tag{4.17}$$

We emphasise that the main result of this section, Proposition 4.8, holds as well for other inner products and under the assumption that $\mathrm{d}J(\Omega)$ is merely an element of the dual space of $\mathcal{H}$, i.e., it is compatible with using the volume formula of the shape derivative.

Lemma 4.6.
There exists a unique minimiser $\mathbf{U}_\alpha \in \mathring{H}^1(\Omega, \mathbb{R}^2)$ *of* (4.16) *and* $\mathbf{U}_\alpha$ *satisfies*

$$\frac{1}{\alpha}(\mathrm{B}\mathbf{U}_\alpha, \mathrm{B}\mathbf{V})_{[L^2]^2} + (\mathrm{D}\mathbf{U}_\alpha, \mathrm{D}\mathbf{V})_{[L^2]^{2\times 2}} = \int_{\partial\Omega} \mathbf{g}\cdot\mathbf{V}\,\mathrm{d}s \tag{4.18}$$

for all $\mathbf{V} \in \mathring{H}^1(\Omega;\mathbb{R}^2)$. *If* $\mathbf{U}_\alpha \in C^2(\Omega;\mathbb{R}^2)\cap C^1(\overline{\Omega};\mathbb{R}^2)$ *then* $\mathbf{U}_\alpha$ *satisfies the corresponding Euler-Lagrange equations:*

$$\Delta\mathbf{U}_\alpha = 0 \quad \text{in } \Omega, \tag{4.19a}$$

$$\mathrm{D}\mathbf{U}_\alpha\mathbf{n} + \hat{\mathrm{D}}\mathbf{U}_\alpha\mathbf{n} + \alpha\,\mathrm{D}\mathbf{U}_\alpha\mathbf{n} = \alpha\mathbf{g} \quad \text{on } \partial\Omega, \tag{4.19b}$$

where

$$\hat{\mathrm{D}}\mathbf{U} = \begin{pmatrix} -\partial_y U_2 & \partial_x U_2 \\ \partial_y U_1 & -\partial_x U_1 \end{pmatrix}. \tag{4.20}$$

Proof. It follows from Poincaré's inequality that $\mathbf{U} \mapsto \|\,\mathrm{D}\mathbf{U}\|_{[L^2]^{2\times 2}}$ is a norm on $\mathring{H}^1(\Omega;\mathbb{R}^2)$ that is equivalent to the norm $\mathbf{U}\mapsto\|\mathbf{U}\|_{[H^1]^2} := \sqrt{\|\mathbf{U}\|_{[L^2]^2} + \|\,\mathrm{D}\mathbf{U}\|_{[L^2]^{2\times 2}}}$. This implies that the functional (4.16) is strictly convex on $\mathring{H}^1(\Omega;\mathbb{R}^2)$. Furthermore, it is lower semi-continuous and hence by convexity also weakly lower semi-continuous on this space. As a result, existence of a unique minimiser follows from the direct method of calculus of variations.

Multiplying with α, the first order necessary and sufficient optimality condition of (4.16) reads: find $\mathbf{U}_\alpha \in \mathring{H}^1(\Omega,\mathbb{R}^2)$ such that

$$\int_\Omega \mathrm{B}\mathbf{U}_\alpha\cdot\mathrm{B}\mathbf{V} + \alpha\,\mathrm{D}\mathbf{U}_\alpha\colon\,\mathrm{D}\mathbf{V}\,\mathrm{d}x = \alpha\int_{\partial\Omega}\mathbf{g}\cdot\mathbf{V}\,\mathrm{d}s \quad \text{for all } \mathbf{V}\in\mathring{H}^1(\Omega;\mathbb{R}^2). \tag{4.21}$$

Denoting $\mathbf{U}_\alpha = (U_1, U_2)$ and $\mathbf{V} = (V_1, V_2)$, by the definition of B, using integration by parts twice and Schwarz's theorem, we have that

$$\begin{aligned}
&\int_\Omega \mathrm{B}\mathbf{U}_\alpha\cdot\mathrm{B}\mathbf{V}\,\mathrm{d}x && (4.22)\\
&= \int_\Omega \partial_x U_1\partial_x V_1 + \partial_y U_2\partial_y V_2 - \partial_y U_2\partial_x V_1 - \partial_x U_1\partial_y V_2 && (4.23)\\
&\qquad + \partial_y U_1\partial_y V_1 + \partial_y U_1\partial_x V_2 + \partial_x U_2\partial_y V_1 + \partial_x U_2\partial_x V_2\,\mathrm{d}x\\
&= \int_\Omega \mathrm{D}\mathbf{U}_\alpha\colon\,\mathrm{D}\mathbf{V} - \partial_y U_2\partial_x V_1 - \partial_x U_1\partial_y V_2 + \partial_y U_1\partial_x V_2 + \partial_x U_2\partial_y V_1\,\mathrm{d}x && (4.24)\\
&= \int_\Omega \mathrm{D}\mathbf{U}_\alpha\colon\,\mathrm{D}\mathbf{V} - \partial_y U_2\partial_x V_1 - \partial_x U_1\partial_y V_2 - \underbrace{\partial_x\partial_y}_{=\partial_y\partial_x}U_1V_2 - \underbrace{\partial_y\partial_x}_{=\partial_x\partial_y}U_2V_1\,\mathrm{d}x && (4.25)\\
&\qquad + \int_{\partial\Omega}\partial_y U_1 V_2 n_1 + \partial_x U_2 V_1 n_2\,\mathrm{d}s
\end{aligned}$$

$$= \int_\Omega \mathrm{D}\mathbf{U}_\alpha \colon \mathrm{D}\mathbf{V} - \partial_y U_2 \partial_x V_1 - \partial_x U_1 \partial_y V_2 + \partial_x U_1 \partial_y V_2 + \partial_y U_2 \partial_x V_1 \,\mathrm{d}x \tag{4.26}$$

$$+ \int_{\partial\Omega} \partial_y U_1 V_2 n_1 + \partial_x U_2 V_1 n_2 - \partial_x U_1 V_2 n_2 - \partial_y U_2 V_1 n_1 \,\mathrm{d}s$$

$$= \int_\Omega \mathrm{D}\mathbf{U}_\alpha \colon \mathrm{D}\mathbf{V} \,\mathrm{d}x + \int_{\partial\Omega} \partial_y U_1 V_2 n_1 + \partial_x U_2 V_1 n_2 - \partial_x U_1 V_2 n_2 - \partial_y U_2 V_1 n_1 \,\mathrm{d}s. \tag{4.27}$$

Using this identity in (4.21) we obtain

$$\int_\Omega -(1+\alpha)\,\mathrm{div}(\mathrm{D}\mathbf{U}_\alpha)\cdot\mathbf{V}\,\mathrm{d}x + \int_{\partial\Omega} (\mathrm{D}\mathbf{U}_\alpha\mathbf{n} + \hat{\mathrm{D}}\mathbf{U}_\alpha\mathbf{n} + \alpha\,\mathrm{D}\mathbf{U}_\alpha\mathbf{n})\cdot\mathbf{V}\,\mathrm{d}s = \alpha\int_{\partial\Omega}\mathbf{g}\cdot\mathbf{V}\,\mathrm{d}s, \tag{4.28}$$

and (4.19) follows. □

It turns out that if $\mathbf{g}$ is compatible, that is, if the Cauchy–Riemann Neumann problem (4.8) admits a solution, then this solution is also a minimiser of (4.16).

Corollary 4.7.
Let $\mathbf{g} \in C(\partial\Omega;\mathbb{R}^2)$ be given. Suppose that (4.8) admits a solution $\mathbf{U} \in C^2(\Omega;\mathbb{R}^2) \cap C^1(\overline{\Omega};\mathbb{R}^2)$. Then $\mathbf{U}$ (up to a constant) is the minimiser of (4.16) and satisfies the Neumann problem

$$\Delta\mathbf{U} = 0 \quad \text{in } \Omega, \tag{4.29a}$$

$$\mathrm{D}\mathbf{U}\mathbf{n} = \mathbf{g} \quad \text{on } \partial\Omega. \tag{4.29b}$$

Proof. If $\mathbf{U}$ is a solution to (4.8) then $\Delta\mathbf{U} = 0$ in Ω and also $\mathrm{D}\mathbf{U} = -\hat{\mathrm{D}}\mathbf{U}$ on $\partial\Omega$. Therefore, $\mathbf{U}$ solves

$$\Delta\mathbf{U} = 0 \quad \text{in } \Omega, \tag{4.30}$$

$$(\mathrm{D}\mathbf{U} - \mathrm{D}\mathbf{U} + \alpha\,\mathrm{D}\mathbf{U})\mathbf{n} = \alpha\mathbf{g} \quad \text{on } \partial\Omega, \tag{4.31}$$

and thus owing to Lemma 4.6 $\mathbf{U}$ is the unique minimiser (up to a constant) of (4.16). □

The previous lemma only gives a characterisation of the minimiser of (4.16) in the case that the Cauchy–Riemann Neumann boundary value problem admits a solution. In general, this problem does not have a solution and as a result the minimisers of (4.16) cannot be holomorphic. However, we can show that they are nearly holomorphic by splitting the space $\mathring{H}^1(\Omega;\mathbb{R}^2)$ into a holomorphic and a nonholomorphic part. In the following proposition we think of $\mathcal{H} = \mathring{H}^1(\Omega;\mathbb{R}^2)$ but $\mathcal{H}$ could be any Hilbert space continuously embedded into $H^1(\Omega;\mathbb{R}^2)$.

Proposition 4.8.
Let $\mathcal{H} = \mathring{H}^1(\Omega;\mathbb{R}^2)$ and consider the orthogonal decomposition $\mathcal{H} = V \oplus V^\perp$ for $V = \{\mathbf{U} \in \mathcal{H} : \mathrm{B}\mathbf{U} = 0\}$. Furthermore, let $\boldsymbol{\varphi}_\mathbf{g}$ be the unique solution to

$$(\boldsymbol{\varphi}_\mathbf{g}, \mathbf{V})_\mathcal{H} = \int_{\partial\Omega} \mathbf{g} \cdot \mathbf{V} \,\mathrm{d}s \quad \text{for all } \mathbf{V} \in \mathcal{H}, \tag{4.32}$$

where $\mathbf{g} \in L_2(\partial\Omega)$ is a given function. Denote $\boldsymbol{\varphi}_\mathbf{g} = \boldsymbol{\varphi}_\mathbf{g}^{(1)} + \boldsymbol{\varphi}_\mathbf{g}^{(2)}$ for $\boldsymbol{\varphi}_\mathbf{g}^{(1)} \in V$ and $\boldsymbol{\varphi}_\mathbf{g}^{(2)} \in V^\perp$ and decompose the solution $\mathbf{U}_\alpha = \mathbf{U}_\alpha^{(1)} + \mathbf{U}_\alpha^{(2)}$ of (4.16) in the same way. Then

(i) $\mathbf{U}_\alpha^{(1)} = \boldsymbol{\varphi}_\mathbf{g}^{(1)}$ *for all* $\alpha > 0$.

(ii) $\mathbf{U}_\alpha^{(2)}$ *satisfies*

$$(\mathrm{B}\mathbf{U}_\alpha^{(2)}, \mathrm{B}\mathbf{V}^{(2)})_{[L^2]^2} + \alpha(\mathbf{U}_\alpha^{(2)}, \mathbf{V}^{(2)})_\mathcal{H} = \alpha \int_{\partial\Omega} \mathbf{g} \cdot \mathbf{V}^{(2)} \,\mathrm{d}s \tag{4.33}$$

for all $\mathbf{V}^{(2)} \in V^\perp$.

(iii) There exists a constant $C > 0$ independent of α such that

$$\|\mathbf{U}_\alpha^{(2)}\|_\mathcal{H} \le \|\boldsymbol{\varphi}_\mathbf{g}^{(2)}\|_\mathcal{H} \le C\|\mathbf{g}\|_{[L^2(\partial\Omega)]^2}. \tag{4.34}$$

(iv) We have

$$\mathbf{U}_\alpha \to \boldsymbol{\varphi}_\mathbf{g}^{(1)} \qquad \text{strongly in} \quad \mathcal{H} \text{ as } \alpha \to 0, \tag{4.35}$$

$$\frac{1}{\sqrt{\alpha}}\mathrm{B}(\mathbf{U}_\alpha) \to 0 \qquad \text{strongly in} \quad L^2(\Omega;\mathbb{R}^2) \text{ as } \alpha \to 0. \tag{4.36}$$

Proof. ad (i): First we observe that using the weak form of (4.32) the optimality condition (4.21) is equivalent to: find $\mathbf{U}_\alpha^{(1)} \in V$ and $\mathbf{U}_\alpha^{(2)} \in V^\perp$ such that

$$\begin{aligned}(\mathrm{B}\mathbf{U}_\alpha^{(2)}, \mathrm{B}\mathbf{V}^{(2)})_{[L^2]^2} + \alpha(\mathbf{U}_\alpha^{(1)}, \mathbf{V}^{(1)})_\mathcal{H} + \alpha(\mathbf{U}_\alpha^{(2)}, \mathbf{V}^{(2)})_\mathcal{H} \\ = \alpha(\boldsymbol{\varphi}_\mathbf{g}^{(1)}, \mathbf{V}^{(1)})_\mathcal{H} + \alpha(\boldsymbol{\varphi}_\mathbf{g}^{(2)}, \mathbf{V}^{(2)})_\mathcal{H}\end{aligned} \tag{4.37}$$

for all $\mathbf{V}^{(1)} \in V$ and $\mathbf{V}^{(2)} \in V^\perp$. By testing (4.37) with $\mathbf{V}^{(2)} = 0$ we obtain

$$(\mathbf{U}_\alpha^{(1)} - \boldsymbol{\varphi}_\mathbf{g}^{(1)}, \mathbf{V}^{(1)})_\mathcal{H} = 0 \tag{4.38}$$

for all $\mathbf{V}^{(1)} \in V$. This means that $\mathbf{U}_\alpha^{(1)} - \boldsymbol{\varphi}_\mathbf{g}^{(1)} \in V^\perp$, but since by definition $\mathbf{U}_\alpha^{(1)} - \boldsymbol{\varphi}_\mathbf{g}^{(1)} \in V$ we conclude that $\mathbf{U}_\alpha^{(1)} = \boldsymbol{\varphi}_\mathbf{g}^{(1)}$.

ad (ii) & (iii): Testing (4.37) with $\mathbf{V}^{(1)} = 0$ we immediately obtain (4.33). To obtain

the bound for $\mathbf{U}_\alpha^{(2)}$ we test (4.33) with $\mathbf{V}^{(2)} = \mathbf{U}_\alpha^{(2)}$ and use the continuity of the trace to obtain that

$$\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}^2 \leq \|\mathbf{g}\|_{[L^2(\partial\Omega)]^2}\|\mathbf{U}_\alpha^{(2)}\|_{[L^2(\partial\Omega)]^2} \preceq \|\mathbf{g}\|_{[L^2(\partial\Omega)]^2}\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}. \tag{4.39}$$

Dividing by $\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}$ gives the result.

ad (iv): First note that $V^\perp$ is closed and convex and thus by the Hahn-Banach Separation Theorem $V^\perp$ is also weakly closed. By (iii) we then have that for every null-sequence (α_n), there exists $\mathbf{U}_0^{(2)} \in V^\perp$ and a subsequence (α_{n_k}) of (α_n) such that $\mathbf{U}_{\alpha_{n_k}}^{(2)} \rightharpoonup \mathbf{U}_0^{(2)}$. Therefore, using (4.33) we find

$$\begin{aligned}(\mathrm{B}\mathbf{U}_0^{(2)}, \mathrm{B}\mathbf{U}_0^{(2)})_{[L^2]^2} &= \lim_{k\to\infty} (\mathrm{B}\mathbf{U}_{\alpha_{n_k}}^{(2)}, \mathrm{B}\mathbf{U}_0^{(2)})_{[L^2]^2} \\ &\overset{(4.33)}{=} \lim_{k\to\infty} -\alpha_{n_k}(\mathbf{U}_{\alpha_{n_k}}^{(2)}, \mathbf{U}_0^{(2)})_{\mathcal{H}} + \alpha_{n_k}\int_{\partial\Omega} \mathbf{g}\cdot\mathbf{U}_0^{(2)}\,\mathrm{d}s = 0.\end{aligned} \tag{4.40}$$

Thus, $\mathrm{B}\mathbf{U}_0^{(2)} = 0$, which means $\mathbf{U}_0^{(2)} \in V$ and hence $\mathbf{U}_0^{(2)} = 0$. Since the null-sequence (α_n) was arbitrary and since the limit is unique we obtain that $\mathbf{U}_\alpha^{(2)} \rightharpoonup 0$ in $\mathcal{H}$ as $\alpha \to 0$.

Testing (4.33) with $\mathbf{V}_\alpha^{(2)} = \mathbf{U}_\alpha^{(2)}$, we obtain (using the continuity of the trace operator)

$$\lim_{\alpha\to 0}\|\mathrm{B}\mathbf{U}_\alpha^{(2)}\|_{[L^2]^2}^2 = -\lim_{\alpha\to 0}\underbrace{\alpha\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}^2}_{\text{bounded}} + \lim_{\alpha\to 0}\underbrace{\alpha\int_{\partial\Omega}\mathbf{g}\cdot\mathbf{U}_\alpha^{(2)}\,\mathrm{d}s}_{\text{bounded}} = 0. \tag{4.41}$$

This shows $\mathrm{B}\mathbf{U}_\alpha^{(2)} \to 0$ in $L^2(\Omega;\mathbb{R}^2)$ as $\alpha \to 0$. To show that $\mathbf{U}_\alpha^{(2)} \to 0$ strongly in $\mathcal{H}$ as $\alpha \to 0$, we show that its norm converges to 0. We test (4.33) again with $\mathbf{V}_\alpha^{(2)} = \mathbf{U}_\alpha^{(2)}$ and taking into account that we may assume $\alpha < 1$ we obtain the estimate

$$\begin{aligned}\alpha\|\mathrm{B}\mathbf{U}_\alpha^{(2)}\|_{[L^2]^2}^2 + \alpha\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}^2 &\leq \|\mathrm{B}\mathbf{U}_\alpha^{(2)}\|_{[L^2]^2}^2 + \alpha\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}^2 \\ &= \alpha\int_{\partial\Omega}\mathbf{g}\cdot\mathbf{U}_\alpha^{(2)}\,\mathrm{d}s\end{aligned} \tag{4.42}$$

and thus dividing by α it follows that

$$\lim_{\alpha\to 0}\|\mathbf{U}_\alpha^{(2)}\|_{\mathcal{H}}^2 \leq -\lim_{\alpha\to 0}\|\mathrm{B}\mathbf{U}_\alpha^{(2)}\|_{[L^2]^2}^2 + \lim_{\alpha\to 0}\int_{\partial\Omega}\mathbf{g}\cdot\mathbf{U}_\alpha^{(2)}\,\mathrm{d}s = 0, \tag{4.43}$$

and hence $\mathbf{U}_\alpha^{(2)}$ converges strongly to 0. Using this strong convergence and dividing (4.41) by α we obtain $\mathrm{B}(\frac{\mathbf{U}_\alpha^{(2)}}{\sqrt{\alpha}}) \to 0$ in $L^2(\Omega;\mathbb{R}^2)$ as $\alpha \to 0$.

□

4.3.2 Nearly conformal shape gradients with mixed boundary conditions

Often a part of the boundary $\partial\Omega$ is clamped, meaning that only a part of the shape can freely move. The next lemma shows that the space of clamped conformal mappings is essentially zero. The key reason for this is that nontrivial holomorphic functions only have so called isolated zeros [Gam01, p. 155–156].

Definition 4.9.
Let E be a set and let $z_0 \in E$, we say that z_0 is an isolated point *of the set E if there is $\rho > 0$ such that $|z - z_0| \geq \rho$ for all points $z \in E$ other than z_0.*

Now let $\Gamma_D \subset \partial\Omega$ be a measurable subset of the boundary with at least one point that is not isolated. We define $\Gamma := \partial\Omega \setminus \Gamma_D$ and let $H^1_\Gamma(\Omega;\mathbb{R}^2) := \{\mathbf{U} \in H^1(\Omega;\mathbb{R}^2) : \mathbf{U}|_{\Gamma_D} = 0\}$.

Lemma 4.10.
Let $\Omega \subset \mathbb{R}^2$ be an open set and let $\Gamma_D \subset \partial\Omega$ have at least one point that is not isolated. Then $\mathbf{U} \in H^1_\Gamma(\Omega;\mathbb{R}^2)$ satisfies $\mathrm{B}\mathbf{U} = 0$ a.e. on Ω if and only if $\mathbf{U} = 0$ a.e. on Ω.

Proof. It was proven in [GM78, Thm. 9] that a function that satisfies the Cauchy–Riemann equations in a distributional sense is equal to a holomorphic function almost everywhere, hence there exists a holomorphic function $\tilde{\mathbf{U}}$ with $\tilde{\mathbf{U}} = \mathbf{U}$ a.e., and in particular $\tilde{\mathbf{U}}|_{\Gamma_D} \equiv 0$. Since Γ_D has at least one point that is not isolated, [Gam01, p. 156] implies that $\tilde{\mathbf{U}} = 0$ and hence $\mathbf{U} = 0$ a.e.. □

This means that a decomposition into a holomorphic and a nonholomorphic part as in Proposition 4.8 is not possible for this case and that the solution will always be nonholomorphic. Since we cannot hope for a perfectly conformal deformation, the goal now is to control where the deformation is close to conformal and where we can accept nonconformality. To that end we introduce a weighting function $\mu \in L^\infty(\Omega)$.

Problem 4.11.
We study the following relaxation of (4.12)*: find a minimiser $\mathbf{U}_\alpha \in \mathring{H}^1(\Omega;\mathbb{R}^2)$ of*

$$\min_{\mathbf{U}\in\mathring{H}^1(\Omega;\mathbb{R}^2)} \frac{1}{2}\left(\frac{1}{\alpha}\|\mu\mathrm{B}\mathbf{U}\|^2_{[L^2]^2} + \|\,\mathrm{D}\mathbf{U}\|^2_{[L^2]^{2\times 2}}\right) - \mathrm{d}J(\Omega)[\mathbf{U}], \quad \alpha > 0. \tag{4.44}$$

Although the weight function μ is arbitrary, a choice yielding good numerical results on which we report later, is given by

$$\mu(x) := \left(\frac{\epsilon}{d_{\partial\Omega}(x) + \epsilon}\right)^{1/2}, \tag{4.45}$$

where $d_{\partial\Omega}(x) := \inf_{y\in\partial\Omega}|x-y|$ denotes the distance function associated with $\partial\Omega$ and $\epsilon > 0$ is a small parameter. The intuition behind this choice is that it enforces more holomorphicity near the boundary of Ω which is justified by the fact that grid points near the boundary are subject to larger deformations. The corresponding Euler-Lagrange equation to (4.44) reads: find $\mathbf{U} \in H^1_\Gamma(\Omega;\mathbb{R}^2)$ such that

$$\frac{1}{\alpha}(\mu\mathrm{B}\mathbf{U}, \mu\mathrm{B}\mathbf{V})_{[L^2]^2} + (\mathrm{D}\mathbf{U}, \mathrm{D}\mathbf{V})_{[L^2]^{2\times 2}} = \mathrm{d}J(\Omega)[\mathbf{V}], \tag{4.46}$$

for all $\mathbf{V} \in H^1_\Gamma(\Omega;\mathbb{R}^2)$. Existence and uniqueness of a minimiser $\mathbf{U}$ follows by the same reasoning as in the previous case.

4.3.3 $\mathrm{CR}(\alpha) + H(\mathrm{sym})$-shape gradients

As we will see in the numerical examples, deformations obtained using the standard H^1 inner product often lead to poor mesh quality. In [SS16] it was observed that a significant improvement can be achieved by calculating shape gradients with respect to the inner product induced by the symmetric bilinear form arising in the linear elasticity equations. The motivation behind this choice is to model the domain as an elastic medium and the shape derivative as a force acting on it — the gradient is then the deformation obtained by solving the linear elasticity equations.

We define the symmetric and antisymmetric parts of the derivative $\mathrm{D}\mathbf{U}$ in the usual way through

$$\mathrm{E}\mathbf{U} := \frac{1}{2}(\mathrm{D}\mathbf{U} + \mathrm{D}\mathbf{U}^\top) \text{ and } \mathrm{W}\mathbf{U} := \frac{1}{2}(\mathrm{D}\mathbf{U} - \mathrm{D}\mathbf{U}^\top). \tag{4.47}$$

We have $\mathrm{D}\mathbf{U} = \mathrm{E}\mathbf{U} + \mathrm{W}\mathbf{U}$, and in fact this decomposition is orthogonal with respect to the Frobenius inner product for matrices. Therefore, for any $\mathbf{U} \in H^1(\Omega;\mathbb{R}^2)$ it holds that $\|\mathrm{D}\mathbf{U}\|^2_{[L^2]^{2\times 2}} = \|\mathrm{E}\mathbf{U}\|^2_{[L^2]^{2\times 2}} + \|\mathrm{W}\mathbf{U}\|^2_{[L^2]^{2\times 2}}$.

Now, observe that $\|\mathrm{W}\mathbf{U}\|^2_{[L^2]^{2\times 2}} = \|\partial_x U_2 - \partial_y U_1\|^2_{L^2}$ and hence attempting to minimise this term, as is done when using the standard H^1 inner product, could be counterproductive when also trying to closely satisfy the Cauchy–Riemann equations, as they require $\partial_x U_2 + \partial_y U_1 = 0$.

In order for the symmetric part of the gradient to induce a norm on $H^1(\Omega;\mathbb{R}^2)$, we have to remove the null-space of the skew-symmetric part. We define

$$H(\mathrm{sym},\Omega) := \{\mathbf{U} \in L^2(\Omega;\mathbb{R}^2) : \mathrm{E}\mathbf{U} \in L^2(\Omega;\mathbb{R}^{2\times 2})\}. \tag{4.48}$$

By Korn's inequality [KO88, Thm. 2 and Thm. 4] we know that $H(\mathrm{sym},\Omega) = H^1(\Omega;\mathbb{R}^2)$, and that

$$\mathring{H}(\mathrm{sym},\Omega) := \{\mathbf{U} \in H(\mathrm{sym},\Omega) : \int_\Omega \mathbf{U}\,\mathrm{d}x = 0 \text{ and } \int_\Omega \partial_y U_1 + \partial_x U_2\,\mathrm{d}x = 0\} \tag{4.49}$$

is a Hilbert space when equipped with the inner product given by

$$(\mathbf{U}, \mathbf{V})_{\mathring{H}(\mathrm{sym})} := (\mathrm{E}\mathbf{U}, \mathrm{E}\mathbf{U})_{[L^2]^{2\times 2}}. \tag{4.50}$$

This bilinear form corresponds to the weak form of the linear elasticity equations for a material with Lamé parameters $\lambda = 0$ and $\mu = \frac{1}{2}$.

As before, this inner product can be augmented with a Cauchy–Riemann penalisation to improve conformality. The resulting inner product is given by

$$(\mathbf{U}, \mathbf{V})_{\mathring{H}(\mathrm{sym})+\mathrm{CR}(\alpha)} := \frac{1}{\alpha}(\mathrm{B}\mathbf{U}, \mathrm{B}\mathbf{V})_{[L^2]^2} + (\mathrm{E}\mathbf{U}, \mathrm{E}\mathbf{V})_{[L^2]^{2\times 2}}. \tag{4.51}$$

4.4 Numerical experiments

In this section we study three numerical examples. The first example considers a simply-connected domain with the entire boundary free to move. For this ideal case we will obtain deformations leading to very high mesh quality. We then study a case that is not simply-connected and for which no conformal mapping can be found. Lastly, we study the classical example of energy minimisation in slow flow.

For all problems in this section the space of deformations $\mathcal{H}$ is discretised using first order Lagrangian elements, i.e. we are optimising for the location of the vertices of the mesh.

4.4.1 Levelset example

We begin by considering a simple levelset example: the goal is to shrink a circle to a clover-like shape, as seen in Figure 4.1. Mathematically this is achieved by minimising the shape function $J(\Omega) = \int_\Omega f \,\mathrm{d}x$ where f is given by

$$\begin{aligned} f(x,y) =& (\sqrt{(x-a)^2 + by^2} - 1)(\sqrt{(x+a)^2 + by^2} - 1) \\ & (\sqrt{bx^2 + (y-a)^2} - 1)(\sqrt{bx^2 + (y+a)^2} - 1) - \varepsilon, \end{aligned} \tag{4.52}$$

where $a = 4/5$, $b = 2$ and $\varepsilon = 0.001$

This example illustrates two challenges that mesh deformation methods often face. First, the initial mesh has to be significantly compressed; if this compression is not performed evenly one can quickly obtain overlapping mesh elements. Secondly, the final mesh contains areas of high curvature; this often leads to highly stretched elements.

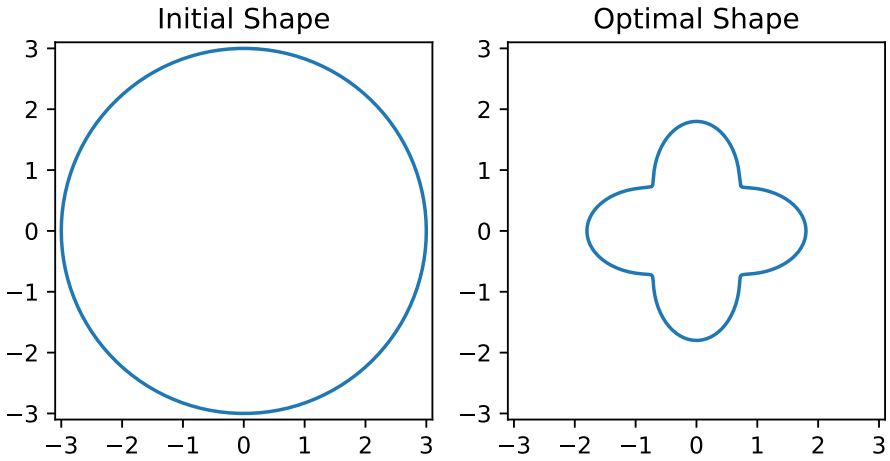


Figure 4.1: The initial shape (left) is a circle of radius 3; the minimiser of J is given by the clover-like shape on the right.

We compare four different inner products to calculate the shape gradients:

$$\begin{aligned}
(\mathbf{U},\mathbf{V})_{\mathring{H}^1} &= (\mathrm{D}\mathbf{U},\mathrm{D}\mathbf{V})_{[L^2]^{2\times 2}},\\
(\mathbf{U},\mathbf{V})_{\mathring{H}(\mathrm{sym})} &= (\mathrm{E}\mathbf{U},\mathrm{E}\mathbf{V})_{[L^2]^{2\times 2}},\\
(\mathbf{U},\mathbf{V})_{\mathrm{CR}(\alpha)+\mathring{H}^1} &= \frac{1}{\alpha}(\mathrm{B}\mathbf{U},\mathrm{B}\mathbf{V})_{[L^2]^2} + (\mathrm{D}\mathbf{U},\mathrm{D}\mathbf{V})_{[L^2]^{2\times 2}},\\
(\mathbf{U},\mathbf{V})_{\mathrm{CR}(\alpha)+\mathring{H}(\mathrm{sym})} &= \frac{1}{\alpha}(\mathrm{B}\mathbf{U},\mathrm{B}\mathbf{V})_{[L^2]^2} + (\mathrm{E}\mathbf{U},\mathrm{E}\mathbf{V})_{[L^2]^{2\times 2}}.
\end{aligned} \tag{4.53}$$

Comparison of mesh quality for small α

We begin by choosing $\alpha = 10^{-2}$; this leads to deformations that are close to perfectly conformal. In Figure 4.2 we show the optimal shape obtained from each of the four inner products in (4.53). We note that the optimisation fails for the standard $(\cdot,\cdot)_{\mathring{H}^1}$ inner product because the mesh overlaps (cf. first row in Figure 4.2). The inner product that uses the symmetric part of the gradient performs significantly better and using it the L-BFGS algorithm converges to the expected optimal shape. However, looking closer at the area of high curvature we can see that some of the elements have been significantly stretched (cf. second row in Figure 4.2). If we add the Cauchy–Riemann regularisation to either of the two inner products we can see that all triangles remain close to unilateral (cf. third and fourth row in Figure 4.2). The nearly conformal mappings achieve this by changing the size of the elements: elements are shrunk where large deformations are necessary and magnified elsewhere.



Figure 4.2: Optimal shapes obtained using the different inner products. First row: $(\cdot,\cdot)_{\mathring{H}^1}$. Second row: $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$. Third row: $(\cdot,\cdot)_{\mathrm{CR}(10^{-2})+\mathring{H}^1}$. Fourth row: $(\cdot,\cdot)_{\mathrm{CR}(10^{-2})+\mathring{H}(\mathrm{sym})}$.

We can quantify these findings by considering the distribution of the element quality $\eta(K) = h(K)/(2\rho(K))$.

For a range of η we count the fraction of cells that satisfy $\eta(K) \leq \eta$, see Figure 4.3.

The initial mesh was obtained using Gmsh [GR09] and we can see that most (98.4%) elements have a mesh quality of better than 1.2 (brown solid line). In particular, Table 4.1 shows that none of the elements have mesh quality larger than 1.5. We do not plot the mesh quality distribution for the unaugmented $(\cdot,\cdot)_{\mathring{H}^1}$ inner product as the mesh degenerates. For the $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$ inner product (blue dashed line with squares) we observe a clear loss in quality, with now only 68.5% of elements having a quality of better than 1.2. In addition, the worst element now has a quality of 3.3.

Augmenting either of the two inner products with the Cauchy–Riemann term and choosing $\alpha = 0.01$ (green lines), recovers the initial mesh quality almost entirely as can be seen both in Figure 4.3 and Table 4.1.

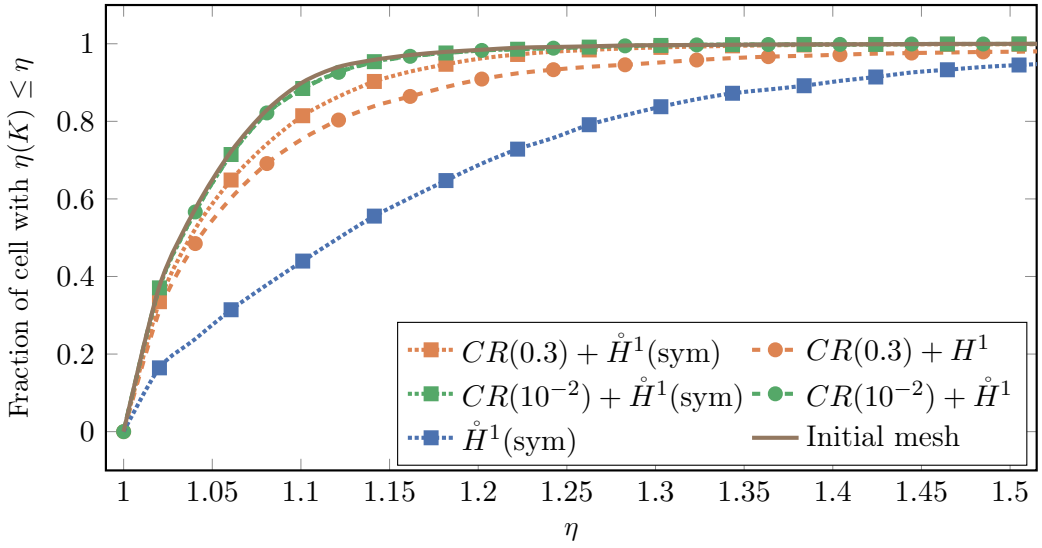


Figure 4.3: Mesh quality as measured via (4.2).

	Percentage of cells with $\eta(K)$ greater than					$\max_K \eta(K)$
	1.5	1.75	2.0	2.5	3.0	
Initial mesh	0.0%	0.0%	0.0%	0.0%	0.0%	1.5
$\mathring{H}(\mathrm{sym})$	5.6%	1.9%	0.7%	0.2%	0.1%	3.3
$CR(0.3) + \mathring{H}(\mathrm{sym})$	0.1%	0.0%	0.0%	0.0%	0.0%	1.6
$CR(10^{-2}) + \mathring{H}(\mathrm{sym})$	0.0%	0.0%	0.0%	0.0%	0.0%	1.5
$CR(0.3) + \mathring{H}^1$	2.1%	1.3%	0.8%	0.5%	0.3%	6.3
$CR(10^{-2}) + \mathring{H}^1$	0.1%	0.0%	0.0%	0.0%	0.0%	1.5

Table 4.1: The $(\cdot,\cdot)_{\mathring{H}^1}$ inner product yields a number of severely stretched elements. Adding the Cauchy–Riemann regularisation with small enough α leads to a mesh in which the worst element quality $\max_K \eta(K)$ in the final mesh is the same as compared to the initial mesh.

Behaviour for larger α

In order to achieve deformations that are nearly conformal, Proposition 4.8 suggests that we should choose α very small. However, while this does indeed lead to high mesh quality, there are two disadvantages to picking α too small. First, the linear mapping associated with the Riesz map becomes poorly conditioned and calculating the gradient becomes more computationally expensive. However, this is usually not a problem as the bottleneck for most applications lies in the calculation of the state and the adjoint equation and the calculation of the gradient is comparatively cheap. Secondly, restricting the optimisation to mappings that are almost entirely conformal means that the algorithm will need more steps to converge; this can be seen in Figure 4.4 where we plot the norm of the gradient at each L-BFGS iteration. For a fair comparison we plot the norm of the gradient in the $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$ inner product for all cases.

For $\alpha = 10^{-3}$, the gradient is almost four orders of magnitude larger than without the added Cauchy–Riemann terms after the same number of iterations. As α is increased, this difference is significantly reduced: for $\alpha = 10^{-2}$ the magnitude of the gradient differs by roughly one order of magnitude. However, Figure 4.3 and Table 4.1 suggest that such a small value of α is not necessary. A value of $\alpha = 0.3$ already results in significant improvement of the mesh quality (orange lines in Figure 4.3), especially when using the $(\cdot,\cdot)_{\mathrm{CR}(\alpha)+\mathring{H}(\mathrm{sym})}$ inner product. In Figure 4.5 we show a close-up of the high curvature area for this value of α, and we can see that the nearly conformal mappings obtained from the inner product $(\cdot,\cdot)_{\mathrm{CR}(\alpha)+\mathring{H}(\mathrm{sym})}$ result in very good mesh quality. Using the $(\cdot,\cdot)_{\mathring{H}^1}$ inner product, without any Cauchy–Riemann augmentation, the algorithm would fail due to overlapping mesh elements; the augmented version now converges but still admits visibly stretched elements. We highlight that Figure 4.4 shows no noticeable difference in speed for these two variants compared to the unaugmented inner product.

To summarise, we note that a balance must be struck between picking α very small to enforce the conformality strongly, and picking α large to ensure speed.

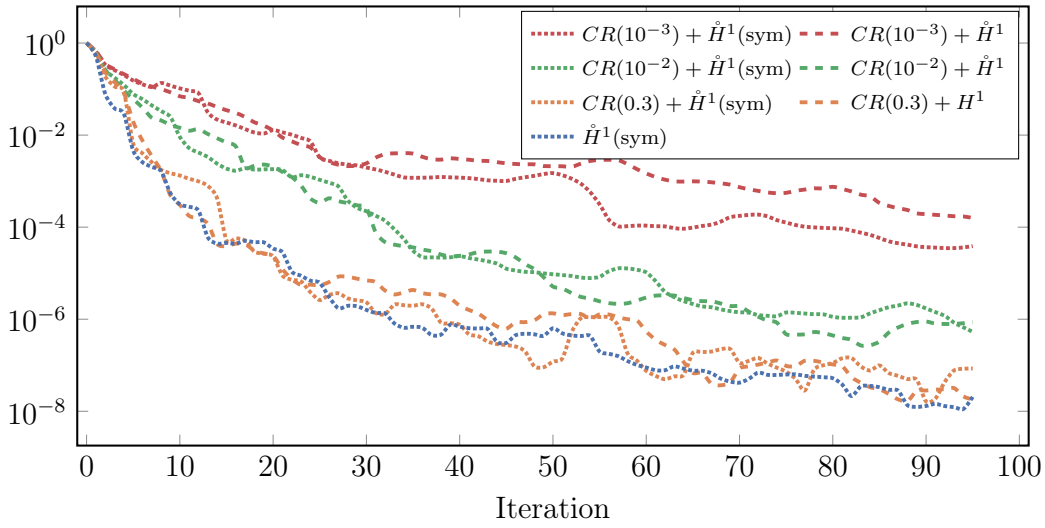


Figure 4.4: Convergence history of the L-BFGS algorithm. The algorithm is terminated after 100 steps.

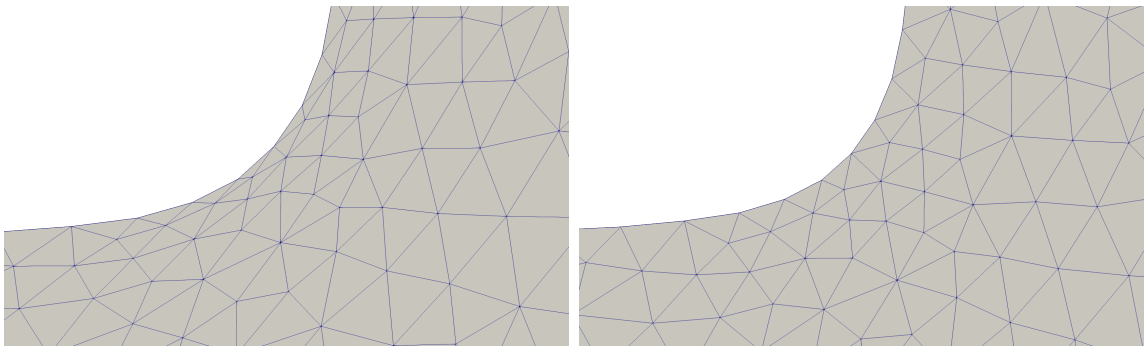


Figure 4.5: Close up of the obtained shape using the $(\cdot,\cdot)_{\mathrm{CR}(\alpha)+\mathring{H}^1}$ (left) and $(\cdot,\cdot)_{\mathrm{CR}(\alpha)+\mathring{H}(\mathrm{sym})}$ (right) inner products for $\alpha = 0.3$.

4.4.2 A negative example: Annulus deformation

We saw in the previous section that if a conformal mapping between the initial and the optimal shape exists, then our proposed inner products yield deformations resulting in good mesh quality. However, the Riemann mapping does not hold for domains that are not simply-connected. The simplest example is that of two annuli: let $A(r, R) := \{z : \ r < |z| < R\}$ denote an annulus of outer radius R and inner radius r. Schottky [Sch77], [IKO11, Theorem 1.2] proved that the annulus $A(r, R)$ can be conformally mapped to the annulus $A(r', R')$ if and only if $\frac{R}{r} = \frac{R'}{r'}$.

Regarding mesh deformation, this is a somewhat sobering result as it implies that two annuli that do not have the same radii-ratio can never be mapped to each other

without loss of mesh quality. Furthermore, in [IKO11, Theorem 1.4] it was proven that a harmonic mapping only exists if

$$\frac{R'}{r'} \geq \frac{1}{2}\left(\frac{R}{r} + \frac{r}{R}\right). \tag{4.54}$$

To illustrate this, we pick $R = R' = 1$ and $r = \frac{1}{2}$ and choose $\Omega = A(r, R)$ for the initial domain. Now for $r' \in (0, 1)$ we define $f_{r'}(\mathbf{x}) = |\mathbf{x} - 1||\mathbf{x} - r'|$ and the corresponding objective $J_{r'}(\Omega) := \int_\Omega f_{r'}(\mathbf{x})\,\mathrm{d}\mathbf{x}$. The annulus $A(r', R')$ is the global minimiser of this function and we note that solving (4.54) for r' gives that a harmonic map between the initial and the optimal shape exists for $r' \leq 0.8$.

For $r' \in (0.5, 0.8)$, all four inner products result in nearly identical deformations, as shown in Figure 4.6 for $r' = 0.75$. For $r' = 0.8$, the mesh is highly distorted and elements near the inner boundary are flattened; the methods all fail for $r' = 0.85$, aligning with the result of [IKO11].

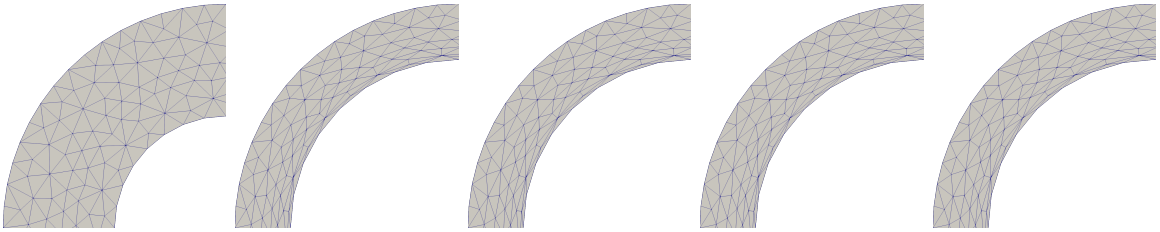


Figure 4.6: Initial mesh (left) and meshes obtained for $A(0.75, 1.0)$ using inner products $(\cdot,\cdot)_{\mathring{H}^1}$, $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$, $(\cdot,\cdot)_{\mathrm{CR}(10^{-2})+\mathring{H}^1}$, and $(\cdot,\cdot)_{\mathrm{CR}(10^{-2})+\mathring{H}(\mathrm{sym})}$ for $\alpha = 10^{-2}$.

This shows that while for simply-connected domains large deformations can be performed without loss of mesh quality, once the topology of the domain changes the problem of mesh deformation becomes significantly harder. For this specific problem the deformations can be improved by weighting the inner product to avoid the strong compression at the inner boundary, but we are not aware of any deformation method induced by an inner product that handles this problem without such modifications.

4.4.3 Energy minimisation in slow flow

We now consider a classical example in shape optimisation: the minimisation of dissipated energy in a fluid governed by Stokes' equations. Let $D \subset \mathbb{R}^2$ be a rectangular channel $(-3, +3) \times (-2, +2)$ with a circular obstacle at $(0, 0)$ of radius $1/2$ as shown in Figure 4.7. We denote the part of the channel without the obstacle by $\Omega := (-3, +3) \times (-2, +2) \setminus \overline{B_{1/2}(0)}$.

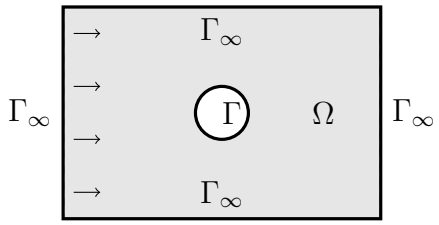


Figure 4.7: Computational domain for the BVP (4.55).

Given a far-field velocity $\mathbf{u}_\infty$, the velocity field $\mathbf{u}$ and the pressure p are then governed by the following boundary value problem:

$$\begin{aligned} -\nabla \cdot (2\mathrm{E}\mathbf{u}) + \nabla p &= 0 \qquad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \qquad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{u}_\infty \quad \text{on } \Gamma_\infty, \\ \mathbf{u} &= 0 \qquad \text{on } \Gamma. \end{aligned} \tag{4.55}$$

We choose these boundary conditions over a more natural outflow condition for comparability with existing literature on energy minimisation in Stokes flow [Pir74; SS16]. The corresponding weak form is given by: find $p \in L^2(\Omega)$ with $\int_\Omega p \,\mathrm{d}x = 0$ and $\mathbf{u} \in [H^1(\Omega)]^2$ with $\mathbf{u}|_{\Gamma_\infty} = \mathbf{u}_\infty$ and $\mathbf{u}|_\Gamma = 0$ such that

$$\int_\Omega 2\mathrm{E}\mathbf{u} : \mathrm{E}\mathbf{v} - p(\nabla \cdot \mathbf{v}) - q(\nabla \cdot \mathbf{u}) \,\mathrm{d}x = 0 \tag{4.56}$$

for all $q \in L^2(\Omega)$ and $\mathbf{v} \in [H^1(\Omega)]^2$ with $\mathbf{v}|_{\Gamma_\infty \cup \Gamma} = 0$. We discretise the state equation using a classical Taylor-Hood $[\mathbb{P}_2]^2 - \mathbb{P}_1$ discretisation [HT74]. The dissipated energy of (4.56) can be calculated by the integral

$$J(\Omega) = \int_\Omega \mathrm{E}\mathbf{u} : \mathrm{E}\mathbf{u} \,\mathrm{d}x. \tag{4.57}$$

We require the volume and the barycentre of the obstacle $\Omega_o := D \setminus \overline{\Omega}$ to remain constant; these constraints are enforced using an augmented Lagrangian method. For a more detailed treatment of the Stokes problem with these constraints we refer to [SS16].

The optimal shape for this problem is the well known ogive [Pir74] and depicted in Figure 4.8. Mesh deformation methods often struggle with this problem at the leading and trailing edge, as creating the sharp tips represents a significant deformation. Due to Lemma 4.10 we know that there is no conformal mapping between the initial and the optimal shape. Hence, we consider a weighting μ as in (4.45) (using $\epsilon = 0.01$) in order to distribute the nonconformality over the entire volume of the mesh and obtain good

mesh quality. The inner products $(\cdot,\cdot)_{\mathring{H}^1}$ and $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$ without Cauchy–Riemann augmentation both lead to overlapping meshes for this problem; we avoid this by weighting them with the same μ. When referring to the different inner products we omit μ for brevity as it is the same for all inner products considered in this section.

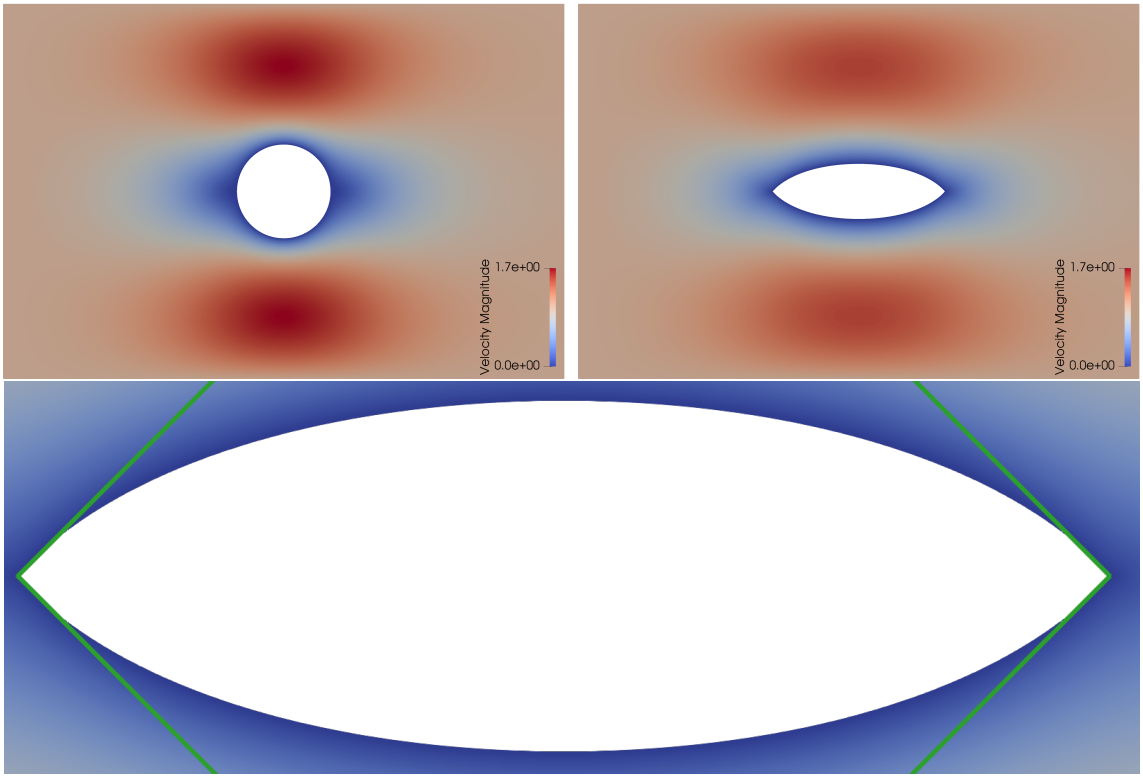


Figure 4.8: Top row: Initial shape and optimal shape for the Stokes energy minimisation problem subject to volume and barycentre constraints. Bottom: Close up of optimal shape comparing with wedges of angle 90°.

The obtained flow-fields for the initial and the optimal shape are presented in Figure 4.8. The methods all behave similarly in the interior of the mesh; however, clear differences can be observed near the tips of the obstacle. In Figure 4.9 we show the meshes near the tip obtained by using the classical $\mathring{H}^1$ and $\mathring{H}(\mathrm{sym})$ inner products both without and with Cauchy–Riemann augmentation. The mesh obtained from the $\mathring{H}^1$ inner product (top left) contains stretched elements and a highly degenerated element at the tip. Using $\mathring{H}(\mathrm{sym})$ (bottom left) results in a better mesh with minor stretch of the elements. Augmenting either of the two inner products with the proposed Cauchy–Riemann term (top right and bottom right) yields further improvement and results in a mesh without noticeable stretch. As in the levelset example, the nearly conformal mappings achieve this by shrinking the elements where the deformation

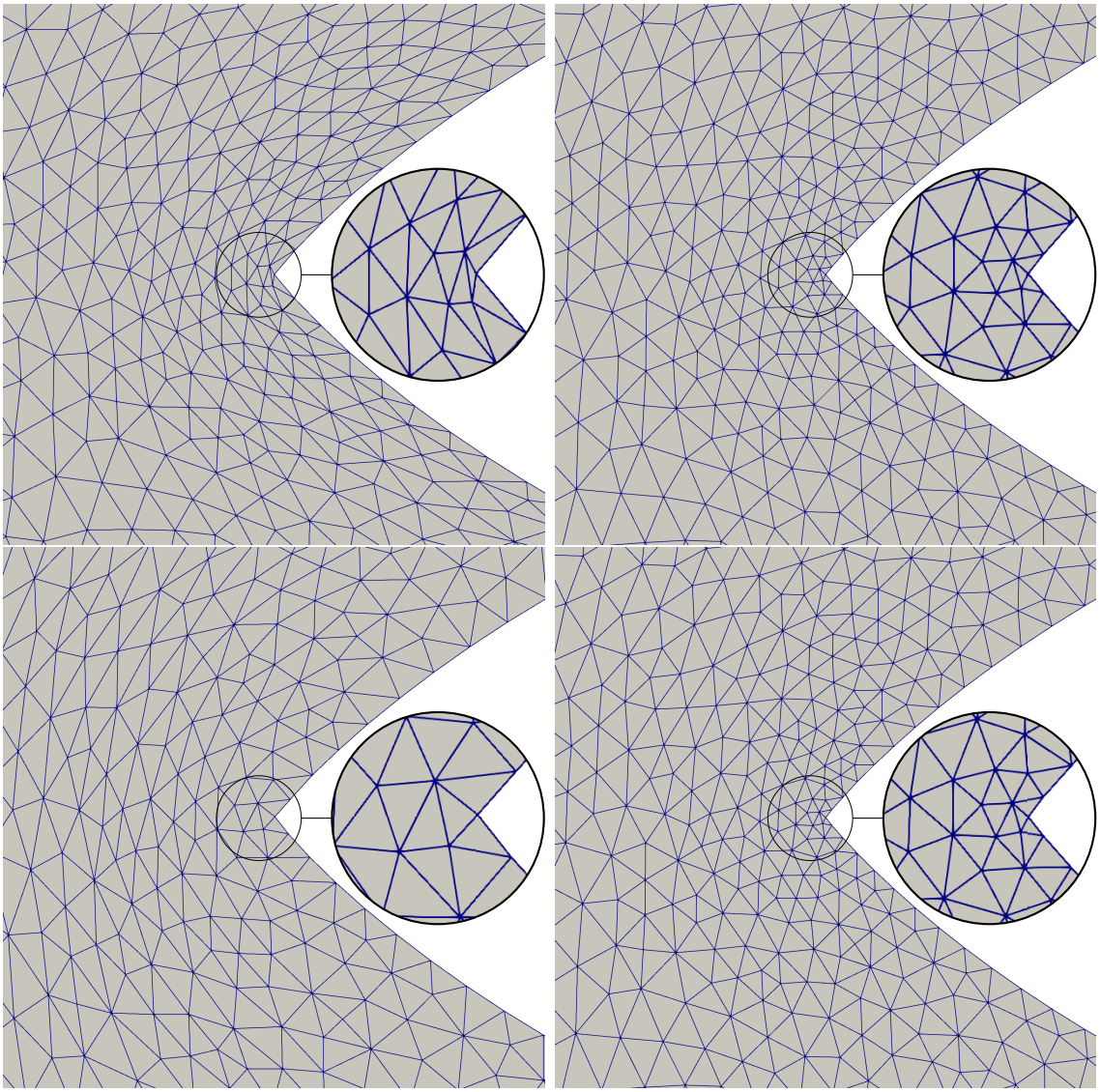


Figure 4.9: Mesh at the tip of the ogive. Augmenting an inner product with the Cauchy–Riemann term yields meshes without noticeably stretched elements. Top left: $(\cdot,\cdot)_{\mathring{H}^1}$. Top right: $(\cdot,\cdot)_{\mathrm{CR}(10^{-2})+\mathring{H}^1}$. Bottom left: $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$. Bottom right: $(\cdot,\cdot)_{\mathrm{CR}(10^{-2})+\mathring{H}(\mathrm{sym})}$.

is largest. We quantify the observed improvement by looking at the distribution of the quality measure $\eta(K)$. For a range of η we count the fraction of cells that satisfy $\eta(K) \leq \eta$, see Figure 4.10. As we know that there cannot be a conformal mapping between the initial and the optimal shape we expect some loss of quality. However, we observe that using the Cauchy–Riemann augmentation reduces the loss significantly. This becomes even more apparent when looking at the fraction of cells

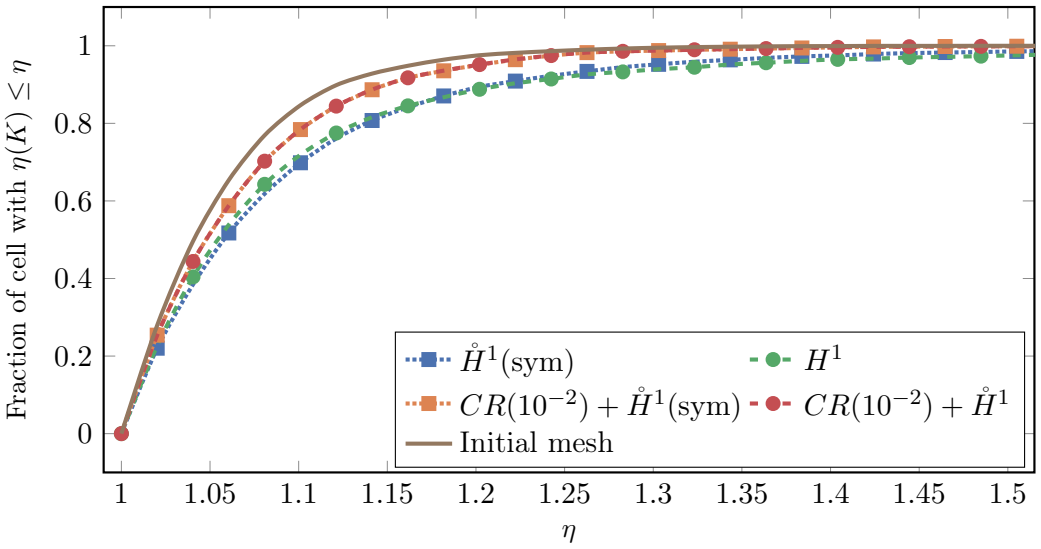


Figure 4.10: Fraction of cells K that satisfy $\eta(K) \leq \eta$ for varying η. While the deformations obtained from the regularised inner products are not able to retain the initial mesh quality, they result in a clear improvement over the nonregularised inner products.

with large values $\eta(K)$, see Table 4.2. In the original mesh the worst element quality is $\max_K \eta(K) = 1.5$. Deforming it using $(\cdot,\cdot)_{\mathring{H}^1}$ yields a mesh with $\max_K \eta(K) = 5.8$, using $(\cdot,\cdot)_{\mathring{H}(\mathrm{sym})}$ yields $\max_K \eta(K) = 2.6$. Regularizing either of these with the Cauchy–Riemann term, improves this to $\max_K \eta(K) = 1.6$. The convergence to the optimal shape is similar for all four compared inner products, as shown in Figure 4.11.

	Percentage of cells with $\eta(K)$ greater than					$\max_K \eta(K)$
	1.5	1.75	2.0	2.5	3.0	
Initial mesh	0.0%	0.0%	0.0%	0.0%	0.0%	1.5
$\mathring{H}(\mathrm{sym})$	1.5%	0.4%	0.1%	0.0%	0.0%	2.6
$CR(10^{-2}) + \mathring{H}(\mathrm{sym})$	0.1%	0.0%	0.0%	0.0%	0.0%	1.6
$\mathring{H}^1$	2.5%	0.9%	0.4%	0.1%	0.0%	5.8
$CR(10^{-2}) + \mathring{H}^1$	0.1%	0.0%	0.0%	0.0%	0.0%	1.6

Table 4.2: The $(\cdot,\cdot)_{\mathring{H}^1}$ inner product yields a number of severely stretched elements. Adding the Cauchy–Riemann regularisation leads to a mesh in which the worst element quality $\max_K \eta(K)$ in the final mesh is only slightly larger than in the initial mesh.

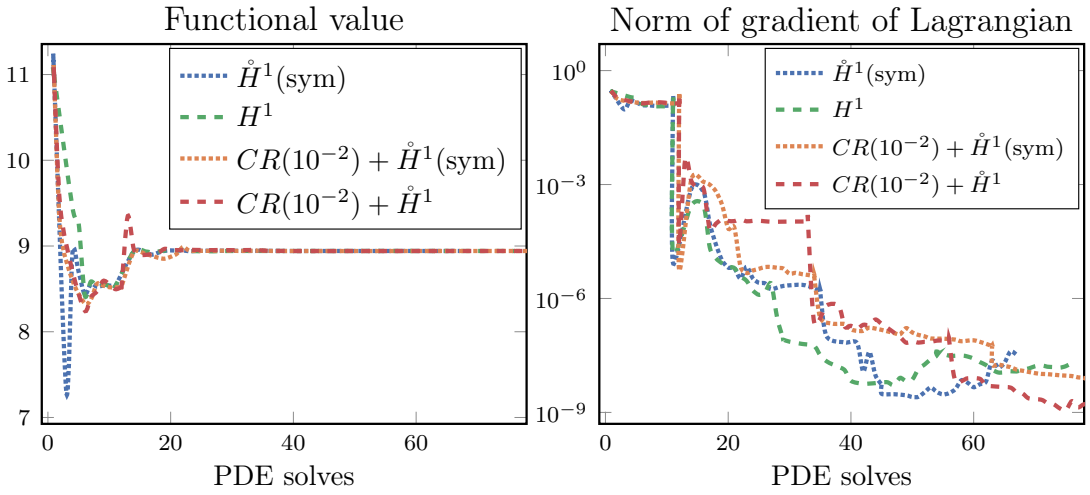


Figure 4.11: The convergence is similar for all four considered inner products. The optimisation was stopped after 5 iterations of the augmented Lagrangian method.

4.5 Discussion of the three dimensional case

We consider only two dimensional problems in this chapter, as the Cauchy–Riemann equations are an inherently two dimensional concept. However, we briefly outline how one could attempt to extend the work presented here to higher dimensions. In general, a mapping $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ is angle preserving if it satisfies the system of nonlinear equations [IM01, Section 2.1]

$$\mathrm{B}^d\mathbf{T} := \mathrm{D}\mathbf{T}^\top \mathrm{D}\mathbf{T} - (\det(\mathrm{D}\mathbf{T}))^{2/d}\mathbf{I}_d = 0. \tag{4.58}$$

One can show that this equation is equivalent to the Cauchy–Riemann equations when $d = 2$, which is why these equations are also known as the Cauchy–Riemann system [IM01, p. 34]. For $d \geq 3$ however, this equation is highly nonlinear and furthermore overdetermined, which in turn implies that the exact solutions for (4.58) can be parametrised by finitely many parameters. In addition, the powerful statement of the Riemann mapping theorem does not extend to higher dimensions.

While this means that we cannot expect results as good as in two dimensions, we make some remarks how one could use the Cauchy–Riemann system in three dimensions. One approach is to linearise B^d around $\mathbf{T} = \mathbf{Id}$, resulting in

$$\hat{\mathrm{B}}^d\mathbf{X} := (\mathrm{D}\mathbf{X}^\top + \mathrm{D}\mathbf{X}) - \frac{2}{d}\operatorname{div}(\mathbf{X}). \tag{4.59}$$

One could then proceed as previously and augment a given inner product $(\cdot,\cdot)_H$ on $H^1(\Omega;\mathbb{R}^3)$ and define

$$(\mathbf{U},\mathbf{V})_{H+\mathrm{CR}(\alpha)} := (\mathbf{U},\mathbf{V})_H + \frac{1}{\alpha}(\hat{\mathrm{B}}^3\mathbf{U},\hat{\mathrm{B}}^3\mathbf{V})_{L^2}. \tag{4.60}$$

When the initial guess is close to the optimal shape we can then hope that the deformation is close to conformal. Alternatively, one could augment the shape functional itself, e.g. by considering

$$\hat{\mathcal{J}}_\Omega(\mathbf{X}) = \mathcal{J}_\Omega(\mathbf{X}) + \frac{1}{\alpha}\left\|\mathrm{B}^d(\mathbf{Id}+\mathbf{X})\right\|_{L^2}^2. \tag{4.61}$$

However, apart from the rare case that a conformal mapping between the initial and the optimal shape of $\mathcal{J}_\Omega$ exists, the minimisers of $\mathcal{J}_\Omega$ and $\hat{\mathcal{J}}_\Omega$ do not coincide. We do not explore these ideas further at this stage.

Code availability

The code for the numerical experiments from this chapter is available at `https://github.com/florianwechsung/ThesisNumerics`. For reproducibility, all major Firedrake components as well as the code used to run these examples have been archived on Zenodo [Fir19; Wec19c; Wec19a].

Part II

Large scale solvers for the Navier–Stokes equations

Chapter 5

Augmented Lagrangian preconditioning for the stationary incompressible Navier–Stokes equations

Many problems in science and engineering involve the simulation of the flow of a fluid or gas. Examples include the flow of air around aeroplanes or cars, the flow of blood, or the flow of water in a pipe or around a boat. Frequently these simulations naturally give rise to shape optimisation problems: an engineer who solves for the airflow around a race car often does this in order to find a design that improves a quantity of interest such as the drag or downforce.

The Reynolds number, defined as $\mathrm{Re} = \frac{UL}{\nu}$ where U is the characteristic velocity, L is the characteristic length scale of the flow, and $\nu > 0$ is the kinematic viscosity of the fluid, is an important dimensionless number governing the nature of the flow. At very low Reynolds number, the flow is laminar and its simulation is well understood [SW94]; as the Reynolds number increases, the flow becomes more complex and the simulation difficulty increases accordingly.

We consider the stationary, incompressible, Newtonian Navier–Stokes equations. Given a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, a decomposition of the boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, a forcing term $\mathbf{f}$ and boundary data $\mathbf{g}$, their strong form is: find a velocity $\mathbf{u}$ and pressure p such that

$$-\nabla \cdot (2\nu \mathrm{E}\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = f \qquad \text{in } \Omega, \tag{5.1a}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{5.1b}$$

$$\mathbf{u} = \mathbf{g} \qquad \text{on } \Gamma_D, \tag{5.1c}$$

$$2\nu \mathrm{E}\mathbf{u} \cdot \mathbf{n} = p\mathbf{n} \qquad \text{on } \Gamma_N, \tag{5.1d}$$

where $\mathrm{E}\mathbf{u} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$. In the limit of short length scales, low velocity, or high viscosity the advection term can be dropped and one obtains the Stokes equations, which we already studied in Section 4.4.3:

$$-\nabla \cdot (2\nu \mathrm{E}\mathbf{u}) + \nabla p = f \qquad \text{in } \Omega, \tag{5.2a}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{5.2b}$$

$$\mathbf{u} = \mathbf{g} \qquad \text{on } \Gamma_D, \tag{5.2c}$$

$$2\nu \mathrm{E}\mathbf{u} \cdot \mathbf{n} = p\mathbf{n} \qquad \text{on } \Gamma_N. \tag{5.2d}$$

The viscosity ν is often omitted as it can be absorbed into the solution.

The importance of the Stokes and Navier–Stokes equations has motivated a great deal of research on algorithms for their solution; for a general overview of the field, see the textbooks of Turek [Tur99], Elman, Silvester & Wathen [ESW14], or Brandt [BL11].

Remark 5.1. *It is well known that stationary solutions to the Navier–Stokes equations cease to be stable for large Reynolds number (roughly at* $\mathrm{Re} \sim 10^2$–10^4*, depending on the geometry). However, even then a fast solver for the stationary equations is valuable for implicit timestepping schemes with large timesteps or for bifurcation analysis.*

For boundary data $\mathbf{g} \in H^{1/2}(\Gamma_D; \mathbb{R}^d)$, we define

$$V_{\mathbf{g}} := \{\mathbf{v} \in H^1(\Omega; \mathbb{R}^d) : \mathbf{v}|_{\Gamma_D} = \mathbf{g}\}. \tag{5.3}$$

The weak form of (5.1) that we consider is: find $(\mathbf{u}, p) \in V_{\mathbf{g}} \times Q$ such that

$$(2\nu \mathrm{E}\mathbf{u}, \mathrm{E}\mathbf{v}) + (\mathbf{u} \cdot \nabla\mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in V_0, \tag{5.4}$$

$$-(q, \nabla \cdot \mathbf{u}) = 0 \qquad \forall q \in Q, \tag{5.5}$$

where $(\cdot, \cdot)$ denotes the L^2 inner product, and $\langle \mathbf{f}, \mathbf{v} \rangle$ denotes the dual pairing of $\mathbf{f} \in H^{-1}(\Omega; \mathbb{R}^d)$ and $\mathbf{v}$. If $|\Gamma \setminus \Gamma_D| = 0$, then a suitable space Q for the pressure is given by $\mathring{L}^2 := \{q \in L^2 : \int_\Omega q \, \mathrm{d}x = 0\}$, otherwise we set $Q := L^2(\Omega)$.

These equations become highly nonlinear as ν decreases. Given an initial guess $\mathbf{u}_0$ that satisfies the boundary conditions, Newton's method applied to these equations results in linear systems of the form: find $(\mathbf{u}, q) \in V_0 \times Q$ such that

$$(2\nu \mathrm{E}\mathbf{u}, \mathrm{E}\mathbf{v}) + (\mathbf{u} \cdot \nabla\mathbf{w}, \mathbf{v}) + (\mathbf{w} \cdot \nabla\mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}^{(1)}, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in V_0, \tag{5.6}$$

$$-(q, \nabla \cdot \mathbf{u}) = \langle f^{(2)}, q \rangle \qquad \forall q \in Q, \tag{5.7}$$

for appropriate right hand sides $\mathbf{f}^{(1)}$ and $f^{(2)}$ and where $\mathbf{w}$ is the velocity solution in the last iteration. An alternative to Newton's method is the Picard iteration, which corresponds to dropping one of the advection terms, i.e.

$$(2\nu \mathrm{E}\mathbf{u}, \mathrm{E}\mathbf{v}) + (\mathbf{w} \cdot \nabla \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}^{(1)}, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in V_0, \tag{5.8}$$

$$-(q, \nabla \cdot \mathbf{u}) = \langle f^{(2)}, q \rangle \qquad \forall q \in Q. \tag{5.9}$$

Picard iteration has two advantages compared to Newton's method: first, it can be shown to be globally convergent for large enough viscosity [Kar82, Theorem 4.3], and second, the bilinear form involving $\mathbf{u}$ and $\mathbf{v}$ is coercive for any viscosity ν. While Newton's method does not exhibit these properties, it yields quadratic convergence close to the true solution [GR86, §4.6.3]. In this thesis we will consider only the equations arising from Newton iteration.

5.1 Preconditioning Strategies

After discretisation using suitable spaces $V_h \subset V_0$, $Q_h \subset Q$ (discussed in detail in Chapters 6 and 7), at each Newton step a nonsymmetric linear system of saddle point type must be solved:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ c \end{bmatrix}, \tag{5.10}$$

where A is the discrete linearised momentum operator, B^T is the discrete gradient operator, B is the discrete divergence operator, and $\mathbf{u}$ and p are the updates to the coefficients for velocity and pressure respectively. The solution of these equations is complicated by several features of the system: first, it is nonsymmetric due to the advection term, and second, it is indefinite due to its saddle-point like structure.

One strategy to solve these systems is to employ a monolithic multigrid iteration on the entire system with a suitable coupled relaxation method, such as the algorithms of Vanka [Van86] or Braess & Sarazin [BS97]. Vanka iteration works well for moderate Reynolds numbers [Tur99], but the iteration counts have been observed to increase significantly once the Reynolds number becomes large [BO06].

An alternative approach to solving (5.10) is to build preconditioners based on block factorisations [MGW00; Ips01; BGL05; ESW14; Wat15]:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}^{-1} = \begin{bmatrix} I & -A^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix} \tag{5.11}$$

where S is the *Schur complement* given by

$$S = -BA^{-1}B^T. \tag{5.12}$$

It was shown in [MGW00], that in fact the diagonal preconditioner

$$P = \begin{bmatrix} A^{-1} & 0 \\ 0 & -S^{-1} \end{bmatrix} \tag{5.13}$$

results in a matrix PA with only 3 distinct eigenvalues. While the matrix S is usually dense and hence not formed exactly, this motivates preconditioners of the form (5.11) or (5.13) where A^{-1} and S^{-1} are replaced by approximations. Cheap approximations have to be devised on a PDE specific basis.

For the Stokes equations, the Schur complement is spectrally equivalent to the viscosity-weighted pressure mass matrix [SW94]. A mesh independent preconditioner can be obtained by approximating the inverse of the top-left block with an AMG cycle and the Schur-complement with the diagonal of the pressure mass matrix. We note that this preconditioner can also be derived using a functional analytic approach as it is spectrally equivalent to the Riesz-map in the appropriate function space [MW11; GHS14; HS15]. For the Navier–Stokes equations this choice still yields mesh independent convergence and is effective for very small Reynolds numbers, but the convergence deteriorates with Reynolds number [ES96; ESW14].

The pressure convection-diffusion (PCD) approach [KLW02] constructs an auxiliary convection-diffusion operator on the pressure space, and hypothesises that a certain commutator is small. This yields an approximation to the Schur complement inverse that involves the inverse of the Laplacian on the pressure space, the application of the auxiliary convection-diffusion operator, and the inverse of the pressure mass matrix. The least-squares commutator (LSC) approach [Elm+06] is based on a similar idea, but derives the commutator algebraically. Both of these approaches yield mesh independent iteration counts and perform well for moderate Reynolds numbers. However, as the Reynolds number increases, the iterations counts grow at a rate between $\sim \mathrm{Re}^{1/3}$ and $\sim \mathrm{Re}^{1/2}$ depending on the exact variant of the preconditioner and problem under consideration [KLW02]. One might ask whether this growth is due to the Schur-complement approximation or simply because AMG becomes a poor approximation for A^{-1}. It was demonstrated in [ESW14, Table 9.3] that the ideal versions of the preconditioners, i.e. when all solves are performed exactly, admit similar growth of iteration counts. This suggests that it is the approximation of the Schur-complement that degrades as the Reynolds number increases.

5.2 Augmented Lagrangian preconditioning

In [Ols02; BO06] Benzi & Olshanskii propose a different approach: they modify the equations in a way that does not change their solution, but forces the Schur-complement to be close to an operator that can easily be inverted. This approach is known as augmented Lagrangian or grad-div stabilisation [OR03; BO06; GGV03; OB08; BO11; FG83; Kob95].

The discrete variant of this approach replaces (5.10) with

$$\begin{bmatrix} A+\gamma B^T W B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{b}+\gamma B^T W c \\ c \end{bmatrix}, \tag{5.14}$$

for some matrix W. We see immediately that a solution to (5.10) is also a solution to (5.14) and vice versa. The Schur-complement however, does change.

Lemma 5.2.

The Schur-complement of the matrix in (5.14) *satisfies*

$$S_\gamma^{-1} = S^{-1} - \gamma W \tag{5.15}$$

where S^{-1} *is the Schur-complement of the matrix in* (5.10)*.*

Proof. We recall the Sherman-Morrison-Woodbury identity [Hag89]: for matrices A, U, C and V such that A, C and $C^{-1}+VA^{-1}U$ are invertible, it holds that

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}. \tag{5.16}$$

We use this identity twice to obtain

$$\begin{aligned} S_\gamma^{-1} &= -(B(A+\gamma B^T W B)^{-1}B^T)^{-1} \\ &= -(B(A^{-1} - A^{-1}B^T(\frac{1}{\gamma}W^{-1} + BA^{-1}B^T)^{-1}BA^{-1})B^T)^{-1} \\ &= -(\underbrace{BA^{-1}B^T}_{-S} - \gamma BA^{-1}B^T(W^{-1}+BA^{-1}B^T)^{-1}BA^{-1}B^T)^{-1} \\ &= (S + S(\frac{1}{\gamma}W^{-1} - S)^{-1}S)^{-1} \\ &= (S^{-1} - S^{-1}S(\frac{1}{\gamma}W^{-1} - S + SS^{-1}S)^{-1}SS^{-1}) \\ &= S^{-1} - \gamma W. \end{aligned} \tag{5.17}$$

□

Given any approximation $\tilde{S}^{-1} \approx S^{-1}$, we approximate the new Schur complement by $\tilde{S}_\gamma^{-1} = \tilde{S}^{-1} - \gamma W$. In the previous section we remarked that the approximation $S \approx -\nu M_p$, where M_p is the pressure mass matrix, results in a mesh independent preconditioner. This suggests the choice $W = M_p^{-1}$, as then

$$S_\gamma^{-1} = S^{-1} - \gamma M_p^{-1} \approx -(\nu + \gamma) M_p^{-1}. \tag{5.18}$$

Clearly the approximation becomes better as $\gamma \to \infty$ *for fixed mesh size and viscosity*, in the sense that $\left|\left|\left| S_\gamma^{-1} + (\nu+\gamma) M_p^{-1} \right|\right|\right| / \left|\left|\left| S_\gamma^{-1} \right|\right|\right| \to 0$. In [BO06; Ols02] the eigenvalues of the preconditioned system are studied for the above choice of W and it is shown that the choice $\gamma \sim \nu^{-1}$ yields uniformly bounded eigenvalues in ν and h.

However, it is well known that the eigenvalues of a matrix do not characterise the convergence of GMRES for a linear system [GPS96]. Instead, it is sufficient to bound the field-of-values of the preconditioned system [Sta97, Theorem 3.2], [EE01, Corollary 6.2], [OT14, §1.3]. This analysis was performed in [BO11] for the augmented Lagrangian preconditioner for the Oseen problem, using general results of Loghin & Wathen [LW04]. One of the key ingredients in this analysis is that the momentum operator is coercive with constant ν. They use this to prove that the choice $\gamma \sim \nu^{-1}$ results in an optimal preconditioner (assuming exact solves of the momentum block). However, the momentum operator of the Newton linearisation of (5.1) is only coercive for $\nu > \nu_0$ for some problem-dependent ν_0 [GR86, p. 300]. Fortin & Glowinski remark [FG83, p. 85] that this is typically a very restrictive condition: for ν large enough so that $\nu > \nu_0$ the Stokes approximation itself is usually adequate. This proof strategy would therefore require significant extension to apply to the Newton linearisation considered here.

In practice, Benzi & Olshanskii [BO06] observe that a constant choice of γ yields mesh independent and nearly Reynolds number independent results. We will develop solvers that are robust to large values of γ, and hence we simply choose γ large. In the experiments of Section 6.4, we take the value $\gamma = 10^4$, to match the largest Reynolds number considered. Considering an implementation of this scheme, we note that a triple matrix product like $B^T W B$ is both expensive to compute and store. However, for $W = M_p^{-1}$ adding $\gamma B^T M_p^{-1} B$ to the linear system corresponds to augmenting the weak form with a term

$$\gamma c(\mathbf{u}, \mathbf{v}) = \gamma \int_\Omega \Pi_{Q_h}(\nabla \cdot \mathbf{u}) \Pi_{Q_h}(\nabla \cdot \mathbf{v}) \, \mathrm{d}x, \tag{5.19}$$

where Π_{Q_h} is the projection onto Q_h. When the pressure space consists of discontinuous functions, then this projection can be evaluated cheaply and the integral can be

assembled directly. We will later also study a discretisation that satisfies $\mathrm{div}(V_h) \subset Q_h$, in which case the projection reduces to the identity.

Remark 5.3. *The same augmentation but without the projection onto Q_h is also known as grad-div stabilisation as it corresponds to the weak form of $-\gamma\nabla\nabla\cdot\mathbf{u}$. This stabilisation has several advantageous properties. For example, it significantly improves the pressure-robustness of discretisations where the incompressibility constraint is not enforced pointwise [OR03; HR12; Joh+17]. It also arises in other contexts in the numerical analysis of* (5.1)*. For example, Boffi & Lovadina [BL97] showed that the addition of the term $h^{-1/2}(\nabla\cdot\mathbf{u}, \nabla\cdot\mathbf{v})_{L^2(\Omega)}$ to the weak form of the $[\mathbb{P}_2]^2-\mathbb{P}_0$ discretisation of* (5.1) *improves its convergence order. It also arises in the iterated penalty [Tem68; BS08] and artificial compressibility [Cho67] methods for the Stokes and Navier–Stokes equations.*

To demonstrate the efficacy of the approach, we show iteration counts for the ideal version of the augmented Lagrangian preconditioner, that is when the top-left block is solved exactly. The iteration counts are shown in Table 5.1: as γ is increased, the preconditioner becomes a very good approximation of the true inverse of the Jacobian. To summarise, the augmented Lagrangian approach adds little cost to the assembly of the system but improves the approximation of the Schur-complement significantly. However, as we discuss in the next section, the price to be paid for this improvement

γ	Reynolds number				
	10	100	1000	5000	10000
0	13.00	25.33			
1	7.50	5.33	5.00	4.00	3.50
10^2	2.00	2.00	1.67	1.50	1.50
10^4	1.50	1.00	1.00	1.00	1.00

Table 5.1: Average number of outer Krylov iterations per Newton step for a two dimension lid-driven cavity problem on a 32×32 uniform grid using the ideal version of the augmented Lagrangian preconditioner. For $\gamma = 0$ the solver exceeds the limit of 100 iterations at $\mathrm{Re} = 300$.

is that the top-left block becomes much harder to solve.

5.3 Solving the top-left block

Let us denote

$$A_\gamma := A + \gamma B^T M_p^{-1} B. \tag{5.20}$$

While it is understood how to solve linear systems involving A, the added augmented Lagrangian term introduces extra difficulty. The matrix $B^T M_p^{-1} B$ has a large nullspace, consisting of all vector fields with vanishing projected divergence. This implies that as γ becomes large, A_γ becomes nearly singular and standard iterative solvers for A become ineffective at solving systems involving A_γ.

We demonstrate this in Table 5.2, where we apply standard geometric multigrid with Jacobi smoothing [MM16] and algebraic multigrid [FY02] to the A_γ system. We clearly see that convergence degrades quickly as the ratio γ/ν is increased.

Refinements	Degrees of freedom	γ/ν							
		0	1	10	10^2	10^3	10^4	10^6	10^8
Geometric multigrid									
1	578	5	5	7	20	>100	>100	>100	>100
2	2 178	5	5	7	24	>100	>100	>100	>100
3	8 450	5	5	7	26	>100	>100	>100	>100
4	33 282	5	5	7	27	>100	>100	>100	>100
Algebraic multigrid									
1	578	7	7	11	28	57	87	>100	>100
2	2 178	8	7	11	29	74	>100	>100	>100
3	8 450	7	7	12	34	93	>100	>100	>100
4	33 282	8	8	14	39	>100	>100	>100	>100

Table 5.2: Iteration counts for a system involving the top-left block A_γ only. Excerpt from Table 6.2 in Section 6.4.2.

A key insight of [BO06; OB08] was that a specialised multigrid algorithm could be built for A_γ with convergence that does not deteriorates as $\gamma/\nu \to \infty$ by applying the seminal work of Schöberl [Sch99b; Sch99a], who developed parameter robust multigrid methods in the context of nearly incompressible elasticity. The method involves a custom smoother that effectively treats errors in the kernel of the discrete divergence term and a specialised prolongation operator whose continuity constant is independent of γ and ν. Using this scheme for the top-left block, the augmented Lagrangian preconditioner exhibits outer iteration counts that grow only very slowly with Reynolds number [BO06]. However, it is described as difficult to implement [HBH10; BOW11], and so despite promising results in [BO06; OB08], to the best of the author's knowledge, all of the works that use augmented Lagrangian preconditioning and the Schur complement approximation (5.18) employ either matrix factorisation as the inner solver [NW07; RVS08; BB10; HNC11; HR12] or a block-triangular approximation to

A_γ [BOW11; HBH10; BO11; HVK18]. This block-triangular approximation decouples linear systems involving A_γ into d scalar anisotropic advection–diffusion problems, which may be solved with algebraic multigrid techniques. However, this simplicity comes at a price; the scheme is much more sensitive to the choice of γ, and its convergence deteriorates somewhat as the Reynolds number increases [BOW11].

To preserve both robustness and scalability, we choose to implement the geometric multigrid scheme for the full A_γ matrix. In the next section we will review the theory developed by Schöberl for nearly singular problems and describe the requirements for the smoothing and prolongation operators. In [Sch99a; Sch99b] a full convergence proof for the W-cycle and the variable V-cycle is given. However, the nearly singular nature and a potential nonnestedness complicates the analysis and the final result is somewhat weaker than the usual textbook multigrid efficiency. Convergence is proven for a sufficiently high number of smoothing iterations and the bounds on the contraction rate in terms of smoothing iterations are weaker than usual. We do not reproduce the full multigrid convergence proof here, instead we will illustrate the requirement for a parameter robust smoother and prolongation on the basis of an additive two level method as in [Sch99b, Section 3.1.2]. In Chapters 6 and 7 we will then construct smoothing and prolongation operators for specific discretisations and prove that they satisfy the necessary properties for a robust scheme.

5.4 Subspace correction methods for nearly singular problems

Preconditioners for problems that are given as the sum of a positive definite and a positive semidefinite operator have been subject to extensive study, in particular in the context of solving problems in $H(\mathrm{div})$ and $H(\mathrm{curl})$, where problems involving bilinear forms $(\mathbf{u},\mathbf{v}) \mapsto (\mathbf{u},\mathbf{v}) + \gamma(\nabla\cdot\mathbf{u}, \nabla\cdot\mathbf{v})$ and $(\mathbf{u},\mathbf{v}) \mapsto (\mathbf{u},\mathbf{v}) + \gamma(\nabla\times\mathbf{u}, \nabla\times\mathbf{v})$ arise frequently [EW92; VW92; HT97; AFW97; AFW00; HX07]. Ewing & Wang and Vassilevski & Wang studied Schwarz methods for the $H(\mathrm{div})$ case and realised the importance of subspace decompositions that respect the kernel of the divergence. General formulations of this idea were developed by Schöberl [Sch99a; Sch99b; Sch98] (for additive relaxation) and Lee et al. [Lee+07; LWC09] (for multiplicative relaxation).

Both Schöberl and Lee et al. only consider the case of a symmetric system and the author is not aware of any literature that has treated nonsymmetric nearly singular problems. For this reason, the following analysis will consider the Stokes problem only. We will demonstrate in the numerical experiments that the resulting smoother

performs remarkably well even when the advection terms are included. Ignoring advection, the problem associated with the top-left block can be written as

$$a_{h,\gamma}(\mathbf{u},\mathbf{v}) := a(\mathbf{u},\mathbf{v}) + \gamma c_h(\mathbf{u},\mathbf{v}), \tag{5.21}$$

where

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) &:= 2\nu(\mathrm{E}\mathbf{u},\mathrm{E}\mathbf{v}), \\ c_h(\mathbf{u},\mathbf{v}) &:= (\Pi_{Q_h}(\nabla\cdot\mathbf{u}),\Pi_{Q_h}(\nabla\cdot\mathbf{v})). \end{aligned} \tag{5.22}$$

We define the operator $A_{h,\gamma} : V_h \to V_h^*$ by

$$\langle A_{h,\gamma}\mathbf{u},\mathbf{v}\rangle := a_{h,\gamma}(\mathbf{u},\mathbf{v}), \tag{5.23}$$

and we drop the subscript γ to denote the case without augmentation, i.e.

$$\langle A_h\mathbf{u},\mathbf{v}\rangle := a(\mathbf{u},\mathbf{v}). \tag{5.24}$$

In addition, without loss of generality we can assume $\nu = 1$ (if not, we divide the equation by ν and then follow the analysis below with $\hat{\gamma} = \gamma/\nu$).

Many smoothers commonly used in multigrid can be expressed as so called subspace correction methods. We consider a decomposition

$$V_h = \sum_i V_i \tag{5.25}$$

where the sum is not necessarily a direct sum. For each subspace i we denote the natural inclusion by $I_i : V_i \to V_h$ and we define the restriction A_i of $A_{h,\gamma}$ onto V_i as

$$\langle A_i\mathbf{u}_i,\mathbf{v}_i\rangle := \langle A_{h,\gamma}I_i\mathbf{u}_i, I_i\mathbf{v}_i\rangle \qquad \text{for all } \mathbf{u}_i,\mathbf{v}_i \in V_i. \tag{5.26}$$

The additive Schwarz preconditioner associated with the subspace decomposition $\{V_i\}$ is then defined by the action of its inverse:

$$D_{h,\gamma}^{-1} = \sum_i I_i A_i^{-1} I_i^*. \tag{5.27}$$

The method is also known as the parallel subspace correction method [Xu92]. A standard result in the theory of subspace correction methods ([Xu01, Eqn. (4.11)], [Sch99b, Theorem 4.1]) is that

$$\|\mathbf{u}_h\|_{D_{h,\gamma}}^2 = \inf_{\substack{\mathbf{u}_i\in V_i \\ \sum_i \mathbf{u}_i = \mathbf{u}_h}} \sum_i \|\mathbf{u}_i\|_{A_i}^2. \tag{5.28}$$

In order to prove spectral equivalence of $D_{h,\gamma}$ and $A_{h,\gamma}$, we want to obtain a bound of the form

$$c_1 D_{h,\gamma} \le A_{h,\gamma} \le c_2 D_{h,\gamma}, \tag{5.29}$$

where $M \leq N$ means that $\|\mathbf{u}\|_M \leq \|\mathbf{u}\|_N$ for all $\mathbf{u}$. The number of iterations required by the conjugate gradient method for $A_{h,\gamma}$ preconditioned by $D_{h,\gamma}$ then behaves like $\sqrt{c_2/c_1}$ [Wat15, eqn. (2.18)].

Let us study the second inequality in (5.29) first and give a bound for c_2. For i and j we define

$$g_{ij} = \begin{cases} 1, & \text{if } \exists \mathbf{v}_i \in V_i, \mathbf{v}_j \in V_j : |\operatorname{supp}(\mathbf{v}_i) \cap \operatorname{supp}(\mathbf{v}_j)| > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{5.30}$$

indicating interaction between subspaces V_i and V_j. The overlap is then defined as $N_O = \max_i \sum_j g_{ij}$. Let $\mathbf{u}_h = \sum_i I_i \mathbf{u}_i$ be a splitting of $\mathbf{u}_h \in V_h$. Then

$$\begin{aligned} \|\mathbf{u}_h\|^2_{A_{h,\gamma}} &= \sum_{ij} a_{h,\gamma}(I_i\mathbf{u}_i, I_j\mathbf{u}_j) = \sum_{ij} g_{ij} a_{h,\gamma}(I_i\mathbf{u}_i, I_j\mathbf{u}_j) \\ &\leq \sum_{ij} g_{ij} \|\mathbf{u}_i\|_{A_i} \|\mathbf{u}_j\|_{A_j} \leq \frac{1}{2} \sum_{ij} g_{ij} (\|\mathbf{u}_i\|^2_{A_i} + \|\mathbf{u}_j\|^2_{A_j}) \\ &\leq N_O \sum_i \|\mathbf{u}_i\|^2_{A_i}. \end{aligned} \tag{5.31}$$

By (5.28), the statement follows with $c_2 = N_O$. We note that this bound is independent of the PDE.

However, the first inequality in (5.29) is harder to obtain and usually depends not only on the smoother but also on the PDE and the mesh size. We demonstrate this for the case of Jacobi relaxation, i.e. when $V_i = \{\alpha \boldsymbol{\varphi}_i : \alpha \in \mathbb{R}\}$ where $\{\boldsymbol{\varphi}_i\}$ is the set of basis functions of V_h. We note that the decomposition $\mathbf{u}_h = \sum \mathbf{u}_i$, $\mathbf{u}_i \in V_i$ is unique, and hence

$$\begin{aligned} \|\mathbf{u}_h\|^2_{D_{h,\gamma}} &= \sum_i \|\mathbf{u}_i\|^2_{A_{h,\gamma}} \preceq (1+\gamma) \sum_i \|\mathbf{u}_i\|^2_1 \preceq \frac{1+\gamma}{h^2} \sum_i \|\mathbf{u}_i\|^2_0 \\ &\preceq (1+\gamma) h^{-2} \|\mathbf{u}_h\|^2_0 \preceq (1+\gamma) h^{-2} \|\mathbf{u}_h\|^2_{A_{h,\gamma}}. \end{aligned} \tag{5.32}$$

This bound is parameter dependent and degrades as $\gamma \to \infty$.

In order to obtain a bound independent of γ, we require a subspace decomposition that respects the nullspace of the singular operator, which we denote by

$$\mathcal{N}_h = \{\mathbf{v}_h \in V_h : \Pi_{Q_h}(\nabla \cdot \mathbf{v}_h) = 0\}. \tag{5.33}$$

To give an intuition, we consider a $\mathbf{u}_0 \in \mathcal{N}_h$. If the subspace decomposition satisfies

$$\mathcal{N}_h = \sum_i V_i \cap \mathcal{N}_h, \tag{5.34}$$

then $\mathbf{u}_0$ can be written as

$$\mathbf{u}_0 = \sum_i \mathbf{u}_{0,i}, \qquad \mathbf{u}_{0,i} \in V_i \cap \mathcal{N}_h. \tag{5.35}$$

We redo the calculation in (5.32) but use the fact that each of the $\mathbf{u}_{0,i}$ are divergence-free, to obtain

$$\|\mathbf{u}_0\|^2_{D_{h,\gamma}} \leq \sum_i \|\mathbf{u}_{0,i}\|^2_{A_{h,\gamma}} \preceq \sum_i \|\mathbf{u}_{0,i}\|^2_1. \tag{5.36}$$

We now make this idea rigorous and prove γ independent spectral equivalence of $D_{h,\gamma}$ and $A_{h,\gamma}$. A key assumption, which we will need to check for each element and subspace decomposition individually, is that the splitting in (5.35) is stable, so that the last term in (5.36) can be bounded.

Proposition 5.4 ([Sch99b, Theorem 4.1])**.**
Let $\{V_i\}$ be a subspace decomposition of V_h with overlap N_O and assume that the pair $V_h \times Q_h$ is inf-sup stable for the mixed problem

$$B((\mathbf{u}, p), (\mathbf{v}, q)) := a(\mathbf{u}, \mathbf{v}) + (\nabla \cdot \mathbf{v}, p) + (\nabla \cdot \mathbf{u}, q). \tag{5.37}$$

Assume that $\mathbf{u}_h \in V_h$ and $\mathbf{u}_0 \in \mathcal{N}_h$ satisfy

$$\begin{aligned} \inf_{\substack{\mathbf{u}_h = \sum \mathbf{u}_i \\ \mathbf{u}_i \in V_i}} \sum_i \|\mathbf{u}_i\|_1^2 &\leq c_1(h) \|\mathbf{u}_h\|_0^2 \\ \inf_{\substack{\mathbf{u}_0 = \sum \mathbf{u}_{0,i} \\ \mathbf{u}_{0,i} \in \mathcal{N}_h \cap V_i}} \sum_i \|\mathbf{u}_{0,i}\|_1^2 &\leq c_2(h) \|\mathbf{u}_0\|_0^2. \end{aligned} \tag{5.38}$$

Then it holds that

$$(c_1(h) + c_2(h))^{-1} D_{h,\gamma} \preceq A_{h,\gamma} \leq N_O D_{h,\gamma}, \tag{5.39}$$

with constants independent of γ.

Proof. Let $\mathbf{u}_h \in V_h$, and consider a decomposition $\mathbf{u}_h = \mathbf{u}_0 + \mathbf{u}_1$ obtained by solving

$$B((\mathbf{u}_1, p_1), (\mathbf{v}_h, q_h)) = (\nabla \cdot \mathbf{u}_h, q_h) \qquad \text{for all } (\mathbf{v}_h, q_h) \in V_h \times Q_h. \tag{5.40}$$

Testing with $\mathbf{v}_h = 0$ we obtain that $\Pi_{Q_h}(\nabla \cdot \mathbf{u}_1) = \Pi_{Q_h}(\nabla \cdot \mathbf{u}_h)$ and hence $\Pi_{Q_h}(\nabla \cdot \mathbf{u}_0) = 0$. Furthermore, by stability we have

$$\begin{aligned} \|\mathbf{u}_1\|_1 &\preceq \sup_{\substack{\mathbf{v}_h \in V_h \\ q_h \in Q_h}} \frac{B((\mathbf{u}_1, p_1), (\mathbf{v}_h, q_h))}{\|(\mathbf{v}_h, q_h)\|} \\ &\overset{(5.40)}{\leq} \sup_{\substack{\mathbf{v}_h \in V_h \\ q_h \in Q_h}} \frac{\|\Pi_{Q_h}(\nabla \cdot \mathbf{u}_h)\|_0 \|q_h\|_0}{\|(\mathbf{v}_h, q_h)\|} \\ &\leq \|\Pi_{Q_h}(\nabla \cdot \mathbf{u}_h)\|_0 \end{aligned} \tag{5.41}$$

and hence $\|\mathbf{u}_1\|_1 \preceq \|\mathbf{u}_h\|_1$ and $\|\mathbf{u}_1\|_1 \preceq \gamma^{-1/2}\|\mathbf{u}_h\|_{A_{h,\gamma}}$. Using $\mathbf{u}_0 = \mathbf{u}_h - \mathbf{u}_1$ we obtain in addition that $\|\mathbf{u}_0\|_1 \preceq \|\mathbf{u}_h\|_1$ and conclude

$$\begin{aligned}\|\mathbf{u}_h\|_{D_h}^2 &\leq \inf_{\substack{\mathbf{u}_1=\sum \mathbf{u}_{1,i}\\ \mathbf{u}_{1,i}\in V_i}} \sum_i \underbrace{\|\mathbf{u}_{1,i}\|_{A_{h,\gamma}}^2}_{\leq(1+\gamma)\|\mathbf{u}_{1,i}\|_1^2} + \inf_{\substack{\mathbf{u}_0=\sum \mathbf{u}_{0,i}\\ \mathbf{u}_{0,i}\in \mathcal{N}_h\cap V_i}} \sum_i \underbrace{\|\mathbf{u}_{0,i}\|_{A_{h,\gamma}}^2}_{=\|\mathbf{u}_{0,i}\|_1^2} \\ &\overset{(5.38)}{\preceq} (1+\gamma)c_1(h)\|\mathbf{u}_1\|_0^2 + c_2(h)\|\mathbf{u}_0\|_0^2 \\ &\preceq (1+\gamma)c_1(h)\|\mathbf{u}_1\|_1^2 + c_2(h)\|\mathbf{u}_0\|_1^2 \\ &\preceq (c_1(h)+c_2(h))\|\mathbf{u}_h\|_{A_{h,\gamma}}^2.\end{aligned} \tag{5.42}$$

□

We will usually see bounds that behave as $c_1 \sim h^{-2}$ and $c_2 \sim h^{-4}$. The reason for the fast growth of c_2 is that we will construct the splitting into divergence-free functions by expressing $\mathbf{u}_0 = \nabla \times \mathbf{\Phi}$ for a potential $\mathbf{\Phi} \in H^2(\Omega;\mathbb{R}^d)$, introducing second derivatives into the estimates. This quartic growth of c_2 complicates a standard multigrid analysis, as one needs to show that a coarse-grid solve reduces the error in a way so that $D_{h,\gamma}$ acts on it in a mesh independent way. The analysis simplifies if instead we consider an additive coarse-grid solve.

Let V_H be a coarse-grid space (not necessarily satisfying $V_H \subset V_h$), let $P_H : V_H \to V_h$ be a prolongation operator, and let $A_{H,\gamma} : V_H \to V_H^*$ correspond to the PDE on the coarse space V_H. Then we define

$$C_{h,\gamma}^{-1} = P_H A_{H,\gamma}^{-1} P_H^* + \sum_i I_i A_i^{-1} I_i^* \tag{5.43}$$

to be the additive Schwarz preconditioner with coarse-grid correction.

Proposition 5.5 ([Sch99b, Lemma 3.5])**.**
Assume that

1. *the overlap of local spaces V_i is bounded by N_O;*
2. *the prolongation $P_H : V_H \to V_h$ is continuous in the energy norm with a constant independent of γ, i.e.*

$$\|P_H\mathbf{u}_H\|_{A_{h,\gamma}} \leq C_P\|\mathbf{u}_H\|_{A_{H,\gamma}} \qquad \text{for all } \mathbf{u}_H \in V_H; \tag{5.44}$$

3. *there exists an interpolation operator $I_H : V_h \to V_H$ that is continuous with a constant independent of γ, i.e.*

$$\|I_H\mathbf{u}_h\|_{A_{H,\gamma}} \leq C_I\|\mathbf{u}_h\|_{A_{h,\gamma}} \qquad \text{for all } \mathbf{u}_h \in V_h; \tag{5.45}$$

4. the difference $\mathbf{u}_f = \mathbf{u}_h - P_H I_H \mathbf{u}_h$ *satisfies*

$$\inf_{\substack{\mathbf{u}_i \in V_i \\ \sum_i \mathbf{u}_i = \mathbf{u}_f}} \sum_i \|\mathbf{u}_i\|^2_{A_{h,\gamma}} \leq C_S \|\mathbf{u}_h\|^2_{A_{h,\gamma}} \qquad \textit{for all } \mathbf{u}_h \in V_h \tag{5.46}$$

with a constant independent of γ.

Then the additive Schwarz preconditioner with coarse-grid correction satisfies

$$(C_I^2 + C_S)^{-1} C_{h,\gamma} \leq A_{h,\gamma} \leq (1 + N_O) \max\{C_P^2, 1\} C_{h,\gamma}. \tag{5.47}$$

Remark 5.6. *One can show that requirements 2.)-4.) are in fact necessary to obtain an optimal two-level preconditioner.*

Proof. Let $\mathbf{u}_h = P_H \mathbf{u}_H + \sum_i I_i \mathbf{u}_i$ be an arbitrary splitting, and denote

$$\Omega_i = \cup_{\mathbf{v} \in V_i} \operatorname{supp}(\mathbf{v}), \tag{5.48}$$

then

$$\begin{aligned}
\|\mathbf{u}_h\|^2_{A_{h,\gamma}} &= \|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}} + 2 \sum_i a_{h,\gamma}(P_H \mathbf{u}_H, I_i \mathbf{u}_i) + \sum_{ij} a_{h,\gamma}(I_i \mathbf{u}_i, I_j \mathbf{u}_j) \\
&\leq \|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}} + 2 \sum_i \|P_H \mathbf{u}_H\|_{A_{h,\gamma}(\Omega_i)} \|\mathbf{u}_i\|_{A_i} + N_O \sum_i \|\mathbf{u}_i\|^2_{A_i} \\
&\leq \|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}} + \sum_i \|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}(\Omega_i)} + \sum_i \|\mathbf{u}_i\|^2_{A_i} + N_O \sum_i \|\mathbf{u}_i\|^2_{A_i} \\
&\leq (1 + N_O)\big(\|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}} + \sum_i \|\mathbf{u}_i\|^2_{A_i}\big) \\
&\leq \max\{1, C_P^2\}(1 + N_O)\big(\|\mathbf{u}_H\|^2_{A_{H,\gamma}} + \sum_i \|\mathbf{u}_i\|^2_{A_i}\big).
\end{aligned} \tag{5.49}$$

Since the splitting was arbitrary, we obtain

$$A_{h,\gamma} \leq (1 + N_O) \max\{C_P^2, 1\} C_{h,\gamma}. \tag{5.50}$$

For the lower bound, we consider the splitting $\mathbf{u}_h = P_H I_H \mathbf{u}_h + (\mathbf{u}_h - P_H I_H \mathbf{u}_h) = P_H I_H \mathbf{u}_h + \mathbf{u}_f$ and calculate

$$\begin{aligned}
\|\mathbf{u}_h\|^2_{C_{h,\gamma}} &\leq \|I_H \mathbf{u}_h\|^2_{A_{H,\gamma}} + \inf_{\substack{\mathbf{u}_i \in V_i \\ \sum_i \mathbf{u}_i = \mathbf{u}_f}} \sum_i \|\mathbf{u}_i\|^2_{A_{h,\gamma}} \\
&\leq C_I^2 \|\mathbf{u}_h\|_{A_{h,\gamma}} + C_S \|\mathbf{u}_h\|^2_{A_{h,\gamma}} \\
&= (C_I^2 + C_S) \|\mathbf{u}_h\|^2_{A_{h,\gamma}}.
\end{aligned} \tag{5.51}$$

□

This proposition has two implications. First, though we cannot expect to bound $\|\mathbf{u}_h\|_{D_{h,\gamma}} \preceq \|\mathbf{u}_h\|_{A_{h,\gamma}}$ for all $\mathbf{u}_h \in V_h$ independently of γ and h, it is enough to be able to prove these bounds for functions of the form $\mathbf{u}_f = \mathbf{u}_h - P_H I_H \mathbf{u}_h$. Secondly, we require a robust prolongation operator to map coarse grid to fine grid functions. To get an intuition for this requirement, we calculate

$$\begin{aligned} \|\mathbf{u}_H\|^2_{A_{H,\gamma}} &= \|\mathbf{u}_H\|^2_{A_H} + \gamma \|\Pi_{Q_H}(\nabla \cdot \mathbf{u}_H)\|^2_0 \\ \|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}} &= \|P_H \mathbf{u}_H\|^2_{A_h} + \gamma \|\Pi_{Q_h}(\nabla \cdot (P_H \mathbf{u}_H))\|^2_0. \end{aligned} \tag{5.52}$$

The key difficulty lies in the second term of this norm. To see this, observe that for an element $\mathbf{u}_H \in \mathcal{N}_H$ the second term in $\|\mathbf{u}_H\|^2_{A_{H,\gamma}}$ vanishes, but since it does not necessarily hold that $P_H \mathbf{u}_H \in \mathcal{N}_h$, the corresponding term in $\|P_H \mathbf{u}_H\|^2_{A_{h,\gamma}}$ might be large.

To avoid this, we must modify the prolongation operator to map fields that are discretely divergence-free on the coarse grid to fields that are (nearly) discretely divergence-free on the fine grid. We now describe a modification of the standard prolongation operator that satisfies this conditions. This type of modification goes back to Schöberl's work, though we give a different derivation and proof.

Let $\mathbf{u}_H \in \mathcal{N}_H$ be a discretely divergence-free function on the coarse-grid and denote the standard prolongation induced by the interpolation operator on the finite element space by $P_H \mathbf{u}_H$. We are interested in finding a small perturbation $\tilde{\mathbf{u}}_h$ such that $P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h \in \mathcal{N}_h$. This is for example given by solving

$$\begin{aligned} \min_{\tilde{\mathbf{u}}_h \in V_h} \quad & a(\tilde{\mathbf{u}}_h, \tilde{\mathbf{u}}_h) \\ \text{s.t.} \quad & \Pi_{Q_h}(\nabla \cdot \tilde{\mathbf{u}}_h) = \Pi_{Q_h}(\nabla \cdot P_H \mathbf{u}_H). \end{aligned} \tag{5.53}$$

This corresponds to solving a Stokes like problem in $V_h \times Q_h$. We will now relax this problem in two aspects. First, we do not need to enforce that $P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h$ has zero discrete divergence, but it is enough if it is suitably small, i.e. we can instead find $\tilde{\mathbf{u}}_h \in V_h$ that minimises

$$\min_{\tilde{\mathbf{u}}_h \in V_h} a(\tilde{\mathbf{u}}_h, \tilde{\mathbf{u}}_h) + \gamma \|\Pi_{Q_h}(\nabla \cdot (P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h))\|^2_0. \tag{5.54}$$

This corresponds to solving

$$a_{h,\gamma}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = \gamma(\Pi_{Q_h}(\nabla \cdot P_H \mathbf{u}_H), \Pi_{Q_h}(\nabla \cdot \tilde{\mathbf{v}}_h)) \quad \text{for all } \tilde{\mathbf{v}}_h \in V_h. \tag{5.55}$$

Clearly at this stage we have not gained much, since we now need to solve a global problem involving the nearly singular bilinear form $a_{h,\gamma}$. However, it turns out that

under certain assumptions that we will state in the following proposition, one can instead solve the same problem on smaller spaces $\tilde{V}_h \subset V_h$ and $\tilde{Q}_h \subset Q_h$:

$$\min_{\tilde{\mathbf{u}}_h \in \tilde{V}_h} a(\tilde{\mathbf{u}}_h, \tilde{\mathbf{u}}_h) + \gamma \|\Pi_{\tilde{Q}_h}(\nabla \cdot (P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h))\|_0^2. \tag{5.56}$$

or equivalently: find $\tilde{\mathbf{u}}_h \in \tilde{V}_h$ such that

$$a_{h,\gamma}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = \gamma(\Pi_{\tilde{Q}_h}(\nabla \cdot (P_H \mathbf{u}_H)), \Pi_{\tilde{Q}_h}(\nabla \cdot \tilde{\mathbf{v}}_h)) \quad \text{for all } \tilde{\mathbf{v}}_h \in \tilde{V}_h. \tag{5.57}$$

How exactly one chooses these subspaces will depend on the discretisation under consideration and will be studied in detail in the next two chapters.

Proposition 5.7 (Robust prolongation)**.**
Assume we can split $Q_h = \tilde{Q}_H \oplus \tilde{Q}_h$ and that $\tilde{Q}_H \subset Q_H$. Let $P_H : V_H \to V_h$ be a prolongation operator that is continuous in the $\|\cdot\|_1$ norm and preserves the divergence with respect to $\tilde{Q}_H$, i.e.

$$(\nabla \cdot (P_H \mathbf{v}_H), \tilde{q}_H) = (\nabla \cdot \mathbf{v}_H, \tilde{q}_H) \qquad \text{for all } \tilde{q}_H \in \tilde{Q}_H, \mathbf{v}_H \in V_H. \tag{5.58}$$

Assume in addition that there exists a $\tilde{V}_h \subset V_h$ such that

$$(\nabla \cdot \tilde{\mathbf{v}}_h, \tilde{q}_H) = 0 \qquad \text{for all } \tilde{q}_H \in \tilde{Q}_H, \tilde{\mathbf{v}}_h \in \tilde{V}_h. \tag{5.59}$$

and such that the pairing $\tilde{V}_h \times \tilde{Q}_h$ is inf-sup stable, i.e.

$$\inf_{\tilde{q}_h \in \tilde{Q}_h} \sup_{\tilde{\mathbf{v}}_h \in \tilde{V}_h} \frac{(\tilde{q}_h, \nabla \cdot \tilde{\mathbf{v}}_h)}{\|\tilde{\mathbf{v}}_h\|_1 \|\tilde{q}_h\|_0} \geq c \tag{5.60}$$

for some mesh independent $c > 0$. For $\mathbf{u}_H \in V_H$, define $\tilde{\mathbf{u}}_h$ as the solution to

$$a_{h,\gamma}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = \gamma(\Pi_{Q_h}(\nabla \cdot (P_H \mathbf{u}_H)), (\Pi_{Q_h}(\nabla \cdot (\tilde{\mathbf{v}}_h)))) \qquad \text{for all } \tilde{\mathbf{v}}_h \in \tilde{V}_h. \tag{5.61}$$

Then the prolongation $\tilde{P}_H : V_H \to V_h$ defined by

$$\tilde{P}_H \mathbf{u}_H = P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h \tag{5.62}$$

is continuous in the energy norm.

Remark 5.8. *The problems in* (5.57) *and* (5.61) *are equivalent by the assumption in* (5.59)*.*

Remark 5.9. *This prolongation operator is very similar to the one used by [Sch99b, Theorem 4.2] and [BO06, Lemma 5.1]. The difference is that in Schöberl's work the problem in* (5.61) *is replaced with*

$$a_{h,\gamma}(\tilde{\mathbf{u}}_h, \tilde{v}_h) = a_{h,\gamma}(P_H \mathbf{u}_H, \tilde{\mathbf{v}}_h) \qquad \text{for all } \tilde{\mathbf{v}}_h \in \tilde{V}_h. \tag{5.63}$$

We use the version in (5.61) *for two reasons: first, it reduces to the standard prolongation for* $\gamma = 0$*, and second, it can be viewed as a local version of the global problem in* (5.55)*, where as solving a global version of* (5.63) *would lead to* $\tilde{\mathbf{u}}_h = P_H \mathbf{u}_H$ *and hence* $\tilde{P}_H \mathbf{u}_H = 0$*.*

The proofs of Schöberl and Benzi & Olshanskii are based on an equivalent mixed problem. We give a different proof motivated by the formulation as an optimisation problem and use the existence of a Fortin operator due to inf-sup stability.

Proof of Proposition 5.7. We denote

$$J(\tilde{\mathbf{v}}_h) := a(\tilde{\mathbf{v}}_h, \tilde{\mathbf{v}}_h) + \gamma \|\Pi_{\tilde{Q}_h}(\nabla \cdot (P_H \mathbf{u}_H - \tilde{\mathbf{v}}_h))\|_0^2, \tag{5.64}$$

and observe that $\tilde{\mathbf{u}}_h$ is the unique minimiser of J in $\tilde{V}_h$. By inf-sup stability of the pairing $\tilde{V}_h \times \tilde{Q}_h$ there exists a continuous Fortin operator $I : V \to \tilde{V}_h$ that satisfies

$$\Pi_{\tilde{Q}_h}(\nabla \cdot (I\mathbf{v})) = \Pi_{\tilde{Q}_h}(\nabla \cdot \mathbf{v}) \quad \text{for all } \mathbf{v} \in V. \tag{5.65}$$

Let $\bar{\mathbf{u}}_h := I(P_H \mathbf{u}_H) \in \tilde{V}_h$ and observe that

$$J(\tilde{\mathbf{u}}_h) \leq J(\bar{\mathbf{u}}_h) = a(\bar{\mathbf{u}}_h, \bar{\mathbf{u}}_h) \preceq \|P_H \mathbf{u}_H\|_1^2 \preceq \|\mathbf{u}_H\|_1^2. \tag{5.66}$$

We conclude

$$\begin{aligned}
&\|\tilde{P}_H \mathbf{u}_H\|_{A_{h,\gamma}}^2 \\
\leq &\|P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h\|_{A_h}^2 + \gamma \|\Pi_{Q_h}(\nabla \cdot (P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h))\|_0^2 \\
\preceq &\underbrace{\|P_H \mathbf{u}_H\|_{A_h}^2}_{\preceq \|\mathbf{u}_H\|_{A_h}^2} + \underbrace{\|\tilde{\mathbf{u}}_h\|_{A_h}^2 + \gamma \|\Pi_{\tilde{Q}_h}(\nabla \cdot (P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h))\|_0^2}_{= J(\tilde{\mathbf{u}}_h) \underset{(5.66)}{\preceq} \|\mathbf{u}_H\|_1^2} + \gamma \|\Pi_{\tilde{Q}_H}(\nabla \cdot (P_H \mathbf{u}_H - \tilde{\mathbf{u}}_h))\|_0^2 \\
\preceq &\|\mathbf{u}_H\|_1^2 + \gamma \|\underbrace{\Pi_{\tilde{Q}_H}(\nabla \cdot (P_H \mathbf{u}_H))}_{\underset{(5.58)}{=} \Pi_{\tilde{Q}_H}(\nabla \cdot \mathbf{u}_H)}\|_0^2 + \gamma \|\underbrace{\Pi_{\tilde{Q}_H}(\nabla \cdot \tilde{\mathbf{u}}_h)}_{\underset{(5.59)}{=} 0}\|_0^2 \\
\preceq &\|\mathbf{u}_H\|_1^2 + \gamma \|\Pi_{\tilde{Q}_H}(\nabla \cdot \mathbf{u}_H)\| \\
\preceq &\|\mathbf{u}_H\|_{A_{H,\gamma}}^2.
\end{aligned} \tag{5.67}$$

□

Remark 5.10. *Note that if* $\operatorname{div}(V_H) \subset Q_H$*, then discretely divergence-free vectorfields on the coarse grid are in fact exactly divergence-free. If in addition* $V_H \subset V_h$*, then the natural inclusion is a continuous prolongation operator. This corresponds to the situation that was studied by Lee et al. [LWC09] for nearly singular elasticity and hence they do not require special prolongation operators. However, as we will see in the next two chapters,* H^1*-conforming, stable discretisations for the Stokes and Navier–Stokes equations are often either nonnested or not exactly divergence-free.*

Chapter 6

Augmented Lagrangian preconditioning for a low order discretisation

In the previous chapter we described the requirements for a multigrid scheme with γ robust performance for the top-left block A_γ that arises in the augmented Lagrangian formulation. We will now study the specific case of a low order discretisation with piecewise constant pressures. For this case the discrete divergence corresponds directly to the flux across edges/facets in 2D/3D, simplifying the characterisation of divergence-free vector fields.

The chapter is organised as follows. In Section 6.1 we describe the two dimensional case that Schöberl, Benzi & Olshanskii studied and construct robust smoothing and prolongation operators for the $[\mathbb{P}_2]^2-\mathbb{P}_0$ element. We then extend their work to three dimensions. While the same general strategy applies in three dimensions, in Section 6.2 we will explain that the $[\mathbb{P}_2]^3-\mathbb{P}_0$ element cannot be used, as it would lead to ill-posed problems as part of the prolongation. We will propose appropriate alternative finite element discretisations and matching prolongation operators that result in a robust multigrid scheme.

The study will have ignored the advection term up to that point. In Section 6.3 we discuss the inclusion of the advection term and the stabilisation scheme that we employ. We present an implementation of the full solver in Section 6.4, based on the Firedrake finite element library and the developed PCPATCH preconditioner in PETSc. We demonstrate robust performance up to $\mathrm{Re} = 10\,000$ in two dimensions and $\mathrm{Re} = 5\,000$ in three dimensions for two classical benchmark problems.

6.1 Robust smoothing and prolongation in two dimensions

Given a simply-connected domain Ω, we consider a simplicial mesh $\mathcal{M}_H$, that is $\cup_{K\in\mathcal{M}_H} K = \overline{\Omega}$ with $\mathrm{int}(K_1)\cap\mathrm{int}(K_2) = \emptyset$ for all $K_1 \neq K_2 \in \mathcal{M}_H$. A fine grid $\mathcal{M}_h$, $h = H/2$ is then obtained by regular refinement of the simplices in $\mathcal{M}_H$. As before we only consider the two level case in the exposition, but the approach extends in the usual way to arbitrarily many levels. The function spaces we consider in two dimensions are given by

$$V_h := \{\mathbf{v} \in H^1(\Omega;\mathbb{R}^2) : \mathbf{v}|_K \in [P^2(K)]^2 \ \forall K \in \mathcal{M}_h\}, \tag{6.1}$$

$$Q_h := \{q \in L^2(\Omega) : q|_K \in P^0(K) \ \forall K \in \mathcal{M}_h\}, \tag{6.2}$$

with coarse grid spaces V_H and Q_H defined in the same way on $\mathcal{M}_H$. If the velocity is prescribed on all of $\partial\Omega$, then we require in addition that the pressures in Q_H and Q_h integrate to zero.

6.1.1 Prolongation

We begin by studying the prolongation operator that maps V_H to V_h. Since the spaces are nested, the standard prolongation P_H given by interpolation at the degrees of freedom is equal to the identity operator. However, as discussed at the end of the previous chapter, a velocity field that is divergence-free with respect to Q_H does not necessarily need to be divergence-free with respect to the (larger) space Q_h. In order to use Proposition 5.7 we denote

$$\begin{aligned} \tilde{Q}_H &:= Q_H, \\ \tilde{Q}_h &:= \{q_h \in Q_h : \Pi_{Q_H} q_h = 0\}, \end{aligned} \tag{6.3}$$

and by definition

$$Q_h = \tilde{Q}_H \oplus \tilde{Q}_h. \tag{6.4}$$

The space $\tilde{Q}_h$ represents the 'extra pressure functions' obtained by refining the mesh. Since $V_H \subset V_h$ and the prolongation is just the identity, we immediately obtain that the divergence with respect to functions in $\tilde{Q}_H$ is preserved, i.e.

$$(\nabla\cdot(P_H\mathbf{u}_H), \tilde{q}_H) = (\nabla\cdot\mathbf{u}_H, \tilde{q}_H) \quad \forall \mathbf{u}_H \in V_H, \tilde{q}_H \in \tilde{Q}_H. \tag{6.5}$$

Hence, if a function is discretely divergence-free on the coarse grid, after prolongation it remains discretely divergence-free with respect to functions in $\tilde{Q}_H$. In order to

remove any divergence with respect to functions in $\tilde{Q}_h$, we will solve a local problem in each cell. To this end, we define

$$\tilde{V}_h \coloneqq \{\mathbf{v}_h \in V_h : \mathbf{v}_h|_{\partial K} = 0 \; \forall K \in \mathcal{M}_H\}. \tag{6.6}$$

The space $\tilde{V}_h$ consists of velocity degrees of freedom in V_h that are in the interior of coarse cells $K \in \mathcal{M}_H$, as shown in Figure 6.1. We note that though we always show regular grids, this is not required.

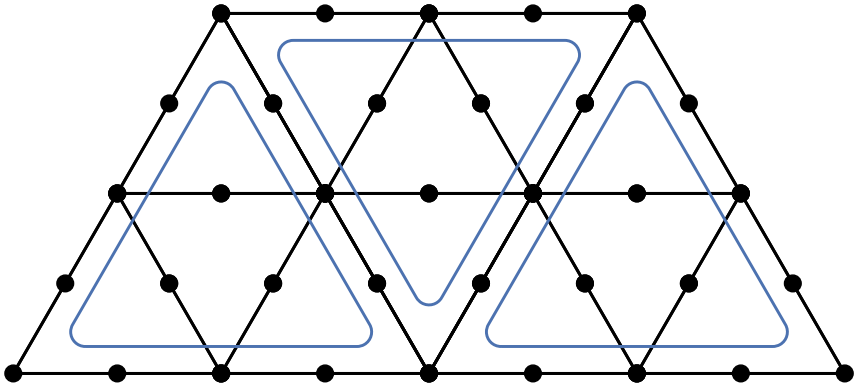


Figure 6.1: The subspace $\tilde{V}_h$ consists of local patches collecting 6 degrees of freedom inside each coarse-grid cell.

Since functions in $\tilde{V}_h$ vanish on the boundary of coarse grid cells, we obtain

$$(\nabla \cdot \tilde{\mathbf{u}}_h, \tilde{q}_H) = 0 \quad \forall \tilde{\mathbf{u}}_h \in \tilde{V}_h, \; \tilde{q}_H \in \tilde{Q}_H. \tag{6.7}$$

Lastly we note that the pairing $\tilde{V}_h \times \tilde{Q}_h$ is inf-sup stable and hence by Proposition 5.7, a robust prolongation operator $\tilde{P}_H$ is then given by

$$\tilde{P}_H \mathbf{u}_H = \mathbf{u}_H - \tilde{\mathbf{u}}_h \tag{6.8}$$

where $\tilde{\mathbf{u}}_h \in \tilde{V}_h$ solves

$$a_{h,\gamma}(\tilde{\mathbf{u}}_h, \tilde{\mathbf{v}}_h) = \gamma(\Pi_{Q_h}(\nabla \cdot \mathbf{u}_H), \Pi_{Q_h}(\nabla \cdot \tilde{\mathbf{v}}_h)) \qquad \text{for all } \tilde{\mathbf{v}}_h \in \tilde{V}_h. \tag{6.9}$$

We highlight that the space $\tilde{V}_h$ decouples into small patch solves involving only 6 degrees of freedom per coarse grid cell as shown in Figure 6.1. This is crucial for an efficient implementation.

6.1.2 Relaxation

In the previous chapter we established that the key requirement for a subspace decomposition $\{V_i\}$ is that it decomposes the nullspace of the divergence. In the case of piecewise continuous pressures, this is satisfied by choosing subspaces corresponding to patches around vertices of the mesh. More specifically, for each vertex v_i in the mesh $\mathcal{M}_h$, its *star* is the patch of elements sharing v_i:

$$\operatorname{star}(v_i) \coloneqq \bigcup_{K \in \mathcal{M}_h \,:\, v_i \in K} K. \tag{6.10}$$

The subspace decomposition we consider is given by subspaces V_i associated with each vertex v_i defined by

$$V_i \coloneqq \{\mathbf{u}_h \in V_h : \operatorname{supp}(\mathbf{u}_h) \subset \operatorname{star}(v_i)\}. \tag{6.11}$$

We call the resulting subspace correction method a *star iteration*, it is shown in Figure 6.2. For the reader's convenience, we repeat the argument of [Sch99b, Sec-

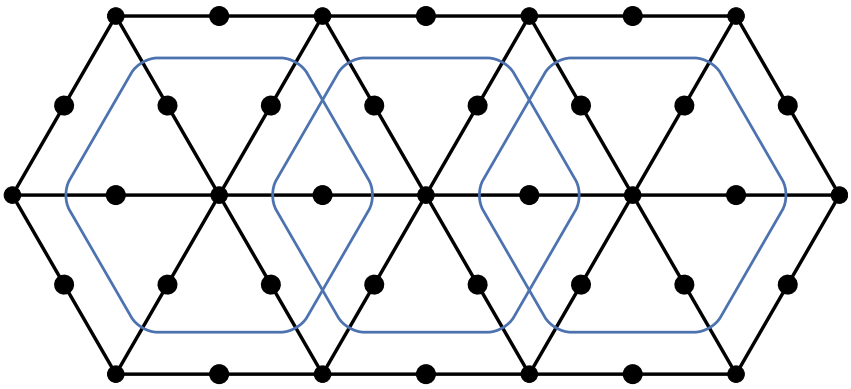


Figure 6.2: The *star* patch around a vertex collects the degrees of freedom inside the cells attached to this vertex. The overlap is given by $N_O = 2$.

tion 4.1.2] to show that this decomposition satisfies the requirement in Proposition 5.5 for a robust relaxation. The proof for this Lemma requires the assumption that Ω is simply-connected. In practice, we observe good performance for the multigrid scheme also for domains that do not satisfy this assumption, as the nonlocal solenoidal vector fields are captured by the coarse grid solve.

Proposition 6.1.

Consider the subspace decomposition defined in (6.11) *and denote by* $D_{h,\gamma}$ *the subspace*

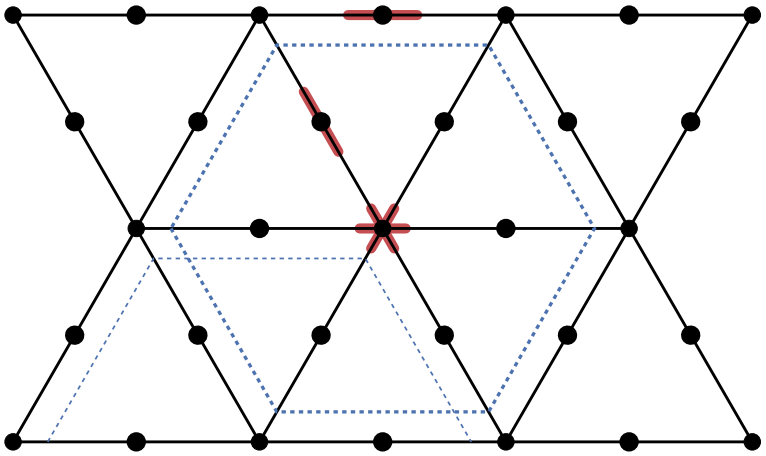


Figure 6.3: The integration regions around each degree of freedom (highlighted in red) are chosen to either be entirely inside or entirely outside of each domain Ω_i (dashed).

correction method associated with the decomposition $\{V_i\}$. *Then* $\mathbf{u}_h \in V_h$ *and* $\mathbf{u}_0 \in \mathcal{N}_h$ *satisfy*

$$\inf_{\substack{\mathbf{u}_h = \sum \mathbf{u}_i \\ \mathbf{u}_i \in V_i}} \sum_i \|\mathbf{u}_i\|_1^2 \preceq h^{-2} \|\mathbf{u}_h\|_0^2 \tag{6.12}$$

$$\inf_{\substack{\mathbf{u}_0 = \sum \mathbf{u}_{0,i} \\ \mathbf{u}_{0,i} \in \mathcal{N}_h \cap V_i}} \sum_i \|\mathbf{u}_{0,i}\|_1^2 \preceq h^{-4} \|\mathbf{u}_0\|_0^2. \tag{6.13}$$

Moreover, it holds that

$$(h^{-2} + h^{-4})^{-1} D_{h,\gamma} \preceq A_{h,\gamma} \leq N_O D_{h,\gamma} \tag{6.14}$$

where N_0 *is the maximum number of overlapping subspaces in the mesh. In particular, the bounds are independent of* γ.

Proof. The second inequality in (6.14) was proven in (5.31) and estimate (6.12) follows by performing the same calculation as in (5.32) but with $\gamma = 0$. It remains to prove the splitting estimate for divergence-free vector fields.

Step 1: Decomposition of divergence-free vector fields. We observe that a discretely divergence-free vector field $\mathbf{u}_0 \in \mathcal{N}_h$ can be suitably modified in the *interior* of each cell to become continuously divergence-free by solving a local Stokes problem. Denote this continuously divergence-free vector field by $\tilde{\mathbf{u}}$ and recall that then $\tilde{\mathbf{u}} = \nabla \times \mathbf{\Phi}$ for some vector field $\mathbf{\Phi} \in H_0^2(\Omega)$ with $\|\mathbf{\Phi}\|_2 \preceq \|\tilde{\mathbf{u}}\|_1$ [GR86, p. 42]. Now pick a covering $\{\Omega_i\}_i$ such that Ω_i is strictly contained in $\operatorname{star}(v_i)$ in the sense, that for any $x \in \Omega_i$ it

holds $B(x, h/10) \subset \operatorname{star}(v_i)$. Choosing a partition of unity $\{\rho_i\}_i$ such that for all i

$$\begin{aligned} \|\rho_i\|_{L^\infty} &\leq 1 \\ \|\rho_i\|_{W^{1,\infty}} &\preceq h^{-1} \\ \|\rho_i\|_{W^{2,\infty}} &\preceq h^{-2} \\ \operatorname{supp}(\rho_i) &\subset \Omega_i, \end{aligned} \tag{6.15}$$

we define $\mathbf{\Phi}_i = \rho_i \mathbf{\Phi}$ and obtain a decomposition

$$\mathbf{\Phi} = \sum_i \mathbf{\Phi}_i. \tag{6.16}$$

Now let $I_1 : V \to V_h$ be a Scott–Zhang interpolation operator [SZ90]. We construct the interpolation operator to integrate over a region of size $h/10$ of the facets close to each degree of freedom only, as shown in Figure 6.3. Together with the requirement on the covering $\{\Omega_i\}_i$ this implies a locality property

$$I_1(\mathbf{v}) \in V_i \text{ for all } \mathbf{v} \in V \text{ s.t. } \operatorname{supp}(\mathbf{v}) \subset \Omega_i. \tag{6.17}$$

The operator can be chosen such that $I_1(\mathbf{u}_h) = \mathbf{u}_h$. Furthermore, define $I_2 : V \to V_h$ as in the classical proof for inf-sup stability of the $[\mathbb{P}_2]^2 - \mathbb{P}_0$ element [Bof+08, Proposition 3.1]:

$$\begin{aligned} I_2(\mathbf{v})(M) &= 0, \quad \text{for all vertices } M, \\ \int_E I_2(\mathbf{v}) \,\mathrm{d}s &= \int_E \mathbf{v} \,\mathrm{d}s, \quad \text{for all edges } E. \end{aligned} \tag{6.18}$$

Now define $I(\mathbf{v}) = I_1(\mathbf{v}) + I_2(\mathbf{v} - I_1(\mathbf{v}))$, then it holds that

$$\begin{aligned} I(\mathbf{v}_h) &= \mathbf{v}_h \quad \text{for all } \mathbf{v}_h \in V_h, \\ \int_E I(\mathbf{v}) \,\mathrm{d}s &= \int_E \mathbf{v} \,\mathrm{d}s \quad \text{for all } \mathbf{v} \in V, \\ I(\mathbf{v}) &\in V_i \text{ for all } \mathbf{v} \in V \text{ s.t. } \operatorname{supp}(\mathbf{v}) \subset \Omega_i. \end{aligned} \tag{6.19}$$

Hence we can define

$$\mathbf{u}_i := I(\nabla \times \mathbf{\Phi}_i) \in V_i \tag{6.20}$$

and conclude that

$$\sum_i \mathbf{u}_i = \sum_i I(\nabla \times \mathbf{\Phi}_i) = I(\nabla \times \mathbf{\Phi}) = I(\tilde{\mathbf{u}}) = \mathbf{u}_h. \tag{6.21}$$

Lastly, using the fact that we are considering piecewise constant pressures, $\mathbf{u}_i \in V_i \cap \mathcal{N}_h$ follows from

$$\int_K \nabla \cdot \mathbf{u}_i \,\mathrm{d}x = \int_{\partial K} \mathbf{u}_i \cdot \mathbf{n} \,\mathrm{d}s = \int_{\partial K} (\nabla \times \mathbf{\Phi}_i) \cdot \mathbf{n} \,\mathrm{d}s = \int_K \nabla \cdot (\nabla \times \mathbf{\Phi}_i) \,\mathrm{d}x = 0 \tag{6.22}$$

for any element $K \in \mathcal{M}_h$.

Step 2: Splitting estimate for divergence-free vector fields. The bound in (6.13) can now be obtained by calculating

$$\begin{aligned}
&\inf_{\substack{\mathbf{u}_0=\sum \mathbf{u}_{0,i} \\ \mathbf{u}_{0,i}\in\mathcal{N}_h\cap V_i}} \sum_i \|\mathbf{u}_{0,i}\|_1^2 \\
&\preceq \sum_i \|\mathbf{u}_i\|_1^2 = \sum_i \|I_h(\nabla\times(\rho_i\mathbf{\Phi}))\|_1^2 \\
&\preceq \sum_i \|\rho_i\mathbf{\Phi}\|_2^2 \preceq \sum_i h^{-4}\|\mathbf{\Phi}\|_{0,\Omega_i}^2 + h^{-2}\|\mathbf{\Phi}\|_{1,\Omega_i}^2 + \|\mathbf{\Phi}\|_{2,\Omega_i}^2 \\
&\preceq h^{-4}\|\mathbf{u}_0\|_0^2.
\end{aligned} \tag{6.23}$$

Finally, the claim in (6.14) follows from Proposition 5.4 □

We have proven that the block Jacobi method induced by the spaces $\{V_i\}$ yields a γ robust but mesh dependent relaxation. By adding a coarse grid solve, we obtain a preconditioner that is also mesh independent.

Proposition 6.2 ([Sch99b, § 4.2.1])**.**
There exists a continuous interpolation operator $I_H : V_h \to V_H$ such that for every $\mathbf{u}_h \in V_h$ the difference

$$\mathbf{u}_f = \mathbf{u}_h - \tilde{P}_H I_H \mathbf{u}_h \tag{6.24}$$

satisfies

$$\inf_{\substack{\mathbf{u}_i\in V_i \\ \sum_i \mathbf{u}_i=\mathbf{u}_f}} \sum_i \|\mathbf{u}_i\|_{A_{h,\gamma}}^2 \le C_S \|\mathbf{u}_h\|_{A_{h,\gamma}}^2. \tag{6.25}$$

Hence the additive two-level solver given by

$$C_{h,\gamma}^{-1} = P_H A_{H,\gamma}^{-1} P_H^* + \sum_i I_i A_i^{-1} I_i^* \tag{6.26}$$

satisfies the requirements of Proposition 5.5 and is both γ and h robust.

Proof. First we construct a Fortin operator as in the proof of Lemma 6.1, i.e. a continuous linear map $I_H : V \to V_H$ that satisfies

$$\int_E I_H \mathbf{v}_H \,\mathrm{d}s = \int_E \mathbf{v}_H \,\mathrm{d}s \tag{6.27}$$

for every edge E in the coarse-grid $\mathcal{M}_H$. Then for every $\mathbf{v}_h \in V_h$ we have by continuity $\|I_H\mathbf{v}_h\|_1 \preceq \|\mathbf{v}_h\|_1$ and since edge integrals are preserved we have

$$\|\Pi_{Q_H}(\nabla\cdot(I_H\mathbf{v}_h))\|_0 = \|\Pi_{Q_H}(\nabla\cdot\mathbf{v}_h)\|_0 \le \|\Pi_{Q_h}(\nabla\cdot\mathbf{v}_h)\|_0 \tag{6.28}$$

and hence

$$\|I_H \mathbf{v}_h\|_{A_{H,\gamma}} \preceq \|\mathbf{v}_h\|_{A_{h,\gamma}}. \tag{6.29}$$

Note that the prolongation operator $\tilde{P}_H$ is exact on coarse grid edges E, and hence

$$\int_E \mathbf{u}_f \, \mathrm{d}s = 0. \tag{6.30}$$

We now split $\mathbf{u}_f = \mathbf{u}_0 + \mathbf{u}_1$ by solving for $\mathbf{u}_1 \in \tilde{V}_h$ and $p_1 \in \tilde{Q}_h$ that satisfy

$$B((\mathbf{u}_1, p_1), (\tilde{\mathbf{v}}, \tilde{q})) = (\nabla \cdot \mathbf{u}_f, q_h) \quad \text{for all } \tilde{\mathbf{v}} \in \tilde{V}_h, \tilde{q} \in \tilde{Q}_h. \tag{6.31}$$

Then we have

$$\Pi_{\tilde{Q}_h}(\nabla \cdot \mathbf{u}_1) = \Pi_{\tilde{Q}_h}(\nabla \cdot \mathbf{u}_f) \tag{6.32}$$

and since $0 = \Pi_{\tilde{Q}_H}(\nabla \cdot \mathbf{u}_1) = \Pi_{\tilde{Q}_H}(\nabla \cdot \mathbf{u}_f)$ and $Q_h = \tilde{Q}_H \oplus \tilde{Q}_h$ we have

$$\Pi_{Q_h}(\nabla \cdot \mathbf{u}_1) = \Pi_{Q_h}(\nabla \cdot \mathbf{u}_f). \tag{6.33}$$

In addition, by stability (following the same argument as in (5.41)) we have

$$\|\mathbf{u}_1\|_1 \preceq \sup_{\substack{\mathbf{v} \in \tilde{V}_h \\ \tilde{q} \in \tilde{Q}_h}} \frac{B((\mathbf{u}_1, p_1), (\tilde{\mathbf{v}}, \tilde{q}))}{\|(\tilde{\mathbf{v}}, \tilde{q})\|} \leq \sup_{\substack{\mathbf{v} \in \tilde{V}_h \\ \tilde{q} \in \tilde{Q}_h}} \frac{\|\Pi_{\tilde{Q}_h}(\nabla \cdot \mathbf{u}_f)\|_0 \|q\|_0}{\|(\tilde{\mathbf{v}}, \tilde{q})\|} \tag{6.34}$$

and hence $\|\mathbf{u}_1\|_1 \preceq \|\mathbf{u}_f\|_1$, $\|\mathbf{u}_1\|_1 \preceq \gamma^{-1/2} \|\mathbf{u}_f\|_{A_{h,\gamma}}$ and $\|\mathbf{u}_0\|_1 \preceq \|\mathbf{u}_f\|_1$. Note that since $\mathbf{u}_1$ vanishes on coarse-grid edges, we additionally have $\|\mathbf{u}_1\|_0 \preceq H \|\mathbf{u}_1\|_1$.

Now proceed as in the proof of Lemma 6.1 and modify $\mathbf{u}_0$ in the interior of each cell so that it becomes divergence-free. Denote this vector field by $\tilde{\mathbf{u}}_0$ and then pick $\mathbf{\Phi} \in H_0^2(\Omega)$ so that $\nabla \times \mathbf{\Phi} = \tilde{\mathbf{u}}_f$. On a coarse grid edge E with tangent $\mathbf{t}$ it then holds

$$\int_E \nabla \mathbf{\Phi} \cdot \mathbf{t} \, \mathrm{d}s = \int_E (\nabla \times \mathbf{\Phi}) \cdot \mathbf{n} \, \mathrm{d}s = \int_E \mathbf{u}_0 \cdot \mathbf{n} \, \mathrm{d}s = \int_E \mathbf{u}_f \cdot \mathbf{n} \, \mathrm{d}s = 0, \tag{6.35}$$

implying that $\mathbf{\Phi}$ has the same value on every coarse grid vertex, and since $\mathbf{\Phi}$ is zero on the boundary, we know that this value is zero. The $\mathbb{P}_1$ interpolant of $\mathbf{\Phi}$ vanishes, and hence we obtain that

$$\|\mathbf{\Phi}\|_0 \preceq H^{-2} \|\mathbf{\Phi}\|_2 \text{ and } \|\mathbf{\Phi}\|_1 \preceq H^{-1} \|\mathbf{\Phi}\|_2. \tag{6.36}$$

Proceeding as in (6.23), we obtain

$$\|\mathbf{u}_0\|_{A_{h,\gamma}}^2 = \inf_{\substack{\mathbf{u}_0 = \sum \mathbf{u}_{0,i} \\ \mathbf{u}_{0,i} \in \mathcal{N}_h \cap V_i}} \sum_i \|\mathbf{u}_{0,i}\|_1^2 \preceq \|\mathbf{\Phi}\|_2^2 \preceq \|\tilde{\mathbf{u}}_0\|_1^2 \preceq \|\mathbf{u}_f\|_1^2. \tag{6.37}$$

Finally,

$$\|\mathbf{u}_f\|_{D_{h,\gamma}}^2 \leq \|\mathbf{u}_0\|_{D_{h,\gamma}}^2 + \|\mathbf{u}_1\|_{D_{h,\gamma}}^2 \preceq \|\mathbf{u}_f\|_{A_{h,\gamma}}^2. \tag{6.38}$$

□

Remark 6.3. *Pressure elements other than $\mathbb{P}_0$ have been considered for the augmented Lagrangian preconditioner. Benzi & Olshanskii [BO06, Table 6.2] also present results for the $[\mathbb{P}_1 iso \mathbb{P}_2]^2 - \mathbb{P}_1$ pair, where the pressure mass matrix solve in Π_{Q_h} is approximated by the inverse of a diagonal matrix. However, for this element pairing the developed multigrid scheme is not independent of the ratio γ/ν and hence as ν decreases, γ has to be decreased correspondingly. This in turn leads to worse control of the Schur complement and consequent growth in iteration counts.*

6.2 Robust smoothing and prolongation in three dimensions

6.2.1 Prolongation

We now turn to the three dimensional case. Unfortunately, in three dimensions the choice $[\mathbb{P}_2]^3 - \mathbb{P}_0$ results in a pairing $\tilde{V}_h - \tilde{Q}_h$ in the local solves for the prolongation that is not inf-sup stable. This can be seen by counting degrees of freedom: $[\mathbb{P}_2]^3$ only has degrees of freedom on edges and vertices. Since there are zero vertices and only one edge not on the boundary of the refined coarse tetrahedron (see Figure 6.4), we have $\dim(\tilde{V}_h) = 3$. On the other hand, the pressure space satisfies $\dim(\tilde{Q}_h) = 7$ (one dimension is fixed by the nullspace). The local solve can therefore not be well-posed.

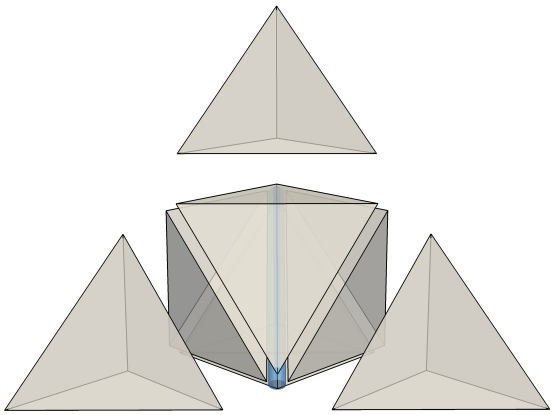


Figure 6.4: The uniform refinement of a coarse mesh tetrahedron yields eight fine mesh tetrahedra. Only the edge highlighted in blue does not lie entirely on the boundary of the tetrahedron.

Increasing the degree of the velocity space to piecewise cubic polynomials, i.e. choosing the element pair $[\mathbb{P}_3]^3-\mathbb{P}_0$, introduces additional degrees of freedom on the facets and results in a stable element pair. However, this element is extremely expensive while being suboptimal by two orders for the velocity. Alternatively, Bernardi & Raugel [BR85b; BR85a] suggest enriching the piecewise linear velocity space with bubble functions on each facet[1]. While it is only necessary to add a single bubble function for the normal component of the velocity on each facet, this adds significant complexity to the implementation as these functions are not affine equivalent; they require a Piola transform to preserve the normal orientation. This means that the basis functions associated with vertices and those associated with facets need to be pulled-back differently, complicating the implementation. For this reason we choose instead to enrich the space with facet bubbles for all three components of the velocity, obtaining the $[\mathbb{P}_1 \oplus B_3^F]^3-\mathbb{P}_0$ element. As can be seen in Figure 6.5, this results in an element with significantly fewer degrees of freedom than $[\mathbb{P}_3]^3-\mathbb{P}_0$. We also show the $[\mathbb{P}_2 \oplus B_3^F]^3$ element in Figure 6.5. It turns out that this element satisfies a particular property that is useful in the prolongation that we will discuss later.

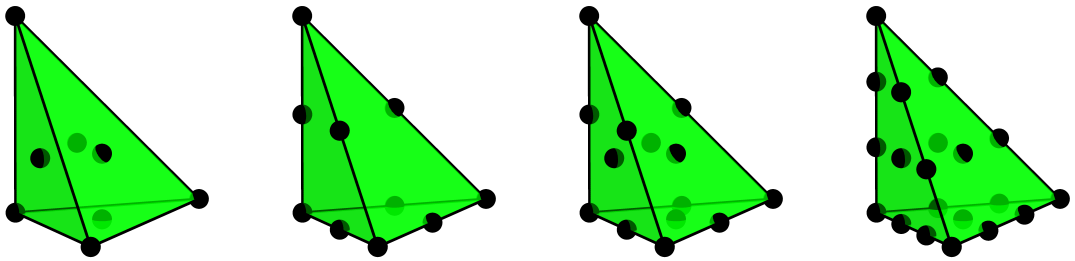


Figure 6.5: The $\mathbb{P}_1 \oplus B_3^F$, $\mathbb{P}_2$, $\mathbb{P}_2 \oplus B_3^F$ and $\mathbb{P}_3$ elements.

Though the $[\mathbb{P}_1 \oplus B_3^F]^3-\mathbb{P}_0$ element satisfies the inf-sup condition, it introduces a different difficulty: it violates $V_H \subset V_h$. The nonnestedness is demonstrated in Figure 6.6; a coarse bubble cannot be interpolated exactly by functions in V_h. In particular, this means that the flux across coarse grid faces is not preserved, hence violating condition (5.58) required for the robust prolongation operator. A brief calculation shows that every coarse grid bubble is interpolated by four fine grid bubbles: one with coefficient 1, the other three with coefficient 1/2. From this it follows immediately that the integral of the prolonged bubble is equal to $(1+3\cdot\frac{1}{2})/4 = \frac{5}{8}$ of the integral of the coarse bubble. Hence, when using a hierarchical basis, since the

[1]The bubble function on each facet is the product of the barycentric coordinates that are nonzero on that facet.

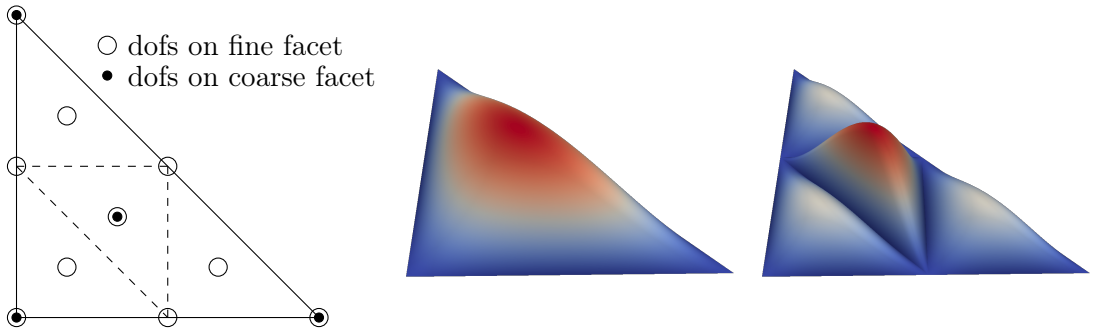


Figure 6.6: Left: Degrees of freedom on the facet of a coarse cell and its refinement. Middle: Bubble function on a coarse facet. Right: Prolongation of a bubble function.

piecewise linear basis functions are prolonged exactly we can obtain a prolongation that satisfies (5.58) by simply multiplying the coefficients of the fine grid bubble functions by 8/5.

Lemma 6.4.
Let $I_h^{\mathbb{P}_1}, I_h^{\mathbb{P}_1 \oplus B_3^F} : C^0 \to V_h$ be the standard interpolation operators associated with the $\mathbb{P}_1$ and the $\mathbb{P}_1 \oplus B_3^F$ finite elements. For $\mathbf{v} \in C^0(\Omega)$, define the interpolation

$$P_h(\mathbf{v}) = I_h^{\mathbb{P}_1}\mathbf{v} + \gamma I_h^{\mathbb{P}_1 \oplus B_3^F}(\mathbf{v} - I_h^{\mathbb{P}_1}\mathbf{v}) \tag{6.39}$$

where $\gamma = \frac{5}{8}$. Then

$$\begin{aligned} \|\mathbf{v} - P_h\mathbf{v}\|_0 &\preceq h^2\|\mathbf{v}\|_2 \quad \forall \mathbf{v} \in H^2, \\ \|P_h\mathbf{v}_H\|_1 + h^{-1}\|\mathbf{v}_H - P_h\mathbf{v}_H\|_0 &\preceq \|\mathbf{v}_H\|_1 \quad \forall \mathbf{v}_H \in V_H. \end{aligned} \tag{6.40}$$

Furthermore, for any $\mathbf{v}_H \in V_H$ and $q_H \in Q_H$, it holds

$$\int_\Omega (\nabla \cdot (P_h\mathbf{v}_H)) q_H \,\mathrm{d}x = \int_\Omega (\nabla \cdot \mathbf{v}_H) q_H \,\mathrm{d}x. \tag{6.41}$$

Proof. The first statement in (6.40) follows from the Bramble Hilbert Lemma and a scaling argument since P_h is invariant on piecewise affine functions [Cia02, Theorem 3.1.6]. To prove the second statement, one follows the argument of [Zha90, Proposition 2.2] to prove that the approximation property holds for $I_h^{\mathbb{P}_1}$ and $I_h^{\mathbb{P}_1 \oplus B_3^F}$. The idea is to construct Scott-Zhang [SZ90] operators (for which we know that optimal approximation holds) that are equivalent to $I_h^{\mathbb{P}_1}$ and $I_h^{\mathbb{P}_1 \oplus B_3^F}$ on the finite element space. We then have for $k \in \{0, 1\}$

$$\begin{aligned} &h^{k-1}\|P_h\mathbf{v}_H\|_k \\ \preceq &h^{k-1}\|I_h^{\mathbb{P}_1}\mathbf{v}_H - \mathbf{v}_H\|_k + h^{k-1}\|I_h^{\mathbb{P}_1 \oplus B_3^F}(\underbrace{\mathbf{v}_H - I_h^{\mathbb{P}_1}\mathbf{v}_H}_{\in V_H})\|_k \\ \preceq &\|\mathbf{v}_H\|_1 + \|\mathbf{v}_H - I_h^{\mathbb{P}_1}\mathbf{v}_H\|_1 \\ \preceq &\|\mathbf{v}_H\|_1. \end{aligned} \tag{6.42}$$

The last statement follows by the discussion preceding the Lemma. □

The robust prolongation operator can then be built in the same manner as explained in Section 6.1.1. To demonstrate the need for the rescaling of the facet bubble function, we show the residual of the outer flexible GMRES iteration for the linear solve in the first Newton step at Re = 10 for a lid-driven cavity problem (see section 6.4.5 for details) in Table 6.1. Without modifying the prolongation of the facet bubbles, we observe no convergence.

Iteration	Residual with bubble scaling	Residual without bubble scaling
0	3.50	3.50
1	2.34×10^{-2}	3.50
2	2.04×10^{-2}	3.50
3	4.50×10^{-3}	3.50
4	6.79×10^{-4}	3.50
5	1.36×10^{-4}	3.49
6	1.33×10^{-4}	3.48
7	2.19×10^{-5}	3.48

Table 6.1: Residual of the outer flexible GMRES solver when employing the $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ element. It is necessary to modify the prolongation operator to achieve convergence with this element.

Lastly, we consider the $[\mathbb{P}_2 \oplus B_3^F]^3 - \mathbb{P}_0$ element. While it is also nonnested, it turns out that the interpolation is exact *on the facets* of each coarse cell and hence flux preserving. To see this, observe that the cubic facet bubble function is only quadratic on the newly introduced edges of a regularly refined facet, as they are parallel to the edges of the coarse facet and therefore one of the barycentric coordinates is constant. The coarse bubble function is therefore prolonged exactly. This means that despite the velocity spaces not being nested, the fluxes across coarse grid facets are preserved and one does not need a modification as just described for the $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ element. However, in our preliminary numerical experiments the simpler prolongation was outweighed by the cost of the larger number of degrees of degrees of freedom, and hence we use $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ for the numerical experiments in Section 6.4.

6.2.2 Smoothing

As in two dimensions, we consider subspaces $\{V_i\}$ given by velocity functions with support in the star of vertices $\{v_i\}$, where $\text{star}(v_i)$ denotes all cells attached to the vertex v_i. The proof for parameter robustness of the smoother in three dimensions is almost entirely analogous to the two dimensional case, with the difference that the

Fortin operator is now chosen to preserve integrals over facets and not edges. For a divergence-free vector field $\mathbf{u}$, the existence of a potential $\mathbf{\Phi}$ that satisfies the bound $\|\mathbf{\Phi}\|_2 \preceq \|\mathbf{u}\|_1$ is now guaranteed by [CM10]. However, we note that the bound was only proven for domains that are starlike with respect to some ball (i.e. in particular for convex domains).

A statement similar to Proposition 6.2 is harder to obtain as we cannot prove similar bounds to (6.36). Instead we refer to the proof for the multiplicative multigrid method in [Sch99a] that does not require these strong bounds.

6.3 Including the advection term

So far we have neglected the terms arising from the linearisation of the advection term. Since we apply a Newton linearisation, the top left block corresponds to solves of the form: find $\mathbf{u} \in V_h$ such that

$$(2\nu \mathrm{E}\mathbf{u}, \mathrm{E}\mathbf{v}) + (\mathbf{w} \cdot \nabla \mathbf{u}, \mathbf{v}) + (\mathbf{u} \cdot \nabla \mathbf{w}, \mathbf{v}) + \gamma(\Pi_{Q_h}(\nabla \cdot \mathbf{u}), \Pi_{Q_h}(\nabla \cdot \mathbf{v})) = \langle \mathbf{f}, \mathbf{v} \rangle \quad (6.43)$$

for all $\mathbf{v} \in V_h$, while the Picard linearisation yields: find $\mathbf{u} \in V_h$ such that

$$(2\nu \mathrm{E}\mathbf{u}, \mathrm{E}\mathbf{v}) + (\mathbf{w} \cdot \nabla \mathbf{u}, \mathbf{v}) + \gamma(\Pi_{Q_h}(\nabla \cdot \mathbf{u}), \Pi_{Q_h}(\nabla \cdot \mathbf{v})) = \langle \mathbf{f}, \mathbf{v} \rangle \quad (6.44)$$

for all $\mathbf{v} \in V_h$. It is well known that straightforward Galerkin discretisations of advection-dominated problems are oscillatory [BH82; Tur99; QV08; ESW14]. Several approaches have been developed to avoid these issues, for example by adding a small amount of artificial viscosity as in the case of Streamline Upwind/Petrov Galerkin (SUPG) or Galerkin Least Squares (GLS) or by enriching the space with bubble functions. In addition, it has been observed that mesh-dependent SUPG stabilisation is highly advantageous for multigrid smoothers on advection-dominated problems [Ram99; Tur99]. To apply SUPG stabilisation, we recall the strong form of the momentum residual given by

$$\mathcal{L}(\mathbf{u}, p) = -\nabla \cdot (2\nu \mathrm{E}\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathbf{f}. \quad (6.45)$$

Note that since we consider piecewise constant pressures, $\nabla p = 0$ and hence the residual only depends on $\mathbf{u}$, $\mathcal{L}(\mathbf{u}, p) = \mathcal{L}(\mathbf{u})$. SUPG stabilisation then corresponds to adding the following term to the weak form:

$$\mathrm{SUPG}(\mathbf{u}, \mathbf{v}) := \int_\Omega \delta(\mathbf{u}) \mathcal{L}(\mathbf{u}) \cdot \big((\mathbf{u} \cdot \nabla)\mathbf{v}\big) \, \mathrm{d}x. \quad (6.46)$$

Here $\delta(\mathbf{u})$ is a weighting function that should be small in regions where the flow is well-resolved and large where stabilisation is necessary. The particular form employed in this work is

$$\delta(\mathbf{u}) = \delta_d \left(\frac{4\|\mathbf{u}\|^2}{h^2} + \frac{144\nu^2}{h^4} \right)^{-1/2}, \tag{6.47}$$

with $\delta_d = 1$ in two dimensions and $\delta_d = 1/20$ in three dimensions. To the best of the author's knowledge this form was first suggested in [SHJ91, eq. (3.58)]. It is important to take account of the dependence of δ on the (unknown) solution $\mathbf{u}$ when taking the derivatives required by Newton's method; in this work, these derivatives are calculated automatically and symbolically by the Unified Form Language [Aln+14]. Note that the added terms only depend on $\mathbf{u}$ and $\mathbf{v}$, hence they only affect the top-left block of the Jacobian. This is crucial, as a pollution of the off-diagonal blocks would complicate the Schur complement approximation.

The Picard linearisation is easier to solve but sacrifices quadratic convergence of the nonlinear solver. Several authors have reported success with geometric multigrid for scalar analogues of (6.44) without the grad-div term, using a combination of line/plane relaxation and SUPG stabilisation [Ram99; OR04; WE06]. Olshanskii and Benzi [OB08] and Elman et al. [ELW03] apply preconditioners built on the Picard linearisation (6.44) to the Newton linearisation (6.43), with good results.

Numerical experiments indicated that the additive star iteration alone was not effective as a relaxation method for (6.43). (Benzi and Olshanskii [BO06] used a multiplicative star iteration with multiple directional sweeps, but we wished to avoid this as its performance varies with the core count in parallel.) We investigated the multiplicative composition of additive star iterations and plane smoothers, and while this led to a successful multigrid cycle, the plane smoothers were quite expensive (involving many 2D solves) and were also difficult to parallelise on arbitrary unstructured grids where the parallel decomposition does not divide into planes. While the additive star iteration alone is not effective as a relaxation for (6.43), we found that a few iterations of GMRES preconditioned by the additive star iteration remains contractive even for low viscosities. This point merits further analysis but will not be analysed further in this work. This relaxation method also has the advantage that it is easy to parallelise, with convergence independent of the parallel decomposition.

6.4 Numerical Results

6.4.1 Algorithm details

A graphical representation of the entire algorithm is shown in Figure 6.7. We employ simple continuation in Reynolds number as a globalisation device, as Newton's method is not globally convergent. To improve convergence, the `l2` line search method in PETSc [Bru+15], which aims to reduce the l^2 norm of the residual, is used.

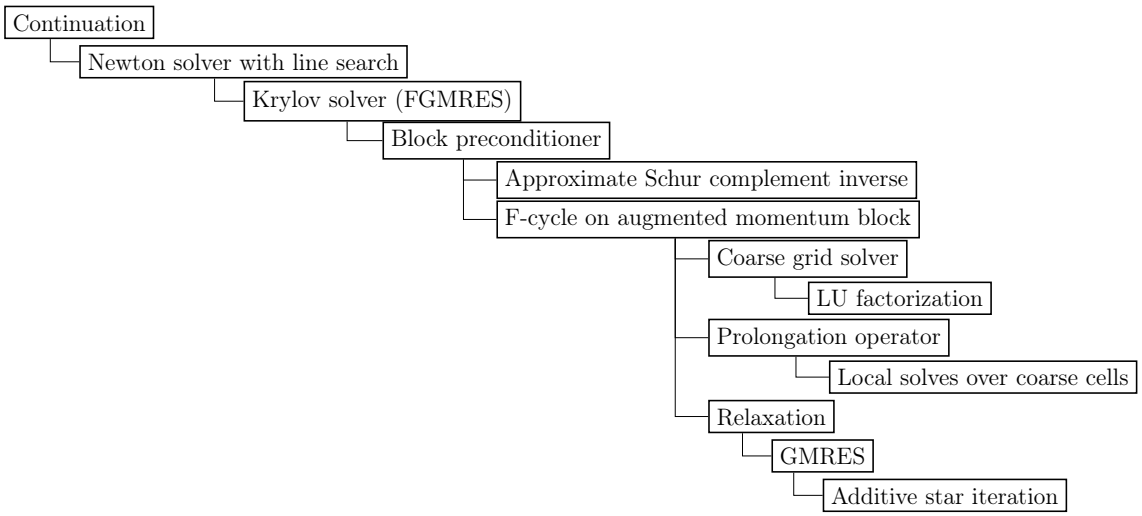


Figure 6.7: An outline of the algorithm for solving (5.1).

We use flexible GMRES [Saa93] as the outermost solver for the linearised Newton system, as we employ GMRES in the multigrid relaxation. If the pressure is only defined up to a constant, then the appropriate nullspace is passed to the Krylov solver and the solution is orthogonalized against the nullspace at every iteration. We use the full block factorisation preconditioner

$$P^{-1} = \begin{pmatrix} I & -\tilde{A}_\gamma^{-1} B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} \tilde{A}_\gamma^{-1} & 0 \\ 0 & \tilde{S}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -B\tilde{A}_\gamma^{-1} & I \end{pmatrix} \tag{6.48}$$

with approximate inner solves $\tilde{A}_\gamma^{-1}$ and $\tilde{S}^{-1}$ for the augmented momentum block and the Schur complement respectively. The diagonal, upper and lower triangular variants described in [MGW00; Ips01] also converge well, but these took longer runtimes in preliminary experiments.

We use one application of a full multigrid cycle [BL11, Figure 1.2] using the components described in Sections 6.1 and 6.2 for $\tilde{A}_\gamma^{-1}$. The problem on each level is constructed by rediscretisation; fine grid functions, such as the current iterate in the Newton scheme, are transferred to the coarse levels via injection. On each level

the SUPG stabilisation is performed with parameters corresponding to the mesh in question. For each relaxation sweep we perform 6 (in 2D) or 10 (in 3D) GMRES iterations preconditioned by the additive star iteration; at lower Reynolds numbers this can be reduced, but we found that these expensive smoothers represented the optimal tradeoff between inner and outer work at higher Reynolds numbers. The problem on the coarsest level is solved with the SuperLU_DIST sparse direct solver [LD03]. For scalability, the coarse grid solve is agglomerated onto a single compute node using PETSc's telescoping facility [May+16]. As all inner solvers are additive, the convergence of the solver is independent of the parallel decomposition (up to roundoff).

The solver is implemented in the Firedrake library which relies on PETSc for the solution of linear systems. PETSc provides implementations of all standard Krylov-methods, as well as interfaces to direct solvers and many preconditioners. The particular strength of the library lies in the ability to arbitrarily nest linear solvers, which we rely on heavily. To perform the local solves as part of the smoothing and the prolongation, a new preconditioner in PETSc called PCPATCH was developed. This was done in collaboration with Lawrence Mitchell and Patrick Farrell and extended the previous work by Kirby and Mitchell in [KM18]. PCPATCH relies on the DMPlex unstructured mesh component [KK05; KK09] in PETSc. The user provides a topological description of the patches (e.g. *star around each vertex*), PCPATCH then assembles the local problems, solves the systems and combines the solutions either additively or multiplicatively.

Differently from Benzi & Olshanskii, we also modify the restriction operator so that it is the adjoint to the modified prolongation. This relationship is assumed in the multigrid convergence proof by Schöeberl and we observed that it improves the robustness of the solver at very little extra cost.

6.4.2 Solving the top-left block only: nearly incompressible elasticity

We begin by studying the performance of the multigrid scheme when applied to the top-left block only, without any advection. This corresponds to solving the linear elasticity equations in the nearly incompressible regime. The model problem we study is given by: find $\mathbf{u} \in V_h$ such that

$$\int_\Omega \mathrm{E}\mathbf{u} : \mathrm{E}\mathbf{v}\,\mathrm{d}x + \hat{\gamma}\int_\Omega \Pi_{Q_h}(\nabla\cdot\mathbf{u})\Pi_{Q_h}(\nabla\cdot\mathbf{v})\,\mathrm{d}x = \int_\Omega \mathbf{f}\cdot\mathbf{v}\,\mathrm{d}x \tag{6.49}$$

for all $\mathbf{v} \in V_h$. We choose $\Omega = [0,1]^d$, homogeneous Dirichlet boundary conditions on $\partial\Omega$ and set $\mathbf{f} = (1,1)$ in two and $\mathbf{f} = (1,1,1)$ in three dimensions. Though this problem

is symmetric, we use the same configuration involving GMRES as described above, as this will be necessary once the nonsymmetric advection and stabilisation terms from the Navier–Stokes equations are added. The parameter $\hat{\gamma}$ will correspond to the ratio γ/ν used in the augmented Lagrangian preconditioner. The largest parameter studied, $\hat{\gamma} = 10^8$, corresponds to the case of $\nu = 10^{-4}$ and $\gamma = 10^4$.

We compare four different configurations of the solver: for the relaxation we compare the robust relaxation method against standard point Jacobi, and for the grid transfers we compare the robust scheme developed above to a standard grid transfer based on point interpolation. In addition, we also show iteration counts when using the boomeramg algebraic multigrid routine from the hypre library [FY02]. Results in two and three dimensions are shown in Tables 6.2 and 6.3. We clearly see that both robust smoothers and robust prolongation operators are crucial for well behaved iteration counts as $\hat{\gamma}$ is increased. The results are qualitatively similar in two and three dimensions, though iteration counts are larger in the latter case.

Refinements	Degrees of freedom	$\hat{\gamma}$							
		0	1	10	10^2	10^3	10^4	10^6	10^8
Robust relaxation & robust transfer									
1	578	3	3	4	4	4	4	4	4
2	2 178	4	4	4	5	5	5	5	5
3	8 450	4	4	4	5	5	5	5	5
4	33 282	3	4	4	4	4	4	4	4
Robust relaxation & standard transfer									
1	578	3	3	4	5	7	26	>100	>100
2	2 178	4	4	5	6	12	>100	>100	>100
3	8 450	4	4	5	7	20	>100	>100	>100
4	33 282	3	4	5	7	>100	>100	>100	>100
Jacobi relaxation & robust transfer									
1	578	5	5	6	16	81	>100	>100	>100
2	2 178	5	5	6	17	>100	>100	>100	>100
3	8 450	5	5	6	17	>100	>100	>100	>100
4	33 282	5	5	6	17	>100	>100	>100	>100
Jacobi relaxation & standard transfer									
1	578	5	5	7	20	>100	>100	>100	>100
2	2 178	5	5	7	24	>100	>100	>100	>100
3	8 450	5	5	7	26	>100	>100	>100	>100
4	33 282	5	5	7	27	>100	>100	>100	>100
Algebraic multigrid									
1	578	7	7	11	28	57	87	>100	>100
2	2 178	8	7	11	29	74	>100	>100	>100
3	8 450	7	7	12	34	93	>100	>100	>100
4	33 282	8	8	14	39	>100	>100	>100	>100

Table 6.2: Iteration counts in two dimensions for the $[\mathbb{P}_2]^2-\mathbb{P}_0$ element for five different geometric and algebraic multigrid variants. The geometric multigrid results are obtained with a 4×4 coarse grid.

Refinements	Degrees of freedom	$\hat{\gamma}$ 0	1	10	10^2	10^3	10^4	10^6	10^8
Robust relaxation & robust transfer									
1	21 771	3	4	5	7	8	8	8	8
2	166 803	5	5	6	9	9	10	10	10
3	1 305 891	6	6	7	9	11	11	11	11
Robust relaxation & standard transfer									
1	21 771	3	4	5	7	10	16	>100	>100
2	166 803	5	5	6	9	13	42	>100	>100
3	1 305 891	6	6	8	10	14	>100	>100	>100
Jacobi relaxation & robust transfer									
1	21 771	6	6	7	15	68	>100	>100	>100
2	166 803	8	8	9	18	>100	>100	>100	>100
3	1 305 891	9	9	10	19	>100	>100	>100	>100
Jacobi relaxation & standard transfer									
1	21 771	6	6	7	14	79	>100	>100	>100
2	166 803	9	9	9	17	>100	>100	>100	>100
3	1 305 891	9	9	10	19	>100	>100	>100	>100
Algebraic multigrid									
1	21 771	13	15	12	32	>100	>100	>100	>100
2	166 803	14	17	13	32	>100	>100	>100	>100
3	1 305 891	15	17	14	34	>100	>100	>100	>100

Table 6.3: Iteration counts in three dimensions for the $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ element for five different geometric and algebraic multigrid variants. The geometric multigrid results are obtained with a $4 \times 4 \times 4$ coarse grid.

6.4.3 Solver verification with the method of manufactured solutions

In order to verify the implementation and the convergence of the $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ element we employ the method of manufactured solution. We start by considering the pressure and velocity field proposed in [STH89], which is rescaled to the $[0,2]^2$ square. This results in $\mathbf{u} = (u_1, u_2)$ with

$$\begin{aligned} u_1(x,y) &= \frac{1}{4}(x-2)^2 x^2 y \left(y^2-2\right) \\ u_2(x,y) &= -\frac{1}{4} x \left(x^2-3x+2\right) y^2 \left(y^2-4\right) \\ \tilde{p}(x,y) &= \frac{xy\left(3x^4-15x^3+10x^2y^2-30x\left(y^2-2\right)+20\left(y^2-2\right)\right)}{5\mathrm{Re}} \\ &\quad -\frac{1}{128}(x-2)^4 x^4 y^2 \left(y^4-2y^2+8\right) \\ p(x,y) &= \tilde{p} - \frac{1}{4}\int_{[0,2]^2} \tilde{p}(x,y)\,\mathrm{d}x = \tilde{p} + \frac{1408}{33075} - \frac{8}{5\mathrm{Re}}. \end{aligned} \tag{6.50}$$

As we are primarily interested in the three dimensional case, we extend the vector field into the z dimension via $\mathbf{u}(x,y,z) = (u_1(x,y), u_2(x,y), 0)$ and consider the domain $\Omega = [0,2]^3$. The pressure remains the same as in two dimensions. In order to avoid the flow being perfectly aligned with the mesh, we consider an unstructured coarse grid.

To demonstrate that the error convergence is independent of γ, we run the solver for values $\gamma = 1$ and $\gamma = 10^4$. Figure 6.8 shows the error between the computed velocity and pressure and their known analytical solutions for $\mathrm{Re} = 10$, $\mathrm{Re} = 100$ and $\mathrm{Re} = 500$. A large value of γ does not impact the convergence.

Due to the $\mathbb{P}_0$ discretisation for the pressure we expect, and see, first order convergence of the pressure and of the gradient of the velocity. However, though we would hope to see second order convergence of the velocity in the L^2 norm, for the case of $\mathrm{Re} = 500$ we observe only first order convergence on coarse meshes. This may be explained by the added SUPG stabilisation not being adjoint consistent, preventing the use of the typical Aubin-Nitsche duality argument [Arn+02], [OD09, p. 3508], [Hic+13]. Only for sufficiently fine meshes the convergence order improves, as the weight in the SUPG stabilisation (6.47) decreases with the mesh size.

Comparing the velocity error for fixed mesh size, we observe that the error increases significantly as the Reynolds number is increased from $\mathrm{Re} = 10$ to $\mathrm{Re} = 500$. This behaviour is expected for a discretisation that does not enforce the divergence constraint exactly. In the next chapter, we will extend the solver to an exactly divergence-free discretisation and see Reynolds-robust errors for the velocity.

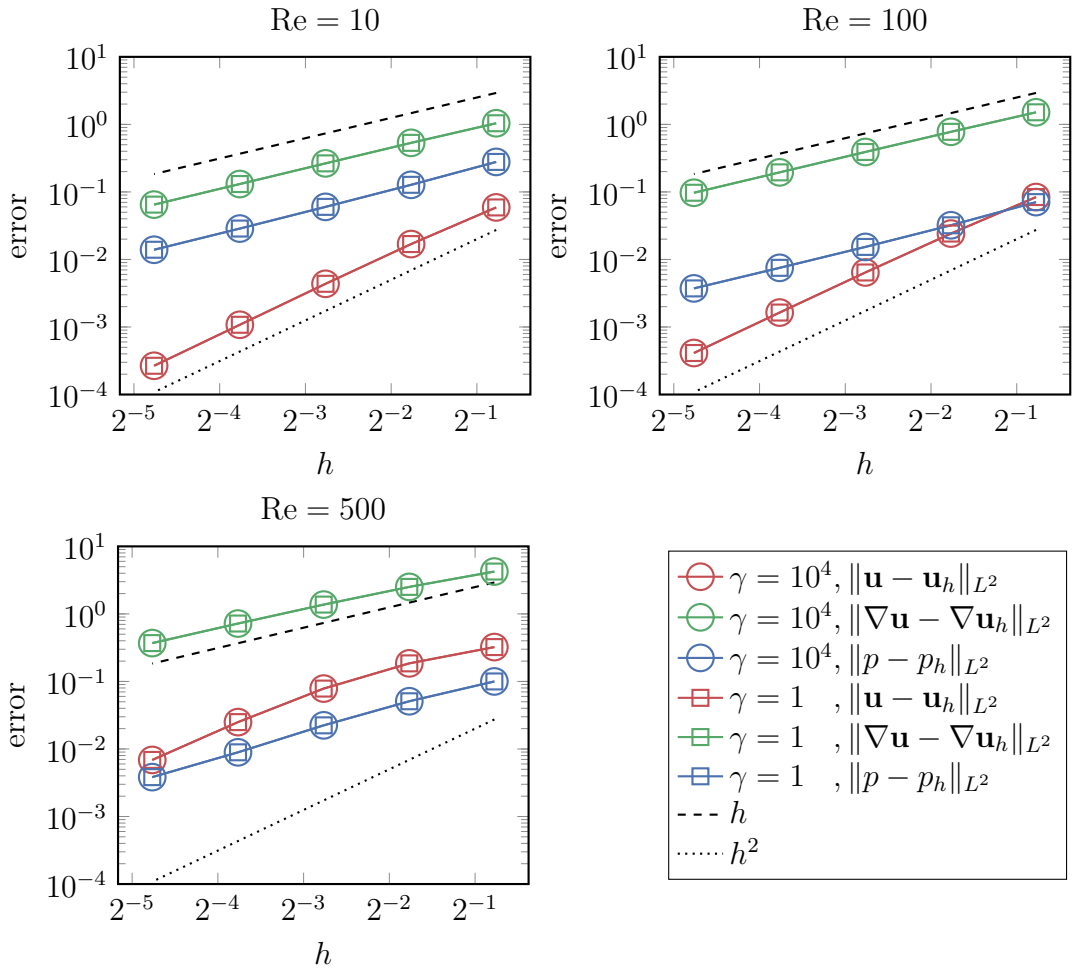


Figure 6.8: Convergence of the computed velocity and pressure field as the mesh is refined.

6.4.4 Two-dimensional experiments

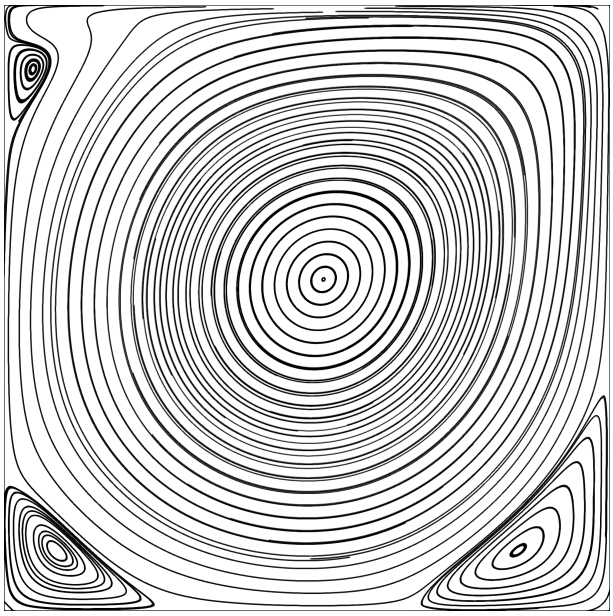


Figure 6.9: Streamlines for the two dimensional lid-driven cavity problem at $\mathrm{Re} = 2500$. The domain is given by the $[0, 2] \times [0, 2]$ square. The boundary condition on the top is given by a horizontal velocity field $\mathbf{u}(x, y) = (x^2(2 - x)^2, 0)$ and the other boundaries are equipped with a no-slip condition.

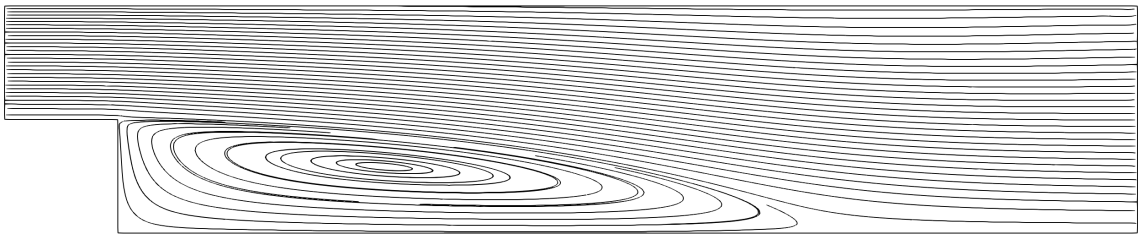


Figure 6.10: Streamlines for the two dimensional backwards-facing step problem at $\mathrm{Re} = 200$. The domain is given by $([0, 10] \times [0, 2]) \setminus ([0, 1) \times [0, 1))$. The inflow condition at the top-left boundary is given by a horizontal velocity field $\mathbf{u}(x, y) = (4(2 - y)(y - 1), 0)$, a natural outflow condition is enforced on the right and the other boundaries are equipped with a no-slip condition.

We consider two representative benchmark problems: the regularised lid-driven cavity and backward-facing step problems, shown in Figures 6.9 and 6.10, and described in detail in [ESW14, Examples 8.1.2 and 8.1.3]. For each experiment, we fix a coarse grid and vary the number of refinements to vary the size of the problem under consideration; all refinements are used in the multigrid iteration, to ensure that the

convergence does not deteriorate as more levels are employed. We employ the $[\mathbb{P}_2]^2-\mathbb{P}_0$ element for all two dimensional experiments. To investigate the performance of the solver with Reynolds number, the problem is first solved for $\mathrm{Re} = 10$, then $\mathrm{Re} = 100$, and then in steps of 100 until $\mathrm{Re} = 10\,000$, with the solution for the previous value of Re used as initial guess for the next. The initial guess for $\mathrm{Re} = 10$ is obtained by solving the Stokes equations. The augmented Lagrangian parameter is set to $\gamma = 10^4$ in these and all subsequent experiments.

The linear solves are terminated with an absolute tolerance of 10^{-10} in the ℓ_2-norm and a relative tolerance of 10^{-9}. The nonlinear solves are terminated with an absolute tolerance of 10^{-8} and a relative tolerance of 10^{-9}. As each outer iteration of the Krylov method does a *fixed* amount of work (i.e. all subproblems are solved with a fixed number of iterations, not to a specified tolerance), the solver scales well with mesh size and Reynolds number if the iteration counts remain approximately constant.

For comparison, we solve the same problems using the reference implementations of the PCD and LSC preconditioners in version 3.5 of IFISS [ERS14], up to $\mathrm{Re} = 1\,000$, as IFISS does not employ stabilisation of the advection term. For both of these preconditioners we use the variant that takes corrections for the boundary conditions into account and we solve the inner problems in the Schur complement approximation using an algebraic multigrid solver. We employ the hybrid strategy suggested by [ESW14, p. 391] that uses a single sweep of ILU(0) on the finest level and two iterations of point-damped Jacobi for pre- and post-smoothing on all coarsened levels. A relative tolerance of 10^{-6} is set for the Krylov solver and an absolute tolerance of 10^{-8} for the Newton solver.

We begin by considering the regularised lid-driven cavity problem. Iteration counts using the PCD and LSC preconditioners are shown in Table 6.4. Using PCD the iteration counts increase by about a factor of approximately 6 as the Reynolds number is increased from $\mathrm{Re} = 10$ to $\mathrm{Re} = 1\,000$. For LSC we observe a larger increase in iteration counts: with growth varying between factors of 6 and 10 for different meshes.

The coarse grid used for the augmented Lagrangian preconditioner is the 16×16 grid of triangles of negative slope. The results are shown in Table 6.5. We observe at most a doubling of iteration counts as the Reynolds number is increased from $\mathrm{Re} = 10$ to $\mathrm{Re} = 1\,000$. As the Reynolds number is increased further to $\mathrm{Re} = 10\,000$ iteration counts increase further, by a factor of 2 to 3. Overall we observe that the preconditioner becomes more robust as the mesh is refined and the flow is resolved better. For the finest mesh iteration counts less than quadruple over a range of 4

$1/h$	Degrees of freedom	Reynolds number		
		10	100	1 000
2^6	1.25×10^4	24.5/22.5	42.0/49.3	157.0/205.7
2^7	4.97×10^4	25.5/21.0	42.7/43.3	149.0/207.3
2^8	1.98×10^5	26.0/23.0	44.0/38.0	137.0/180.0

Table 6.4: Average number of outer Krylov iterations per Newton step for the 2D regularised lid-driven cavity problem with PCD/LSC preconditioner.

orders of magnitude for the Reynolds number. Finally, we study the nonregularised

Refinements	Degrees of freedom	Reynolds number				
		10	100	1 000	5 000	10 000
1	1.05×10^4	2.50	4.00	5.00	8.00	15.00
2	4.15×10^4	2.50	3.33	5.00	8.50	9.50
3	1.65×10^5	2.50	3.33	4.33	7.50	9.00
4	6.57×10^5	2.50	3.00	3.67	8.00	8.50

Table 6.5: Average number of outer Krylov iterations per Newton step for the 2D regularised lid-driven cavity problem with augmented Lagrangian preconditioner.

version of the lid-driven cavity. Here the boundary condition at the top of the domain is replaced with a constant value of $\mathbf{u} = (1, 0)^T$, leading to a discontinuity of the boundary conditions in the two top corners. This makes the problem more challenging. The iteration counts are shown in Table 6.6: the behaviour is qualitatively similar to the regularised case, though iteration counts at high Reynolds number are slightly larger.

Refinements	Degrees of freedom	Reynolds number				
		10	100	1 000	5 000	10 000
1	1.05×10^4	3.00	4.67	5.67	9.50	14.00
2	4.15×10^4	2.50	3.67	6.00	10.00	13.00
3	1.65×10^5	2.50	3.33	5.67	9.00	11.50
4	6.57×10^5	2.50	3.00	4.67	9.00	11.50

Table 6.6: Average number of outer Krylov iterations per Newton step for the 2D nonregularised lid-driven cavity problem with augmented Lagrangian preconditioner.

For the backward-facing step we observe that the performance is dependent on the resolution of the coarse grid. We consider two experiments, one starting with a coarse grid consisting of 6 475 vertices and 12 546 elements (labeled A) and one

consisting of 25 194 vertices and 49 583 elements (labeled B). Both triangular meshes were generated with Gmsh [GR09]. For mesh A, we observe that the iteration counts for large Reynolds numbers show the solver degrades somewhat as the mesh is refined, see Table 6.7. Using the finer coarse grid B alleviates this problem. The bottom half of Table 6.7 shows that iteration counts only marginally grow as we increase from $\mathrm{Re} = 10$ to $\mathrm{Re} = 10\,000$. Since we did not see robustness issues using very coarse grids for the symmetric example in Section 6.4.2, this suggests that the scheme requires a 'fine enough' coarse grid solve to capture the advection terms adequately.

Refinements	Degrees of freedom	Reynolds number				
		10	100	1 000	5 000	10 000
coarse grid A						
1	2.53×10^5	2.67	3.00	4.00	11.00	10.00
2	1.01×10^6	2.67	2.25	3.50	8.00	18.50
3	4.02×10^6	3.50	3.33	3.50	7.50	10.50
4	1.61×10^7	6.50	3.67	2.50	6.50	18.00
coarse grid B						
1	9.95×10^5	2.33	2.25	3.50	3.50	5.50
2	3.97×10^6	3.00	2.67	3.50	3.50	4.50
3	1.59×10^7	3.00	3.33	2.50	3.50	4.00

Table 6.7: Average number of outer Krylov iterations per Newton step for the 2D backward-facing step problem for two different coarse grids with augmented Lagrangian preconditioner.

The results for PCD and LSC on the backwards-facing step are shown in Table 6.8. The iteration counts approximately treble as we increase from $\mathrm{Re} = 10$ to $\mathrm{Re} = 1\,000$.

$1/h$	Degrees of freedom	Reynolds number		
		10	100	1 000
2^6	5.91×10^4	27.5/21.5	38.0/35.0	167.0/152.3
2^7	2.36×10^5	29.0/23.0	38.0/32.3	122.3/114.0
2^8	9.38×10^5	29.0/31.5	37.5/33.0	106.0/97.3

Table 6.8: Average number of outer Krylov iterations per Newton step for the 2D backwards-facing step problem with PCD/LSC preconditioner.

6.4.5 Three-dimensional experiments

The lid-driven cavity and backward-facing step problems can both be extended to three dimensions in a natural way. For the lid-driven cavity, we consider the cube $\Omega = [0, 2]^3$ with no-slip boundary conditions on all sides apart from the top boundary $\{y = 2\}$. On the top boundary we enforce $\mathbf{u}(x, y, z) = (x^2(2-x)^2z^2(2-z)^2, 0, 0)^T$. The three dimensional backwards-facing step is given by $\Omega = (([0, 10] \times [1, 2]) \cup ([1, 10] \times [0, 1])) \times [0, 1]$. We enforce the inflow condition $\mathbf{u}(x, y, z) = (4(2 - y)(y - 1)z(1 - z), 0, 0)^T$ on the left boundary $\{x = 0\}$, a natural outflow condition on the right boundary $\{x = 10\}$, and no-slip boundary conditions on the remaining boundaries.

Two aspects of the solver were modified compared to the version used in two dimensions. First, we observe that reducing the size of the SUPG stabilisation by a factor of 1/20 improves convergence significantly. Secondly, all tolerances in the linear and nonlinear solvers are relaxed to 10^{-8}. The three-dimensional experiments were both run for $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ discretisations of up to one billion degrees of freedom on ARCHER, the UK national supercomputer. Images of the solutions are shown in Figures 6.11 and 6.12.

Refinements	Degrees of freedom	Reynolds number				
		10	100	1 000	2 500	5 000
1	2.13×10^6	4.50	4.33	4.50	4.00	3.50
2	1.69×10^7	4.00	4.33	4.50	4.00	5.00
3	1.35×10^8	4.00	3.67	3.50	3.00	8.00
4	1.08×10^9	3.50	3.67	3.00	2.50	5.00

Table 6.9: Average number of outer Krylov iterations per Newton step for the 3D regularised lid-driven cavity problem.

Refinements	Degrees of freedom	Reynolds number				
		10	100	1 000	2 500	5 000
1	2.56×10^6	5.00	4.67	4.00	4.50	4.00
2	2.01×10^7	5.00	4.67	5.00	4.00	8.00
3	1.60×10^8	5.50	4.67	3.50	3.00	5.00
4	1.27×10^9	5.50	3.67	3.00	3.00	4.00

Table 6.10: Average number of outer Krylov iterations per Newton step for the 3D backwards-facing step problem.

As for the two-dimensional case, we see only very little variation of the iteration counts with Reynolds number over this range.

To stress the solver further, the lid-driven cavity with 2 refinements (1.7×10^7 degrees of freedom) was run until failure. Iteration counts remain stable until $\mathrm{Re} = 7\,000$, then begin to increase, with eventual failure of convergence at $\mathrm{Re} = 7\,700$.

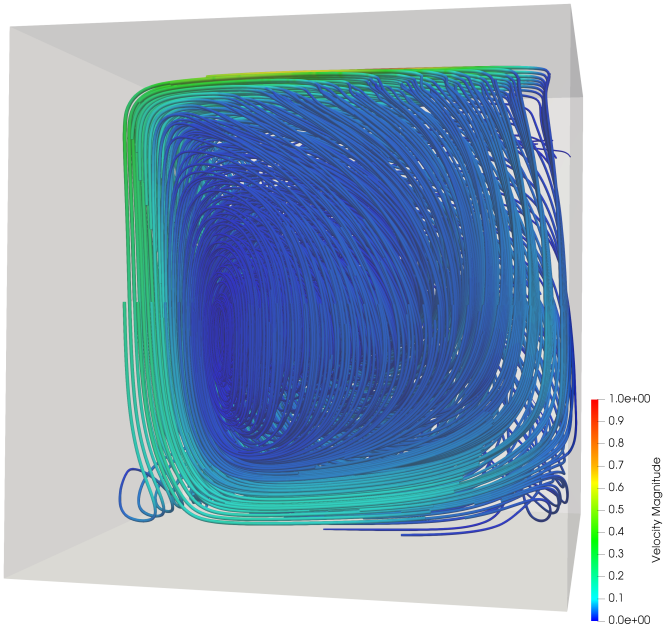


Figure 6.11: Interior view of the streamtubes of the 3D lid-driven cavity at $\mathrm{Re} = 5\,000$. The color denotes speed.

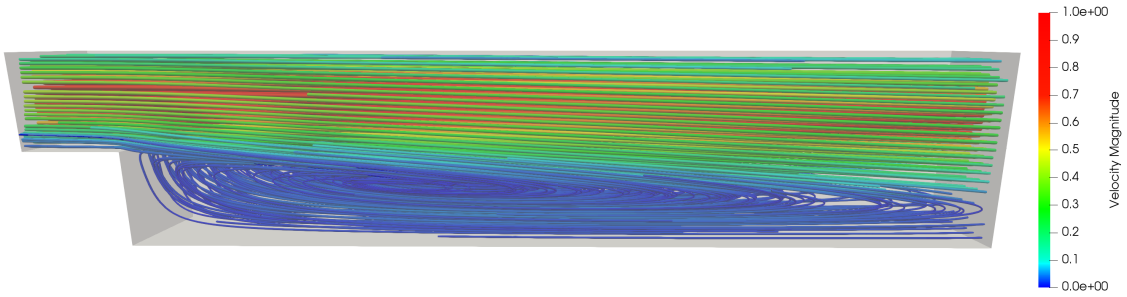


Figure 6.12: Interior view of the streamtubes of the 3D backwards-facing step at $\mathrm{Re} = 1\,000$. The color denotes speed.

6.4.6 Computational performance

Having seen that the algorithmic scalability of the solver is good, with well-controlled iteration counts, we now consider the computational performance.

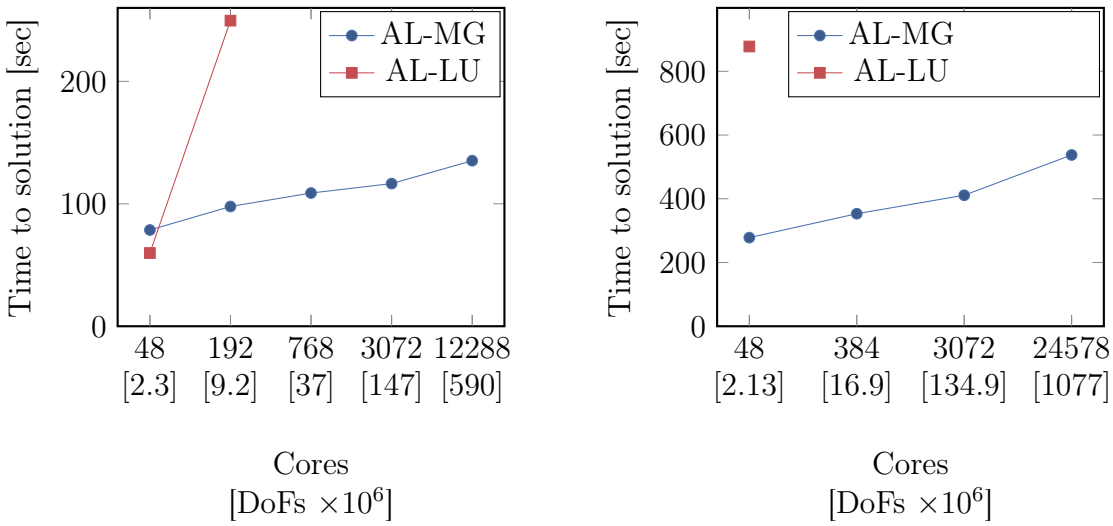
Scaling

In Figures 6.13a and 6.13b we perform a weak scaling analysis: as the problem size is increased the number of MPI processes is increased simultaneously so that the number of degrees of freedom per process stays constant. We show the aggregate time to solution for the first three continuation steps. Perfect weak scaling corresponds to a constant time to solution as the problem size is increased, however, since the complexity of a multigrid cycle is only asymptotically linear, we expect some increase in runtime.

In two dimensions we see an increase in runtime by $\sim 72\%$ as the problem is increased by a factor of 256. In three dimensions we see approximately a doubling in runtime as the problem size is increased by a factor of 512. We attribute this increase to two factors: firstly, the coarse grid solve is performed on a single node, hence the matrix needs to be gathered onto that node which introduces significant communication. The alternative would be a distributed solve, but this was slower in our experiments due to communication during the solve. Secondly, there is a load imbalance in the default mesh distribution. Standard finite element mesh partitioning schemes result in a well-balanced partition of cells. However, for the block Jacobi smoother to have perfect load balance the number of vertices (around which the blocks are built) owned by each process must also be equal. We notice this problem in particular in three dimensions. Calculating the ratio between the largest number of vertices on a core and the average number of vertices, we observe that this ratio increases gradually from for the smallest problem ~ 1.28 to ~ 1.95 for the biggest problem. The development of a mesh partitioning scheme that balances both cells and vertices is ongoing work.

Finally, we remark that both in two and three dimensions we observe a jump in runtime for the largest problem. This can be understood by studying the network topology of the ARCHER supercomputer: a so-called *group* is formed of 376 nodes and within such a group every node is interconnected, enabling fast communication. Since a node has 24 cores, this means that problems running on up to 9024 cores fit in a single group. Past that, the communication cost increases due to the extra cost of communication between groups over optical interconnect.

In addition, we compare the runtime to that of the ideal version of the preconditioner where we use the MUMPS [Ame+00] distributed memory sparse direct solver for the top-left block (denoted by AL-LU in the plot). While for smaller problems a direct solver is faster, we see a clear performance advantage of the multigrid scheme once the problem size exceeds a few million degrees of freedom.



(a) Weak scaling of the two-dimensional lid-driven cavity.

(b) Weak scaling of the three-dimensional lid-driven cavity.

Figure 6.13: Weak scaling of time to solution over the first three continuation steps. AL-LU: augmented Lagrangian preconditioner with sparse direct solve for the top-left block. AL-MG: augmented Lagrangian preconditioner with multigrid scheme for top-left block.

Runtime comparison to SIMPLE

We compare the solver to an implementation of SIMPLE [Pat80, Section 6.7] in the same software framework. We select the lid-driven cavity in two dimensions with three refinements (1.6×10^5 degrees of freedom) as a representative problem. The SIMPLE preconditioner is given by

$$P^{-1}_{\text{SIMPLE}} = \begin{pmatrix} I & -\operatorname{diag}(A)^{-1}B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} \tilde{A}^{-1} & 0 \\ 0 & \tilde{S}^{-1}_{\text{SIMPLE}} \end{pmatrix} \begin{pmatrix} I & 0 \\ -B\tilde{A}^{-1} & I \end{pmatrix}, \tag{6.51}$$

where

$$\tilde{S}_{\text{SIMPLE}} = -B^T \operatorname{diag}(A)^{-1} B \tag{6.52}$$

and no grad-div augmentation is employed. $\tilde{A}^{-1}$ is approximated by one full multigrid cycle of the ML algebraic multigrid library [Gee+06]; $\tilde{S}^{-1}_{\text{SIMPLE}}$ is approximated with one V cycle of ML[2].

The results for several continuation steps are shown in Table 6.11. The computations were performed in serial. Each SIMPLE iteration is approximately 20 times faster than an augmented Lagrangian iteration, but the lower cost per iteration is outweighed by the greater number of iterations required.

[2]For fairness, we do not use exact inner solves, since our solver also does not use exact inner solves. Of the algebraic multigrid libraries available in PETSc, ML performed the best.

Reynolds number	Augmented Lagrangian		SIMPLE	
	Total iterations	Time (min)	Total iterations	Time (min)
10	6	0.14	827	1.00
50	6	0.15	1137	1.38
100	7	0.17	1409	1.70
150	7	0.17	1525	1.84
200	7	0.17	1544	1.86

Table 6.11: Iteration count and runtime comparison against the SIMPLE preconditioner.

Code availability

The subspace correction preconditioner has been incorporated into PETSc and Firedrake. The code for the Navier–Stokes solver and the numerical experiments in this chapter can be found at `https://github.com/florianwechsung/alfi/`. For reproducibility, all major Firedrake components as well as the code used to run these examples have been archived on Zenodo [Fir19; Wec19b].

Chapter 7

Augmented Lagrangian preconditioning for divergence-free finite element pairs

In the previous chapter we studied a preconditioner for the Navier–Stokes equations that sees only very modest iteration growth as the Reynolds number is increased. However, the scheme is tightly coupled to the use of piecewise constant elements for the pressure and does not enforce the divergence constraint exactly. As a consequence, in the numerical experiments we observed that the velocity error increased with the Reynolds number.

This problem can be illustrated by studying a priori error estimates for the Stokes equations: the discrete solution $\mathbf{u}_h \in V_h$ to the Stokes problem can be shown to satisfy [Joh+17, (3.5)]

$$\|\nabla(\mathbf{u} - \mathbf{u}_h)\|_{L^2(\Omega)} \leq 2 \inf_{\tilde{\mathbf{u}}_h \in \mathcal{N}_h} \|\nabla(\mathbf{u} - \tilde{\mathbf{u}}_h)\|_{L^2(\Omega)} + \nu^{-1} \inf_{q_h \in Q_h} \|p - q_h\|_{L^2(\Omega)}, \tag{7.1}$$

where $\mathcal{N}_h$ is the space of discretely divergence-free vector fields. For large values of ν^{-1} the second term in this estimate eventually dominates.

Though the spaces V and Q satisfy $\operatorname{div}(V) = Q$, many of the arguably most popular finite element discretisation for the Stokes and Navier–Stokes equations such as the Taylor-Hood, the Mini or the $[\mathbb{P}_2]^2$–$\mathbb{P}_0$ element, do not satisfy the finite dimensional analogue, that is $\operatorname{div}(V_h) = Q_h$. The reason for this is that the inf-sup condition requires a certain richness of the velocity space, that often leads to $\operatorname{div}(V_h) \supsetneq Q_h$, implying that the divergence constraint is not enforced exactly. This relaxation in turn results in the appearance of the pressure in the velocity error estimate.

For choices $V_h \times Q_h$ that satisfy $\operatorname{div}(V_h) = Q_h$, one can improve the error estimate

for the velocity to be independent of the pressure:

$$\|\nabla(\mathbf{u}-\mathbf{u}_h)\|_{L^2(\Omega)} \leq 2 \inf_{\tilde{\mathbf{u}}_h \in \mathcal{N}_h} \|\nabla(\mathbf{u}-\tilde{\mathbf{u}}_h)\|_{L^2(\Omega)}. \tag{7.2}$$

This property is known as *pressure robustness*. Other advantages of such discretisations are discussed in [Joh+17]. The analysis is more complicated for the Navier–Stokes equations but progress has recently been made in [ALM18]. We also mention that similar results are available for the time dependent case [LR19].

There are several choices of element pairs that satisfy our requirements for pressure robustness. They include $H(\mathrm{div}) - L^2(\Omega)$ discretisations, such as the Raviart-Thomas and Brezzi-Douglas-Marini families [CKS06; WY07]; (hybrid) DG schemes [CKS04; CCS05]; or H^1-conforming approaches such as the Scott-Vogelius $\mathbb{P}_k - \mathbb{P}_{k-1}^{\mathrm{disc}}$ pair. The former two options allow for arbitrary order approximations, but at the expense of more complex analysis: the discretisation of the momentum equation now requires penalty terms to weakly enforce H^1-conformity. To avoid these penalty terms and to remain H^1-conforming, we focus on the Scott-Vogelius element pair.

The remainder of this chapter is structured as follows: In Section 7.1 we discuss inf-sup stability of the Scott-Vogelius element. Stability is only guaranteed on meshes with a particular structure, resulting in a nonnested hierarchy. In Section 7.2 we develop prolongation and smoothing operators. The prolongation will need to handle the nonnestedness of the hierarchy and the subspaces in the smoother now need to capture exactly divergence-free functions. A stabilisation scheme to avoid oscillations in the advection dominated regime is discussed in Section 7.3 and numerical results are presented in Section 7.4.

7.1 Stability of the Scott-Vogelius element

The stability of the Scott-Vogelius element is a topic of ongoing research. In two dimensions, Scott & Vogelius proved [SV85] that the element is inf-sup stable for $k \geq 4$ if the mesh does not have so called *singular vertices*. In three dimensions, it was proven more recently in [Zha11a] that the element is stable for $k \geq 6$ on uniform meshes. The stability on general tetrahedral meshes continues to be an open question.

On barycentrically refined meshes, however, the pair is known to be stable for polynomial order $k = d$, see [Qin94, Section 4.6] for the 2D case and [Zha04] for the 3D case. If one is willing to consider the more complicated Powell-Sabin split, the order can be reduced further to $k = d - 1$ [Zha08; Zha11b]. The two refinement patterns are shown for the two dimensional case in Figure 7.1.

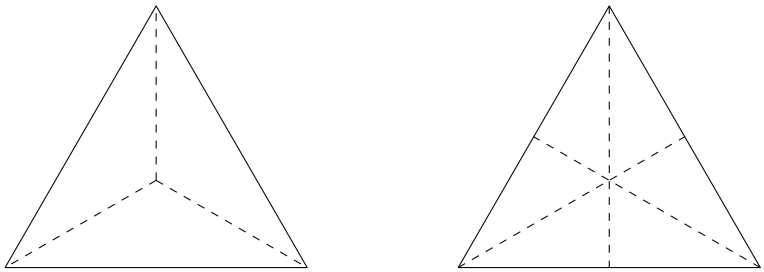


Figure 7.1: Barycentrically refined triangle, also known as Alfeld split, on the left and Powell-Sabin split on the right.

In this chapter we will consider the case of $k = d$ on barycentrically refined meshes, but the analysis applies in the same way to the Powell-Sabin split.

In the context of a multigrid scheme, the requirement for barycentrically refined elements has some implications for our mesh hierarchy. First, note that repeatedly barycentrically refining a mesh leads to degenerate triangles. Furthermore, it is not known whether regularly refining a mesh, on which an element pair with $k = d$ is stable, always results in a refined mesh for which stability is maintained. Consequently, the multigrid hierarchy has to be built differently. Given a domain Ω, we consider a simplicial mesh $\mathcal{M}_h = \{K^h\}$ with $\cup_{K^h \in \mathcal{M}_h} K^h = \overline{\Omega}$ and $(K_1^h)^\circ \cap (K_2^h)^\circ = \emptyset$. The elements $K^h \in \mathcal{M}_h$ will be referred to as the *macro cells*. For each level h, we obtain the mesh $\hat{\mathcal{M}}_h$ by barycentric refinement, that is, for each element $K^h \in \mathcal{M}_h$ we obtain $d+1$ many elements $\hat{K}_i^h$, $0 \leq i \leq d$ and

$$\hat{\mathcal{M}}_h = \{\hat{K}_i^h : 0 \leq i \leq d, K^h \in \mathcal{M}_h\}. \tag{7.3}$$

The function spaces on $\hat{\mathcal{M}}_h$ are then given by

$$V_h := \{\mathbf{v} \in H^1(\Omega; \mathbb{R}^d) : \mathbf{v}|_{\hat{K}} \in [P^d(\hat{K})]^d \ \forall \hat{K} \in \hat{\mathcal{M}}_h\}, \tag{7.4}$$

$$Q_h := \{q \in L^2(\Omega) : q|_{\hat{K}} \in P^{d-1}(\hat{K}) \ \forall \hat{K} \in \hat{\mathcal{M}}_h\}. \tag{7.5}$$

We construct the hierarchy as follows: we start with an initial coarse triangulation of the domain, given by $\mathcal{M}_H$. We obtain $\mathcal{M}_h$, $h = \frac{1}{2}H$, by uniform refinement of the initial mesh. Both $\mathcal{M}_H$ and $\mathcal{M}_h$ are then refined barycentrically to obtain $\hat{\mathcal{M}}_H$ and $\hat{\mathcal{M}}_h$. Note that though $\mathcal{M}_H$ and $\mathcal{M}_h$ form a nested hierarchy, this is not true for $\hat{\mathcal{M}}_H$ and $\hat{\mathcal{M}}_h$. This two-level approach canonically extends to many levels; a hierarchy of three levels is shown in Figure 7.2.

We will see in Section 7.2.2 that having a macro element structure at every level not only guarantees inf-sup stability, but is also crucial in order to define a robust relaxation method.

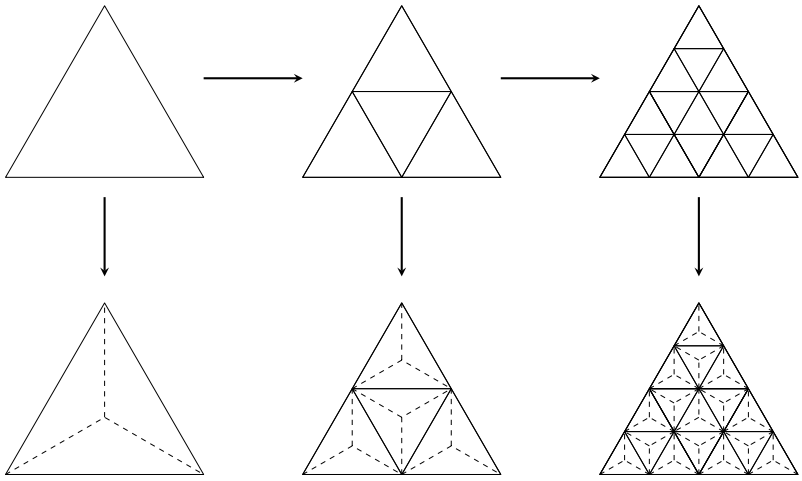


Figure 7.2: A three level barycentrically refined multigrid hierarchy.

7.2 Solving the top-left block

We proceed as in the previous chapter and begin by studying the augmented top-left block without the advection terms:

$$a_{h,\gamma}(\mathbf{u},\mathbf{v}) := a(\mathbf{u},\mathbf{v}) + \gamma c_h(\mathbf{u},\mathbf{v}), \tag{7.6}$$

where

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) &:= (2\nu \mathrm{E}\mathbf{u}, \mathrm{E}\mathbf{v}), \\ c_h(\mathbf{u},\mathbf{v}) &:= (\Pi_{Q_h}(\nabla\cdot\mathbf{u}), \Pi_{Q_h}(\nabla\cdot\mathbf{v})). \end{aligned} \tag{7.7}$$

We note that since $\mathrm{div}(V_h) \subset Q_h$, the projection Π_{Q_h} reduces to the identity mapping.

7.2.1 Robust prolongation

For nested function spaces, exact enforcement of the divergence simplifies the prolongation significantly. Discretely divergence-free functions on the coarse grid are exactly divergence-free, thus if the prolongation is the identity, they remain divergence-free on the fine grid. This property was implicitly used in [LWC09] and [WZ14] to build robust solvers for the elasticity equations using the Scott-Vogelius element with $k \geq 4$ on standard, nested mesh hierarchies in two dimensions. In our case, due to the barycentric refinement at each level, we do not have nested function spaces and hence the prolongation is not exact and a divergence-free function on the coarse grid may be prolonged to a function on the fine grid with nonzero divergence. However, we observe that interpolation is exact on the boundaries of coarse grid macro cells. This

means that flux across these boundaries is preserved and hence the divergence with respect to functions in

$$\tilde{Q}_H \coloneqq \{q \in L^2 : q \equiv \text{const on coarse grid macro cells } K \in \mathcal{M}_H\}, \tag{7.8}$$

is preserved, i.e. we have for the standard interpolation operator $P_H : V_H \to V_h$ that

$$(\nabla \cdot \mathbf{u}_H, \tilde{q}_H) = (\nabla \cdot (P_H \mathbf{u}_H), \tilde{q}_H) \quad \text{for all } \mathbf{u}_H \in V_H,\ \tilde{q}_H \in \tilde{Q}_H. \tag{7.9}$$

Hence requirement (5.58) of Proposition 5.7 is satisfied. To correct for any extra divergence gained due to interpolation inside a coarse grid macro cell we will perform local solves again. To this end, we define the spaces

$$\begin{aligned} \tilde{Q}_h &\coloneqq \{q_h \in Q_h : \Pi_{\tilde{Q}_H} q_h = 0\} \\ \tilde{V}_h &\coloneqq \{\mathbf{v}_h \in V_h : \operatorname{supp}(\mathbf{v}_h) \subset K \text{ for some } K \in \mathcal{M}_H\}. \end{aligned} \tag{7.10}$$

The space $\tilde{V}_h$ consists of local patches of velocity degrees of freedom contained in coarse grid macro cells, as shown in Figure 7.3. As in the low order case, we highlight that these patches decouple and hence solves involving $\tilde{V}_h$ can be performed in parallel, leading to scalable performance. By [Qin94, Section 4.6 & 8.1] and [Zha04] the pair

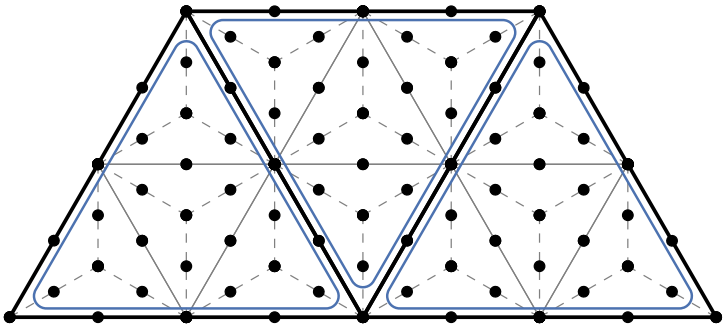


Figure 7.3: Degrees of freedom of the $\tilde{V}_h$ on which we perform local solves to obtain a robust prolongation in two dimensions.

$\tilde{V}_h \times \tilde{Q}_h$ is inf-sup stable in both two and three dimensions, so it remains to check

$$(\nabla \cdot \tilde{\mathbf{v}}_h, \tilde{q}_H) = 0 \quad \text{for all } \tilde{q}_H \in \tilde{Q}_H, \tilde{\mathbf{v}}_h \in \tilde{V}_h, \tag{7.11}$$

which follows from the requirement that the support of vector fields $\tilde{\mathbf{v}}_h \in \tilde{V}_h$ is contained in coarse grid macro cells and the definition of $\tilde{Q}_H$. A robust prolongation operator can now be constructed as in Proposition 5.7.

7.2.2 Robust smoothing

In the previous two chapters we have established that the core requirement to obtain a parameter robust relaxation is a subspace decomposition that also decomposes the kernel of the singular operator. In order to find such a decomposition, we now discuss how one can characterise exactly divergence-free functions in V_h.

Characterisation of the kernel using exact de Rham complexes We begin by recalling some fundamental de Rham complexes and then describe how they can be used to construct a subspace decomposition that decomposes the kernel of the divergence.

The smooth de Rham complex in two dimensions is given by

$$\mathbb{R} \xrightarrow{\text{id}} C^\infty(\Omega) \xrightarrow{\text{curl}} [C^\infty(\Omega)]^2 \xrightarrow{\text{div}} C^\infty(\Omega) \xrightarrow{\text{null}} 0, \tag{7.12}$$

and in three dimensions

$$\mathbb{R} \xrightarrow{\text{id}} C^\infty(\Omega) \xrightarrow{\text{grad}} [C^\infty(\Omega)]^3 \xrightarrow{\text{curl}} [C^\infty(\Omega)]^3 \xrightarrow{\text{div}} C^\infty(\Omega) \xrightarrow{\text{null}} 0. \tag{7.13}$$

Such a complex is called exact, if the kernel of an operator is given by the range of the preceding operator in the sequence, e.g. when $\operatorname{range}\operatorname{curl} = \ker \operatorname{div}$. It is well known that these complexes are exact, precisely when the domain is simply-connected [AFW06, p. 18]. Such an exactness property is of interest here, because it allows us to characterise divergence-free vector fields as the curls of potentials.

Several lower regularity variants of these complexes exist, arguably the most well known ones being

$$\mathbb{R} \xrightarrow{\text{id}} H^1(\Omega) \xrightarrow{\text{curl}} H(\operatorname{div}, \Omega) \xrightarrow{\text{div}} L^2(\Omega) \xrightarrow{\text{null}} 0, \tag{7.14}$$

$$\mathbb{R} \xrightarrow{\text{id}} H^1(\Omega) \xrightarrow{\text{grad}} H(\operatorname{curl}, \Omega) \xrightarrow{\text{curl}} H(\operatorname{div}, \Omega) \xrightarrow{\text{div}} L^2(\Omega) \xrightarrow{\text{null}} 0. \tag{7.15}$$

In the last decades, a significant effort has been made to find finite element spaces that form exact subsequences of (7.14) and (7.15) [AFW06].

Example 7.1. *In three dimensions one can obtain a discrete, exact subsequence by discretising H^1 with continuous Lagrange elements, $H(\operatorname{curl}, \Omega)$ with Nedelec edge elements, $H(\operatorname{div}, \Omega)$ with Nedelec face elements and L^2 with discontinuous Lagrange elements.*

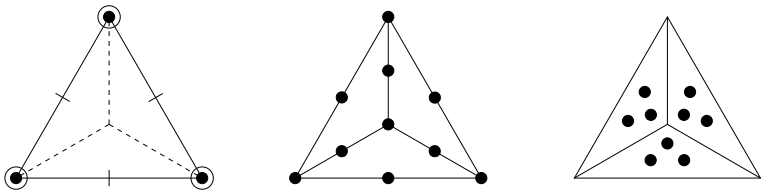


Figure 7.4: A 2D exact Stokes complex on barycentrically refined meshes.

For this work, we are interested in characterising the kernel of the divergence of vector fields with H^1 regularity. Hence, we study the following complexes with more regularity, that are sometimes referred to as the Stokes complexes,

$$\mathbb{R} \xrightarrow{\text{id}} H^2(\Omega) \xrightarrow{\text{curl}} [H^1(\Omega)]^2 \xrightarrow{\text{div}} L^2(\Omega) \xrightarrow{\text{null}} 0, \tag{7.16}$$

and in three dimensions

$$\mathbb{R} \xrightarrow{\text{id}} H^2(\Omega) \xrightarrow{\text{grad}} H^1(\text{curl}, \Omega) \xrightarrow{\text{curl}} [H^1(\Omega)]^3 \xrightarrow{\text{div}} L^2(\Omega) \xrightarrow{\text{null}} 0, \tag{7.17}$$

where $H^1(\text{curl}, \Omega) = \{u \in [H^1(\Omega)]^3 : \text{curl}\, u \in [H^1(\Omega)]^3\}$. Discrete complexes of this form are much harder to construct and often result in high order polynomials due to the high regularity requirements.

Assume now that we have been given a discrete exact subsequence of (7.16) or (7.17) of the form

$$\ldots \to \Sigma_h \xrightarrow{\text{curl}} V_h \xrightarrow{\text{div}} Q_h \xrightarrow{\text{null}} 0. \tag{7.18}$$

Then for a divergence-free discrete vector field $\mathbf{u}_h \in V_h$, we can write it as the curl of a potential $\mathbf{\Phi}_h \in \Sigma_h$. Assume Σ_h has a basis given by $\{\mathbf{\Phi}_j\}$, then $\mathbf{\Phi}_h$ can be written as $\mathbf{\Phi}_h = \sum_j c_j \mathbf{\Phi}_j$ for some coefficients c_j. Now we can define a divergence-free decomposition of $\mathbf{u}_h$ as $\mathbf{u}_h = \sum_j \mathbf{u}_j$ where $\mathbf{u}_j = c_j \nabla \times \mathbf{\Phi}_j$. Hence, a subspace decomposition $\{V_i\}$ such that $\nabla \times \mathbf{\Phi}_j \in V_i$ for some i for any basis function $\mathbf{\Phi}_j$, decomposes the kernel. To understand how to choose a decomposition $\{V_i\}$ that satisfies this property, we have to look at least at the support of the basis function $\{\mathbf{\Phi}_j\}$.

In two dimensions and on barycentrically refined meshes, choosing Σ_h to be the Hsieh-Clough-Tocher (HCT) finite element space together with continuous $[\mathbb{P}_2]^2$ finite element functions for V_h and discontinuous $\mathbb{P}_1$ finite element functions for Q_h yields an exact discrete complex [Joh+17, p. 514]. The three elements are displayed in Figure 7.4. For a given vertex v_i in the macro mesh $\mathcal{M}_h$, we define the macro $\text{star}(v_i)$ of the vertex as the union of all macro elements touching the vertex. We then see that for every

basis function $\mathbf{\Phi}_j$ there exists a vertex v_i such that $\operatorname{supp}(\mathbf{\Phi}_j) \subset \operatorname{macro\,star}(v_i)$. Hence, also $\operatorname{supp}(\nabla \times \mathbf{\Phi}_j) \subset \operatorname{macro\,star}(v_i)$ and if we define

$$V_i = \{\mathbf{v} \in V_h : \operatorname{supp}(\mathbf{v}) \subset \operatorname{macro\,star}(v_i)\} \tag{7.19}$$

then these subspaces decompose the kernel. More recently in [FGN18] an $H^1(\operatorname{curl}, \Omega)$-conforming element on barycentrically refined tetrahedral meshes was introduced that forms an exact sequence with piecewise cubic continuous velocities and piecewise quadratic discontinuous pressures. Hence, by the same argument we obtain that the macro star around vertices provides a decomposition of the kernel of the divergence in three dimensions.

Characterisation of the kernel by a localised Fortin operator In the previous section we introduced discrete exact sequences as a tool to construct a subspace decomposition that also decomposes the kernel of the divergence. When such an exact sequence exists, the approach is clearly attractive since the subspace decomposition can be found by simply studying the support of the basis functions in Σ_h. However, an exact sequence only guarantees the existence of some $\mathbf{\Phi}_h \in \Sigma_h$ so that $\nabla \times \mathbf{\Phi}_h = \mathbf{u}_h$, but does not make statements about its norm. The proof for exactness in [FGN18] for example is based on a counting argument and hence does not provide any bounds.

In two dimensions, this is not an issue as it is straightforward to obtain an element in Σ_h with bounded norm, as we will now argue. For a divergence-free vector field we know (see [GR86, Theorem 3.3]) that there exists a $\mathbf{\Phi} \in H_0^2(\Omega)$ such that $\nabla \times \mathbf{\Phi} = \mathbf{u}_h$ and $\|\mathbf{\Phi}\|_2 \preceq \|\mathbf{u}_h\|_1$. Since $\nabla\times$ in two dimensions simply corresponds to the rotated gradient, we see that any two $\mathbf{\Phi}$ that satisfy $\nabla \times \mathbf{\Phi} = \mathbf{u}_h$ are equal up to a constant, and hence we have in fact $\mathbf{\Phi} \in \Sigma_h$.

In three dimensions, the second step in this argument fails. It was proven by [CM10] that the *regularised Poincaré* operator provides a bounded linear map

$$R : [H^s(\Omega)]^3 \to [H^{s+1}(\Omega)]^3 \text{ s.t. } \nabla \times R(\mathbf{u}) = \mathbf{u} \text{ for all } \mathbf{u} \in H^s \text{ with } \nabla \cdot \mathbf{u} = 0, \tag{7.20}$$

for any s and domain Ω that is star-like with respect to some ball. However, although the norm of the potential obtained from this map is bounded by the norm of $\mathbf{u}$, we cannot directly infer this property for the discrete potential as the uniqueness property that we exploited in two dimensions does not hold. In fact, we can add any gradient to the potential $\mathbf{\Phi}$ and still preserve $\nabla \times \mathbf{\Phi} = \mathbf{u}$.

We briefly recall the approach in Chapter 6 to construct the subspace decomposition. We constructed a covering $\{\Omega_i\}$ of the domain and picked a corresponding partition

of unity $\{\rho_i\}$. We then constructed a particular interpolation operator $I : V \to V_h$ and defined $\mathbf{u}_i = I(\nabla \times (\rho_i \mathbf{\Phi}))$. The operator I was essentially the same one as in the classical proof for the inf-sup stability of the $[\mathbb{P}_2]^2 - \mathbb{P}_0$ element, with minor modifications so that it uses only values on element boundaries. However, as of writing the author is not aware of a similar construction of a Fortin operator for the Scott-Vogelius element in either two or three dimensions that could be modified.

While we do know that a Fortin operator must exist for the Scott-Vogelius element, we do not a priori know whether it has a locality property similar to (6.17). We will instead use the fact that the Scott-Vogelius element is known to be inf-sup stable on a single macro element to construct a global Fortin operator by suitably *gluing together* local Fortin operators on each macro element. We will formulate the statement and proof for the special case of the Scott-Vogelius element on barycentric refinements, but the same idea applies to Powell-Sabin splits or other macro elements for which inf-sup stability is known. This approach is attractive as it does not require the explicit construction of a $[H^2(\Omega)]^2/H^1(\mathrm{curl}, \Omega)$-conforming finite element.

Lemma 7.2 (Localised Fortin operator)**.**
There exists a linear map $I_h : V \to V_h$ such that

- *I_h is continuous,*
- *$(q_h, \nabla \cdot (I_h(\mathbf{v}))) = (q_h, \nabla \cdot \mathbf{v})$ for all $q_h \in Q_h$ and $\mathbf{v} \in V$,*
- *$I_h(\mathbf{v}_h) = \mathbf{v}_h$ for $\mathbf{v}_h \in V_h$,*
- *there exists a covering $\{\Omega_i\}$ of Ω such that for all vertices v_i in the macro mesh it holds*

$$I_h(\mathbf{v}) \in V_i \text{ for all } \mathbf{v} \in V \text{ such that } \mathrm{supp}(\mathbf{v}) \subset \Omega_i \tag{7.21}$$

where $\{V_i\}$ are the subspaces defined in (7.19)*.*

Remark 7.3. *The first three conditions are the classical conditions for a Fortin operator. The last condition is necessary to ensure that the decomposition is local.*

Proof. The idea is to combine a global Fortin operator that preserves the discrete divergence with respect to pressures that are constant on macro cells with suitable local Fortin operators. We begin by defining the space of constant pressures on $\mathcal{M}_h$:

$$\tilde{Q}_h := \{q_h \in Q_h : q_h|_K \equiv \text{const } \forall K \in \mathcal{M}_h\}. \tag{7.22}$$

Now let $\tilde{I}_h : V \to V_h$ be a Fortin operator for the pair $V_h \times \tilde{Q}_h$ constructed as in the proof of Lemma 6.1. The integration regions for the Scott-Zhang operator underlying

$\tilde{I}_h$ and the covering $\{\Omega_i\}$ that satisfies the locality property are shown in Figure 7.5. We note that this operator is the identity on V_h and preserves the discrete divergence with respect to $\tilde{Q}_h$.

To treat the divergence with respect to the remaining pressures in $Q_h \setminus \tilde{Q}_h$ we now consider the macro elements separately. For each such macro triangle K we define the spaces $V_{h,0}(K) = \{\mathbf{v}_h|_K : \mathbf{v}_h \in V_h,\ \text{supp}(\mathbf{v}_h) \subset K\}$ and $Q_h(K) = \{q_h|_K : q_h \in \tilde{Q}_h^{\perp}\}$. The space $V_{h,0}(K)$ consists of velocity fields supported in a macro cell and the space $Q_h(K)$ consists of pressures on a macro cell that integrate to zero. We note that $Q_h = \sum_{K\in\mathcal{M}_h} Q_h(K) \oplus \tilde{Q}_h$. Since the Scott-Vogelius element is inf-sup stable on each macro element, we know that there exist Fortin operators $I_h^K : V(K) \to V_{h,0}(K)$, where $V(K) = \{\mathbf{v}|_K : \mathbf{v} \in V\}$, such that

- $I_h^K(\mathbf{v}_h) = \mathbf{v}_h$ for $\mathbf{v}_h \in V_{h,0}(K)$
- I_h^K is bounded as a map $V(K) \mapsto V_{h,0}(K)$
- $(q_h, \nabla\cdot\mathbf{v}) = (q_h, \nabla\cdot(I_h^K(\mathbf{v})))$ for all $q_h \in Q_h(K)$

for all K, [For77], [EG04, Lemma 4.19].

Now we define

$$I_h(\mathbf{v}) = \tilde{I}_h(\mathbf{v}) + \sum_K I_h^K((\mathbf{v} - \tilde{I}_h(\mathbf{v}))|_K). \tag{7.23}$$

Clearly, I_h is linear and $I_h(\mathbf{v}_h) = \mathbf{v}_h$ for all $\mathbf{v}_h \in V_h$. In addition, I_h is continuous with continuity constant only dependent on the continuity constant of $\tilde{I}_h$ and the local Fortin operators, and I_h satisfies the locality property in (7.21).

Furthermore, we note that the discrete divergence of vector fields in $V_{h,0}(K)$ with respect to $\tilde{Q}_h$ is zero. It follows that

$$\begin{aligned}
&(q_h, \nabla\cdot(I_h(\mathbf{v}))) \\
=&\underbrace{(q_h, \nabla\cdot(\tilde{I}_h(\mathbf{v})))}_{=(q_h,\nabla\cdot\mathbf{v})} + \sum_K \underbrace{(q_h, \nabla\cdot(\overbrace{I_h^K((\mathbf{v} - \tilde{I}_h(\mathbf{v}))|_K)}^{\in V_{h,0}(K)}))}_{=0} \\
=&(q_h, \nabla\cdot\mathbf{v})
\end{aligned} \tag{7.24}$$

for all $q_h \in \tilde{Q}_h$ and $\mathbf{v} \in V$.

Lastly, we show that I_h preserves the discrete divergence with respect to the local pressures in $Q_h(K)$. For $\mathbf{v} \in V$, $K \in \mathcal{M}_h$, and $q_h \in Q_h(K)$, we have

$$\begin{aligned}
&(q_h, \nabla\cdot(I_h(\mathbf{v}))) \\
&= (q_h, \nabla\cdot(\tilde{I}_h(\mathbf{v}))) + (q_h, \nabla\cdot(I_h^K((\mathbf{v} - \tilde{I}_h\mathbf{v})|_K))) \\
&= (q_h, \nabla\cdot(\tilde{I}_h(\mathbf{v}))) + (q_h, \nabla\cdot(\mathbf{v} - \tilde{I}_h(\mathbf{v}))) \\
&= (q_h, \nabla\cdot\mathbf{v}),
\end{aligned} \tag{7.25}$$

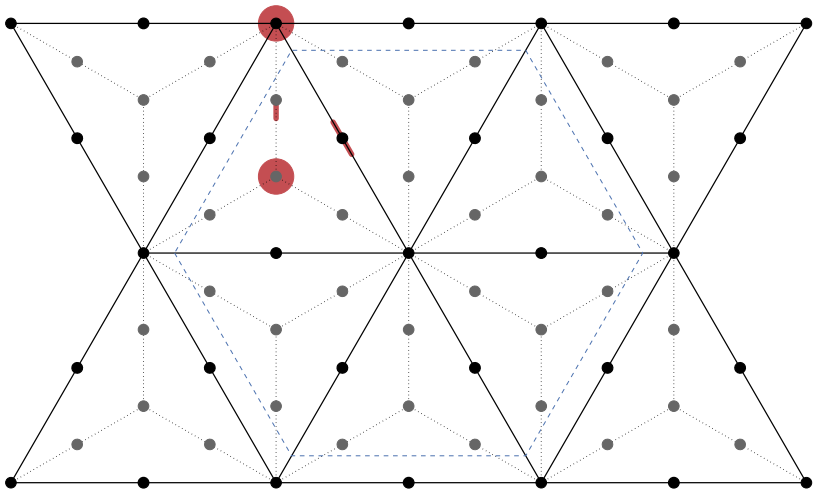


Figure 7.5: Domain Ω_i around a vertex v_i (blue dashed line), covering all degrees of freedom inside macro star(v_i). The figure also shows integration regions (in red) for vertex and edge degrees of freedom. Note that the integration regions are chosen so that only those associated with macro star(v_i) intersect with Ω_i.

as desired. □

Once we have obtained this localised Fortin operator, the proof for the existence of a splitting is now the same as in Proposition 6.1.

Proposition 7.4.

Assume that Ω is simply-connected, consider the subspace decomposition defined in (7.19) and denote by $D_{h,\gamma}$ the subspace correction method associated with the decomposition $\{V_i\}$. Then $\mathbf{u}_h \in V_h$ and $\mathbf{u}_0 \in \mathcal{N}_h$ satisfy

$$\inf_{\substack{\mathbf{u}_h=\sum \mathbf{u}_i \\ \mathbf{u}_i \in V_i}} \sum_i \|\mathbf{u}_i\|_1^2 \preceq h^{-2}\|\mathbf{u}_h\|_0^2 \tag{7.26}$$

$$\inf_{\substack{\mathbf{u}_0=\sum \mathbf{u}_{0,i} \\ \mathbf{u}_{0,i} \in \mathcal{N}_h \cap V_i}} \sum_i \|\mathbf{u}_{0,i}\|_1^2 \preceq h^{-4}\|\mathbf{u}_0\|_0^2. \tag{7.27}$$

Moreover, it holds that

$$(h^{-2}+h^{-4})^{-1} D_{h,\gamma} \preceq A_{h,\gamma} \leq N_O D_{h,\gamma} \tag{7.28}$$

where N_O is the maximum number of overlapping subspaces in the mesh. In particular, the bounds are independent of γ.

Proof. Exactly as in Lemma 6.1 but now pick the covering and the interpolation operator constructed in Lemma 7.2. □

7.3 Stabilisation for the advection terms

In the previous chapter we stabilised the advection terms using standard SUPG stabilisation. Since we considered piecewise constant pressures, the pressure gradient vanished and hence only the top-left block was modified. For the Scott-Vogelius element this is not the case, and SUPG stabilisation results in a modification of the upper off-diagonal block, introducing additional coupling between the velocity and the pressure [BFH06, p. 1249], [Gel+05, p. 250]. In fact, in numerical experiments we observe that even when solving the linear system using a direct solver, the outer Newton solver fails to converge already for $\mathrm{Re} \sim 50$ for the two dimensional lid-driven cavity. Furthermore, the modified top-right block needs to be taken into account when adding the augmented Lagrangian term and in the multigrid scheme for the top-left block.

In 1976 Douglas & Dupont [DD76] suggested an interior penalty scheme that penalises a jump of the derivative across facets:

$$S(\mathbf{u}, \mathbf{v}) = \sum_{K \in \hat{\mathcal{M}}_h} \frac{1}{2} \int_{\partial K} \delta h_{\partial K}^2 \left[\!\left[\nabla \mathbf{u}\right]\!\right] \cdot \left[\!\left[\nabla \mathbf{v}\right]\!\right] \mathrm{d}s, \tag{7.29}$$

where $[\![\nabla \mathbf{u}]\!]$ denotes the jump of the gradient, $h_{\partial K}$ is a function giving the facet size, and δ is a free parameter. The term vanishes when the velocity has H^2 regularity.

This scheme has received renewed attention and it was shown in [BH04] that it successfully stabilises advection-dominated problems and has subsequently been used to stabilise the Stokes [BH06] and the Oseen equations [BFH06; BL08]. In addition to not introducing any additional coupling of the pressure and the velocity, an advantage of this scheme is that it is adjoint consistent and preserves the order of convergence in the L^2 norm [BH04; BL08]. We demonstrate this in Section 7.4.2.

Let us now consider the effect of adding the term in (7.29) on the multigrid scheme for the top-left block. Since S vanishes for functions that have continuous gradients, we have added another bilinear form to our system that has a nontrivial kernel consisting of C^1 vector fields. As the weight $\delta h_{\partial K}^2$ is small, the impact is not as significant as that of the grad-div term, but for very high Reynolds number or coarse meshes, we still observe reduced performance of the multigrid scheme. From Proposition 5.4, we know that in order for the smoother to be robust, the subspace decomposition must provide a decomposition of the kernel. In two dimensions, this is satisfied if $k \geq 3$, as the macro star around vertices then captures the support of the HCT element. In three dimensions the lowest degree for a local basis for C^1 vector fields on barycentrically refined meshes is $k = 5$.

We emphasise that this argument is not rigorous, as it does not take into account any potential interaction between the two singular terms. In addition, it is not clear how one would construct a cheap, robust prolongation operator. However, in the numerical experiments we will see that the scheme is noticeably more robust for $k = 3$ in two dimensions and $k = 5$ in three dimensions.

7.4 Numerical results

We present numerical results for the same benchmark problems as before. The general setup of the solver is the same as previously shown in Figure 6.7, but we employ the mesh hierarchy, the smoother, and the transfer operators that were developed in this chapter. The free parameter δ in the stabilisation (7.29) is chosen to be cell dependent as $\delta|_K = 5 \cdot 10^{-3} \|\mathbf{u}|_K\|_\infty$.

7.4.1 Solving the top-left block only: nearly incompressible elasticity

We begin again by studying the performance for the top-left solve only without any advection. Since for the Scott-Vogelius element the projection onto the pressure space is given by the identity, the model problem we study is given by: find $\mathbf{u} \in V_h$ such that

$$\int_\Omega \mathrm{E}\mathbf{u} : \mathrm{E}\mathbf{v} \,\mathrm{d}x + \hat{\gamma} \int_\Omega (\nabla \cdot \mathbf{u})(\nabla \cdot \mathbf{v}) \,\mathrm{d}x = \int_\Omega \mathbf{f} \cdot \mathbf{v} \,\mathrm{d}x \tag{7.30}$$

for all $\mathbf{v} \in V_h$. As in Section 6.4.2, we choose $\Omega = [0, 1]^d$, homogeneous Dirichlet boundary conditions on $\partial\Omega$ and set $\mathbf{f} = (1, 1)$ in two and $\mathbf{f} = (1, 1, 1)$ in three dimensions.

In contrast to the results obtained previously for the $[\mathbb{P}_2]^2 - \mathbb{P}_0$ and the $[\mathbb{P}_1 \oplus B_3^F]^3 - \mathbb{P}_0$ element, we now observe good multigrid performance with the standard prolongation up to values of $\hat{\gamma} = 10^4$. Only for very large values of $\hat{\gamma}$, the modified prolongation is necessary. Recall that the purpose of the modified prolongation operator is essentially to map (discretely) divergence-free vector fields from the coarse-grid to (discretely) divergence-free vector fields on the fine grid. Since for the Scott-Vogelius discretisation, a discretely divergence-free vector field is exactly divergence-free, the only introduced divergence can stem from the interpolation error between the nonnested hierarchies. The observation that the multigrid scheme without special prolongation is robust to such large values of γ suggests that the divergence introduced by this error is very small.

Refinements	Degrees of freedom	$\hat{\gamma}$							
		0	1	10	10^2	10^3	10^4	10^6	10^8
Robust relaxation & robust transfer									
1	1 602	3	4	5	6	6	5	4	3
2	6 274	3	4	5	6	6	5	4	3
3	24 834	3	4	5	6	6	5	4	3
4	98 818	3	4	5	6	5	5	4	3
Robust relaxation & standard transfer									
1	1 602	3	3	5	6	6	7	28	>100
2	6 274	3	4	5	6	6	8	>100	>100
3	24 834	3	4	5	6	6	8	>100	>100
4	98 818	3	4	5	6	6	8	>100	>100
Jacobi relaxation & robust transfer									
1	1 602	7	8	16	41	83	>100	>100	>100
2	6 274	7	9	18	47	>100	>100	>100	>100
3	24 834	7	9	18	47	>100	>100	>100	>100
4	98 818	7	9	18	47	>100	>100	>100	>100
Jacobi relaxation & standard transfer									
1	1 602	7	8	16	45	>100	>100	>100	>100
2	6 274	7	9	18	53	>100	>100	>100	>100
3	24 834	7	9	18	55	>100	>100	>100	>100
4	98 818	7	9	18	55	>100	>100	>100	>100
Algebraic multigrid									
1	1 602	11	13	22	55	>100	>100	>100	>100
2	6 274	11	14	26	64	>100	>100	>100	>100
3	24 834	12	15	29	74	>100	>100	>100	>100
4	98 818	13	16	33	83	>100	>100	>100	>100

Table 7.1: Iteration counts in two dimensions for the $[\mathbb{P}_2]^2$ element for five different geometric and algebraic multigrid variants. The geometric multigrid results are obtained with a 4×4 coarse grid.

Refinements	Degrees of freedom	$\hat{\gamma}$ 0	1	10	10^2	10^3	10^4	10^6	10^8
Robust relaxation & robust transfer									
1	23 871	3	3	3	4	5	5	4	2
2	185 115	3	4	4	6	6	6	5	3
3	1 458 867	3	4	5	6	7	7	6	4
Robust relaxation & standard transfer									
1	23 871	3	3	3	4	5	6	9	>100
2	185 115	3	4	4	6	7	7	27	>100
3	1 458 867	3	4	5	6	8	9	>100	>100
Jacobi relaxation & robust transfer									
1	23 871	8	9	17	51	>100	>100	>100	>100
2	185 115	10	12	21	60	>100	>100	>100	>100
3	1 458 867	10	12	22	63	>100	>100	>100	>100
Jacobi relaxation & standard transfer									
1	23 871	8	9	17	64	>100	>100	>100	>100
2	185 115	10	12	22	74	>100	>100	>100	>100
3	1 458 867	10	12	22	76	>100	>100	>100	>100
Algebraic multigrid									
1	23 871	20	23	38	>100	>100	>100	>100	>100
2	185 115	24	28	48	>100	>100	>100	>100	>100
3	1 458 867	27	32	54	>100	>100	>100	>100	>100

Table 7.2: Iteration counts in three dimensions for the $[\mathbb{P}_3]^3$ element for five different geometric and algebraic multigrid variants. The geometric multigrid results are obtained with a $2 \times 2 \times 2$ coarse grid.

7.4.2 Verification and pressure robustness

In order to validate the implementation and the claim for Reynolds number independent errors of the velocity field, we study the same test case as in Section 6.4.3, but in two dimensions. We observe the expected second order convergence of the velocity gradient and of the pressure as the mesh is refined, see Figure 7.6. To compare with the discretisation from the previous chapter, we perform the same tests for the $[\mathbb{P}_2]^2-\mathbb{P}_0$ element on the same barycentrically refined mesh and with the stabilisation defined in (7.29). As in Section 6.4.3 we see only first order convergence and a growing velocity error as the Reynolds number is increased. This is in contrast to the $[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$ Scott-Vogelius element, which exhibits Reynolds-robust error estimates. The reason for this is the weak enforcement of the divergence constraint for the $[\mathbb{P}_2]^2-\mathbb{P}_0$ element

that can be seen in Figure 7.6. The solutions obtained using the Scott-Vogelius element are divergence-free up to solver tolerances. In addition, we remark that with the interior penalty type stabilisation we now observe the desired convergence rates for the L^2 error of the velocity, even for high Reynolds numbers and coarse meshes.

7.4.3 Algorithmic performance in two and three dimensions

As in the previous chapter, we study the lid-driven cavity and the backwards facing step benchmark problems.

Two dimensions We begin by studying the regularised lid-driven cavity. The setup is the same as in the previous chapter, apart from the coarse grid, which we choose to be 10×10 instead of 16×16. Since the grid is barycentrically refined at every level, this leads to a comparable problem size. As discussed in the previous section, for $k = 2$ we do not expect the $[\mathbb{P}_k]^2 - \mathbb{P}_{k-1}^{\text{disc}}$ pair to be fully robust due to nullspace of the stabilisation terms. In Table 7.3 can see that iteration counts are stable from $\text{Re} = 10$ to $\text{Re} = 1\,000$ but then increase. This increase becomes less prominent as the mesh is refined and the stabilisation term is reduced. If we choose $k = 3$ (in order to capture the nullspace of the stabilisation term with the relaxation method), iteration counts decrease significantly and robustness is improved.

Discretisation	Degrees of freedom	Reynolds number				
		10	100	1 000	5 000	10 000
$[\mathbb{P}_2]^2 - \mathbb{P}_1^{\text{disc}}$						
1	1.70×10^4	4.50	5.00	6.67	18.00	25.50
2	6.75×10^4	4.00	4.33	5.67	16.00	29.00
3	2.69×10^5	4.00	4.00	4.00	10.00	18.00
4	1.08×10^6	3.00	3.33	2.67	6.50	11.50
$[\mathbb{P}_3]^2 - \mathbb{P}_2^{\text{disc}}$						
1	3.62×10^4	2.50	2.67	3.33	8.00	11.50
2	1.44×10^5	2.50	2.67	2.33	5.50	9.50
3	5.77×10^5	2.00	2.67	2.00	4.00	5.50
4	2.31×10^6	2.00	2.67	2.00	2.50	5.00

Table 7.3: Average number of outer Krylov iterations per Newton step for the 2D regularised lid-driven cavity problem using the Scott-Vogelius discretisation for $k = 2$ and $k = 3$.

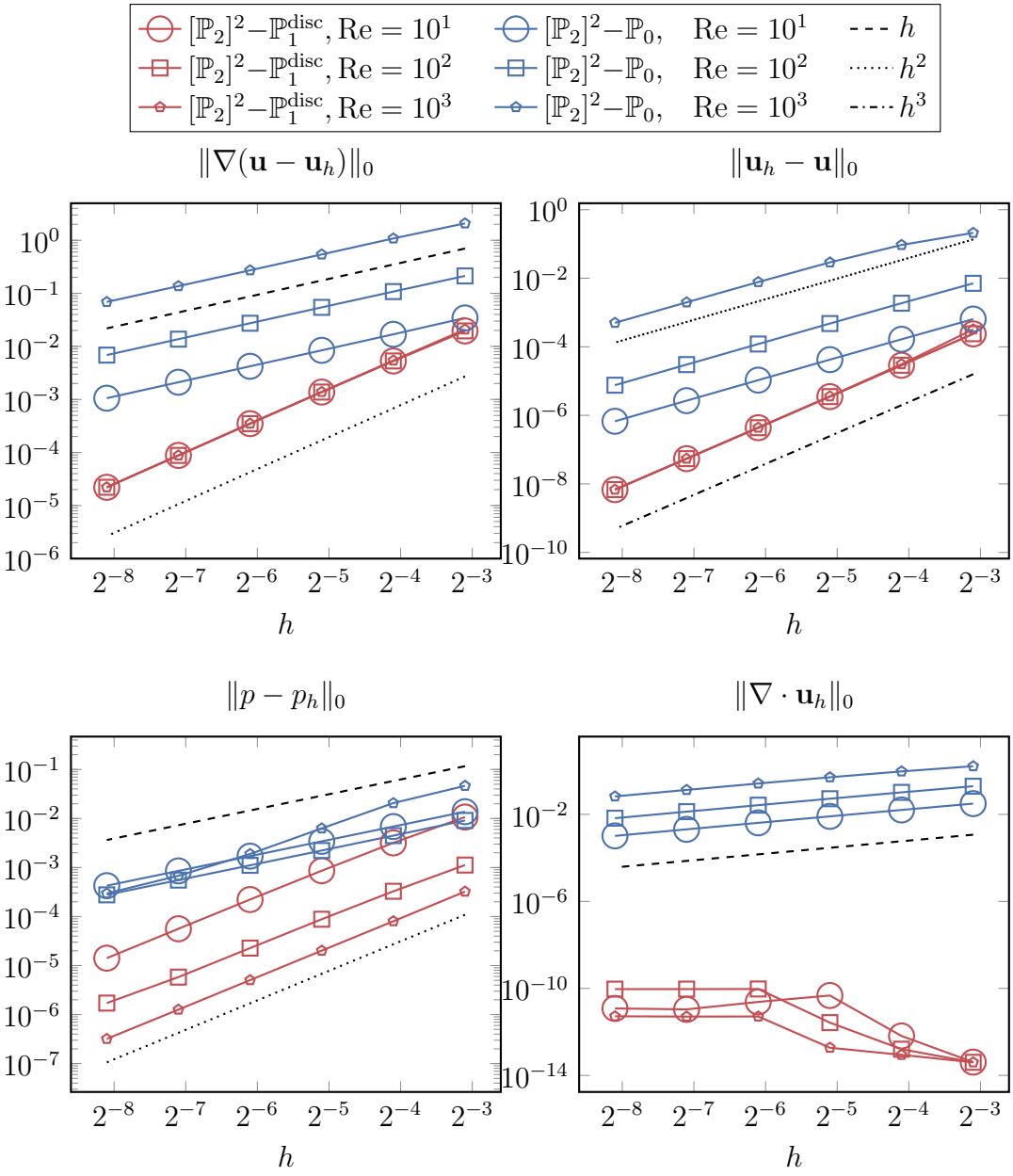


Figure 7.6: Velocity and pressure error as well as L^2 norm of the divergence for the $[\mathbb{P}_2]^2$–$\mathbb{P}_0$ and the $[\mathbb{P}_2]^2$–$\mathbb{P}_1^{\text{disc}}$ element for different Reynolds numbers.

In addition to iteration counts, we also study the impact of the larger patches on the runtime of the solver. We observe that for comparable problems, the runtime is approximately two to three times longer for the $[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$ element. This is not surprising, as the patches in the relaxation and prolongation contain roughly an order of magnitude more degrees of freedom than in the low order case. However, the convergence study in the previous section shows that when the problem has sufficient regularity and even for moderately high Reynolds number, the Scott-Vogelius discretisation leads to smaller approximation errors even on significantly coarser meshes. In addition, we highlight that for high Reynolds number the runtime for the $[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$ element is competitive to the $[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$ element despite significantly more degrees of freedom. This is due to the improved robustness of this element at high Reynolds number.

Discretisation	Degrees of freedom	Reynolds number				
		10	100	1 000	5 000	10 000
$[\mathbb{P}_2]^2-\mathbb{P}_0$	6.57×10^5	1.85	2.17	2.70	5.32	5.67
$[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$	1.08×10^6	3.27	3.51	2.97	6.04	10.16
$[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$	2.31×10^6	7.04	8.30	7.01	8.01	13.30

Table 7.4: Runtime per Newton step (in seconds) for the 2D regularised lid-driven cavity using the $[\mathbb{P}_2]^2-\mathbb{P}_0$, $[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$, and $[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$ element pairs. Measured on two nodes of a Cray XC30 with two 12-core E5-2697v2 CPUs per node.

For the backwards-facing step the results are qualitatively similar, c.f. Table 7.5. We observe growth of the iteration counts on coarse meshes past $\text{Re} = 1\,000$, but see significantly improved robustness as the mesh is refined. As before for the lid-driven cavity, the performance of the solver can be improved noticeably by choosing the $[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$ element.

Three dimensions As in the previous chapter, we now study the three dimensional variants of the two benchmark problems. We study iteration counts both with and without adding the stabilisation terms in (7.29).

Results for the lid-driven cavity are shown in Table 7.6. Both with and without stabilisation we observe iteration counts that approximately double as the Reynolds number is increased from $\text{Re} = 10$ to $\text{Re} = 5\,000$.

We show results for the backward-facing step in Table 7.7 Without stabilisation, we observe iteration counts that approximately triple over the same range of Reynolds numbers. However, when adding stabilisation, iteration counts increase significantly

Discretisation	Degrees of freedom	Reynolds number				
		10	100	1 000	5 000	10 000
$[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$						
1	4.79×10^5	3.67	3.25	5.00	14.50	19.50
2	1.91×10^6	3.67	3.25	4.00	8.50	13.50
3	7.64×10^6	4.50	4.33	3.00	5.00	9.50
$[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$						
1	1.02×10^6	2.00	2.00	2.50	4.50	6.00
2	4.10×10^6	2.50	2.33	1.50	2.00	3.00
3	1.64×10^7	2.50	3.33	2.00	1.50	2.00

Table 7.5: Average number of outer Krylov iterations per Newton step for the 2D backwards-facing step problem using the Scott-Vogelius discretisation for $k = 2$ and $k = 3$.

and blow up for very high Reynolds number. We attribute this to the issue raised in Section 7.3: the stabilisation term itself has a large nullspace (consisting of C^1 vector fields) that is not captured by the relaxation induced by the macro star around vertices. If we choose $k = 5$ we know that a local basis for C^1 functions exists. Indeed, we see in Table 7.7, that iteration counts for the $[\mathbb{P}_5]^3-\mathbb{P}_4^{\text{disc}}$ element are significantly more robust. Though we would not recommend this as a practical discretisation due to its high computational cost and memory requirements, the results demonstrate the necessity of capturing the nullspace of all singular operators in order to obtain a robust scheme.

Refinements	Degrees of freedom	Reynolds number				
		10	100	1 000	2 500	5 000
$k = 3$ without stabilisation						
1	1.03×10^6	3.00	3.67	3.50	4.00	5.00
2	8.22×10^6	3.50	3.67	4.00	4.00	4.00
3	6.55×10^7	3.00	3.33	3.50	3.50	4.00
$k = 3$ with stabilisation						
1	1.03×10^6	3.00	4.00	4.50	5.00	6.00
2	8.22×10^6	3.50	4.00	5.50	6.00	6.50
3	6.55×10^7	3.00	3.33	5.00	6.00	7.50

Table 7.6: Average number of outer Krylov iterations per Newton step for the 3D regularised lid-driven cavity problem.

Refinements	Degrees of freedom	Reynolds number 10	100	1 000	2 500	5 000
$k = 3$ without stabilisation						
1	3.85×10^6	4.50	4.33	5.33	9.00	15.00
2	3.06×10^7	5.00	5.33	5.33	10.00	12.00
$k = 3$ with stabilisation						
1	3.85×10^6	4.50	5.33	7.33	11.50	13.50
2	3.06×10^7	5.00	6.33	12.50	14.00	154.00
$k = 5$ with stabilisation						
1	3.81×10^6	1.50	1.67	2.00	3.50	4.00
2	3.03×10^7	2.00	1.67	2.00	2.50	5.00

Table 7.7: Average number of outer Krylov iterations per Newton step for the 3D backwards-facing step problem. The results for $k = 5$ were obtained on a coarser mesh to have a comparable number of degrees of freedom to the case of $k = 3$.

Code availability

The code for the Navier–Stokes solver and the numerical experiments in this chapter can be found at `https://github.com/florianwechsung/alfi/`. For reproducibility, all major Firedrake components as well as the code used to run these examples have been archived on Zenodo [Fir19; Wec19b].

Part III

Combination and conclusions

Chapter 8

Shape optimisation with geometric multigrid solvers

The goal of this chapter is to develop an approach that enables the use of the previously developed solver for the Navier–Stokes equations for shape optimisation problems. Recall that in order to define robust prolongation operators, the multigrid schemes that we developed rely heavily on the nested structure of the mesh hierarchy. This requirement raises two issues. If the boundary of the physical domain Ω^{phys} is curved, nested refinements of the coarse grid $\mathcal{M}_H$ have the same boundary and do not approximate $\partial\Omega^{\text{phys}}$ as $h \to 0$. The alternative is to adjust the boundary after each refinement, however, then the nestedness property is lost. Both approaches are demonstrated in Figure 8.1. The second issue is closely related: the optimal shape is obtained by deforming the mesh. If we move the vertices of the fine grid, nestedness of the hierarchy is not necessarily preserved. Alternatively, we can move the vertices of the coarse grid and deform the fine meshes correspondingly. This preserves nestedness of the hierarchy but restricts the set of representable shapes.

8.1 Nestedness preserving high-order deformations

The approach we suggest to address both issues is based on curved meshes and the ideas of isoparametric finite elements. Let us start by considering an initial coarse triangulation $\mathcal{M}_H$ on a polytope $\hat{\Omega}$ that approximates Ω^{phys}. For some element degree $p \geq 2$ we consider the Lagrange finite element space

$$\mathcal{H}_H(\hat{\Omega}) := \{\mathbf{T} \in H^1(\hat{\Omega}; \mathbb{R}^d) : \mathbf{T}|_K \in [P^p(K)]^d \ \forall K \in \mathcal{M}_H\}. \tag{8.1}$$

Now we construct a vector field $\mathbf{T}_H \in \mathcal{H}_H(\hat{\Omega})$ such that

$$\Omega \approx \Omega^{\text{phys}} \quad \text{where} \quad \Omega := \mathbf{T}_H(\hat{\Omega}). \tag{8.2}$$

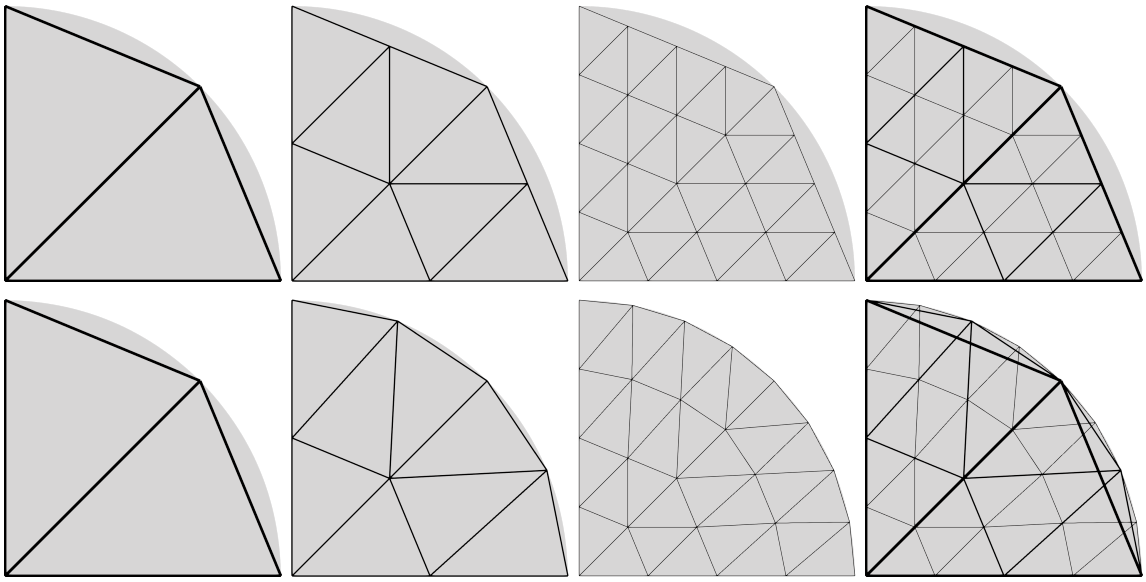


Figure 8.1: Coarse grid, first refinement, second refinement, and superimposed mesh hierarchy obtained from an initial triangulation. Top row: without boundary adjustment. Bottom row: with boundary adjustment after each refinement.

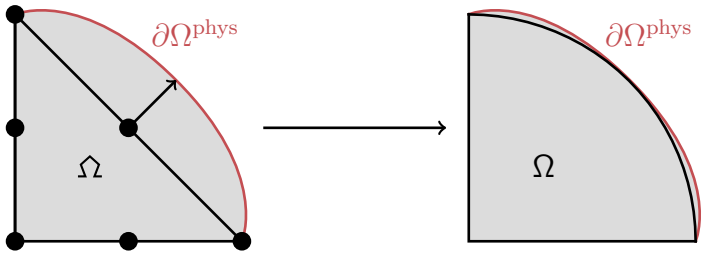


Figure 8.2: The smooth boundary $\partial\Omega$ can be approximated by deforming an initial affine mesh with a piecewise quadratic deformation.

This can be done by projecting each facet or edge degree of freedom of $\mathcal{H}_H(\Omega)$ that lies on the boundary of Ω onto $\partial\Omega^{\text{phys}}$, as shown in Figure 8.2. Firedrake has an interface to OpenCascade, so that the mapping $\mathbf{T}_H$ can be obtained automatically if the geometry is provided in STEP format, a commonly used file format in Computer-aided Design.

We now construct a mesh hierarchy as follows. The initial triangulation $\mathcal{M}_H$ is refined regularly to obtain a nested, fine triangulation $\mathcal{M}_h$. We then define the space $\mathcal{H}_h(\Omega)$ on $\mathcal{M}_h$ analogously to $\mathcal{H}_H(\Omega)$, and since the hierarchy is nested we have $\mathcal{H}_H(\Omega) \subset \mathcal{H}_h(\Omega)$. Hence, we can define $\mathbf{T}_h = P_H^{\mathcal{H}}(\mathbf{T}_H)$, where $P_H^{\mathcal{H}} : \mathcal{H}_H(\Omega) \to \mathcal{H}_h(\Omega)$ is the prolongation operator given by finite element interpolation and is equal to the identity. Deforming the coarse grid with $\mathbf{T}_H$ and the refined grid with $\mathbf{T}_h$ results again in a nested hierarchy, as shown in Figure 8.3. The degree p of the deformation is chosen large enough so that the deformed coarse mesh is a good approximation of

the physical domain.

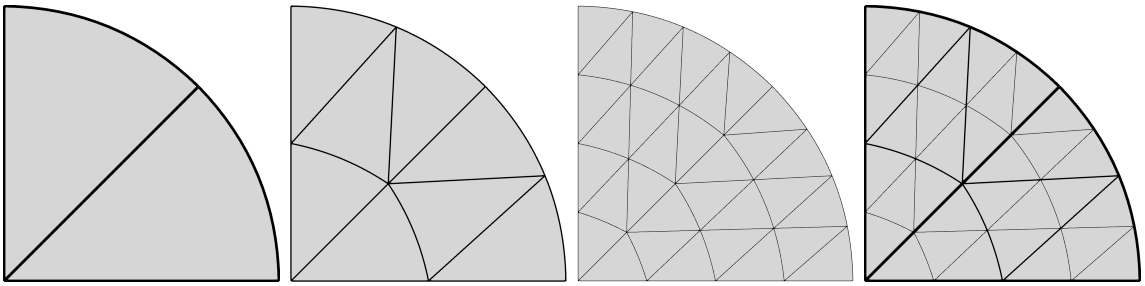


Figure 8.3: Left: initial mesh deformed with a piecewise quadratic. Middle: refinements of the coarse mesh deformed with the prolonged deformation from the coarse grid. Right: superimposed mesh hierarchy demonstrating that nestedness is preserved.

We now repeat the same approach in order to perform shape optimisation: the control space is given by high order deformations defined on the coarse-grid; this deformation is then prolonged onto the fine grids in order to deform the entire mesh hierarchy in a way that preserves nestedness.

To that end, we define $\mathcal{H}_H(\Omega)$ as the push-forward of $\mathcal{H}_H(\hat{\Omega})$ under $\mathbf{T}_H$:

$$\mathcal{H}_H(\Omega) := \{\mathbf{X} \circ \mathbf{T}_H^{-1} : \mathbf{X} \in \mathcal{H}_H(\hat{\Omega})\}, \tag{8.3}$$

and $\mathcal{H}_h(\Omega)$ is defined analogously. In an abuse of notation, we denote the prolongation by $P_H^{\mathcal{H}} : \mathcal{H}_H(\Omega) \to \mathcal{H}_h(\Omega)$. The shape optimisation problem can then be formulated as

$$\underset{\mathbf{X}_H \in \mathcal{H}_H(\Omega)}{\text{minimise}} \; J((\mathbf{Id} + P_H^{\mathcal{H}} \mathbf{X}_H)(\Omega)). \tag{8.4}$$

This approach is demonstrated in Figure 8.4 for the simple levelset problem that we discussed in Section 4.4.1. Note that the shape derivative on the coarse grid is obtained by applying the adjoint of the prolongation operator to the shape derivative on the fine grid.

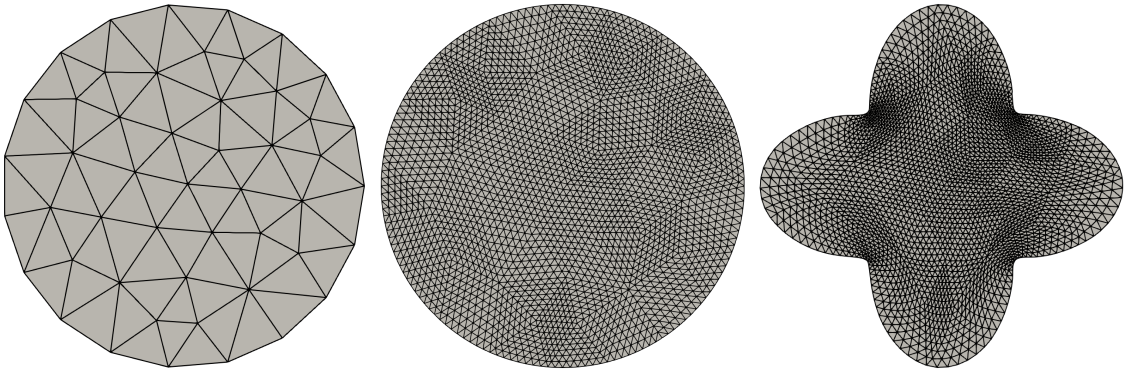


Figure 8.4: Left: coarse triangulation $\mathfrak{Omega}$ of a disk. Middle: Third refinement of the triangulation of Ω obtained by deforming $\mathfrak{Omega}$ with piecewise cubics. Right: Optimal shape given by $(\mathbf{Id} + P_H^{\mathcal{H}}\mathbf{X}_H)(\Omega)$ for some deformation $\mathbf{X}_H \in \mathcal{H}_H(\Omega)$.

Denote $\mathbf{T}_\mathbf{X} = \mathbf{Id} + P_H^{\mathcal{H}}\mathbf{X}_H$, then the function spaces for velocity and pressure on the deformed domain are given by

$$\begin{aligned} V_h(\mathbf{T}_\mathbf{X}(\Omega)) &:= \{\mathbf{v} \circ \mathbf{T}_h^{-1} \circ \mathbf{T}_\mathbf{X}^{-1} : \mathbf{v} \in V_h(\mathfrak{Omega})\} \\ Q_h(\mathbf{T}_\mathbf{X}(\Omega)) &:= \{q \circ \mathbf{T}_h^{-1} \circ \mathbf{T}_\mathbf{X}^{-1} : q \in Q_h(\mathfrak{Omega})\}. \end{aligned} \tag{8.5}$$

Note that though these functions are not necessarily polynomials, they are still the push forward of polynomials on the reference element but with respect to a different mapping. In particular the strategy from Chapter 3 to automatically calculate shape derivatives still applies.

Remark 8.1. *Though the same approach can in principle be used to obtain nested barycentrically refined mesh hierarchies, the Scott-Vogelius element does not enforce the divergence constraint exactly on curved meshes. The reason for this is that while* $\operatorname{div}(V_h(\mathfrak{Omega})) \subset Q_h(\mathfrak{Omega})$*, the corresponding property is not satisfied on* Ω*, i.e.* $\operatorname{div}(V_h(\Omega)) \not\subset Q_h(\Omega)$*. To see this, let* $\mathbf{u} = \mathbf{v} \circ \mathbf{T}_h^{-1} \in V_h(\Omega)$ *and calculate*

$$\operatorname{div}(\mathbf{u}) = \operatorname{tr}(\nabla \mathbf{T}_h^{-1}(\nabla \mathbf{v}) \circ \mathbf{T}_h^{-1}). \tag{8.6}$$

If $\mathbf{v}$ *is a polynomial of degree* k*,* $\nabla \mathbf{v}$ *is a polynomial of degree* $k-1$ *hence the entries in* $(\nabla \mathbf{v}) \circ \mathbf{T}_h^{-1}$ *are in* $Q_h(\Omega)$*. However, when* $\mathbf{T}_h$ *is not affine, then this property is not given for the product* $\nabla \mathbf{T}_h^{-1}(\nabla \mathbf{v}) \circ \mathbf{T}_h^{-1}$*. It may be possible to remedy this issue by using a different mapping between functions on the affine mesh and the curved mesh, but for now we focus on the low order discretisation from Chapter 6. We will later briefly remark on* $H(\operatorname{div})$ *conforming discretisations that are exactly divergence-free even on curved meshes.*

Remark 8.2. *Multigrid solvers in the context of shape optimisation were also investigated by [SW17]. However, for the problems they consider the requirements for the grid transfer operator are less strict and they do not have the same nestedness requirements as here. Hence, they consider standard affine meshes and deformations and interpolate the deformation from the fine grid to the coarser grids. That way the coarse grid 'moves with the fine grid', but the hierarchy is not exactly nested, similar to the bottom row of Figure 8.1. They see good multigrid performance for problems involving diffusion and elasticity equations.*

8.2 Regularisation

We now make three minor modifications to the optimisation problem that will allow us to solve more challenging shape optimisation problems at higher Reynolds numbers. First, we recall that we observed kinks in the boundary of the pipe example that we studied in Section 3.4, see Figure 3.4. In order to obtain more regular deformations, we augment the inner product with the bilinear form that arises in the C^0 interior penalty discretisation of the biharmonic operator [BS05; WKG06; Bre11]. Denoting the set of elements in the mesh by $\{K\}$ and the set of edges / facets by $\{E\}$, we let

$$\begin{aligned} a_{\mathrm{ip}}(\mathbf{U}, \mathbf{V}) = &\sum_K \int_K \Delta \mathbf{U} \cdot \Delta \mathbf{V} \,\mathrm{d}x - \sum_E \int_E \{\!\{\Delta \mathbf{U}\}\!\} \left[\!\left[\frac{\partial \mathbf{V}}{\partial \mathbf{n}_E} \right]\!\right] \mathrm{d}s \\ &- \sum_E \int_E \{\!\{\Delta \mathbf{V}\}\!\} \left[\!\left[\frac{\partial \mathbf{U}}{\partial \mathbf{n}_E} \right]\!\right] \mathrm{d}s + \sigma \sum_E \frac{1}{h_E} \int_E \left[\!\left[\frac{\partial^2 \mathbf{U}}{\partial \mathbf{n}_e^2} \right]\!\right] \left[\!\left[\frac{\partial^2 \mathbf{V}}{\partial \mathbf{n}_E^2} \right]\!\right] \mathrm{d}s, \end{aligned} \tag{8.7}$$

where $[\![\cdot]\!]$ denotes the jump and $\{\!\{ \cdot \}\!\}$ denotes the average across a facet. The parameter σ needs to be chosen large enough in order to guarantee coercivity [Bre11, eqn. (3.23)]. We pick $\sigma = 10$.

The biharmonic operator has already been observed to be well suited for mesh deformation in [Hel03] and was used in the context of shape optimisation in [Keu15, § 2.10.2.]. In Section 8.4 we will compare the shape obtained with and without the biharmonic augmentation.

The inner product that we consider is then given by

$$(\mathbf{U}, \mathbf{V})_{\mathcal{H}} := (\mathrm{E}(\mathbf{U}), \mathrm{E}(\mathbf{V}))_{L^2} + \alpha (\mathrm{B}\mathbf{U}, \mathrm{B}\mathbf{V})_{L^2} + \beta a_{\mathrm{ip}}(\mathbf{U}, \mathbf{V}), \tag{8.8}$$

where E denotes the symmetrised gradient and B denotes the Cauchy–Riemann operator from Chapter 4. We set $\alpha = 0$ in three dimensions.

For the second modification, we recall that in Section 2.4 we discussed that for a smooth domain the shape derivative should be zero for deformations with a trace that vanishes in normal direction. However, this is not the case for the discretised problem. Figure 8.5 shows the magnitude of the discretised shape derivative for an example problem; though small, there are noticeable contributions in the interior of the domain.

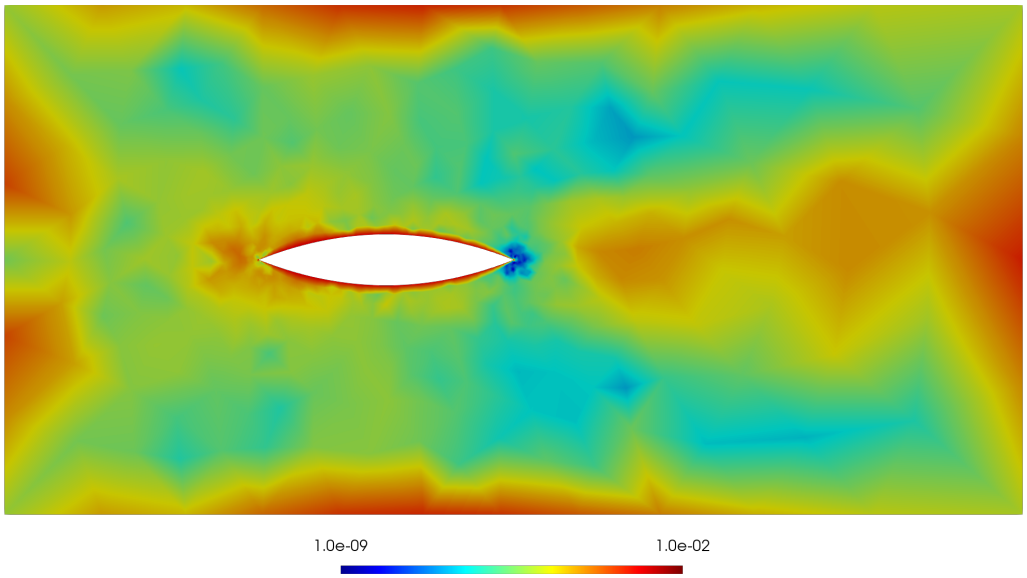


Figure 8.5: Magnitude of the shape derivative of the drag of an obstacle immersed in a fluid.

We recall the definition of the shape gradient,

$$(\nabla J_\Omega(\mathbf{X}), \mathbf{V})_{\mathcal{H}} = \mathrm{d}J_\Omega(\mathbf{X}; \mathbf{V}) \text{ for all } \mathbf{V} \in \mathcal{H}. \tag{8.9}$$

For the continuous problem the derivative is supported on the boundary, and hence, denoting by $\mathcal{H}_0$ the space of deformations in $\mathcal{H}$ with zero trace, we have that

$$(\nabla J_\Omega(\mathbf{X}), \mathbf{V})_{\mathcal{H}} = 0 \quad \text{for all } \mathbf{V} \in \mathcal{H}_0. \tag{8.10}$$

Since the deformation $\mathbf{X}$ is simply a linear combination of the gradients throughout the optimisation, this means that $\mathbf{X}$ also satisfies

$$(\mathbf{X}, \mathbf{V})_{\mathcal{H}} = 0 \quad \text{for all } \mathbf{V} \in \mathcal{H}_0. \tag{8.11}$$

We can regularise the discretised problem by enforcing the same property, that is we solve

$$\begin{aligned} \underset{\mathbf{X}_H \in \mathcal{H}_H(\Omega)}{\text{minimise}} \quad & J((\mathbf{Id} + P_H^{\mathrm{D}} \mathbf{X}_H)(\Omega)) \\ \text{subject to} \quad & (\mathbf{X}_H, \mathbf{V})_{\mathcal{H}} = 0 \text{ for all } \mathbf{V} \in \mathcal{H}_0(\Omega). \end{aligned} \tag{8.12}$$

Remark 8.3. *The approach is similar to the one proposed in [Etl+18]. Etling et. al. go a step further and require at each step that the deformation direction can be obtained as the solution of a Neumann problem, that is, they do not only remove interior contributions of the derivative but also those with respect to tangential deformations of the boundary. However, since the normal changes as the domain is deformed, this is a constraint on each optimisation step rather than on the whole deformation, and it is not clear how one could formulate an optimisation problem on the reference domain that enforces this requirement at each step.*

Finally, to remove the invariance with respect to tangential deformations of the boundary, we add a small Tikhonov regularisation of the form $\zeta\|\mathbf{X}_H\|^2_{\mathcal{H}}$ to the objective. We choose $\zeta = 10^{-4}$ in the experiments to follow. In Section 8.3 we show an example for which the mesh degenerates due to spurious contributions to the shape derivative.

The shape derivatives for the examples shown below were almost all calculated automatically using UFL; only the projection in the augmented Lagrangian term involved a manual calculation for the shape derivative. As in Chapter 4.4.3 we enforce the equality constraints in the optimisation problem using an augmented Lagrangian algorithm; the inner, unconstrained optimisation problems are solved using L-BFGS. We use piecewise quadratic deformations on the coarse grid for all examples. The criterion developed in Corollary 2.13 for diffeomorphisms is not enforced explicitly in the optimisation, but we use it at each step to check if the deformation is too large, and reduce the step size if necessary.

We now revisit two of the problems that we already studied in previous chapters.

8.3 Drag minimisation of an immersed object

In Section 4.4.3 we studied the drag minimisation of an object immersed in Stokes flow. In this section we replace the PDE constraint with the Navier–Stokes equations and we use the preconditioner developed in Chapter 6 to solve the state and adjoint equation. We want to allow for the angle of the tips to change, i.e. we do not expect a smooth deformation, hence we choose $\beta = 10^{-3}$ in (8.8). We start with an initial shape that is similar to what was obtained for Stokes flow in Chapter 4 and perform the optimisation for Reynolds numbers 125, 250, 500, and 1 000. The coarse mesh is unstructured and consists of 1 406 cells. We perform three refinements resulting in $4.5 \cdot 10^5$ degrees of freedom for the PDE constraint.

The obtained shapes are shown in Figure 8.6, and the optimal shape and the velocity field for $\mathrm{Re} = 1\,000$ are shown in 8.7. The optimal shape for Stokes flow that

we obtained in Section 4.4.3 was symmetric. This is expected as the equations are linear and the objective does not change if the flow direction is reversed. Here we clearly see that as the Reynolds number increases and the PDE constraint becomes more nonlinear, the optimal shape becomes asymmetric.

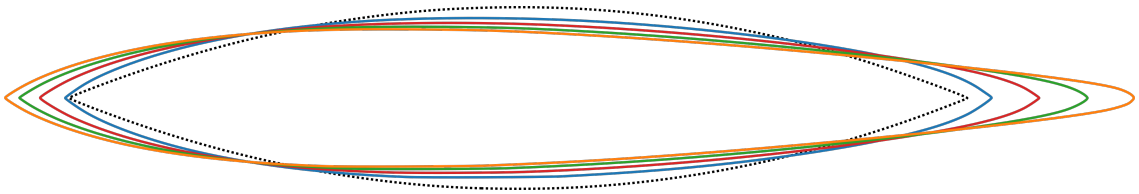


Figure 8.6: Initial shape and optimal shapes for an immersed obstacle for Reynolds numbers 125, 250, 500, and 1 000.

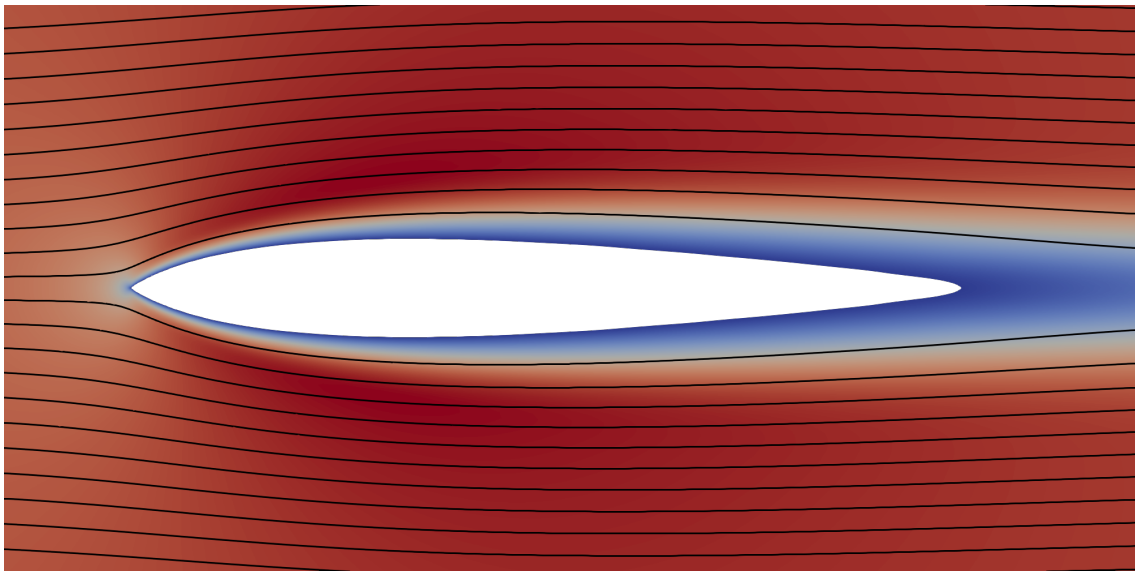


Figure 8.7: Optimal shape and velocity field for $\mathrm{Re} = 1\,000$. The colour denotes speed.

We demonstrate the necessity for the Tikhonov regularisation and the PDE constraint on the control in Figure 8.8. Without this regularisation we obtain spurious deformations in the interior once the shape is close to the optimal shape, eventually causing the solver for the PDE constraint to fail.

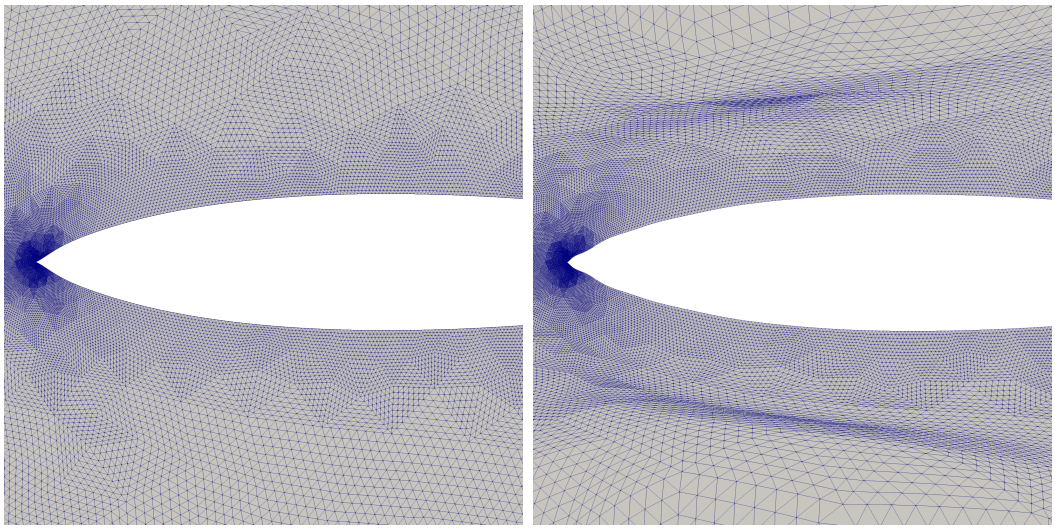


Figure 8.8: Shape obtained without Tikhonov regularisation and control constraint after 50 iterations and after 84 optimisation steps. We see significant deformations in the interior of the domain.

Finally, in Figure 8.9 we show the objective value, the norm of the gradient of the Lagrangian, the constraint violation and the average number of Krylov iterations per Newton step for each Reynolds number. The norm of the gradient is reduced by approximately three orders of magnitude and the violation of the volume and barycentre constraints is of order 10^{-5}–10^{-4}. We note that despite the domain not satisfying the assumption of being simply-connected that we made in Chapter 6, the preconditioner remains effective. Iteration counts are slightly higher than in Chapter 6 as tighter solver tolerances were chosen to obtain accurate gradients

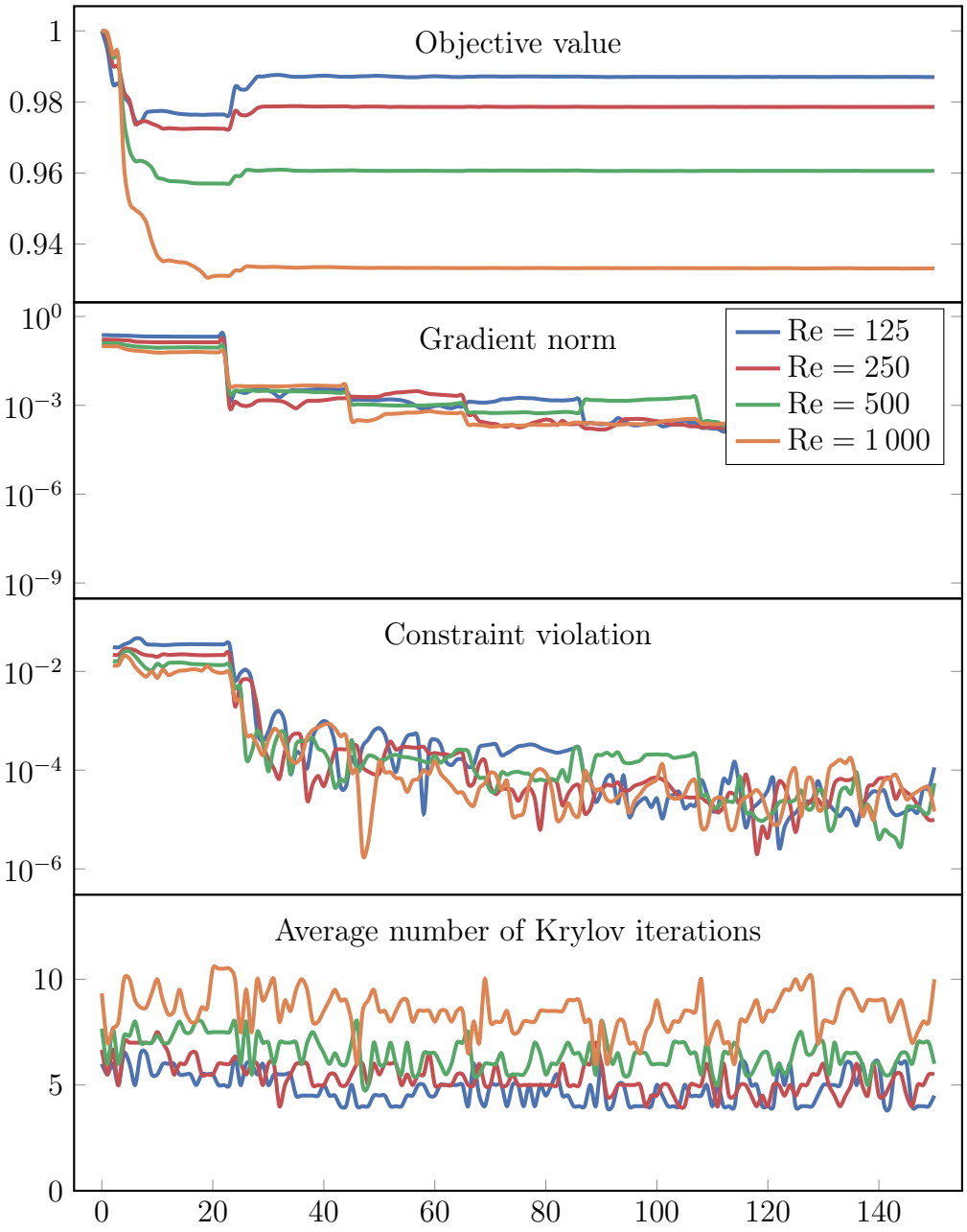


Figure 8.9: Convergence history for the problem of minimising the drag of an obstacle immersed in a fluid subject to a volume and barycentre constraint.

8.4 Optimisation of a pipe

We now revisit the optimisation of a pipe to reduce energy losses from Section 3.4, though this time we will perform the optimisation both in two and in three dimensions and at higher Reynolds number. The coarse mesh in two dimensions is again unstructured and consists of 6 394 cells. We perform two refinements resulting in $5.1 \cdot 10^5$

degrees of freedom for the PDE constraint.

The initial and the optimal shape for Reynolds numbers 125, 250, 500, and 1 000 are shown in Figure 8.10. For low Reynolds number the optimal shape is close to a straight connection between the two fixed ends. As the Reynolds number is increased, the optimal shape becomes more 'S' shaped and avoids sharp bends and high curvature.

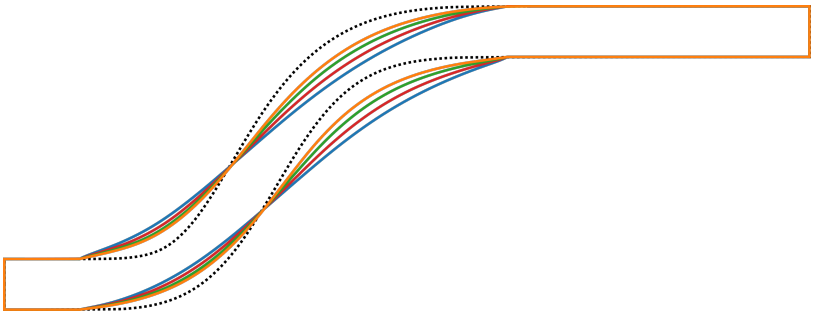


Figure 8.10: Initial shape and optimal shapes for the pipe problem for Reynolds numbers 125, 250, 500, and 1 000.

To motivate the use of the biharmonic operator in the inner product, we take a closer look at the part of the boundary in the bottom left where the boundary condition on the deformation changes from being fixed to being free. In Figure 8.11 we show close ups for Reynolds number 125 with and without the biharmonic term in the inner product. Without the biharmonic term we see a tangled mesh after only 10 iterations of the optimisation scheme and in fact the solver fails in the next iteration. In contrast, when using the biharmonic term we still observe a kink, but the mesh does not degenerate throughout the optimisation.

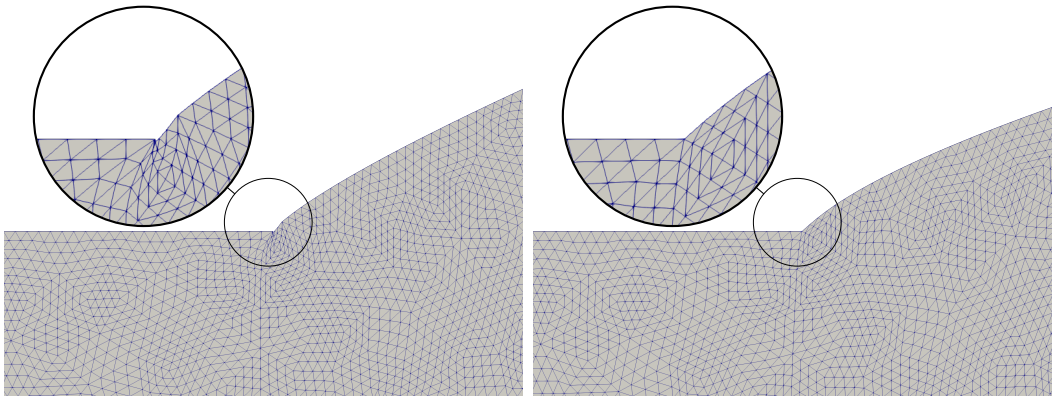


Figure 8.11: Left: without the biharmonic term in the inner product the mesh degenerates after only 10 optimisation steps and the solver for the PDE constraint fails. Right: mesh obtained after 150 optimisation steps when including the biharmonic term.

The convergence history is shown in Figure 8.12, and we see that the energy loss is reduced by 15%–20% depending on the Reynolds number. We observe a reduction of the gradient by between four and five orders of magnitude and a violation of the volume constraint of at most order 10^{-5} after 200 iterations. The faster convergence for higher Reynolds number may be explained by the initial guess being closer to the optimal shape for these cases.

As for the obstacle example, we see well controlled iteration counts of the augmented Lagrangian solver throughout the optimisation.

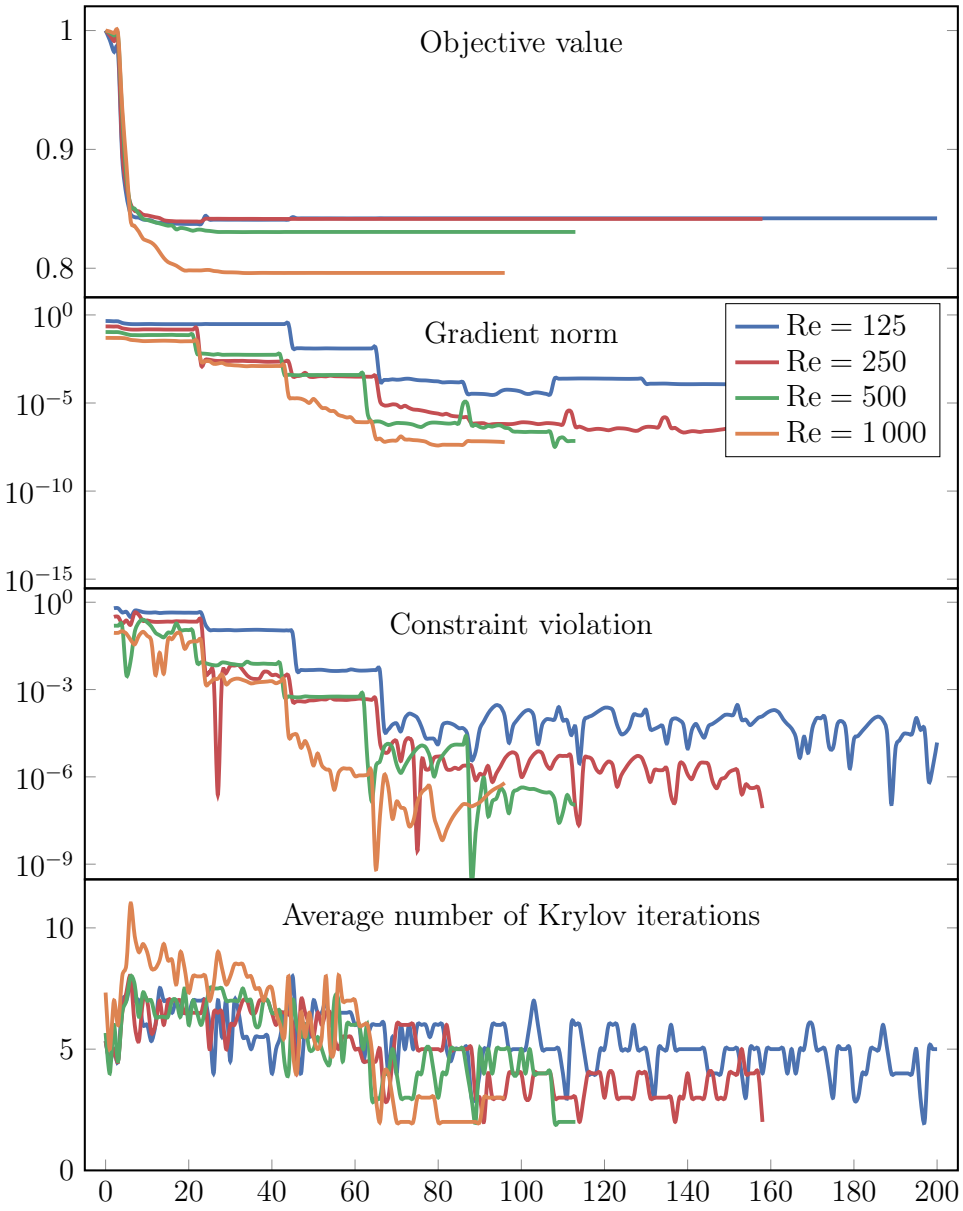


Figure 8.12: Convergence history for the two dimensional pipe shape optimisation problem.

Lastly we study a three dimensional version of the pipe problem. We use the $[\mathbb{P}_2 \oplus B_3^F] - \mathbb{P}_0$ element here as we noted that the continuity constant of the prolongation, when using the $[\mathbb{P}_1 \oplus B_3^F] - \mathbb{P}_0$ element, increased as the mesh was deformed, hence affecting the performance of the preconditioner. In addition, we increase the Tikhonov regularisation to $3 \cdot 10^{-5}$. We start with a coarse mesh of 59 547 cells and refine it once to obtain a problem with $5.5 \cdot 10^6$ degrees of freedom.

We show the obtained shapes obtained for Reynolds numbers 125, 250, 500, and 1 000 in Figure 8.13. The velocity field for $\mathrm{Re} = 500$ is shown in Figure 8.14.

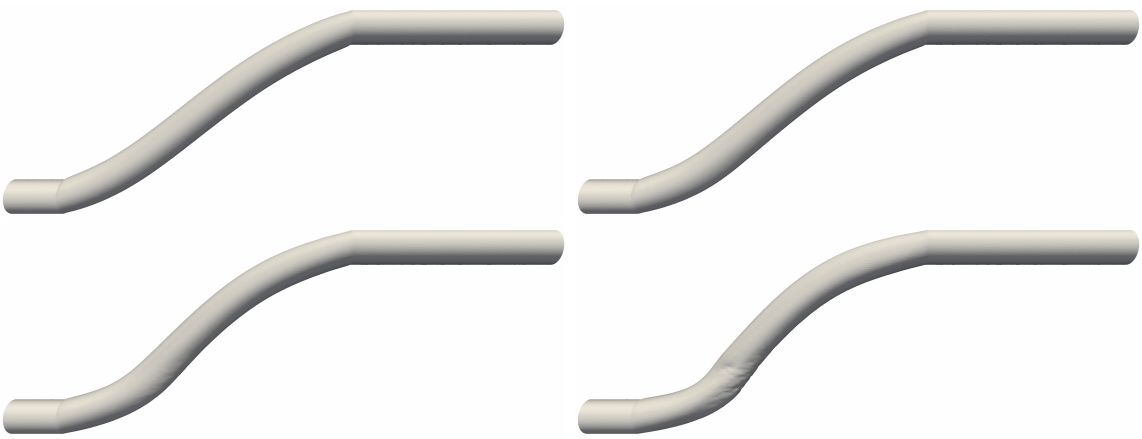


Figure 8.13: Optimal shape for the three dimensional pipe optimisation problem. Top-left: $\mathrm{Re} = 125$. Top-right: $\mathrm{Re} = 250$. Bottom-left: $\mathrm{Re} = 500$. Bottom-right: $\mathrm{Re} = 1,000$.

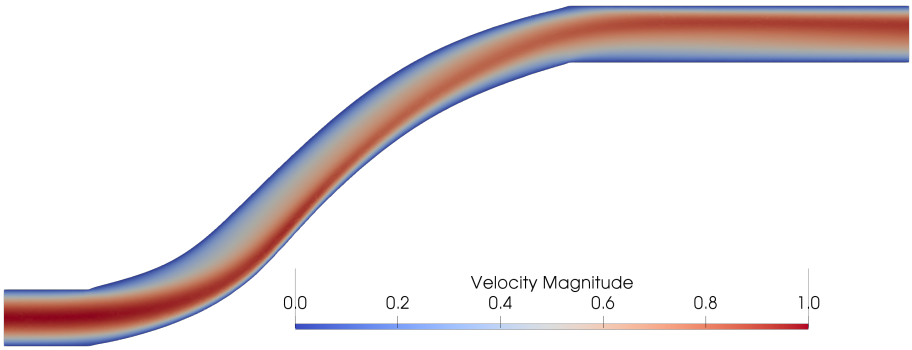


Figure 8.14: Velocity field for the optimal shape at $\mathrm{Re} = 500$.

For $\mathrm{Re} = 500$ we observe some roughness in the lower part of the pipe which increases for $\mathrm{Re} = 1\,000$. We also observe in Figure 8.15 that for the lower Reynolds numbers we see good convergence as the gradient is reduced by approximately three orders of magnitude and the constraint violation is of order 10^{-4}. The convergence for

the $\mathrm{Re} = 1\,000$ case is not as good which we attribute to the relatively coarse mesh that cannot resolve the flow fully at high Reynolds number. Finally, we note that the performance of the preconditioner is very good throughout the optimisation for all four Reynolds numbers. Convergence is achieved after approximately four Krylov iterations per Newton step.

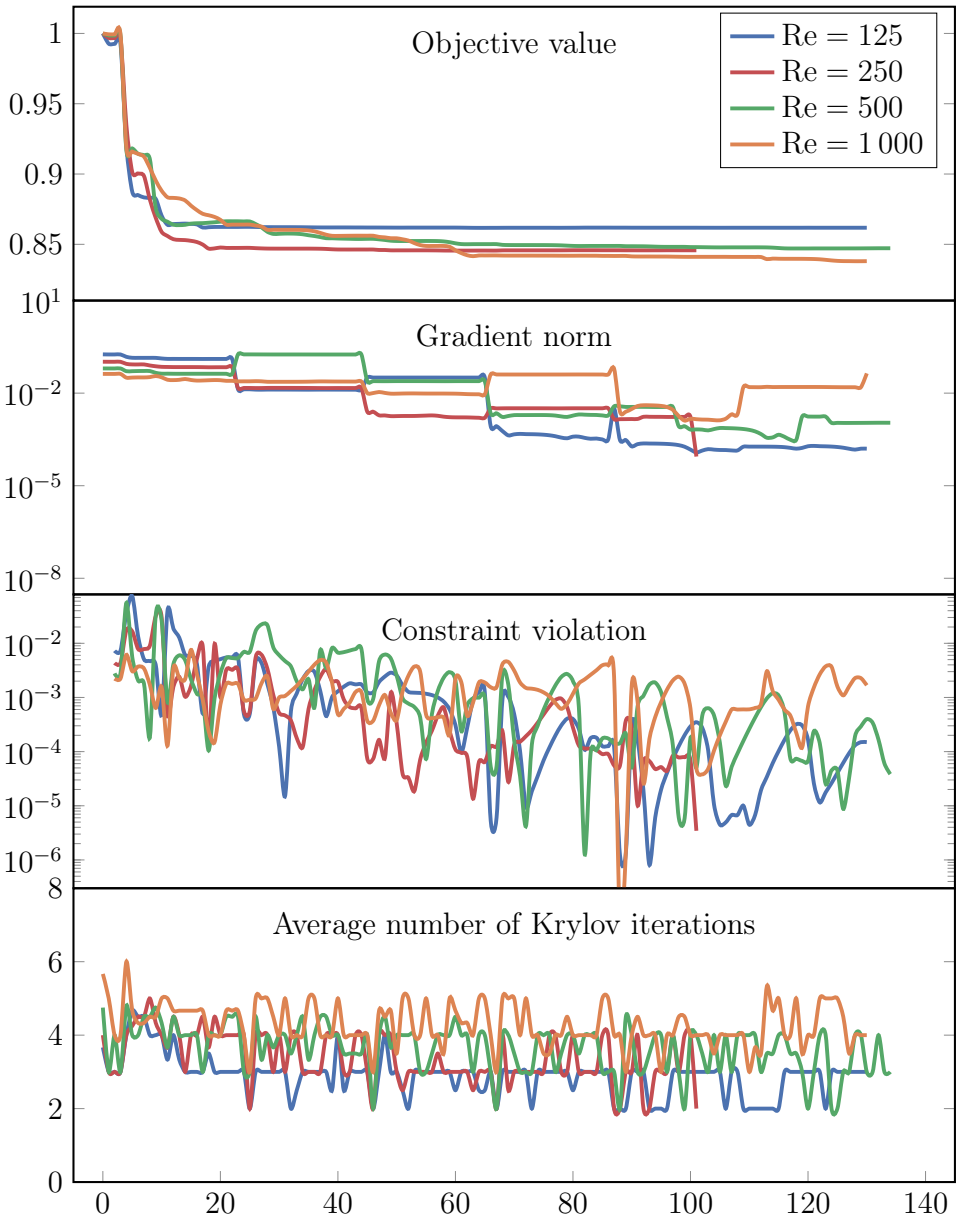


Figure 8.15: Convergence history for the three dimensional pipe shape optimisation problem.

Code availability

The code for the numerical experiments from this chapter is available at `https://github.com/florianwechsung/ThesisNumerics`. For reproducibility, all major Firedrake components as well as the code used to run these examples have been archived on Zenodo [Fir19; Wec19c; Wec19a; Wec19b].

Chapter 9

Conclusions and outlook

In this thesis we have studied several questions that arise when performing shape optimisation, with a focus on the case when the optimisation is constrained by the incompressible Navier–Stokes equations.

9.1 Shape Optimisation

The first part of the thesis was concerned with two general aspects of shape optimisation. In Chapter 3 we introduced a reformulation of the classical shape derivative in the context of finite elements. We expressed the shape derivative as the derivative with respect to the mapping between the reference and the physical element. This viewpoint enables a straightforward implementation of automated shape differentiation in the UFL software package and in other finite element codes. This new feature in UFL is now the basis for the automated shape optimisation capabilities of the dolfin-adjoint and Fireshape libraries.

In Chapter 4 we focussed on mesh deformation. In the context of finite elements, the domain is given by a mesh, and we desire to find deformations that preserve the angles in the mesh. As there are many deformations that transform a given initial shape to the optimal shape, we were able to exploit this nonuniqueness to pick a deformation with particularly good properties. By augmenting the inner product with a least-squares formulation of the Cauchy–Riemann equations we obtained shape gradients in two dimensions that are nearly conformal. Denoting this inner product by $(\cdot,\cdot)_{\mathrm{CR}(\alpha)+\mathcal{H}}$, we showed that in the limit $\alpha \to 0$, the nonconformal part of the shape gradient vanishes.

The Riemann mapping theorem guarantees that there exists a conformal mapping between any two simply-connected domains in two dimensions. Aligning with this result, in our numerical examples we observed that, in this case, the new inner product

indeed yields deformations that preserve angles in the mesh and we were able to perform large shape deformations without mesh stretching or tangling. We noted that a balance has to be struck: very small values of α yield highly conformal deformations but slow down the optimisation and result in a harder problem for the Riesz map.

When the domain is not simply-connected or when the deformations are clamped at a part of a boundary we cannot expect to find a conformal mapping between the initial and the optimal shape. For this case we introduced a weighting into the inner product that emphasises conformality of the deformation close to the boundary. Using this weighting function we demonstrated an improvement of the mesh quality for the classical problem of an object immersed in Stokes flow.

The approach presented is based on the Cauchy–Riemann equations and hence does not naturally extend to three dimensions. In fact, we remark that in higher dimensions the analogue to the Riemann mapping theorem does not hold and hence we cannot expect there to be a conformal mapping between two domains, not even when they are simply-connected.

For this reason it would be desirable to formulate other notions for the quality of deformations and the resulting meshes, that translate more naturally to three dimensions. In [Rui+19] a mesh deformation strategy is proposed that relies on minimising a distortion measure of the mesh. However, the measure is nonlinear in the deformation and hence cannot be incorporated into the inner product. Here it may be desirable to take the approach outlined in Section 2.4 and to introduce a control space $\mathcal{E}$ on the boundary, to formulate an extension operator $E : \mathcal{E} \to \mathcal{D}$ based on [Rui+19], and to then minimise the objective $\tilde{J}_\Omega(e) = J_\Omega(E(e))$.

9.2 Robust large-scale solvers for the incompressible Navier–Stokes equations

In the second part of the thesis we focussed on a particular PDE constraint given by the steady, incompressible Navier–Stokes equations. The development of preconditioners for these equations has been a highly active field for the last two decades. In Chapter 5 we studied the augmented Lagrangian preconditioner by Benzi & Olshanskii [BO06] that modifies the problem and shifts the difficulty from finding a good Schur complement approximation to finding robust solvers for a modified top-left block. For nearly singular symmetric problems like the one arising in the top-left block of the augmented Lagrangian formulation, Schöberl developed the necessary theory for multigrid preconditioners with robust performance with respect to the augmented

Lagrangian parameter γ and the viscosity ν. We outlined the framework developed by Schöberl and argued that the two key ingredients for a robust multigrid scheme are a relaxation and a prolongation operator with parameter independent performance. For the relaxation one can obtain a robust scheme by considering a subspace correction method built on a subspace decomposition that decomposes the nullspace of the singular operator. A robust prolongation can be obtained by modifying the standard prolongation operator to map discretely divergence-free vector fields on the coarse grid to (nearly) divergence-free vector fields on the fine grid.

Up to this point the discussion did not specify a particular discretisation. In Chapter 6 we began by studying the $[\mathbb{P}_2]^2-\mathbb{P}_0$ element that was already considered by Benzi & Olshanskii. We described the construction of robust relaxation and prolongation schemes in two dimensions and then argued that the same element cannot be used in three dimensions. We discussed several alternatives and developed a tailored prolongation scheme that preserves the divergence, leading to a robust scheme in three dimensions. We presented a fully parallel and open-source implementation in Firedrake and PETSc and solved problems with up to a billion degrees of freedom on more than 24 000 cores, demonstrating that the solver is highly suitable for large scale computations. In two dimensions and on the finest grids we observed at most a quadrupling of iteration counts as the Reynolds number is increased from $\mathrm{Re} = 10$ to $\mathrm{Re} = 10\,000$. In three dimensions we observed robust performance up to Reynolds number $\mathrm{Re} = 5\,000$, with iteration counts staying nearly constant from $\mathrm{Re} = 10$ to $\mathrm{Re} = 5\,000$.

The work in Chapter 6 relied on the use of piecewise constant pressures, as this allowed us to identify the discrete divergence with the flux across facets. This is used both in the construction of the robust relaxation and robust prolongation operators. In Chapter 7 we turned our attention to the Scott-Vogelius discretisation. In addition to higher order convergence, the element is particularly attractive as it enforces the divergence constraint pointwise. This property in turn yields convergence of the velocity error that is robust in the Reynolds number. This was not the case for the discretisation in the previous chapter, for which we saw velocity errors that grew as the Reynolds number was increased.

As the Scott-Vogelius element is not stable for arbitrary meshes, we focus on the case of barycentrically refined meshes. The resulting hierarchies are nonnested, but a robust prolongation operator could be obtained using a suitable local modification of the standard prolongation. A robust relaxation method is obtained by considering a subspace correction method based on the macro star around vertices. We showed

two distinct approaches to derive this relaxation scheme, one based on discrete exact sequences and one based on building a particular Fortin operator by exploiting inf-sup stability on the macro elements. We emphasise that the second approach extends beyond the barycentrically refined case to meshes with different macro structures for which inf-sup stability is known.

To avoid the loss of adjoint consistency and the strong coupling due to the SUPG stabilisation utilised in Chapter 6, we employed an interior penalty type stabilisation. We confirmed using the method of manufactured solutions that the velocity and the pressure error converge at the expected rate and that the velocity error is now independent of the Reynolds number.

In two dimensions, we observed modest growth of iteration counts for the $[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$ element and slower growth or even constant iteration counts when the $[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$ element is used. We argued that the improved robustness in the higher order case is due to the nullspace of the stabilisation term (consisting of C^1 vector fields), which is captured by the relaxation when cubic velocity elements are used.

In three dimensions, the results were qualitatively similar: for the lowest order inf-sup stable discretisation, $[\mathbb{P}_3]^3-\mathbb{P}_2^{\text{disc}}$, we saw growth of the iteration counts due to the difficulty introduced by the stabilisation term. In three dimensions the kernel of the stabilisation is captured by quintic polynomials, and indeed we observed significantly improved robustness for the $[\mathbb{P}_5]^3-\mathbb{P}_4^{\text{disc}}$ element.

This work raises several questions and potential directions for future research. We begin by discussing the choice of stabilisation scheme. While the $[\mathbb{P}_3]^2-\mathbb{P}_2^{\text{disc}}$ discretisation that is required in two dimensions for robust performance is not impractical, the fifth order discretisation required for full robustness in three dimensions is very expensive. Here Powell-Sabin splits become attractive, as one can construct a local basis for C^1 functions consisting of piecewise quadratic functions on these splits both in two and three dimensions [GLN19, p. 1]. This means that we expect the scheme to be fully robust when employing the $[\mathbb{P}_2]^2-\mathbb{P}_1^{\text{disc}}$ or $[\mathbb{P}_2]^3-\mathbb{P}_1^{\text{disc}}$ element. However, this split is more difficult to implement and also increases the size of the local problems for the relaxation further. An alternative to the interior penalty type stabilisation is local projection stabilisation [BB04]. This stabilisation scheme is again adjoint consistent and was used successfully for the Stokes and Navier–Stokes equations [LRL08], but introduces other difficulties into the implementation as it augments the variational form with a term involving projections onto a coarse grid space.

A further question associated with advection is the question of the performance of the smoother for the top-left block. The theory we described only covers the symmetric

case without advection, and for this case we see complete robustness even on very coarse grids. When advection is added, we observe that a certain resolution of the coarse grid seems to be necessary for good performance. It would be desirable to both understand the nonsymmetric case better and also to find a way to be able to use coarse grids with fewer degrees of freedom while retaining robustness.

An alternative class of finite element pairs that enforce the divergence constraint exactly are given by $H(\mathrm{div})$-L^2 conforming discretisations. These elements are attractive for two reasons. First, they are inf-sup stable even at low order and on arbitrary meshes. Secondly, they enforce the divergence constraint exactly, even on nonaffine meshes. In addition, they are part of an exact discrete sequence and hence it is straightforward to characterise the kernel of the divergence. Local subspaces given by the star around vertices capture the kernel and yield a robust relaxation scheme. This has already been exploited by Arnold, Falk, and Winther [AFW97; AFW00] to develop robust multigrid methods for $H(\mathrm{div})$ and $H(\mathrm{curl})$ Riesz maps. This is attractive as the macro element structure and the resulting macro star required for the Scott-Vogelius element means that the subspaces for the relaxation involve many degrees of freedom. In three dimensions the size of the problems to be solved for each subspace can exceed a thousand degrees of freedom. This number is much smaller for the standard star around vertices.

A second advantage of these discretisations is that they do not require special prolongation operators. Since the divergence constraint is enforced exactly and a nested hierarchy can be used, discretely divergence-free vector fields on the coarse grid are exactly divergence-free and hence also divergence-free on the fine grid, since the prolongation is simply the identity.

We also note that for certain elements there exist interpolation operators $I : V \to V_h$ that preserve the divergence [GS03; XZ10]. Using such an operator for the prolongation could remove the need for the nestedness of the mesh hierarchy. This would simplify the use of the solver in the context of shape optimisation, enable anisotropic mesh adaptation, and allow for geometry fitting as the mesh is refined without requiring higher order finite elements.

Although we focussed on the steady Navier–Stokes equations, we highlight that the work extends directly to fully implicit time discretisations of the transient case [HR12]. In addition, by allowing for spatially varying viscosity, the ideas could be extended to the Reynolds-averaged Navier–Stokes equations. This opens up a large new class of engineering problems for which we do not expect a steady, laminar solution.

Thinking more broadly, the augmented Lagrangian approach is also attractive for many other mixed problems such as those arising in incompressible hyperelasticity, magnetohydrodynamics, or the modelling of liquid crystals. In all of these cases the approximation of the Schur complement can be simplified significantly by adding a singular term corresponding to the constraint to the top-left block. To obtain a fast solver, one then needs to locally characterise the kernel of the constraint in order to develop a robust and efficient multigrid scheme.

Bibliography

[AFW00] D. N. Arnold, R. S. Falk, and R. Winther. "Multigrid in $H(\mathrm{div})$ and $H(\mathrm{curl})$". In: *Numerische Mathematik* 85.2 (2000), pp. 197–217. DOI: 10.1007/pl00005386.

[AFW06] D. N. Arnold, R. S. Falk, and R. Winther. "Finite element exterior calculus, homological techniques, and applications". In: *Acta Numerica* 15 (2006), pp. 1–155. DOI: 10.1017/S0962492906210018.

[AFW97] D. N. Arnold, R. S. Falk, and R. Winther. "Preconditioning in $H(\mathrm{div})$ and applications". In: *Mathematics of Computation* 66.219 (1997), pp. 957–984. DOI: 10.1090/S0025-5718-97-00826-0.

[AGL05] J. Ahrens, B. Geveci, and C. Law. "Paraview: an end-user tool for large data visualization". In: *The visualization handbook* 717 (2005). DOI: 10.1016/b978-012387582-2/50038-1.

[ALM18] N. Ahmed, A. Linke, and C. Merdon. "Towards pressure-robust mixed methods for the incompressible Navier Stokes equations". In: *Computational Methods in Applied Mathematics* 18.3 (2018), pp. 353–372. DOI: 10.1515/cmam-2017-0047.

[Aln+14] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. "Unified Form Language: a domain-specific language for weak formulations of partial differential equations". In: *ACM Transactions on Mathematical Software* 40.2 (2014), 9:1–9:37. DOI: 10.1145/2566630.

[Ame+00] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. "MUMPS: a general purpose distributed memory sparse solver". In: *International Workshop on Applied Parallel Computing*. Springer. 2000, pp. 121–130. DOI: 10.1007/3-540-70734-4_16.

[And+11] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. "Efficient topology optimization in MATLAB using 88 lines of code". In: *Structural and Multidisciplinary Optimization* 43.1 (2011), pp. 1–16. DOI: 10.1007/s00158-010-0594-7.

[Arn+02] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. "Unified analysis of discontinuous Galerkin methods for elliptic problems". In: *SIAM Journal on Numerical Analysis* 39.5 (2002), pp. 1749–1779. DOI: 10.1137/S0036142901384162.

[Att96] H. Attouch. "Viscosity solutions of minimization problems". In: *SIAM Journal on Optimization* 6.3 (1996), pp. 769–806. DOI: {10.1137/S1052623493259616}.

[Bal+16] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. Smith, S. Zampini, H. Zhang, and H. Zhang. *PETSc Web page*. 2016. URL: http://www.mcs.anl.gov/petsc.

[Bal+18] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, D. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. Smith, S. Zampini, H. Zhang, and H. Zhang. *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.10. Argonne National Laboratory, 2018. URL: http://www.mcs.anl.gov/petsc.

[Bal+97] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. "Efficient management of parallelism in object oriented numerical software libraries". In: *Modern Software Tools in Scientific Computing*. Birkhäuser Press, 1997, 163–202. DOI: 10.1007/978-1-4612-1986-6_8.

[BB04] R. Becker and M. Braack. "A two-level stabilization scheme for the Navier-Stokes equations". In: *Numerical Mathematics and Advanced Applications*. Springer Berlin Heidelberg, 2004, pp. 123–130. DOI: 10.1007/978-3-642-18775-9_9.

[BB10] S. Börm and S. L. Borne. "$\mathcal{H}$-LU factorization in preconditioners for augmented Lagrangian and grad-div stabilized saddle point systems". In: *International Journal for Numerical Methods in Fluids* 68.1 (2010), pp. 83–98. DOI: 10.1002/fld.2495.

[BC03] B. Bourdin and A. Chambolle. "Design-dependent loads in topology optimization". In: *ESAIM: Control, Optimisation and Calculus of Variations* 9 (2003), pp. 19–48. DOI: 10.1051/cocv:2002070.

[Beg05] H. Begehr. "Boundary value problems in complex analysis I". In: *Bol. Asoc. Mat. Venezolana* 12 (2005), pp. 65–85.

[Bel+97] J. A. Bello, E. Fernández-Cara, J. Lemoine, and J. Simon. "The differentiability of the drag with respect to the variations of a Lipschitz domain in a Navier–Stokes flow". In: *SIAM Journal on Control and Optimization* 35.2 (1997), pp. 626–640. DOI: 10.1137/S0363012994278213.

[Ben89] M. P. Bendsøe. "Optimal shape design as a material distribution problem". In: *Structural Optimization* 1.4 (1989), pp. 193–202. DOI: 10.1007/BF01650949.

[Ber10] M. Berggren. "A unified discrete–continuous sensitivity analysis method for shape optimization". In: *Applied and Numerical Partial Differential Equations*. Vol. 15. 2010, pp. 25–39. DOI: 10.1007/978-90-481-3239-3_4.

[BFH06] E. Burman, M. A. Fernández, and P. Hansbo. "Continuous interior penalty finite element method for Oseen's equations". In: *SIAM Journal on Numerical Analysis* 44.3 (2006), pp. 1248–1274. DOI: 10.1137/040617686.

[BGL05] M. Benzi, G. H. Golub, and J. Liesen. "Numerical solution of saddle point problems". In: *Acta Numerica* 14 (2005), pp. 1–137. DOI: 10.1017/S0962492904-000212.

[BH04] E. Burman and P. Hansbo. "Edge stabilization for Galerkin approximations of convection–diffusion–reaction problems". In: *Computer Methods in Applied Mechanics and Engineering*. Recent Advances in Stabilized and Multiscale Finite Element Methods 193.15 (2004), pp. 1437–1453. DOI: 10.1016/j.cma.2003.12.032.

[BH06] E. Burman and P. Hansbo. "Edge stabilization for the generalized Stokes problem: A continuous interior penalty method". In: *Computer Methods in Applied Mechanics and Engineering* 195.19 (2006), pp. 2393–2410. DOI: 10.1016/j.cma.2005.05.009.

[BH82] A. N. Brooks and T. J. R. Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 32.1-3 (1982), pp. 199–259. DOI: 10.1016/0045-7825(82)90071-8.

[BL08] E. Burman and A. Linke. "Stabilized finite element schemes for incompressible flow using Scott–Vogelius elements". In: *Applied Numerical Mathematics* 58.11 (2008), pp. 1704–1719. DOI: 10.1016/j.apnum.2007.11.001.

[BL11] A. Brandt and O. E. Livne. *Multigrid techniques: 1984 guide with applications to fluid dynamics*. Rev. ed. Philadelphia: Society for Industrial and Applied Mathematics, 2011. DOI: 10.1137/1.9781611970753.

[BL97] D. Boffi and C. Lovadina. "Analysis of new augmented Lagrangian formulations for mixed finite element schemes". In: *Numerische Mathematik* 75.4 (1997), pp. 405–419. DOI: 10.1007/s002110050246.

[BO06] M. Benzi and M. A. Olshanskii. "An augmented Lagrangian-based approach to the Oseen problem". In: *SIAM Journal on Scientific Computing* 28.6 (2006), pp. 2095–2113. DOI: 10.1137/050646421.

[BO11] M. Benzi and M. A. Olshanskii. "Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier–Stokes problem". In: *SIAM Journal of Numerical Analysis* 49.2 (2011), pp. 770–788. DOI: 10.1137/100806485.

[Bof+08] D. Boffi, F. Brezzi, L. F. Demkowicz, R. G. Durán, R. S. Falk, and M. Fortin. *Mixed Finite Elements, Compatibility Conditions, and Applications*. Vol. 1939. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-78319-0.

[BOW11] M. Benzi, M. A. Olshanskii, and Z. Wang. "Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations". In: *International Journal for Numerical Methods in Fluids* 66.4 (2011), pp. 486–508. DOI: 10.1002/fld.2267.

[BR85a] C. Bernardi and G. Raugel. "A conforming finite element method for the time-dependent Navier–Stokes equations". In: *SIAM Journal on Numerical Analysis* 22.3 (1985), pp. 455–473. DOI: 10.1137/0722027.

[BR85b] C. Bernardi and G. Raugel. "Analysis of some finite elements for the Stokes problem". In: *Mathematics of Computation* 44.169 (1985), pp. 71–79. DOI: 10.2307/2007793.

[Bra+09] C. Brandenburg, F. Lindemann, M. Ulbrich, and S. Ulbrich. "A continuous adjoint approach to shape optimization for Navier Stokes flow". In: *Optimal control of coupled systems of partial differential equations*. Springer, 2009, pp. 35–56. DOI: 10.1007/978-3-7643-8923-9_2.

[Bra+12] C. Brandenburg, F. Lindemann, M. Ulbrich, and S. Ulbrich. "Advanced numerical methods for PDE constrained optimization with application to optimal design in Navier Stokes flow". In: *Constrained Optimization and Optimal Control for Partial Differential Equations*. Springer, 2012, pp. 257–275. DOI: 10.1007/978-3-0348-0133-1_14.

[Bra82] A. Brandt. "Guide to multigrid development". In: *Multigrid Methods*. Vol. 960. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 220–312. DOI: 10.1007/BFb0069930.

[Bre11] S. C. Brenner. "C^0 interior penalty methods". In: *Frontiers in Numerical Analysis - Durham 2010*. Vol. 85. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 79–147. DOI: 10.1007/978-3-642-23914-4_2.

[Bro81] P. Brown. "A noninteractive method for the automatic generation of finite element meshes using the Schwarz-Christoffel transformation". In: *Computer Methods in Applied Mechanics and Engineering* 25.1 (1981), pp. 101–126. DOI: 10.1016/0045-7825(81)90071-2.

[Bru+15] P. R. Brune, M. G. Knepley, B. F. Smith, and X. Tu. "Composing scalable nonlinear algebraic solvers". In: *SIAM Review* 57.4 (2015), pp. 535–565. DOI: 10.1137/130936725.

[BS05] S. C. Brenner and L.-Y. Sung. "C^0 interior penalty methods for fourth order elliptic boundary value problems on polygonal domains". In: *Journal of Scientific Computing* 22-23.1-3 (2005), pp. 83–118. DOI: 10.1007/s10915-004-4135-7.

[BS08] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Third edition. Vol. 15. Texts in Applied Mathematics. Springer-Verlag New York, 2008. DOI: 10.1007/978-0-387-75934-0.

[BS97] D. Braess and R. Sarazin. "An efficient smoother for the Stokes problem". In: *Applied Numerical Mathematics* 23.1 (1997), pp. 3–19. DOI: 10.1016/S0168-9274(96)00059-1.

[CCS05] J. Carrero, B. Cockburn, and D. Schötzau. "Hybridized globally divergence-free LDG methods. Part I: the Stokes problem". In: *Mathematics of Computation* 75.254 (2005), pp. 533–564. DOI: 10.1090/S0025-5718-05-01804-1.

[Cho67] A. J. Chorin. "A numerical method for solving incompressible viscous flow problems". In: *Journal of Computational Physics* 2.1 (1967), pp. 12–26. DOI: 10.1016/0021-9991(67)90037-x.

[Cia02] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Vol. 40. Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898719208.

[CKS04] B. Cockburn, G. Kanschat, and D. Schötzau. "A locally conservative LDG method for the incompressible Navier-Stokes equations". In: *Mathematics of Computation* 74.251 (2004), pp. 1067–1096. DOI: 10.1090/S0025-5718-04-01718-1.

[CKS06] B. Cockburn, G. Kanschat, and D. Schötzau. "A note on discontinuous Galerkin divergence-free solutions of the Navier-Stokes equations". In: *Journal of Scientific Computing* 31.1-2 (2006), pp. 61–73. DOI: 10.1007/s10915-006-9107-7.

[CM10] M. Costabel and A. McIntosh. "On Bogovskiĭ and regularised Poincaré integral operators for de Rham complexes on Lipschitz domains". In: *Mathematische Zeitschrift* 265.2 (2010), pp. 297–320. DOI: 10.1007/s00209-009-0517-8.

[Con95] J. B. Conway. *Functions of One Complex Variable*. 2nd ed. 11. New York: Springer-Verlag, 1995. DOI: 10.1007/978-1-4612-6313-5.

[DD76] J. Douglas and T. Dupont. "Interior penalty procedures for elliptic and parabolic Galerkin methods". In: *Computing Methods in Applied Sciences*. Vol. 58. Springer Berlin Heidelberg, 1976, pp. 207–216. DOI: 10.1007/BFb0120591.

[Dwi09] R. P. Dwight. "Robust mesh deformation using the linear elasticity equations". In: *Computational Fluid Dynamics 2006*. Springer Berlin Heidelberg, 2009, pp. 401–406. DOI: 10.1007/978-3-540-92779-2_62.

[DZ11] M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. Society for Industrial and Applied Mathematics, 2011. DOI: 10.1137/1.9780898719826.

[EE01] M. Eiermann and O. G. Ernst. "Geometric aspects of the theory of Krylov subspace methods". In: *Acta Numerica* 10 (2001). DOI: 10.1017/S096249290100-0046.

[EG04] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Vol. 159. Applied Mathematical Sciences. New York, NY: Springer New York, 2004. DOI: 10.1007/978-1-4757-4355-5.

[EG15] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 2015. DOI: 10.1201/b18333.

[EH18] T. Etling and R. Herzog. "Optimum experimental design by shape optimization of specimens in linear elasticity". In: *SIAM Journal on Applied Mathematics* 78.3 (2018), pp. 1553–1576. DOI: 10.1137/17M1147743.

[Elm+06] H. C. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. "Block preconditioners based on approximate commutators". In: *SIAM Journal on Scientific Computing* 27.5 (2006), pp. 1651–1668. DOI: 10.1137/040608817.

[ELW03] H. C. Elman, D. Loghin, and A. J. Wathen. "Preconditioning techniques for Newton's method for the incompressible Navier-Stokes equations". In: *BIT Numerical Mathematics* 43.5 (2003), pp. 961–974. DOI: 10.1023/b:bitn.0000014565.86918.df.

[ERS14] H. C. Elman, A. Ramage, and D. J. Silvester. "IFISS: a computational laboratory for investigating incompressible flow problems". In: *SIAM Review* 56.2 (2014), pp. 261–273. DOI: 10.1137/120891393.

[ES18] M. Eigel and K. Sturm. "Reproducing kernel Hilbert spaces and variable metric algorithms in PDE-constrained shape optimization". In: *Optimization Methods and Software* 33.2 (2018), pp. 268–296. DOI: 10.1080/10556788.2017.1314471.

[ES96] H. C. Elman and D. Silvester. "Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations". In: *SIAM Journal on Scientific Computing* 17.1 (1996), pp. 33–46. DOI: 10.1137/0917004.

[ESW14] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014. DOI: 10.1093/acprof:oso/9780199678792.001.0001.

[Etl+18] T. Etling, R. Herzog, E. Loayza, and G. Wachsmuth. "First and second order shape optimization based on restricted mesh deformations". In: *arXiv:1810.10313* (2018).

[EW92] R. E. Ewing and J. Wang. "Analysis of the Schwarz algorithm for mixed finite elements methods". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 26.6 (1992), pp. 739–756. DOI: 10.1051/m2an/1992260607391.

[Far+13] P. E. Farrell, D. A. Ham, S. W. Funke, and M. E. Rognes. "Automated derivation of the adjoint of high-level transient finite element programs". In: *SIAM Journal on Scientific Computing* 35.4 (2013), pp. C369–C393. DOI: 10.1137/120873558.

[FG83] M. Fortin and R. Glowinski. *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*. Vol. 15. Studies in Mathematics and Its Applications. Elsevier Science Ltd, 1983. DOI: 10.1002/zamm.19850651211.

[FGN18] G. Fu, J. Guzman, and M. Neilan. "Exact smooth piecewise polynomial sequences on Alfeld splits". In: *arXiv:1807.05883 [math]* (2018). arXiv: 1807.05883.

[Fir19] Firedrake-Zenodo. *Software used in 'Shape optimisation examples for 'Shape Optimisation and Robust Solvers for Incompressible Flow"*. 2019. DOI: 10.5281/zenodo.3369183. URL: https://zenodo.org/record/3369183.

[FMW19] P. E. Farrell, L. Mitchell, and F. Wechsung. "An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier-Stokes equations at high Reynolds number". In: *SIAM Journal on Scientific Computing* (2019). accepted.

[For77] M. Fortin. "An analysis of the convergence of mixed finite element methods". In: *RAIRO. Analyse numérique* 11.4 (1977), pp. 341–354. DOI: 10.1051/m2an/1977110403411.

[FP01] L. Formaggia and S. Perotto. "New anisotropic a priori error estimates". In: *Numerische Mathematik* 89.4 (2001), pp. 641–667. DOI: 10.1007/s002110100273.

[FY02] R. D. Falgout and U. M. Yang. "Hypre: A library of high performance preconditioners". In: *Computational Science — ICCS 2002*. Vol. 2331. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 632–641. DOI: 10.1007/3-540-47789-6_66.

[Gam01] T. W. Gamelin. *Complex Analysis*. Undergraduate Texts in Mathematics. New York, NY: Springer New York, 2001. DOI: 10.1007/978-0-387-21607-2.

[Gar+16] H. Garcke, C. Hecht, M. Hinze, C. Kahle, and K. F. Lam. "Shape optimization for surface functionals in Navier-Stokes flow using a phase field approach". In: *Interfaces and Free Boundaries* 18.2 (2016), pp. 219–261. ISSN: 1463-9963. DOI: 10.4171/IFB/363.

[Gee+06] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. *ML 5.0 smoothed aggregation user's guide*. Tech. rep. SAND2006-2649, Sandia National Laboratories, 2006.

[Gel+05] T. Gelhard, G. Lube, M. A. Olshanskii, and J.-H. Starcke. "Stabilized finite element schemes with LBB-stable elements for incompressible flows". In: *Journal of Computational and Applied Mathematics* 177.2 (2005), pp. 243–267. DOI: 10.1016/j.cam.2004.09.017.

[GGV03] C. Greif, G. H. Golub, and J. M. Varah. "Augmented lagrangian techniques for solving saddle point linear systems". In: *SIAM Journal on Matrix Analysis and Applications* 27 (2003), pp. 779–792.

[GHS14] A. Günnel, R. Herzog, and E. Sachs. "A note on preconditioners and scalar products in Krylov subspace methods for self-adjoint problems in Hilbert space". In: *Electron. Trans. Numer. Anal.* 41 (2014), pp. 13–20.

[GLN19] J. Guzman, A. Lischke, and M. Neilan. "Exact sequences on Powell-Sabin splits". In: *arXiv:1904.05466* (2019).

[GM78] J. D. Gray and S. A. Morris. "When is a function that satisfies the Cauchy-Riemann equations analytic?" In: *The American Mathematical Monthly* 85.4 (1978), p. 246. DOI: 10.2307/2321164.

[GPS96] A. Greenbaum, V. Pták, and Z. Strakoš. "Any nonincreasing convergence curve is possible for GMRES". In: *SIAM Journal on Matrix Analysis and Applications* 17.3 (1996), pp. 465–469. DOI: 10.1137/S0895479894275030.

[GR09] C. Geuzaine and J.-F. Remacle. "Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. DOI: 10.1002/nme.2579.

[GR86] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Vol. 5. Springer Series in Computational Mathematics. Springer, 1986. DOI: 10.1002/zamm.19870671119.

[Gri11] P. Grisvard. *Elliptic Problems in Nonsmooth Domains*. Society for Industrial and Applied Mathematics, 2011. DOI: 10.1137/1.9781611972030.

[GS03] V. Girault and L. R. Scott. "A quasi-local interpolation operator preserving the discrete divergence". In: *Calcolo* 40.1 (2003), pp. 1–19. DOI: 10.1007/s100920300000.

[Hag89] W. W. Hager. "Updating the inverse of a matrix". In: *SIAM Review* 31.2 (1989), pp. 221–239. DOI: 10.1137/1031049.

[Ham+19] D. A. Ham, L. Mitchell, A. Paganini, and F. Wechsung. "Automated shape differentiation in the Unified Form Language". In: *Structural and Multidisciplinary Optimization* (2019). accepted.

[HBH10] S. Hamilton, M. Benzi, and E. Haber. "New multigrid smoothers for the Oseen problem". In: *Numerical Linear Algebra with Applications* (2010), pp. 557–576. DOI: 10.1002/nla.707.

[Hel03] B. T. Helenbrook. "Mesh deformation using the biharmonic operator". In: *International Journal for Numerical Methods in Engineering* 56.7 (2003), pp. 1007–1021. DOI: 10.1002/nme.595.

[Hic+13] J. E. Hicken, J. Li, O. Sahni, and A. A. Oberai. "Adjoint consistency analysis of residual-based variational multiscale methods". In: *Journal of Computational Physics* 255 (2013), pp. 396–406. DOI: 10.1016/j.jcp.2013.07.039.

[Hin+09] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer, 2009. DOI: 10.1007/978-1-4020-8839-1.

[HKT08] P. Hajlasz, P. Koskela, and H. Tuominen. "Sobolev embeddings, extensions and measure density condition". In: *Journal of Functional Analysis* 254.5 (2008), pp. 1217–1234. DOI: 10.1016/j.jfa.2007.11.020.

[HL19] R. Hohmann and C. Leithäuser. "Shape optimization of a polymer distributor using an Eulerian residence time model". In: *SIAM Journal on Scientific Computing* 41.4 (2019), B625–B648. DOI: 10.1137/18M1225847.

[HMT07] S. Hofmann, M. Mitrea, and M. Taylor. "Geometric and transformational properties of Lipschitz domains, Semmes-Kenig-Toro domains, and other classes of finite perimeter domains". In: *Journal of Geometric Analysis* 17.4 (2007), pp. 593–647. DOI: 10.1007/BF02937431.

[HNC11] X. He, M. Neytcheva, and S. S. Capizzano. "On an augmented Lagrangian-based preconditioning of Oseen type problems". In: *BIT Numerical Mathematics* 51.4 (2011), pp. 865–888. DOI: 10.1007/s10543-011-0334-4.

[HP15] R. Hiptmair and A. Paganini. "Shape optimization by pursuing diffeomorphisms". In: *Computational Methods in Applied Mathematics* 15.3 (2015). DOI: 10.1515/cmam-2015-0013.

[HPS15] R. Hiptmair, A. Paganini, and S. Sargheini. "Comparison of approximate shape gradients". In: *BIT* 55.2 (2015), pp. 459–485. DOI: 10.1007/s10543-014-0515-z.

[HR12] T. Heister and G. Rapin. "Efficient augmented Lagrangian-type preconditioning for the Oseen problem using grad-div stabilization". In: *International Journal for Numerical Methods in Fluids* 71.1 (2012), pp. 118–134. DOI: 10.1002/fld.3654.

[HS15] R. Herzog and E. Sachs. "Superlinear convergence of Krylov subspace methods for self-adjoint problems in Hilbert space". In: *SIAM Journal on Numerical Analysis* 53.3 (2015), pp. 1304–1324. DOI: 10.1137/140973050.

[HT74] P. Hood and C. Taylor. "Navier-Stokes equations using mixed interpolation". In: *Finite element methods in flow problems* (1974), pp. 121–132.

[HT97] R. Hiptmair and A. Toselli. "Overlapping Schwarz methods for vector valued elliptic problems in three dimensions". In: *IMA Volumes in Mathematics and its Applications*. 1997. DOI: 10.1007/978-1-4612-1176-1_8.

[HVK18] X. He, C. Vuik, and C. M. Klaij. "Combining the augmented Lagrangian preconditioner with the SIMPLE Schur complement approximation". In: *SIAM Journal on Scientific Computing* 40.3 (2018), A1362–A1385. DOI: 10.1137/17M1144775.

[HX07] R. Hiptmair and J. Xu. "Nodal auxiliary space preconditioning in H(curl) and H(div) spaces". In: *SIAM Journal on Numerical Analysis* 45.6 (2007), pp. 2483–2509. DOI: 10.1137/060660588.

[IKO11] T. Iwaniec, L. V. Kovalev, and J. Onninen. “The Nitsche conjecture”. In: *Journal of the American Mathematical Society* 24.2 (2011), pp. 345–345. DOI: 10.1090/S0894-0347-2010-00685-6.

[IM01] T. Iwaniec and G. Martin. *Geometric Function Theory and Non-linear Analysis*. Oxford mathematical monographs. Oxford ; New York: Oxford University Press, 2001. DOI: 10.1112/S0024609302241569.

[Ips01] I. C. F. Ipsen. “A note on preconditioning nonsymmetric matrices”. In: *SIAM Journal on Scientific Computing* 23.3 (2001), pp. 1050–1051. DOI: 10.1137/S1064827500377435.

[ISW18] J. A. Iglesias, K. Sturm, and F. Wechsung. “Two-dimensional shape optimization with nearly conformal transformations”. In: *SIAM Journal on Scientific Computing* 40.6 (2018), A3807–A3830. DOI: 10.1137/17M1152711.

[Ive82] D. C. Ives. “Conformal grid generation”. In: *Applied Mathematics and Computation* 10-11 (1982), pp. 107–135. DOI: 10.1016/0096-3003(82)90189-8.

[Joh+17] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz. “On the divergence constraint in mixed finite element methods for incompressible flows”. In: *SIAM Review* 59.3 (2017), pp. 492–544. DOI: 10.1137/15m1047696.

[Kar82] O. Karakashian. “On a Galerkin–Lagrange multiplier method for the stationary Navier–Stokes equations”. In: *SIAM Journal on Numerical Analysis* 19.5 (1982), pp. 909–923. DOI: 10.1137/0719066.

[Keu15] M. M. Keuthen. “Second Order Shape Optimization with Geometric Constraints”. PhD thesis. Technische Universität München, 2015.

[KK05] M. G. Knepley and D. A. Karpeev. *Flexible representation of computational meshes*. Tech. rep. ANL/MCS-P1295-1005. Argonne National Laboratory, 2005.

[KK09] M. G. Knepley and D. A. Karpeev. “Mesh algorithms for PDE with Sieve I: mesh distribution”. In: *Scientific Programming* 17.3 (2009), pp. 215–230. DOI: 10.1155/2009/948613.

[KLW02] D. Kay, D. Loghin, and A. Wathen. “A preconditioner for the steady-state Navier–Stokes equations”. In: *SIAM Journal on Scientific Computing* 24.1 (2002), pp. 237–256. DOI: 10.1137/S106482759935808X.

[KM18] R. C. Kirby and L. Mitchell. “Solver composition across the PDE/linear algebra barrier”. In: *SIAM Journal on Scientific Computing* 40.1 (2018), pp. C76–C98. DOI: 10.1137/17M1133208.

[Knu07] P. Knupp. “Remarks on mesh quality, in ’45th AIAA Aerospace Sciences Meeting and Exhibit’”. In: *Sandia National Laboratories, New Mexico, USA* (2007).

[KO88] V. A. Kondrat'ev and O. A. Oleinik. "Boundary-value problems for the system of elasticity theory in unbounded domains. Korn's inequalities". In: *Russian Mathematical Surveys* 43.5 (1988), pp. 65–119. DOI: 10.1070/RM1988v043n-05ABEH001945.

[Kob95] G. M. Kobel'kov. "On solving the Navier–Stokes equations at large Reynolds numbers". In: *Russian Journal of Numerical Analysis and Mathematical Modelling* 10.1 (1995). DOI: 10.1515/rnam.1995.10.1.33.

[KU14] M. Keuthen and M. Ulbrich. "Moreau-Yosida regularisation in shape optimization with geometric constraints". In: *Computational Optimization and Applications* 62.1 (2014), pp. 181–216. DOI: 10.1007/s10589-014-9661-0.

[LD03] X. S. Li and J. W. Demmel. "SuperLU_DIST: a scalable distributed-memory sparse direct solver for unsymmetric linear systems". In: *ACM Transactions on Mathematical Software* 29.2 (2003), pp. 110–140. DOI: 10.1145/779359.779361.

[Lee+07] Y.-J. Lee, J. Wu, J. Xu, and L. Zikatanov. "Robust subspace correction methods for nearly singular systems". In: *Mathematical Models and Methods in Applied Sciences* 17.11 (2007), pp. 1937–1963. DOI: 10.1142/s0218202507002522.

[LMW12] A. Logg, K.-A. Mardal, and G. Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*. Vol. 84. Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-23099-8.

[LR19] A. Linke and L. G. Rebholz. "Pressure-induced locking in mixed methods for time-dependent (Navier–)Stokes equations". In: *Journal of Computational Physics* 388 (2019), pp. 350–356. DOI: 10.1016/j.jcp.2019.03.010.

[LRL08] G. Lube, G. Rapin, and J. Löwe. "Local projection stabilization of finite element methods for incompressible flows". In: *Numerical Mathematics and Advanced Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 481–488. DOI: 10.1007/978-3-540-69777-0_57.

[LW04] D. Loghin and A. J. Wathen. "Analysis of preconditioners for saddle-point problems". In: *SIAM Journal on Scientific Computing* 25.6 (2004), pp. 2029–2049. DOI: 10.1137/S1064827502418203.

[LWC09] Y.-J. Lee, J. Wu, and J. Chen. "Robust multigrid method for the planar linear elasticity problems". In: *Numerische Mathematik* 113.3 (2009), pp. 473–496. DOI: 10.1007/s00211-009-0232-8.

[May+16] D. A. May, P. Sanan, K. Rupp, M. G. Knepley, and B. F. Smith. "Extreme-scale multigrid components within PETSc". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. 2016. DOI: 10.1145/2929908.2929913.

[MFD19] S. Mitusch, S. Funke, and J. Dokken. "Dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake". In: *Journal of Open Source Software* 4.38 (2019), p. 1292. DOI: 10.21105/joss.01292.

[MGW00] M. F. Murphy, G. H. Golub, and A. J. Wathen. "A note on preconditioning for indefinite linear systems". In: *SIAM Journal on Scientific Computing* 21.6 (2000), pp. 1969–1972. DOI: 10.1137/S1064827599355153.

[Mic72] A. M. Micheletti. "Metrica per familglie di domini limitati a proprietá generiche degli autovalori". In: *Annali della Scuola Normale Superiore di Pisa* 26.3 (1972), pp. 683–694.

[MM16] L. Mitchell and E. H. Müller. "High level implementation of geometric multigrid solvers for finite element problems: Applications in atmospheric modelling". In: *Journal of Computational Physics* 327 (2016), pp. 1–18. DOI: 10.1016/j.jcp.2016.09.037.

[Mor+12] P. Morin, R. H. Nochetto, M. S. Pauletti, and M. Verani. "Adaptive finite element method for shape optimization". In: *ESAIM: Control, Optimisation and Calculus of Variations* 18.4 (2012), pp. 1122–1149. DOI: 10.1051/cocv/2011192.

[MS76] F. Murat and J. Simon. "Etude de problemes d'optimal design". In: *Optimization Techniques Modeling and Optimization in the Service of Man Part 2*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1976, pp. 54–62.

[MW11] K.-A. Mardal and R. Winther. "Preconditioning discretizations of systems of partial differential equations". In: *Numerical Linear Algebra with Applications* 18.1 (2011), pp. 1–40. DOI: 10.1002/nla.716.

[Noc80] J. Nocedal. "Updating quasi-Newton matrices with limited storage". In: *Mathematics of Computation* 35.151 (1980), p. 773. DOI: 10.2307/2006193.

[NP02] A. Novruzi and M. Pierre. "Structure of shape derivatives". In: *Journal of Evolution Equations* 2.3 (2002), pp. 365–382. DOI: 10.1007/s00028-002-8093-y.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Berlin: Springer, 2006. DOI: 10.1007/978-0-387-40065-5.

[NW07] A. C. de Niet and F. W. Wubs. "Two preconditioners for saddle point problems in fluid flows". In: *International Journal for Numerical Methods in Fluids* 54.4 (2007), pp. 355–377. DOI: 10.1002/fld.1401.

[OB08] M. A. Olshanskii and M. Benzi. "An augmented Lagrangian approach to linearized problems in hydrodynamic stability". In: *SIAM Journal on Scientific Computing* 30.3 (2008), pp. 1459–1473. DOI: 10.1137/070691851.

[OD09] T. A. Oliver and D. L. Darmofal. "Analysis of dual consistency for discontinuous Galerkin discretizations of source terms". In: *SIAM Journal on Numerical Analysis* 47.5 (2009), pp. 3507–3525. DOI: 10.1137/080721467.

[Ols02] M. A. Olshanskii. "A low order Galerkin finite element method for the Navier-Stokes equations of steady incompressible flow: a stabilization issue and iterative methods". In: *Computer Methods in Applied Mechanics and Engineering* 191.47-48 (2002), pp. 5515–5536. DOI: 10.1016/S0045-7825(02)00513-3.

[OR03] M. A. Olshanskii and A. Reusken. "Grad-div stablilization for Stokes equations". In: *Mathematics of Computation* 73.248 (2003), pp. 1699–1719. DOI: 10.1090/s0025-5718-03-01629-6.

[OR04] M. A. Olshanskii and A. Reusken. "Convergence analysis of a multigrid method for a convection-dominated model problem". In: *SIAM Journal on Numerical Analysis* 42.3 (2004), pp. 1261–1291. DOI: 10.1137/s0036142902418679.

[OT14] M. A. Olshanskii and E. E. Tyrtyshnikov. *Iterative Methods for Linear Systems: Theory and Applications*. Applied mathematics. Philadelphia: Society for Industrial and Applied Mathematics, 2014. DOI: 10.1137/1.9781611973464.

[Pat80] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Series in computational methods in mechanics and thermal sciences. New York: Hemisphere Publ. Co, 1980. DOI: 10.1201/9781482234213.

[PEA15] F. Palacios, T. D. Economon, and J. J. Alonso. "Large-scale aircraft design using SU2". In: *53rd AIAA Aerospace Sciences Meeting*. Kissimmee, Florida: American Institute of Aeronautics and Astronautics, 2015. DOI: 10.2514/6.2015-1946.

[PG97] N. A. Pierce and M. B. Giles. "Preconditioned multigrid methods for compressible flow calculations on stretched meshes". In: *Journal of Computational Physics* 136.2 (1997), pp. 425–445. DOI: 10.1006/jcph.1997.5772.

[Pir74] O. Pironneau. "On optimum design in fluid mechanics". In: *Journal of Fluid Mechanics* 64.1 (1974), pp. 97–110. DOI: 10.1017/S0022112074002023.

[PWF18] A. Paganini, F. Wechsung, and P. E. Farrell. "Higher-order moving mesh methods for PDE-constrained shape optimization". In: *SIAM Journal on Scientific Computing* 40.4 (2018), A2356–A2382. DOI: 10.1137/17M1133956.

[Qin94] J. Qin. "On the Convergence of Some Low Order Mixed Finite Elements for Incompressible Fluids". PhD thesis. Pennsylvania State University, 1994.

[QV08] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Vol. 23. Springer Series in Computational Mathematics. Springer, 2008. DOI: 10.1007/978-3-540-85268-1.

[Ram99] A. Ramage. "A multigrid preconditioner for stabilised discretisations of advection-diffusion problems". In: *Journal of Computational and Applied Mathematics* 110.1 (1999), pp. 187–203. DOI: 10.1016/s0377-0427(99)00234-4.

[Rat+16] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. Mcrae, G.-T. Bercea, G. R. Markall, P. H. J. Kelly, D. A. Ham, and P. H. J. Kelly. "Firedrake: automating the finite element method by composing abstractions". In: *ACM Transactions on Mathematical Software* 43.24 (2016). DOI: 10.1145/2998441.

[Riz81] A. Rizzi. "Computational mesh for transonic airfoils". In: *Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves*. Wiesbaden: Vieweg+Teubner Verlag, 1981, pp. 222–253. DOI: 10.1007/978-3-663-14008-5_18.

[RK14] D. Ridzal and D. Kouri. *Rapid Optimization Library*. 2014. URL: https://trilinos.org/packages/rol/.

[Rui+19] E. Ruiz-Gironés, A. Gargallo-Peiró, J. Sarrate, and X. Roca. "Automatically imposing incremental boundary displacements for valid mesh morphing and curving". In: *Computer-Aided Design* 112 (2019), pp. 47–62. DOI: 10.1016/j.cad.2019.01.001.

[RVS08] M. ur Rehman, C. Vuik, and G. Segal. "A comparison of preconditioners for incompressible Navier-Stokes solvers". In: *International Journal for Numerical Methods in Fluids* 57.12 (2008), pp. 1731–1751. DOI: 10.1002/fld.1684.

[Saa93] Y. Saad. "A flexible inner-outer preconditioned GMRES algorithm". In: *SIAM Journal on Scientific Computing* 14.2 (1993), pp. 461–469. DOI: 10.1137/0914028.

[Sam04] J. Samareh. "Aerodynamic shape optimization based on free-form deformation". In: *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*. 2004. DOI: 10.2514/6.2004-4630.

[Sch10] S. Schmidt. "Efficient Large Scale Aerodynamic Design Based on Shape Calculus". PhD thesis. University of Trier, Germany, 2010.

[Sch14] S. Schmidt. "A two stage CVT / Eikonal convection mesh deformation approach for large nodal deformations". In: (2014). eprint: arXiv:1411.7663 (cs.NA).

[Sch18] S. Schmidt. "Weak and strong form shape Hessian's and their automatic generation". In: *SIAM Journal on Scientific Computing* 40.2 (2018), pp. C210–C233. DOI: 10.1137/16M1099972.

[Sch77] F. Schottky. "Über die conforme Abbildung mehrfach zusammenhängender ebener Flächen". In: *Journal für die reine und angewandte Mathematik* 83 (1877).

[Sch98] J. Schöberl. “Robust multigrid preconditioning for parameter-dependent problems I: the Stokes-type case”. In: *Multigrid Methods V*. Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, 1998, pp. 260–275. DOI: 10.1007/978-3-642-58734-4_15.

[Sch99a] J. Schöberl. “Multigrid methods for a parameter dependent problem in primal variables”. In: *Numerische Mathematik* 84.1 (1999), pp. 97–119. DOI: 10.1007/s002110050465.

[Sch99b] J. Schöberl. “Robust Multigrid Methods for Parameter Dependent Problems”. PhD thesis. Linz, Austria: Johannes Kepler Universität Linz, 1999.

[SHJ91] F. Shakib, T. J. R. Hughes, and Z. Johan. “A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 89.1-3 (1991), pp. 141–219. DOI: 10.1016/0045-7825(91)90041-4.

[Sig01] O. Sigmund. “A 99 line topology optimization code written in MATLAB”. In: *Structural and Multidisciplinary Optimization* 21.2 (2001), pp. 120–127. DOI: 10.1007/s001580050176.

[Sim80] J. Simon. “Differentiation with respect to the domain in boundary value problems”. In: *Numerical Functional Analysis and Optimization* 2.7-8 (1980), pp. 649–687. DOI: 10.1080/01630563.1980.10120631.

[SP98] O. Sigmund and J. Petersson. “Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima”. In: *Structural and Multidisciplinary Optimization* 16.1 (1998), pp. 68–75. DOI: 10.1007/BF01214002.

[SS16] V. Schulz and M. Siebenborn. “Computational comparison of surface metrics for PDE constrained shape optimization”. In: *Computational Methods in Applied Mathematics* 16.3 (2016), pp. 485–496. DOI: 10.1515/cmam-2016-0009.

[SSW18] S. Schmidt, M. Schütte, and A. Walther. “Efficient numerical solution of geometric inverse problems involving Maxwell’s equations using shape derivatives and automatic code generation”. In: *SIAM Journal on Scientific Computing* 40.2 (2018), B405–B428. DOI: 10.1137/16M110602X.

[Sta97] G. Starke. “Field-of-values analysis of preconditioned iterative methods for nonsymmetric elliptic problems”. In: *Numerische Mathematik* 78.1 (1997), pp. 103–117. DOI: 10.1007/s002110050306.

[STH89] T. Shih, C. Tan, and B. Hwang. “Effects of grid staggering on numerical schemes”. In: *International Journal for Numerical Methods in Fluids* 9.2 (1989), pp. 193–212. DOI: https://doi.org/10.1002/fld.1650090206.

[Stu16] K. Sturm. “Convergence of Newton’s method in shape optimisation via approximate normal functions”. In: (2016). arXiv: 1608.02699.

[SV85] L. R. Scott and M. Vogelius. “Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 19.1 (1985), pp. 111–143. DOI: 10.1051/m2an/1985190101111.

[SW17] M. Siebenborn and K. Welker. “Algorithmic aspects of multigrid methods for optimization in shape spaces”. In: *SIAM Journal on Scientific Computing* 39.6 (2017), B1156–B1177. DOI: 10.1137/16M1104561.

[SW94] D. Silvester and A. J. Wathen. “Fast iterative solution of stabilised Stokes systems. Part II: Using general block preconditioners”. In: *SIAM Journal on Numerical Analysis* 31.5 (1994), pp. 1352–1367. DOI: 10.1137/0731070.

[SZ90] L. R. Scott and S. Zhang. “Finite element interpolation of nonsmooth functions satisfying boundary conditions”. In: *Mathematics of Computation* 54.190 (1990), pp. 483–493. DOI: 10.1090/S0025-5718-1990-1011446-7.

[Tem68] R. Temam. “Une méthode d’approximation de la solution des équations de Navier–Stokes”. In: *Bulletin de la Société Mathématique de France* 98.4 (1968), pp. 115–152. DOI: 10.24033/bsmf.1662.

[Tur99] S. Turek. *Efficient Solvers for Incompressible Flow Problems: an Algorithmic and Computational Approach*. Springer Berlin Heidelberg, 1999. DOI: 10.1007/978-3-642-58393-3.

[Van86] S. P. Vanka. “Block-implicit multigrid solution of Navier-Stokes equations in primitive variables”. In: *Journal of Computational Physics* 65.1 (1986), pp. 138–158. DOI: 10.1016/0021-9991(86)90008-2.

[VW92] P. S. Vassilevski and J. Wang. “Multilevel iterative methods for mixed finite element discretizations of elliptic problems”. In: *Numerische Mathematik* 63.1 (1992), pp. 503–520. DOI: 10.1007/BF01385872.

[Wat15] A. J. Wathen. “Preconditioning”. In: *Acta Numerica* 24 (2015), pp. 329–376. DOI: 10.1017/S0962492915000021.

[WE06] C.-T. Wu and H. C. Elman. “Analysis and comparison of geometric and algebraic multigrid for convection-diffusion equations”. In: *SIAM Journal on Scientific Computing* 28.6 (2006), pp. 2208–2228. DOI: 10.1137/060662940.

[Wec19a] F. Wechsung. *Fireshape/fireshape: thesis florian wechsung*. 2019. DOI: 10.5281/zenodo.3461013. URL: https://doi.org/10.5281/zenodo.3461013.

[Wec19b] F. Wechsung. *Florianwechsung/alfi: thesis submission*. 2019. DOI: 10.5281/zenodo.3369207. URL: https://doi.org/10.5281/zenodo.3369207.

[Wec19c] F. Wechsung. *Florianwechsung/thesisnumerics: thesis*. 2019. DOI: 10.5281/zenodo.3463176. URL: https://doi.org/10.5281/zenodo.3463176.

[Wes04] P. Wesseling. *An Introduction to Multigrid Methods*. Corr. reprint. Philadelphia: R.T. Edwards, 2004.

[WKG06] G. N. Wells, E. Kuhl, and K. Garikipati. "A discontinuous Galerkin method for the Cahn–Hilliard equation". In: *Journal of Computational Physics* 218.2 (2006), pp. 860–877. DOI: 10.1016/j.jcp.2006.03.010.

[WR] F. Wechsung and C. Richardson. *PyROL*. URL: https://bitbucket.org/pyrol/pyrol/.

[WY07] J. Wang and X. Ye. "New finite element methods in computational fluid dynamics by H(div) elements". In: *SIAM Journal on Numerical Analysis* 45.3 (2007), pp. 1269–1286. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/060649227.

[WZ14] J. Wu and H. Zheng. "Parallel subspace correction methods for nearly singular systems". In: *Journal of Computational and Applied Mathematics* 271 (2014), pp. 180–194. DOI: 10.1016/j.cam.2014.04.012.

[Xu01] J. Xu. "The method of subspace corrections". In: *Journal of Computational and Applied Mathematics* 128.1-2 (2001), pp. 335–362. DOI: 10.1016/S0377-0427(00)00518-5.

[Xu92] J. Xu. "Iterative methods by space decomposition and subspace correction". In: *SIAM Review* 34.4 (1992), pp. 581–613. DOI: 10.1137/1034116.

[XZ10] X. Xu and S. Zhang. "A new divergence-free interpolation operator with applications to the Darcy–Stokes–Brinkman equations". In: *SIAM Journal on Scientific Computing* 32.2 (2010), pp. 855–874. DOI: 10.1137/090751049.

[Zha04] S. Zhang. "A new family of stable mixed finite elements for the 3D Stokes equations". In: *Mathematics of Computation* 74.250 (2004), pp. 543–555. DOI: 10.1090/s0025-5718-04-01711-9.

[Zha08] S. Zhang. "On the P1 Powell-Sabin divergence-free finite element for the Stokes equations". In: *Journal of Computational Mathematics* 26.3 (2008), pp. 456–470.

[Zha11a] S. Zhang. "Divergence-free finite elements on tetrahedral grids for $k \geq 6$". In: *Mathematics of Computation* 80.274 (2011), pp. 669–669. DOI: 10.1090/S0025-5718-2010-02412-3.

[Zha11b] S. Zhang. "Quadratic divergence-free finite elements on Powell–Sabin tetrahedral grids". In: *Calcolo* 48.3 (2011), pp. 211–244. DOI: 10.1007/s10092-010-0035-4.

[Zha90] S. Zhang. "Optimal-order nonnested multigrid methods for solving finite element equations I: on quasi-uniform meshes". In: *Mathematics of Computation* 55.191 (1990), p. 23. DOI: 10.2307/2008790.