An Adaptive Finite Element Method for the Incompressible Navier–Stokes Equations on Time-dependent Domains

Roland Becker

November 16, 1995

Contents

Chapter 1

Introduction and Notations

In this thesis we develop an algorithm for the numerical solution of the incompressible Navier–Stokes equations on domains varying in time. Numerical algorithms for the Navier–Stokes equations are used for the simulation of gas and liquid flows. In many engineering and biomedical applications the equations are given on a domain with moving boundaries. Two examples for this, the flow in a water pump and the blood flow through a contracted artery are described below.

In addition to the difficulties inherent to the Navier–Stokes equations, the motion of the boundaries is a supplementary source of errors. We propose a new approach to the analysis of this error which leads to optimal results. In the same way as physics propose the mathematical equations as a reproduction of the laws the real flow is governed by and thereby admitting model errors, numerical mathematics propose a finite model accessible to computers as an approximation to the mathematical equations. The control of the resulting discretization error is the objective of adaptive algorithms. We propose a new strategy for the implementation of such an algorithm. The developed techniques ensure that the number of operations per unknown of the adaptive algorithm is comparable to the common case of uniformly refined meshes. We can therefore fully profit from the local structure of an adaptively refined mesh generated by an a posteriori error estimator.

In view of the water pump problem, we show the stability of certain discretizations of the Stokes equations on anisotropic meshes, in the sense that

the stability constant does not depend on the aspect ratio. Known results suggest a linear dependence on this parameter which is unacceptable in many cases.

For the discretization of the incompressible flow equations we use an adaptive finite element method which is based on the Galerkin least squares approach with (bi-,tri-)linear finite elements, see [24], [27] and [51]. This method leads to a systematic discretization of optimal order that simultaneously deals with the two difficulties of the incompressible Navier–Stokes equations, the velocity-pressure coupling and the dominance of the convection term for higher Reynolds numbers. In comparison to the pure Galerkin discretization, stability is gained through the addition of least squares terms. A proper weighting leads to a balance between pure Galerkin and stabilizing terms. In this way one gains stability without destroying accuracy. The method is inherently consistent in the sense that a strong solution of the partial differential equation satisfies the discrete equations.

The discrete equations are solved by a multigrid algorithm that uses the canonical grid transfer operators given by the finite element spaces. Some care has to be taken since the discrete bilinear forms vary from level to level due to the dependence of the stabilizing terms on the mesh parameter.

As the triangulation of the domain as well as the domain itself changes in each time-step, there is the need of an accurate time discretization. In contrast to the case of a fixed domain the shape functions (and not only the coefficients of the solution) depend on time, reflecting the fact that the equation is posed on a non-cylindrical region in space-time. Besides the development of an accurate time discretization, we analyze various difficulties that arise in the context of moving boundary value problems. This includes the construction of adaptive algorithms and the question of stability of the discretization on stretched meshes.

There are many situations for flow simulations where the known temporal variation of the domain has to be taken into account. We choose two representative examples which give raise for different problems. Our first example is the blood flow through an artery which is contracted by a muscle. This can be modeled by an incompressible channel flow where part of the wall is moving up and down (see [41]). The domain with a plot of the global flow

picture is shown in figure 1.1. At the inflow we describe a parabolic profile according to Poiseuille flow, whereas the outflow is modeled by a natural boundary condition. The second example is the flow in a water pump where

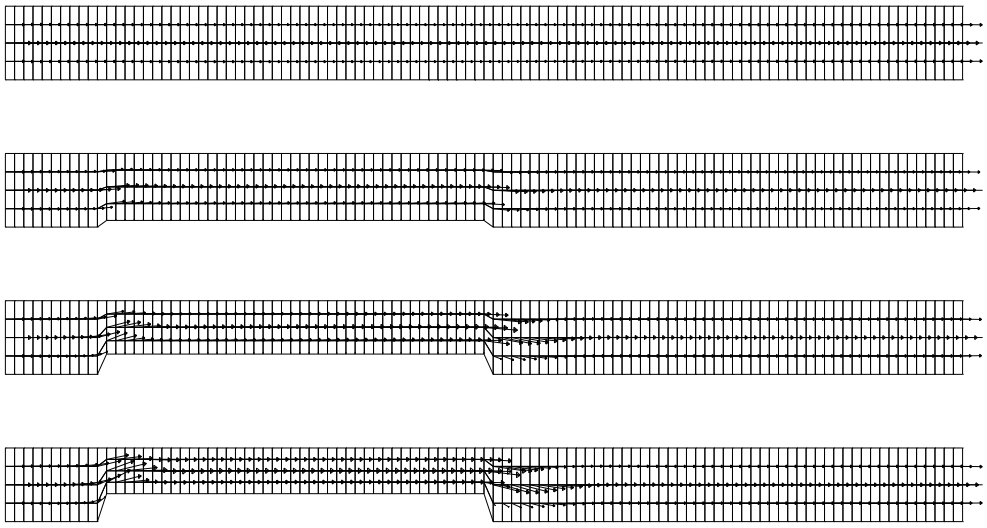

Figure 1.1: Flow in a contracted artery

a piston moves in a cylinder. A simple two-dimensional version of this pump is depicted in figure 1.2. Being driven by the movement of the piston the flow satisfies homogeneous Dirichlet boundary conditions on the left part of the boundary of the upper channel when the piston moves up and on the right part when the piston moves down. These boundary conditions simulate the action of the valves of the pump. The movement of the piston is described by a linear stretching with a factor periodic in time. Although the model is very simple it reflects two important difficulties. First, the mesh moving with the piston becomes very stretched when the piston arrives at the top. Second, there are strong singularities at the corners of the domain. They are most prominent at the points where the piston moves along the wall, because the boundary condition has a jump here.

As illustrated by the above examples, from the mathematical point of view there are several interesting questions which arise in the context of flow equa-

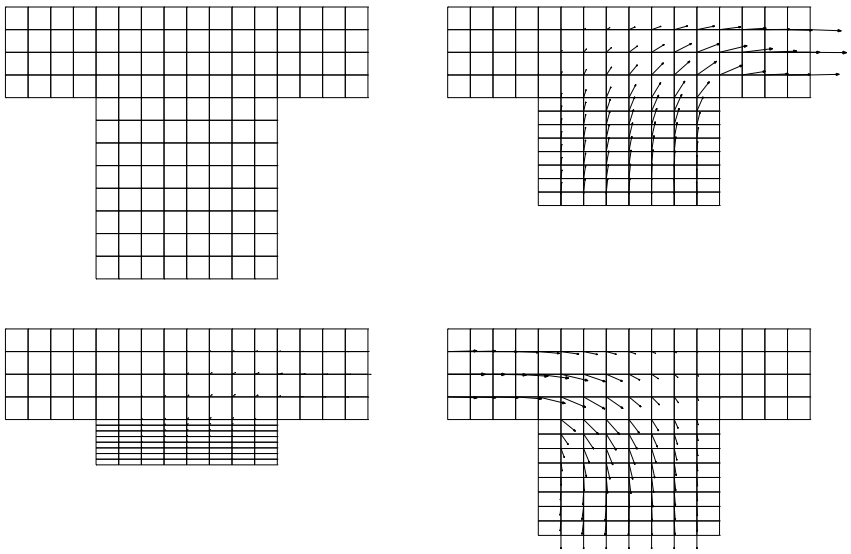

Figure 1.2: Flow in a water pump

tions with moving boundaries. First of all, the problem of an accurate temporal discretization appears. It is no longer sufficient to discretize the time derivative; the variation of the domain also has to be considered. One possible approach to this issue is the transformation of the variable domain onto a fixed one, introducing time-dependent coefficients in the transformed equations. This, however, requires the knowledge of a smooth global transformation, which might be difficult to obtain for complicated domains, if it exists at all. Further, it seems to be difficult to generalize this approach to situations where the movement of the boundary is not known in advance, as for nonstationary free boundary problems.

Therefore we describe an algorithm for the time discretization which works on the original problem. Under the assumption that a smooth transformation is given at least locally, we derive error estimates of optimal order. This is carried out in the case of the heat equation.

Another important problem which arises through the variability of the boundary is the triangulation of the domain. The usual finite element theory

supposes certain conditions on the triangulation, as for example the uniform shape condition (see the Notations), in order to derive stability of the discretization and to obtain optimal error estimates. As for the piston problem, the elements of the triangulation might become stretched as time evolves (see figure 1.2), therefore introducing a large constant in the quasi-uniformity condition. In order to deal with this difficulty we analyze the discretization of the Stokes equations on arbitrarily stretched meshes in the case of the Galerkin least squares method. We show that the right choice of stabilization leads to a stability constant which does not depend on the aspect ratio. We further derive an error estimate which separates the element extensions and the derivatives of the continuous solution according to the directions.
Because the movement of the domain introduces additional sources of errors in comparison to the case of a fixed domain and since the flow pattern may change considerably, the use of adaptive mesh refinement and de-refinement is an important feature for the applications under consideration. An example for this is the piston problem where we need a strong local refinement in the vicinity of the singularities. This will be demonstrated later.
Our approach to adaptivity is guided by the idea of efficiency. All components of the algorithm are designed in such a way that we obtain a similar work per cell as compared to a modern finite element code without adaptivity. In order to achieve this goal all components have to be carefully designed. There are three major ingredients for an adaptive algorithm.
First, a flexible grid handling is needed. Making use of the continuity of the finite element functions under consideration we describe a simple non-hierarchical discretization on meshes with hanging nodes. This avoids the complicated introduction of auxiliary matching elements.
Second, a reasonable error indicator is needed for the local mesh refinement. On the one hand, it should give an upper bound for the actual error to be able to lead to a reliable error control. On the other hand, it should be efficient in order to avoid over-refinement. We derive a residual based a posteriori error estimator for the Stokes equations following the general approach of Johnson et. al. (see [23]).
Third, in the context of locally refined meshes the question of the iterative solution of the discrete systems has to be considered. Although the number

of arithmetic operations decreases on adaptively refined meshes as compared to the case of global refinement the conditioning of the system is in general not better since it depends on the smallest element size in the mesh. We propose a multigrid algorithm which uses the hierarchy created by the adaptive grid refinement. Since all operations are done on the locally refined part of the meshes, it leads to an algorithm with optimal complexity for the solution of the discrete equations on a given adaptively refined grid if the spectral radius of the iteration matrix is independent of the number of levels. A proof of a level-independent bound for the error reduction is given by Bramble and Pasciak, see [13]. We therefore obtain an algorithm with overall optimal complexity provided that the refinement is efficient and the adaptive algorithm is close to being optimal in the sense that we obtain the final mesh within as few number of iterations as possible.

The combination of discretization and solution of the discrete equations makes it possible to develop a stopping criterion that allows to stop the iteration when the iteration error is in the order of the discretization error. Since this kind of estimators involves a dual problem, the stability of which has to be estimated in some sense, the case of the nonlinear Navier-Stokes equations is considerably more complicated (see [35]). We therefore do not discuss the question of estimating the stability constant. Although this excludes the possibility of quantitative error control, the principals of the adaptive algorithm are the same.

We demonstrate the effectiveness of our approach to adaptivity by some benchmark computations which point out the important gain which can be realized by adaptive algorithms.

The organization of this thesis is the following: In the second chapter we present our numerical results for the above mentioned moving boundary test problems. We further describe and analyze the discretization on variable domains. We define an oblique time derivative which allows us to apply the usual energy techniques in order to derive optimal order error estimators. The following chapter describes the adaptive algorithm together with the multigrid scheme on locally refined meshes. Chapter four develops an a posteriori error estimator for the Navier–Stokes equations. We also derive an a posteriori indicator for the iteration error. In chapter fife we proof stabil-

ity of the weighted least squares formulation with low order finite elements applied to the stationary Stokes equations. The final chapter describes the discretization of the incompressible Navier–Stokes equations. We also present a multigrid algorithm for the resulting discrete equations and apply the adaptive algorithm to a two-dimensional cylinder flow benchmark problem. The error indicator is based on a straightforward generalization of the a posteriori estimator for the Stokes equations given in chapter four.
The computations presented in this thesis are done with the finite element library DEAL.

Notations

Throughout this thesis we use the following notations:
$\Omega \subset \mathbb{R}^d$ denotes a bounded domain with Lipschitz boundary $\partial\Omega$ in two or three dimensions, $d = 2, 3$.
We use the usual notations for Lebesgue and Sobolev spaces, $L_p(\Omega)$ and $H^m_{(0)}(\Omega)$.
The triangulation of Ω is denoted by $\mathcal{T}$ and consists of a set of elements K with edges (or faces in three dimensions) Γ. The set of nodes N of the triangulation is denoted by $\mathcal{N}$. The diameter of K and the the diameter of its biggest inscribed ball are denoted by h_K and ρ_K respectively. The following quasi-uniformity condition is usually required to be fulfilled with a moderate constant (see [19]):

$$\frac{h_K}{\sigma_K} \leq C \quad \forall K \in \mathcal{T}. \tag{1.1}$$

We will weaken this constant in order to allow the use of anisotropic meshes which lead to a big constant C in (1.1).
The finite element spaces which are considered in this thesis are either the space of continuous element-wise linear functions or the space of element-wise constant functions, for example on a triangular mesh $\mathcal{T}$ in the first case

$$V_h = \{v_h : v_h|K \in P^1(K)\ \forall K \in \mathcal{T} \text{ and } v_h \text{ is continuous}\},$$

where $P^1(K)$ denotes the space of linear functions on K. Abusing the notation, we will shortly write $V_h = P^1$. On quadrilateral elements we use the usual isoparametric finite element functions, see [19].
At several places we will use an interpolation operator I_h. Possible choices are nodal interpolation (for continuous functions) or L_2-projection. They have the following interpolation properties:

$$\|\nabla\,(u - I_h u)\| \leq h\, C_I \|\nabla^2 u\|, \qquad \text{and} \tag{1.2}$$

$$\|u - I_h u_h\| \leq h^\alpha C_I \|\nabla^\alpha u\|, \quad \alpha = 0, 1. \tag{1.3}$$

The (different) constants C_I depend on the mesh constant C in (1.1).

Chapter 2

Moving Boundary Problems

In this chapter we present the examples of fluid problems involving time-dependent domain described in the introduction.
We further analyze the method which is used to discretize the equations in time. Since the difficulties arising from the movement of the mesh are not related to the Navier-Stokes equations, we consider an elliptic model problem. It is shown that the order of classical one-step methods as the Crank-Nicholson scheme is maintained under certain regularity assumptions on the movement. The movement of the domain is expressed by a transformation on a fixed region. Instead of working on the fixed domain, we develop a discretization on the deforming domain. The advantage of this approach is the possible generalization to the case where the transformation is given locally from time-step to time-step and to space-time elements. The analysis presented here, however, requires a regularity assumption on the transformation.
The algorithm is illustrated for the examples considered in the first sections.

2.1 Flow in a Channel with a Moving Indentation

We investigate a two-dimensional unsteady flow of an incompressible fluid in a channel with a time-dependent indentation in one wall. The initial condition for the velocity is steady Poiseuille flow and the indentation moves in and out sinusoidally. The geometry of the flow domain $\Omega(t)$ is shown in figure 2.1. The starting domain is $\Omega(0) = [-8, 18] \times [0, 1]$. The shape of the

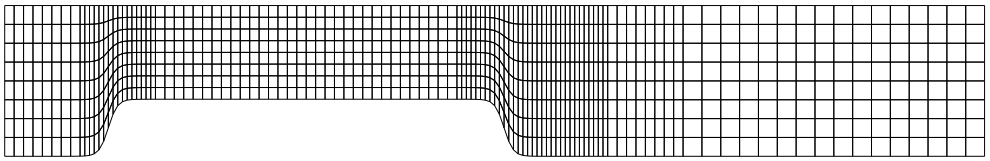

Figure 2.1: Geometry of the channel (scaled in y by a factor 4)

indentation which is centered in $x = 0$ is taken from Pedley and Stephanoff (see [41]). The height of the moving bottom wall is given by

$$y(x) = \begin{cases} 0 & \text{for } -8 < x < -x_3 \\ 0.5h(t)(1 + \tanh(a(x + x_2))) & \text{for } -x_3 < x < -x_1 \\ h(t) & \text{for } -x_1 < x < x_1 \\ 0.5h(t)(1 - \tanh(a(x - x_2))) & \text{for } x_1 < x < x_3 \\ 0 & \text{for } x_3 < x < 18 \end{cases}, \tag{2.1}$$

where $x_1 = 4$, $x_3 = 6.5$, $x_2 = 0.5(x_1 + x_3)$ and $a = 4.14$. The function $h(t)$ describing the temporal variation of the indentation is given by:

$$h(t) = 0.5\epsilon\,(1 - \cos(2\pi\frac{t - t_0}{T})),$$

with ϵ specifying the maximum blockage of the channel cross section and T indicating the oscillation period. The governing parameters are Reynolds number Re, Strouhal number St and amplitude parameter ϵ. For given viscosity ν, maximal inflow velocity u_0 and channel width L, these parameters are defined by:

$$Re = \frac{u_0 L}{\nu} \qquad St = \frac{L}{u_0 T}.$$

St is a frequency number that measures the speed of the wall movement relative to the mean velocity.

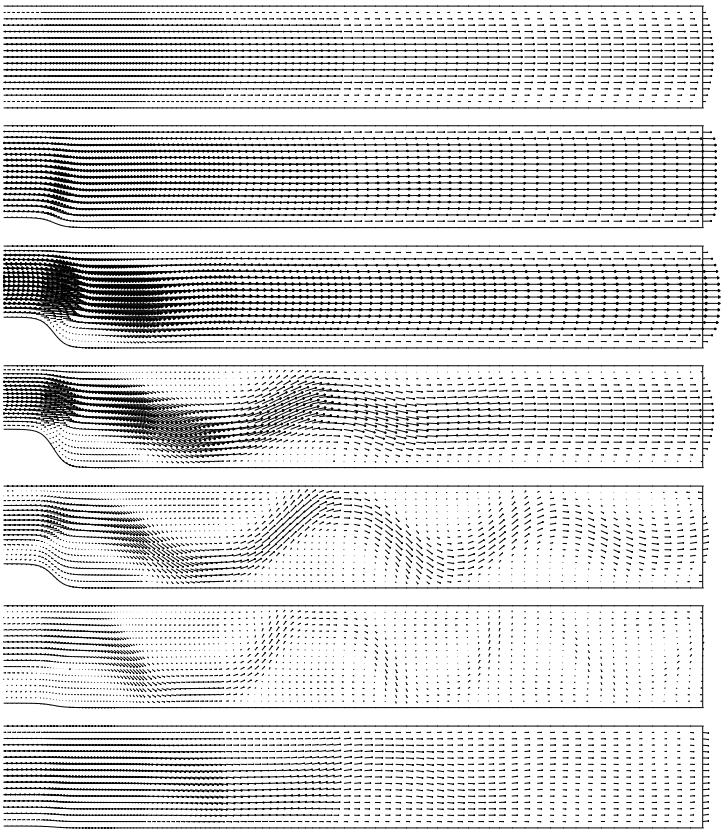

Figure 2.2: Velocity during one cycle of contraction (scaled)

Experiments and theory are described in [41]. The flow becomes oscillatory downstream of the indentation and a propagating train of waves appears, separation points and wave crests depending on the governing parameters. The experimental data indicate that the dependence on St is more important than that on Re for a realistic range of parameters.

The motivation for the investigation in [41] is to understand incompressible fluid flow through collapsible tubes such as arteries and veins. Theoretical models indicate that the dynamics of the oscillations are strongly influenced by the energy dissipation in the flow through the collapsible segment of the tube, see [6]. Numerical experiments using finite difference or finite volume discretizations are reported in [44], [46] and [21].

The temporal development of the flow computed on a mesh with 3584 ele-

ments and a timestep of $k = 0.0125$ is shown in figure 2.2. The channel is scaled by a factor of 2 in y-direction and only the part downstream of the indentation is shown. The parameters are chosen as $St = 0.038$ and $Re = 610$, a case described in [41]. Upstream of the indentation, the flow is not affected by the movement of the wall. Figure 2.2 shows that, concerning the general flow picture, the present computation is in good agreement with the cited ones. At time $t = 0.26$, separation appears in the lee of the indentation. A second vortex appears at $t = 0.42$ on the opposite side. After this, vortices continue building up alternately at the upper and lower wall. The times agree with those given by [44] and [21]. Since precise quantitative values are not given in the literature we cannot do a closer comparison.
We investigate the convergence properties with respect to the timestep k. In figure 2.3 the temporal development of the pressure and the first velocity component at point (8.0,0.5) are shown. From figure 2.3, the first order

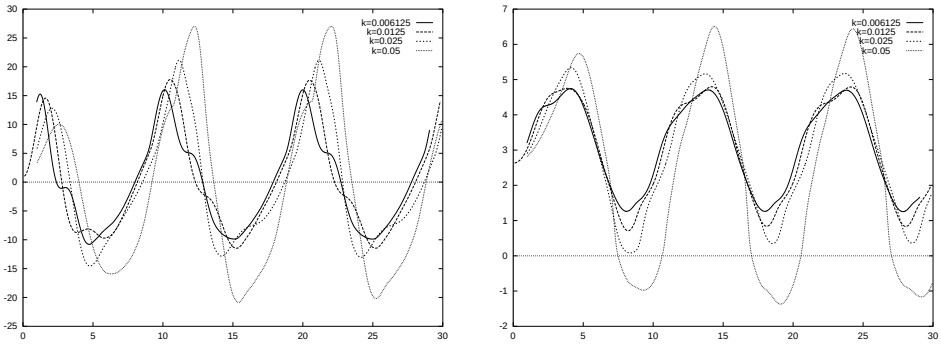

Figure 2.3: Pressure (left) and u-velocity (right) values for different timesteps

convergence is obvious. This indicates, that the order of the DG(0)-scheme, which is used here for the time-discretization, is preserved in the moving boundary case. A proof for this in the case of the heat equation is given later. As expected, the convergence of the velocity is better than that of the pressure. The timesteps considered here are small in comparison to computations without moving boundaries, such as the flow around a cylinder (see [53]). However, the shape of the pressure curve in figure 2.3 is still varying for the smallest timesteps. We therefore choose the smallest timestep, $k = 0.006125$, in the following computations.

Next, we present a comparison between the flows corresponding to different Reynolds and Strouhal numbers. For this, we chose the value of the pressure in (8.0,0.5). The upper picture in figure 2.4 compares the temporal development of the pressure for $Re = 410, 510, 610, 710$ at $St = 0.038$, whereas the lower one shows the results for $Re = 610$ and $St = 0.028, 0.038, 0.048, 0.058$. The values are chosen in accordance with the experiences from [6]. It is clear from these results that the flow is more sensitive to variations in St than for variations in Re for the range of values considered here.

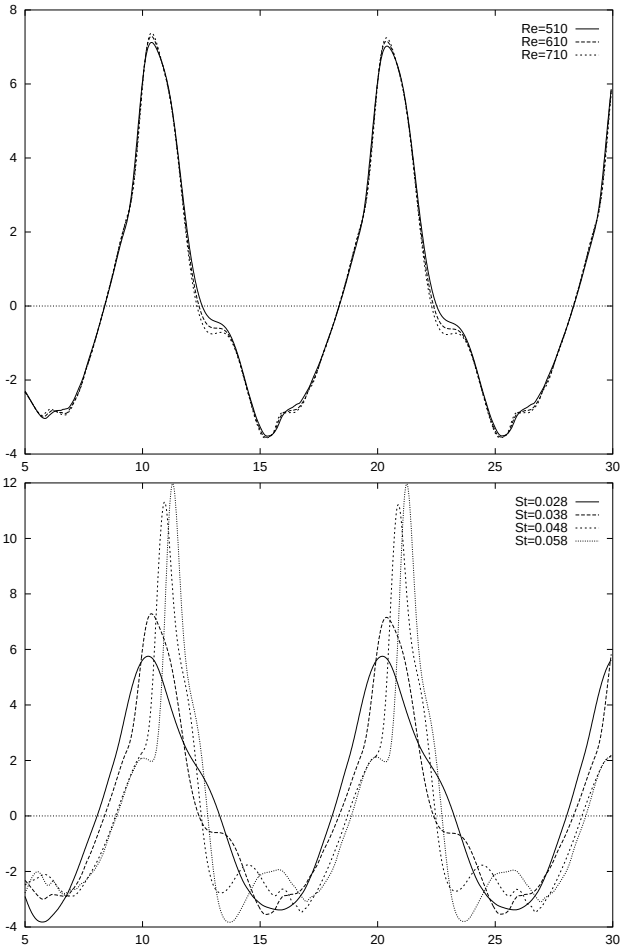

Figure 2.4: Temporal development of pressure value for varying Re (above) and for varying St (below)

2.2 Flow in a Water Pump

In this section we describe a simple model for a water pump. The geometry of the pump is a cylinder in which a piston moves up and down. The cylinder ends in a channel perpendicular to it. In order to allow the water to come in and to move out the channel has a valve on each side. When the piston moves upwards the left valve is closed and the right one is open. The situation is reversed when the piston moves downwards.
The flow in this pump is described by the incompressible Navier-Stokes equations. The movement of the piston is simulated by a temporal deformation of the cylinder. On the walls of the cylinder we have no-slip boundary conditions, whereas on the bottom of the piston we have to prescribe the velocity of the flow equal to the speed of the pistons motion. The action of the valves is modeled by appropriate boundary conditions: The closed state of the valve is simulated by homogeneous Dirichlet data for the velocity, while the open state is modeled by a Neumann boundary condition,

$$\nu \frac{\partial u}{\partial n} + p \cdot n = 0.$$

For a mathematical analysis of this condition see ([30]).
This set of time-dependent boundary conditions is summarized in figure 2.5. The left and right snapshots show the piston moving up and down respectively. The position of the piston is described by $y(t) = \alpha(t)\eta$ with a reference

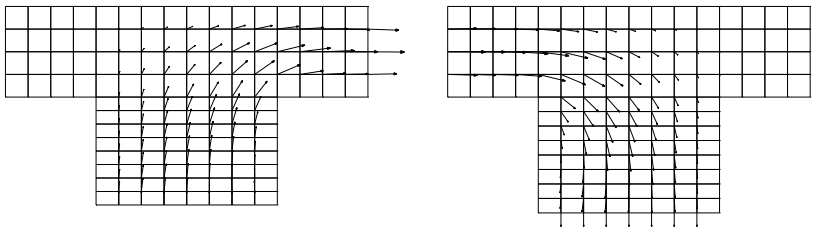

Figure 2.5: Boundary conditions for the water pump

coordinate η. We suppose the piston to be driven by a connection rod and

get the following formula for α:

$$\alpha(t) = r\sin(\lambda t) + \sqrt{l^2 - r\cos(\lambda t)^2},$$

where r denotes the radius of a moving circle on which the lower endpoint of the rod is moving and l is the rod's length. We additionally introduce the parameter λ which controls the speed of the movement. In our computations we chose l and r in such a way that the piston moves between $y = -2$ and $y = -.25$, obtaining a compression factor of 1:8. The parameter λ is equal to 0.1 and the viscosity ν in the Navier-Stokes equations is equal to 0.01.

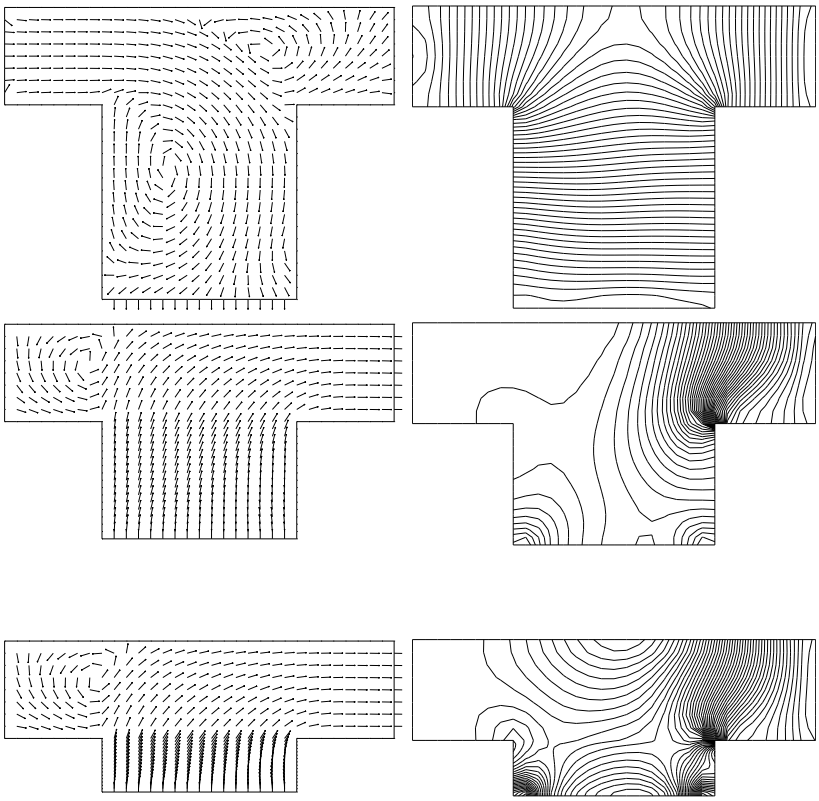

Figure 2.6: Velocity field and pressure isolines while the piston moves upwards

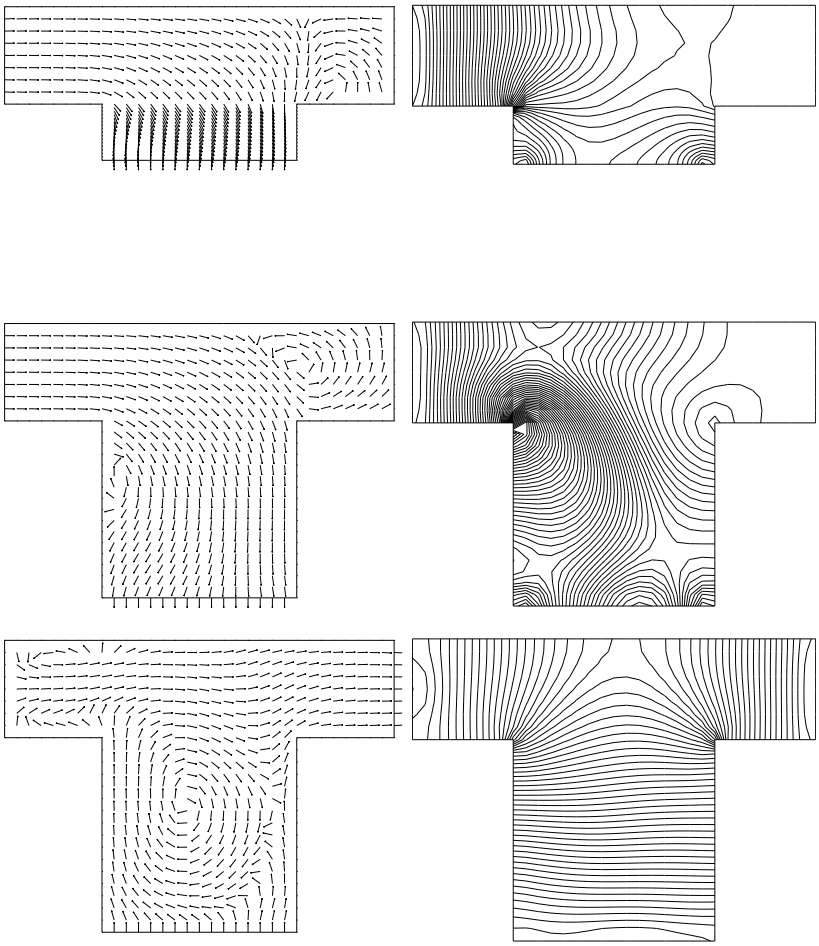

Figure 2.7: Velocity field and pressure isolines while the piston moves downwards

Figures 2.6 and 2.7 present the flow pattern during one cycle. The right pictures show isolines of the pressure. The temporal development of velocity and pressure values in the channel is shown in figures 2.8 and 2.9.
Due to the discontinuity of the time-dependent boundary conditions for the velocity between the bottom of the piston and the cylinder wall, the question of regularity for the solutions arises. The situation is similar to the "driven cavity" flow. Indeed we have a singularity in the pressure as well as for the velocity gradient. The existent of a unique solution for discontinuous boundary data in certain weighted Sobolev spaces which control the singularities is proven by Solonnikov, see [48]. In order to deal with the singularities of the

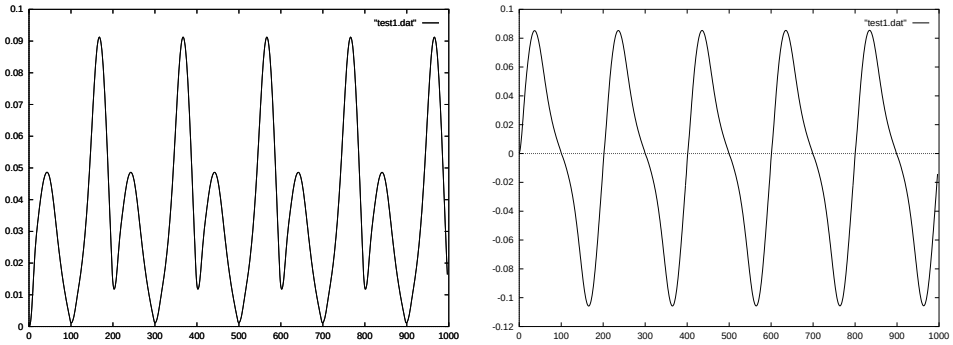

Figure 2.8: Evolution of vertical and horizontal velocity components in $(0,0)$

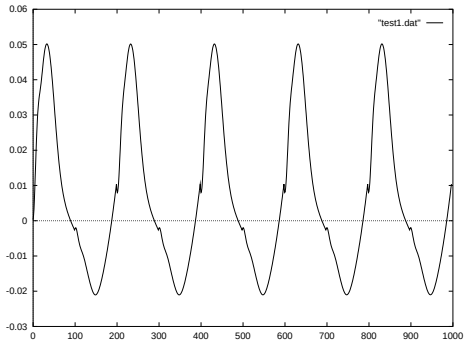

Figure 2.9: Evolution of the pressure in $(0,0)$

solution it seems to be natural to use local grid refinement near the corners. The history of a locally refinement during one cycle of motion is shown in figure 2.10. The mesh refinement is controlled by an a posteriori error estimator that will be presented in the following chapters. The singularities appearing in this problem have different sources. First, the singularity at the edges where the cylinder ends in the channel is caused by the reentrant corners. Second, the singularity at the point where the piston slides along the cylinder is caused by the discontinuity of the boundary conditions for the velocity. The cylinder wall enforces no slip conditions while on the bottom of the piston we must describe the velocity of the pistons motion. The impact of this boundary jump is difficult to analyze since it is coupled with the motion of the boundary. From the numerical point of view it is important to know

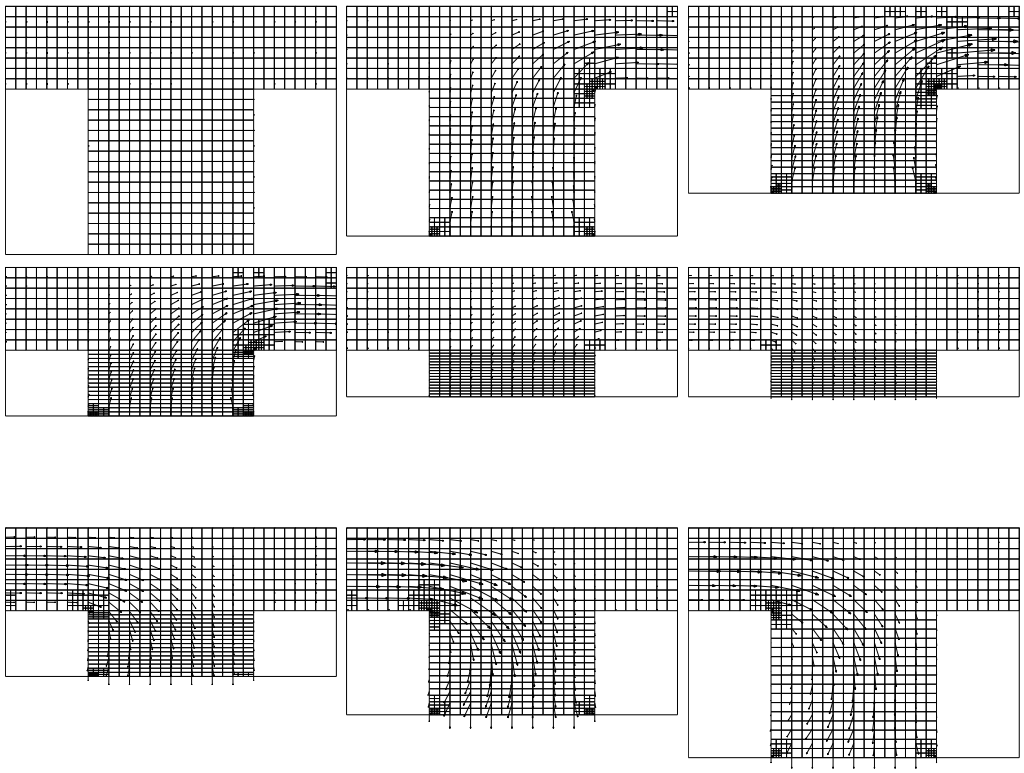

Figure 2.10: Evolution of the grid and corresponding flow pattern

whether these singularities destroy the accuracy of computational values at a given distance from the edges. In order to demonstrate this pollution effect we perform the following numerical test. We compare the temporal change of the pressure and the two velocity components at the origin for different refinement strategy. The first strategy does not use any local refinement. The second one uses local refinement at the upper corners only, whereas the third strategy allows a local refinement everywhere. The number of additional elements is restricted to ten per cent of the number of elements used for the global refinement. The result of this test is shown in figures 2.11, 2.12 and 2.13. The solid lines represent the computation on a globally refined mesh, whereas the dashed lines represent the values obtained on the locally refined meshes. Although the point of evaluation is chosen closer to the upper singularities, comparison between the two adaptive strategies shows that

the influence of the piston singularity is much stronger than the corner singularity. The maximal pressure values differ even at this location far from the piston by about ten per cent.

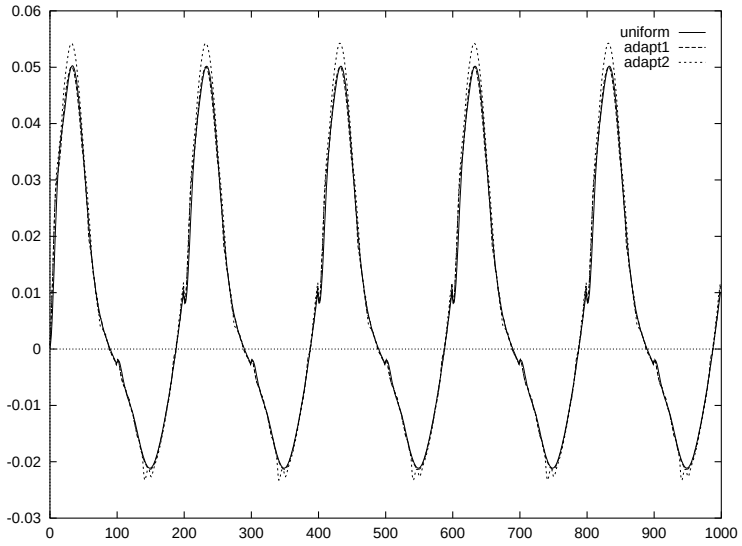

Figure 2.11: Comparison between uniform and adaptive refinement (pressure)

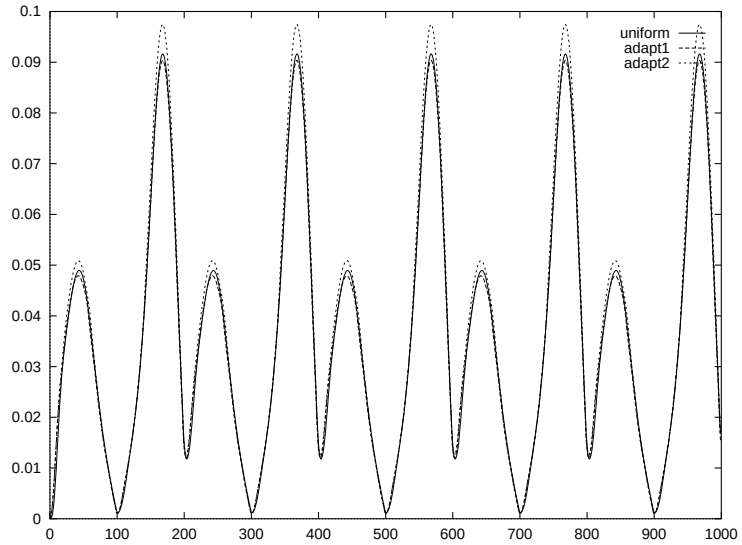

Figure 2.12: Comparison between uniform and adaptive refinement (u-velocity)

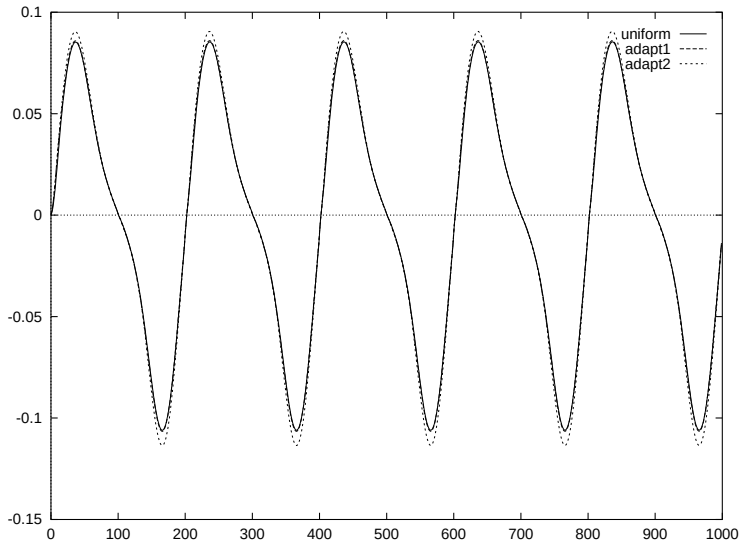

Figure 2.13: Comparison between uniform and adaptive refinement (v-velocity)

2.3 Time Discretization

In this section we develop an algorithm for the discretization of time dependent partial differential equations on a domain with moving boundaries.
The problem has been considered by several different authors. For example, Jamet, [32] and [33], Lesaint and Touzani [36] consider variable domains in view of free boundary equations. Lynch and Gray [39], Tezduyar, Behr and Liou [49] and Nomura and Hughes [40] aim at flow problems with moving boundaries. Error estimates are obtained by the first authors. In [32] a non-optimal estimate which treats time and space variables simultaneously is given. The result is improved for the Crank-Nicholson scheme applied to a one-dimensional problem discretized by finite differences in [33] and [36].
Supposing that the deformation of the domain can be expressed by a smooth transformation to a fixed region, we proof that the rate of convergence of a time-stepping scheme carries over to the case of moving boundaries. This is done by the introduction of an oblique time-derivative in weighted Sobolev spaces in order to allow the application of energy techniques from [50]. In the case of the Crank-Nicholson time-stepping for the one-dimensional heat equation, the scheme presented here is similar to the one considered in [33]. The major drawback of our analysis is the assumption on the regularity of the transformation describing the movement of the domain. The concrete form of the quantities which enter the discretization are given for the fluid problems presented above.

2.3.1 Investigation of the continuous problem

In this section we investigate the following parabolic problem on a domain with moving boundaries:

$$\begin{aligned} \frac{\partial u}{\partial t} - Au &= f \quad \text{in } \Omega_t \\ u &= 0 \quad \text{on } \partial\Omega_t \\ u(0) &= u_0. \end{aligned} \tag{2.2}$$

We suppose that the movement of the domain is described by a diffeomorphism Φ_t with $J_{\Phi_t} \geq \bar{J} > 0$, whose inverse will be denoted by Ψ_t. This

means:

$$\Omega_t = \Phi_t(\Omega_0)$$

Further, we suppose the operator $A = a_{i,j}\partial_{x_i x_j} + a_i \partial_{x_i} + a_0$ to be $H_0^1(\Omega_t)$–elliptic. Our first goal is to show that (2.2) admits a unique solution. For this we consider a weak formulation in an appropriate Hilbert space. This framework allows us the application of the Galerkin method to derive a finite element discretization.

Definition 2.1. *For a given time interval $I = (0,T)$ and $k \in R$, we define:*

$$L^2(I, H^k(\Omega_t)) = \left\{ u : I \to H^k(\Omega_t) \text{ measurable and } \int_I \|u\|^2_{H^k(\Omega_t)}\, dt < \infty \right\}.$$

$$H^1(I, H^k(\Omega_t)) = \Big\{ u : I \to H^k(\Omega_t) \,:\, \exists\, \tilde{u} \in L^2(I, H^k(\Omega_t)) \text{, such that}$$
$$f.a.a \;\; t, s \in I \quad u(t) - u(s) = \int_s^t \tilde{u}(\tau)\, d\tau \Big\}.$$

In order to obtain an operator which behaves on the time–dependent Sobolev spaces like the ordinary time derivative, we further define:

$$\begin{aligned} \delta_t &= \partial_t - \beta \cdot \nabla \quad \text{with} \\ \beta &= D\Psi_t^{-1} \frac{\partial \Psi_t}{\partial t}. \end{aligned}$$

We then have:

Lemma 2.1.

$$\delta_t \left[H^1(I, H^k(\Omega_t)) \right] \subset L^2(I, H^k(\Omega_t))$$

Proof. We use the fact that a C^∞-Diffeomorphism $\Phi : \Omega \to \tilde{\Omega}$ induces an isomorphism $\Phi^* : H^k(\tilde{\Omega}) \to H^k(\Omega)$.
The identity $\Phi(t, \Psi(x)) = x$ gives $\frac{\partial \Phi}{\partial t} + D\Phi \frac{\partial \Psi}{\partial t} = 0$,resp., $\frac{\partial \Phi}{\partial t} = -\beta$.
We define $v \in H^1(I, H^k(\Omega))$ by $v(t, \xi) = u(t, \Phi_t(\xi))$. Then we have $\partial_t v = \delta_t u$. □

The behavior of δ_t on Ω_t is described in figure 2.14 compared to ∂_t on the cylindrical domain $I \times \Omega$.

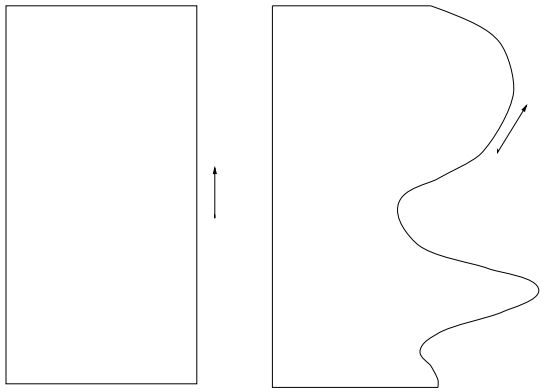

Figure 2.14: Definition of δ_t

Next we introduce a scalar product which takes into account the movement of the domain:

$$(u, v)_\rho := \int_{\Omega_t} uv\, \rho\, dx$$

The corresponding spaces will be indexed by ρ .

Lemma 2.2. *There is a weight ρ with $\bar{\rho} \geq \rho \geq \underline{\rho} > 0$, such that*

$$\frac{d}{dt}(u, v)_\rho = (\delta_t u, v)_\rho + (v, \delta_t u)_\rho$$

Proof. It is sufficient to show that

$$\frac{d}{dt}\int_{\Omega_t} \rho f\, dx = \int_{\Omega_t} \rho\, \delta_t f\, dx.$$

Transformation yields:

$$\begin{aligned}
\frac{d}{dt}\int_{\Omega_t} \rho f\, dx &= \frac{d}{dt}\int_{\Omega_0} \rho(t, \Phi_t) f(t, \Phi_t) J_{\Phi_t}\, d\xi \\
&= \int_{\Omega_0} \rho\, \delta_t f J_{\Phi_t}\, d\xi + \int_{\Omega_0} \delta_t \rho f J_{\Phi_t}\, d\xi + \int_{\Omega_0} \rho f \frac{d}{dt} J_{\Phi_t}\, d\xi \\
&= \int_{\Omega_t} \rho\, \delta_t f\, dx + \int_{\Omega_t} \left\{ \delta_t \rho - \rho \frac{d}{dt} \log J_{\Phi_t} \right\} f\, dx
\end{aligned}$$

The equation for ρ in braces is equivalent to:

$$\widetilde{\rho}_t - \beta \cdot \nabla \widetilde{\rho} = 0 \quad \text{and } \rho = J_{\Psi_t} \widetilde{\rho}.$$

For appropriate initial conditions this linear first order equation has a positive solution. □

Remark 2.1. *We describe Φ, β and ρ for the piston example. The domain corresponding to the cylinder ($y \geq 0$) has the following form:*

$$[-1,1] \times [\alpha(t), 0],$$

the transformations are therefore given by:

$$\Phi_t(\xi,\eta) = (\xi, -\alpha(t)\eta) \quad \Psi_t(x,y) = (x, -\frac{y}{\alpha(t)}).$$

We thus get:

$$\beta = (0, -y\frac{\alpha_t}{\alpha}) \quad \textit{and } \rho = -\alpha^{-1}.$$

On the part of the domain corresponding to the channel, we have $\Phi = Id$ and therefore $\beta = 0$ and $\rho = 1$. The derivative of β as well as the weight ρ therefore have a discontinuity at $y = 0$.
For the channel problem the transformation has the following form:

$$\Phi_t(\xi,\eta) = (\xi, -\alpha(\xi,t)\eta),$$

with α given by 2.1. The quantities β and ρ are given by formulae similar to the above ones.

Now we rewrite (2.2) as:

$$\begin{aligned} \delta_t u + \beta \cdot \nabla u - Au &= f \quad \text{in } \Omega_t \\ u &= 0 \quad \text{on } \partial\Omega_t. \end{aligned} \tag{2.3}$$

A weak formulation of this is given by:
Find $u \in H^1_{0,\rho}(\Omega_t)$, such that

$$(\delta_t u, v)_\rho + (\kappa \cdot \nabla u, v)_\rho + a(u,v) = (f,v)_\rho \qquad \forall v \in H^1_{0,\rho}(\Omega_t), \tag{2.4}$$

where we set $\kappa = a_{i,j}(\log \rho)_{x_j} + \beta$ for abbreviation and $a(\cdot,\cdot)$ denotes the bilinear form corresponding to the operator A. The solution u of (2.4) does not depend on ρ since the weighted norm is equivalent to the original one. We are now in a situation similar to the case of a fixed domain. For instance, we can apply the classical energy techniques to proof existence and uniqueness. For example, choosing $v = u$ in (2.4) yields an a priori estimate for $\|u(t)\|_\rho$. The above lemma allows us to choose $v = \delta_t u$, which gives an estimate for $\|\delta_t u\|_\rho$. We can therefore apply the Rellich theorem to a Galerkin sequence of (2.4), $\widetilde{u}_n = u_n \circ \Psi_t$. The ellipticity of operator A guarantees uniqueness in the usual way. We therefore obtain the following result:

Proposition 2.1. *The variational equation (2.4) admits a unique solution.*

2.3.2 Semi–discretization

First we discretize (2.4) in time. Given a triangulation of Ω_0 and a corresponding finite element space $V_h(0) = \operatorname{span}\{\varphi_i^0\} \subset H^1_{0,\rho}(\Omega_0)$ we construct $V_h(t) = \operatorname{span}\{\varphi_i\}$ by:

$$\varphi_i(t,x) = \varphi_i^0(\Psi_t(x)). \tag{2.5}$$

The semi–discrete analogue of (2.4) is the following: Find $u_h \in V_h(t)$, such that

$$(\delta_t u_h, v_h)_\rho + (\kappa \cdot \nabla u_h, v_h)_\rho + a(u_h, v_h) = (f, v_h)_\rho \qquad \forall v_h \in V_h(t)$$

Our special choice of basis functions has the following consequence:

Lemma 2.3. *The functions defined by (2.5) have the following properties:*

$$\frac{\partial \varphi_i}{\partial t} = \beta \cdot \nabla \varphi_i,$$
$$\delta_t\,[H^1(I, V_h)] \subset L^2(I, V_h).$$

Proof. By construction:

$$\frac{\partial \varphi_i}{\partial t} = \frac{\partial \Psi_t}{\partial t} \text{ und } \nabla \varphi_i = D\Psi_t^T \nabla \varphi_i^0.$$

A function $v \in H^1(I, V_h)$ can be written as:

$$\begin{aligned} v(t,x) &= \sum c_i(t)\varphi_i(t,x) \quad \text{such that} \\ \delta_t v &= \sum \frac{dc_i}{dt}\varphi_i + \sum c_i \frac{\partial \varphi_i}{\partial t} - \sum c_i \beta \cdot \nabla \varphi_i \\ &= \sum \frac{dc_i}{dt}\varphi_i. \end{aligned}$$

□

For the following error analysis we introduce the following bilinear form in order to guarantee positivity:

$$\tilde{a}(u,v) = \int_{\Omega_t} \{a_{i,j} u_{x_i} v_{x_j} + (a_i + a_{i,j_{x_j}}) u_{x_i} v + \tilde{a_0} uv\} \rho \, dx + \int_{\Omega_t} \kappa_i u_{x_i} v \rho \, dx. \tag{2.6}$$

We choose $\tilde{a_0}$ such that

$$\begin{aligned} \tilde{a}(u,u) &\geq C_1(t) \|\nabla u\|_\rho^2 \\ |\tilde{a}(u,v)| + |\tilde{a}_t(u,v)| &\leq C_2(t) \|\nabla u\|_\rho \|\nabla v\|_\rho. \end{aligned}$$

In the following we consider the problem:

$$(\delta_t u, v)_\rho + \tilde{a}(u,v) = (f,v)_\rho \qquad \forall v \in H^1_{0,\rho}(\Omega_t) \tag{2.7}$$

By means of the transformation $\tilde{u} = \exp((\tilde{a_0} - a_0)t)u$, this formulation is equivalent to (2.4). The discrete version of (2.7) is:

$$(\delta_t u_h, v_h)_\rho + \tilde{a}(u_h, v_h) = (f, v_h)_\rho \qquad \forall v_h \in V_h.$$

Theorem 2.1. *Let the transformation Φ_t be C^2 and the above assumptions on the operator A be fulfilled. Then we have the following error estimate for the semi-discretization of (2.7):*

$$\|e\|_\rho \leq C h^2 |u|_{\rho,2},$$

where the constant C only depends on an interpolation constant and the above constants C_1 and C_2.

Proof. We get the usual error identity for $e = u - u_h$:

$$(\,\delta_t e, v_h)_\rho + \tilde{a}(e, v_h) = 0 \qquad \forall v_h \in V_h.$$

Let $R : V \to V_h$ be the following projector:

$$\tilde{a}(u - Ru, v_h) = 0 \qquad \forall v_h \in V_h. \tag{2.8}$$

Decomposition of e in

$$e = u - u_h = (u - Ru) + (Ru - u_h) = \epsilon + \theta$$

yields

$$\|\theta(t)\|_\rho \leq \|\theta(0)\|_\rho + \int_0^t \|\,\delta_t \epsilon\|_\rho \, ds$$

and:

$$\|e(t)\|_\rho \leq \|\epsilon(0)\|_\rho + \|e(0)\|_\rho + \|\epsilon(t)\|_\rho + \int_0^t \|\,\delta_t \epsilon\|_\rho \, ds.$$

The following lemma concludes the estimation. □

Lemma 2.4.

$$\begin{aligned} \|\epsilon\|_\rho &\leq M_1 h^2 |u|_{\rho,2} \\ \|\,\delta_t \epsilon\|_\rho &\leq M_2 h^2 \left(|\,\delta_t u|_{\rho,2} + |u|_{\rho,2}\right) \\ \textit{with} \qquad M_1 &= CLC_2^2/C_1 \qquad M_2 = CLC_2^3/C_1. \end{aligned}$$

The constants C are independent of time.

Proof. Let φ be a solution of $A\varphi = f$ with homogeneous boundary values. Then we have

$$|\varphi|_{\rho,2} \leq L(\Omega_t)\|f\|_\rho, \tag{2.9}$$

where the constant L depends only on the convexity of the domain. The stability estimate (2.9) follows from Lemma 2.2. On the other hand, the

interpolation error estimate for finite elements is also independent of Ω .
We consider the dual problem with its discretization:

$$\begin{aligned} \tilde{a}(v,\varphi) &= (\epsilon, v)_\rho \qquad \forall v \in V \\ \tilde{a}(v_h,\varphi_h) &= (\epsilon, v_h)_\rho \qquad \forall v_h \in V_h, \end{aligned}$$

and have:

$$C_1\|\nabla\epsilon\|_\rho^2 \le \tilde{a}(\epsilon,\epsilon) = \tilde{a}(\epsilon,\epsilon - v_h) \le C_2\|\nabla\epsilon\|_\rho\|\nabla(\epsilon - v_h)\|_\rho,$$

it follows that

$$\begin{aligned} \|\nabla\epsilon\|_\rho &\le C_2/C_1 \inf_{v_h\in V_h}\|\nabla(u - v_h)\|_\rho \le CC_2/C_1 h|u|_{\rho,2}. \\ \|\epsilon\|_\rho^2 &= \tilde{a}(\epsilon,\varphi) = \tilde{a}(\epsilon,\varphi - \varphi_h) \le C_2\|\nabla\epsilon\|_\rho\|\nabla(\varphi - \varphi_h)\|_\rho, \end{aligned}$$

and so

$$\|\epsilon\|_\rho^2 \le CC_2 h\|\nabla\epsilon\|_\rho|\varphi|_{\rho,2} \le LCC_2h^2|u|_{\rho,2}\|\epsilon\|_\rho.$$

The weighted interpolation estimates are valid under the regularity assumption. Differentiation of (2.8) leads to:

$$\tilde{a}(\,\delta_t\epsilon, v_h) + \tilde{a}_t(\epsilon, v_h) = 0 \qquad \forall v_h \in V_h. \tag{2.10}$$

This yields:

$$\begin{aligned} C_1\|\nabla\,\delta_t\epsilon\|_\rho^2 &\le \tilde{a}(\,\delta_t\epsilon,\,\delta_t\epsilon) \\ &= \tilde{a}(\,\delta_t\epsilon,\,\delta_t\epsilon - v_h) - \tilde{a}_t(\epsilon, v_h - \,\delta_t\epsilon) + \tilde{a}_t(\epsilon,\,\delta_t\epsilon) \\ &\le C_2(\|\nabla\,\delta_t\epsilon\|_\rho + \|\nabla\epsilon\|_\rho)\|\nabla(\,\delta_t\epsilon - v_h)\|_\rho + C_2\|\nabla\,\delta_t\epsilon\|_\rho\|\nabla\epsilon\|_\rho, \end{aligned}$$

such that

$$\|\nabla\,\delta_t\epsilon\|_\rho \le CC_2/C_1h|\,\delta_t u|_{\rho,2} + CC_2^2/C_1h|u|_{\rho,2}$$

Let φ and φ_h be solutions of the dual problems with right hand side $\delta_t\epsilon$.

$$\begin{aligned} \|\,\delta_t\epsilon\|_\rho^2 &= \tilde{a}(\,\delta_t\epsilon,\varphi) = \tilde{a}(\,\delta_t\epsilon,\varphi - \varphi_h) - \tilde{a}_t(\epsilon,\varphi_h - \varphi) + \tilde{a}_t(\epsilon,\varphi) \\ &\le CC_2\,\{\|\nabla\,\delta_t\epsilon\|_\rho + \|\nabla\epsilon\|_\rho\}\,\|\nabla(\varphi - \varphi_h)\|_\rho + C_2\|\epsilon\|_\rho|\varphi|_{\rho,2} \\ &\le CC_2\,\{\|\nabla\,\delta_t\epsilon\|_\rho + \|\nabla\epsilon\|_\rho\}\,.hL\|\,\delta_t\epsilon\|_\rho + CC_2L\|\epsilon\|_\rho|\,\delta_t\epsilon|_{\rho,2} \end{aligned}$$

This gives:

$$\begin{aligned}\|\,\delta_t\epsilon\|_\rho \leq & CLC_2\left\{h\|\nabla\,\delta_t\epsilon\|_\rho + h\|\nabla\epsilon\|_\rho + \|\epsilon\|_\rho\right\}\\ \leq & CL\left\{hC_2C_2^2/C_1\left(h|\,\delta_t u|_{\rho,2} + h|u|_{\rho,2}\right)\right.\\ & \left. + h^2C_2^2/C_1|u|_{\rho,2} + C_2^3/C_1h^2|u|_{\rho,2}\right\},\end{aligned}$$

and therefore

$$\|\,\delta_t\epsilon\|_\rho \leq CLh^2C_2^3/C_1\left\{|\,\delta_t u|_{\rho,2} + |u|_{\rho,2}\right\}.$$

This achieves the proof. □

2.3.3 Full discretization

Using the expansion $u_h = \sum \hat{u}_i(t)\varphi_i(t,x)$, the above semi-discretization leads to the following set of ordinary differential equations for the coefficients $\hat{u}_i$:

$$M(t)\,\frac{d}{dt}\hat{u}(t) + \left(A(t) + R(t)\right)\hat{u}(t) = f(t),$$

where M, A and R denote the mass–, stiffness– and "mesh–movement"–matrices. We can now apply any of the well-known methods for stiff ordinary differential equations like implicit Euler, Crank-Nicholson , fractional-theta-scheme or discontinuous Galerkin to complete the discretization. The algorithm takes for the first two methods the form:

$$\begin{aligned}(M_n + \theta k(A_n + R_n)\hat{u}_n = & (M_{n-1} - (1-\theta)k(A_{n-1} + R_{n-1})\hat{u}_{n-1}\\ & + k\theta f_n + k(1-\theta)f_{n-1},\end{aligned}$$

where $\theta = 1$ and $\theta = 1/2$ correspond to the implicit Euler and Crank-Nicholson schemes. The index n of the matrices indicates evaluation of the integrals at time t_n on the corresponding mesh Ω_n. The fractional-theta-scheme is variant of the above θ-scheme where one timestep consists of three micro-timesteps with different θ and k. The discontinuous Galerkin method uses a weak formulation of the ordinary differential equation in order to derive a finite element discretization. Third order can be achieved by the use of linear discontinuous shape functions in time, see [22]. For the evaluation of the integrals we use the two point Gauss formula in the following tests.

Table 2.1 gives the results for one-dimensional numerical tests with domain $\Omega_t = (0, \alpha(t)), \alpha = 1 + \frac{1}{2}\sin(2\pi t)$. We use a uniform mesh with $h_0 = 0.001$ for Ω_0 and the time for the evaluation of the error e is $T = 6$. The timestep k is chosen as $k = 2^{-l}$. The right hand side is chosen to match the exact solution $u = (1 + \cos(t))\sin(x)$.

l	Euler	p	C-N	p	F-T	p	DG(1)	p
1	$5.03 - 2$	1.4	$2.81 - 2$	3.3	$2.33 - 2$	1.3	$9.46 - 3$	2.6
2	$1.65 - 2$	1.6	$4.40 - 3$	2.6	$7.08 - 3$	1.7	$1.69 - 3$	2.4
3	$6.55 - 3$	1.3	$1.13 - 3$	1.9	$1.60 - 3$	2.1	$2.16 - 4$	2.9
4	$2.84 - 3$	1.2	$2.85 - 4$	1.9	$3.73 - 4$	2.1	$2.80 - 5$	2.9
5	$1.32 - 3$	1.1	$7.14 - 5$	1.9	$8.96 - 5$	2.0	$3.71 - 6$	2.9
6	$6.36 - 4$	1.0	$1.79 - 5$	1.9	$2.26 - 5$	1.9	$5.39 - 7$	2.8

Table 2.1: L_2-error and order of convergence p for moving boundaries

From table 2.1 we see that the four time-stepping schemes considered here maintain their order of convergence in the case of moving meshes.

Chapter 3

Adaptive Finite Elements

In this chapter we describe our approach to adaptivity. After the presentation of the adaptive iteration, we explain a simple refinement algorithm which uses hanging nodes. This considerably simplifies mesh generation, especially in three dimensions. The heart of the algorithm is an a-posteriori error estimator which is used to design a local refinement (or de-refinement) criterion. As has been proposed in [7] we use the techniques of Johnson et al. for the estimation of the discretization error to derive a stopping criterion for the multigrid iteration which is used to invert the discrete system in each step. We further describe a modification of the residuum estimator which improves its efficiency.

An adaptive refinement algorithm leads to a hierarchy of grids that can be naturally used for a multigrid solver. Known to have optimal complexity and the property of using good initial guesses, the multigrid algorithm seems to be an appropriate solver within an adaptive solution procedure. While adaptivity leads to a near–optimal number of elements N, the use of multigrid should ensure an overall $O(N)$–complexity. Unlike in the case of global refinement, N is no longer proportional to the number of levels L. This causes the need of a proper definition of the multigrid procedure. As in the globally refined case we proceed in a level-wise fashion. The levels for the multigrid algorithm correspond to the number of refinements of a coarse grid element. In order to construct a multigrid algorithm with optimal complexity for locally refined meshes it is therefore important to restrict all arithmetic

operations for a certain level to the refined regions corresponding to it.
It is known that smoothing on the interior of the local refinements is sufficient to obtain a level–independent convergence rate (see [13]). However, it is also necessary to do the other operations, like calculating the residuals, on these same points only. This affords a proper organization of the iteration. Furthermore the algorithm should be able to treat refinement and coarsening on any level. This requires a flexible data structure. We will describe in detail our approach to this issue.
In the following chapters the solution algorithm presented here for the Poisson equation will be applied to the Stokes and Navier–Stokes equations as well as to the equations resulting from a splitting scheme in the case of the nonstationary incompressible flow equations.

3.1 Adaptive Algorithm

For the sake of simplicity we consider the Poisson equation on a given domain $\Omega \subset \mathbb{R}^d, d = 2, 3$ with homogeneous Dirichlet boundary data:

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{3.1}$$

Its weak formulation is written in the usual form:

$$\begin{aligned} &\text{Find } u \in V, \text{ such that:} \\ &a(u, \varphi) = (f, \varphi) \quad \forall \varphi \in V, \end{aligned}$$

where V denotes the Sobolev space $H_0^1(\Omega)$ and $a(\cdot, \cdot)$ is the Dirichlet form. The classical finite element method constructs an approximation V_h of V depending on a fixed triangulation of the domain Ω with mesh parameter h. In order to assess the solution u_h of the discrete equations, one needs an a posteriori estimate that involves only quantities depending on the given data and u_h. This is different from the usual a priori estimates which bound the error by a certain power of h times a bound for a norm involving the derivatives of the solution u. A posteriori error estimates thus have the following general form:

$$\|u - u_h\| \leq C_1 \eta(u_h), \tag{3.2}$$

where $\|\cdot\|$ is a given norm and η is the error indicator. The constant C_1 has to be known a priori in order to obtain a reliable error bound from (3.2). It is also important to have an estimate of the form

$$C_2\eta(u_h) \leq \|u - u_h\|. \tag{3.3}$$

This estimate insures efficiency of an algorithm which uses η for a local mesh refinement, provided that the constant C_2 is known to be bounded below. The two constants C_1 and C_2 thus control the efficiency quotient $q = \|u - u_h\|/\eta$, since $q \leq C_1/C_2$.
For residual based estimators, η is evaluated element-wise. If the norm chosen in (3.2) is the global energy- or L_2-norm, η takes the form $\eta^2 = \sum_{K\in\mathcal{T}} \eta_K^2$ with:

$$\eta_K^2 = h_K^\alpha \|R(u_h)\|_K^2,$$

where h_K denotes the local mesh parameter and $R(u_h)$ is the residual of the discrete solution with respect to the continuous operator. The exact form of R depends on the differential equation. A posteriori error estimates of this type have been developed for various kinds of partial differential equations (see [23], [57], and the references cited therein). They can be used for two purposes. First, we can decide whether a given discrete solution can be accepted with respect to a given tolerance. Second, we can reduce the error by refining the elements where the estimator η has large values, supposing a local behavior of the error.
An adaptive algorithm is an iteration which aims to produce a solution u_h satisfying

$$\eta(u_h) \leq \mathrm{TOL}, \tag{3.4}$$

with a given tolerance TOL. In order to achieve (3.4) with the least work possible we modify the mesh successively. A reasonable strategy to do this is to aim at the equi-distribution of the element-wise contributions of the estimator. This idea is based on the following heuristic argument.
Let us be given a mesh $\mathcal{T}_0$ which satisfies (3.4) and which equilibrates element-wise contributions, that is $\eta_K = \bar{\eta} \quad \forall K \in \mathcal{T}_0$. We construct another mesh

$\mathcal{T}_1$ with the same number of elements by refining a cell K and by coarsening another cell L. In order to compare the contributions over $K \cup \mathrm{L}$ for both grids, which are denoted by η_0^2 and η_1^2, we suppose the following local behavior of η_K:

$$\frac{1}{C} h_K^\alpha \leq \eta_K^2 \leq C h_K^\alpha. \tag{3.5}$$

We then have $\eta_0^2 = 5\,\bar{\eta}^2$ and

$$\eta_1^2 \geq \frac{1}{C^2}\,(2^\alpha 4 + (\frac{1}{2})^2)\,\bar{\eta}^2. \tag{3.6}$$

This leads to:

$$\frac{\eta_1^2}{\eta_0^2} \geq \frac{1}{C^2}\,\frac{2^\alpha 4 + (\frac{1}{2})^2}{5}. \tag{3.7}$$

For the L_2-norm we have $\alpha = 4$. This implies that the η_1 is bigger than η_0, if (3.5) holds with a constant $C \leq 3.58$.

The above reasoning relies upon some questionable assumptions. The estimate (3.5) is in general difficult to obtain. Global effects, due to pollution for example, are not at all taken into account. But even if (3.5) holds with a reasonable constant, a refinement algorithm based on the bisection of elements is in principle not able to equilibrate the element-wise contributions. This indicates that the construction of a local refinement criterion needs some care.

In each step of the iteration we thus refine the given mesh and invert the resulting discrete equations. The solution process can be put on a solid basis by incorporating also an estimator $\tilde{\eta}$ for the iteration error. This has been proposed in [7]. Being able to bound the additional error term by a computable quantity we can stop the iteration when the iteration error is lower than the discretization error.

Table 3.1 summarizes the essential structure of this adaptive algorithm. Steps 1 and 5 are necessary to preserve the previous solution as initial guess for the multigrid procedure, while step 3 aims to avoid large jumps in the grid function h.

The algorithm follows the simple strategy that it is sufficient to solve the

1. Save the current solution values
2. Refine (or coarsen) the grid according to the error estimator η by adding new elements to the hierarchy (or by deleting elements)
3. Smooth the obtained grid by refining additional elements
4. Create a numbering for the given grid
5. Restore the values of the previous solution to the newly allocated vectors
6. Iterate the discrete solution via multigrid until $\tilde{\eta} \leq \kappa\eta$, with a constant $\kappa < 1$.
7. Quit, if the error estimator η is sufficiently small

Table 3.1: Adaptive algorithm

discrete equations up to an accuracy in the order of the discretization error. This is taken into account by the introduction of $\tilde{\eta}$. Since the a posteriori estimate involving η requires exact resolution of the discrete system, it is important to guarantee that η may be evaluated for the given iterated approximate solution.

From the construction of this adaptive algorithm it is clear that we have to be able to add elements or take them off easily. The multigrid algorithm described below is an ideal solver on these varying meshes, since it uses just loops over the elements of the refinement levels.

Although different refinement strategies could be used, we only consider the simplest one using bisection. The refined mesh may have hanging nodes. In view of three dimensional applications it is desirable to avoid auxiliary elements that are usually introduced in order to obtain a regular triangulation. To avoid these difficulties we describe a simple way to deal with hanging nodes. A simple grid with two hanging nodes is presented in figure 3.1.

The definition of the discretization presents no difficulty in the frame of conforming finite elements. The discrete space V_h is defined by the continuity requirement of the finite element functions. In order to avoid quadrilaterals with five degrees of freedom we do not associate finite element function to

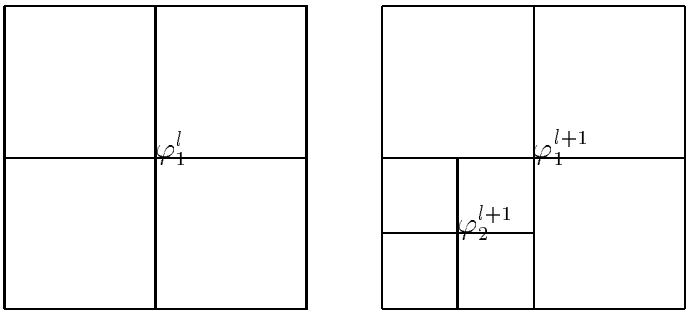

Figure 3.1: Refinement with hanging nodes

the hanging nodes. The refinement in figure 3.1 from level l to level $l+1$ introduces a new function φ_2^{l+1}. The Lagrangian representation implies the following change for φ_1.

$$\varphi_1^{l+1} = \varphi_1^l - \frac{1}{4}\varphi_2^{l+1}$$

This function is shown in figure 3.2. For implementational convenience it

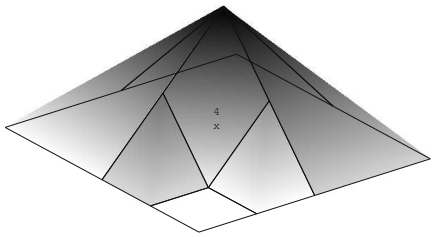

Figure 3.2: Basis function (only one element is refined)

is advisable to introduce virtual functions to the hanging nodes and set the corresponding degrees of freedoms according to the continuity requirement. In this way all computations can be done in the usual element-by-element fashion. We explain a simple way of dealing with the hanging nodes in the context of the multigrid algorithm below.

3.2 A residual-based error estimator

The residual–based a posteriori error estimator from [23] in the case of error control in L_2 reads as follows:

$$\eta \equiv C_I C_S \sum_K \left\{ h_K^4 \|f\|_K^2 + h_K^3 \sum_{\Gamma \subset K} \|[\frac{\partial u_h}{\partial n}]_\Gamma\|_{L_2(\Gamma)}^2 \right\}. \tag{3.8}$$

C_I and C_S denote an interpolation constant and a stability constant respectively. The jump over an edge Γ is denoted by $[\cdot]_\Gamma$.

Numerical examples in [7] indicate that this estimator leads to an h–independent quotient of estimated to true error. Its behavior with respect to the exact solution u, however, is less satisfactory, because the efficiency index q introduced above varies by a factor of up to ten for different exact solutions. One remedy to this problem, that is due to the dominance of the f–term, is the use of the L_2–projection P_h as interpolation operator instead of the nodal interpolation. This allows replacing $\|f\|_K$ by $\|f - P_h f\|_K$ as mentioned in [23]. We can now further substitute the term $h_K^4 \|f\|_K^2$ by $h_K^6 \|\nabla f\|_K^2$. On the one hand this modification requires the evaluation of the gradient of the right hand side which might not be known explicitly. The L_2–projection, on the other hand, does not seem to be appropriate for a local error estimator since it has a worse local behavior than a local interpolation operator.

We therefore present another modification of (3.8) which relies on a formula for integrals over an edge of a finite element. This kind of formula is also been used in the analysis of superconvergence for finite elements, see for example [11].

For a given edge Γ of an element K we denote by ∂^- the directional derivative normal to Γ and by h^- the element extension in the same direction. Furthermore, φ_Γ stands for a finite element function that is 1 on Γ and vanishes on the opposite edge. For sake of simplicity, only parallelograms are considered.

Lemma 3.1. *Let K be a parallelogram and $f \in H^1(K)$. Then there holds:*

$$\int_\Gamma f \, dS = \frac{|\Gamma|}{|K|} \int_K f(x,y)\, dxdy + \frac{h_\Gamma^- |\Gamma|}{|K|} \int_K \varphi_\Gamma \partial_\Gamma^- f(x,y)\, dxdy. \tag{3.9}$$

Proof. On the reference element $\hat{K} = (-1,1)^2$ with reference edge $\hat{\Gamma} = \{\eta = 1\}$ there holds:

$$\begin{aligned}\int_{\hat{\Gamma}} f\,dS &= \int_{-1}^{1} f(\xi,1)\,d\xi = \frac{1}{2}\int_{\hat{K}} f(\xi,\eta)\,d\xi d\eta + \int_{\hat{K}} \frac{\eta+1}{2} f_\eta(\xi,\eta)\,d\xi d\eta \\ &= \frac{1}{2}\int_{\hat{K}} f(\xi,\eta)\,d\xi d\eta + \int_{\hat{K}} \hat{\varphi}_{\hat{\Gamma}}(\xi,\eta)\,\partial_{\hat{\Gamma}}^{-} f(\xi,\eta)\,d\xi d\eta,\end{aligned}$$

where $\hat{\varphi}_{\hat{\Gamma}} = \sum_{N_i \subset \hat{\Gamma}} \hat{\varphi}_i$. For an arbitrary parallelogram K let $T_K : \hat{K} \to K$ be the linear transformation, that maps $\hat{K}$ to K. Its Jacobian is denoted by J_T. $T_K|_{\hat{\Gamma}}$ is a parameterization of Γ. The derivative of $T_K|_{\hat{\Gamma}}$ is denoted by DT. Thus we have:

$$\begin{aligned}\int_{\Gamma} f(x,y)\,dS &= \int_{\hat{\Gamma}} f(T)(DT^*DT)^{1/2}\,dS = \frac{|\Gamma|}{2}\int_{\hat{\Gamma}} f(T)\,d\xi d\eta \\ &= \frac{|\Gamma|}{2}\left\{\frac{1}{2}\int_{\hat{K}} f(T)\,d\xi d\eta + \int_{\hat{K}} \hat{\varphi}_{\hat{\Gamma}}\partial_{\hat{\Gamma}}^{-} f(T)\,d\xi d\eta\right\} \\ &= \frac{2|\Gamma|}{|K|}\left\{\frac{1}{2}\int_{\hat{K}} f(T)J_T\,d\xi d\eta + \int_{\hat{K}} \hat{\varphi}_{\hat{\Gamma}}\partial_{\hat{\Gamma}}^{-} f(T)J_T\,d\xi d\eta\right\} \\ &= \frac{|\Gamma|}{|K|}\int_K f(x,y)\,dxdy + \frac{h_\Gamma^{-}|\Gamma|}{|K|}\int_K \varphi_\Gamma \partial_\Gamma^{-} f(x,y)\,dxdy.\end{aligned}$$

□

Whereas the case of triangles, or parallelepipeds in three dimensions, can be treated in the same way, the case of non-parallelograms leads to additional terms, that will deteriorate the estimator. These terms, however, will become negligible if we use a regular refinement algorithm, since the angles which measure the derivation from a parallelogram tend to zero quadratically when h goes to zero.

To see how the formula (3.9) can be used to modify the estimator we shortly review the proof that (3.8) actually bounds the error. We denote the error by $e = u - u_h$. Let z be the solution of the continuous dual problem:

$$\begin{aligned}-\Delta z &= e \quad \text{in } \Omega \\ z &= 0 \quad \text{on } \partial\Omega.\end{aligned} \tag{3.10}$$

Then we have with z_h being the nodal interpolant of z:

$$\begin{aligned}\|e\|^2 &= (\nabla e, \nabla z) = (\nabla e, \nabla(z - z_h)) \\ &= (f, z - z_h) - (\nabla u_h, \nabla(z - z_h)) \\ &= \sum_K (f, z - z_h)_K - (\nabla u_h, \nabla(z - z_h))_K \\ &= \sum_K (f + \Delta u_h, z - z_h)_K - (\frac{\partial u_h}{\partial n}, z - z_h)_{\partial K}.\end{aligned}$$

We now apply the edge formula to get:

$$\begin{aligned}&= \sum_K \left\{ \int_K (f + \Delta u_h)(z - z_h)\, dx \right\} - \frac{1}{2} \sum_K \left\{ \sum_{\Gamma \subset K} [\partial_n u_h]_\Gamma \int_\Gamma (z - z_h)\, dx \right\} \\ &= \sum_K \left\{ \int_K (f + \Delta_h u_h)(z - z_h)\, dx \right\} \\ &\quad - \sum_K \left\{ \sum_{\Gamma \subset K} \frac{|\Gamma|}{2|K|} [\partial_n u_h]_\Gamma \int_K \varphi_\Gamma h_\Gamma^- (\partial_\Gamma^- (z - z_h))\, dx \right\}.\end{aligned}$$

For abbreviation we set:

$$\Delta_h u_h|_K \equiv \Delta u_h|_K - \frac{1}{2} \sum_{\Gamma \subset K} \frac{|\Gamma|}{|K|} [\partial_n u_h]_\Gamma \quad \text{and} \tag{3.11}$$

$$R_K(u_h) \equiv \frac{1}{2} \frac{1}{\sqrt{3}} \sum_{\Gamma \subset K} \frac{|\Gamma|}{|K|^{\frac{1}{2}}} |[\partial_n u_h]_\Gamma|. \tag{3.12}$$

This leads to the estimate:

$$\begin{aligned}\|e\|^2 \leq & \sum_K \left\{ \int_K (f + \Delta_h u_h)(z - z_h)\, dx \right\} \\ & + \sum_K \left\{ \sum_{\Gamma \subset K} \frac{|\Gamma|}{2|K|} [\partial_n u_h]_\Gamma \|\varphi_\Gamma\|_K \|h_\Gamma^- \partial_\Gamma^- (z - z_h)\|_K \right\} \\ \leq & \sum_K \left\{ \|f + \Delta_h u_h\|_K + R_K(u_h) \right\} \|z - z_h\|_K \\ \leq & C_I^\alpha \left\{ \sum_K h_K^{2\alpha} (\|f + \Delta_h u_h\|_K^2 + R_K(u_h)^2) \right\}^{1/2} \|\nabla^\alpha z\|\end{aligned}$$

and using the stability estimate $\|\nabla^\alpha z\| \leq C_S^\alpha \|e\|$ for the dual problem (3.10)

$$\leq C_I^\alpha C_S^\alpha \left\{ \sum_K h_K^{2\alpha} (\|f + \Delta_h u_h\|_K^2 + R_K(u_h)^2) \right\}^{1/2} \|e\|.$$

On orthogonal grids Δ_h corresponds to an element-wise 5–point finite difference stencil. We therefore get a smaller element-wise residual in comparison to (3.8), since the f-term is balanced by $\Delta_h u_h$.

3.3 Multigrid method on locally refined meshes

First we define several spaces of finite element functions that will be used in the definition of the multigrid algorithm on locally refined meshes. The implementational details, like the treatment of hanging nodes on the inter-level boundaries, are based on these definitions. We show how we can treat hanging nodes in a simple way – using nodal values – by the multigrid algorithm, yielding a conforming discretization with well known properties. The use of this technique considerably simplifies the mesh generation (no need of auxiliary matching triangles) and, at the same time avoids non–nested spaces or a supplementary treatment at the interior boundaries.A typical grid which we have in mind is shown in figure 3.3. First we have to introduce some sets

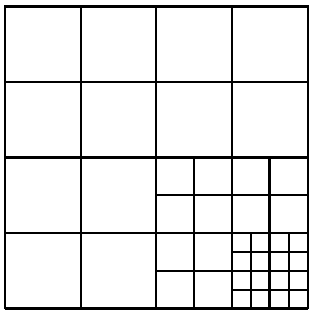

Figure 3.3: Structure of a locally refined grid with hanging nodes

of elements, that will be used in the definition of the finite element spaces. We start with a given coarse grid $\mathcal{T}_0$ consisting of elements E_0. During the adaptive refinement, we create a hierarchy of levels, indexed by $l = 0, \cdots, L$, by subdividing certain elements. The elements of level l, i. e., all the elements that are the l–th refinement of some K_0 in E_0, are denoted by E_l. All elements that are not further refined are called active. To define the appropriate spaces we associate finite element functions in the usual way with a

given grid. The spaces we need are the following:

$$\tilde{V}_l = \left\{\varphi_l : \mathrm{supp}(\varphi_l) = \cup_{K\in\tilde{E}} K, \tilde{E} \subset E_l\right\},$$
$$\hat{V}_l = \left\{\varphi \in \tilde{V}_l : \text{ some part of } \mathrm{supp}(\varphi) \text{ is active}\right\}$$
$$V_l = \cup_{k\le l} \hat{V}_k \cup \tilde{V}_l.$$

The definition of these finite element spaces is illustrated in figure 3.4 in the case of the above grid with two levels of refinement. The thick lines indicate the regions on which the corresponding functions have their support. The

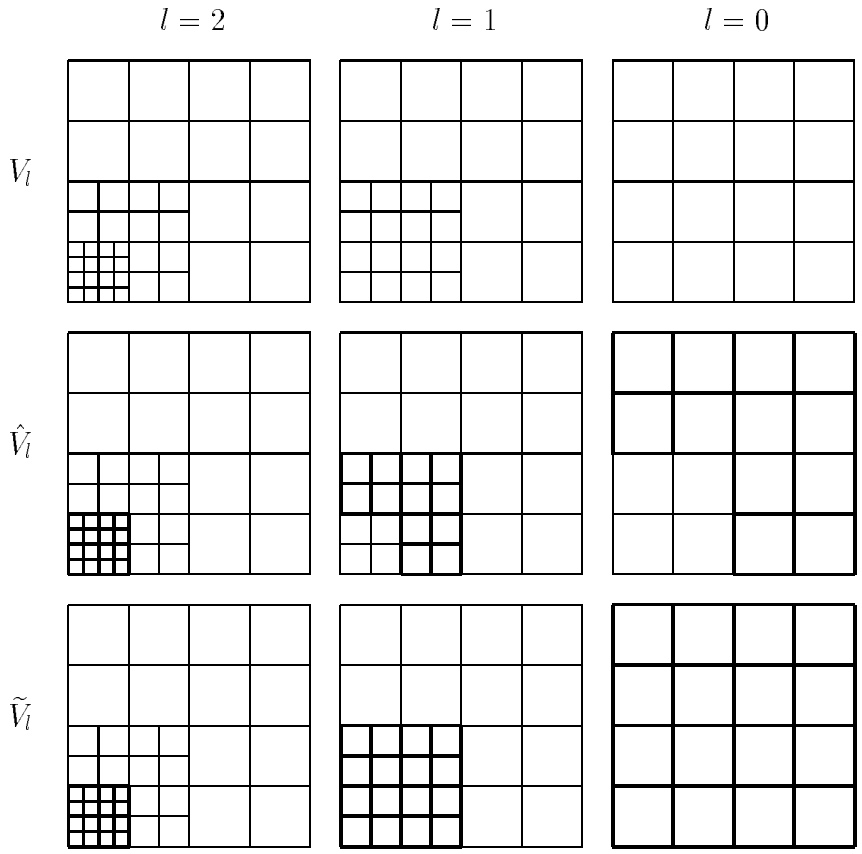

Figure 3.4: Definition of the spaces V_l $\hat{V}_l$ and $\tilde{V}_l$

multigrid algorithm uses the nested spaces $V_l \subset V, 1 \le l \le L$. Next we define

the discrete operators which are used in the multigrid algorithm.

$$(A_l\varphi,\psi) = a(\varphi,\psi) \quad \forall\varphi,\psi \in V_l$$
$$(\hat{A}_l\varphi,\psi) = a(\varphi,\psi) \quad \forall\varphi,\psi \in \hat{V}_l.$$

Further we denote the L_2–projection on V_l by P_l and the L_2–projection on $\tilde{V}_l$ by $\tilde{P}_l$. The smoothing operator on level l is denoted by S_l. Since smoothing takes place only on the refined part, S_l acts only on the space $\tilde{V}_l$. Thus, it is the identity on $V_l - \tilde{V}_l$.
The usual recursive definition of the V-cycle, setting $MG_l(b_l, u_l)$ for the the multigrid operator on level l with right hand side b_l and initial guess u_l, looks as follows: In the case of symmetric elliptic equations it is shown in [13] that

1. Pre-smoothing $v_l = u_l + S_l(b_l - A_l u_l)$
2. Correction $w_l = v_l + MG_{l-1}(0, P_{l-1}(b_l - A_l w_l))$
3. Post-smoothing $MG_l(b_l, u_l) = w_l + S_l(b_l - A_l w_l)$

Table 3.2: Basic multigrid algorithm

the multigrid algorithm described above converges with a contraction rate independent of the number of levels.
The difficulty for the implementation is that the calculation has to take place on the spaces $\tilde{V}_l$ in order to have optimal complexity. We split up the computations on the spaces V_l in order to replace them by the more convenient spaces $\tilde{V}_l$. This allows us to do all necessary calculations in a loop over all elements of level l.
Next we describe some details of the implementation of the multigrid algorithm. First, we note that there is a difficulty with algorithm (3.3) on locally refined grids. The entries of the solution vector on the lower levels, being corrections, will of course tend to zero, while they tend to the solution on the highest level. But the space $\tilde{V}_l$ consists of two different parts, the space $\hat{V}_l$ that belongs to V_L on the one hand, and the refined part $\tilde{V}_l - \hat{V}_l$ on the other hand. If we use vectors corresponding to $\tilde{V}_l$, we have the problem that one part of the components carries a different information than the other.

This difficulty is overcome by the use of the following version of the multigrid algorithm, which uses a translation on the lower grids to ensure the convergence of u_l to the solution for $l < L$. This modification has been introduced for the purpose of non–linear multigrid algorithms (see [26]) and is known as "full approximation scheme".

1. Pre-smooth $v_l = u_l + S_l(b_l - A_l u_l)$
2. Correction $w_l = v_l + MG_{l-1}(P_{l-1}v_l, P_{l-1}(b_l - A_l w_l) + A_{l-1}w_{l-1})$
3. Post-smooth $MG_l(b_l, u_l) = w_l + S_l(b_l - A_l w_l)$

Table 3.3: Multigrid algorithm with shift

Next we describe the organization of this algorithm, which allows us the use of the spaces $\tilde{V}_l$. The recursive algorithm **mgstep**(l,u,f,w,d) , where l denotes the level, u the current iterate, f the right hand side, w the vector for the shift in table 3.3, and d an auxiliary vector needed for the calculation of the defects is shown in table 3.4. This algorithm is called in the iteration loop

1. $if(l = 0)$ solve; $else$ smooth on $\tilde{V}_l$
2. compute the residual on $\tilde{V}_l$
3. restrict to $\tilde{V}_{l-1}$
4. shift with $A_{l-1}w_{l-1}$
5. call $mgstep(l-1, u, f, w, d)$
6. compute $d_{l-1} = u_{l-1} - w_{l-1}$
7. update u_l by the prolongation of d_{l-1}
8. smooth on $\tilde{V}_l$

Table 3.4: Recursive algorithm **mgstep**(l,u,f,w,d)

given in table 3.5.
We next describe the operations which have to be performed on each $\tilde{V}_l$. Since smoothing takes place only on the refined regions, the application of

Loop over i
• Calculate the residual for the stopping criterion
• Set $w = u$
• Call $mgstep(L, u, f, w, d)$

Table 3.5: Multigrid iteration

the smoothing operator poses no problem. The computation of the defect on level l, which provides the inter-level transfer, can be broken up into successive calculations on the spaces $\hat{V}_k, k \leq l$. The defect on the interfaces is thus calculated in two steps. Having restricted the contributions from the elements on the higher level l, it is sufficient to add those from the coarser grid when we have passed to the level $l-1$. This is possible because the values belonging to $\tilde{V}_k, k < l$ of the solution $u_l \in V_l$ are unchanged on level l. Let us now examine in more detail the treatment of hanging nodes. In order to proceed in the element-wise manner, we introduce virtual basis functions $\tilde{\varphi}_i$ on the refined elements, as described in the following figure: Remarking

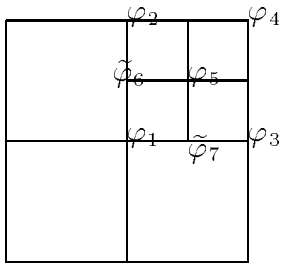

Figure 3.5: Additional basis functions

that the conforming functions φ_i are a linear combination of the functions $\tilde{\varphi}_i$ with coefficients given by the restriction operator, we can easily calculate the correct value of the residual on the level interfaces by means of the additional functions $\tilde{\varphi}_i$. Some precaution should be taken, however, for the virtual values on the hanging nodes. Before computing the action of the linear operators A_l, these values have to be the mean of the neighboring ones (averaging). After this calculations the coefficients have to be brought back

to the basis φ (distributing).
In the following we present some numerical experiences with the multigrid iteration described above. First we test the algorithm on grids with different levels of local refinement. To this end we subdivide the lower right corner of an uniform grid m times, see figure 3.3. In the following figure we give the reduction rates for the V–cycle with 2 Jacobi iterations as pre– and post-smoothing. This example is of course artificial since the refinement does not have any effect on the global discretization error. Figure 3.6 shows the ex-

l—m	0	1	2	3	4
1	0.03	0.02	0.06	0.08	0.09
2	0.06	0.05	0.07	0.09	0.10
3	0.06	0.07	0.09	0.10	0.10
4	0.07	0.08	0.09	0.10	0.10
5	0.08	0.09	0.09	0.10	0.10
6	0.08	0.09	0.09	0.10	0.11
7	0.09	0.09	0.09	0.11	0.12

Figure 3.6: Reduction rate for edge refining

perimental reduction rate measured over the first ten iterations for different levels of global refinements l. We conclude from figure 3.6 that there is no deterioration due to the local refinement.
The next example is more realistic. We solve the Poisson equation on the unit square with an approximate delta load. The refinement is done according to the error estimator described above. In order to measure the reduction rate we solve the discrete equations in each refinement step with an accuracy of ten digits. In the next figure we show the number of elements, the number of iterations and the reduction rate for each step. We use the V–cycle with one step of Gauss–Seidel as pre– and post-smoothing. The last example is meant as a test for the multigrid algorithm in the case where we do not have full elliptic regularity. For this, we choose the Poisson equation with right hand side $f = 1$ on a domain with a slit. This is a good test for the multigrid algorithm since the values of the solution far from the slit vary considerably

i	No. El	$\hat{\rho}$
1	64	0.064
2	196	0.144
3	532	0.105
4	1552	0.105
5	4648	0.093
6	15412	0.091
7	46864	0.096
8	97660	0.107
9	141160	0.117

Figure 3.7: Reduction rate for delta load

during the refinement procedure, due to the pollution effect. Since we smooth only locally the transport of information is managed by inter-level transfer. In figure 3.8 we show a grid occurring in the calculations. From this, the highly localized structure of the mesh is apparent. The finest elements have a level of 14 refinements. The total number of elements, however, is only 2968. As the theory presented in [13] indicates, the V–cycle still gives a re-

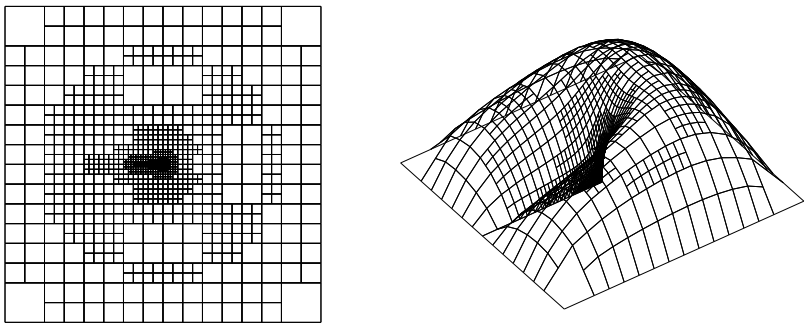

Figure 3.8: Grid and Solution for slit problem

duction rate independent of the number of refinement levels. The results are presented in figure 3.9. We additionally give the number of iterations and the reduction factor for the variable V–cycle. This multigrid cycling which

uses more smoothing iterations on the lower levels, has been advocated in [13]. Since the variable V–cycle affords the same amount of work as the W–

Adaptive Steps		V-cycle	Var. V-cycle
i	No. El	$\hat{\rho}$	$\hat{\rho}$
1	64	0.117	0.064
2	256	0.142	0.074
3	1024	0.168	0.085
4	4084	0.189	0.094
5	4450	0.214	0.130
6	5254	0.222	0.138
7	6574	0.233	0.154
8	12010	0.230	0.171
9	14842	0.236	0.178

Figure 3.9: Reduction rate for slit domain

cycle, the use of this algorithm does not pay in this example, since the final computational time is shorter for the simple V–cycle procedure.

Chapter 4

Error estimators for the Stokes Equations

This section describes the application of the adaptive algorithm presented in the previous chapter to the Stokes equations. While most of the implementational details are similar to the case of the Poisson equation, the key point is the error estimator. Again, we need an a posteriori error estimator as well for the discretization error as for the iteration error.
In the first section, we briefly introduce the discretization of the Stokes equations by stabilized finite elements. The following section presents the estimator for the discretization error. It is a generalization of the well-known residual based indicator for the Poisson equation, as proposed by Babuska and Rheinboldt [4] or Eriksson and Johnson [22]. For the Stokes equations, a posteriori estimates have been presented by Verfürth [56] and Bernardi et al. [5]. Usually the velocity error is estimated in the H^1– and the pressure error in the L_2–norm. As the quantity of physical interest in flow computations is the L_2–norm of the velocity, representing the kinetic energy, we develop an a posteriori estimator for this norm. We proof that the true error is bounded by the estimator, leading to a reliable error control. The proof of efficiency which could be done by comparison with sharp a priori estimates is not carried out here. The corresponding estimates can be obtained following the lines of [22]. We prefer to demonstrate the performance of the estimator by numerical computations.

The next section presents an a posteriori estimator for the iteration error. This leads to an effective stopping criterion for the multigrid iteration. The final section is devoted to numerical tests for these estimators.
The a posteriori estimators presented here involve several constants. In principle there is an interpolation constant C_I and a stability constant C_S as in the theory of [23]. While the first one only depends on the characteristics of the grid, the second constant reflects the stability properties of the dual problem. In the case of the linear Stokes equations this constant can be easily estimated. For the non-linear Navier–Stokes equations, however, the estimation of this constants is an important problem (see [35]).
Since the estimator consists of several terms, there are also different constants in front of each contribution. The calibration of these constants, which is important to obtain an efficient upper bound for the error, is done by preliminary computations on a coarse grid with a known solution. This is possible because the estimators are independent of the mesh size and of the exact solution.

4.1 Discretization of the Stokes equations

We discretize the Stokes equations with Dirichlet boundary values,

$$\begin{aligned} -\Delta u + \nabla p &= f \quad \text{in } \Omega \\ \nabla \cdot u &= 0 \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

The weak formulation of this system of equations is given by:

$$\begin{aligned} (\nabla u, \nabla v) - (p, \nabla \cdot v) &= (f, v) \quad \forall v \in V \\ (\nabla \cdot u, q) &= 0 \qquad \forall q \in Q, \end{aligned}$$

where V and Q denote the spaces $H_0^1(\Omega)$ and $L^2(\Omega)/R$.
The standard Galerkin discretization consists in replacing the infinite dimensional Hilbert spaces V and Q by discrete spaces V_h and Q_h:

$$\begin{aligned} (\nabla u_h, \nabla v_h) - (p_h, \nabla \cdot v_h) &= (f, v_h) \quad \forall v_h \in V_h \\ (\nabla \cdot u_h, q_h) &= 0 \qquad \forall q_h \in Q_h. \end{aligned}$$

It is well known that this approach does not lead to a stable discretization unless the finite dimensional spaces fulfill the LBB condition,

$$\inf_{p_h \in Q_h} \sup_{v_h \in V_h} \frac{(p_h, \nabla \cdot v_h)}{\|\nabla v_h\|} \geq \gamma \|p_h\|,$$

see for example [25]. An alternative to the construction of finite element spaces satisfying this compatibility condition is presented by Hughes et al. (see [31]). The idea is to modify the discrete bilinear form in order to get a stable discretization. This is usually done by adding mesh-dependent least squares terms to the Galerkin formulation. For the choice of linear spaces for the velocities and the pressure the discretization reads as follows:

$$\begin{aligned} (\nabla u_h, \nabla v_h) - (p_h, \nabla \cdot v_h) &= (f, v_h) && \forall v_h \in V_h \\ (\nabla \cdot u_h, q_h) + \sum_K \delta_K (\nabla q_h, \nabla p_h)_K &= \sum_K \delta_K (f, \nabla q_h)_K && \forall q_h \in Q_h, \end{aligned}$$

where $\delta_K \sim h_K^2$. The structure of the system is changed by the appearance of a pressure stabilization term $c(p_h, q_h) \equiv \sum_K \delta_K (\nabla p_h, \nabla q_h)_K$. We also note that the term involving the Laplacian of the velocities vanishes for (isoparametric) (bi-)linear elements. This means that the discrete system is still symmetric.

If we use a piecewise constant pressure space ($Q_h = Q^0$) the discrete system takes the following form:

$$\begin{aligned} (\nabla u_h, \nabla v_h) - (p_h, \nabla \cdot v_h) &= (f, v_h) && \forall v_h \in V_h \\ (\nabla \cdot u_h, q_h) + \sum_K \delta_K \sum_{\Gamma \subset K} ([p_h]_\Gamma, [q_h]_\Gamma)_\Gamma &= 0 && \forall q_h \in Q_h, \end{aligned}$$

where $[\cdot]_\Gamma$ denotes the jump over an element edge Γ. The stabilizing form c clearly has again the meaning of a weighted discrete Laplacian. Note that consistency is satisfied in the sense that the jump terms vanish for a continuous pressure.

Due to the change of the bilinear form, stability for the pressure is now implied by the following generalized LBB condition:

$$\inf_{p_h \in Q_h} \left\{ \sup_{v_h \in V_h} \frac{(p_h, \nabla \cdot v_h)^2}{\|\nabla v_h\|^2 \|p_h\|^2} + \frac{c(p_h, p_h)}{\|p_h\|^2} \right\} = \gamma^2 > 0, \tag{4.1}$$

independently of h. This is easily seen in the following way:
Choosing the pair $\{u, p\}$ as test functions leads to

$$\|\nabla u\|^2 + c(p,p) = (f,u)$$

and thus

$$\|\nabla u\|^2 + c(p,p) \leq 2\,\|f\|_{-1}^2.$$

For the pair of test functions $\{-v^*, 0\}$, where v^* attains the sup in (4.1) we get

$$(p, \nabla v^*) = (f, v^*) - (\nabla u, \nabla v^*),$$

leading to

$$\frac{(p,v^*)^2}{\|\nabla v^*\|^2} \leq 2\,(\|f\|_{-1}^2 + \|\nabla u\|^2).$$

Combining the two estimations gives:

$$\|\nabla u\|^2 + \gamma^2 \|p\|^2 \leq 8\,\|f\|_{-1}^2$$

Remark 4.1. *The mechanism of the stabilization is as follows: Applying an inverse estimate shows that*

$$c(p,p) \leq \|p\|^2.$$

The pressure functions for which the usual inf-sup condition does not hold ("unstable modes") are the functions maximizing the quotient $c(p,p)/\|p\|^2$ and will thus be controlled by the additional term. On the other hand, smooth pressures lead to a small c and are handled by the divergence term.

Remark 4.2. *The correct choice of the bilinear form c is crucial for this discretization. On anisotropically refined grids with elements of the form $K = [0, h_x] \times [0, h_y]$ for example, the following bilinear forms are appropriate (see the following chapter):*

$$c(p_h, q_h) = \delta \sum_K h_x^2 (\partial_x p_h, \partial_x q_h)_K + h_y^2 (\partial_y p_h, \partial_y q_h)_K \quad \text{for } Q_h = Q^1,$$

and

$$c(p_h, q_h) = \delta \sum_K \sum_{\Gamma \subset K} h_\Gamma^- ([p_h]_\Gamma [q_h]_\Gamma)_\Gamma \, ds \quad \textit{for } Q_h = Q^0.$$

Remark 4.3. *The addition of the stabilization terms leads to the following discrete system: denoting the stiffness matrix by A, the divergence matrix by B, and the matrix corresponding to the stabilization by C, it gets the following form:*

$$\begin{bmatrix} A & -B^* \\ B & C \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

where the matrix C appears in the pressure diagonal block. As in the case of the pure Galerkin discretization, the LBB constant γ is related to the lowest eigenvalue of the Schur complement $S = C + BA^{-1}B^*$. This can be seen using the Rayleigh quotient for the positive definite matrix S:

$$\begin{aligned}
\lambda_{min}(S) &= \inf \frac{(Sp,p)}{\|p\|^2} = \inf \left\{ \frac{(BA^{-1}B^*p,p)}{\|p\|^2} + \frac{(Cp,p)}{\|p\|^2} \right\} \\
&= \inf \left\{ \frac{(AA^{-1}B^*p, A^{-1}B^*p)}{\|p\|^2} + \frac{(Cp,p)}{\|p\|^2} \right\} \\
&= \inf \left\{ \sup \frac{(AA^{-1}B^*p,u)^2}{(Au,u)\|p\|^2} + \frac{(Cp,p)}{\|p\|^2} \right\} \\
&= \inf \left\{ \sup \frac{(p,Bu)^2}{(Au,u)\|p\|^2} + \frac{(Cp,p)}{\|p\|^2} \right\} \\
&= \gamma^2
\end{aligned}$$

Remark 4.4. *The Galerkin least squares discretization is related to the so called MINI element, introduced by Arnold, Brezzi and Fortin in [2] for triangular meshes. The velocity space is enlarged by a bubble function on each element in order to ensure the inf–sup condition to hold. Elimination of the velocity degrees of freedom indeed leads to a stabilization term for the pressure in the divergence equation. Denoting the bubble functions by φ_b and writing $u = \bar{u}\varphi + u_b\varphi_b$, leads to:*

$$u_b = (\nabla\varphi_b, \nabla\varphi_b)^{-1} \left\{ (\nabla\cdot\varphi_b, \varphi)p + (f, \varphi_b) \right\},$$

where we use the fact the $(\nabla\varphi_b, \nabla\varphi) = 0$. *Inserting the expression for* u_b *in the divergence equation finally gives:*

$$(\nabla\cdot\bar{u}, \xi) + \sum_K \frac{\|\varphi_b^K\|^2}{\|\nabla\varphi_b^K\|_K^2}(\nabla\xi, \nabla p)_K = \sum_K \frac{\|\varphi_b^K\|^2}{\|\nabla\varphi_b^K\|_K^2}(f, \nabla\xi)_K$$

The size of the stabilizing parameter corresponding to the MINI element is therefore $\delta_K \sim \|\varphi_b^K\|^2/\|\nabla\varphi_b^K\| \sim |K|$.

The combination of the stability of the scheme with the usual interpolation properties of (bi-)linear finite elements leads to the following convergence results, which can be found in [10] and [29]:

Theorem 4.1. *Let* $e_u = u - u_h$ *and* $e_p = p - p_h$. *If the continuous problem is* H^2*–regular, there holds for* $m = 0, 1$*:*

$$\|e_u\|_m + \|e_p\|_{m-1} \leq Ch^{2-m}\left(\|u\|_2 + \|p\|_1\right)$$

Numerical results confirming the above estimates and a numerical study of the effect of the stabilization parameter are reported in [29].

4.2 A posteriori estimates for the discretization error

In this section we present an error estimator for the discretization error. It is based on the evaluation of element-wise residuals in a similar way as the error estimator for the Poisson equation. The derivation of the estimator runs along the lines presented by Johnson et al., see [23]. First, a dual problem is used to represent the error. Next, we use the projection property of the finite element discretization combined with the "strong" stability properties of the dual problem to produce the necessary power of the discretization parameter h. The estimator consists of four residual terms, similar to the one for the Poisson equation. A fourth term comes from the stabilization and takes into account the fact that we have Galerkin "orthogonality" with respect to the stabilized bilinear form a_δ instead of a. Compared to the estimator proposed in [56], which controls the velocity error in H^1 and the pressure in L_2, we additionally allow the norms to be one order lower for both. Error control in

L_2 for the velocities seems to be more adequate in flow computations since this norm presents a physical quantity, the kinetic energy.
The following theorem proofs the reliability of the estimator by showing that the true error is bounded by a constant times the sum of the residuals. As mentioned above, there are two different constants involved. The first one, denoted C_I, is a mesh dependent interpolation constant, whereas the second one, C_S, measures the stability of the dual problem.

Theorem 4.2. *Let the Stokes equations be H^2–regular. Then, there holds for $m = 0, 1$:*

$$\|u - u_h\|_m + \|p - p_h\|_{m-1} \leq \eta_m(u_h, p_h),$$

with

$$\begin{aligned}\eta_m(u_h, p_h) \equiv C_S C_I \Bigg[& \left\{ \sum_K h_K^{2(2-m)} \|f + \Delta_h u_h - \nabla_h p_h\|_K^2 \right\}^{1/2} \\ & + \left\{ \sum_K h_K^{2(\frac{3}{2}-m)} \sum_{\Gamma \subset K} \|[p_h \cdot n - \frac{\partial u_h}{\partial n}]\|_\Gamma^2 \right\}^{1/2} \\ & + \left\{ \sum_K h_K^{2(1-m)} \|\nabla \cdot u_h\|_K^2 \right\}^{1/2} + \left\{ \sum_K \delta_K^2 \|\nabla p_h - f\|_K^2 \right\}^{1/2} \Bigg]\end{aligned}$$

where

$$(\Delta_h u_h - \nabla_h p_h)|_K \equiv (\Delta u_h - \nabla p)|_K + \frac{|\Gamma|}{2|K|} \sum_{\Gamma \subset K} [p \cdot n - \frac{\partial u_h}{\partial n}]_\Gamma$$

The H^{-1}-norm for the pressure is understood in the following sense:

$$\|p\|_{-1} = \sup_{\varphi \in H^1/R} \frac{(p, \varphi)}{\|\varphi\|_1}.$$

Remark 4.5. *We use the modification of the estimator presented before for the Poisson equation.*

Proof. We use interpolation operators $I_h : V \to V_h$ and $J_h : Q \to Q_h$ with the usual properties.

For abbreviation we set $X \equiv V \times Q$ and denote the elements of X as couples $\{u, p\}$. We further introduce the bilinear form a describing the weak formulation:

$$a(\{u,p\},\{v,q\}) \equiv (\nabla u, \nabla v) - (p, \nabla \cdot v) + (\nabla \cdot u, q),$$

and the stabilized version

$$a_\delta(\{u,p\},\{v,q\}) \equiv (\nabla u, \nabla v) - (p, \nabla \cdot v) + (\nabla \cdot u, q) + \sum_K \delta_K (\nabla q, \nabla p)_K.$$

Next, we recall the error equation for our discretization, that is implied by the consistency of the method:

$$a_\delta(\{e_u, e_p\}, \{v_h, q_h\}) = 0 \quad \forall \{v_h, q_h\} \in X_h,$$

where $e_u = u - u_h$ and $e_p = p - p_h$. The fact that this "Galerkin orthogonality" comes up in the modified bilinear form gives us an additional term in the estimator.
We start with the case $m = 1$.
Let $\{w, r\}$ be the solution of the following continuous dual problem:

$$a(\{v,q\},\{w,r\}) = (\nabla e_u, \nabla v) + (e_p, q) \quad \forall \{v,q\} \in X.$$

This corresponds to the following Stokes system:

$$\begin{aligned} -\Delta w - \nabla r &= -\Delta e_u \\ -\nabla \cdot w &= e_p. \end{aligned}$$

Now we get with $w_h = I_h w$ and $r_h = J_h r$:

$$\begin{aligned} \|\nabla e_u\|^2 + \|e_p\|^2 &= a(\{e_u, e_p\}, \{w, r\}) = \\ & a(\{e_u, e_p\}, \{w, r\}) - a_\delta(\{e_u, e_p\}), (\{w_h, r_h\}) = \\ & \underbrace{a(\{e_u, e_p\}, \{w - w_h, r - r_h\})}_{\text{I}} + \underbrace{(a - a_\delta)(\{e_u, e_p\}, \{w_h, r_h\})}_{\text{II}} \end{aligned}$$

The first term can be estimated in the usual way:

$$
\begin{aligned}
\mathrm{I} &= a(\{e_u, e_p\}, \{w - w_h, r - r_h\}) = \\
&\quad (\nabla e_u, \nabla(w - w_h) - (e_p, \nabla\cdot(w - w_h)) - (\nabla\cdot e_u, r - r_h) \\
&= (\nabla u, \nabla(w - w_h)) - (p, \nabla\cdot(w - w_h)) - (\nabla\cdot u, r - r_h) \\
&\quad - (\nabla u_h, \nabla(w - w_h)) + (p_h, \nabla\cdot(w - w_h)) + (\nabla\cdot u_h, r - r_h) \\
&= \sum_K (f, w - w_h)_K + (\Delta u_h - \nabla p, w - w_h)_K \\
&+ \int_{\partial K} (-\frac{\partial u_h}{\partial n} + p\cdot n, w - w_h)\, dS + (\nabla\cdot u_h, r - r_h)_K \\
&= \sum_K (f + \Delta_h u_h - \nabla_h p_h, w - w_h) \\
&\quad + \frac{1}{2}\sum_\Gamma [p\cdot n - \frac{\partial u_h}{\partial n}]_\Gamma \int_\Gamma (w - w_h)\, dS + (\nabla\cdot u_h, r - r_h)_K \\
&\leq C_I \left[\left\{ \sum_K h_K^2 \|f + \Delta_h u_h - \nabla_h p_h\|_K^2 \right\}^{1/2} \right. \\
&\quad \left. + \kappa \left\{ \sum_{K,\Gamma\subset K} h_K \|[p\cdot n - \frac{\partial u_h}{\partial n}]\|_\Gamma^2 \right\}^{1/2} \right] \|w\|_1 \\
&\quad + C_I \left\{ \sum_K \|\nabla\cdot u_h\|_K \right\}^{1/2} \|r\|_0
\end{aligned}
$$

For the second term we have:

$$
\begin{aligned}
\mathrm{II} &= (a - a_\delta)(\{e_u, e_p\}, \{w_h, r_h\}) \\
&= (a - a_\delta)(\{u, p\}, \{w_h, r_h\}) - (a - a_\delta)(\{u_h, p_h\}, \{w_h, r_h\}) \\
&= -(f, r_h)_\delta + c(p_h, r_h) = \sum_K \delta_K (\nabla p_h - f, \nabla r_h) \\
&\leq (\sum_K \|\nabla p_h - f\|_K^2)^{1/2} (\sum_K \delta_K \|\nabla r_h\|_K^2)^{1/2} \\
&\leq (\sum_K \|\nabla p_h - f\|_K^2)^{1/2} \|r\|
\end{aligned}
$$

We conclude using the stability estimate $\|w\|_1 + \|r\| \leq C_S \|e_u\|_1 + \|e_p\|$.
For the case $m = 0$, we first note that the H^{-1}-norm can be expressed in the following way: denoting by T the solution operator of the Laplace equation with Neumann boundary conditions, there holds:

$$\|p\|_{-1}^2 \sim (p, Tp).$$

Writing $q = Tp$ or

$$(\nabla q, \nabla \varphi) = (p, \varphi) \quad \forall \varphi \in H^1(\Omega)/R,$$

we have:

$$\|p\|_{-1} = \sup_{\varphi} \frac{(p,\varphi)}{\|\nabla \varphi\|} = \sup_{\varphi} \frac{(\nabla q, \nabla \varphi)}{\|\nabla \varphi\|} \leq \|\nabla q\| = (p,q)^{1/2}$$

and

$$\|p\|_{-1}^2 = \sup \frac{(p,\varphi)^2}{\|\nabla \varphi\|^2} \geq \frac{(p,q)^2}{\|\nabla q\|^2} = \frac{(p,q)\|\nabla q\|^2}{\|\nabla q\|^2} = (p,q)$$

Now we consider the following dual problem:

$$a(\{v,q\},\{w,r\}) = (e_u, v) + (Te_p, q) \quad \forall \{v,q\} \in X,$$

corresponding to:

$$\begin{aligned} -\Delta w - \nabla r &= e_u \\ -\nabla \cdot w &= Te_p, \end{aligned}$$

This leads to

$$\|e_u\|^2 + \|e_p\|_{-1}^2 = \mathrm{I} + \mathrm{II},$$

with terms similar to the above ones. This time we can use stronger norms to gain the higher powers of h:

$$\begin{aligned} \mathrm{I} \leq C_I \Bigg[& \left\{ \sum_K h_K^4 \|f + \Delta_h u_h - \nabla_h p_h\|_K^2 \right\}^{1/2} \\ & + \kappa \left\{ \sum_{K, \Gamma \subset K} h_K^3 |[p \cdot n - \frac{\partial u_h}{\partial n}]|^2 \right\}^{1/2} \Bigg] \|w\|_2 \\ & + C_I \left\{ \sum_K h_K^2 \|\nabla \cdot u_h\|_K \right\}^{1/2} \|r\|_1 \end{aligned}$$

and

$$\mathrm{II} \leq (\sum_K h_K^2 \|\nabla p_h - f\|^2)^{1/2} \|r\|_1$$

The stability estimate

$$\|w\|_2 + \|r\|_1 \leq C_S \, (\|e_u\| + \|e_p\|_{-1}),$$

concludes the estimation. Its validity follows from the classical regularity result of Cattabriga [16] and the fact that $\|Tp\|_1 \leq \|p\|_{-1}$. □

4.3 A posteriori estimates for the iteration error

We shortly recall the definition of the multigrid algorithm. Let $V_0 \subset V_1 \subset \cdots \subset V_L$ be a given nested hierarchy of finite element spaces. We denote by $P_l : V \to V_l$ and by $\Pi_l : V \to V_l$ the L_2 and Stokes–projections respectively. Denoting the discrete Stokes system by S, there holds $P_k \, S_l = S_k \, \Pi_k$, $k \leq l$. We further denote by $\tilde{u}_h = u_L$ a given approximate velocity field, the error $e_u = u - u_L$ of which is to be estimated. We employ analogous notations for the pressure.

Theorem 4.3. *Let the Stokes equations be H^2–regular. Then there holds for $m = 0, 1$:*

$$\|e\|_m \leq \eta(\tilde{u}_h, \tilde{p}_h)_m + \tilde{\eta}(\tilde{u}_h, \tilde{p}_h)_m$$

with η_m as above and $\tilde{\eta}_m$ defined as follows:

$$\begin{aligned}\tilde{\eta}((u_L, p_L)) \equiv \sum_{l=1}^{L} & h_l^{2-m} \|P_l(f - A_L u_L - B_L^* p_L)\| \\ & + h_l^{1-m} \|P_l(g - B_L u_L - C_L p_L)\|.\end{aligned}$$

Remark 4.6. *In the above sum "high frequency errors" are weighted differently from the lower ones. This is in accordance with the standard multigrid theory.*

Proof. We abbreviate the couples $\{u, p\}, \{v, q\}, \{w, r\}$ by x, y, z. Let z be the solution of the dual problem:

$$a(y, z) = (e, y) \quad \forall y \in V \times Q$$

Choosing $z_l = I_l z$ with some appropriate interpolation operator $I_l : X \to X_l$ and setting $z_{-1} = 0$, $a_{-1} = 0$ we have:

$$\begin{aligned}\|e\|^2 = a(e,z) &= a(e,z) - a(e,z_L) + a(e,z_L) \\ &= a(e, z - z_L) + a_L(e,z_L) + (a - a_L)(e,z_L) \\ &= a(e, z - z_L) + (a - a_L)(e,z_L) + a_L(e,z_L) \\ &\leq \eta(x_L) + a_L(e,z_L)\end{aligned}$$

The first term corresponds to the discretization error and the second one is bounded as follows:

$$\begin{aligned}a_L(e,z_L) = a_L(e, \sum_{l=0}^{L} z_l - z_{l-1}) &= \sum_{l=0}^{L} a(e, z_l - z_{l-1}) \\ &= \sum_{l=0}^{L} (f, z_l - z_{l-1}) - a_L(x_L, z_l - z_{l-1}) \\ &= \sum_{l=0}^{L} (f, z_l - z_{l-1}) - (S_l \Pi_l x_L, z_l - z_{l-1}) \\ &= \sum_{l=0}^{L} (f, z_l - z_{l-1}) - (P_l S_L x_L, z_l - z_{l-1}) \\ &= \sum_{l=0}^{L} (P_l(P_L f - S_L x_L), z_l - z_{l-1}) \\ &\leq \tilde{C_I} \sum_{l=1}^{L} \Big\{ h_l^{2-m} \|P_l(f - A_L u_L - B_L^* p_L)\| \\ &+ h_l^{1-m} \|g - B_L u_L - C_L p_L\|)(\|w\|_{2-m} + \|r\|_{1-m} \Big\}.\end{aligned}$$

This finally gives:

$$\|e\| \leq \eta(u_L, p_L) + \tilde{\eta}(u_L, p_L).$$

□

Remark 4.7. *In the case of inherited bilinear forms, the residual $d_l = P_l d_L$ with $d_L = P_L f - S_L x_L$ can be identified with the usual multigrid residuals, see [7].*
In each case the coefficient vector $\hat{d_L}$ of $d_L = \sum_i \hat{d_L}^i \varphi_L^1$ is easily calculated:

$$\hat{d_L} = M^{-1}(b - S_L x_L)$$

and $\hat{d}_l$ is just the restriction of $\hat{d}_L$.

4.4 Numerical tests for the error estimators

This section illustrates the behavior of the error estimators. We will first present some result for the discretization error estimator. In order to compare the estimated with the true error we compute the following exact solution:

$$u = (\sin(\pi x)\cos(\pi y), -\cos(\pi x)\sin(\pi y)),$$
$$p = xy.$$

Figure 4.1 shows the distribution of the true and estimated errors. Darker gray levels correspond to higher values.

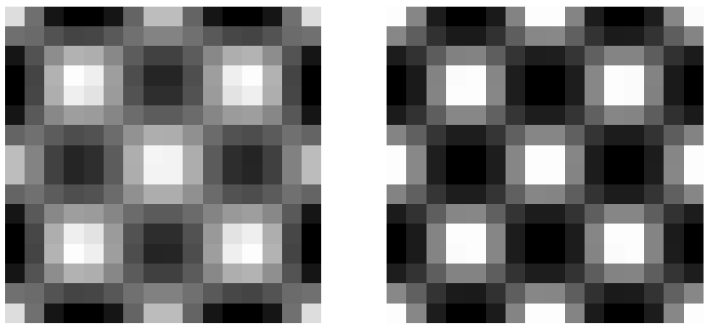

Figure 4.1: True (left) and estimated (right) element-wise errors, Test1

It is clear from figure 4.1 that the qualitative behavior is satisfactory. The structure of the error is well represented by the estimator. In order to obtain a good quantitative upper bound for the error it is necessary to calibrate the constants in front of the different terms of the estimator. It is clear from the analysis in the preceding section that there are different interpolation as well as stability constants involved in the different terms. Further, there might be cancelations in the terms that are estimated by the Cauchy inequality.

For these reasons we prefer estimating these constants by preliminary computations on coarse grids. The estimator which we obtain in this way leads to a reasonable efficiency. In our computations for various test problems, the quotient of estimated to true error was about 2 in most cases and about 5 in the worst case.
In order to demonstrate the behavior of the estimator for different solutions, we next consider another given function with a different behavior:

$$u = ((1-x^2)^2(1-y^2)y, -(1-y^2)^2(1-x^2)x),$$
$$p = xy + x^3y^3.$$

These tables show that the qualitative structure of the error is well repre-

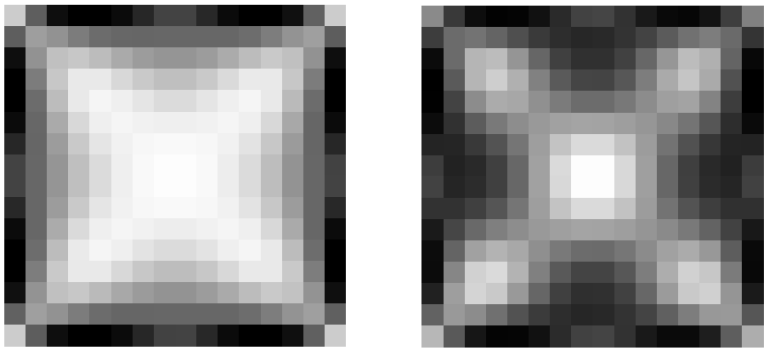

Figure 4.2: True (left) and estimated (right) element-wise errors,Test2

sented by the estimator. This implies that the mesh would be refined in the correct regions. In both cases the error is slightly overestimated with a factor of 1.3 for the first test and 1.8 for the second one.
To get insight into the behavior of the different terms of η, we show the distribution of the individual contribution in figure 4.3. For this, we choose the regularized driven cavity as test problem. In order to get smooth solutions, the boundary condition on the top is chosen to be a polynomial which vanishes at the corners in contrast to the standard lid driven cavity with

Figure 4.3: Contribution from "div-","jump-" and "residual-" term

discontinuous boundary data. This clearly shows that the different terms do not have the same behavior. Therefore no one of them should be skipped. We finally present a locally refined grid after 3 iterations of the adaptive algorithm using the a posteriori estimator presented above.

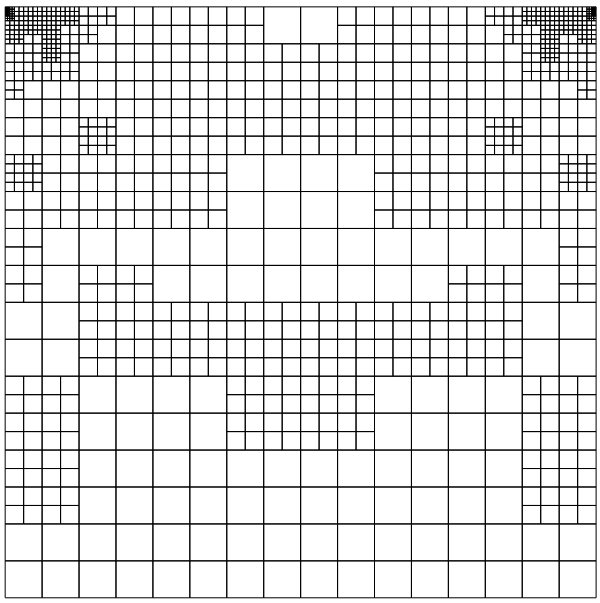

Figure 4.4: Locally refined grid for the lid driven cavity

We now turn our attention to the estimator for the iteration error. In our first numerical tests, we show that the stopping criterion gives acceptable results independently of h and the exact solution u. We therefore calculate two different test cases on a rather coarse and a fine grid with 256 and 16384 elements respectively. We choose the same exact solutions as before. We present the results for $m = 0$, estimating the L_2-norm of the velocity error, since this case is clearly more sensitive to the constants involved in the estimator. In this case, the pressure error is measured in the H^{-1}-norm. We replace this norm by a weighted L_2-norm, $\|h\,(p - p_h)\|_{L_2}$. The error is thus slightly underestimated, since

$$\|p\|_{-1} = \sup_{\varphi} \frac{(p,\varphi)}{\|\nabla\varphi\|} \geq \sup_{\varphi_h} \frac{(p,\varphi_h)}{\|\nabla\varphi_h\|} \geq \|h\,p\|,$$

by an inverse estimate.

In tables 4.1 and 4.2 we give the values for the error e and the two estimators η and $\tilde{\eta}$ during 8, respectively 10 iterations. The last column gives the

	# Elements 256				
it	e	$\eta + \tilde{\eta}$	η	$\tilde{\eta}$	q
0	$1.24e+00$	$9.82e+00$	$7.15e-01$	$9.10e+00$	0.13
1	$3.87e-01$	$8.08e-01$	$2.47e-01$	$5.61e-01$	0.48
2	$2.60e-01$	$3.23e-01$	$2.51e-01$	$7.16e-02$	0.81
3	$2.04e-01$	$2.65e-01$	$2.47e-01$	$1.74e-02$	0.77
4	$1.96e-01$	$2.51e-01$	$2.46e-01$	$4.73e-03$	0.78
5	$1.98e-01$	$2.47e-01$	$2.46e-01$	$1.29e-03$	0.80
6	$2.00e-01$	$2.47e-01$	$2.46e-01$	$1.07e-03$	0.81
7	$2.01e-01$	$2.46e-01$	$2.46e-01$	$3.43e-04$	0.82
8	$2.01e-01$	$2.46e-01$	$2.46e-01$	$1.77e-04$	0.82

Table 4.1: Stokes, Test 1

quotient q of the estimated to the true error, which measures the efficiency of the estimator.

In table 4.1 we present the convergence history for Test1 on a coarse mesh

with 256 elements. Table 4.2 shows the behavior for a fine grid with 16384 elements. Comparing the quotient of the estimated to the true error illus-

	# Elements 16384				
it	e	$\eta + \tilde{\eta}$	η	$\tilde{\eta}$	q
0	$1.54e+00$	$1.10e+01$	$1.99e-01$	$1.08e+01$	0.14
1	$6.61e-01$	$5.27e-01$	$5.19e-02$	$4.75e-01$	1.25
2	$7.57e-02$	$1.21e-01$	$1.77e-02$	$1.03e-01$	0.62
3	$3.03e-02$	$3.89e-02$	$7.05e-03$	$3.19e-02$	0.78
4	$1.10e-02$	$1.21e-02$	$2.33e-03$	$9.85e-03$	0.90
5	$4.40e-03$	$4.72e-03$	$1.48e-03$	$3.23e-03$	0.93
6	$2.15e-03$	$2.14e-03$	$1.07e-03$	$1.06e-03$	1.00
7	$1.22e-03$	$1.39e-03$	$1.03e-03$	$3.57e-04$	0.88
8	$9.31e-04$	$1.14e-03$	$1.02e-03$	$1.20e-04$	0.81
9	$8.59e-04$	$1.06e-03$	$1.02e-03$	$4.07e-05$	0.81
10	$8.36e-04$	$1.03e-03$	$1.02e-03$	$1.39e-05$	0.81

Table 4.2: Stokes, Test 1

trates that the constants are independent of the mesh parameter h. We also note that we would have stopped the multigrid iteration after 3 respectively 8 steps.

In tables 4.3 and 4.4 we present the results for the second test case. This time we would have stopped the multigrid iteration after 3 respectively 9 steps. Comparing the actual error in these steps to the final one shows that the stopping criterion is reliable and efficient.

Finally, we demonstrate the the interaction of the two estimators by reporting the iteration history of the complete adaptive algorithm for the above test cases. The tolerance is increased during the iteration in order to achieve a smoother transition. We stop the iteration if the estimated iteration order is ten times less than the estimated discretization error. This insures that the main part of the error results from the mesh. In this way we are sure to refine the grid according to the discretization error. The tolerance for the first test case is 0.002, whereas for the second case, we we choose 0.001. The

	# Elements 256				
it	e	$\eta + \tilde{\eta}$	η	$\tilde{\eta}$	q
0	$7.71e-01$	$2.81e+00$	$1.02e-01$	$2.70e+00$	0.27
1	$1.61e-01$	$3.72e-01$	$1.10e-01$	$2.61e-01$	0.43
2	$7.90e-02$	$1.34e-01$	$9.83e-02$	$3.65e-02$	0.59
3	$5.79e-02$	$1.04e-01$	$9.79e-02$	$6.11e-03$	0.56
4	$5.04e-02$	$1.00e-01$	$9.79e-02$	$2.10e-03$	0.50
5	$4.86e-02$	$9.84e-02$	$9.79e-02$	$4.39e-04$	0.49
6	$4.82e-02$	$9.82e-02$	$9.79e-02$	$2.22e-04$	0.49
7	$4.81e-02$	$9.81e-02$	$9.80e-02$	$1.17e-04$	0.49
8	$4.81e-02$	$9.80e-02$	$9.80e-02$	$6.55e-05$	0.49

Table 4.3: Stokes, Test 2

	# Elements 16384				
it	e	$\eta + \tilde{\eta}$	η	$\tilde{\eta}$	q
0	$1.31e+00$	$3.23e+00$	$7.35e-04$	$3.23e+00$	0.40
1	$1.05e-01$	$2.45e-01$	$2.53e-02$	$2.20e-01$	0.43
2	$3.75e-02$	$7.27e-02$	$7.45e-03$	$6.53e-02$	0.52
3	$1.37e-02$	$2.74e-02$	$2.91e-03$	$2.45e-02$	0.50
4	$6.26e-03$	$1.08e-02$	$1.25e-03$	$9.59e-03$	0.58
5	$2.85e-03$	$4.38e-03$	$6.00e-04$	$3.78e-03$	0.65
6	$1.34e-03$	$1.95e-03$	$4.61e-04$	$1.49e-03$	0.68
7	$6.86e-04$	$1.00e-03$	$4.13e-04$	$5.92e-04$	0.68
8	$3.81e-04$	$6.41e-04$	$4.07e-04$	$2.34e-04$	0.59
9	$2.63e-04$	$4.98e-04$	$4.05e-04$	$9.29e-05$	0.53
10	$2.23e-04$	$4.41e-04$	$4.04e-04$	$3.66e-05$	0.51

Table 4.4: Stokes, Test 2

values of the estimators η and $\tilde{\eta}$ as well as the true error and the efficiency quotient q are given in tables 4.5 and 4.6. We can see from tables 4.5 and 4.6 that in most cases we need one multigrid iteration to get the iteration er-

N	it	η	$\tilde{\eta}$	$\eta + \tilde{\eta}$	e	q
16	1	$2.57e-01$	$5.48e-02$	$3.12e-01$	$3.16e-01$	1.01
16	2	$2.56e-01$	$1.51e-02$	$2.71e-01$	$3.18e-01$	1.17
64	1	$1.34e-01$	$1.00e-02$	$1.44e-01$	$8.29e-02$	0.57
256	1	$4.19e-02$	$1.57e-03$	$4.35e-02$	$2.09e-02$	0.48
1024	1	$1.16e-02$	$2.64e-04$	$1.19e-02$	$5.20e-03$	0.43
4036	1	$3.22e-03$	$6.21e-05$	$3.28e-03$	$1.30e-03$	0.39
4708	1	$2.76e-03$	$1.21e-05$	$2.77e-03$	$1.09e-03$	0.39
14740	1	$1.02e-03$	$1.00e-06$	$1.02e-03$	$4.69e-04$	0.45

Table 4.5: Adaptive Iteration, Test 1

N	it	η	$\tilde{\eta}$	$\eta + \tilde{\eta}$	e	q
16	1	$7.29e-02$	$3.18e-02$	$1.04e-01$	$8.29e-02$	0.79
16	2	$7.25e-02$	$1.41e-02$	$8.67e-02$	$8.57e-02$	0.98
16	3	$7.23e-02$	$6.48e-03$	$7.88e-02$	$8.72e-02$	1.10
64	1	$3.39e-02$	$9.28e-03$	$4.32e-02$	$2.04e-02$	0.47
64	2	$3.39e-02$	$2.85e-03$	$3.67e-02$	$2.14e-02$	0.58
256	1	$1.12e-02$	$2.01e-03$	$1.32e-02$	$5.47e-03$	0.41
256	2	$1.12e-02$	$4.76e-04$	$1.16e-02$	$5.49e-03$	0.47
1000	1	$3.56e-03$	$4.02e-04$	$3.97e-03$	$1.55e-03$	0.39
1000	2	$3.56e-03$	$1.73e-04$	$3.74e-03$	$1.47e-03$	0.39
3232	1	$1.40e-03$	$7.18e-05$	$1.47e-03$	$7.36e-04$	0.50
4252	1	$8.80e-04$	$1.34e-05$	$8.94e-04$	$4.64e-04$	0.51

Table 4.6: Adaptive Iteration, Test 2

ror sufficiently small. Only the first adaptive iterations need more multigrid cycles.

In figures 4.5 and 4.6 we show the locally refined meshes produced by the last iterations of the algorithm. As can be seen from figure 4.5, the algorithm tends to refine the boundary in the fourth adaptive step. This is due to the fact, that the stabilization introduces a higher error near the boundary (see

Blum, [10]). A certain global accuracy is needed for the estimator to take this effect into account.

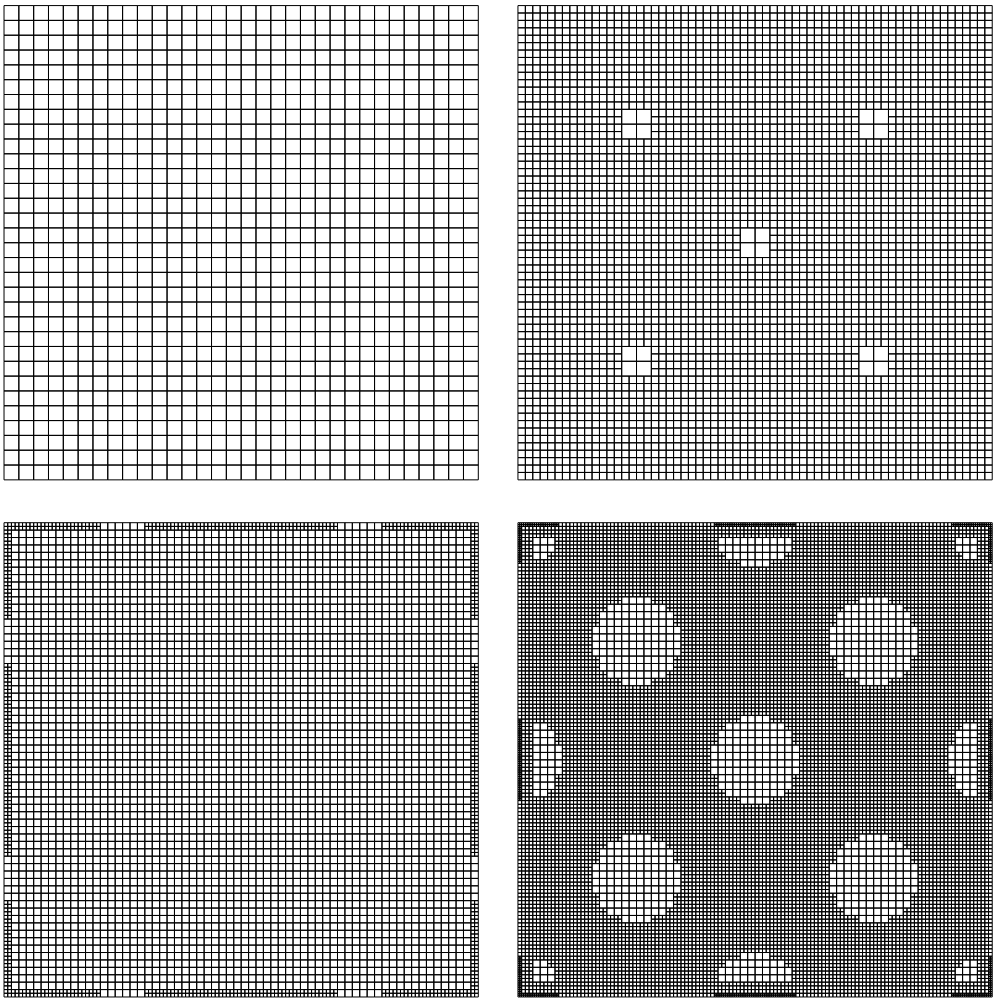

Figure 4.5: Mesh during the 4 last iterations, Test1

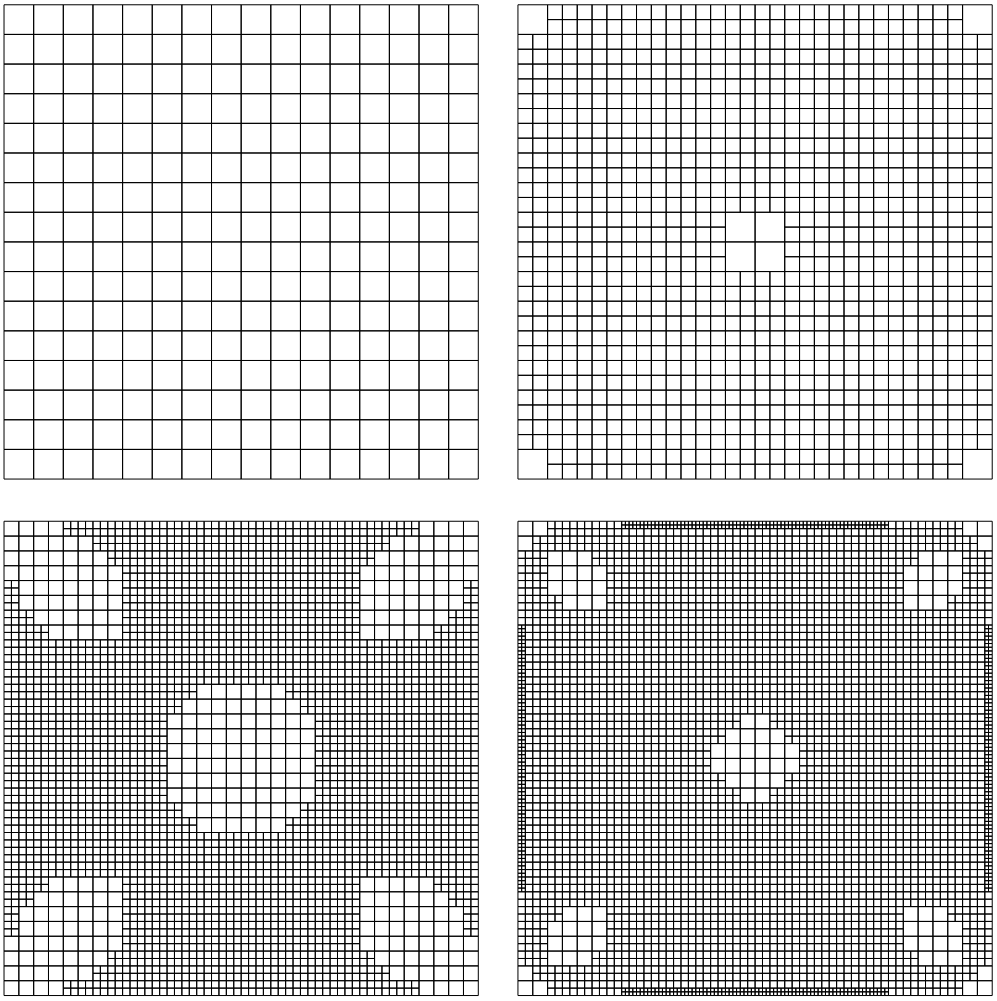

Figure 4.6: Mesh during the 4 last iterations, Test2

Chapter 5

Discretization of the Stokes equations on anisotropic grids

For moving boundary problems, the tracking of the boundary may naturally lead to extended elements as, e.g., in the piston problem. The use of stretched meshes is also a common praxis in many other applications. One example is the resolution of laminar boundary layers in viscous flows.
It is important to know, whether certain discretizations are stable on these grids, in the sense that the stability constants do not depend on the element aspect ratio. For the incompressible Navier-Stokes equations, this means that the classical inf-sup condition should hold independently of the stretching factor. From the theoretical point of view, however, it is not clear, if this holds, since the classical proofs require the triangulation to be quasi-uniform. This assumption is required for the use of the nodal interpolation operator as well as Clément's generalization to non-smooth functions [20].
We develop an alternative interpolation operator that takes the direction of the stretching into account. It is shown to yield a stable interpolation in $H^1(\Omega)$. This construction is then applied to stabilized low order discretizations of the Stokes equations and it is shown that the correct choice of the discrete bilinear form yields a stable approximation.
The stability result can be used to derive error estimates which distinguish between the element extensions in the different directions. We present an estimation in the gradient norm for the velocities and the L_2-norm for the

pressure. Our goal is only to give an indication of how to use anisotropic grids when physical quantities vary strongly only in some directions, rather than to develop a complete theory. The issue of L_2-error estimates is left out, since the duality argument usually used for this does not yield optimal result on anisotropic meshes.

We proceed in the following way: First, only two dimensional orthogonal grids are considered. They are also required to have moderate variations of the aspect ratio. This is usually no restriction, since any practical grid refinement algorithm will lead to meshes fulfilling this condition. Whereas the generalization to the three dimensional case is straightforward, the treatment of non-orthogonal grids is more complicated. A simple definition of an interpolation operator via coordinate transformation is given in the latter case.

In the context of discretizations which rely on weighted stabilization techniques the question of the correct choice of the stabilization parameter arises. For the Q^1/Q^1-element on non-uniform elements which are aligned to the Cartesian coordinate axes and have width h_x and h_y in x- and y-direction respectively, there are at least three possibilities of stabilization:

$$\begin{aligned} c_{\mathrm{I}}(p,q) &= \sum_K |K|(\nabla p, \nabla q)_K, \\ c_{\mathrm{II}}(p,q) &= \sum_K \operatorname{diam}(K)^2(\nabla p, \nabla q)_K, \qquad \text{and} \\ c_{\mathrm{III}}(p,q) &= \sum_K \left\{ h_x^2(\partial_x p, \partial_x q)_K + h_y^2(\partial_y p, \partial_y q)_K \right\}. \end{aligned}$$

The first one is related to the MINI-element, see the discussion in Remark 4.4. In the second definition the measure of the element as mesh-parameter is replaced by the diameter. The choice of c_{III} is motivated by the requirement that the stabilizing term has to be bounden in L_2 (see Remark 4.1). It has been pointed out in [9] that only c_{III} yields good numerical results on anisotropic meshes, thus indicating that the MINI-element (in a generalized form for quadrilateral elements) cannot be expected to be robust in the above sense.

We will finally present numerical tests for the different stabilization techniques to confirm our theoretical results.

In order to invert the discrete system of equations we propose a smoother for the multigrid iteration which is locally adapted to the stretching of the elements.

5.1 Interpolation operator on anisotropic grids

We shortly recall the notations introduced before: We denote the elements of a triangulation $\mathcal{T}$ by K, the edges by Γ and the nodes by N_i, $1 \leq i \leq N$. For a given element K the set of neighboring elements is abbreviated by $\mathcal{N}(K)$. In a similar way $\mathcal{N}(N_i)$ stands for the elements and $\mathcal{E}(N_i)$ for the set of edges containing N_i.

We will first consider the case of two dimensional orthogonal grids. This means that all edges of the triangulation are parallel to the coordinate axes. Generalizations to curved and three dimensional meshes will be discussed later.

We have to make a further assumption on the mesh in order to avoid large variations in space of the element aspect ratios and of the direction of anisotropic refinement. For this, we define an element-wise constant mesh function which measures the directional extensions of an element K:

$$h_x^K = \sum_{N_i \in K} \frac{\partial \varphi_i}{\partial x}(M_K),$$

where φ_i denotes the Lagrange basis function belonging to the node N_i and M_K is the midpoint of the element. $h_y^K$is defined in the same way. The following assumption is crucial for our analysis:

There is a moderate constant κ such that:

$$\begin{aligned} &\frac{1}{\kappa} h_x^L \leq h_x^K \leq \kappa h_x^L \text{ and} \\ &\frac{1}{\kappa} h_y^L \leq h_y^K \leq \kappa h_y^L \quad \forall L \in \mathcal{N}(K). \end{aligned} \tag{5.1}$$

The constant κ will enter our estimates. A grid with large κ looks for instance like the one shown in figure 5.1.

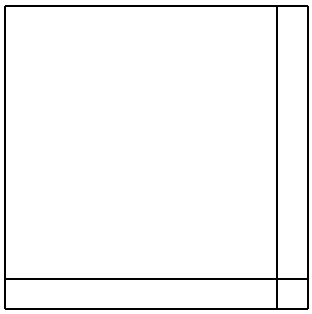

Figure 5.1: Grid with large κ

The element-wise aspect ratio $\sigma_K = \max(\frac{h_x^K}{h_y^K}, \frac{h_y^K}{h_x^K})$ is however allowed to be large.

The interpolation operator we will define below is an adaptation of the construction proposed by Scott and Zhang in [47] for the interpolation of rough functions on quasi-uniform meshes.

To each node i we associate an edge Γ_i by:

$$\Gamma_i \equiv \mathrm{ArgMax}_{\Gamma \in \mathcal{E}(N_i)} |\Gamma|. \tag{5.2}$$

If the maximum is not unique we may pick any of the largest edges. We further associate to i a linear function ψ_i, defined by:

$$\int_{\Gamma_i} \psi_i \varphi_j \, ds = \delta_{i,j} \quad \forall 1 \leq j \leq N.$$

For a given function $u \in H_0^1(\Omega)$ we define $I_h : H_0^1(\Omega) \mapsto V_h$ by:

$$I_h u \equiv \sum_i \int_{\Gamma_i} u \psi_i \, ds \, \varphi_i. \tag{5.3}$$

Taking special care of the boundary nodes, we can achieve that I_h preserves boundary values as in [47] by defining the values of I_h for the boundary nodes as integrals over element edges which also belong to the boundary, provided that the the anisotropic refinement is orthogonal to the boundary. This is not a severe restriction since strong gradients of the solution are expected to be orthogonal to the boundary.

Since $I_h \varphi_j = \varphi_j \, \forall j$ this operator leaves the discrete functions unchanged. It is shown in [47] that this definition leads to a stable optimal-order interpolation

on quasi-uniform grids. We show in the following that our special choice of edges implies the same properties under assumption (5.1) even on stretched meshes.

Theorem 5.1. *Let the mesh $\mathcal{T}$ be orthogonal and satisfy condition (5.1). Then, there holds,*

$$\|\nabla I_h u\| \leq C\|\nabla u\|, \tag{5.4}$$

with a constant C only depending on κ.

Proof. We distinguish between low and high aspect ratio elements. In the first case we can directly apply the results of Scott and Zhang, because our assumption on the grid implies that the aspect ratio stays low in the neighborhood of a given element.
Let us consider the case of a high aspect ratio element, which is, without loss of generality, supposed to be stretched in x-direction as shown in figure 5.2.

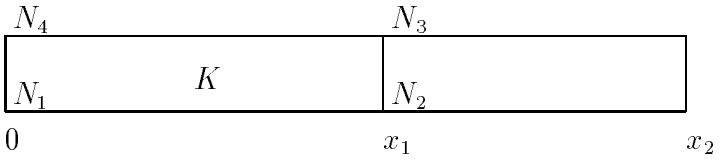

Figure 5.2: Stretched element K

We denote by h_x and h_y the widths of K in x- and y-direction. By (5.1) the surrounding elements are stretched in the same direction. We can thus suppose that the interpolation is given by $I_h u|_K = \sum_i \alpha_i \varphi_i$, where φ_i is the basis function belonging to N_i (c.f. figure 5.2). The coefficients α are given by:

$$\alpha_1 = \int_0^{x_1} \psi_1(\xi) u(\xi, 0)\, d\xi \quad \alpha_2 = \int_{x_1}^{x_2} \psi_2(\xi) u(\xi, 0)\, d\xi$$
$$\alpha_3 = \int_{x_1}^{x_2} \psi_2(\xi) u(\xi, h_y)\, d\xi \quad \alpha_4 = \int_0^{x_1} \psi_1(\xi) u(\xi, h_y)\, d\xi,$$

where $x_1 = h_x$ and $|x_2 - x_1| \leq \kappa h_x$.
Now we estimate the x- and y-derivatives separately by using the fact that

$$\|\partial_y u_h\|^2 \sim |K| \, (\frac{|\alpha_4 - \alpha_1|^2}{h_y^2} + \frac{|\alpha_3 - \alpha_2|^2}{h_y^2})$$

and

$$\|\partial_x u_h\|^2 \sim |K| \, (\frac{|\alpha_3 - \alpha_4|^2}{h_x^2} + \frac{|\alpha_2 - \alpha_1|^2}{h_x^2}).$$

Let us estimate the first term for the y-derivative:

$$\begin{aligned} \alpha_4 - \alpha_1 &= \int_0^{x_1} \psi_1 u(\xi, 0) \, d\xi - \int_0^{x_1} \psi_1 u(\xi, h_y) \, d\xi \\ &= \int_0^{h_x} \int_0^{h_y} \psi_1 \partial_y u(\xi, \eta) \, d\eta d\xi, \end{aligned}$$

using $\|\psi_1\|^2_{L^\infty(0,h_x)} \leq 1/h_x$ we get

$$|\alpha_4 - \alpha_1| \leq \frac{1}{h_x} \|\partial_y u\|_{L_1(K)},$$

such that

$$\frac{|\alpha_4 - \alpha_1|^2}{h_y^2} \leq \frac{1}{h_x^2 h_y^2} |K| \|\partial_y u\|^2.$$

Estimating the other term similarly leads to:

$$\|\partial_y u_h\|_K \leq \sqrt{2} \|\partial_y u\|_{\mathcal{N}(K)}.$$

Now we turn our attention to the x-derivative. We set $\lambda = 1/\kappa$ and note that $\frac{x_2 - x_1}{x_1} = \lambda$.

$$\begin{aligned} \alpha_3 - \alpha_4 &= \int_0^{x_1} \psi_1(\xi) u(\xi, 0) \, d\xi - \int_{x_1}^{x_2} \psi_2(\xi) u(\xi, 0) \, d\xi \\ &= \int_0^{x_1} \psi_1(\xi) u(\xi, y) \, d\xi - \int_{x_1}^{x_2} \psi_2(\xi) u(\xi, y) \, d\xi \\ &- \int_0^{x_1} \psi_1(\xi) \int_0^y \partial_y u(\eta, \xi) \, d\eta d\xi + \int_{x_1}^{x_2} \psi_1(\xi) \int_0^y \partial_y u(\eta, \xi) \, d\eta d\xi \end{aligned}$$

transformation of the second integral yields:

$$
\begin{aligned}
&= \int_0^{x_1} \psi_1(\xi) u(\xi, y)\, d\xi - \int_0^{x_1} \psi_1(\xi) u(x_1 + \lambda\xi, y)\, d\xi \\
&+ \int_0^{x_1} (\psi_1(\xi) - \psi_2(x_1 + \lambda\xi)\lambda) u(x_1 + \lambda\xi, y)\, d\xi \\
&- \int_0^{x_1} \psi_1(\xi) \int_0^{y} \partial_y u(\eta, \xi)\, d\eta d\xi + \int_{x_1}^{x_2} \psi_1(\xi) \int_0^{y} \partial_y u(\eta, \xi)\, d\eta d\xi \\
&= \mathrm{I} + \mathrm{II} + \mathrm{III}\,.
\end{aligned}
$$

Next we estimate the different terms:

$$
\begin{aligned}
\mathrm{I} &= \int_0^{x_1} \psi_1(\xi) u(\xi, y)\, d\xi - \int_0^{x_1} \psi_1(\xi) u(x_1 + \lambda\xi, y)\, d\xi \\
&= \int_0^{x_1} \psi_1(\xi) \int_{\xi}^{x_1 + \lambda\xi} \partial_x u(s, y)\, ds d\xi
\end{aligned}
$$

integrating with respect to y gives:

$$
\begin{aligned}
h_y\, |\,\mathrm{I}\,| &\le |K|^{1/2} \|\partial_x u\|_{\mathcal{N}(K)} \\
\frac{|K|}{h_x^2}\, |\,\mathrm{I}\,|^2 &\le \|\partial_x u\|_{\mathcal{N}(K)}.
\end{aligned}
$$

For the second term we have

$$
\mathrm{II} = \int_0^{x_1} (\psi_1(\xi) - \psi_2(x_1 + \lambda\xi)\lambda) u(x_1 + \lambda\xi, y)\, d\xi = 0,
$$

since $\psi_1(\xi) - \psi_2(x_1 + \lambda\xi)\lambda$ is a linear function that vanishes at the end points. Finally the last term is estimated as follows:

$$
|\,\mathrm{III}\,| \le \frac{2}{h_x}\, |K| \|\partial_y u\|_{\mathcal{N}(K)}
$$

which shows that:

$$
\frac{|K|}{h_x^2}\, |\,\mathrm{III}\,|^2 \le \frac{h_y^2}{h_x^2}\, \|\partial_y u\|^2_{\mathcal{N}(K)}.
$$

Putting things together we obtain:

$$
\|\partial_x u_h\| \le C \|\nabla u\|,
$$

which concludes the proof. □

5.2 Stability of the stabilized Q^1/Q^0-element

We prove the stability of the bilinear velocity, constant pressure element on arbitrarily stretched meshes that fulfill condition (5.1) of the preceding section. This discretization has the following form:

$$\begin{aligned} (\nabla u_h, \nabla v) - (p_h, \nabla \cdot v) &= (f, v) \quad \forall v \in V_h \\ (\nabla \cdot u_h, q) + c_0(p_h, q) \quad &= 0 \qquad\quad \forall q \in Q_h, \end{aligned} \tag{5.5}$$

with $V_h = Q^1$ and $Q_h = Q^0$. Denoting by h_Γ^- the mean width in the direction orthogonal to Γ of the elements adjacent to Γ, the stabilization is given by:

$$c_0(p_h, q_h) \equiv \sum_\Gamma h_\Gamma^- \int_\Gamma [p]_\Gamma [q]_\Gamma \, ds.$$

Theorem 5.2. *Let the mesh $\mathcal{T}$ fulfill assumption (5.1). Then, there exists a constant γ independent of the aspect ratio, such that:*

$$\inf_{p_h \in Q_h} \left\{ \sup_{v_h \in V_h} \frac{(p_h, \nabla \cdot v_h)}{\|\nabla v_h\| \|p_h\|} + \frac{c_0(p_h, p_h)^{1/2}}{\|p_h\|} \right\} = \gamma > 0. \tag{5.6}$$

Proof. First we construct a slightly smaller pressure space by requiring the pressure to be constant on couples of elements in the direction of the stretching (c.f. figure 5.3).

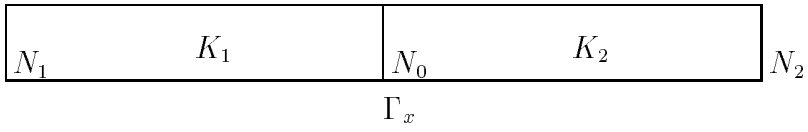

Figure 5.3: Domain of constant pressure value

Eventually we cluster more than two elements if the mesh has an odd number of elements in the considered direction. Denoting by Γ_x the edges of the macro elements in x-direction, we modify the interpolation operator in order to assure that

$$\int_{\Gamma_x} I_h u \, ds = \int_{\Gamma_x} u \, ds \quad \forall \, \Gamma_x. \tag{5.7}$$

This is done in the following way: Let $\tilde{\mathcal{T}}$ be the mesh composed of the macro elements $K = K_1 \cup K_2$. We define the values of the interpolation operator $\tilde{I}_h u$ at the corresponding nodes as before. Considering the additional basis functions in a hierarchical way, the coefficient α_0 of φ_0 is given as a correction,

$$\alpha_0 = \frac{1}{\int_{\Gamma_x} \varphi_0 \, ds} \, (\int_{\Gamma_x} u \, ds - \alpha_1 \int_{\Gamma_x} \varphi_1 \, ds - \alpha_2 \int_{\Gamma_x} \varphi_2 \, ds),$$

in order to satisfy (5.7). The H^1-stability of this operator can be seen as before since $h_x^2 \, \|\partial_x u\|^2 \sim |K| \, (|\alpha_1 - \alpha_2|^2 + |\alpha_0|^2)$ and α_0 is again a weighted difference of edge integrals in the long direction.
For a macro-element-wise constant pressure $\bar{p}_h$, let $v \in H_0^1(\Omega)$ be a function with $(\bar{p}_h, \nabla \cdot v) \geq \gamma \|\bar{p}_h\|^2$ and $\|\nabla v\| \leq \|\bar{p}_h\|$. The function v exists by virtue of the LBB-condition for the continuous problem (see [25]). We set $v_h = \tilde{I}_h v$, where $\tilde{I}_h$ denotes the modified interpolation operator. This yields:

$$\begin{aligned}
(\bar{p}_h, \nabla \cdot v_h) &= (\bar{p}_h, \nabla \cdot v) + (\bar{p}_h, \nabla \cdot (v_h - v)) \\
&\geq \gamma \|\bar{p}_h\|^2 - \sum_{\Gamma \in \tilde{\mathcal{T}}} [\bar{p}_h]_\Gamma \int_\Gamma (v_h - v) \, ds \\
&\geq \gamma \|\bar{p}_h\|^2 - \left(\sum_{\Gamma_y} |K| [\bar{p}_h]_{\Gamma_y}^2 \right)^{1/2} \left(\sum_{\Gamma_y} |K|^{-1} | \int_{\Gamma_y} (v_h - v) \, ds|^2 \right)^{1/2}
\end{aligned}$$

due to (5.7) and the Cauchy inequality. We proceed by applying the following estimate:

$$\int_{\Gamma_y} (v_h - v) \, ds \leq C \, \|\nabla (v - v_h)\|_{L_1(\mathcal{N}(K))}. \tag{5.8}$$

Note that the first integral in (5.8) is along the short edges. We can therefore apply the Bramble–Hilbert lemma. The analogue estimate for integrals over Γ_x does not hold independently of the aspect ratio. We obtain:

$$\begin{aligned}
&(\bar{p}_h, \nabla \cdot v_h) \\
&\quad \geq \gamma \|\bar{p}_h\|^2 - \left(\sum_{\Gamma_y} h_x \int_{\Gamma_y} [\bar{p}_h]^2 \, ds \right)^{1/2} \left(\sum_{\Gamma_y} |K|^{-1} \|\nabla (v_h - v)\|_{L_1(\mathcal{N}(K))}^2 \right)^{1/2} \\
&\quad \geq \gamma \|\bar{p}_h\|^2 - \left(\sum_{\Gamma_y} h_x \int_{\Gamma_y} [\bar{p}_h]^2 \, ds \right)^{1/2} \|\nabla v\|,
\end{aligned}$$

where we used th H^1-stability of $\tilde{I}_h$.
For a given element-wise constant pressure p_h we take $\bar{p}_h$ as the projection on the macro-element-wise constant space. We estimate the interpolation error by:

$$\|p_h - \bar{p}_h\|^2 \leq \sum_{\Gamma_x} \int_{\Gamma_x} h_y [\bar{p}_h]^2 \, ds. \tag{5.9}$$

Using the notation $p_1 = p|K_1$, $p_2 = p|K_2$ with the numbering of figure 5.3, (5.9) is proven by:

$$\begin{aligned} \|p_h - \bar{p}_h\|^2 &\sim |K| \left\{ (p_1 - \frac{p_1 + p_2}{2})^2 + (p_2 - \frac{p_1 + p_2}{2})^2 \right\} \\ &= |K| \, (p_1 - p_2)^2 = h_y \int_{\Gamma_x} [p_h]_{\Gamma_x}. \end{aligned}$$

Applying (5.9) we obtain:

$$\begin{aligned} (p_h, \nabla \cdot v_h) &= (\bar{p}_h, \nabla \cdot v_h) + (p_h - \bar{p}_h, \nabla \cdot v_h) \\ &\geq \gamma \|\bar{p}_h\|^2 - \left[\left(\sum_{\Gamma_y} h_x \int_{\Gamma_y} [\bar{p}_h]^2 \, ds \right)^{1/2} - \left(\sum_{\Gamma_x} h_y \int_{\Gamma_x} [\bar{p}_h]^2 \, ds \right)^{1/2} \right] \|\nabla v\| \\ &\geq \gamma \|\bar{p}_h\|^2 - c_0 (p_h, p_h)^{1/2} \|\nabla v\| \\ &\geq \left(\gamma \|\bar{p}_h\| - c_0 (p_h, p_h)^{1/2} \right) \|\nabla v\|. \end{aligned}$$

Dividing by $\|p_h\|$ and $\|\nabla v_h\|$ yields (5.6). □

5.3 Stability of the stabilized Q^1/Q^1-element

We examine the stability of the bilinear velocity–bilinear pressure element on stretched grids. This discretization reads:

$$\begin{aligned} (\nabla u_h, \nabla v) - (p_h, \nabla \cdot v) &= (f, v) \quad && \forall v \in V_h \\ (\nabla \cdot u_h, q) + c_1(p_h, q) \quad &= 0 && \forall q \in Q_h, \end{aligned} \tag{5.10}$$

with $V_h = Q^1$ and $Q_h = Q^1$. The stabilization is given by:

$$c_1(p_h, q_h) \equiv \sum_K \left\{ h_x^2 (\partial_y p_h, \partial_y q_h)_K^2 + h_y^2 (\partial_y p_h, \partial_y q_h)_K^2 \right\}.$$

Theorem 5.3. *Let the mesh $\mathcal{T}$ fulfill assumption (5.1). Then, there exists a constant γ independent of the aspect ratio, such that:*

$$\inf_{p_h \in Q_h} \left\{ \sup_{v_h \in V_h} \frac{(p_h, \nabla \cdot v_h)}{\|\nabla v_h\| \|p_h\|} + \frac{c_1(p_h, p_h)^{1/2}}{\|p_h\|} \right\} = \gamma > 0. \tag{5.11}$$

For the proof of this theorem we need the following estimates:

Lemma 5.1. *Let $\bar{p}_h$ be the L_2-projection of the bilinear p_h on the constants. There holds:*

$$\|p_h\| \leq \|\bar{p}_h\| \tag{5.12}$$

$$c_0(\bar{p}_h, \bar{p}_h) \leq c_1(p_h, p_h) \tag{5.13}$$

$$\|p_h - \bar{p}_h\|^2 \leq c_1(p_h, p_h). \tag{5.14}$$

Proof. The first estimate is inherent for the L_2-projection.
For (5.13) we adopt the numbering of figure 5.4.

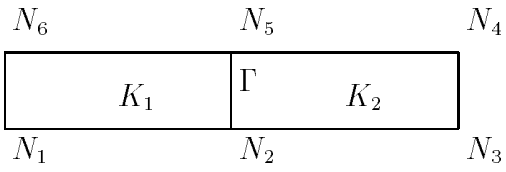

Figure 5.4: Numbering for the proof of the lemma

We thus have with $p_i = p_h(N_i)$:

$$[\bar{p}_h]_\Gamma^2 = \left(\frac{p_1 + p_2 + p_5 + p_6}{4} - \frac{p_2 + p_3 + p_4 + p_5}{4} \right)^2 = \frac{1}{2} \left(\frac{p_1 + p_6}{2} - \frac{p_3 + p_4}{2} \right)^2$$
$$\sim \frac{\bar{h_\Gamma}^2}{|K|} \|\partial_\Gamma^- p_h\|_{K_1 \cup K_2}^2,$$

leading to

$$h_\Gamma^- \int_\Gamma [\bar{p}_h]^2 \, ds = |K| [\bar{p}_h]_\Gamma^2 \leq \bar{h_\Gamma}^2 \|\partial_\Gamma^- p_h\|_{K_1 \cup K_2}^2.$$

Since

$$p_h(x, y) - p_h(s, t) = \int_t^y \partial_y p_h(x, \eta) \, d\eta + \int_s^x \partial_x p_h(\xi, y) \, d\xi,$$

integrating with respect to s and t gives:

$$|p_h(x,y) - \bar{p}_h| \le \frac{1}{h_x h_y} \int_0^{h_x} \int_0^{h_y} \left[\int_0^{h_y} |\partial_y p_h| \, d\eta + \int_0^{h_x} |\partial_x p_h| \, d\eta \right] ds \, dt$$
$$\le \int_0^{h_y} |\partial_y p_h(x,\eta)| \, d\eta + \int_0^{h_x} |\partial_y p_h(\xi,y)| \, d\xi.$$

Applying the Cauchy inequality and integrating again finally yields:

$$\|p_h - \bar{p}_h\|_K^2 \le h_x^2 \|\partial_x p_h\|_K^2 + h_y^2 \|\partial_y p_h\|_K^2$$

□

Proof of the theorem. Let p_h be a given bilinear pressure. We choose $\bar{p}_h$ to be the L_2-projection of p_h onto the space of piecewise constant functions. We use the stability estimate of the last section, denoting by v_h the velocity that attains the supremum for $\bar{p}_h$ in (5.6),

$$\frac{(p_h, \nabla \cdot v_h)}{\|\nabla v_h\|} = \frac{(\bar{p}_h, \nabla \cdot v_h)}{\|\nabla v_h\|} + \frac{(p_h - \bar{p}_h, \nabla \cdot v_h)}{\|\nabla v_h\|},$$

by (5.6) and (5.14)

$$\ge \gamma \|\bar{p}_h\| - c_0 (\bar{p}_h, \bar{p}_h)^{1/2} - c_1 (p_h, p_h)^{1/2}$$

and by 5.12 and 5.13

$$\ge \gamma \|p_h\| - c_1 (p_h, p_h)^{1/2}.$$

(5.11) follows immediately. □

5.4 Error estimates

The stability results of the preceding sections are applied to the derivation of error estimates on stretched orthogonal grids. We additionally need some results for the interpolation error of I_h defined by (5.3). Using the invariance of I_h on the discrete space V_h and its stability, we can derive interpolation estimates which take the anisotropy of the mesh into account. Similar results have been obtained by Apel and Dobrowolski (see [1]). The next lemma collects the interpolation estimates.

Lemma 5.2. *Under the above assumptions on the triangulation $\mathcal{T}$, we have:*

$$\|\partial_x(u - I_h u)\| \leq C \left(h_x \|\partial_x^2 u\| + h_y \|\partial_x \partial_y u\| + \frac{h_y^2}{h_x} \|\partial_y^2 u\| \right) \tag{5.15}$$

$$\|\partial_y(u - I_h u)\| \leq C \left(h_x \|\partial_x \partial_y u\| + h_y \|\partial_y^2 u\| \right) \tag{5.16}$$

$$\|p - J_h p\| \leq C \left(h_x \|\partial_x p\| + h_y \|\partial_y p\| \right) \tag{5.17}$$

Remark 5.1. *For the nodal interpolation operator, (5.15) holds without the last term. The presence of this term for the anisotropic interpolation operator, however, does not harm the error estimate since it is of smaller size than the other terms if the anisotropic refinement is along the direction of high y-derivatives of u.*

Proof. Let us first proof that (5.15) and (5.16) are correct for the nodal interpolation operator N_h. For abbreviation we set $u_h = N_h u$ and $f = \partial_x(u - u_h)$. On the reference element $\hat{K}$ we have:

$$\int_{-1}^{1} \partial_x u(\xi, \pm 1)\, d\xi = u(1, \pm 1) - u(-1, \pm 1) = u_h(1, \pm 1) - u_h(-1, \pm 1),$$

and therefore

$$\int_{-1}^{1} f(\xi, \pm 1)\, d\xi = 0.$$

Furthermore, we have

$$\begin{aligned} f(x, y) - f(t, -1) &= (f(x, y) - f(t, y)) + (f(t, y) - f(t, -1)) \\ &= \int_t^x \partial_x f(\xi, y)\, d\xi + \int_{-1}^{y} \partial_y f(t, \eta)\, d\eta, \end{aligned}$$

leading to

$$f(x, y) = \int_{-1}^{1} \int_t^x \partial_x f(\xi, y)\, d\xi\, dt + \int_{-1}^{1} \int_{-1}^{y} \partial_y f(t, \eta)\, d\eta\, dt,$$

and so

$$\|\partial_x(u - u_h)\|_{\hat{K}} \leq C \left(\|\partial_x^2 u\|_{\hat{K}} + \|\partial_x \partial_y (u - u_h)\|_{\hat{K}} \right).$$

But

$$\|\partial_x\partial_y u_h\|_{\hat{K}} \leq C\|\partial_x\partial_y u\|_{\hat{K}}$$

since

$$\partial_x\partial_y u_h = \int_{\hat{K}} \partial_x\partial_y u \, dxdy.$$

Transformation to the reference element now yields:

$$\begin{aligned}
\|\partial_x(u-u_h)\|_K &\leq C\frac{|K|}{h_x}\,\|\hat{\partial}_x(\hat{u}-\hat{u}_h)\|_{\hat{K}} \\
&\leq C\frac{|K|}{h_x}\,\left(\|\hat{\partial}_x^2\hat{u}\|_{\hat{K}} + \|\hat{\partial}_x\hat{\partial}_y\hat{u}\|_{\hat{K}}\right) \\
&\leq C\left(h_x\|\partial_x^2 u\| + h_y\|\partial_x\partial_y u\|\right).
\end{aligned}$$

In order to proof the results for I_h, we note that

$$u - I_h u = (u - N_h u) + (N_h u - I_h u) = (u - N_h u) + I_h(u - u_h).$$

From the prof of the stability estimate (5.4) we further see that

$$\|\partial_y I_h W\| \leq C\|\partial_y w\| \quad \text{and} \quad \|\partial_x I_h W\| \leq C\left(\|\partial_x w\| + \frac{h_y}{h_x}\|\partial_y w\|\right).$$

The estimates (5.15) and (5.16) follow immediately. □

Theorem 5.4. *Let the solution $\{u,p\}$ of the Stokes equations be in $H^2(\Omega)\times H^1(\Omega)$. Then, for $\{u_h,p_h\}$ defined by (5.5) or (5.10), there holds:*

$$\begin{aligned}
&\|\nabla(u-u_h)\| + \|p-p_h\| \qquad (5.18)\\
&\qquad \leq C\sum_{K\in\mathcal{T}}\Big\{h_x^K\|u_{xx}\|_{\mathcal{N}(K)} + \max(h_x^K,h_y^K)\|u_{xy}\|_{\mathcal{N}(K)} \\
&\qquad + h_y^K\|u_{yy}\|_{\mathcal{N}(K)} + h_x^K\|p_x\|_{\mathcal{N}(K)} + h_y^K\|p_y\|_{\mathcal{N}(K)}\Big\}, \qquad (5.19)
\end{aligned}$$

with a constant C not depending on the aspect ratio. The local mesh parameters are defined by $h_y^K = \max_{L\in\mathcal{N}(K)} h_y^L$ and similar for h_x^K. Assuming the mesh size to vary smoothly, h_y^K and h_x^K can be expressed in terms of local quantities on each K.

Proof. We have, using the Galerkin character of our discretization,

$$\|\nabla(u-u_h)\| \leq C\left(\|\nabla(u-I_h u)\| + \|p-J_h p\|\right)$$

and

$$\gamma\|p-p_h\| \leq C\left(\|\nabla(u-I_h u)\| + \|p-J_h p\|\right).$$

Since the constant γ does not depend on the mesh parameters, (5.18) follows by applying the interpolation results. □

5.5 Generalizations to non-orthogonal and three dimensional meshes

First we consider the case of non-orthogonal meshes in two dimensions. In order to apply the above theory we construct a transformation T which locally maps the given triangulation $\mathcal{T}$ to an orthogonal one (c.f. figure 5.5). We use the transformation T_K of a given element K to the reference element to obtain a piecewise linear function. Noting that the eigenvalues of this transformation describe the stretching of the element, we may decompose $T_K = D_K T_K^0$, where D_K corresponds to a diagonal matrix with the eigenvalues of T_K as entries. We construct the orthogonal stretched mesh $\tilde{\mathcal{T}}$ and the transformation T by glueing together the T_K^0. We suppose that $\tilde{\mathcal{T}}$ satisfies (5.1) and additionally require the following bounds for T and for its inverse S:

$$\|DT\|_{L_\infty} + \|DS\|_{L_\infty} \leq \alpha, \tag{5.20}$$

where α is a moderate constant. Denoting by w the transformation of a H^1-function v and by $w_h = I_h^0$ its anisotropic interpolate, we define

$$I_h v \equiv w_h(T) \tag{5.21}$$

Let us consider the question of stability of I_h. Using (5.20), we have:

$$\|\nabla I_h v\| = \|DT^*\nabla w_h\| \leq \alpha\|\nabla w_h\| \leq \alpha\|\nabla w\| = \alpha\|DS^*\nabla v\| \leq \alpha^2\|\nabla v\|.$$

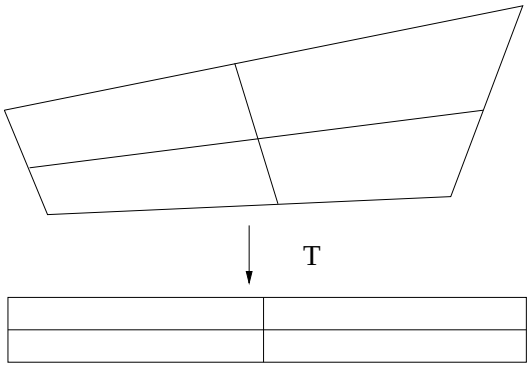

Figure 5.5: Transformation to an orthogonal grid

Finally we consider the three dimensional case. Since the elements may be stretched in one or two directions (see figure 5.6), we have to modify the assumption (5.1) in order to discard abrupt changes of the aspect ratio in any of the directions.

The definition of the interpolation operator (5.3) has to be changed in the

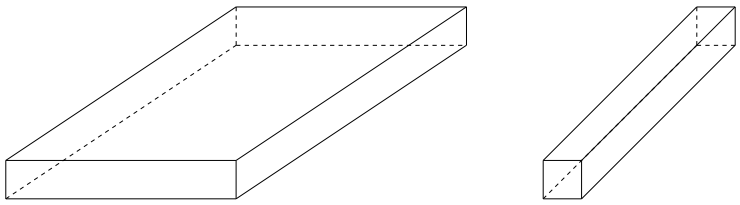

Figure 5.6: Stretched elements in 3D

following way: Depending on whether one or two directions are stretched, we have to associate either an edge or a face integral to a given node. Similarly, in the stability proof for the Q^1/Q^0-element the auxiliary pressure space has to be constant on the four elements neighboring a macro face in the second case. With these modifications our proofs carry immediately over to the three dimensional case.

5.6 Numerical Tests

The matrix system corresponding to the stabilized discretization of the Stokes equations (5.10) reads in matrix notation

$$\begin{bmatrix} A & B \\ -B^T & \alpha C_i \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} ,$$

where C_i corresponds to the stabilizing bilinear form $c_{\rm I}$, $c_{\rm II}$ or $c_{\rm III}$. The Schur complement of the main diagonal block A is $S = \alpha C_i - B^T A^{-1} B$. Then, the stability constant γ in (5.11) is given by

$$\gamma = \lambda_{min}(M^{-1}S)^2 ,$$

where M denotes the mass matrix corresponding the the pressure space. Relation (5.11) together with an upper bound for $\lambda_{max}(M^{-1}S)$ implies that S is spectrally equivalent to M. The question is whether this remains valid if the aspect ratio tends to infinity. In view of the relation (5.11), the continuity constant, which bounds $\lambda_{max}(B^T A^{-1} B)$, can be estimated independently of σ. Hence, we may estimate γ by counting the number of CG–iterations (preconditioned by the mass matrix) needed to invert the Schur complement. For these test calculations we have chosen a sequence of uniformly anisotropic grids obtained by one-directional refinements. The results are given in table 5.1. The increase of the stability constant for $c_{\rm I}$ reflects the fact that

	No. of Iterations						
$C_i \vert \sigma$	2	4	8	16	32	64	128
1	8	18	39	98	559	*	*
2	8	18	39	88	193	*	*
3	8	16	29	31	29	27	24

Table 5.1: Number of cg–iterations

$\lambda_{min}(M^{-1}S) \sim \sigma^{-1}$, whereas the bad behavior of $c_{\rm II}$ can be explained by $\lambda_{max}(M^{-1}S) \sim \sigma$ (over–stabilization), as we only have $c_{\rm II}(p,q) \leq \sigma \|p\| \|q\|$. We also see that the anisotropic stabilization using $c_{\rm III}$ leads to an aspect

ratio independent behavior.
For illustrating the relevance of the above error estimate (5.18) , we consider the accuracy of the different stabilizations. For this we construct a sequence of grids with a fixed number of nodes and increasing aspect ratio. This is done by pulling nodes exponentially to the boundary. The results of this test are shown in table 5.2, where L denotes the number of refinement steps used in the generation of the computational mesh.

		$\|\|p - p_h\|\|$			
c_i	L	$\sigma = 100$	$\sigma = 500$	$\sigma = 1000$	
	3	$2.10 - 2$	$2.84 - 2$	$6.17 - 2$	
1	4	$1.37 - 2$	$7.84 - 3$	$5.77 - 2$	$O(\Gamma)$
	5	$1.12 - 2$	$4.69 - 3$	$5.82 - 2$	
	3	$3.57 - 2$	$4.33 - 2$	$1.80 - 1$	
2	4	$1.27 - 2$	$1.63 - 2$	$1.70 - 2$	$O(1)$
	5	$5.13 - 3$	$3.60 - 1$	$2.15 - 1$	
	3	$4.95 - 2$	$5.44 - 2$	$5.51 - 2$	
3	4	$1.45 - 2$	$1.63 - 2$	$1.70 - 2$	$O(h^2)$
	5	$3.90 - 3$	$4.42 - 3$	$4.76 - 3$	

Table 5.2: Accuracy of the various versions of the stabilized Q_1/Q_1–Stokes element

5.7 Solution Algorithm

Finally, we briefly describe our concept of the multigrid algorithm used for solving the discrete Stokes problems (5.10) and also the related discrete Navier-Stokes problems. Within a stationary or nonstationary Navier-Stokes calculation a simple fixed–point iteration is employed for treating the non–linearity, as described for instance in [52]. The resulting linearized systems are then solved in each step by a multigrid algorithm on a sequence of hierarchical meshes. In our test calculations these meshes have been generated by systematic subdivision (bisection of element edges) of a macro-decomposition

of the domain Ω.
In this multigrid algorithm the generic restrictions and prolongations are used. The smoother is an adaptation of the block–Gauss–Seidel iteration introduced by Vanka [54]. Within an iteration sweep, to each nodal point the patch consisting of the four surrounding elements is associated and the corresponding velocity and pressure unknowns are updated simultaneously. A detailed presentation with numerical results is given in chapter 6.
From the classical multigrid–theory for elliptic equations we know that point–iterations lose their smoothing property for large aspect ratios. The remedy is the use of a smoother which becomes a direct solver in the limit $\sigma \to \infty$. Consequently, since our smoother is similar to a point–Gauss–Seidel iteration, we expect problems in the case of strongly stretched grids. Our strategy to overcome this difficulty is as follows:
Since we expect the mesh aspect ratio to be large only in a small part of the computational domain, we should use an *adaptive* smoother. This means that we will combine the point–smoother with a more robust version just where we need it, for instance on elements with a large σ_K. In this approach the nodes are grouped in the direction of the anisotropic mesh refinement and iterated implicitly leading to a process which may be termed "string-wise"–block–Gauss–Seidel method. Our algorithm works as follows:
– Define a set of lines (strings) in the direction of the anisotropic mesh refinement which can be determined from the matrix entries.
– Along these strings, use a direct band–solver for updating the corresponding local velocity and pressure values. This process is controlled by a switch aspect ratio σ_s. Figure 5.7 illustrates the process by showing those areas of the computational domain where the point–smoother is adaptively modified into a "string"–smoother for $\sigma_s = 2, 3, 10$. For testing the effectiveness of the above stabilization strategy we have calculated a jet flow in a channel. This more realistic example shows that the discussed stability properties are also crucial in Navier–Stokes calculations. The coarse grid and the boundary conditions are described by figure 5.8. Table 5.3 contains the multigrid convergence rates in dependence of the viscosity ν and the mesh aspect ratio σ, using 2 adaptive ($\sigma_s = 5$) pre– and post-smoothing sweeps. For low Reynolds numbers the algorithm behaves well independently of the aspect ratio. For

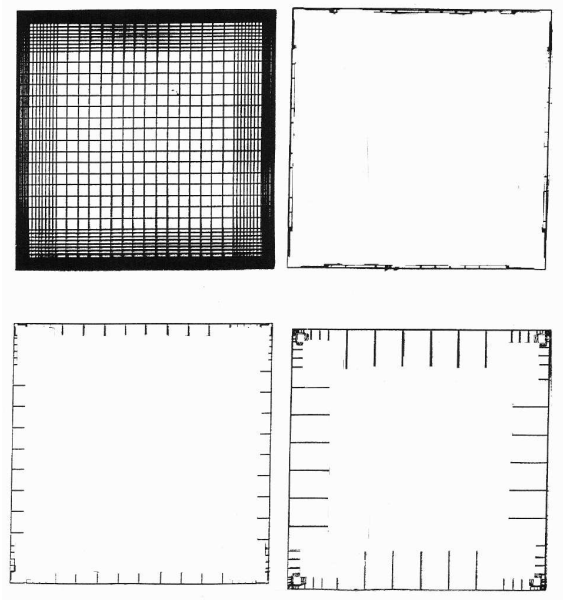

Figure 5.7: Adaptively chosen "strings" for a "driven cavity" grid

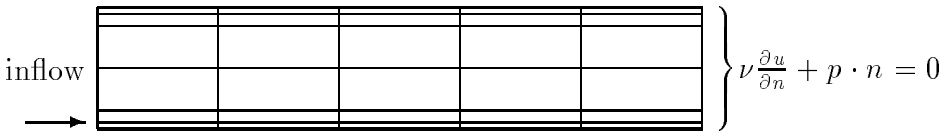

Figure 5.8: Coarse grid and boundary conditions for the jet flow

higher Reynolds numbers the convergence speed deteriorates. This indicates that the discretization of the convective terms needs some additional care on anisotropic meshes. To conclude this discussion we present some numerical

	ρ_{mg}			
$\nu \mid \sigma$	50	100	500	1000
1	0.30	0.31	0.33	0.33
10^{-2}	0.30	0.33	0.40	0.45
10^{-3}	0.30	0.40	0.42	0.54

Table 5.3: Multigrid convergence rates for the jet problem depending on ν and σ

results for the case with $\nu = 1/3000$, where the upper picture shows the effect of a wrong stabilization.

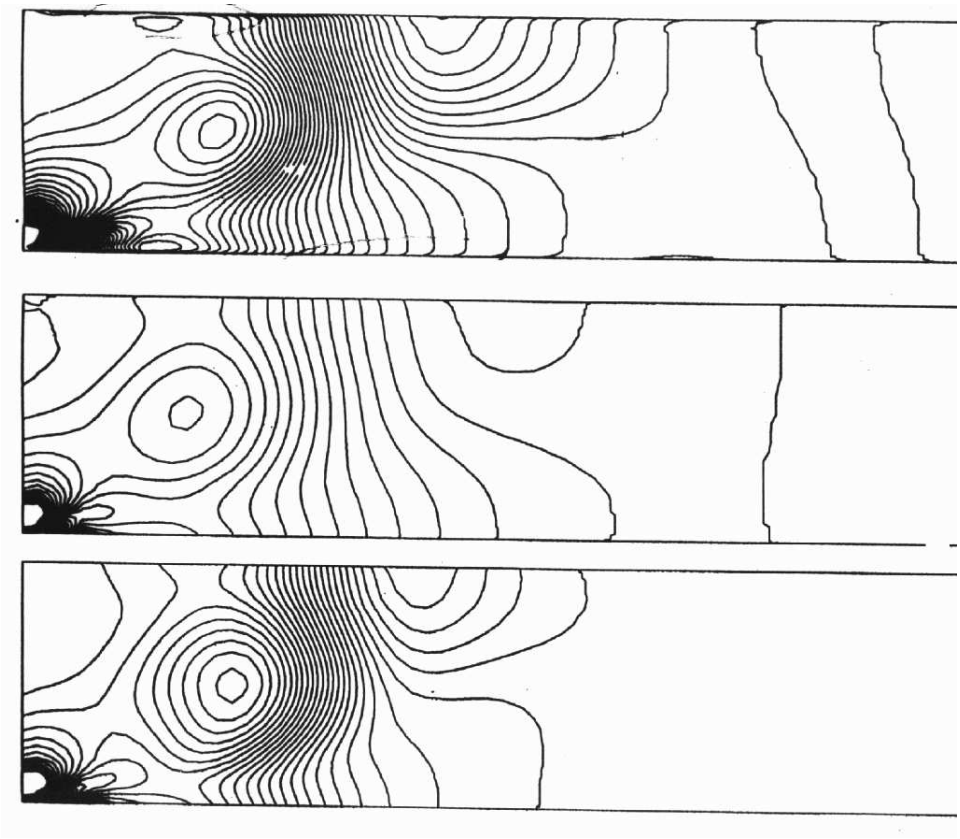

Figure 5.9: Pressure isolines for the jet problem calculated with isotropic (upper), "Mini"– (middle) and anisotropic (lower) stabilization

Chapter 6

Finite elements for the Navier–Stokes equations

The generalization of the techniques described in the preceding chapters to the non-linear incompressible Navier-Stokes equations poses some additional problems. The discretization has to be stabilized not only in the sense of the velocity-pressure coupling, but also with respect to dominant convection in the case of high Reynolds number. This is achieved by adding further weighted least squares terms to the discrete equations as in the streamline diffusion method for the convection-diffusion equation. In this way we get a natural finite element discretization, which introduces the correct amount of artificial viscosity without sacrificing accuracy since numerical viscosity is added only in stream-wise direction. The systematic approach insures that the method is consistent in the sense that a strong solution of the Navier-Stokes equations also satisfies the discrete equations.

We further describe a generalization of the residual based error estimator for the Stokes equations to the non-linear equations. The estimator is used for the automatic mesh refinement. We demonstrate the advantage of the adaptive algorithm in the context of some standard but non-trivial test problems for incompressible flow solvers.

For the nonstationary equations we discuss the application of a pressure correction scheme. The advantage of this splitting method is a considerable reduction of computational complexity in the case of transient flows.

6.1 The stationary case

In this section we discuss the discretization of the stationary incompressible Navier–Stokes equations with the stabilized Q^1/Q^1–element. The new difficulty occurring here is the possible dominance of the convection term. Further we describe a multigrid algorithm for the coupled equations. The smoothing operator is a patch-wise block Gauss–Seidel iteration which will be described below. Finally we present some results for 2-D test computations.

The stationary incompressible Navier-Stokes equations are the following system of equations, involving the velocities u and the pressure p of an incompressible fluid at stationary state:

$$\begin{aligned} u \cdot \nabla u - \nu \Delta u + \nabla p &= f \\ \nabla \cdot u &= 0, \end{aligned} \tag{6.1}$$

posed on a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ and supplemented with the following boundary conditions:

$$\begin{aligned} &\text{(rigid wall)} \quad u = 0 \quad \text{on } \Gamma_0 \\ &\text{(inflow)} \qquad u = u_0 \quad \text{on } \Gamma_1 \\ &\text{(outflow)} \quad \nu \frac{\partial u}{\partial n} + p \cdot n = 0 \quad \text{on } \Gamma_2, \end{aligned}$$

where $\Gamma_0 \cup \Gamma_1 \cup \Gamma_2 = \partial\Omega$ and n denotes the outer unit normal. The boundary condition on Γ_0 and on Γ_1 is a Dirichlet condition, whereas on Γ_2 a Neumann-type condition is prescribed. Under well-known (and in most cases unrealistic) conditions on the domain Ω, the volume forces f, the normalized viscosity ν and the boundary data, the equations admit a unique solution (see [25]).

6.1.1 Discretization by the Galerkin-least-squares method

The incompressible Navier Stokes equations have two different sources of difficulties that have to be overcome by a stable discretization. First, for increasing Reynolds numbers (small ν), the flow becomes convection dominated and the operator acting on the velocities changes behavior. A standard

finite element technique to deal with scalar convection-diffusion equations is the streamline diffusion method [34]. It is known to give an accurate and stable discretization in the whole range of the diffusion parameter. A precise error analysis can be found in [59].
The second difficulty is the velocity-pressure coupling. A stabilizing technique for this issue has been discussed in the previous chapter.
Since both techniques rely on the modification of the standard Galerkin formulation, it is quite natural to combine them to treat the two difficulties simultaneously through a so-called Galerkin-least-squares approach. This is explained in the following.
We start with the weak formulation of (6.1). For simplicity we consider the case of homogeneous Dirichlet boundary conditions:

$$\begin{aligned} &\text{Find } u \in V \text{ and } p \in Q \text{ such that:} \\ &(u \cdot \nabla u, v) + \nu(\nabla u, \nabla v) - (p, \nabla \cdot v) = (f, v) \quad \forall v \in V, \\ &(\nabla \cdot u, q) = 0 \qquad \forall q \in Q, \end{aligned} \tag{6.2}$$

where $V = H_0^1(\Omega)^2$ and $Q = L_2(\Omega)/R$ in the case of Dirichlet boundary conditions.
We obtain the standard Galerkin discretization by choosing finite element subspaces $V_h \subset V$ and $Q_h \subset Q$. As mentioned before this does not yield a stable algorithm unless the spaces fulfill the discrete LBB-condition. Therefore the approach is modified.
The streamline diffusion method for the convection-diffusion equation introduces a stabilizing term by the use of an additional test function of the form $\delta u \cdot \nabla v$. In a similar way, the pressure stabilization adds the term $\alpha \nabla q$. Furthermore we introduce the term $\tau \nabla \cdot v$ as a test function for the divergence equation, as proposed in [27]. The introduction of the additional least squares terms is done in an element-wise fashion. This implies that the weighting parameters δ, α and τ depend on the element and will be subscribed correspondingly. Putting all the terms together, we obtain the following set of equations:

$$
\begin{aligned}
(u\cdot\nabla u, v) &+ \nu(\nabla u, \nabla v) - (p, \nabla\cdot v) \\
&+ \sum_K (u\cdot\nabla u - \nu\Delta u + \nabla p,\, \delta_K u\cdot\nabla v + \alpha_K \nabla q)_K \\
&= (f, v) + \sum_K (f, \delta_K u\cdot\nabla v + \alpha_K\nabla q)_K \qquad \forall v \in V, \\
(\nabla\cdot u, q) \quad &+ \sum_K (\nabla\cdot u,\, \tau_K \nabla\cdot v)_K = 0 \qquad \forall q \in Q.
\end{aligned}
$$

This discretization has been analyzed for different choices of the parameters α, δ and τ in [27], [24] and [51]. The method is consistent in the sense that the stabilizing terms vanish for a strong solution of (6.1). We only sketch the error analysis in order to clarify the role of the parameters and motivate our choice. To this end it is sufficient to consider the linearized equations:

$$
\begin{aligned}
\beta\cdot\nabla u - \nu\Delta u + \nabla p &= f \\
\nabla\cdot u &= 0,
\end{aligned}
$$

where β is a given divergence free velocity field, for example the one from a previous step of a fixed point iteration.
We introduce the following bilinear forms:

$$
\begin{aligned}
a(\{u,p\},\{v,q\}) &= \nu(\nabla u, \nabla v) + (\beta\cdot\nabla u, v) - (p, \nabla\cdot v) + (\nabla\cdot u, q), \\
s(\{u,p\},\{v,q\}) &= \sum_K (-\nu\Delta u + \cdot\nabla u + \nabla p,\, \delta_K \beta\cdot\nabla v\beta + \alpha_K\nabla q)_K \\
&\quad + \sum_K (\nabla\cdot u,\, \tau_K\nabla\cdot v)_K, \\
\tilde{a} &= a + s.
\end{aligned}
$$

Further, we only consider the case of finite element spaces defined by piecewise (bi-)linear functions for both variables. The continuous and discrete solutions are characterized by:

$$
\begin{aligned}
a(\{u,p\},\{v,q\}) &= (f, v) \qquad\qquad \forall\{v,q\} \in V\times Q \\
\tilde{a}(\{u_h,p_h\},\{v_h,q_h\}) &= (f, v_h) \\
&\quad + \sum_K (f,\, \beta_K\cdot\nabla u_h + \alpha_K\nabla q_h) \quad \forall\{v_h,q_h\} \in V_h\times Q_h.
\end{aligned}
$$

Denoting the error by $\{e_u, e_p\} = \{u - u_h, p - p_h\}$, we obtain by consistency:

$$
\tilde{a}(\{e_u, e_p\},\{v_h,q_h\}) = 0 \quad \forall\{v_h,q_h\} \in V_h\times Q_h. \tag{6.3}
$$

First we note that the required symmetry of the coupling of the terms $\beta \cdot \nabla u$ and ∇p directly imposes that $\alpha_K = \delta_K$. Next we introduce a δ_K-dependent metric,

$$|||\{u_h, p_h\}|||^2 \equiv \nu \|\nabla u_h\|^2 + \sum_K \delta_K \|\beta \cdot \nabla u_h + \nabla q_h\|_K^2.$$

This choice immediately implies the following stability estimate:

$$\tilde{a}(\{v, q\}, \{v, \}) \geq |||\{v, q\}|||^2.$$

Splitting the error as $e_u = (u - I_h u) + (I_h u - u_h) = \xi_u + \eta_u$ and similarly for the pressure error, we get:

$$|||\{\eta_u, \eta_p\}|||^2 \leq \tilde{a}(\{\eta_u, \eta_p\}, \{\eta_u, \eta_p\}) = -\tilde{a}(\{\eta_u, \eta_p\}, \{\xi_u, \xi_p\}),$$

due to (6.3). Observing that the symmetric terms of $\tilde{a}$ can be estimated by $|||\cdot|||$, we get:

$$\begin{aligned} |||\{\eta_u, \eta_p\}|||^2 \leq \frac{1}{4}|||\{\eta_u, \eta_p\}|||^2 + |||\{\xi_u, \xi_p\}|||^2 \\ + |(\beta \cdot \nabla \xi_u, \eta_u)| + |(\xi_p, \nabla \cdot \eta_u)| + |(\nabla \cdot \xi_u, \eta_p)|. \end{aligned}$$

The last terms can be estimated element-wise by

$$\begin{aligned} \delta_K \|\beta \cdot \nabla \eta_u\|_K^2 + \frac{1}{\delta}\|\xi_u\|_K^2 + \tau_K \|\nabla \cdot \eta_u\|_K^2 \\ + \frac{1}{\tau_K}\|\xi_p\|_K^2 + \delta_K \|\nabla \eta_p\|_K^2 + \frac{1}{\delta_K}\|\xi_u\|_K^2. \end{aligned}$$

Absorbing the η-terms yields:

$$\begin{aligned} |||\{\eta_u, \eta_p\}|||_K^2 \leq (\nu + \delta_K \|\beta\|_{K,\infty} + \tau_K)\, h_K^2 \|\nabla^2 u\|_K^2 + \delta_K h_K^2 \|\nabla^2 p\|_K^2 \\ + \frac{h^4}{\delta_K}\|\nabla^2 u\|_K^2 + \frac{h^4}{\tau_K}\|\nabla^2 p\|_K^2. \end{aligned}$$

It is obvious that the term $(\xi_p, \nabla \cdot \eta_u)$ can alternatively be estimated using $\nu \|\nabla u\|$. This shows that we may put $\tau = 0$ for large ν. Although the above analysis is not precise, it can be used for the choice of the parameters, leading to

$$\delta_K \sim \tau_K \sim h_K \max\left(\frac{h_K}{\|\beta\|_{K,\infty}} - \nu, 0\right).$$

This parameter design has been proposed in [24].

6.1.2 Multigrid for the stationary equations

First we describe the system of equations arising from the Galerkin-least-squares discretization. For this purpose we denote by φ_i, $i = 1, N$ the set of basis functions and expand the pressure and velocity functions in the usual way:

$$p = \sum \hat{p}_i \, \varphi_i \quad u = \sum \hat{u}_i \, \varphi_i,$$

where $(\hat{u}_i)_{1 \le i \le N}$ denotes the vector of the components of the velocity degrees of freedom and similar for the pressure. We obtain the following system of equations:

$$\begin{bmatrix} A + N + S & -B^* + T^* \\ B + T & C \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}.$$

Denoting the test functions by ψ, the different matrices are given by:

$$\begin{aligned} A &= \nu(\nabla\psi, \nabla\varphi) & N &= (\psi, \hat{u} \cdot \nabla\varphi) \\ B &= (\psi, \nabla \cdot \varphi) & S &= \sum_K \delta_K(\hat{u} \cdot \nabla\psi, \hat{u} \cdot \nabla\varphi)_K + \tau_K(\nabla \cdot \psi, \nabla \cdot \varphi)_K \\ T &= \sum_K \delta_K(\nabla\psi, u \cdot \nabla\varphi)_K & C &= \sum_K \delta_K(\nabla\psi, \nabla\varphi)_K. \end{aligned}$$

This nonlinear system is solved by a quasi-Newton iteration, which produces a system of linear equations to be solved in each step. Next we describe a multigrid algorithm to invert these coupled linear equations.
We use the canonic finite element multigrid algorithm with L^2-projection as restriction and the inclusion as prolongation operator. The smoothing operator, however, requires more attention since the system is neither symmetric nor positive definite. As smoothing operator we employ a block Gauss-Seidel iteration similar to the one proposed in [54] for the staggered grid finite difference discretization. A smoothing step consists of a loop over the pressure degrees of freedom, where we simultaneously update the corresponding pressure value together with the velocity unknowns coupling with it. The systematic of this coupling is shown in figure 6.1. Having calculated the residuals r_i and r for the velocity and the pressure respectively, we obtain

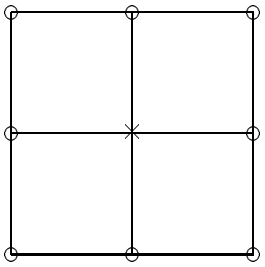

Figure 6.1: Patch for the Vanka-smoother

the following local system for the velocity- and pressure-updates, z_i and z:

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 & -b_1 \\ 0 & a_{22} & \cdots & 0 & -b_1 \\ 0 & 0 & \cdots & a_{nn} & -b_n \\ b_1 & b_2 & \cdots & b_n & c \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \\ z \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_n \\ r \end{bmatrix}.$$

We have simplified this system by neglecting the coupling terms between the velocities (a_{12} etc.). Due to its simple structure the obtained matrix can be inverted within two sweeps over the involved unknowns.
The idea of this construction is to get velocity updates which satisfy the divergence equation with respect to the test function defining the patch. In the same way the pressure update is the Lagrange multiplier for this system. Note that the solver described above does not make any use of the special structure caused by the stabilization. The behavior of the smoother is demonstrated by a standard driven cavity computation with different Reynolds numbers in table 6.1. We also list the number of fixed point iterations, denoted by FPI. The number of elements is N.

Re—N	1600	6400	25600	FPI
1	0.081	0.096	0.121	4
100	0.098	0.099	0.130	6
1000	0.227	0.245	0.168	9
5000	0.285	0.368	0.370	18

Table 6.1: Multigrid convergence rates and number of nonlinear iterations for the driven cavity

6.1.3 Examples for the adaptive algorithm

In this section we present some numerical results for the adaptive algorithm. The error estimator we use is the natural generalization of the one described for the Stokes equations. For the L_2-norm it takes the form:

$$\eta(u_h, p_h) \equiv C_S C_I \left[\left\{ \sum_K h_K^4 \| f - u_h \cdot \nabla u_h + \Delta_h u_h - \nabla_h p_h \|_K^2 \right\} \right.$$
$$+ \left\{ \sum_K h_K^3 \sum_{\Gamma \subset K} \| [p_h \cdot n - \frac{\partial u_h}{\partial n}] \|_{L_2(\Gamma)}^2 \right\} + \left\{ \sum_K h_K^4 \| \nabla \cdot u_h \|_K^2 \right\}$$
$$\left. + \left\{ \sum_K \delta_K^2 \| u_h \cdot \nabla u_h + \nabla p_h - f \|_K^2 \right\} \right]^{1/2} .$$

As mentioned above we set $C_S = 1$, leaving the question of error control aside. In figure 6.2 we show the flow structure and an adaptive grid with ~ 5000 elements for the standard lid driven cavity test problem. The Reynolds number is equal to 1000. Due to the pressure singularities there is a strong refinement in the upper corners. In contrast to the lower Reynolds number case the algorithm tends to refine considerably more the upper right corner. Note that the reattachement points of the secondary vortices are also refined. This indicates that the estimator (6.4) reflects the important flow features even if it is supposed to control the global L_2-error, which is probably not the best measure to get a good resolution of small vortices. The next example is the flow around a cylinder (a test problem taken from the DFG benchmark [8]). This allows us to compare the performance of our adaptive algorithm with other highly developed computer codes for the incompressible Navier–Stokes equations. The quantities of interest are the pressure difference on the cylinder and the C_W- and C_A-values. Only the pressure difference is considered here because the corresponding approximate values from the benchmark seem to be more reliable than the other quantities. We also give the values of the calculated pressure at specific points.

The geometric configuration is the following: The channel has a height of 0.41 m and a length of 2.2 m. The diameter of the cylinder which is fixed at 0.15 m from the inflow at a height of 0.15 m is 0.1 m. The Reynolds number for the stationary problem we consider here is 20. The first computations are done on the almost uniform grid Grid1 (c.f. 6.3). The pressure values

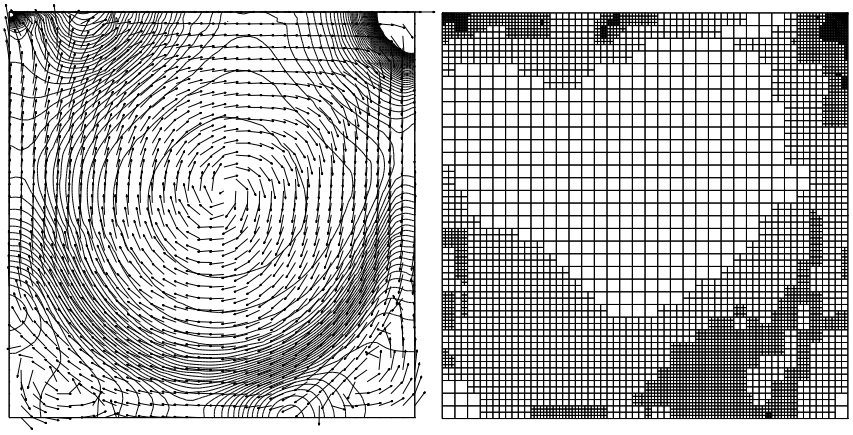

Figure 6.2: Vector plot and pressure isolines (left) and locally refined mesh (right) for the driven cavity computation

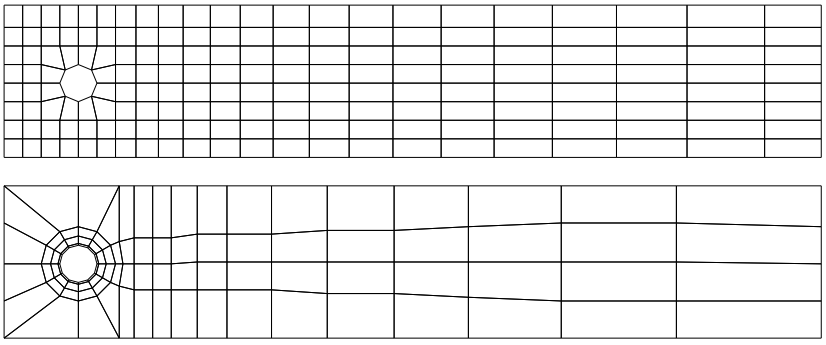

Figure 6.3: Coarse grids 1 and 2 for the benchmark problem

at the points $N_1 = (0.05, 0.15)$, $N_2 = (0.25, 0.15)$, $N_3 = (0.15, 0.05)$ and $N_4 = (0.15, 0.025)$ are given in table 6.2. Table 6.3 shows the approximate pointwise errors of the pressure at N_1 and the approximate pointwise error of the first velocity component at N_3. We also give approximate values for the

L_2-error, which are obtained by measuring the differences of two consecutive solution on the common coarse grid. We next show the results obtained

i	$\#n$	$p(N_1)$	$p(N_2)$	$p(N_3)$	$p(N_4)$
1	2268	$1.1246e-01$	$3.0776e-03$	$2.5843e-02$	$2.6580e-02$
2	8664	$1.2087e-01$	$1.0363e-02$	$2.6497e-02$	$2.7264e-02$
3	33840	$1.2678e-01$	$1.3171e-02$	$2.6551e-02$	$2.7305e-02$
4	133728	$1.2969e-01$	$1.4210e-02$	$2.6267e-02$	$2.7016e-02$
5	531648	$1.3104e-01$	$1.4553e-02$	$2.6135e-02$	$2.6883e-02$

Table 6.2: Pressure values on Grid 1, uniform refinement

i	$\#n$	$\|\|e_p\|\|$	$\|e_p(N_1)\|$	$\|\|e_u\|\|$	$\|e_u(N_3)\|$
1	2268	$5.690e-04$	$1.562e-02$	$1.687e-03$	$2.460e-03$
2	8664	$2.679e-04$	$8.410e-03$	$7.396e-04$	$4.440e-03$
3	33840	$1.056e-04$	$5.910e-03$	$2.557e-04$	$3.208e-03$
4	133728	$3.708e-05$	$2.911e-03$	$8.586e-05$	$1.336e-03$
5	531648	$1.261e-05$	$1.341e-03$	$2.528e-05$	$4.228e-04$

Table 6.3: Approximate Errors on Grid 1, uniform refinement

by computations on the "hand-fitted" grid 2. This mesh has been used by Turek in his benchmark computations, see [8]. The same quantities as above are reported in tables 6.4 and 6.5. From these results it is obvious that

i	$\#n$	$p(N_1)$	$p(N_2)$	$p(N_3)$	$p(N_4)$
1	1296	$1.1657e-01$	$1.0251e-02$	$2.8505e-02$	$2.9251e-02$
2	4896	$1.2640e-01$	$1.3980e-02$	$2.7526e-02$	$2.8361e-02$
3	19008	$1.3036e-01$	$1.4878e-02$	$2.6925e-02$	$2.7689e-02$
4	74880	$1.3144e-01$	$1.4793e-02$	$2.6328e-02$	$2.7087e-02$
5	297216	$1.3184e-01$	$1.4746e-02$	$2.6122e-02$	$2.6871e-02$

Table 6.4: Pressure values on Grid 2, uniform refinement

	Approximate Errors, Grid2				
i	$\#n$	$\|\|e_p\|\|$	$\|e_p(N_1)\|$	$\|\|e_u\|\|$	$\|e_u(N_3)\|$
1	1296	$1.070e-03$	$3.107e-02$	$3.639e-03$	$6.524e-03$
2	4896	$3.023e-04$	$9.838e-03$	$1.319e-03$	$5.519e-03$
3	19008	$9.857e-05$	$3.954e-03$	$3.897e-04$	$1.616e-03$
4	74880	$4.395e-05$	$1.082e-03$	$1.268e-04$	$2.473e-04$
5	297216	$1.616e-05$	$3.998e-04$	$3.960e-05$	$3.560e-05$

Table 6.5: Approximate Errors on Grid 2, uniform refinement

the "hand-tuned" grid 2 saves approximately 50 per cent of the unknowns as compared to the uniform grid 1 at the same accuracy.
Next we present the results of an self-adaptive computation. In figure 6.4 the sequence of locally refined meshes is shown. In table 6.6 we present the solution values at the points $N_1 \cdots N_4$. Table 6.7 compares the pressure difference

i	$\#n$	$p(N_1)$	$p(N_2)$	$p(N_3)$	$p(N_4)$
1	849	$1.0713e-01$	$2.0473e-03$	$2.6144e-02$	$2.6772e-02$
2	1410	$1.1687e-01$	$1.0879e-02$	$2.7364e-02$	$2.8269e-02$
3	2604	$1.2535e-01$	$1.3875e-02$	$2.7955e-02$	$2.8736e-02$
4	5355	$1.2891e-01$	$1.4933e-02$	$2.7887e-02$	$2.8655e-02$
5	10353	$1.3089e-01$	$1.5097e-02$	$2.7466e-02$	$2.8204e-02$
6	21720	$1.3187e-01$	$1.4962e-02$	$2.7092e-02$	$2.7840e-02$
7	40674	$1.3207e-01$	$1.4826e-02$	$2.6754e-02$	$2.7525e-02$
8	86529	$1.3213e-01$	$1.4759e-02$	$2.6469e-02$	$2.7219e-02$
9	164301	$1.3227e-01$	$1.4779e-02$	$2.6313e-02$	$2.7067e-02$
10	330930	$1.3231e-01$	$1.4781e-02$	$2.6212e-02$	$2.6962e-02$

Table 6.6: Pressure values, adaptive refinement

of the different grids resp. refinement strategies. The last column contains the computation time on workstation SUN Sparc 10/51. We conclude from table 6.7, that we can reduce the number of unknowns down to 1-5 per cent by the use of the adaptive refinement. Depending on the prescribed precision

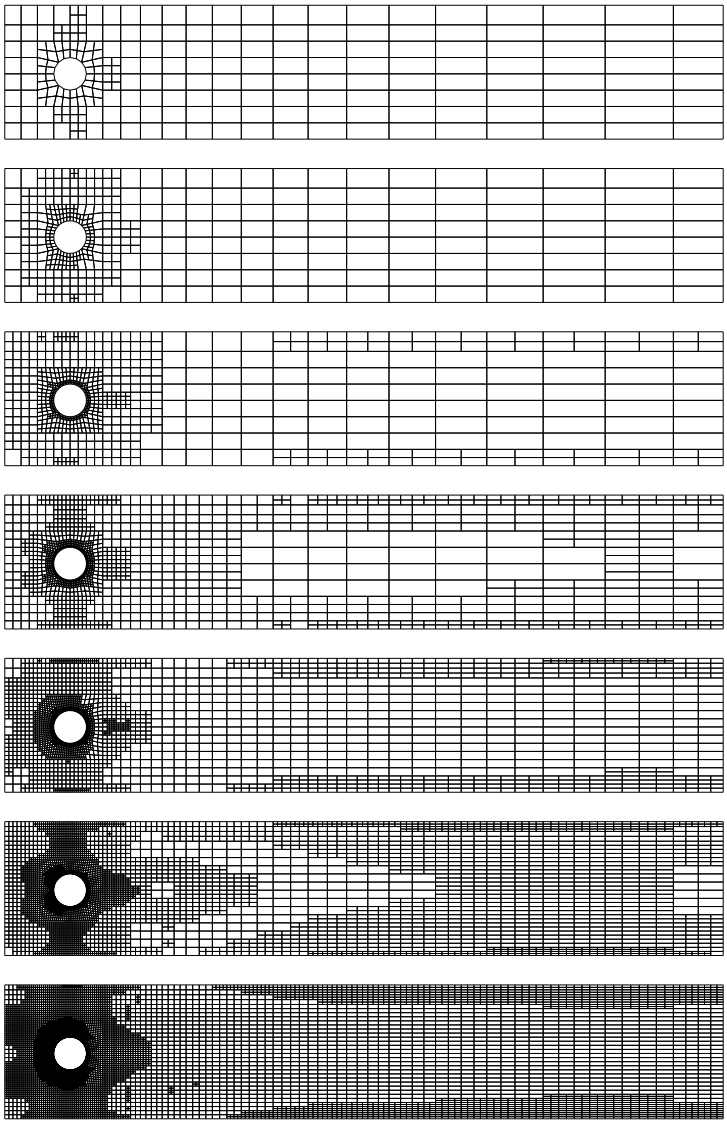

Figure 6.4: Grids during the adaptive refinement

the gain may even be more significant. With about 160.000 unknowns we obtain the same value for the pressure difference than the apparently most precise computations with more than 7.000.000 unknowns done by Schieweck [8].

As could be expected from the construction of the multigrid algorithm on

locally refined meshes this gain directly translates into a saving of computational time.

	Uniform Refinement, Grid1		
i	$\#n$	Δp	$sec.$
1	2268	$1.09389e-01$	4.0
2	8664	$1.10513e-01$	15.2
3	33840	$1.13617e-01$	61.7
4	133728	$1.15488e-01$	254.2
5	531648	$1.16486e-01$	1001.1
	Uniform Refinement, Grid2		
1	1296	$1.06318e-01$	3.3
2	4896	$1.12428e-01$	11.7
3	19008	$1.15484e-01$	46.9
4	74880	$1.16651e-01$	195.1
5	297216	$1.17098e-01$	796.2
	Adaptive Refinement, Grid1		
1	825	$1.05085e-01$	2.2
2	1362	$1.05990e-01$	2.9
3	2616	$1.11482e-01$	5.4
4	5334	$1.13978e-01$	11.3
5	10281	$1.15794e-01$	23.4
6	21546	$1.16915e-01$	51.6
7	40584	$1.17253e-01$	108.2
8	86259	$1.17379e-01$	233.0
9	164046	$1.17499e-01$	406.9
10	330930	$1.17530e-01$	830.4

Table 6.7: Pressure difference

6.2 The nonstationary case

The discretization and the solution procedure presented in the previous section can naturally be generalized to the nonstationary case by the use of an adequate discretization of the time derivative. The system resulting from an application of, for example the implicit Euler or Crank-Nicholson scheme differs from the one described above only through the addition of a term of the form $\frac{1}{k}M$ in the velocity block, where M denotes the mass matrix. Since there is no time-derivative acting on the pressure variable, the resulting system behaves in the limit case $k \to 0$ as a mixed discretization of the Neumann problem. It is therefore even for small k a complex system to be solved in each time-step. To avoid the difficulty of inverting such a system with condition number $O(h^{-2})$, we present an adaptation of the classical projection methods of Chorin [18] and Van Kan [55]. An error analysis for these splitting methods on the continuous level is given by Prohl in [42] and [43]. Since these algorithms modify the equations in an inconsistent way, the Galerkin character of the discretization is lost. Temporal and spatial errors get mixed up. The design of an optimal adaptive algorithm is therefore difficult. However, we can also use the projection method as a preconditioner for the coupled system of equation, as proposed by Turek [53]. This means using some kind of Laplace operator as a preconditioner for the pressure Schur complement. This approach has been used in [17]. An analysis can be found in [14].

6.2.1 The pressure correction scheme

We turn our attention to the nonstationary incompressible Navier–Stokes equations:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \cdot \nabla u - \nu \Delta u + \nabla p &= f \\ \nabla \cdot u &= 0, \end{aligned}$$

supplemented with the same boundary conditions as in the stationary case and additionally an initial conditions for the velocity. For simplicity, we consider here time discretization by the implicit Euler method. Other higher-order one-step schemes are readily adopted and will be mentioned later.

In order to point out the main idea, we shortly discuss the splitting method in the framework of a semi-discretization in time. We set for abbreviation $A(w) \equiv w \cdot \nabla - \nu\Delta$ to obtain:

$$\begin{aligned} \frac{u^n - u^{n-1}}{k} + A(u^n)u^n + \nabla p^n &= f^n \\ \nabla \cdot u^n &= 0, \end{aligned} \tag{6.4}$$

with time step k.
The splitting scheme, introducing an intermediate velocity field $\tilde{u}$ reads as follows:

$$1. \quad \frac{\tilde{u}_n - u_{n-1}}{k} + A(\tilde{u}_n)\tilde{u}_n + \sigma\nabla p_{n-1} = f,$$

and

$$\begin{aligned} 2. \qquad \frac{u_n - \tilde{u}_n}{k} + \nabla(p_n - \sigma p_{n-1}) &= 0 \\ \nabla \cdot u_n &= 0. \end{aligned}$$

While the first step advances the velocity field ignoring the divergence constraint, the second step performs a L_2–projection onto the subspace of divergence-free vectors. In the first step the complete boundary data are used to compute $\tilde{u}_n$. The second step uses the homogeneous Neumann condition for the $p_n - \sigma p_{n-1}$. Depending on the parameter σ, which might be chosen equal to 0 or 1, the corresponding Lagrange multiplier is used as the new pressure or as a pressure correction. This leads to the algorithms known as Chorin and Van Kan scheme, respectively.
The correction step can equivalently be written as a Neumann problem:

$$\begin{aligned} -\Delta q &= -\nabla \cdot \tilde{u}_n \quad \text{with} \quad \frac{\partial q}{\partial n}|_{\partial\Omega} = 0 \\ u_n &= \tilde{u}_n - \nabla q \\ p_n &= \sigma p_{n-1} + \frac{1}{k} q. \end{aligned} \tag{6.5}$$

Note that the Neumann equation for q is well-posed, since $\int_\Omega \nabla \cdot \tilde{u}_n \, dx = \int_\Gamma \tilde{u}_n \cdot n \, dS = 0$ due to the (compatible) Dirichlet condition. It is clear

from the velocity update in (6.5) that the projected u_n does not satisfy the Dirichlet boundary conditions (only the normal component is correct.).
A first interpretation of the splitting scheme can be given if we add the two steps in order to obtain the equations

$$\begin{aligned} \frac{u^n - u^{n-1}}{k} + A(\tilde{u}_n)\tilde{u}_n + \nabla p^n &= f^n \\ \nabla \cdot u^n &= 0. \end{aligned}$$

These equations look like (6.4) with the difference that $A(u^n)u^n$ is replaced by $A(\tilde{u}_n)\tilde{u}_n$. Therefore, this system does not allow the same boundary conditions as (6.4). In order to avoid the difficulty with the boundary condition, the schemes can be interpreted in another way. We derive a system involving the velocities $\tilde{u}$. For this, we replace u_{n-1} in the first step using the projection from the previous time step. In this way we arrive at:

$$\begin{aligned} \frac{\tilde{u}_n - \tilde{u}_{n-1}}{k} + A\tilde{u}_n + \nabla(p_{n-1} + \sigma(p_{n-1} - p_{n-2})) &= f \\ \nabla \cdot \tilde{u}_n - k\Delta(p_n - \sigma p_{n-1}) &= 0. \end{aligned} \tag{6.6}$$

For $\sigma = 1$, this results in

$$\begin{aligned} \frac{\tilde{u}_n - \tilde{u}_{n-1}}{k} + A\tilde{u}_n + \nabla(2p_{n-1} - p_{n-2}) &= f \\ \nabla \cdot \tilde{u}_n - k^2\Delta\frac{p_n - p_{n-1}}{k} &= 0, \end{aligned} \tag{6.7}$$

where the pressure gradient is approximated by the second order Adams–Bashforth formula. The divergence equation is disturbed by a second order term similar to the one known from the pressure stabilization.
The interpretation of the projection methods as system (6.6) is the key point in the derivation of error estimates. A theoretical analysis on this basis yielding optimal error estimates can be found in [43].
It is straightforward to construct a fully discrete approximation using (bi)-linear finite elements for the velocity and as well as for the pressure. There is, however, the usual difficulty with finite element spaces which do not satisfy the inf-sup condition. This can be seen in the following way: For a stationary solution the disturbance in the divergence equation vanishes and we end

up with the standard Galerkin discretization of the Navier–Stokes equation. These problems will not occur as long as the consistency error $k\Delta(p_n - \sigma p_{n-1})$ is greater than $h^2\Delta p$.
In order to avoid these problems for small time steps, we warrant stability by modifying the pressure update in the following fashion:

$$p_n = \sigma\frac{k}{k+\delta}p_{n-1} + \frac{1}{k+\delta}q.$$

In the case $\sigma = 1$, this alters the divergence equation in the following way: $\nabla \cdot u_n - h^2\Delta p_n = 0$ and $\nabla\cdot\tilde{u}_n - k^2\Delta\frac{p_n - p_{n-1}}{k} - h^2\Delta p_n = 0$.
Letting a denote the bilinear form corresponding to $A(\tilde{u})$, the discrete splitting scheme reads:

1. $(\frac{\tilde{u}_n - u_{n-1}}{k}, \varphi) + a(\tilde{u}_n, \varphi) + \sigma(\nabla p_{n-1}, \varphi) = (f, \varphi)$
2. $(\nabla q, \nabla\xi) = -(\nabla\cdot\tilde{u}_n, \xi)$
3. $(u_n - \tilde{u}_n, \varphi) - (\nabla q, \varphi) = 0$
4. $p_n = \sigma\frac{k}{k+\delta}p_{n-1} + \frac{1}{k+\delta}q.$

This can again be rewritten as a decoupled system:

$$(\frac{\tilde{u}_n - \tilde{u}_{n-1}}{k}, \varphi) + a(\tilde{u}_n, \varphi) + (\nabla(2p_{n-1} - p_{n-2}), \varphi) = (f, \varphi)$$
$$(\nabla\cdot\tilde{u}_n, \xi) + k^2(\nabla\frac{p_n - p_{n-1}}{k}, \nabla\xi) + \delta(\nabla p_n, \nabla\xi) = 0.$$

6.2.2 Outflow Boundary Condition

The L_2–projection onto J_0 scheme is well defined if $\int_\Gamma \tilde{u}_n\, dS = 0$ (see (6.5)). In the case of outflow boundary conditions there is no reason for this to hold true, since the first step of the splitting scheme decouples the pressure and does not satisfy the divergence condition, even in the spatial mean. Most practical applications, however, as the flow through a channel involve outflow conditions. This is particularly important for problems with moving boundaries where the computational domain changes in size. For a fixed inflow quantity the outflow quantity changes in time and the numerical scheme has to be able to take this into account.Therefore we modify the first step in

order to enforce $\int_\Gamma \tilde{u} \cdot n \, dS = 0$.
We slightly change our notations by setting

$$a(u,v) = (\frac{1}{k}u, v) + (u \cdot \nabla u, v) + \nu(\nabla u, \nabla v) \text{ and}$$
$$(g, \varphi) = (f, \varphi) + (\frac{1}{k}u_{n-1}, v) - \sigma(\nabla p_{n-1}, \varphi).$$

The first step of the splitting scheme just reads:

$$a(\tilde{u}_n, \varphi) = (g, \varphi) \quad \forall \varphi \in V_h.$$

Since we want to impose an additional condition, $b(\tilde{u}_n, 1) = 0$ with the usual divergence bilinear form $b(u,v) = \int \nabla \cdot uv \, dx$, we consider the following saddle point problem:

$$\begin{aligned} a(\tilde{u}_n, v) + b(v, \bar{p}) &= (g, v) \quad \forall v \in V_h \\ b(\tilde{u}_n, q) &= 0 \quad \forall q \in Q_h, \end{aligned}$$

with $Q_h = span(1)$. The real number $\bar{p}$ gives a mean pressure that is needed to compensate for the supplementary equation. This saddle point system can be solved iteratively in a few steps of block Gauss-Seidel since we are in the favorite position to be able to calculate the Schur complement which is also a real number. This iteration can naturally be coupled with the fixed point iteration for the nonlinearity.

References

[1] Apel, T., Dobrowolski, M.: *Anisotropic interpolation with applications to the finite element method*, Computing 47, 277-293 (1992).

[2] Arnold, D.N., Brezzi, F., Fortin, M.: *A stable finite element for the Stokes equations*, Calculo 21 (1984).

[3] Baines, M.J.: *An analysis of the moving finite-element method*, SIAM J. Num. Anal. 28, 1323-1349 (1991).

[4] Babuška, I., Rheinboldt, W.C. : *Error estimates for adaptive finite element computations*, SIAM J. Num. Anal. 15, 736-754 (1978).

[5] Bernardi, C., Bonnon, O., Langouët, C., Métivet, B. : *Residual error indicators for linear problems: Extension to the Navier-Stokes equations*, Proceedings of the 9th Int. Conf. on FE in fluids, Venezia (1995).

[6] Bertram, C.P., Pedley, T.J.: *A mathematical model of unsteady collapsible tube behaviour*, J. Biomed. 15, 39-50 (1982).

[7] Becker, R., Johnson, C., Rannacher, R. : *Adaptive error control for multigrid finite element methods*, to appear in Computing.

[8] DFG-SPP *Abschlußbericht Strömungssimulation mit Hochleistungsrechnern, 1996.*

[9] Becker, R., Rannacher, R. : *Finite element solution of the incompressible Navier-Stokes equations on anisotropically refined meshes*, Proceedings of the Tenth GAMM-Seminar, Notes on Num. Fluid Dyn., Vieweg (1995).

[10] Blum, H.: *Asymptotic error expansion and defect correction in the finite element method*, Habilitationsschrift, Universität Heidelberg, 1991.

[11] Blum, H., Lin, Q., Rannacher, R.: *Asymptotic error expansion and Richardson extrapolation for linear finite elements*, Num. Math, 49, 11-37 (1986).

[12] Bramble, J.: *Multigrid Methods*, Pitman, 1993.

[13] Bramble, J., Pasciak, J.: *New estimates for multilevel algorithms including the V-cycle*, Math. Comp., 60, 447-471 (1993).

[14] Bramble, J., Pasciak, J.: *Iterative techniques for time dependent Stokes problem*, preprint (1994).

[15] Brezzi, R.E., Fortin, T.: *Mixed and Hybrid Finite Element Methods*, Springer, Berlin, 1991.

[16] Cattabriga, L.: *Su un problema al contorno relativo al sistema di equazioni di Stokes*, Rend. Mat. Sem. Univ. Padova 31, 308-340 (1961).

[17] Cahouet, Chabard,: *Some fast 3D finite element solvers for the generalized Stokes problem*, Int. J. Nmer. Meth. Fluids 8, 869-895 (1988).

[18] Chorin, A.J..: *Numerical solution of the Navier-Stokes equations*, Math. Comp. 22, 745-762 (1968).

[19] Ciarlet, P.: *The finite element method for elliptic problems*, North Holland, 1980.

[20] Clément, P.: *Approximation by finite element functions using local regularization*, R.A.I.R.O. Anal. Numer. 9 R2, 77-84 (1975).

[21] Demirdžić, I., Perić, M.: *Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries*, Int. J. Nmer. Meth. Fluids 10, 771-790 (1990).

[22] Eriksson, K., Johnson, C. : *Adaptive finite element methods for parabolic problems: I. A linear model problem*, SIAM J. Num. Anal. 28, 43-77 (1991).

[23] Eriksson, K., Estep, D., Hansbo, P. and Johnson, C.: Adaptive Finite Element Methods, Manuscript 1994, to appear with North Holland.

[24] Franca, L.P., Frey, S.L.: *Stabilized finite element methods: II. The incompressible Navier-Stokes equations*, Comp. Meth. Appl. Mech. Eng.,99, 209-233 (1992).

[25] Girault, V., Raviart, P.A.: *Finite Element Methods for Navier-Stokes Equations*, Springer, Berlin, 1986.

[26] Hackbusch, W.: *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.

[27] Hansbo, P., Szepessy, A.: *A velocity-pressure streamline diffusion method for the incompressible Navier-Stokes equations*, Comp. Meth. Appl. Mech. Eng., 84, 175-192 (1990).

[28] Harig, J.: *Eine robuste und effiziente Finite Elemente Methode zur Lösung der inkompressiblen 3-D Navier-Stokes-Gleichungen auf Vektorrechnern*, Thesis, Universität Heidelberg, 1991.

[29] Harig, J.: *Eine robuste und effiziente Finite Elemente Methode zur Lösung der inkompressiblen 3-D Navier-Stokes-Gleichungen auf Vektorrechnern*, Thesis, Universität Heidelberg, 1991.

[30] Heywood, J., Rannacher, T., Turek, S.: *Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations*, Int. J. Nmer. Meth. Fluids (to appear)

[31] Hughes, T.J.R., Franca, L.P., Balestra, M.: *A new finite element formulation for computational fluid dynamics: V. Cicumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation for the Stokes problem accomodating equal order interpolation*, Comp. Meth. Appl. Mech. Eng. 59, 85-99 (1986).

[32] Jamet, P.: *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain*, SIAM J. Numer. Anal. 15, 912-928 (1978).

[33] Jamet, P.: *Stability and convergence of a generalized Crank-Nicolson scheme on a variable mesh for the heat equation*, SIAM J. Numer. Anal. 17, 530-539 (1980).

[34] Johnson, C.: *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge Univerity Press, 1987.

[35] Johnson, C., Rannacher, R. and Boman, M.: *Numerics and hydrodynamic stability: Towards error control in CFD*, SIAM J. Numer. Anal. 32 (1995).

[36] Lesaint, P., Touzani, R.: *Approximation of the heat equation in a variable domain with application to the Stefan problem*, SIAM J. Numer. Anal. 26, 366-379 (1989).

[37] Lions, J.L.: *Sur les problèmes mixtes pour certains systèmes paraboliques dans des ouverts non cylindriques*, Ann. Inst. Fourier 7, 143-928 (1957).

[38] Ling, S.C., Atabek, H.B.: *A nonlinear analysis of pulsatile flow in arteries*, J. Fluid Mechanics 3, 493-511 (1972).

[39] Lynch, D.R., Gray, W.G.: *Finite element simulation of flow in deforming regions*, J. Comp. Phys. 36, 135-153 (1980).

[40] Nomura, T., Hughes, T.J.R.: *An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body*, Comp. Meth. Appl. Mech. Eng. 95, 115-138 (1992).

[41] Pedley, T.J., Stephanoff, K.D.: *Flow along a channel with a time-dependent indentation in one wall: the generation of vorticity waves*, J. Fluid Mechanics, 160, 337-367 (1985).

[42] Prohl, A.: *Über die Chorinsche Methode zur Lösung der inkompressiblen Navier-Stokes Gleichungen*, Diploma Thesis (1992).

[43] Prohl, A.: *Quasi-Kompressibilitätsmethoden zur Lösung der inkompressiblen Navier-Stokes Gleichungen*, PhD Thesis, 1995.

[44] Ralph, M.E., Pedley, T.J.: *Flow in a channel with a moving indentation*, J. Fluid Mech. 190, 87-112 (1988).

[45] Rannacher, R., Turek, S.: *Simple nonconforming quadrilateral Stokes element*, Num. Meth. Part. Diff. Eqns. 8, 97-111 (1992).

[46] Rosenfeld, M., Kwak, D.: *Time-dependent solutions of viscous incompressible flows in moving co-ordinates*, Int. J. f. Num. Meth. in Fluids, 13, 1311-1328 (1991).

[47] Scott, R., Zhang, D.: *Interpolation of non-smooth functions*, Math. Comp. 13, 1311-1328 (1991).

[48] Solonnikov, V.A.: *On a boundary value problem for the Navier–Stokes equations with discontinuous boundary data*, Rendiconti di Matematica, Serie 7, Volume 10, 757-772 (1990).

[49] Tezduyar, T.E., Behr, M., Liou, J.: *A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/ space-time procedure: 1. The concept and the preliminary numerical tests*, Comp. Meth. Appl. Mech. Eng. 94, 339-351 (1992).

[50] Thomée, V.: *Galerkin finite element method for parabolic equations*, Lecture Notes in Mathematics 1054, Springer, Berlin, 1984.

[51] Tobiska, L., Verfürth, R.: *Analysis of a streamline diffusion finite element method for the Stokes and Navier-Stokes equations*, to appear in SIAM J. Numer. Anal.

[52] Turek, S.: *Tools for simulating non–stationary incompressible flow via discretely divergence–free finite element models* Int. J. Numer. Meth. Fluids, 18, 71–105 (1994).

[53] Turek, S.: *A comparative study of some time-stepping techniques for the incompressible Navier-Stokes equations*, Preprint 95-10 (SFB 359), Universität Heidelberg (1995).

[54] Vanka, S.P.: *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*, J. Comp. Phys. 65, 138-158 (1986).

[55] Van Kan, J.: *A second-order accurate pressure-correction scheme for viscous incompressible flows*, SIAM J. Sci. Stat. Comp. 7, 870-891 (1986).

[56] Verfürth, R.: *A posteriori error estimators for the Stokes equations*, Num. Math. 55, 309-325 (1989).

[57] Verfürth, R.: *A posteriori error estimation and adaptive mesh-refinement techniques*, J. Comp. Apl. Math. (1993).

[58] Wathen, A., Sylvester, D.: *Fast iterative solution of stabilized Stokes systems, Part I. Using simple diagonal preconditioners*, SIAM J. Numer. Anal. 30, 630-649 (1993).

[59] Zhou, G.: *How accurate is the streamline diffusion method finite element method?*, Preprint 95-22 (SFB 359), 1995.