

ADAPTIVE TECHNIQUES FOR COMPUTING SHEAR STRESSES IN BIOMEDICAL FLOWS

by

ELINE SUNDT

THESIS

for the degree of

MASTER OF SCIENCE

(Master i Anvendt matematikk og mekanikk)



*Faculty of Mathematics and Natural Sciences
University of Oslo*

August 2012

*Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo*

Abstract

In this thesis, we investigate the application of duality-based goal-oriented adaptive error control to the computation of shear stresses in biomedical flow problems. As a model problem, we consider the linear Stokes equations.

Adaptive error control for goal functionals expressed as surface integrals, as in the case of shear stresses, require the formulation of a dual Stokes problem where the shear stress goal functional enters as a driving force. This may lead to instabilities (oscillations) in the dual pressure. A partial solution to this problem is to reformulate surface integrals as volume integrals.

We find that the volume formulation leads to significant improvements, both for the stability of the dual pressure and the quality of efficiency indices. Various strategies for approximation of the dual problem, mesh refinement, and representations of shear stress goal functionals are examined. The strategies have been implemented in Python based on the FEniCS/DOLFIN framework and applied to a pair of two-dimensional geometries, including a simple test case on the unit square with known primal and dual analytic solutions, and an idealized model of an aneurysm geometry.

Acknowledgements

This thesis is written as a part of a masters degree in applied mathematics at the University of Oslo, marking the end of six years of studies. There are many people I would like to acknowledge. First of all I would like to express my gratitude to my supervisors Anders Logg and Marie E. Rognes for their invaluable advice and support, and for always being available for guidance. I would also like to thank Simula Research laboratories for providing access to computing resources and for providing a good working environment at their facilities.

I want to thank my fellow students at UiO for giving me a great six years of studies that included eager black-board discussions, late study sessions, trips overseas and numerous festive excursions. A special thanks must go to Øyvind for his support these last months. I would also like to give my appreciation to my good friends of many years for their patience over the last couple of years. Last but not least I must thank my parents, Bente and Jan Eilert for all their love and support, and my sister, Anja, who inspires me always with her courage and determination.

Contents

Contents	iv
1 Introduction	1
2 Simulating fluid flow	3
2.1 Mathematical model	3
2.2 The finite element method	6
2.3 Finite element formulation for the Stokes equations	9
2.4 Finite element formulation for the stationary Navier-Stokes equations	10
3 Adaptivity and goal-oriented error control	11
3.1 A framework for adaptive error control	11
3.2 Application of the framework for adaptive control to the Stokes problem and the stationary Navier-Stokes problem	14
4 Methodology	17
4.1 Adaptivity strategies	17
4.2 Computing goal functionals	19
4.3 Performance evaluation	21
4.4 Tools	22
5 Test cases	23
5.1 Test case I: Two dimensional unit square	23
5.2 Test case II: Idealized saccular aneurysm	26
6 Numerical results	31
6.1 Test case I	31
6.2 TestCase II	53
7 Conclusion and future work	61
A Code	63
A.1 The shear stress goal functional	63

Chapter 1

Introduction

Computational fluid dynamics (CFD) is a scientific field that, with the help of digital computers, produces quantitative predictions or numerical descriptions of fluid flows. It is immensely powerful as a means to increase understanding of fluid phenomena. CFD plays a central role in engineering predictions as a tool to solve industrial problems, and has applications in all areas of engineering. It is used within aerospace engineering, chemical engineering, mechanical and civil engineering to name a few [15, 29].

CFD is never exact. To fulfill the full power of CFD, results must be accurate and computations cost effective. It is crucial that the accuracy of a computed solution may be determined. Duality based goal-oriented error control and adaptivity is a technique for minimization of the error in a quantity of interest, a so-called *goal functional*, taken from a computed solution. Most often our main interest in a simulation will be in some quantity of physical or engineering interest. With error control in a goal functional we exploit the localized nature of such a quantity to construct purposefully designed meshes, thereby allowing for an economic use of computational resources. The goal-oriented framework also provides an estimate of the accuracy of the computed functional of interest. *A posteriori* estimates are derived based on local solution residuals and the solution of an auxiliary adjoint (dual) problem. The dual problem describes the dependence of the global error in the goal functional on local smoothness properties. Goal-oriented adaptivity and error control thus addresses both the aspect of computational work and accuracy in a simulation.

Another field in which CFD has a huge potential is within modelling and imaging of physiological processes [8]. Simulation of blood flow in arteries and computation of the resulting shear stresses offer a promising tool to increase understanding of the causes for the development and rupture of what is known as aneurysms. An aneurysm is a balloon-like dilatation of a segment of the the artery that forms when the structural integrity of the artery wall is compromised due to repetitive hydraulic stress [14]. Figure 1.1 illustrates the phenomenon. As many as 1-6 per cent of the population will during their life time develop aneurysms [13, 23, 19]. Untreated the aneurysm might rupture, which may lead to death or grave disability. Even unruptured an aneurysm may cause damage by interrupting the blood flow or impinging on nearby blood vessels, organs or bone [14].

Wall shear stress is a quantity of importance in the study of blood flow, in particular in the study of flows in blood vessels with vascular malformations. It is believed to be a contributing factor in the development and rupture of aneurysms [19]. Simulations of blood are computationally expensive as they are conducted on geometrically complex domains.



Figure 1.1: Image of an abnormal blood vessel with multiple aneurysms [2].

In this thesis we investigate techniques for employing goal-oriented error control and adaptivity when the quantity of interest is shear stress. We investigate different ways of representing the shear stress and the quality of the derived error estimates and error-indicators. Method investigations do not extend to the case when the shear stresses are induced by fluid flow governed by the Navier-Stokes equations. Nonlinear extensions of the general theory is however still included, as is the application of the finite element method and the goal-oriented framework to the Navier-Stokes equations.

The outline of the thesis is as follows. In [Chapter 2](#) the governing equations for flow considered in this thesis are presented. The general theory of the finite element method along with its application to the governing equations are also presented in [Chapter 2](#). [Chapter 3](#) introduces a framework for adaptive goal-oriented error control. [Chapter 4](#) presents the methodology employed in this thesis. In [Chapter 5](#) we provide a brief introduction to the different test-cases, and the numerical results are presented in [Chapter 6](#). [Chapter 7](#) provides conclusions and future works.

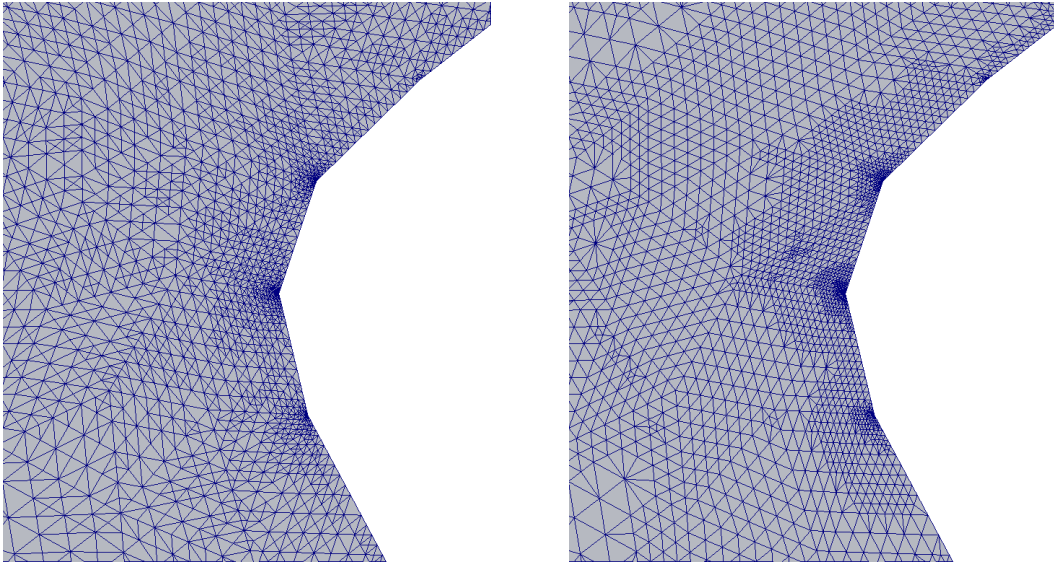


Figure 1.2: Different refinement strategies of a geometry.

Chapter 2

Simulating fluid flow

In this chapter, the governing equations considered in this work are described, together with a finite element discretization of these equations.

2.1 Mathematical model

2.1.1 The steady-state incompressible Navier-Stokes equations

The incompressible Navier-Stokes equations are a set of nonlinear partial differential equations that describe the motion of incompressible Newtonian fluids. They arise from the conservation laws for mass and momentum and a balance of forces according to Newton's laws [25, 15, 5, 20]. For a vector field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, describing the fluid velocity, and a scalar pressure field $p = p(\mathbf{x}, t)$, the full Navier-Stokes equations for viscous Newtonian incompressible fluids read

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \nabla \cdot \sigma + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \quad (2.1)$$

where ρ is the density of the fluid, $\sigma = \sigma(\mathbf{u}, p)$ is the tensor describing stresses in the fluid and \mathbf{f} is a given body force, such as gravity. The stress tensor we consider in this thesis is

$$\sigma(\mathbf{u}, p) = \mu \nabla \mathbf{u} + pI, \quad (2.2)$$

where μ is the dynamic viscosity. We have switched the sign of the pressure to obtain a symmetric matrix system, and we apply the regular velocity gradient instead of the symmetric gradient $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$, which is traditionally used in stress calculations. The equations (2.1) are called the momentum equation and the equation of continuity, respectively. We consider here the time-independent, or stationary, Navier-Stokes equations, and we assume incompressibility of the fluid, meaning a constant density in space and time. All fluids are compressible, but incompressibility is a reasonable assumption when the fluid density is virtually unaffected by pressure changes within the flow [15]. This applies for fluids such as water and oil, and is a common assumption also for blood.

The equation of continuity

The continuity equation is derived from the law of mass conservation, which states that, for an arbitrary fixed volume V , with boundary ∂V , the volume mass will remain constant over time. Thus a decrease of mass within V must equal the rate of outflow through the boundaries. This is described by the equation

$$\frac{d}{dt} \int_V \rho \, dV = - \int_{\partial V} \rho \mathbf{u} \cdot \mathbf{n} \, dS. \quad (2.3)$$

Using the divergence theorem (Green's theorem) [15], the surface integral is transformed into a volume integral, and we obtain

$$\int_V \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) dV = 0. \quad (2.4)$$

For this to hold for an arbitrarily chosen volume, it must be true at any point, and thus the integrand vanishes, leaving us with $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$. This is called the volume conservation equation or the continuity equation. Since the fluid is incompressible, the density is constant in time, and the continuity equation reduces to

$$\nabla \cdot \mathbf{u} = 0. \quad (2.5)$$

The momentum equation

The momentum of the fluid contained within the volume V is conserved. Newton's second law states that the total force acting on a body equals its rate of change of linear momentum, $\int_V \rho \mathbf{u} dV$. From this balance of forces the momentum equation is derived. The forces acting on the fluid are divided into two types: body forces are forces acting from a distance, such as gravity, whereas surface forces are forces that arise from direct contact with the surroundings, and are given by stresses on the domain walls. The stress tensor σ gives the state of stress at a point, so by taking its vector product with the unit normal on the boundary wall, we obtain the stress vector whose normal and tangential components constitute the normal- and shear stresses respectively. This relation of forces is expressed through the equation

$$\begin{aligned} \frac{d}{dt} \int_V \rho \mathbf{u} dV &= \int_{\partial V} \sigma \cdot \mathbf{n} dS + \int_V \mathbf{f} dV \\ &= \int_V \nabla \cdot \sigma + \mathbf{f} dV, \end{aligned} \quad (2.6)$$

where the last equality was obtained by applying Green's theorem to the stress-term. We apply the transport theorem [15] (2.3) to the momentum-term, perform integration by parts on the resulting surface integral and finally apply the continuity equation. We then obtain

$$\begin{aligned} \frac{d}{dt} \int_V \rho \mathbf{u} dV &= \int_V \rho \frac{d\mathbf{u}}{dt} dV + \int_{\partial V} \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dS \\ &= \int_V \rho \frac{d\mathbf{u}}{dt} + \rho (\nabla \cdot \mathbf{u}) \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u} dS \\ &= \int_V \rho \frac{d\mathbf{u}}{dt} + \rho \mathbf{u} \cdot \nabla \mathbf{u} dS. \end{aligned} \quad (2.7)$$

This is inserted into (2.6). The resulting terms can now be collected in one volume integral

$$\int_V \left(\rho \frac{d\mathbf{u}}{dt} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \sigma - \mathbf{f} \right) dV = 0. \quad (2.8)$$

Again we observe that for this to apply to any volume, the integrand must vanish at every point, which yields

$$\rho \left(\frac{d\mathbf{u}}{dt} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \nabla \cdot \sigma + \mathbf{f}. \quad (2.9)$$

This is the momentum equation, which, when put together with (2.5), yields the time-dependent incompressible Navier-Stokes equations.

2.1.2 The Stokes equations

The Stokes equations are a stationary, linearized form of the full Navier-Stokes equations, derived by eliminating the terms on the left side of (2.1), namely the term expressing the time-dependence, $\frac{\partial \mathbf{u}}{\partial t}$, and the nonlinear term, $\mathbf{u} \cdot \nabla \mathbf{u}$, also called the convective term. When convection is removed, we model a more controlled and predictable flow, whereas letting $\frac{\partial \mathbf{u}}{\partial t} = 0$, means the equations are stationary, or time-independent. This means that there is no longer any change over time, and we say that the flow has reached a steady state. The Stokes equations model slow flowing, or creeping, flow. In flow that satisfies the Stokes equations, viscous forces dominate over kinematic forces, which may correspond to a high fluid viscosity or very small velocities. Our model for Stokes flow then reads

$$-\nabla \cdot \sigma(\mathbf{u}, p) = \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \quad (2.10)$$

with $\sigma(\mathbf{u}, p)$ as in (2.2) and \mathbf{f} the body forces.

2.1.3 Boundary conditions

To complete our model problem, we must also describe the fluid behaviour on the domain boundary. When conditions are imposed directly on the fluid velocity or pressure on the boundary, it is called a Dirichlet-type boundary condition. When instead conditions are imposed on the velocity gradient normal to the boundary, it is called a Neumann-type boundary condition. These are the two types of boundary conditions we are concerned with in this thesis. A brief description is given of some common boundary conditions.

An inflow boundary condition is a prescribed fluid velocity at a domain inlet. We can for example have a uniform or a parabolic inlet velocity profile. At a domain outlet we impose a condition of no flow influence or resistance by letting the normal component of the stress tensor, σ , also called the traction vector, equal zero. That is $\sigma(\mathbf{u}, p) \cdot \mathbf{n} = \mathbf{0}$. On the remaining boundary, namely all areas that are not inlet or outlet, we model rigid walls to contain the fluid. Modelling the boundary walls as rigid is in some cases, such as blood flow, not an accurate assumption, but is fairly common nonetheless. Solid walls are modelled by letting the velocity equal zero on the boundary. The no-slip condition means that the fluid velocity relative to the wall is zero, so in the case of rigid walls, this means that the fluid velocity is zero.

2.1.4 Computation of wall shear stress

In this thesis we investigate techniques for conducting error control in the wall shear stresses induced by fluid flow. In Chapter 3, the general theory of adaptive error control is presented, and we see that the goal functional in which we seek error minimization, will appear in the variational form of an auxiliary adjoint (dual) problem derived from our original problem. This means that different goal functionals give rise to different dual solutions. We have applied, in our numerical studies, different techniques for representing the shear stress goal functional: one in which the shear stress is represented in terms of surface integrals, also called the *surface formulation*, and one in which the shear stress is reformulated to contain only integrals over the interior of the domain, a formulation we call the *volume formulation*.

The wall shear stress is a quantity that resides on the domain boundary, and is thus traditionally represented in terms of a line- or surface integral in, respectively, two and three dimensions. If \mathbf{u} is the fluid velocity and p the fluid pressure, then the *surface representation*

of the induced wall shear stress reads

$$\mathcal{M}_S((\mathbf{u}, p)) = \int_{\Gamma_G} \sigma(\mathbf{u}, p) \mathbf{n} \cdot \mathbf{t} \, dS, \quad (2.11)$$

where σ is the stress tensor given in (2.2), \mathbf{n} is the unit normal vector and \mathbf{t} the unit tangent vector on the goal boundary Γ_G . The subscript S indicates that \mathcal{M}_S gives shear stress represented as surface integral.

The technique for deriving the volume formulation is described in [27], and the continuation article [28], for the computation of drag and lift. Here we apply it to the computation of shear stress. The derivation starts with the definition of a function $\mathbf{v}_d \in [H^1(\Omega)]^d$, that is prescribed a certain value on, Γ_G , and set to be zero everywhere else on the domain boundary. The function \mathbf{v}_d should be defined so that we, by multiplying it with the momentum equation, integrating the resulting expression over the domain and performing integration by parts, obtain an expression that equals the goal functional and that consists exclusively of volume integrals. The value of \mathbf{v}_d in the domain interior can be arbitrarily chosen.

The technique is demonstrated on the stationary version of the Navier-Stokes equations given in (2.1). Application to the Stokes equations then entails simply removing the convection-term. We multiply the momentum equation with \mathbf{v}_d , integrate over Ω , and perform integration by parts:

$$\begin{aligned} \mathbf{0} &= \int_{\Omega} \rho (\mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \sigma(\mathbf{u}, p) - \mathbf{f}) \cdot \mathbf{v}_d \, dx \\ &\quad \int_{\Omega} \rho \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v}_d \, dx - \int_{\Omega} \nabla \cdot \sigma(\mathbf{u}, p) \cdot \mathbf{v}_d \, dx - \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_d \, dx \\ &= \int_{\Omega} \rho \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v}_d \, dx + \int_{\Omega} \sigma(\mathbf{u}, p) \cdot \nabla \mathbf{v}_d \, dx - \int_{\partial\Omega} \sigma(\mathbf{u}, p) \mathbf{n} \cdot \mathbf{v}_d \, dx - \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_d \, dx. \end{aligned} \quad (2.12)$$

We define \mathbf{v}_d to equal the tangent vector \mathbf{t} on the part of the boundary on which the goal functional is evaluated, and to be zero on the remaining boundary. The integral expression for the surface shear stress thus appears in (2.12). By inserting \mathbf{v}_d and collecting the volume terms on one side and the boundary terms on the other, we have derived the volume representation of the shear stress.

$$\mathcal{M}_V((\mathbf{u}, p)) = \int_{\Omega} \sigma(\mathbf{u}, p) \nabla \mathbf{v}_d \, dx - \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_d \, dx. \quad (2.13)$$

Since the value of \mathbf{v}_d in the interior can be arbitrarily chosen, we will in our numerical testing apply one volume representation in which \mathbf{v}_d has minimal support in Ω , and one in which it has a larger support, to investigate the effect this has on the corresponding dual solution and generated error estimates.

2.2 The finite element method

Partial differential equations (PDE's) are the underlying equations in a wide range of physical phenomena. Most problems involving partial differential equations can not be solved

analytically. We need to use a numerical technique to discretize the problem and generate an approximate solution. The numerical technique we use in this thesis is the finite element method. It is a powerful and flexible numerical approach in that it with relative ease allows for discretization of complicated domains, application of non-uniform meshes and appropriate boundary conditions. It is the most widely used numerical technique both in analysis and engineering [11, 9].

2.2.1 The strong problem formulation

When solving a PDE, we search for a continuous solution that satisfies the equations as well as some additional conditions that characterize the behaviour on the domain boundaries and the problems initial state. For a general PDE this can be expressed as:

$$\boxed{\text{Find } u \in V \quad \text{such that} \quad \mathcal{L}u = f} \quad (2.14)$$

where $\mathcal{L} : V \rightarrow V'$ is a differential operator from the solution space V into the dual space V' , $f \in V'$, and $u \in V$ is the continuous solution. This initial problem formulation is called the *strong problem* or *strong formulation*, and we say that a solution u of (2.14) solves the problem in a strong way.

2.2.2 The weak problem formulation

Given such a formulation, the finite element method is applied to generate a corresponding approximate solution. The first step is deriving an alternative formulation of the strong form that allows for the equation to be fulfilled with less restrictive differentiability requirements on u , for this reason called the *weak formulation* or the *weak problem*. It is derived by multiplying the strong form with a suitably defined test function v , integrating the resulting equations over the domain Ω , and using integration by parts to remove higher order derivatives of u . In the linear case, it can be phrased as:

$$\boxed{\text{find } u \in V \quad \text{such that} \quad a(u, v) = L(v) \quad \forall v \in \hat{V}} \quad (2.15)$$

where $a : V \times \hat{V} \rightarrow \mathbb{R}$ is a bilinear form, $L : \hat{V} \rightarrow \mathbb{R}$ a linear form and V and \hat{V} are called the trial- and test spaces, respectively. They are spaces of functions with such generalized derivatives, such as Sobolev spaces. For a description of Sobolev spaces and their properties, see [11]. We will often refer to (2.15) as the *variational form*.

2.2.3 The finite element formulation

To introduce the discrete setting, we construct finite dimensional subspaces, $V_h \subset V$, in which to search for the approximate solution. To this end we first divide the domain Ω into a set, $\mathcal{T}_h = \{T\}$, of subdomains with disjoint interiors, also called a tessellation, such that $\Omega = \cup_{T \in \mathcal{T}_h} T$, which we call the cells. We then define local function spaces $\mathcal{P} = \mathcal{P}(T)$ over each cell as well as a way to determine a function $v \in \mathcal{P}$ from the basis for its dual space, which consists of linear functionals on \mathcal{P} . The definition of a finite element, Definition 2.2.1, found in [9] formally describes this.

Definition 2.2.1 (Finite element) *Let*

- (i) T be a bounded closed subset of \mathbb{R}^d with non-empty interior and piecewise smooth boundary (the element domain)
- (ii) $\mathcal{P} = \mathcal{P}(T)$ be a finite-dimensional space of functions on T of dimension n (the space of shape functions)
- (iii) $\mathcal{N} = \{N_1, N_2, \dots, N_k\}$ be a basis for \mathcal{P} (the set of degrees of freedom or nodes)

Then $(T, \mathcal{P}, \mathcal{N})$ is called a **finite element**.

Definition 2.2.2 (Nodal basis of \mathcal{P}) *Let $(T, \mathcal{P}, \mathcal{N})$ be a finite element. The basis $\{\phi_1, \phi_2, \dots, \phi_k\}$ of \mathcal{P} dual to \mathcal{N} (i.e. $N_i(\phi_j) = \delta_{ij}$) is called the **nodal basis** of \mathcal{P} .*

The global finite element space V_h over the entire domain Ω can now be constructed by combining the local function spaces given by the finite elements. By replacing the infinite-dimensional test- and trial spaces with their finite dimensional counterparts, we obtain the restricted variational problem:

$$\boxed{\text{find } u_h \in V_h \quad \text{such that} \quad a(u_h, v_h) = L(v_h) \quad \forall v_h \in \hat{V}_h \subset \hat{V}} \quad (2.16)$$

where u_h is the finite element solution. A function in V_h can be written as a linear combination of the basis for V_h .

$$u_h = \sum_{i=1}^k U_i \phi_i \quad (2.17)$$

where U_i are given by the degrees of freedom. Inserting this ansatz into (2.16), and replacing the test function with an arbitrary basis function, ϕ_j , we get a system of equations where U_i are the unknowns, that we solve to find the finite element solution.

2.2.4 Lagrange elements

There are a wide variety of element choices. We use the most common element, the Lagrange element, in which the spatial cell is a simplex, which is a generalization of a triangle; an interval in one dimension, a triangle in two dimensions and a tetrahedron in three dimensions. The functions over the cells are polynomials, and the degrees of freedom are given simply by function evaluation in the nodes, so a function $v \in \mathcal{P}$ is determined by $N(v) = v(x_i)$. The definition of the Lagrange element, Definition 2.2.3, is found in [19]. The Lagrange element is illustrated in Figure 2.1.

Definition 2.2.3 (Lagrange element) *The Lagrange element (CG_q) is defined for $q = 1, 2, 3$ by*

- T a simplex
- $\mathcal{P} = \mathcal{P}_k(T)$ space of polynomials of degree less than or equal to k
- $N_i(v) = v(x_i)$ function value in the node with global index i

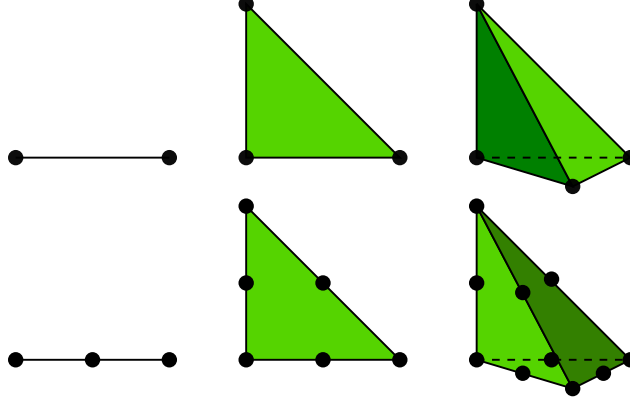


Figure 2.1: The linear (top) and quadratic (bottom) Lagrange elements in one-, two-, and three dimensions. Figures taken from [19].

2.3 Finite element formulation for the Stokes equations

We apply the finite element method to the Stokes equations for a general domain Ω with boundary $\partial\Omega$ partitioned into a Dirichlet and Neumann part, Γ_D and Γ_N , on which we impose Dirichlet and Neumann conditions, respectively. To derive the weak formulation for the Stokes problem, we multiply the strong equations, given in (2.1.2), with the appropriate test functions, integrate over Ω and apply integration by parts. The weak problem for the Stokes equations now reads

$$\begin{aligned}
 &\text{find } (\mathbf{u}, p) \in V \times Q \quad \text{such that} \\
 &\quad a((\mathbf{u}, p), (\mathbf{v}, q)) = L((\mathbf{v}, q)) \quad \forall (\mathbf{v}, q) \in \hat{V} \times \hat{Q} \\
 &\text{where} \\
 &\quad a((\mathbf{u}, p), (\mathbf{v}, q)) = \langle \nabla \mathbf{u}, \nabla \mathbf{v} \rangle + \langle p, \nabla \cdot \mathbf{v} \rangle + \langle \nabla \cdot \mathbf{u}, q \rangle \\
 &\quad L((\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{g}, \mathbf{v} \rangle_{\Gamma_N}
 \end{aligned} \tag{2.18}$$

To find the finite element formulation for the Stokes equations, we apply a mixed finite element method, which refers to the fact that we have different approximation spaces for the different unknowns. In order to obtain a stable system of algebraic equations, the finite element spaces for the velocity and pressure should be chosen so as to satisfy the Babuska-Brezzi condition, derived in [10]. For the Stokes equations, it is known that using the Taylor-Hood element space, defined in [26], assures this. The Taylor-Hood element space is the tensor product of the vector-valued globally continuous space of piecewise quadratics $([CG_2]^d)$ for the velocity, for spatial dimension d , and the scalar globally continuous space of piecewise linears (CG_1) for the pressure. The corresponding continuous solution spaces from which the discrete spaces are induced are the Sobolev spaces $[H^1(\Omega)]^2$ and $L^2(\Omega)$, respectively. The definitions of some of the applied function spaces are

$$\begin{aligned}
 H^1(\Omega) &= \{v \in \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} |v|^2 + |\nabla v|^2 dx \leq \infty\}, \\
 \mathbf{H}^1(\Omega) &= [H^1(\Omega)]^d = \{v = \{v_1, \dots, v_d\} \mid v_i \in H^1(\Omega) \ \forall i = 1, \dots, d\}, \\
 L^2(\Omega) &= \{v \in \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} |v|^2 dx \leq \infty\}. \\
 CG_k &= \{v \in H^1(\Omega) \mid v \in \mathbb{P}_k(T) \ \forall T \in \mathcal{T}_h\}. \\
 \mathbf{CG}_k &= [CG_k]^d = \{v = \{v_1, \dots, v_d\} \mid v_i \in CG_k \ \forall i = 1, \dots, d\},
 \end{aligned} \tag{2.19}$$

where \mathcal{T}_h is a tessellation of cells. The \mathbf{CG}_k spaces are globally continuous, meaning that neighbouring elements coincide in degrees of freedom on common edges. Elements in the corresponding discontinuous Galerkin (\mathbf{DG}_k) spaces do not coincide across common edges. This is illustrated in Figure 2.2. Replacing the continuous solution of (2.18) with $(\mathbf{u}_h, p_h) \in \hat{V}_h \times \hat{Q}_h = \mathbf{CG}_2 \times \mathbf{CG}_1$, we obtain the finite element formulation for the Stokes equations

$$\begin{aligned} &\text{find } (\mathbf{u}_h, p_h) \in V_h \times Q_h \quad \text{such that} \\ &\quad a((\mathbf{u}_h, p_h), (\mathbf{v}, q)) = L((\mathbf{v}, q)) \quad \forall (\mathbf{v}, q) \in \hat{V}_h \times \hat{Q}_h \\ &\text{where} \\ &\quad a((\mathbf{u}_h, p_h), (\mathbf{v}, q)) = \langle \nabla \mathbf{u}_h, \nabla \mathbf{v} \rangle + \langle p_h, \nabla \cdot \mathbf{v} \rangle + \langle \nabla \cdot \mathbf{u}_h, q \rangle \\ &\quad L((\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{g}, \mathbf{v} \rangle_{\Gamma_N} \end{aligned} \tag{2.20}$$

2.4 Finite element formulation for the stationary Navier-Stokes equations

For nonlinear partial differential equations, such as the Navier-Stokes equations, the bilinear form is replaced with a semilinear form, $F(u; v) = a(u, v) - L(v)$, which is nonlinear in all arguments preceding the semi-colon and linear in the remaining arguments. The weak form is derived in the same manner, by multiplication with appropriate test functions, integration over the domain and application of integration by parts. The finite element formulation for the stationary Navier-Stokes equations is the same as for the Stokes equations, only with an added convection-term. The same element space, the Taylor-Hood element space, is used. For the Navier-Stokes equations the weak formulation reads

$$\begin{aligned} &\text{find } (\mathbf{u}, p) \in V \times Q \quad \text{such that} \\ &\quad F((\mathbf{u}, p); (\mathbf{v}, q)) = 0 \quad \forall (\mathbf{v}, q) \in \hat{V} \times \hat{Q} \\ &\text{where} \\ &\quad F((\mathbf{u}, p); (\mathbf{v}, q)) = \langle \mathbf{u} \nabla \mathbf{u}, \mathbf{v} \rangle + \langle \nabla \mathbf{u}, \nabla \mathbf{v} \rangle + \langle p, \nabla \cdot \mathbf{v} \rangle + \langle \nabla \cdot \mathbf{u}, q \rangle \\ &\quad - \langle \mathbf{f}, \mathbf{v} \rangle - \langle \mathbf{g}, \mathbf{v} \rangle_{\Gamma_N} \end{aligned} \tag{2.21}$$

where $F : V \times \hat{V} \rightarrow \mathbb{R}$. The finite element formulation is entirely analogous with (2.21) with V and Q as finite element spaces.

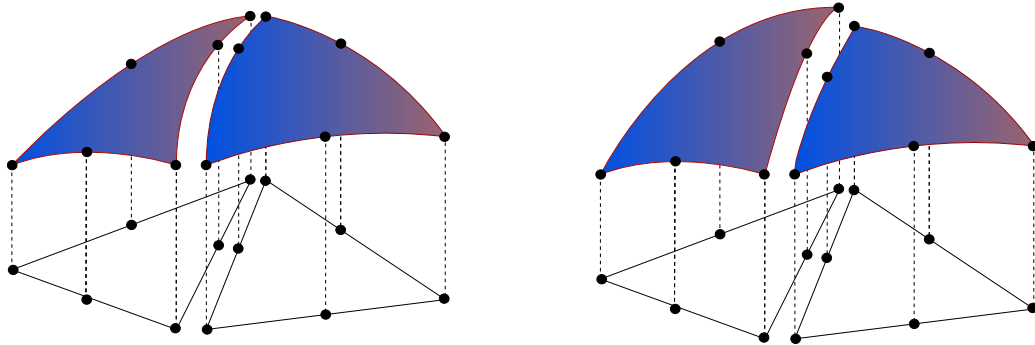


Figure 2.2: Piecewise quadratic functions of globally continuous (left), and globally discontinuous (right) function spaces. Figures taken from [19].

Chapter 3

Adaptivity and goal-oriented error control

In all numerical modelling, we target reliable control of the error in an approximation with optimal use of computational resources. Error estimates reflect the accuracy of the computation and stability of the problem, and can usually be divided into two types; *a priori*- and *a posteriori* error estimates. *A priori* error estimates relate the computational error in an approximation to information about the global continuous solution. To quantify the error in practice, *a priori* error estimates are less suited, since information about the continuous problem is not usually available. We must rather apply *a posteriori* error estimates, where the computational error is related to information about the global discrete solution. Dealing with error estimation in global norms like the energy norm or L^2 -norm does, in most applications, not provide useful bounds for quantities of real physical interest. They also involve global stability constants that are usually unknown, and that are too crude in the sense that they do not properly reflect the sensitivity to local error sources.

Goal-oriented adaptivity and error control applies a duality technique to derive error estimates directly for a quantity of interest taken from the computed solution. These estimates are based on local error indicators, in which local residuals are weighted with the solution of a linearized adjoint (dual) problem. The dual problem expresses the influence of the local residuals on the global error in the quantity of interest. The local error indicators can be utilized to adapt mesh refinement with respect to minimization of the goal functional error. Goal-oriented error estimation thus aims to optimize the use of computational resources needed to control of the accuracy in a given physical quantity of interest. The following is a summary of the general theory of goal oriented adaptivity and error control gathered from [7, 6, 12].

3.1 A framework for adaptive error control

In many applications, our main interest will be some localized physical property of the solution of a simulation. We define a goal functional $\mathcal{M} : V \rightarrow \mathbb{R}$, which expresses this quantity. Since we typically target a high accuracy in a localized quantity of interest, a mesh of high global resolution may be unnecessarily expensive. When we employ goal-oriented adaptivity, refinement can be focused to the areas that give the largest contributions to the total error in \mathcal{M} . We thus aim at obtaining an accurate simulation with a purposeful use of computational resources. The task of goal-oriented adaptive error control is: a minimal computational work to obtain a given level of accuracy in \mathcal{M} , or conversely, minimizing the error in \mathcal{M} , with a

given amount of computational work.

An adaptive method consists of a method for discretization and an adaptive algorithm. This involves an iterative process in which a problem is solved following the prescribed discretization procedure, a stopping-criterion is evaluated using computational feedback, and a strategy of modification is carried out if the criterion is not satisfied. When the stopping-criterion is met, the iterative process terminates and we have obtained a solution guaranteed to fulfil the desired level of accuracy.

The residual error, that is the error generated by inserting the approximated solution into the differential equation, provides a measure of the accuracy, whereas the solution to an auxiliary dual problem gives the stability properties. This duality technique is described in [12] for adaptive methods with estimates on the global solution error. In [7, 6] the technique is extended to goal oriented error control. The main steps involved in deriving *a posteriori* error estimates and error indicators are described below.

3.1.1 The linear case

We first consider the case of linear variational problems and linear goal functionals. In the following section, the theory is extended to include nonlinear variational problems and nonlinear goal functionals. We consider a general linear variational problem

$$\text{find } u \in V \quad \text{such that} \quad a(u, v) = L(v) \quad \forall v \in \hat{V}, \quad (3.1)$$

with $a : V \times \hat{V} \rightarrow \mathbb{R}$ a continuous bilinear form, $L : \hat{V} \rightarrow \mathbb{R}$ a continuous linear form, and V and \hat{V} appropriate Sobolev spaces. In the context of adaptive error control this is called the primal problem, and u the primal solution. The finite element formulation, posed on finite dimensional subspaces $V_h \in V$ and $\hat{V}_h \in \hat{V}$, constructed as described in Section 2.2.3, then reads

$$\text{find } u_h \in V_h \quad \text{such that} \quad a(u_h, v_h) = L(v_h) \quad \forall v_h \in \hat{V}_h. \quad (3.2)$$

This is called the discrete primal problem, and its solution u_h , the discrete primal solution. We assume that a unique discrete solution exists. We will most often refer to (3.2) as the primal problem, rather than the discrete primal. The (weak) residual relative to the approximation u_h , measures the error of inserting the approximate solution, u_h , in the continuous primal problem. The weak residual is defined as

$$r(v) = L(v) - a(u_h, v). \quad (3.3)$$

By Galerkin orthogonality, we have

$$r(v) = L(v) - a(u_h, v) = a(u_h, v) - a(u_h, v) = 0 \quad \forall v \in \hat{V}_h, \quad (3.4)$$

meaning that the residual vanishes for all functions in \hat{V}_h . This property is used in the derivation of the error estimates.

The goal of the computation is to minimize the error in the goal functional with respect to u_h , that is; for a given tolerance TOL, we wish to compute a discrete primal solution, u_h , that solves (3.2) such that

$$\eta = |\mathcal{M}(u) - \mathcal{M}(u_h)| \leq \text{TOL}. \quad (3.5)$$

To derive an *a posteriori* error estimate for the magnitude of $\mathcal{M}(u) - \mathcal{M}(u_h)$, we must first find a way of representing it that involves only known quantities. To this end we define the continuous dual variational problem

$$\text{find } z \in V^* \quad \text{such that} \quad a^*(z, v) = \mathcal{M}(v) \quad \forall v \in \hat{V}^*, \quad (3.6)$$

where V^* and \hat{V}^* are the dual trial- and test spaces, $a^* : V^* \times \hat{V}^* \rightarrow \mathbb{R}$ is the adjoint of a , so that $a^*(u, v) = a(v, u)$. We call z the dual solution. The discrete dual problem is the natural discretization of (3.7), posed on the finite dimensional subspaces $V_h^* \subset V^*$ and $\hat{V}_h^* \subset \hat{V}^*$. The discrete dual reads

$$\text{find } z_h \in V_h^* \quad \text{such that} \quad a^*(z_h, v) = \mathcal{M}(v) \quad \forall v \in \hat{V}_h^*, \quad (3.7)$$

Now we have what we need to find a representation of $\mathcal{M}(u) - \mathcal{M}(u_h)$ in terms of computable functions. We assume that we have $V^* = \hat{V}$ and $\hat{V}^* = \{u - v | u, v \in V\}$. Using linearity of \mathcal{M} and combining (3.1), (3.7) (3.3), we can find a different representation of the error in \mathcal{M} .

$$\begin{aligned} \mathcal{M}(u) - \mathcal{M}(u_h) &= a^*(z, u - u_h) = a(u - u_h, z) \\ &= a(u, z) - a(u_h, z) \\ &= L(z) - a(u_h, z) = r(z). \end{aligned} \quad (3.8)$$

Note that by galerkin orthogonality (3.4), we also have that

$$\mathcal{M}(u) - \mathcal{M}(u_h) = r(z) = r(z - v_h) \quad \forall v_h \in \hat{V}_h. \quad (3.9)$$

3.1.2 Extension to the nonlinear case

We now extend the theory to include nonlinear variational problems and nonlinear goal functionals. The general nonlinear variational problem reads

$$\text{find } u \in V \text{ such that} \quad F(u; v) = 0, \quad \forall v \in \hat{V} \quad (3.10)$$

where V and \hat{V} are a pair of trial and test spaces and $F : V \times \hat{V} \rightarrow \mathbb{R}$ is a semilinear form which is (possibly) nonlinear in arguments preceding the semicolon and linear in remaining arguments. Some definitions are in order:

- The Fréchet derivative:

$$F'(u; \partial u, v) = \frac{\partial}{\partial \epsilon} F(u + \epsilon \partial u; v) |_{\epsilon=0}$$
- The adjoint:

$$F'^*(u; \partial u, v) = F'(u; v, \partial u)$$
- An appropriate average of the Fréchet derivative:

$$\overline{F'}(u, v) = \int_0^1 F'(su + (1-s)u_h; u, v) ds$$

We define the dual problem:

$$\text{find } z \in V^* \text{ such that} \quad \overline{F'}^*(z, v) = \overline{\mathcal{M}'}(v) \quad \forall v \in \hat{V}^* \quad (3.11)$$

$\overline{F'}^*$ is now a bilinear form. We want to use this to find an error representation just like in the linear case. To this end we first apply the chain rule on the average of the Fréchet to obtain the following equality.

$$\begin{aligned}
\overline{F'}(u - u_h, v) &= \int_0^1 F'(su + (1-s)u_h; u - u_h, v) \, ds \\
&= \int_0^1 \frac{\partial}{\partial \epsilon} F(su + (1-s)u_h + \epsilon(u - u_h); v)|_{\epsilon=0} \, ds \\
&= \int_0^1 \frac{\partial}{\partial \epsilon} F(g(\epsilon); v)|_{\epsilon=0} \, ds \\
&\quad (\text{where } g(\epsilon) = su + (1-s)u_h + \epsilon(u - u_h)) \\
&= \int_0^1 \frac{\partial}{\partial g} F(su + (1-s)u_h; v) \frac{\partial g}{\partial \epsilon} \, ds \\
&= (u - u_h) \int_0^1 \frac{\partial}{\partial s} \left(\frac{\partial g}{\partial s} \right)^{-1} F(su + (1-s)u_h; v) \, ds \\
&= (u - u_h)(u - u_h)^{-1} (F(u; v) - F(u_h; v)) \\
&= F(u; v) - F(u_h; v).
\end{aligned} \tag{3.12}$$

This can be shown identically for $\overline{\mathcal{M}'}$, so we have

$$\begin{aligned}
\overline{F'}(u - u_h; v) &= F(u; v) - F(u_h; v), \\
\overline{\mathcal{M}'}(u - u_h) &= \mathcal{M}(u) - \mathcal{M}(u_h).
\end{aligned} \tag{3.13}$$

Using these derived equalities together with (3.10), we can find a representation of the error in the goal functional.

$$\begin{aligned}
\mathcal{M}(u) - \mathcal{M}(u_h) &= \overline{\mathcal{M}'}(u - u_h) = \overline{F'}^*(z, u - u_h) \\
&= \overline{F'}(u - u_h, z) = F(u; z) - F(u_h; z) \\
&= -F(u_h; z) = r(z)
\end{aligned} \tag{3.14}$$

The error in the goal functional is now represented in terms of the weak residual evaluated in the dual solution as for the linear case.

3.2 Application of the framework for adaptive control to the Stokes problem and the stationary Navier-Stokes problem

We derive the dual problem and error representation including redistribution of facet residuals based on the strong and weak formulation for the Stokes equations. They are derived for a general domain Ω , with boundary $\partial\Omega$ divided into a Dirichlet part Γ_D , which may include a homogeneous and non-homogeneous part, and Neumann part Γ_N , a general body force \mathbf{f} and a general goal functional $\mathcal{M}(u, p)$.

3.2.1 Dual problem

When solving the dual problem for the Stokes equations we look for the dual solution $\mathbf{z} = (z_V, z_Q) \in W^*$ such that

$$a^*(\mathbf{z}, \mathbf{w}) = \mathcal{M}(\mathbf{w}),$$

for all $\mathbf{w} = (\mathbf{w}_V, w_Q) \in \hat{W}^*$. Where

$$a^*(\mathbf{z}, \mathbf{w}) = \langle \mu \nabla \mathbf{w}_V, \nabla \mathbf{z}_V \rangle + \langle w_Q, \nabla \cdot \mathbf{z}_V \rangle + \langle \nabla \cdot \mathbf{w}_V, z_Q \rangle.$$

The goal functional \mathcal{M} will vary with the application.

3.2.2 Error representation

Based on the general weak form we find the representation of the error $\mathcal{M}((\mathbf{u}_h, p_h)) - \mathcal{M}((\mathbf{u}, p))$ for the Stokes equations using the weak residual r evaluated at the dual solution \mathbf{z} as described in (3.8). $\mathcal{T}_h = \{T_i\}$ denotes a tessellation of cells with $\Omega = \cup_i T_i$ and $T_i \cap T_j = \emptyset \quad \forall i, j$. We use the definition of the weak residual and divide the residual into a sum of cell contributions. We then apply integration by parts on the cell-terms to obtain a distribution of the cell residuals into interior- and facet contributions, denoted, respectively R_T and $R_{\partial T}$. We denote by v^T the restriction of v to the cell T , and $\sigma_h = \mu \nabla \mathbf{u}_h + p_h$ the discrete shear stress.

$$\begin{aligned}
r((\mathbf{z}_V, z_Q)) &= L((\mathbf{z}_V, z_Q)) - a((\mathbf{u}_h, p_h), (\mathbf{z}_V, z_Q)) \\
&= \langle f, \mathbf{z}_V \rangle + \langle g, \mathbf{z}_V \rangle_{\Gamma_N} - \langle \mu \nabla \mathbf{u}_h, \nabla \mathbf{z}_V \rangle - \langle p_h, \nabla \cdot \mathbf{z}_V \rangle - \langle \nabla \cdot \mathbf{u}_h, z_Q \rangle \\
&= \sum_{T \in \mathcal{T}_h} \langle \mathbf{f}, \mathbf{z}_V \rangle_T + \langle \mathbf{g}, \mathbf{z}_V \rangle_{\partial T \cap \Gamma_N} - \langle \mu \nabla \mathbf{u}_h, \nabla \mathbf{z}_V \rangle_T - \langle p_h, \nabla \cdot \mathbf{z}_V \rangle_T - \langle \nabla \cdot \mathbf{u}_h, z_Q \rangle_T \\
&= \sum_{T \in \mathcal{T}_h} \langle \mathbf{f} + \nabla \cdot (\mu \nabla \mathbf{u}_h + p_h I), \mathbf{z}_V \rangle_T - \langle \nabla \cdot \mathbf{u}_h, z_Q \rangle_T \\
&\quad - \langle (\mu \nabla \mathbf{u}_h + p_h I) \mathbf{n}, \mathbf{z}_V \rangle_{\partial T} + \langle \mathbf{g}, \mathbf{z}_V \rangle_{\partial T \cap \Gamma_N} \\
&= \sum_{T \in \mathcal{T}_h} \langle \mathbf{f}^T + \nabla \cdot \sigma_h^T, \mathbf{z}_V^T \rangle_T - \langle \nabla \cdot \mathbf{u}_h^T, z_Q^T \rangle_T - \langle (\sigma_h^T \mathbf{n}, \mathbf{z}_V^T)_{\partial T} + \langle \mathbf{g}^T, \mathbf{z}_V^T \rangle_{\partial T \cap \Gamma_N} \\
&= \sum_{T \in \mathcal{T}_h} \langle R_V^T, \mathbf{z}_V^T \rangle_T + \langle R_Q^T, z_Q^T \rangle_T - \langle \sigma_h^T \mathbf{n}^T, \mathbf{z}_V^T \rangle_{\partial T} + \langle \mathbf{g}^T, \mathbf{z}_V^T \rangle_{\partial T \cap \Gamma_N} \\
&= \sum_{T \in \mathcal{T}_h} \langle R_T, \mathbf{z}^T \rangle_T - \sum_{T \in \mathcal{T}_h} \langle \sigma_h^T \mathbf{n}^T, \mathbf{z}_V^T \rangle_{\partial T} + \langle \mathbf{g}^T, \mathbf{z}_V^T \rangle_{\partial T \cap \Gamma_N} \\
&= \sum_{T \in \mathcal{T}_h} \langle R_T, \mathbf{z} \rangle_T + \langle R_{\partial T}, \mathbf{z} \rangle_{\partial T}
\end{aligned} \tag{3.15}$$

The error in the goal functional is now represented as a sum of residual contributions from each cell T , one part of which comes from the cell interior and one from the cell facets. So, for a general set of boundary conditions

$$\begin{aligned}
\mathbf{u} &= \mathbf{g} \quad \text{on } \Gamma_D, \\
\sigma(\mathbf{u}, p) \mathbf{n} &= \mathbf{h} \quad \text{on } \Gamma_N,
\end{aligned}$$

on a general domain boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$ where Γ_D and Γ_N are the Dirichlet and Neumann boundary respectively, and for general functions \mathbf{g} and \mathbf{h} , we thus have the formulations of the cell and facet residual contributions for the Stokes equations

$$R_T = \begin{pmatrix} R_V^T \\ R_Q^T \end{pmatrix} = \begin{pmatrix} \mathbf{f}^T + \nabla \cdot (\mu \nabla \mathbf{u}_h^T + p_h^T I) \\ -\nabla \cdot \mathbf{u}_h^T \end{pmatrix}, \tag{3.16}$$

$$R_{\partial T} = \begin{pmatrix} R_{\partial T}^V \\ 0 \end{pmatrix}$$

$$\text{where} \tag{3.17}$$

$$R_{\partial T}^V = \begin{cases} \mathbf{g}^T - \sigma_h^T \mathbf{n}^T & \text{on } \partial T \cap \Gamma_N \\ -\sigma_h^T \mathbf{n}^T & \text{on } \partial T \cap \Gamma_D \end{cases}.$$

Extension to the Navier-Stokes equations yields an added term to the cell residuals given by the convection.

The facet residual on interior facets will contribute to both cells that share the facet. We distribute these interior facet residuals equally between the two cells, and this way obtain indicators that will be either equal, but in general smaller in magnitude than the original formulation. To perform this distribution we return to the residual representation found in (3.15), and consider only the facet contributions, that is the last term, $\langle R_{\partial T}, \mathbf{z} \rangle_{\partial T}$. We switch from cell-wise summation to summation over all mesh edges, perform the distribution, and then switch back to obtain a different representation consisting of contributions from each cell. These representations of the total error are equal, and will sum up to the same total error as a result of cancellations. However, when the localized cell residuals are to be used as indicators, the absolute value is applied, and the redistributed cell representations will give smaller values.

We first note that $\langle R_{\partial T}, \mathbf{z} \rangle_{\partial T} = \sum_{e \in \partial T} \langle R_{\partial T, e}, \mathbf{z} \rangle_e$, where $R_{\partial T, e} = R_{\partial T}|_e$, the restriction to edge e . We use continuous Lagrange function spaces, so the functions are piecewise polynomials which are continuous across boundaries. This is illustrated in Figure 2.2. The function value of the polynomials in neighbouring cells T and T' then coincide on the shared boundary, and here $\mathbf{z}^{T'} = \mathbf{z}^T$. We consider the sum of facet terms.

$$\begin{aligned}
\sum_{T \in \mathcal{T}_h} \langle R_{\partial T}, \mathbf{z} \rangle_{\partial T} &= \sum_{T \in \mathcal{T}_h} \sum_{e \in \partial T} \langle R_{\partial T, e}, \mathbf{z} \rangle_e \\
&= \sum_{\substack{e \in e_{int} \\ e = T \cap T'}} \langle R_{\partial T, e}, \mathbf{z}^T \rangle_e + \langle R_{\partial T', e}, \mathbf{z}^{T'} \rangle_e + \sum_{\substack{e \in e_{ext} \\ e \in \partial T}} \langle R_{\partial T, e}, \mathbf{z}^T \rangle_e \\
&= \sum_{\substack{e \in e_{int} \\ e = T \cap T'}} \langle R_{\partial T, e} + R_{\partial T', e}, \mathbf{z}^T \rangle_e + \sum_{\substack{e \in e_{ext} \\ e \in \partial T}} \langle R_{\partial T, e}, \mathbf{z} \rangle_e \\
&= \sum_{T \in \mathcal{T}_h} \left(\sum_{e \in \partial T \cap \Omega} \langle R_{\partial T, e} + R_{\partial T', e}, \mathbf{z}^T \rangle_e + \sum_{e \in \partial T \cap \partial \Omega} \langle R_{\partial T, e}, \mathbf{z} \rangle_e \right) \\
&= \sum_{T \in \mathcal{T}_h} \sum_{e \in \partial T} \langle [R_{\partial T}]|_e, \mathbf{z} \rangle_e \\
&= \sum_{T \in \mathcal{T}_h} \langle [R_{\partial T}], \mathbf{z}^T \rangle_{\partial T},
\end{aligned} \tag{3.18}$$

where $[\]$ denotes the redistribution over cell facets. We thus have

$$[R_{\partial T}] = \begin{pmatrix} [R_{\partial T}^V] \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\sigma_h^T \mathbf{n}^T + \sigma_h^{T'} \mathbf{n}^{T'}) \\ 0 \end{pmatrix}, \tag{3.19}$$

where the neighbour cell T' is found relative to cell T and edge e

$$T' = \begin{cases} T'(T, e) & \text{if } e \subset e_{int} \\ T & \text{if } e \subset e_{ext} \end{cases}$$

where e_{int} and e_{ext} denote the set of interior and exterior edges, respectively. We have derived two equal representations of the error in the goal functional

$$r(\mathbf{z}) = \sum_{T \in \mathcal{T}_h} \langle R_T, \mathbf{z} \rangle_T + \langle R_{\partial T}, \mathbf{z} \rangle_{\partial T} = \sum_{T \in \mathcal{T}_h} \langle R_T, \mathbf{z} \rangle_T + \langle [R_{\partial T}], \mathbf{z} \rangle_{\partial T}. \tag{3.20}$$

When we apply the absolute value, these representations will, however, in general not be equal. This gives rise to the definition of different error indicators, which we denote η_T .

Chapter 4

Methodology

This chapter describes the methodology used to apply the reviewed theory in the numerical testing.

4.1 Adaptivity strategies

An adaptive method involves a stopping-criterion which guarantees an error below a certain tolerance-level, and a strategy for modification if the criterion is not met. When applying adaptive error-control in a goal functional, the modification procedure is concerned with finding an economic mesh design, with the aim to reduce the error in the goal functional. This entails making choices on several sub-strategies that together make-up the modification procedure. They provide instructions for how to estimate the error in the target quantity, how to create the local error indicators, how to determine which cells are to refined and how to refine them. This section provides an introduction to the sub-strategies that we consider in this thesis. The adaptive algorithm for goal oriented error control and adaptivity is given in [Algorithm 1](#). Uniform refinement is implemented by a simple adjustment in the refinement step by simply letting all the cells be refined.

Algorithm 1 Goal oriented error control and adaptivity.

Given an initial mesh partitioning \mathcal{T}_0 , a goal functional \mathcal{M} , an error tolerance, ε .

- 1: Compute the primal solution u_h
 - 2: Estimate the error in the goal functional, $\eta_h \approx |M(u) - M(u_h)|$
 - 3: **if** $\eta_h < \varepsilon$ **then**
 Done ▷ error is computed within given tolerance
 - 4: Compute error-indicators $\{\eta_T\}$
 - 5: Mark mesh for refinement, based on $\{\eta_T\}$
 - 6: Refine mesh, $\mathcal{T}_k \rightarrow \mathcal{T}_{k+1}$
 - 7: Repeat procedure ▷ (points [1](#) - [6](#))
-

4.1.1 Error estimators and error indicators

Given a computed solution, cell error indicators are formed from computed local residuals and the approximated solution to the linearized dual problem. The error indicators indicate

which cells give the largest contributions to the global error in the target quantity. There are three strategies on how to form the error indicator on a cell T .

$$\begin{aligned}\eta_T^{err} &= |\langle R_T, z \rangle_T + \langle R_{\partial T}, z \rangle_{\partial T}|, \\ \eta_T^{dwr} &= |\langle R_T, z \rangle_T + \langle [R_{\partial T}], z \rangle_{\partial T}|, \\ \eta_T^{cfs} &= |\langle R_T, z \rangle_T| + |\langle [R_{\partial T}], z \rangle_{\partial T}|.\end{aligned}\tag{4.1}$$

where $[\]$ denotes the redistribution of the facet residuals as introduced in [Section 3.2](#). Given a computed solution, we wish to estimate the error in a quantity of interest $\eta_h \approx |\mathcal{M}(u) - \mathcal{M}(u_h)|$. The strategies for estimating the error in the goal functional considered here are

$$\begin{aligned}\eta_h^{err} &= r(w), \\ \eta_h^{dwr} &= \sum_T \eta_T^{dwr}, \\ \eta_h^{cfs} &= \sum_T \eta_T^{cfs}.\end{aligned}\tag{4.2}$$

4.1.2 Dual approximation

In order to evaluate the error representation $\mathcal{M}(u) - \mathcal{M}(u_h) = r(z)$, we need an approximation to the dual solution z . As discussed in [Section 3.1.1](#), we need a to find a dual approximation that is not found in the natural discretization, \hat{V}_h , of the dual solution space, since the residual in such a function vanishes. We consider two strategies for approximating such a function: an approximation found by means of a higher order method, and an extrapolated dual approximation. To distinguish between them, we add, respectively, the subscript H and E :

$$\begin{aligned}\tilde{z}_H &\text{ - Higher order approximation of dual} \\ \tilde{z}_E &\text{ - Extrapolated approximation of dual}\end{aligned}\tag{4.3}$$

Higher order dual approximation

The *higher order* dual approximation is found by solving the dual variational problem on the same mesh as the primal problem, only using one order higher function spaces.

Extrapolated dual approximation

The *extrapolated* dual approximation is found by solving the same order dual problem, that is; using the same mesh and polynomial order as the primal problem, then extrapolating this computed solution to a higher order function space. This technique is developed in [\[22\]](#), and further details of the derivation can be found there.

4.1.3 Mesh marking

Given a set of error indicators η_T , the cells that are to be refinement are marked using a marking strategy and a marking fraction indicating how large a fraction of the marked cells are to be refined. The marking strategies that we consider are Dörfler and fixed fraction.

Dörfler

When we use Dörfler marking, a set of cells, for which the total added error exceeds a fraction of the total error are marked. The Dörfler marking procedure was derived in [\[21\]](#).

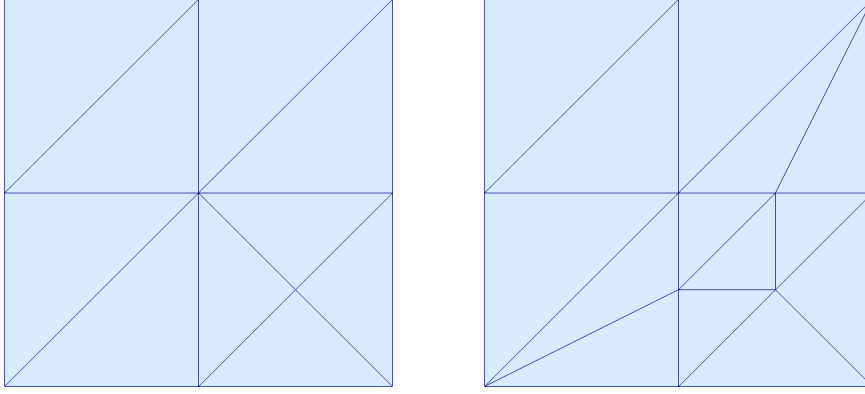


Figure 4.1: Refinement of a single cell using the recursive bisection (to the left) and regular cut refinement algorithm (to the right).

Fixed fraction

The fixed fraction marking procedure is exactly as the name suggests: a fixed percentage of the cells with the highest indicated error are marked for refinement. This percentage is determined by the marking fraction which is also an argument to the adaptive algorithm.

4.1.4 Refinement

The refinement algorithm provides a recipe for how the cells are refined. We consider here the *Rivara recursive bisection* refinement algorithm and the *regular cut* refinement algorithm. The refinement procedures are described below, and the procedures performed on a single cell are illustrated in [Figure 4.1](#).

The Rivara recursive bisection algorithm

With Rivara recursive bisection refinement, which we shall usually refer to as recursive bisection, a triangle is refined by choosing the longest edge and splitting it down the middle so that we have two equal triangles.

The regular cut refinement algorithm

When regular cut refinement is applied, a triangle is subdivided into four equal triangles, and the neighbouring cells that get hanging nodes are refined using the recursive bisection algorithm.

4.2 Computing goal functionals

The goal functionals for which we test adaptive error control in this thesis are: (1) a *manufactured goal functional*, which we utilise to be able to study the performance of the adaptive error control algorithm for smooth primal and dual solutions (2) Three representations of the *wall shear stress goal functional* over some portion of the domain boundary. For the manufactured goal functional, we will derive that possession of known exact primal solutions in this case leads to known exact dual solutions. For the shear stress formulations we provide in this section discussion of some numerical aspects.

4.2.1 The manufactured goal functional

The manufactured goal functional is derived so as to obtain known analytical dual solutions when we have known analytical primal solutions. The manufactured goal functional is defined as

$$\mathcal{M}((\mathbf{u}, p)) = \int_{\Omega} \mathbf{f}_e \cdot \mathbf{u} \, dx + \int_{\Gamma_N} \mathbf{g}_e \cdot \mathbf{u} \, dS, \quad (4.4)$$

for a general boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, with imposed boundary conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{0} \text{ on } \Gamma_D, \\ \sigma(\mathbf{u}, p) \cdot \mathbf{n} &= \mathbf{g} \text{ on } \Gamma_N. \end{aligned}$$

That the Dirichlet boundary conditions are homogeneous, is crucial in order to obtain known exact dual solutions. For simplicity we let $\mathbf{x} = (\mathbf{u}, p)$ denote the primal solutions, $\mathbf{z} = (\mathbf{z}_v, z_q)$ the dual solutions, $\mathbf{y} = (\mathbf{v}, q)$ the primal test functions and $\mathbf{w} = (\mathbf{w}_v, w_q)$ the dual test functions. We then have

$$\begin{aligned} \text{Primal problem:} \quad & \text{find } \mathbf{x} \in W \text{ such that} & a(\mathbf{x}, \mathbf{y}) &= L(\mathbf{y}) & \forall \mathbf{y} \in W \\ \text{Dual problem:} \quad & \text{find } \mathbf{z} \in W^* \text{ such that} & a^*(\mathbf{z}, \mathbf{w}) &= \mathcal{M}(\mathbf{w}) & \forall \mathbf{w} \in \hat{W}^* \end{aligned}$$

The bilinear form a , and linear form L , for the Stokes equations are given in [Section 2.3](#). We have

$$L(\mathbf{y}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} \, dS \quad (4.5)$$

Due to the homogeneous Dirichlet boundary conditions, the primal test and trial function spaces are the same. We have $W^* = \hat{W}$, which in this case is the same as the primal trial space W . This means that the primal and dual solutions are found in the same function space, and $\hat{W} = W$. For the Stokes equations, with the sign of the pressure gradient switched, we know that a is symmetric. If \mathbf{x}_e denotes the exact primal solution, we obtain

$$\begin{aligned} a(\mathbf{x}_e, \mathbf{v}) &= L(\mathbf{v}) = \int_{\Omega} \mathbf{f}_e \cdot \mathbf{v} \, dx + \int_{\Gamma_N} \mathbf{g}_e \cdot \mathbf{v} \, dS \\ &= \mathcal{M}(\mathbf{v}) \end{aligned} \quad (4.6)$$

This leads to $a(\mathbf{x}_e, \mathbf{v}) = a^*(\mathbf{z}, \mathbf{v})$. This is an equation that can be solved for \mathbf{z} , and thus we have obtained known analytical dual solutions. Incidentally, since a is symmetric, we have $a^*(\mathbf{z}, \mathbf{v}) = a(\mathbf{z}, \mathbf{v})$, and thus $\mathbf{z} = \mathbf{x}_e$, and the analytical primal and dual solutions are thus the same.

4.2.2 The shear stress goal functional

The first shear stress representation is the traditional formula which consists of a surface integral over the boundary. The last two are both representations of the shear stress involving integrals over the interior of the domain, one which has a small support in Ω , the other with larger support in Ω , and whose extent will depend on the domain at hand. In this context, computation of volume integrals is preferable to the computation of surface integrals from a numerical point of view [28]. In [27] they also found the volume formulation to yield more accurate results. On a continuous level, when (\mathbf{u}, p) solves the continuous variational problem (2.18), the two formulations (2.11) and (2.13) are equivalent. On a discrete level however, for (\mathbf{u}_h, p_h) that solves (2.20), this does not apply. We can thus not expect that

the computed goal functional values are the same. But we expect that the approximation error will be relatively small and that they will approach approximately the same value when the mesh is refined. If \mathbf{v}_d has a small support in Ω , the computational cost of computing the integrals is smaller. For details on the v_d constructed in the numerical experiments, see [Section 5.1.4](#) and [Section 5.2.2](#).

4.3 Performance evaluation

4.3.1 Evaluating the quality of error estimates

The goal of our adaptive method is to accurately approximate the value of the goal functional with a minimal computational work. To achieve this we need an as accurate as possible estimate of the error in the goal functional, and an efficient strategy for modification when the stop criterion is not met. We make a continuous assessment of the progress based on computational feedback from the adaptive algorithm. To which degree the a posteriori error estimate correctly approximates the actual error in the goal functional is measured in an efficiency index, I_h , which gives the ratio of the error estimate η_h to the actual error η

$$I_h = \frac{\eta_h}{\eta} = \frac{\sum_T \eta_T}{|\mathcal{M}(e)|}. \quad (4.7)$$

The efficiency of the mesh modification strategy is measured in the convergence of the error on the goal functional value, $\eta = |\mathcal{M}(u) - \mathcal{M}(u_h)|$. We provide plots of η versus number of degrees of freedom.

4.3.2 Computing reference values

Reference values for the goal functionals allow for evaluation of the computed error estimates. They are found in one of two ways. If the problem at hand has a known analytical solution, the reference value can be found by hand-calculation. If, on the other hand, no analytical solution is available, the reference value must be found through computation by evaluating the goal functional in an approximate solution computed on a uniformly refined, very fine, mesh. To corroborate that this computed value is correct and found on a sufficiently fine mesh, we supplement with an evaluation of the goal functional on a series of adaptively refined meshes, to assure that they in fact converge towards the same value.

4.3.3 Estimating convergence rates

A common model for the spatial discretization error of a computed solution is $e = e(h) = Ch^r$, where h is the element size and C and r are unknown constants [16, 19]. On a series of increasingly fine meshes, we find the rate at which the corresponding computed solutions approach the exact solution by comparing the discretization errors. If h_j and h_{j+1} are grid-spacings of two consecutive meshes and e_j and e_{j+1} the corresponding solution errors, the exponential, r , of the convergence rate is

$$r = \frac{\log(e_j/e_{j+1})}{\log(h_j/h_{j+1})}. \quad (4.8)$$

r as $h_j \rightarrow 0$ gives the convergence rate. The error e_i is computed in the L^2 - or H^1 -error-norm. If k is the solution polynomial degree, then the optimal convergence rate, l , is $l = k + 1$ in the L^2 -norm and, $l = k$ in the H^1 -norm. DOLFIN provides a function `errornorm`, which approximates the functions by polynomials of high degree by interpolating them to the same high order function space, subtracts the two fields and then computes the integral. It takes

as optional argument the norm-type you wish to use, with default being the L^2 -norm. When the mesh is uniformly refined, we usually employ a halving of the mesh size from one iteration to the next, in which case the denominator of (4.8) vanishes, and we have,

$$r = \log(e_j/e_{j+1}). \quad (4.9)$$

4.4 Tools

The adaptive solver is implemented using the FEniCS finite element software library [1, 18]. The FEniCS project is a tool for automated solution of partial differential equations (PDEs), and is a collaborative project between a number of research institutes around the world. FEniCS is very user friendly, among others because of the close correspondence between the mathematical formulas and the DOLFIN syntax. FEniCS programs can be written both in the Python and C++ programming language. All developed code is written in python [17, 4]. Visualization of results is done using Paraview [3].

Chapter 5

Test cases

In this chapter we introduce the test cases considered in this thesis. The first test case is the two-dimensional unit square, on which we construct known smooth analytical solutions. The second test case is an idealized two dimensional model of a blood vessel with a saccular aneurysm.

On the unit square, we will first investigate adaptive error control in a manufactured goal functional, defined so as to obtain known smooth exact dual solutions. The manufactured goal functional is introduced in [Section 4.2.1](#). It is defined as

$$\mathcal{M}_M((\mathbf{u}, p)) = \int_{\Omega} \mathbf{f}_e \cdot \mathbf{u} \, dx + \int_{\Gamma_N} \mathbf{g}_e \cdot \mathbf{u} \, dS,$$

where (\mathbf{u}, p) is the primal solution, \mathbf{f} is a given body force and \mathbf{g} is the function that gives the Neumann boundary conditions on Γ_N , that is $\sigma((u, p)\mathbf{n} = \mathbf{g}$ on Γ_N , where σ is the stress tensor.

On both domains, we will test adaptive error control in the wall shear stress on a portion of the boundary. We will consider all three representations of the shear stress defined in [Section 2.1.4](#): the surface form, and the volume forms with small and large interior support. The shear stress formulations are distinguished by the added subscripts S , V_1 and V_2 , denoting, respectively, the surface form, the volume with small support and the volume with large support. The three formulations for the shear stress are defined as

$$\begin{aligned} \mathcal{M}_S((\mathbf{u}, p)) &= \int_{\Gamma} h \sigma(\mathbf{u}, p) \mathbf{n} \cdot \mathbf{t} \, dS, \\ \mathcal{M}_{V_1}((\mathbf{u}, p)) &= \int_{\Omega} \nabla \cdot \sigma(\mathbf{u}, p) : \nabla \mathbf{v}_{d_1} \, dx - \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_{d_2} \, dx, \\ \mathcal{M}_{V_2}((\mathbf{u}, p)) &= \int_{\Omega} \nabla \cdot \sigma(\mathbf{u}, p) : \nabla \mathbf{v}_{d_2} \, dx - \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_{d_2} \, dx, \end{aligned} \tag{5.1}$$

where \mathbf{v}_{d_1} and \mathbf{v}_{d_2} has a small and large interior support, respectively, and $h = h((x, y))$, as introduced in [Section 4.2.2](#), is an appropriate smoothing function defined for each individual case. For the volume shear stress formulations, smoothing is incorporated in \mathbf{v}_{d_1} and \mathbf{v}_{d_2} .

5.1 Test case I: Two dimensional unit square

5.1.1 Problem definition

In the first test case we model Stokes flow, and our computational domain, Ω , is the square with width and height of unit length. A no-slip condition is imposed on the left-hand

boundary, and the remaining boundary is given a non-homogeneous Neumann boundary condition corresponding to imposing an induced force. The boundary conditions read

$$\begin{aligned} \mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} + p \mathbf{n} &= \mathbf{g} \quad \text{on } \Gamma_N, \\ \mathbf{u} &= \mathbf{0} \quad \text{on } \Gamma_D, \end{aligned} \quad (5.2)$$

where μ is the dynamic viscosity, the force \mathbf{g} is found from the exact solutions and $\Gamma = \{0\} \times (0, 1)$ and $\Gamma_N = \partial\Omega \setminus \Gamma$ are the parts of the boundary with essential and natural boundary conditions respectively.

5.1.2 Exact solutions

On this domain we can find an exact velocity \mathbf{u}_e , and pressure p_e . In two dimensions, stream functions can be defined, which are used to find the instantaneous streamlines in flow [15]. By letting the velocity be expressed as the curl of the stream function ψ , $\mathbf{u}_e = \nabla \times \psi = (\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x})$, fulfilment of the no-divergence condition is assured, since the divergence of the curl of a vector field is identically zero. We define the stream function as $\psi = x^k \phi(x, y)$ for some $k > 1$, where $\phi(x, y) = \sin(2\pi x) \sin(2\pi y)$. This way the homogeneous Dirichlet boundary condition $\mathbf{u}|_{\Gamma_D} = 0$, is also fulfilled:

$$\mathbf{u}_e = \nabla \times (x^k \phi) = \begin{pmatrix} x^k \frac{\partial \phi}{\partial y} \\ -x^{k-1} (k\phi + x \frac{\partial \phi}{\partial x}) \end{pmatrix}, \quad \nabla \cdot \mathbf{u}_e = \frac{\partial^2 \psi}{\partial x \partial y} - \frac{\partial^2 \psi}{\partial y \partial x} = 0.$$

Since x appears in both terms of \mathbf{u}_e , we have $\mathbf{u}_e|_{\Gamma} = \mathbf{0}$. The exact primal solutions velocity and pressure are given by

$$\begin{aligned} \mathbf{u}_e(x, y) &= \begin{pmatrix} 2\pi x^2 \sin(2\pi x) \cos(2\pi y) \\ -2x \sin(2\pi x) \sin(2\pi y) - 2\pi x^2 \cos(2\pi x) \sin(2\pi y) \end{pmatrix} \\ p_e(x, y) &= \sin(\pi x) \sin(\pi y) \end{aligned} \quad (5.3)$$

The exact velocity and pressure are shown in Figure 5.1. Inserting \mathbf{u}_e and p_e into the momentum equation and Neumann boundary condition, we can find the body force \mathbf{f}_e and the Neumann force \mathbf{g}_e .

$$\begin{aligned} \mathbf{f}_e &= -\nabla \cdot \sigma(\mathbf{u}_e, p_e) \\ &= \begin{pmatrix} -4\nu\pi [\sin(2\pi x)(1 - 4\pi^2 x^2) + 4\pi x \cos(2\pi x)] \cos(2\pi y) - \pi \cos(\pi x) \sin(\pi y) \\ 4\nu\pi [\cos(2\pi x)(3 - 4\pi^2 x^2) - 8\pi x \sin(2\pi x)] \sin(2\pi y) - \pi \sin(\pi x) \cos(\pi y) \end{pmatrix}. \end{aligned} \quad (5.4)$$

The force \mathbf{g}_e is defined as $\mathbf{g}_e = \sigma(\mathbf{u}_e, p_e) \mathbf{n}$ on Γ_N , which spans three sides of the square. By partitioning Γ_N into three disjoint domains, $\Gamma_N = \Gamma_{N_1} \cup \Gamma_{N_2} \cup \Gamma_{N_3}$, we can find the expression for \mathbf{g}_e on each subdomain.

$$\Gamma_{N_1} = \{1\} \times (0, 1), \quad \Gamma_{N_2} = (0, 1) \times \{0\}, \quad \Gamma_{N_3} = (0, 1) \times \{1\}.$$

If we define $\mathbf{g}_i = \mathbf{g}_e|_{\Gamma_{N_i}} = \sigma_e|_{\Gamma_{N_i}} \mathbf{n}_i$, where \mathbf{n}_i is the normal vector on Γ_{N_i} , the restriction of \mathbf{g}_e to Γ_{N_i} , we find that

$$\begin{aligned} \mathbf{g}_1(1, y) &= \nu \frac{\partial \mathbf{u}}{\partial x} = 4\nu\pi \begin{pmatrix} \pi \cos(2\pi y) \\ -2 \sin(2\pi y) \end{pmatrix}, \\ \mathbf{g}_2(x, 0) &= 4\nu\pi \begin{pmatrix} 0 \\ x \sin(2\pi x) + \pi x^2 \cos(2\pi x) \end{pmatrix}, \\ \mathbf{g}_3(x, 1) &= -\mathbf{g}_2(x). \end{aligned} \quad (5.5)$$

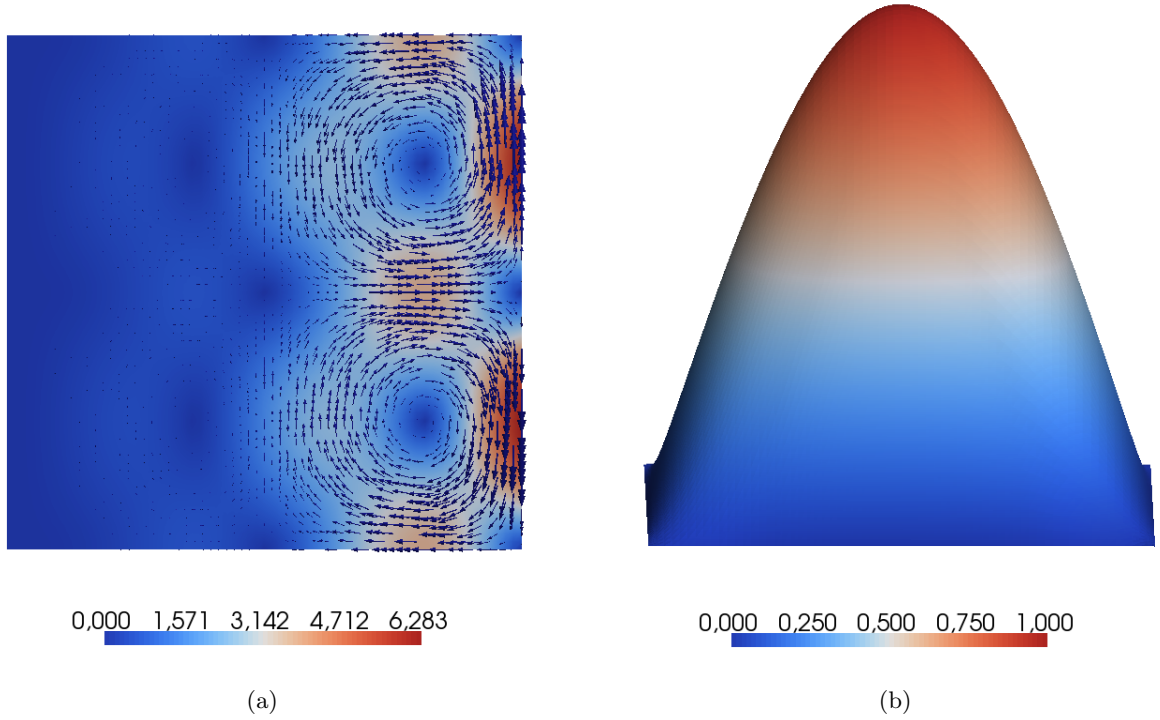


Figure 5.1: Fluid velocity (left) and pressure (right) for the unit square test case on a uniform mesh with 8192 triangles.

5.1.3 The manufactured goal functional

We first consider error control in the manufactured goal functional

$$\mathcal{M}_M((\mathbf{u}, p)) = \int_{\Omega} \mathbf{f}_e \cdot \mathbf{u} \, dx + \int_{\Gamma_N} \mathbf{g}_e \cdot \mathbf{u} \, dS.$$

Since we have homogeneous Dirichlet boundary conditions, and a symmetric bilinear form, considering the manufactured goal leads to known exact dual solutions that coincide with the smooth exact primal solutions. This was derived in [Section 4.2.1](#). We usually refer to the exact primal solutions simply as the exact solutions. This idealized case provides a good opportunity to perform sensitive testing of the effects of varying the various error- and adaptivity parameters in the adaptive algorithm. The reference value for \mathcal{M} is found to be

$$\mathcal{M}_{M,ref} = 37.6481242$$

It is obtained by evaluation of \mathcal{M}_M in the exact solutions on a uniform mesh of approximately 100 000 cells. [Figure 5.2](#) shows that the adaptively obtained computed goal functional values approach the same reference value.

5.1.4 The shear stress goal functional

Next we consider the shear stress over the Dirichlet boundary, at $x = 0$. We will consider the three shear stress representations given in [Section 2.1.4](#). Here, \mathcal{M}_S is the surface formulation, whereas \mathcal{M}_{V_1} and \mathcal{M}_{V_2} are the formulations involving volume integrals in which \mathbf{v}_{d_1} , \mathbf{v}_{d_2} has a small and large interior support respectively. By the construction of the exact solutions,

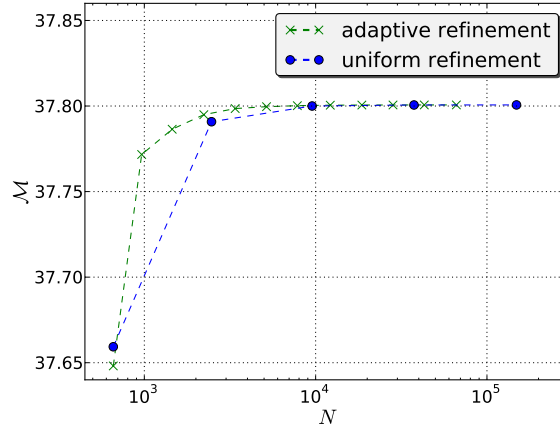


Figure 5.2: Values of the manufactured goal functional plotted against degrees of freedom. Each point marks an iteration. Adaptive meshes are obtained using a Dörfler marking with marking fraction 0.5 and recursive bisection refinement.

there is no shear stress on the Dirichlet boundary. If $\mathbf{u} = (u_1, u_2)$, we have

$$\int_{\Gamma_D} \sigma(\mathbf{u}_e, p_e) \mathbf{n} \cdot \mathbf{t} \, dS = \int_{\Gamma_D} \frac{\partial u_2}{\partial x} \, dS = 0,$$

since $\frac{\partial u_x}{\partial y}(0, y) = 0$. From a physical point of view, it is less interesting to perform error control with respect to a zero shear stress, but as a methodological study of the performance of the adaptive method the results are just as relevant. However, having the goal functional value converge towards zero imposes some numerical issues. As the goal functional value is better approximated, it will approach zero. As will the computational error and the error estimate. This can cause some instabilities, as computed values involved can suddenly fall below machine precision. It can be seen in the results as sudden drops in the efficiency index.

The \mathbf{v}_d with small support is defined by representing \mathbf{v}_d as a continuous piecewise quadratic function, taking the value one in all degrees of freedom associated with the second component on the goal boundary Γ_D , and letting all other degrees of freedom vanish. This results in a vector field that is non-zero only on the cells intersecting the goal boundary. On the other hand, the \mathbf{v}_d with large support is defined by letting the second component of \mathbf{v}_d vary linearly from one to zero from the goal boundary to the right hand side boundary ($x = 1$).

5.2 Test case II: Idealized saccular aneurysm

5.2.1 Problem definition

We consider next an idealized model of a blood vessel with a saccular aneurysm. The original model was a 20mm long and 2mm wide, straight channel with a bulge modelling the aneurysm with a diameter of approximately 3.5mm. To increase flow into the aneurysm, and make derivations of shear stress smoothing applicable to more general cases, the mesh was altered by direct coordinate manipulation. Both domains are illustrated in Figure 5.3. At the inlet the velocity is given a parabolic profile, with a peak velocity of 700mm/s. At the outlet we impose free outflow. On the remaining boundary, which is the channel side walls and over the aneurysm, a no-slip condition is imposed. The boundary conditions read

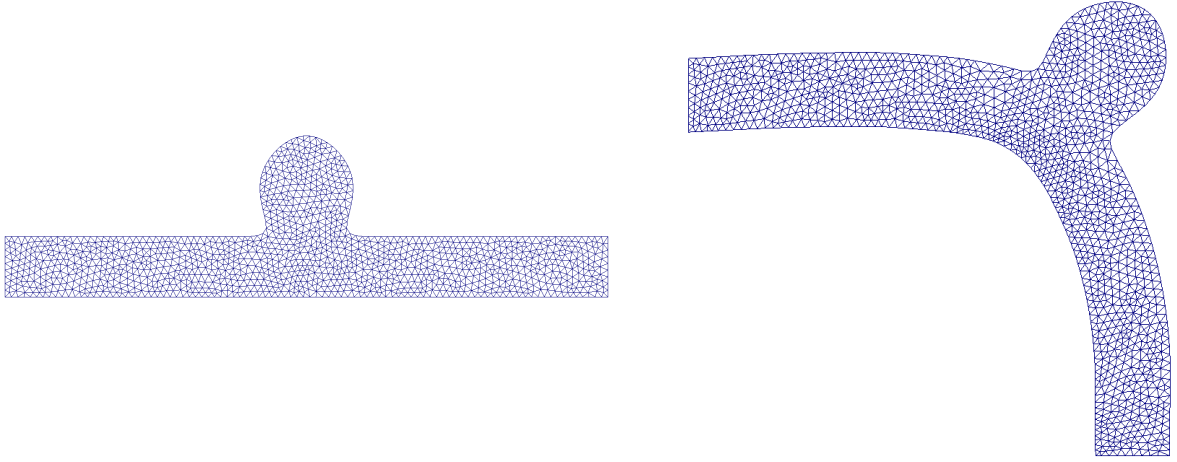


Figure 5.3: Computational domain for the idealized aneurysm. On top the initial model of a straight channel with a bulge and on bottom the manipulated model.

$$\begin{aligned}
 \mathbf{u} &= \mathbf{0} && \text{on } \Gamma, \\
 \mathbf{u} &= h(y) && \text{on } \Gamma_I, \\
 \mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} + p \mathbf{n} &= \mathbf{0} && \text{on } \Gamma_O,
 \end{aligned} \tag{5.6}$$

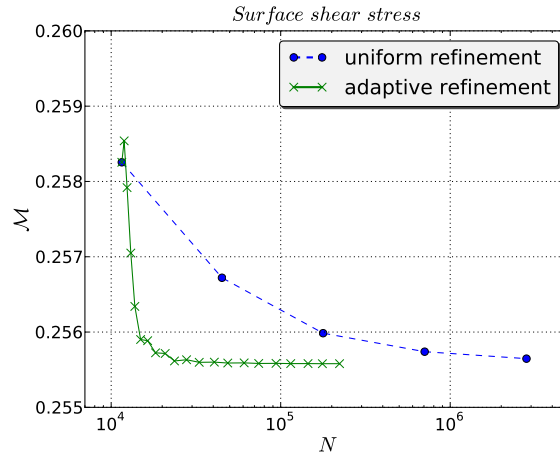
where μ is the dynamic viscosity, and $\Gamma_D = \cup \Gamma_N$ and Γ_O are the parts of the boundary with essential and natural boundary conditions respectively.

5.2.2 The shear stress goal functional

On the domain of the idealized aneurysm, we wish compute the shear stress inside the aneurysm. We will consider the three shear stress representations given in [Section 2.1.4](#). Here, \mathcal{M}_S is the surface formulation, whereas \mathcal{M}_{V_1} and \mathcal{M}_{V_2} are the formulations involving volume integrals in which \mathbf{v}_{d_1} , \mathbf{v}_{d_2} has a small and large interior support, respectively. Reference values for the three shear stress goal functionals are found by comparing the adaptively and uniformly obtained values, as shown in [Figure 5.2.2](#). The reference values are found to be

$$\begin{aligned}
 \mathcal{M}_{S,ref} &= 0.25568, \\
 \mathcal{M}_{V_1,ref} &= 0.25536, \\
 \mathcal{M}_{V_2,ref} &= 0.25555.
 \end{aligned} \tag{5.7}$$

The \mathbf{v}_d with small support is found by representing \mathbf{v}_d as a function in the Crouzieux-Raviart element space. We create a self-defined expression `TangentExpression` subclassing `Expression` that defines the tangent vector by accessing the normal vector on the facets. The tangent expression is then applied to \mathbf{v}_d as a Dirichlet boundary condition. We add interior support to \mathbf{v}_d by solving a Poisson problem on the domain with \mathbf{v}_d as boundary condition on the entire domain boundary. The implementation of \mathbf{v}_d as a $[CG_1]^2$ function is shown in [Figure 5.7](#). For visualization purposes the CR-function is not suited, since it is evaluated in the midpoint of the edges and not in the mesh points. Since both the CG_1 and CR-functions are meant to represent the same function, namely the tangent on the boundary, we use the CG_1 -function to illustrate the implementation of \mathbf{v}_d .



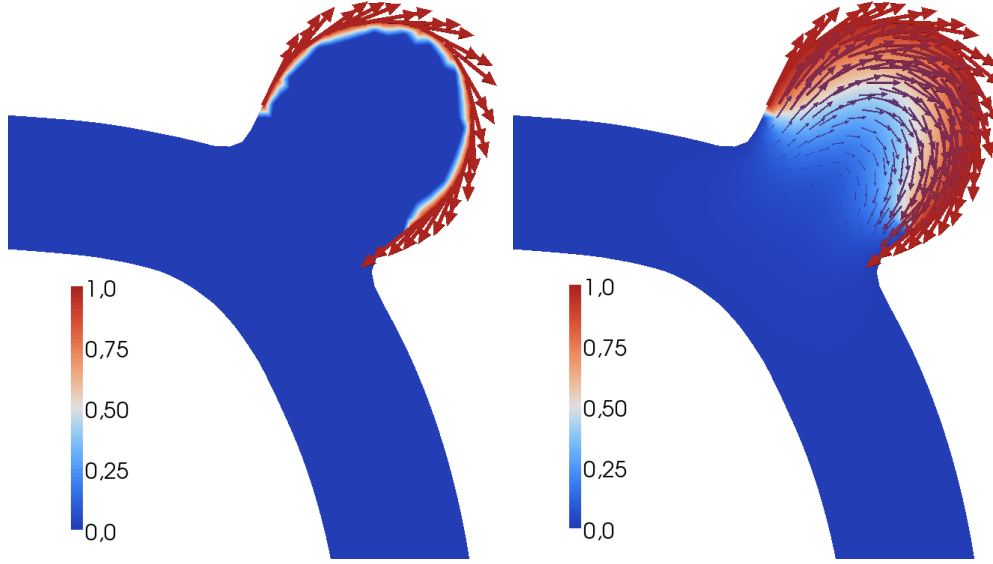


Figure 5.6: \mathbf{v}_d with small (left) and large (right) interior support. Without added smoothing.

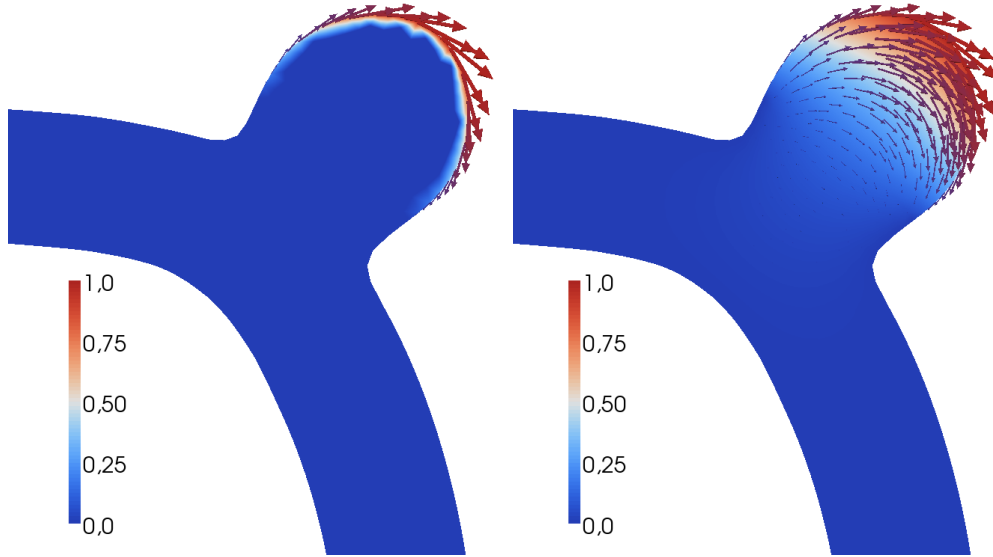


Figure 5.7: \mathbf{v}_d with small (left) and large (right) interior support. With added smoothing.

Chapter 6

Numerical results

In this chapter we present numerical results for the two test cases. The adaptive solver was implemented in DOLFIN 1.0 [18] inspired by the automated framework in DOLFIN 0.9.9 [22]. The complete source code can be found at <http://folk.uio.no/elinesu>.

6.1 Test case I

The velocity flow pattern and the pressure on the unit square are shown in Figure 6.1. When applying a uniform mesh refinement, the discretization error for both the primal velocity and pressure, as measured in the L^2 -norm, decrease at a rate of one higher than the respective degree of approximation, which we call the optimal convergence rate, as we might expect for such smooth solutions. The computed uniform convergence rates are given in Table 6.1.

$N_i \rightarrow N_{i+1}$	8 \rightarrow 16	16 \rightarrow 32	32 \rightarrow 64	64 \rightarrow 128	Final error
	2.947	2.974	2.988	2.995	5.36e-06
	2.193	2.038	2.007	2.001	2.51e-05

Table 6.1: Convergence rate for the error measured in the L^2 -norm for primal velocity (top) and pressure (bottom). Corresponding to a uniform increase in number of cells, N_i , in each direction.

6.1.1 Adaptive error control in the manufactured goal functional

The manufactured goal functional, \mathcal{M}_M , is described in Section 4.2.1. Its area of influence spans the entire domain, so we expect the global error in \mathcal{M}_M to depend on cells throughout the mesh. This might make the adaptive advantages less prominent than when a more localized target quantity is used, such as the shear stress over a portion of the boundary.

Dual solutions and adaptive meshes

When investigating error control in the manufactured goal functional, we obtain dual solutions that coincide with the primal solutions. This was derived in Section 4.2.1. The dual solutions can thus be viewed in Figure 6.1. The velocity magnitudes are the largest on the right side of the domain, particularly along the boundary. Since the primal- and dual solutions are the same, both large residuals in the momentum equation, as well as large magnitudes of the dual velocity will cause the mesh to be most aggressively refined here.

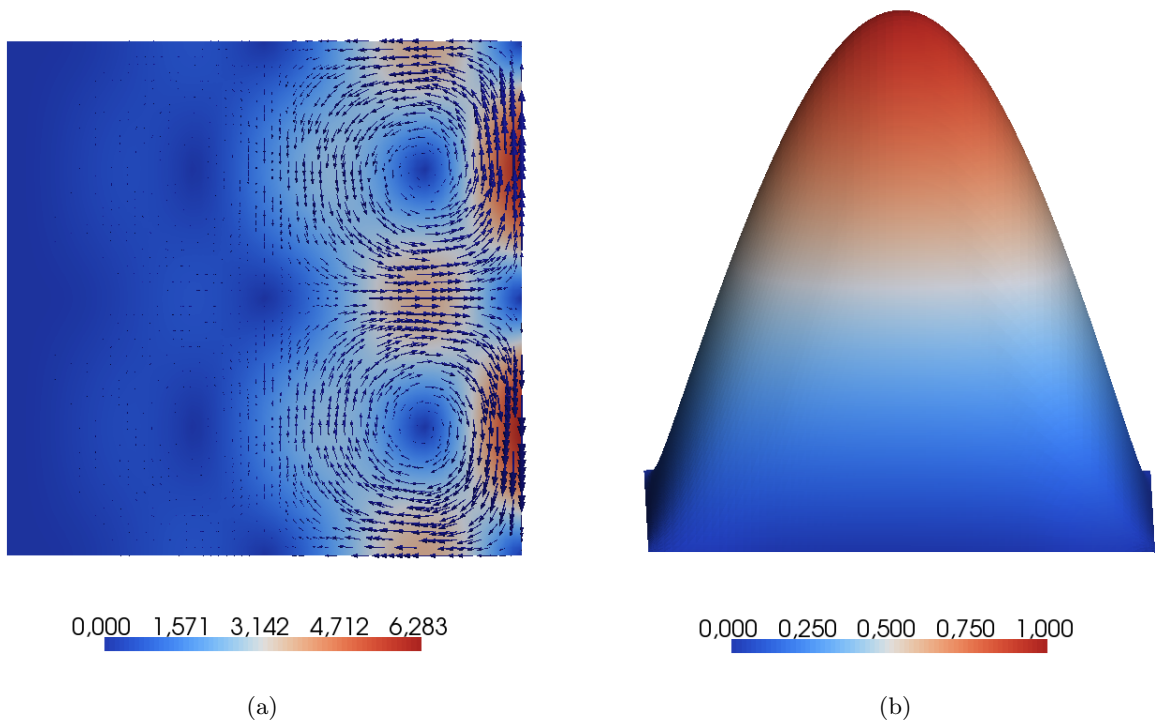


Figure 6.1: Fluid velocity (left) and pressure (right) for the unit square test case computed with Dörfler marking and recursive bisection refinement obtained by 20 iterations. The final mesh has 11334 triangles (51536 degrees of freedom).

Meshes obtained by recursive bisection refinement and regular cut refinement are shown in Figure 6.2. Both refinement algorithms were introduced in Section 4.1.4. We observe that regular cut refinement generates more degrees of freedom than recursive bisection, due to the fact that its tendency to propagate refinement to neighbouring cells is considerably larger than for recursive bisection. As a result it takes fewer iterations to reach the required accuracy when we use the regular cut refinement method. Table 6.2 shows, for a series of adaptive meshes generated using the two refinement algorithms, the average number of cells that are marked to be refined against the average number of cells that are actually refined. For recursive bisection the number of cells that is supposed to be marked correspond well with the number that are in fact refined. For regular cut refinement however, the number of cells that are refined greatly exceeds the number that were marked. This makes for a more aggressive, and one would expect more robust, but on the other hand less precise refinement algorithm. The effect the refinement method has on the generated error estimates is investigated in the context of marking strategies in Section 6.1.1.

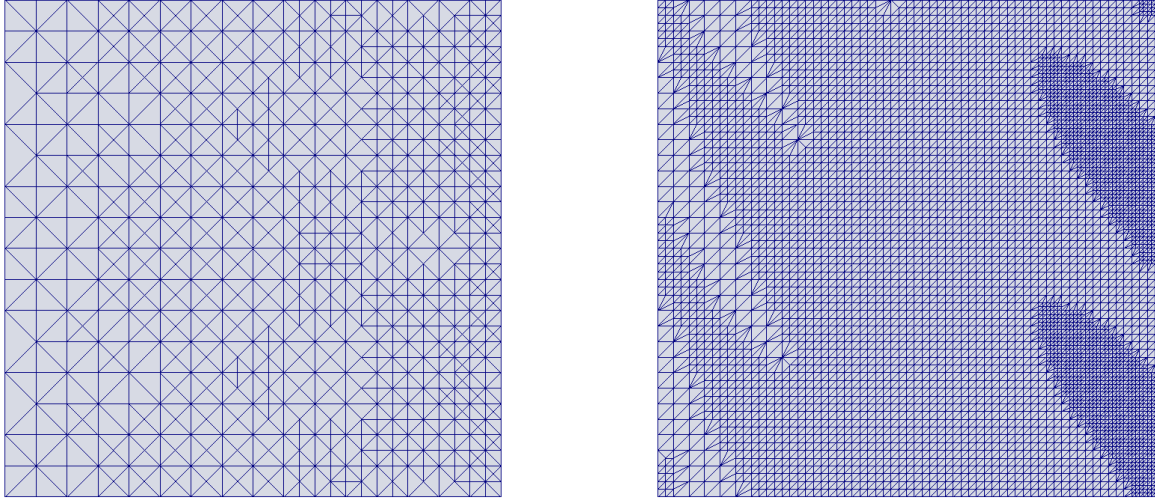


Figure 6.2: Adaptively refined meshes with respect to error-minimization in the manufactured goal functional. Obtained using Dörfler marking and recursive bisection refinement (to the left, 11334 triangles, 20 refinements) and regular cut refinement (to the right, 10127 triangles, 11 refinements).

refinement algorithm	iterations	final cell number	average cells marked (%)	average cells refined (%)
<i>Rivara recursive bisection</i>	20	11334	22.3	23.9
<i>Regular cut</i>	12	14192	13.1	48.3

Table 6.2: Average number of cells marked for refinement and average number of cells that are refined using the two refinement algorithms *Rivara recursive bisection* and *regular cut* refinement with Dörfler marking and fraction 0.5.

Adaptive performance

On account of refinement being required throughout the mesh, it will require a large number of degrees of freedom in order for \mathcal{M}_M to reach a certain level of accuracy. This is demon-

Uniform refinement			Adaptive refinement		
N	η	I_h	N	η	I_h
659	1.5253e-01	0.987369	659	1.5253e-01	0.987369
			1019	2.6666e-02	0.989982
			1688	1.2443e-02	0.991068
2467	1.0667e-02	0.997049	2819	3.3634e-03	0.994635
			4834	1.2597e-03	0.9971
9539	6.9471e-04	0.999282	8178	4.5773e-04	0.997084
			13912	1.5489e-04	0.998674
			23933	5.9854e-05	0.999307
37507	4.4120e-05	0.999822	40075	1.9779e-05	0.999493
			66376	7.2519e-06	0.999770
148739	2.7761e-06	0.999946			

Table 6.3: Number of degrees of freedom, error in \mathcal{M}_M and corresponding efficiency indices for a series of uniform and adaptive meshes obtained using a fixed fraction marking with marking fraction 0.4 and recursive bisection refinement. Refinement level when η is below $1.0\text{e-}04$ and $1.0\text{e-}05$ is marked with blue and red, respectively.

strated in Table 6.3, where we show the error $\eta = |\mathcal{M}_M(u) - \mathcal{M}_M(u_h)|$ and corresponding efficiency indices for a series of meshes generated with uniform and adaptive refinement. The required number of degrees of freedom to reach an accuracy in \mathcal{M}_M of $1.0\text{e-}04$ is almost the same for the two methods. When an accuracy of $1.0\text{e-}05$ is reached, the number of degrees of freedom using uniform refinement is a little more than twice as many as when adaptivity is employed. Considering the added computational work from the additional iterations in the adaptive algorithm, this computational advantage is not significant.

Dual approximation strategies. We consider the quality of the generated error estimates when weighting of the residuals is provided by a higher order dual approximation, \mathbf{w}_H , and extrapolated dual approximation, \mathbf{w}_E . Figure 6.3 shows the errors $\mathcal{M}_M(e)$ and efficiency indices $I_h = \frac{\mathcal{M}(e)}{\eta_h}$, plotted against degrees of freedom for adaptively obtained solutions using the two strategies. The results are very good. Both yield efficiency indices that approach 1.0. \mathbf{w}_H outperforms \mathbf{w}_E , and yields exceptionally accurate error estimates, with efficiency indices that never fall below 0.98. For \mathbf{w}_E the lower bound on the indices is 0.7, and the indices are slightly less stable, but the results are very good nonetheless. In Figure 6.5 we compare the adaptive to the uniformly obtained results for each of the two dual strategies. The results are similar. The adaptive error is slightly smaller than the uniform error, but converges at approximately the same rate.

Error indicators We compare the effect of using different error indicators when both higher order-, and extrapolation provides the weighting. Here we use Dörfler 0.5 marking. The results are shown in Figure 6.6. In both cases the use of error representation indicators yields slightly more stable index values, but more degrees of freedom. Both features are most prominent when the extrapolated dual approximation is used.

Marking parameters We make a comparison between the parameters that together determine how the mesh is refined, namely the marking strategy (Dörfler and fixed fraction), the marking fraction and the refinement algorithm (recursive bisection and regular cut). The marking strategies and refinement algorithms are compared in Figure 6.7, all with a marking fraction of 0.5. Fixed fraction marking gives rise to less oscillations in the efficiency index,

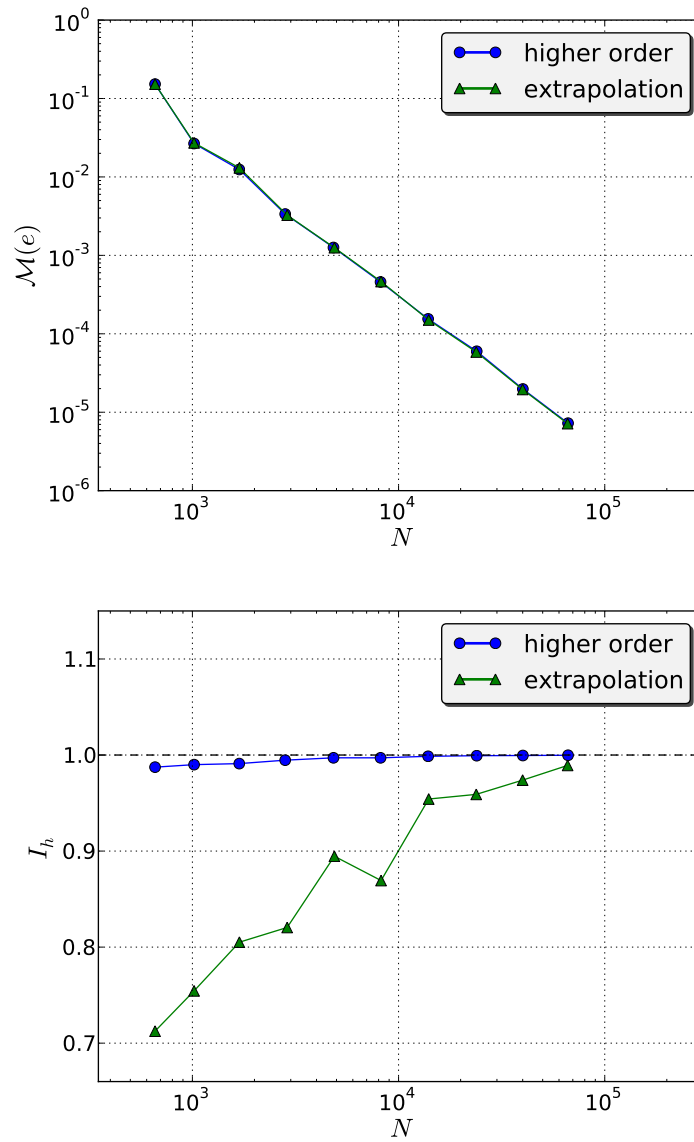


Figure 6.3: Error (top) and efficiency indices (bottom) plotted against degrees of freedom for the higher order and extrapolated dual approximation strategies. With fixed fraction 0.5 marking.

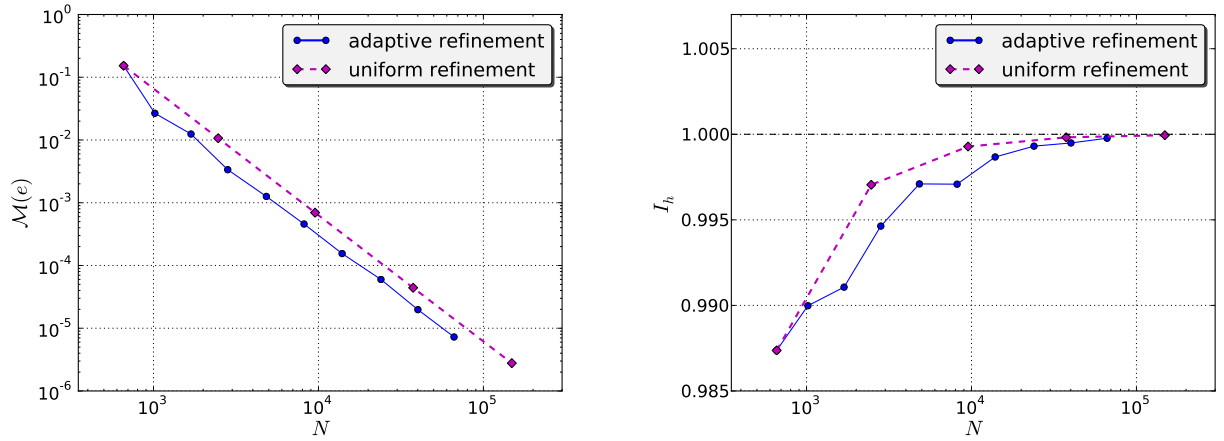


Figure 6.4: Error (left) and efficiency index (right) against degrees of freedom, for the higher order dual approximation strategy under uniform and adaptive refinement.

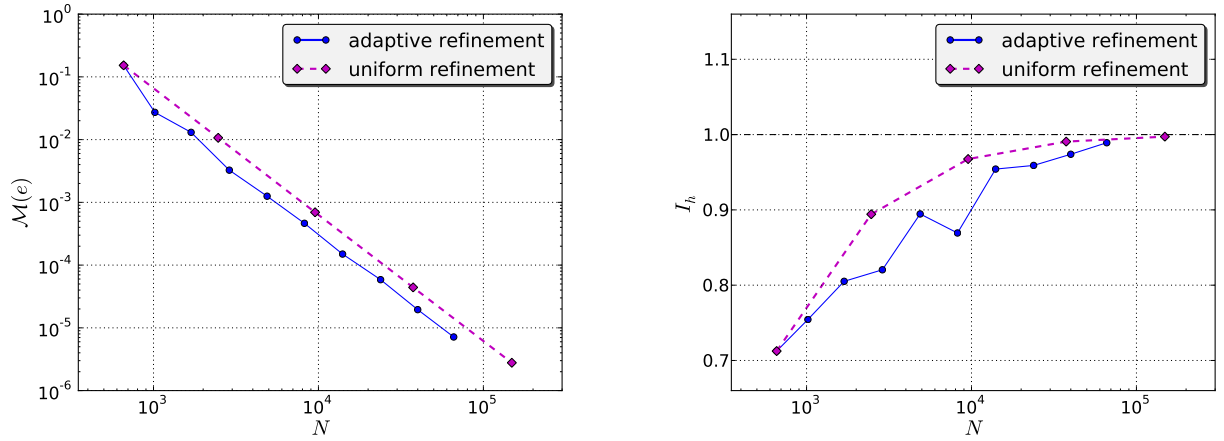


Figure 6.5: Error (left) and efficiency index (right) against degrees of freedom, for the extrapolated dual approximation strategy under uniform and adaptive refinement.

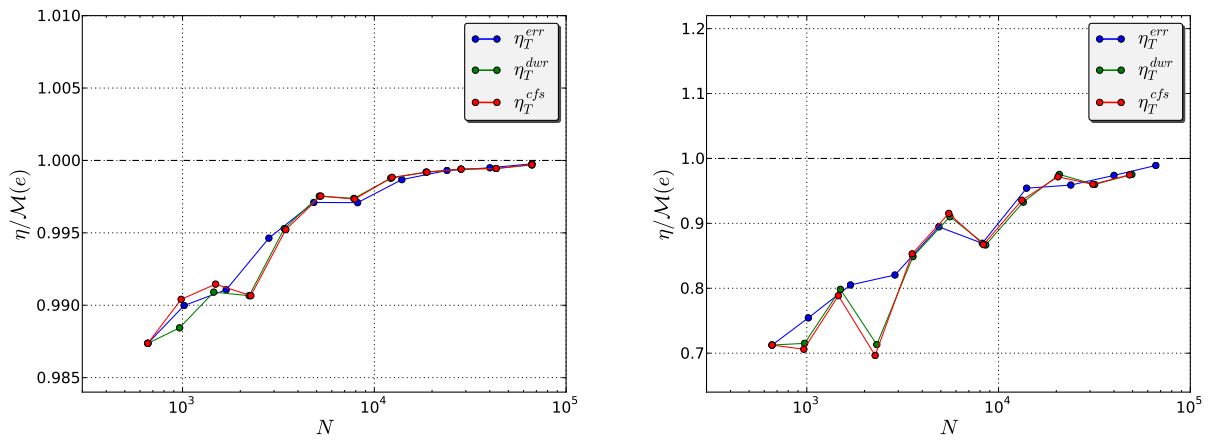


Figure 6.6: Efficiency index for the manufactured goal functional using a higher order dual strategy (left) and extrapolation dual strategy (right).

but in return is more affected by the choice of marking fraction. Dörfler marking is a more adaptive strategy, and it follows approximately the same marking pattern regardless of choice of fraction, as shown in Figure 6.10. In [24] a thorough analysis of the effects of varying the mesh marking fraction was presented for a fluid-structure interaction (FSI) problem. Here was found that the marking fraction had little effect for Dörfler marking but a large effect for fixed fraction marking, just as observed here, and it was concluded that the marking fraction should lie somewhere between 0.3 and 0.5. In this case we need the fraction to be of a certain size to obtain more stable indices. Fixed fraction marking in combination with recursive bisection refinement and a fraction of 0.5 or 0.6 seems to be yield the very best efficiency indices and smallest error. The error is so accurately estimated with all parameter combinations that we must zoom into the details to see the differences, as is done in Figure 6.8.

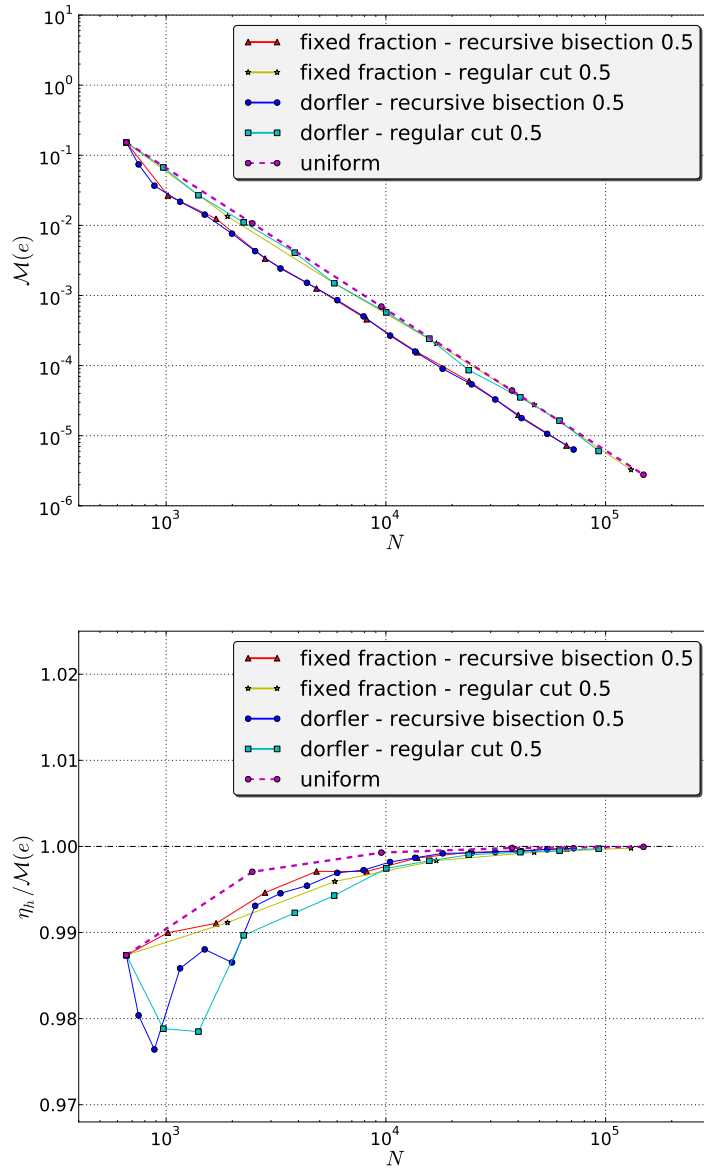


Figure 6.7: Error (top) and index (bottom) for different refinement algorithms and marking strategies all with marking fraction 0.5.

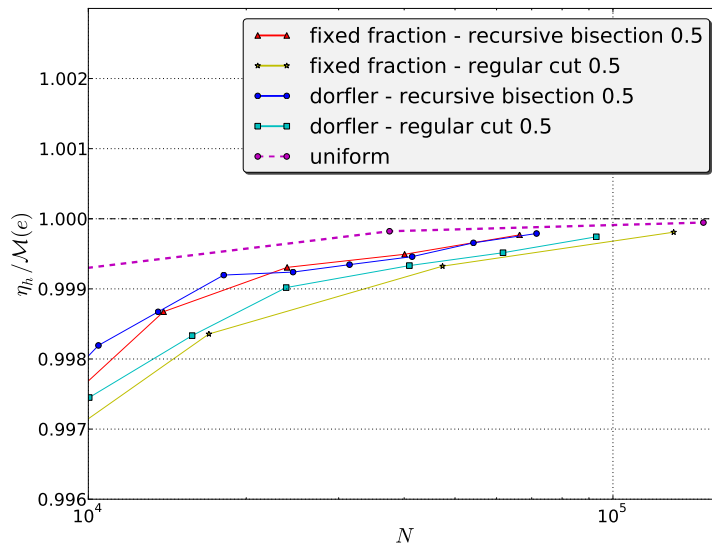


Figure 6.8: Efficiency indices for different refinement algorithms and marking strategies all with marking fraction 0.5.

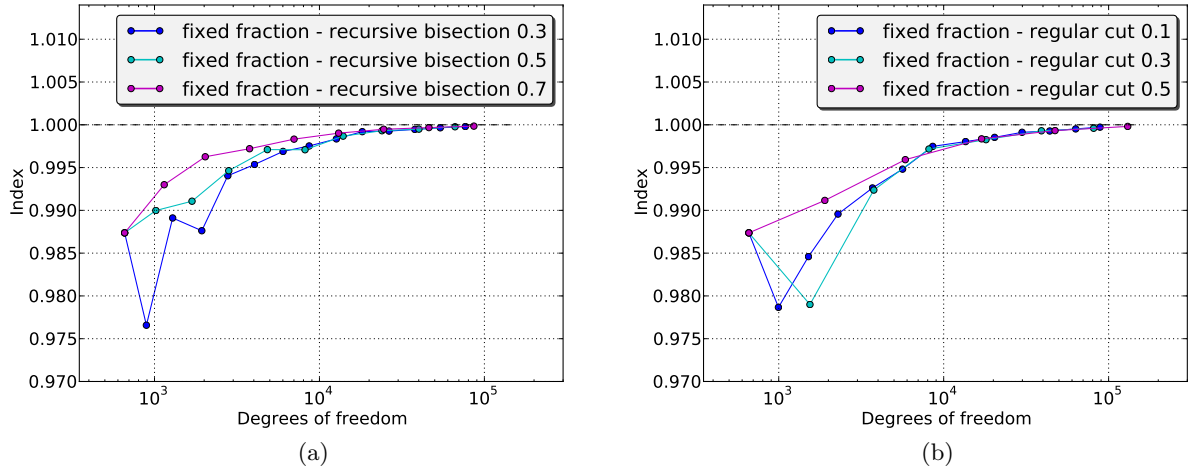


Figure 6.9: Marking fractions compared for fixed fraction marking with recursive bisection (to the right) and regular cut (to the left) refinement. All with higher order dual approximation, η_h^{err} -estimator, and η_T^{err} -indicator.

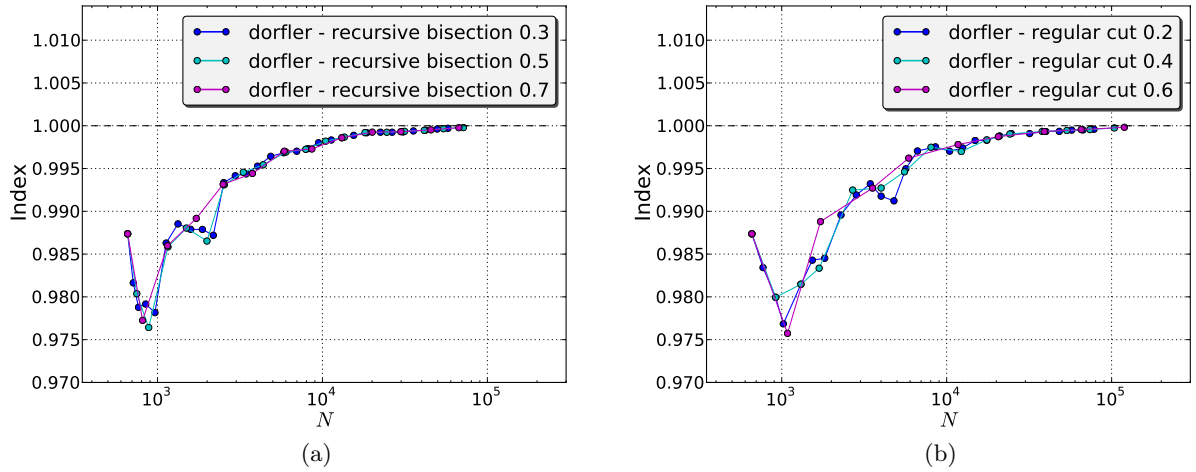


Figure 6.10: Marking fractions compared for drfler marking with recursive bisection (to the right) and regular cut (to the left) refinement. All with higher order dual approximation, η_h^{err} -estimator, and η_T^{err} -indicator.

Convergence of approximations

We consider the convergence of the primal solution, as well as the dual approximations corresponding to the manufactured goal functional, under both uniform and adaptive refinement. In Table 6.1 was given the convergence rates of the primal solution under uniform convergence. Convergence rates for the higher order and extrapolated dual approximations under uniform refinement are shown in Table 6.4 and Table 6.5. Since the primal and dual solutions coincide, results are not included for the same order dual approximation. The higher order- and extrapolated dual approximations are expected to yield a more accurate approximation to the dual solutions. The higher order dual velocity and pressure, as well as the extrapolated dual velocity, has an optimal convergence rate under uniform refinement. The extrapolated dual pressure, however, is one order suboptimal.

In Figure 6.11 the L^2 -error norm is plotted against degrees of freedom for the adaptive and uniform primal solutions, and in Figure 6.12 for the higher order- and extrapolated dual solutions. All the adaptively obtained solutions yield good convergence and approach the uniform rates as the number of degrees of freedom increases. The higher order strategy yields approximately two orders of magnitude more accurate results than the extrapolation strategy.

$N_i \rightarrow N_{i+1}$	8 \rightarrow 16	16 \rightarrow 32	32 \rightarrow 64	64 \rightarrow 128	Final error
	4.007	4.005	4.002	4.001	2.79e-08
	3.402	3.274	3.118	3.041	1.37e-07

Table 6.4: Convergence rate for the error measured in the L^2 -norm for higher order dual velocity (top) and pressure (bottom). Corresponding to a uniform increase in number of cells, N_i , in each direction.

$N_i \rightarrow N_{i+1}$	8 \rightarrow 16	16 \rightarrow 32	32 \rightarrow 64	64 \rightarrow 128	Final error
	3.566	3.830	3.927	3.966	1.94e-06
	1.941	1.962	1.988	1.997	5.61e-05

Table 6.5: Convergence rate for the error measured in the L^2 -norm for extrapolated dual velocity (top) and pressure (bottom). Corresponding to a uniform increase in number of cells, N_i , in each direction.

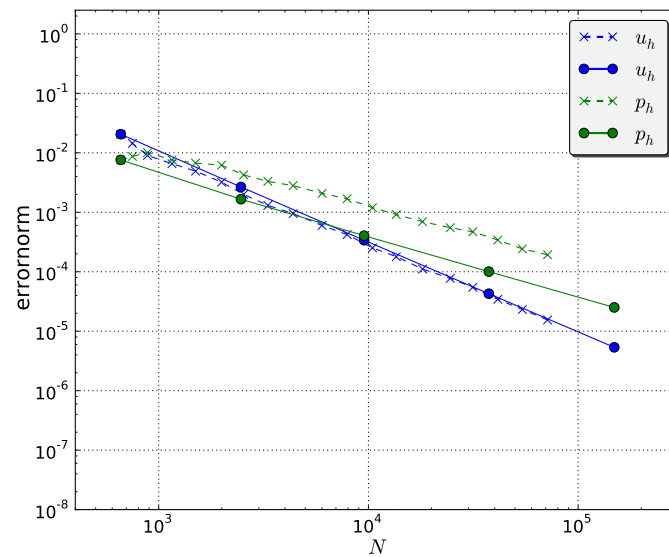


Figure 6.11: L^2 -error of the primal solution under uniform and adaptive refinement. u_h denotes the primal velocity, and p_h denotes the primal pressure. Adaptive rates are given as dotted lines, and uniform rates as solid lines. Adaptive rates obtained using a Dörfler marking with fraction 0.5 and recursive bisection refinement.

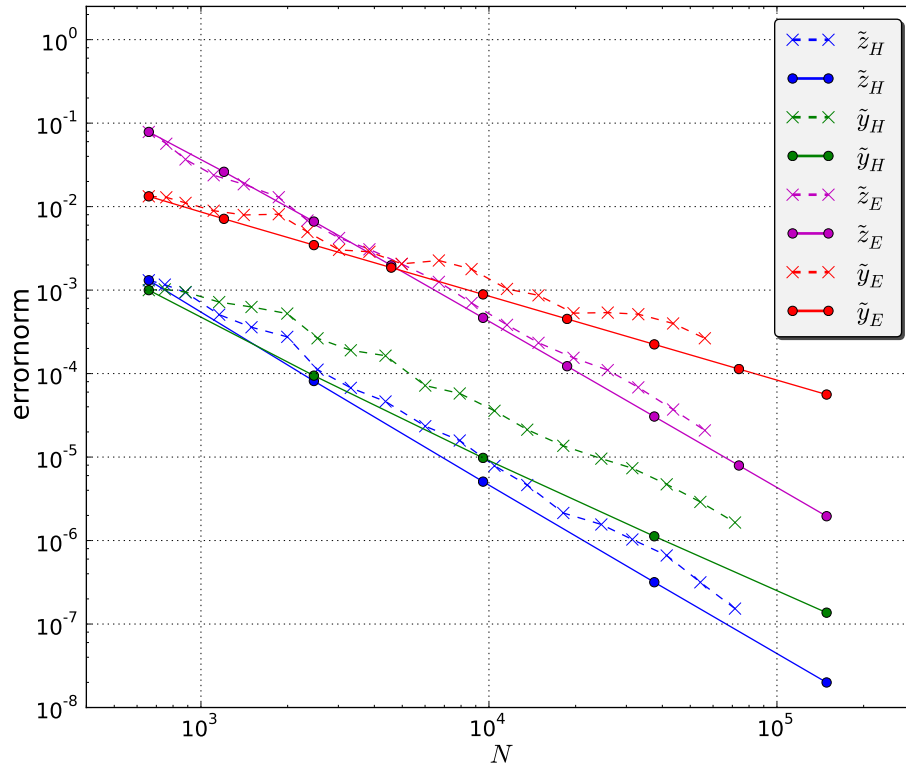


Figure 6.12: L^2 -error of the higher order and extrapolated dual approximations under uniform and adaptive refinement. \tilde{z} and \tilde{y} denote the dual velocity and pressure, respectively. Subscripts H and E denote higher order and extrapolation, respectively. Adaptive rates are given as dotted lines, and uniform rates as solid lines. Adaptive rates obtained using Dörfler marking with fraction 0.5 and recursive bisection refinement.

Adaptive error control in the shear stress goal functional

The shear stress goal functional and the formulations that we apply are introduced in [Section 2.1.4](#). We apply the introduced notation for the representations of the shear stress goal functional. \mathcal{M}_S , \mathcal{M}_{V_1} and \mathcal{M}_{V_2} thus denote, respectively, the surface formulation of the shear stress, the volume formulation of the shear stress with a small interior support and the volume formulation of the shear stress with a large interior support. When something can be taken to mean either of the three, we will simply use \mathcal{M} .

Dual solutions and adaptive meshes

The dual solution generated by the shear stress goal functional has an applied shear stress along the Dirichlet boundary, causing downward fluid movement. The generated dual velocities when the volume shear stress with small and large support constitute the goal functional in the dual variational problem, are shown in [Figure 6.13](#). We see that a large and small interior support in the volume formulation gives rise to significantly different dual velocities. On account of a zero dual velocity on the remaining boundary, hard transitions occur, causing pressure spikes.

The dual solutions, the pressure in particular, are sensitive to the form of \mathcal{M} in the dual variational problem. Using the surface shear stress formulation, \mathcal{M}_S , we experience some numerical instabilities in the dual pressure along the goal boundary. In [Figure 6.14](#) and [Figure 6.15](#) this is illustrated for the higher order dual approximation on a uniform mesh and an adaptive mesh obtained by Dörfler marking with fraction 0.5 and recursive bisection refinement. We see that spikes appear along the boundary on which the shear stress is defined, an effect that is more evident when adaptive refinement is applied. In [Figure 6.14](#), we see that the uniform higher order dual pressure has a spike near the top left corner of the domain, and a subsequent pressure drop and rise along the y -direction. As the mesh is refined uniformly, the lower bound on the pressure will continue to drop, so the uniform pressure will not converge. When the mesh is adaptively refined, the higher order dual pressure is even more unstable, and there develop numerous large spikes along the boundary [Figure 6.15](#). In [Figure 6.16](#) and [Figure 6.17](#) are the corresponding results for extrapolation. This same phenomenon occurs for the extrapolated dual pressure, though on a smaller scale, as [Figure 6.17](#) illustrates. For the uniform extrapolated dual pressure, shown in [Figure 6.16](#), a spike develops at the top corner point. These dual pressures have proved ill-suited as dual solutions. They tend to dominate the error indicators and focus refinement near the goal boundary, the effect of which was clearly demonstrated in the figures.

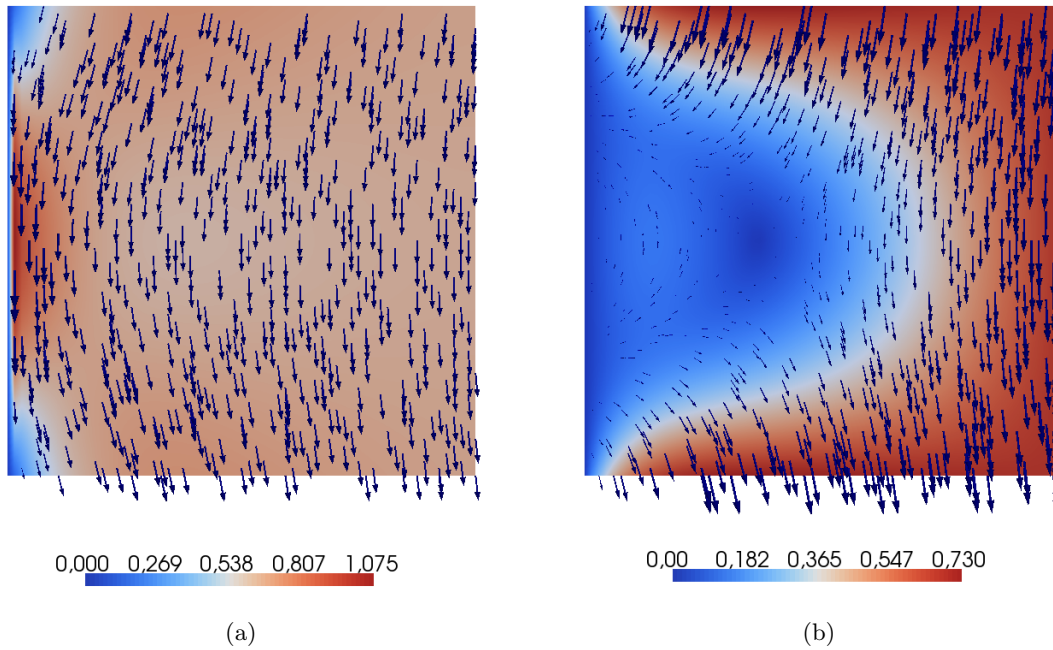


Figure 6.13: The dual velocity corresponding to shear stress with small interior support (to the left), and large interior support (to the right). Found on a uniform mesh with 2048 triangles.

Imposing the volume representation of the shear stress proved effective to improve the dual pressure. In [Figure 6.18](#) and [Figure 6.19](#) are shown, respectively, the higher order and extrapolated dual pressure on the final of a series of adaptive meshes computed using the minimal support volume formulation. With the formulation with larger support, results are similar.

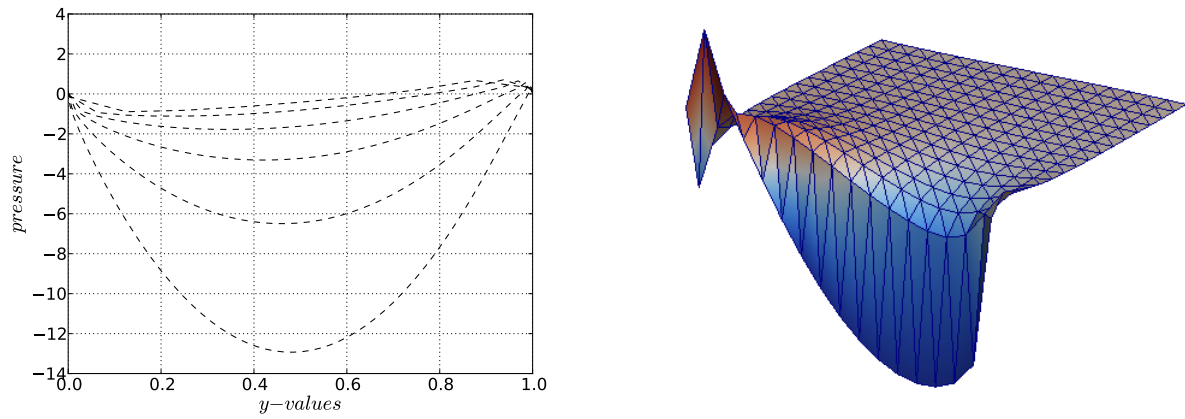


Figure 6.14: To the left values of the higher order dual pressure as a function of y along the Dirichlet boundary on a sequence of increasingly fine uniform meshes. To the right the higher order dual pressure on a 16×16 uniform mesh, scaled with a factor of 0.5. Computed with the *surface formulation* of shear stress.

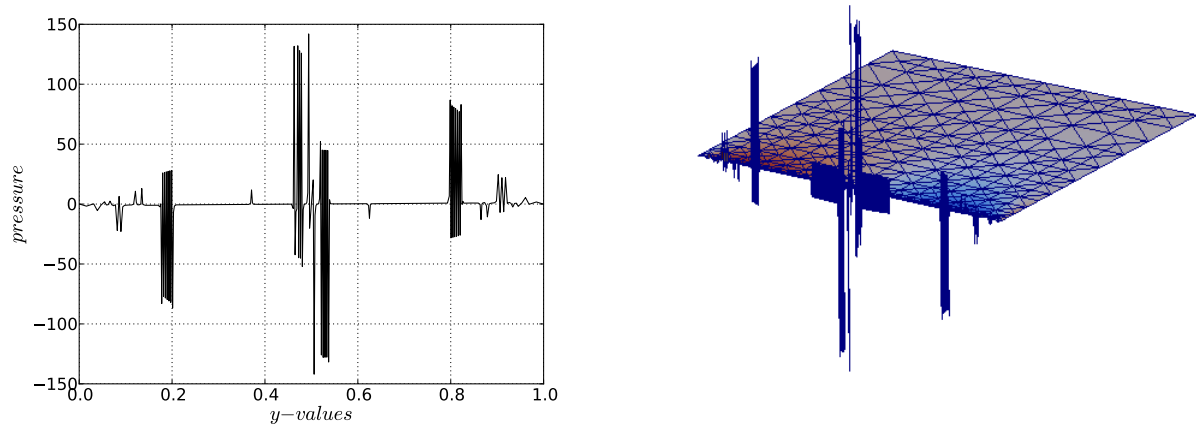


Figure 6.15: To the left values of the higher order dual pressure as a function of y along the Dirichlet boundary on an adaptive mesh with 3283 cells, obtained using Dörfler marking with fraction 0.5 and recursive bisection refinement. To the right the higher order dual pressure on the same mesh scaled with a factor of 0.005. Computed with the *surface formulation* of shear stress.

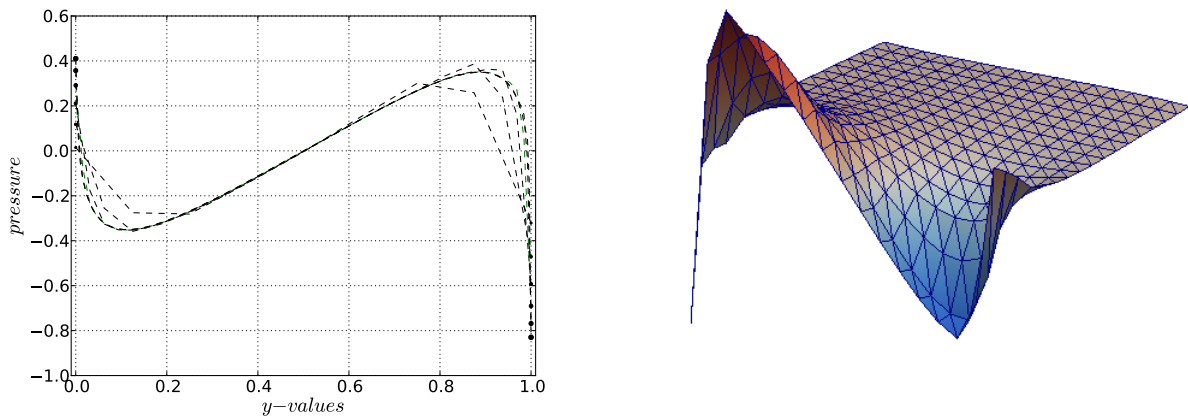


Figure 6.16: To the left the extrapolated dual pressure as a function of y along the Dirichlet boundary on a sequence of increasingly fine uniformly refined meshes. To the right the extrapolated dual pressure on a 16×16 uniform mesh, scaled with a factor of 0.9. Computed with the *surface formulation* of shear stress.

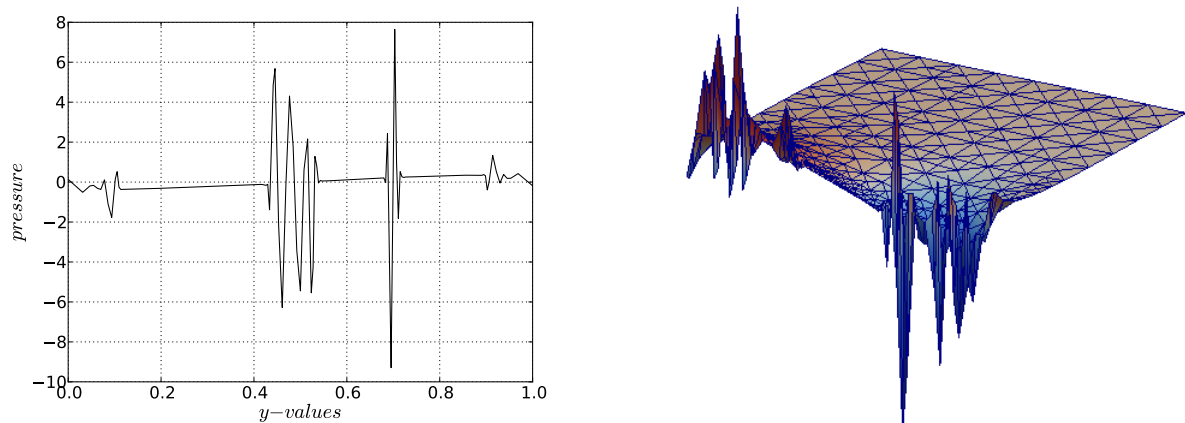


Figure 6.17: To the left the extrapolated dual pressure as a function of y along the Dirichlet boundary on an adaptive mesh with 2746 cells, obtained using Dörfler marking with fraction 0.5 and recursive bisection refinement. To the right the extrapolated dual pressure on the same mesh. Computed with the *surface formulation* of shear stress.

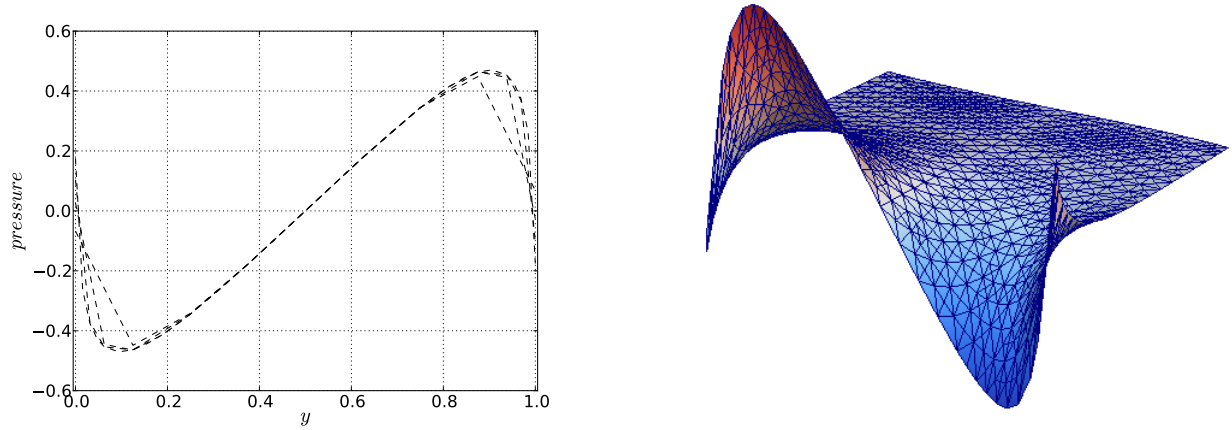


Figure 6.18: To the left the higher order dual pressure as a function of y along the Dirichlet boundary on a sequence of adaptive meshes obtained using Dörfler marking with fraction 0.5 and recursive bisection refinement. To the right the higher order dual pressure on the final mesh with 2100 cells, scaled with a factor of 0.9. Computed with the *volume formulation* of shear stress with minimal interior support.

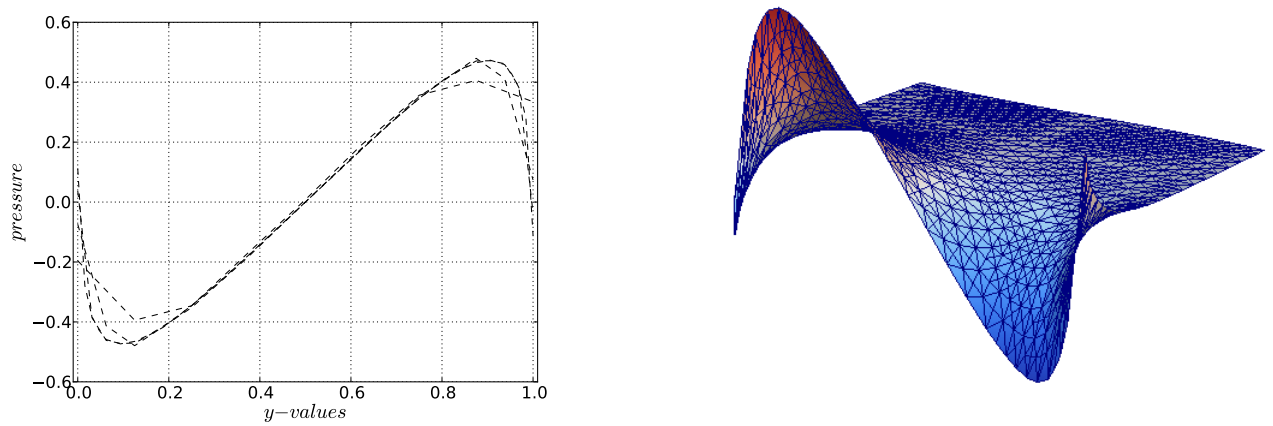


Figure 6.19: To the left the extrapolated dual pressure as a function of y along the Dirichlet boundary on a sequence of adaptive meshes obtained using Dörfler marking with fraction 0.5 and recursive bisection refinement. To the right the extrapolated dual pressure on the final mesh with 2558 cells, scaled with a factor of 0.9. Computed with the *volume formulation* of shear stress with minimal interior support.

Mesheres refined for the three shear stress formulations are shown in [Figure 6.20](#). Here a fixed fraction marking strategy with marking fraction 0.3 and recursive bisection refinement is applied. We see how the surface formulation tends to localize refinement to the goal boundary. The volume formulations yield a quite similar spread of refinement, just somewhat more focused near the goal boundary for the formulation with small support. [Figure 6.21](#) shows generated error indicators on a selected few of the refinement levels in the adaptive algorithm defined by the aforementioned adaptivity arguments. We observe that the surface formulation yields large error indicator values almost exclusively near the goal boundary, whereas the volume formulations yield a more evenly distributed error.

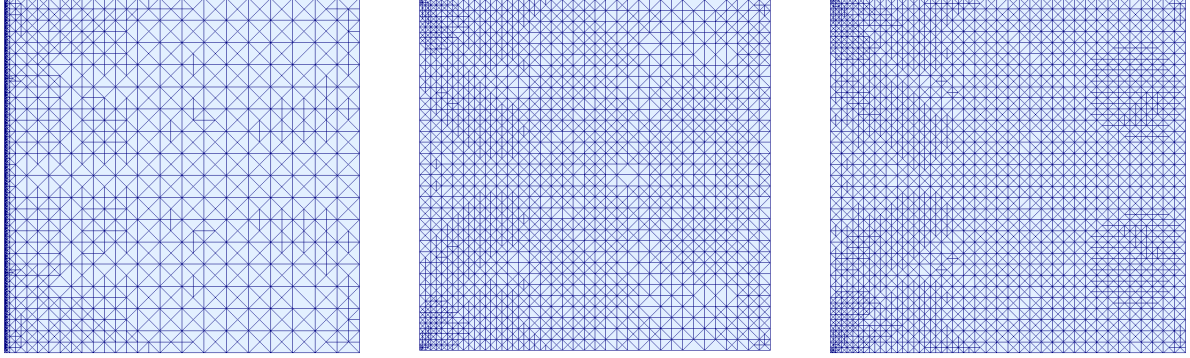


Figure 6.20: Meshes refined with respect to error-minimization in the (from left to right) surface shear stress, volume shear stress with small- and large interior support. Generated using a fixed fraction marking strategy with marking fraction 0.3 and recursive bisection refinement. The meshes consist of, respectively, 4206, 5214 and 5339 triangles.

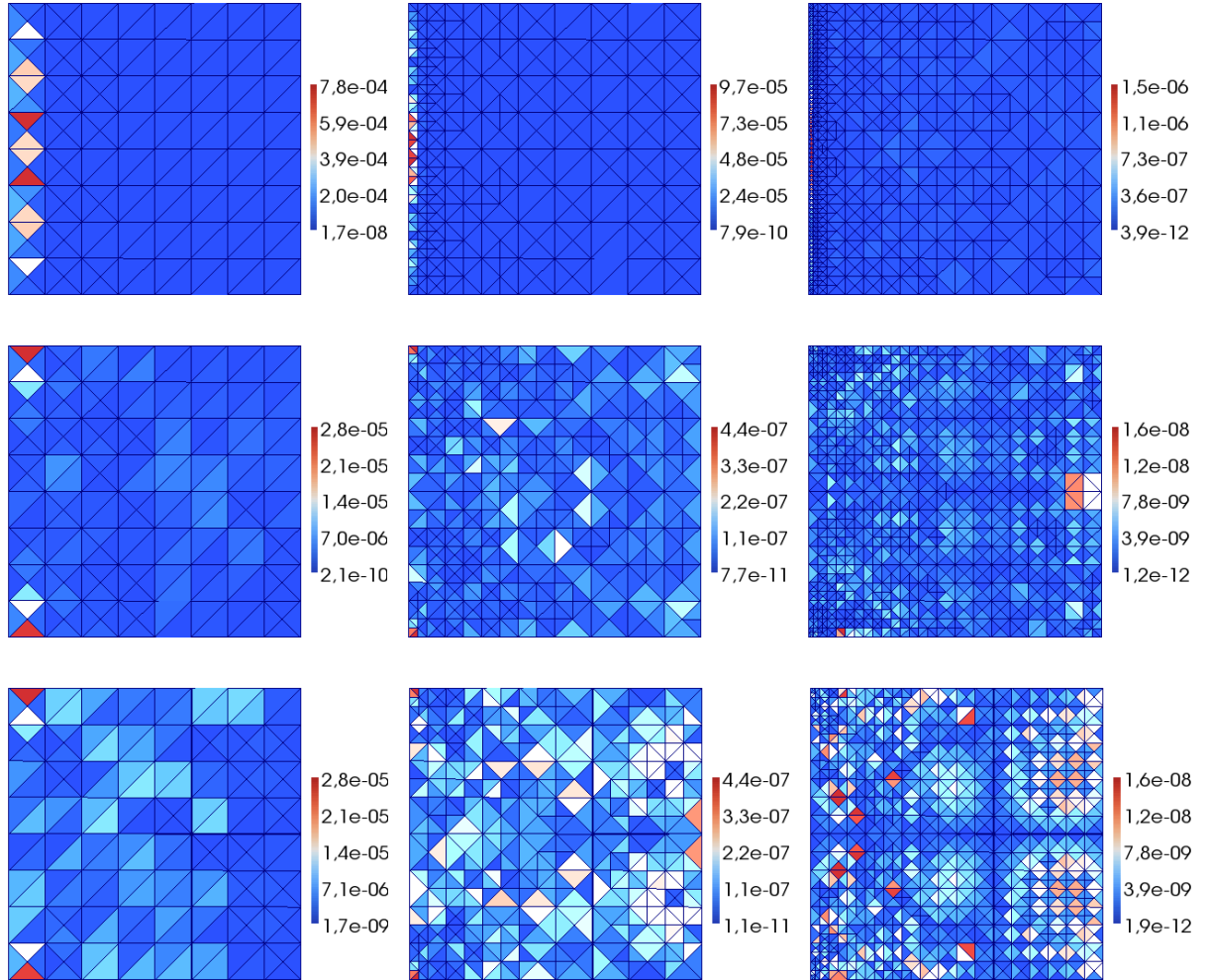


Figure 6.21: Error indicators in the (from left to right) first, fourth and seventh adaptive iteration using the (from top to bottom) surface shear formulation, volume shear formulation with, respectively, small- and large interior support. Using a fixed fraction marking strategy with fraction 0.3.

Adaptive performance

Using the higher order dual approximation yields indices between 0.5 and 2.0 for the volume formulation with small support, as can be seen in Figure 6.1.1. Some oscillations in the efficiency indices occur. When using the volume formulation with a large interior support, the generated efficiency indices are between 0.8 and 1.2. This can be seen in Figure 6.1.1. Some oscillations occur also here, but the results are more stable than for the volume shear formulation with small support. Extrapolated dual approximation only yields good index values when used with the shear stress formulation with large interior support. Otherwise the index values either explode or the error is extremely underestimated. The last drop in index value for adaptive refinement is caused by residuals evaluating to zero. In Table 6.6 are shown generated degrees of freedom, index values and errors in the goal functional on a series of uniformly and adaptively refined meshes. When employing an adaptive marking strategy about a fifth of the degrees of freedom were needed to reach the same accuracy as under uniform refinement, which is a significant improvement on a mesh of this size.

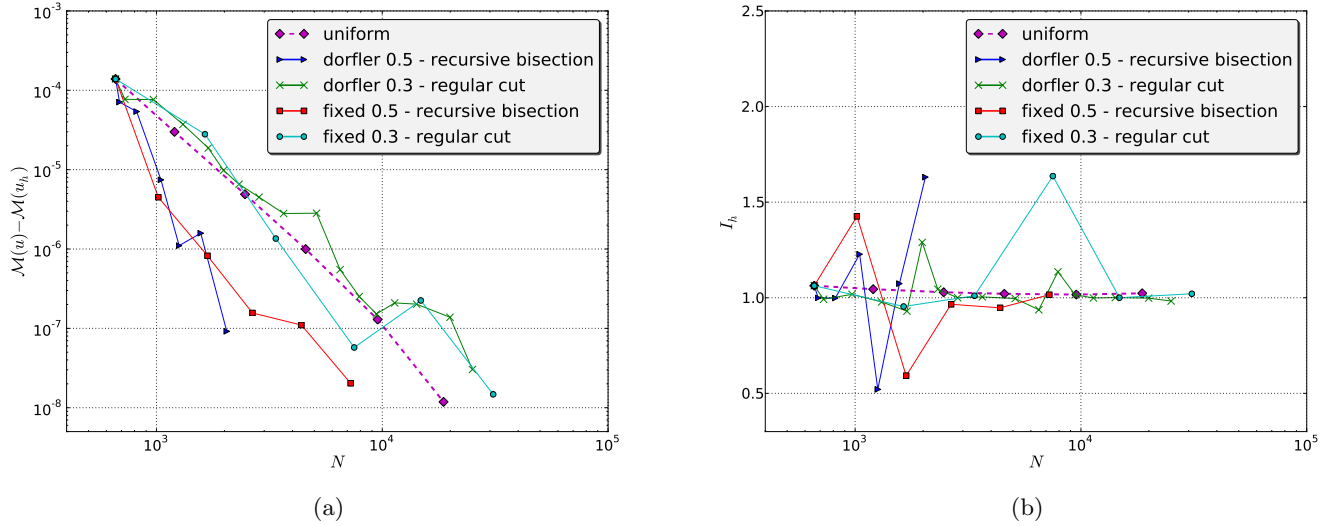
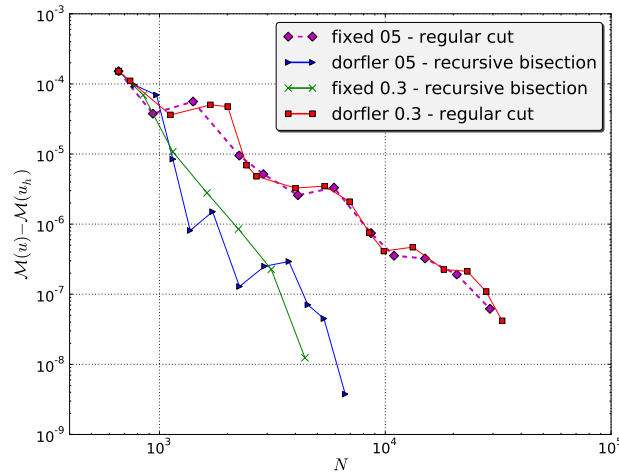


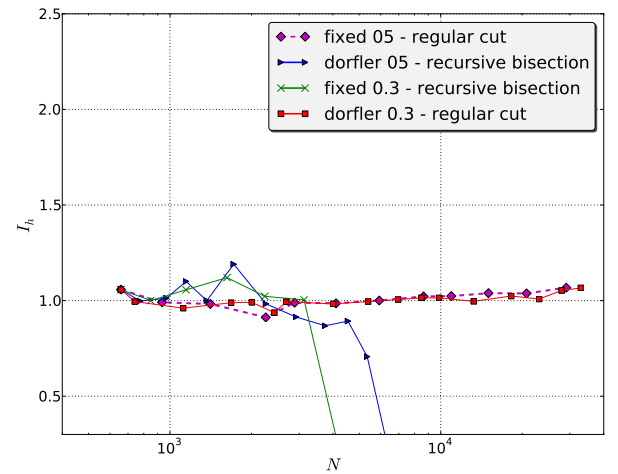
Figure 6.22: Error (left) and efficiency indices (right) for varying marking strategies and refinement strategies all with the optimal choice of marking fraction, for the volume formulation of shear stress with small support.

Uniform refinement			Adaptive refinement		
N	η	I_h	N	η	I_h
659	1.5167e-04	1.057560	659	1.5253e-01	1.057560
			776	9.6185e-05	0.997662
			970	6.9426e-05	1.012435
			1146	8.4471e-06	1.100444
			1368	8.1207e-07	1.000279
			1719	1.5027e-06	1.1905174
2467	5.7974e-06	1.024772	2262	1.2895e-07	0.9831263
			3743	2.9287e-07	0.8684147
			4543	7.0533e-08	0.8927639
			5350	4.4975e-08	0.8071269
			6649	3.7609e-09	0.1178824
9539	1.8801e-07	1.024159			
37507	1.4909e-09	0.971713			

Table 6.6: The Dörfler 0.5 recursive bisection error representation estimator dwr indicator. Last drop in index value for adaptive refinement caused by residuals evaluating to zero (residuals are on level 10^{-10})



(a)



(b)

Figure 6.23: Error (left) and efficiency indices (right) for varying marking strategies and refinement strategies all with the optimal choice of marking fraction, for the volume formulation of shear stress with large support.

Convergence of approximations

Higher order dual approximation yields approximately one order of magnitude more accurate dual approximation than extrapolation, (Figure 6.24) The reference solution is computed on a uniform mesh with approximately 32000 cells. So we have a primal solution with approximately 149000 degrees of freedom and higher order dual solutions with approximately 360000 degrees of freedom.

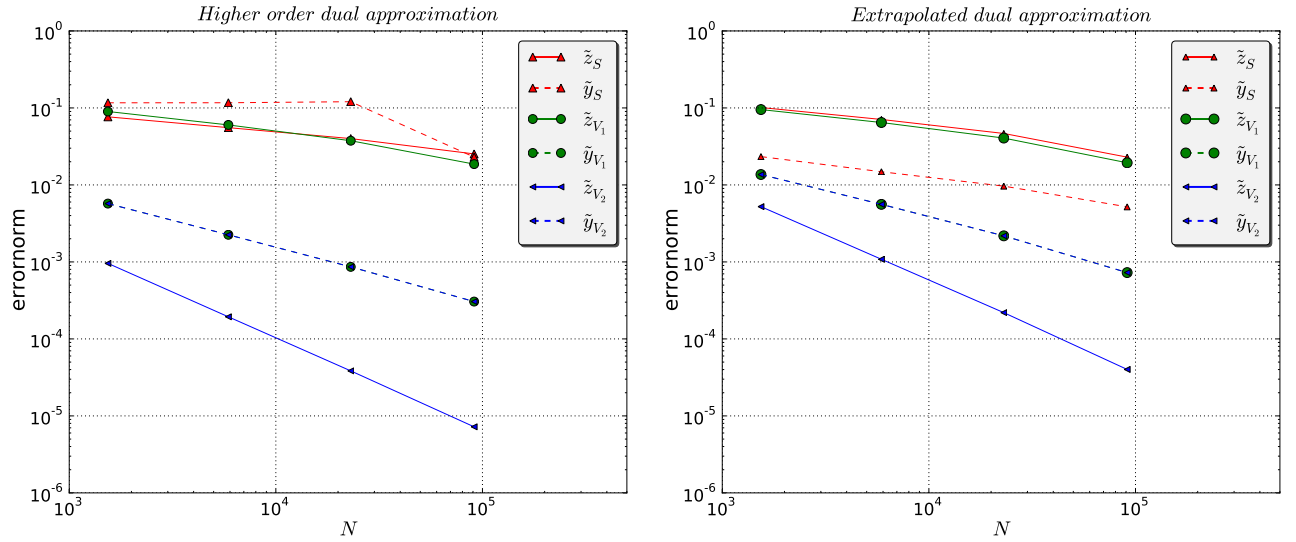


Figure 6.24: L^2 -error of the dual approximations under uniform refinement. \tilde{z} denotes the dual velocity, \tilde{y} denotes the dual pressure. The subscripts S , V_1 and V_2 indicate, respectively, the surface formulation and the volume formulation with small- and large interior support.

6.1.2 Discussion first test case

For test case I with the manufactured goal, efficiency indices are all very close to 1.0, showing excellent estimation properties. The approach thus seems robust for smooth primal and dual solutions.

For smooth primal on unit square, the volume-formulation provides what seems to be excellent dual solutions. The large versus small interior support gave large differences in the dual velocity, but no visible differences in the dual pressure.

The shear stress formulated by surface integrals yields a highly unstable dual pressure along this goal boundary regardless of method for refinement, but especially when adaptive refinement is utilized. This was clearly illustrated in Figures 6.14, 6.15, 6.16 and 6.17 where the generated higher order dual pressures were plotted for both uniform and adaptive mesh refinement. Despite the unstable higher order approximated dual pressure, the quality of the error estimate is very good, and higher order generated efficiency indices between 0.85 and 1.6, provided we employ a refinement method that assures a somewhat spread refinement. The large value spikes that develop for the dual pressure will dominate the error indicators, and thus focus the refinement to the goal boundary. We reach the same accuracy in $\mathcal{M}(e)$ as when applying uniform refinement using a lot fewer degrees of freedom (10000 vs 100000). When using the extrapolated dual approximation, the error is highly underestimated, with

efficiency indices that approach 0.2.

The representation of the goal functional as volume integrals offers a considerable improvement for the stability of the dual pressure, which is illustrated in [Figure 6.18](#). Using volume formulation with \mathbf{v}_d we obtain very good results with a higher order dual approximation. The dual pressure is nice and smooth regardless of the choice of support and order of approximation of \mathbf{v}_d , and the efficiency indices are very good. For the extrapolation dual approximation on the other hand, good results are not as easily obtainable. With \mathbf{v}_d with a large support, and with a sufficiently aggressive refinement approach, we can achieve quite good indices, but results are highly sensitive to choice of parameters. Thus extrapolation is all in all a very unstable choice of dual approximation with respect to computation of shear stress. Higher order is a much more stable dual approximation strategy.

Adaptive refinement proved very effective in terms of accurate computation of shear stress at a considerably lower computational cost. The extrapolation dual approximation method proved unstable and very sensitive to parameter choices, whereas the higher order dual approximation demonstrated the opposite. Good index values could be generated for all shear stress formulations.

For computation of shear stress the cell-facet-split and dual-weighted-residual estimators overestimate a great deal regardless of representation, and are therefore not options as error estimators. We must rather use the error representation estimator in further computations of shear stress.

6.2 TestCase II

The fluid velocity and pressure on the simplified aneurysm are shown in [figure Figure 6.25](#). Note that the switched sign of the pressure in (2.2) yields a counter-intuitive initial negative pressure at the inlet and a rise in pressure across the domain until it reaches its maximum at the outlet. On this test case we consider adaptive error control in the wall shear stress goal functionals \mathcal{M}_S , \mathcal{M}_{V_1} and \mathcal{M}_{V_1} along the surface of the bulge modelling the aneurysm.

Dual solutions and adaptive meshes

The dual solution generated by the shear stress along the aneurysm boundary works to accelerate the dual velocity along the inside of the aneurysm causing a rotational anti-clockwise movement. In [Figure 6.26](#) is shown the dual velocity corresponding to all shear stress formulations, computed on a coarse uniform mesh. For the goal functional with a small support, the velocity follows the rotational movement along the aneurysm boundary. The dual velocity generated by the goal functional with large support, does not seem to follow this pattern, but flows in one direction across the domain.

The dual pressure generated by all shear stress formulations, computed on the initial uniform mesh with 2483 triangles, are shown in [Figure 6.27](#). The dual pressure generated by either of the formulations with small support, that is \mathcal{M}_S and \mathcal{M}_{V_1} , are unstable, and yield oscillations along the goal boundary even at the initial coarse mesh. When applying the volume formulation with large support on the other hand, the generated dual pressure is smooth along the goal boundary. Since the quantities are so small, we have scaled the

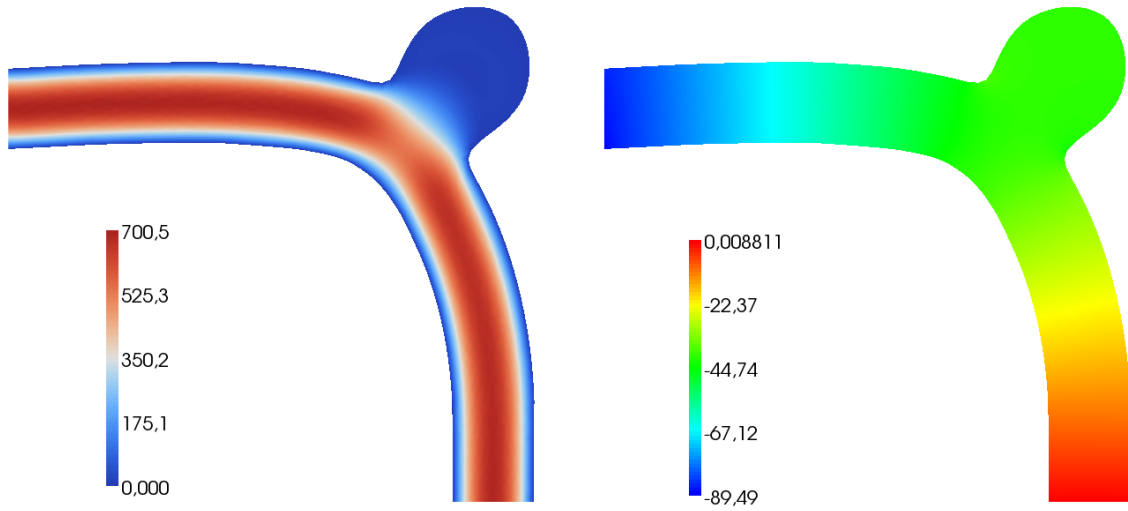


Figure 6.25: Fluid velocity (left) and pressure (right) on the idealized aneurysm. Computed on a uniform mesh with 2438 triangles.

pressure with a factor of 20 to emphasize the smooth features.

In figure [Figure 6.28](#) are shown adaptively obtained dual pressures generated by all three shear stress formulations using a Dörfler marking strategy with marking fraction 0.5 and the regular cut refinement algorithm. We observe that both formulations with small support yield dual pressures with numerous spikes along across the aneurysm boundary, whereas the volume formulation yields a much smoother dual pressure. When we enlarge the features around the aneurysm, however, we see that some numerical instabilities occur also for the volume formulation with a large support.

The shear stress goal functional with small and large support yield quite different adaptive meshes. This is illustrated in [Figure 6.29](#), where we show adaptive meshes generated for the volume formulation with small and large support, obtained using a Dörfler marking strategy with marking fraction 0.5 and regular cut refinement. We observe how a small support tends to yield a more localized refinement.

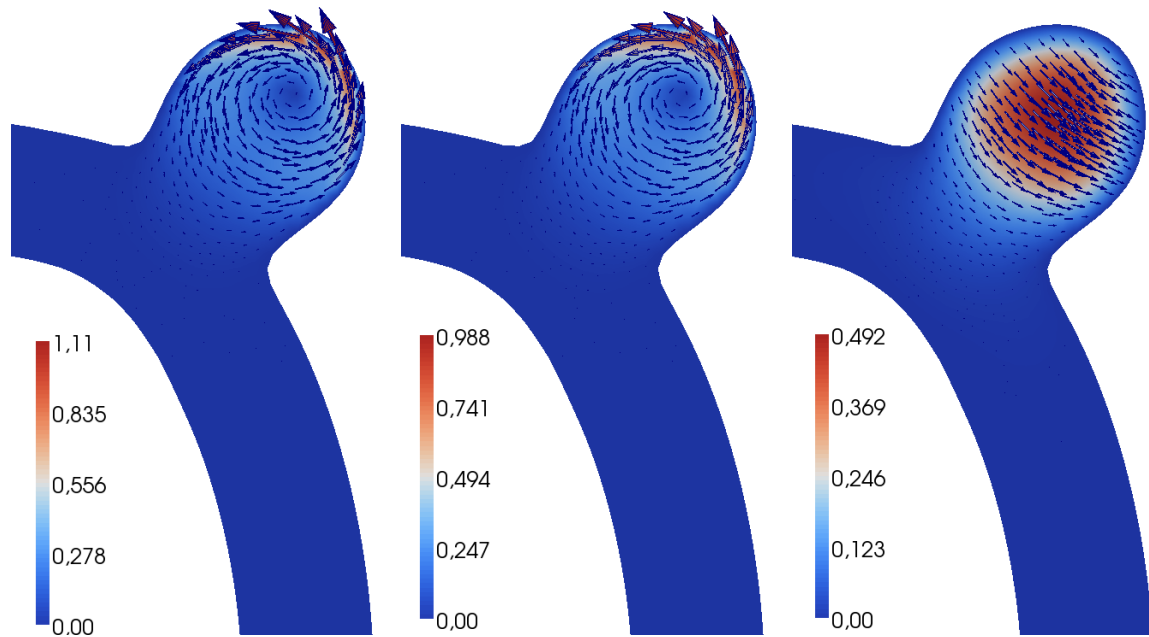


Figure 6.26: Dual velocity for the (from left to right) surface and volume with small and large support shear stress. Computed on the initial, coarse, mesh with 2483 triangles.

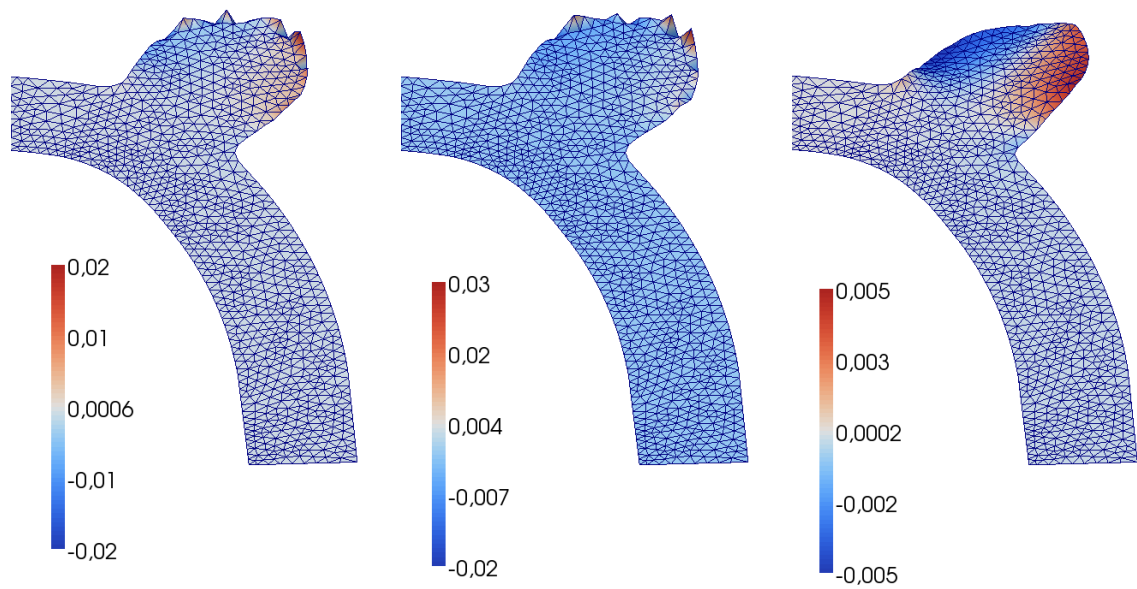


Figure 6.27: Dual pressure for the (from left to right) surface and volume with small and large support shear stress. Computed on the initial, coarse, mesh with 2483 triangles.

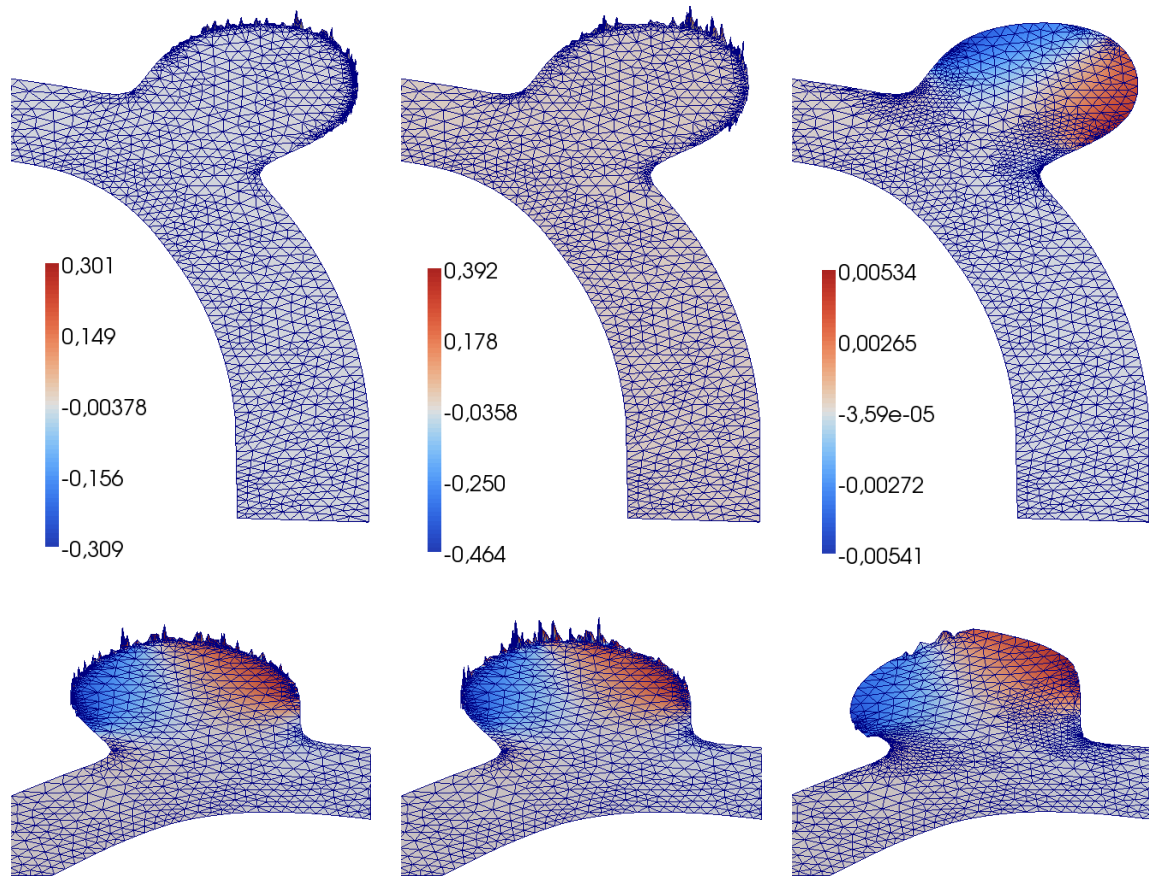


Figure 6.28: Dual pressure for the (from left to right) surface and volume with small and large support shear stress. Computed on adaptive meshes generated with Dörfler marking with marking fraction 0.5 and regular cut refinement. Meshes have, respectively, 3726, 2568 and 3589 cells. Below zoomed in on the aneurysm.

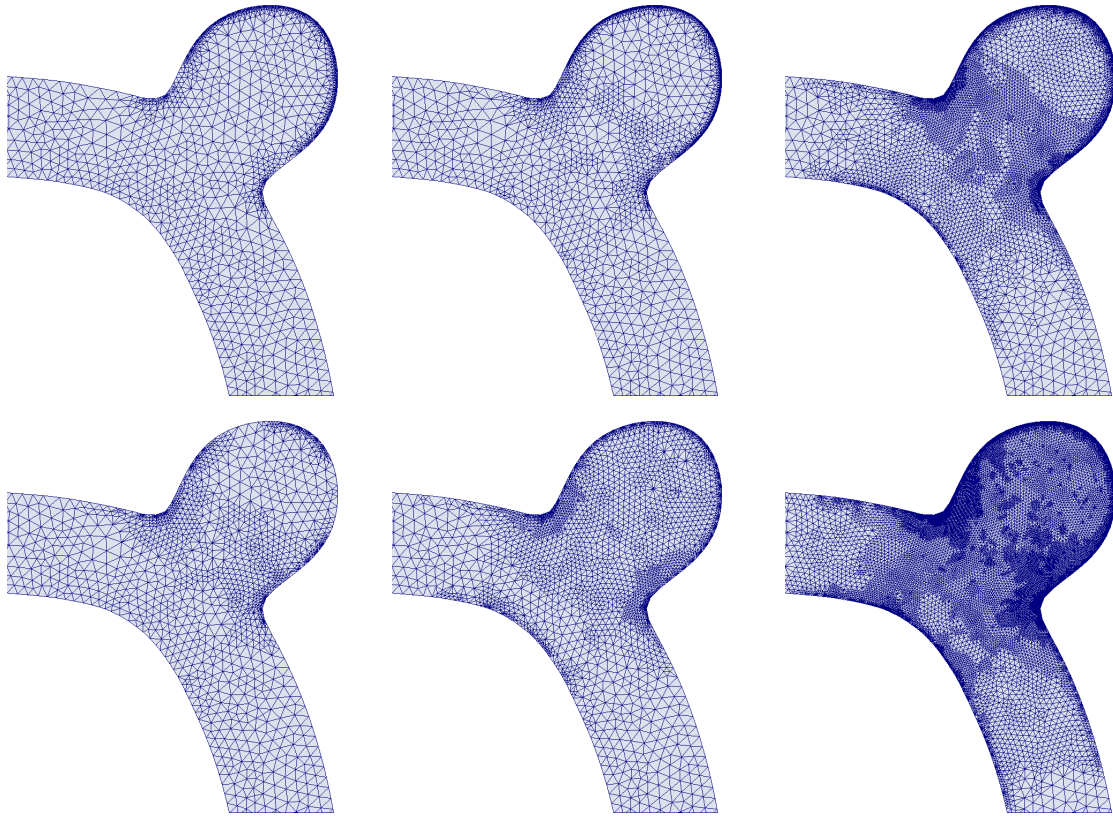


Figure 6.29: Adaptive meshes obtained with respect to the volume formulation with small support (top) and large support (bottom). Meshes have approximately (from left to right) 4500, 8500 and 45000 cells, and are obtained using a Dörfler marking strategy with marking fraction 0.5 and regular cut refinement.

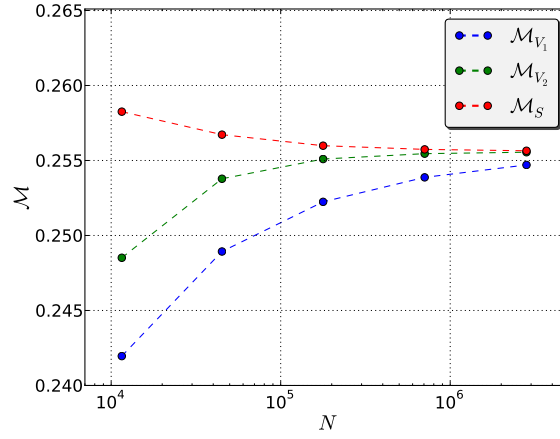


Figure 6.30: Values of the shear stress goal functionals under uniform refinement.

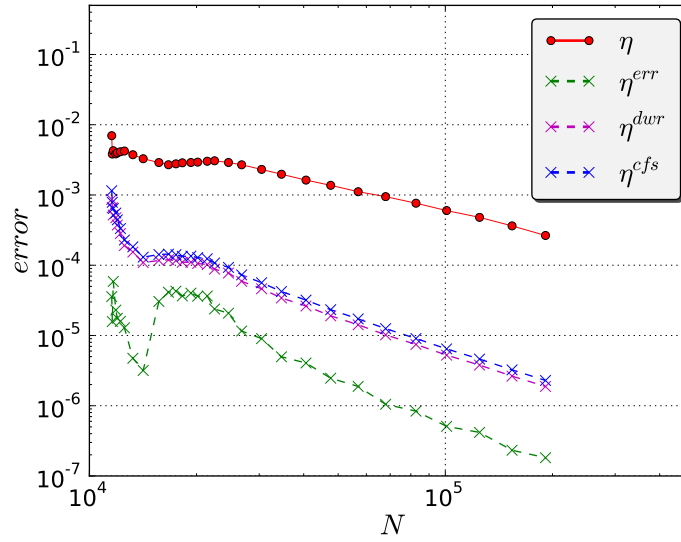


Figure 6.31: Error in the goal functional given by the volume formulation of shear stress using different error estimators. Computed using a Dörfler marking with marking fraction 0.6 and recursive bisection refinement.

Discussion test case II

As we observed in the first test case, the shear stress with small and large support give rise to very different dual velocities. Both the surface formulation and the volume formulation with small support yield a highly unstable dual pressure. More stable dual pressures could be obtained using the volume formulation with large support, but some instabilities were still observed. [Figure 6.31](#) shows that the adaptive refinement did not perform well in terms of generating good quality error estimates. The error was very underestimated. Estimates were somewhat improved by employing estimators that by construction give larger error estimates, but the error was still very underestimated.

Implementing the volume formulation on the aneurysm mesh was more complicated on account of the difficulties with correctly representing the tangent function. As discussed in [Section 4.2.2](#), we want to implement \mathbf{v}_d as a $[CG_1]^d$ function. This presented no problems in the first test case. The reference values for the three formulations of the shear stress were all zero, and generated values of the goal functionals for both uniform and adaptive refinement approached zero.

The goal boundary on the unit square, namely the part of the boundary on which we compute the shear stress, is in $x = 0$, so the tangent vector points in the same direction across all goal boundary facets. On the aneurysm, however, the goal boundary is over the aneurysm, which causes the direction of the tangent vector to vary with the facets. An approximation error is thus introduced due to the fact that we do not correctly represent the tangent vector on the facets. If \mathbf{v}_d is $[CG_1]^d$, for example, the tangent will vary linearly across a facet from the value in first degree of freedom to the value in the second degree of freedom on the facet. We do thus not know what to expect when we use a CG_1 function to represent \mathbf{v}_d . In the numerical testing, we found that, when the tangent was represented as $[CG_1]^d$, the goal functional value did not converge towards any value. In [Figure 5.7](#) we show \mathbf{v}_d implemented as $[CG_1]^d$ with small and large support.

A way to correctly represent the tangent vector is to represent it as a discontinuous function. It requires a discontinuous function to correctly represent the tangent. But This conflict with the requirement that we must be able to apply the gradient on \mathbf{v}_d .

We have tried many different approaches: representing v_d as a CG_1 function did not work. Representing v_d as a DG_0 function did not work either, since we could not apply the gradient. Solving a pde only on the boundary did not work because it was tremendously time-consuming.

What gave the best results, was using the Crouzieux-Raviart element space to represent \mathbf{v}_d . The Crouzieux-Raviart element has evaluation at the midpoint of each edge. The function space made up of Crouzieux-elements is continuous only at edge midpoints. The Crouzieux-Raviart element space, denoted CR, is not H^1 conforming, but allows for application of the gradient, as it is continuous across edges. It thus both represents the tangent vector in a good way since it is discontinuous, and fulfils the requirement that we can apply the gradient. The generated values for all shear stress formulations seemed to converge towards the same value. It was the only tested option, with which we were able to produce a stable dual pressure. This occurred, however, only on a uniform mesh, and of a certain cell size. Under both uniform and adaptive meshes, did the pressure become unstable very quickly. There is no good way of visualizing \mathbf{v}_d when it is CR, since values are set on the edges, and not in the mesh points. [Figure 6.30](#) illustrates that the goal functional values for

the three shear stress formulations converge towards the same value when the CR function space is applied.

Chapter 7

Conclusion and future work

By applying error control in a manufactured goal functional on the unit square, we were able to test the adaptive approach for a case with smooth primal and dual solutions. The generated efficiency indices were all very close to 1.0, showing excellent estimation properties. The approach seems robust for smooth primal and dual solutions.

Applying adaptive error control in wall shear stress, even in this basic test case, proved more challenging. The generated dual solutions were very sensitive to the representation of the shear stress goal functional. When the shear stress was represented in the terms of surface integrals, the dual pressure was of a highly unstable nature, showing large spikes along the boundary of interest. This is not ideal in terms of error indication, since the dual pressure will then dominate the refinement indicators. This will in turn cause refinement to occur almost exclusively near the goal boundary. This applied for both the higher order and the extrapolated dual approximation induced by the surface representation of shear stress.

By instead applying the volume representation of the shear stress, we obtained improved dual pressures for both the higher order and the extrapolated dual approximation strategy. The higher order dual approximation showed good error estimation properties for all three shear stress formulations in terms of quality of the error estimates. The required accuracy in the goal functional was reached at about a third of the degrees of freedom as when the accuracy was met on a uniform mesh. The extrapolation approximation strategy was far less stable. For both the surface formulation and the volume formulation with small interior support acceptable efficiency indices could not be obtained. The error would generally be highly underestimated, and generate significant oscillations in the efficiency indices. When employing the volume formulation with large interior support, the error estimates were slightly improved, but still very sensitive to the employed marking strategy.

Of the techniques for error control in shear stress, the volume formulation with large interior support generated the most stable efficiency indices. Under uniform refinement, this formulation demonstrated better convergence properties for the induced dual approximations.

For the second test case, the results were much less satisfactory. Instabilities were observed in the dual pressure regardless of the choice of goal functional formulation although slight improvements were demonstrated for the volume formulation with large support.

The work in this thesis serves as an initial investigation of these techniques for applying adaptive error control in the shear stress induced by fluid flow. Producing a stable dual pressure proved challenging. The stability of the dual solutions is required to correctly

illustrate the the dependence of the global goal functional error on local discretization errors. Applying the volume representation of the shear stress goal functional showed potential in terms of producing a stable dual pressure. The approximation error that resulted from the conversion to volume-integrals seemed to be small, since the produced goal functional values converged towards the same values. This error could be expected to grow when adding a convection term to the volume formulation. For further conclusions to be drawn, the technique needs to be investigated on more complex domains, both 2D and 3D test problems incorporating both time and convection. Efforts should be made towards finding a good and robust way of representing the tangent vector and reliably computing the dual solution in general.

Appendix A

Code

In this appendix a few code snippets are included. See captions for brief explanation of each.

A.1 The shear stress goal functional

Python code

```
def surface_goal(self, W, x=None):

    if x == None:
        x = Function(W)
        (u, p) = split(x)

    # normal and tangential vectors
    n = FacetNormal(W.mesh())
    t = as_vector((n[1], -n[0]))

    # smoothing function
    h = Expression("4*x[1]*(1-x[1])", degree = 2)

    # smoothed shear stress
    m = h*inner(self.sigma(u, p, self.mu)*n, t)*ds(1)

    return m, x
```

Figure A.1: Implementation of the surface formulation of the shear stress goal functional on the unit square.

Python code

```

def volume_goal(self, goal, W, x=None):

    if x == None:
        x = Function(W)
        (u, p) = split(x)

    vd = generate_vd(W)
    if goal == "volume_shear2":
        v_d = add_support(v_d)

    # volume formulation of shear stress goal functional
    M = inner(self.sigma(u, p, self.mu), grad(v_d))*dx\
        - inner(self.f_ex, v_d)*dx

    return M, x

```

Figure A.2: Implementation of the volume formulation of the shrear stress goal functional. Implementation applies to both the unit square and the idealized aneurysm. The implementation of the function \mathbf{v}_d is applied to each case.

Python code

```

if goal == "volume_shear1":

    V = VectorFunctionSpace(W.mesh(), "CG", 2)
    v_d = Function(V)

    # smoothing
    h = Expression("4*x[1]*(1-x[1])", degree=2)

    bc = DirichletBC(V.sub(1), h, Goal())
    bc.apply(v_d.vector())

elif goal == "volume_shear2":

    V = VectorFunctionSpace(W.mesh(), "CG", 3)

    # smoothing
    h = Expression(("0.0", "4*x[1]*(1-x[1])*(1-x[0])"), degree=3)
    v_d = project(h, V)

```

Figure A.3: Implementation of the function \mathbf{v}_d for the volume formulations with small and large support on the unit square.

Python code

```
def translate(x, y):
    ''' return translated and rotated point '''
    phi = 0.6799989473909490725093281462
    x -= 1.974610
    y += 0.552720
    x_new = x*cos(phi) - y*sin(phi)
    y_new = y*cos(phi) + x*sin(phi)
    return y_new
```

Figure A.4: Method that, given mesh coordinates x and y (for points on the aneurysm), returns the rotated and translated coordinates. Needed in order to introduce smoothing based solely on the x -coordinates.

Python code

```
class Smoothing(Expression):
    def __init__(self, mesh):
        self.mesh = mesh

    def eval(self, values, x):
        g = translate(x[0], x[1])/2.43404902569284065095
        values[0] = g
\begin{python}
\caption{Smoothing function subclassing Expression.}
\end{figure}

\begin{figure}[htbp]
\begin{python}

class TangentExpression(Expression):
    def __init__(self, mesh):
        self.mesh = mesh
    def eval_cell(self, values, x, ufc_cell):
        cell = Cell(self.mesh, ufc_cell.index)
        n = cell.normal(ufc_cell.local_facet)

        #smoothing
        g = translate(x[0], x[1])/2.43404902569284065095
        values[0] = g*n[1]
        values[1] = - g*n[0]
    def value_shape(self):
        return (2,)
```

Figure A.5: Implementation of \mathbf{v}_d with small support and embedded smoothing.

Python code

```

def add_support(self, V, v_d):
    print "adding support to v_d"

    bc = DirichletBC(V, v_d, "on_boundary")

    up = TrialFunction(V)
    vp = TestFunction(V)

    zero = Constant((0.0, 0.0))
    a = inner(grad(up), grad(vp))*dx
    L = inner(zero, vp)*dx

    v_d2 = Function(V)

    solve(a == L, v_d2, bc)
    return v_d2

```

Figure A.6: Add support to v_d by solving a Poisson problem.*Python code*

```

def shear_volume(self, W, x=None):

    goal = self.goal
    plot_ = True
    if x == None:
        x = Function(W)
        plot_ = False
    (u, p) = split(x)

    t = TangentExpression(W.mesh())

    V = VectorFunctionSpace(W.mesh(), "CR", 1)
    v_d = Function(V)

    bc = DirichletBC(V, t, Goal())
    bc.apply(v_d.vector())

    if goal == "volume_shear2":
        v_d = self.add_support(V, v_d)

    M = inner(self.sigma(u, p, self.mu), grad(v_d))*dx -
        inner(self.f_ex, v_d)*dx
    return M, x

```

Figure A.7: Implementation of the volume formulation for the shear stress on the idealized aneurysm.

Bibliography

- [1] FEniCS project. <http://www.fenicsproject.org/>.
- [2] Image of an abnormal blood vessel. <http://commons.wikimedia.org/wiki/File:Fibr.jpg>.
- [3] Paraview. <http://paraview.org/>.
- [4] Python programming language - official website. <http://www.python.org/>.
- [5] M. S. Alnes. Finite element simulations of blood flow in the circle of willis, 2006. <http://folk.uio.no/martinal/master/>.
- [6] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhuser, 2003.
- [7] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 2001.
- [8] D. Birchall. Computational fluid dynamics. *The British Journal of Radiology*, 8, 2009.
- [9] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 2007.
- [10] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *R.A.I.R.O*, 2:129–151, 1974.
- [11] L. C.Evans. *Partial Differential Equations*. the American Mathematical Society, 1998.
- [12] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, 4:105–158, 1995.
- [13] V. Feigin. Stroke epidemiology in the developing world. *The Lancet*, 365(9478):2160–2161, Year.
- [14] K. Johansen. Aneurysms. *Scientific American*, 247:110, 1982.
- [15] P. K. Kundu and I. M.Cohen. *Fluid Mechanics*. Academic Press, 4 edition, 2008.
- [16] H. P. Langtangen. *Computational Partial Differential Equations*. Springer-Verlag, Berlin Heidelberg, 2003.
- [17] H. P. Langtangen. *A Primer on Scientific Programming with Python*. Springer-Verlag Berling Heidelberg, 2009.
- [18] A. Logg. DOLFIN: Automated finite element computing. *ACM transactions on mathematical software*, 37:1–28, 2010.

- [19] A. Logg, K. A. Mardal, and G. N. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Springer Verlag, 2011.
- [20] D. Parker. *Fields, Flows and Waves An introduction to Continuum Models*. Springer, 2003.
- [21] W. D. rfler. A Convergent Adaptive Algorithm for Poisson’s Equation. *SIAM Journal on Numerical Analysis*, 33:1106–1124, 1996.
- [22] M. Rognes and A. Logg. Automated goal oriented error control I: Stationary variational problems. *Submitted to journal*, 2010.
- [23] W. Schievink. Intracranial Aneurysms (review article). *Journal*, 336(1):28, 1997.
- [24] K. Selim. *Adaptive Finite Element Methods for Fluid-Structure Interaction and Incompressible Flow*. PhD thesis, University of Oslo, 2011. <http://folk.uio.no/selim/>.
- [25] H. Sohr. *The Navier-Stokes Equations*. Birkhuser Verlag, 2001.
- [26] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1973.
- [27] J. Volker. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3d navier-stokes equations. *International Journal for Numerical Methods is Fluids*, pages 775–798, 2002.
- [28] J. Volker and G. Matthies. Higher order finite element discretizations in a benchmark problem for incompressible flows. *International Journal for Numerical Methods is Fluids*, 2001.
- [29] J. F. Wendt. *Computational Fluid Dynamics:An Introduction*. Springer-Verlag, 2009.