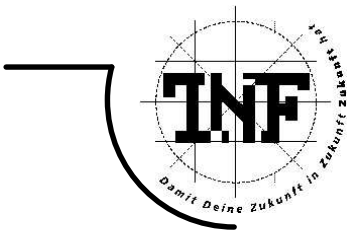
JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis



Algebraic Multigrid Methods for the Numerical Solution of the Incompressible Navier-Stokes Equations

Dissertation

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

Angefertigt am *Institut für Numerische Mathematik*

Begutachter:

A.Univ.Prof. Dipl.-Ing. Dr. Walter Zulehner
Prof. Dr. Arnold Reusken

Eingereicht von:

Dipl.-Ing. Markus Wabro

Mitbetreuung:

O.Univ.Prof. Dipl.-Ing. Dr. Ulrich Langer

Linz, August 2003

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Diese Dissertation wurde bisher weder im In- noch im Ausland als Prüfungsarbeit vorgelegt.

Dipl.-Ing. Markus Wabro

Linz, August 2003

Abstract

If the Navier-Stokes equations for incompressible fluids are linearized using fixed point iterations, the Oseen equations arise. In this thesis we provide concepts for the coupled algebraic multigrid (AMG) solution of this saddle point system, where 'coupled' here is meant in contrast to methods, where pressure and velocity equations are iteratively decoupled, and 'standard' AMG is used for the solution of the resulting scalar problems.

We show how the coarse levels can be constructed (where their stability is an important issue) and which smoothers (known from geometric multigrid methods for saddle point systems) can be used.

To prove the efficiency of our methods experimentally, we apply them to finite element discretizations of various problems (model problems and also more complex industrial settings) and compare them with classical approaches.

Zusammenfassung

Durch die Fixpunkt-Linearisierung der Navier-Stokes Gleichungen für inkompressible Fluide erhält man die sogenannten Oseen Gleichungen. In der vorliegenden Arbeit entwickeln wir Konzepte für die numerische Lösung dieses Sattelpunktsystems durch gekoppelte algebraische Mehrgittermethoden (AMG), wobei "gekoppelt" im Gegensatz zu Vefahren steht, bei denen Druck- und Geschwindigkeitsgleichungen iterativ entkoppelt werden und 'Standard'-AMG zur Lösung der entstehenden skalaren Probleme angewandt wird.

Wir präsentieren Möglichkeiten der Konstruktion der Grobgittersysteme (wobei insbesondere auf deren Stabilität geachtet wird) und der Anwendung von Glättern, welche von geometrischen Mehrgittermethoden für Sattelpunktgleichungen her bekannt sind.

Die Effizienz der entwickelten Methoden wird schließlich experimentell gezeigt, indem sie sowohl für einfachere Modellprobleme als auch für durchaus komplexe industrielle Anwendungen getestet und mit den klassischen Methoden verglichen werden.

Acknowledgements

First I owe a great debt of gratitude to Walter Zulehner, not only for advising my thesis but for arousing my interest and deepening my understanding in the scientific field I am currently working in.

I would also like to thank Ulrich Langer who, as leader of the project and head of the institute I am employed in, provided me with scientific and financial background, and who made constructive suggestions on many details of this thesis, and I want to thank Arnold Reusken for co-refereeing this dissertation.

Special thanks go to Ferdinand Kickinger who supplied me with industrial test cases and with whom I had some very fruitful discussions, and to all other people who are responsible for some aspects of this thesis by "asking the right questions".

This work has been supported by the Austrian Science Foundation "Fonds zur Förderung der wissenschaftlichen Forschung (FWF)" under the grant P14953 "Robust Algebraic Multigrid Methods and their Parallelization".

Contents

Notation

We generally use standard characters for scalar values and scalar functions (p, q,...) and boldface characters for vectors and vector valued functions ($\mathbf{u}$, $\mathbf{v}$,...). We will use the underline notation (which will be introduced in detail in section 2.2) for finite-element vectors associated to scalar or vector valued functions ($\underline{p}$, $\underline{q}$,..., resp. $\underline{\mathbf{u}}$, $\underline{\mathbf{v}}$, ...). The components of vectors are denoted by $(u_1, \dots, u_n)^T = \mathbf{u}$. For matrices we use capital letters (A,...), or component-notation $A = (a_{ij})_{i,j}$.

$\mathcal{G}$ is an open, connected subset of $\mathbb{R}^d$ with space dimension d (generally $d = 2$ or 3), $\partial\mathcal{G}$ its boundary.

Operators

$\mathbf{u} \cdot \mathbf{v}$	$= \sum_{i=1}^{d} u_i v_i$ (scalar product).
$\mathbf{u} \otimes \mathbf{v}$	$= (u_i v_j)_{i,j=1,\dots,d}$ (tensor product).
$\partial_j p$	$= \frac{\partial p}{\partial x_j}$ (partial derivative of p with respect to x_j).
$\partial_j \mathbf{u}$	$= (\partial_j u_i)_{i=1,\dots,d}$.
$\partial_t p$	$= \frac{\partial p}{\partial t}$ (partial derivative of p with respect to t) .
$\partial_t \mathbf{u}$	$= (\partial_t u_i)_{i=1,\dots,d}$.
∇p	$= (\partial_i p)_{i=1,\dots,d}$ (gradient of p).
$\nabla \mathbf{u}$	$= (\partial_i u_j)_{i,j=1,\dots,d}$.
$\operatorname{div} \mathbf{u}$	$= \sum_{i=1}^{d} \partial_i u_i$ (divergence of $\mathbf{u}$).
$(\mathbf{u} \cdot \nabla)\varphi$	$= \sum_{j=1}^{d} u_j \partial_j \varphi$.
$(\mathbf{u} \cdot \nabla)\mathbf{v}$	$= (\sum_{j=1}^{d} u_j \partial_j v_i)_{i=1,\dots,d}$.

Function spaces

$C(\mathcal{G})$	space of continuous functions on $\mathcal{G}$.
$C^k(\mathcal{G})$	space of functions with continuous k-th derivative on $\mathcal{G}$.
$C_0^\infty(\mathcal{G})$	space of infinitely smooth functions with compact support in $\mathcal{G}$.
$C^\infty(\bar{\mathcal{G}})$	space of infinitely smooth functions on $\bar{\mathcal{G}}$.
$L^p(\mathcal{G})$	Lebesgue space of measurable functions q with finite norm $\|q\|_{0,p} = \left(\int_{\mathcal{G}} \|q\|^p\right)^{\frac{1}{p}}$.

$W_p^k(\mathcal{G})$	Sobolev space of functions with k-th derivatives in $L^p(\mathcal{G})$.
$H^k(\mathcal{G})$	$= W_2^k(\mathcal{G})$.
$H_0^1(\mathcal{G})$	the closure of $C_0^\infty(\mathcal{G})$ in $H^1(\mathcal{G})$.
$H^{-1}(\mathcal{G})$	the dual space of $H_0^1(\mathcal{G})$.
$\mathbb{N}$	The natural numbers.
$\mathbb{Z}$	The integer numbers.
$\mathbb{R}$	The real numbers.

Norms

$\|q\|_0$	$= \|q\|_{0,2}$ for $q \in L^2(\mathcal{G})$.
$\|q\|_1$	$= \|\nabla q\|_0$ for $q \in H^1(\mathcal{G})$.
$\|q\|_1$	$= \sqrt{\|q\|_0^2 + \|\nabla q\|_0^2}$ for $q \in H^1(\mathcal{G})$.
$\|\mathbf{v}\|_X$	$= \sqrt{\mathbf{v}^T X \mathbf{v}}$ for $\mathbf{v} \in \mathbb{R}^n$ and a symmetric positive definite matrix $X \in \mathbb{R}^{n\times n}$.
$\|\mathbf{v}\|_{\ell_2}$	$= \sqrt{\mathbf{v}^T \mathbf{v}}$ for $\mathbf{v} \in \mathbb{R}^n$.
$\|Y\|_\alpha$	$= \sup_{0\neq\mathbf{v}\in\mathbb{R}^n} \frac{\|Y\mathbf{v}\|_\alpha}{\|\mathbf{v}\|_\alpha}$ for $Y \in \mathbb{R}^{n\times n}$ (consistent matrix norm to the vector norm $\|.\|_\alpha$).
$\|Y\|_F$	$= \sqrt{\sum_{i,j=1}^n y_{ij}^2}$ for $Y \in \mathbb{R}^{n\times n}$ (Frobenius norm).

Often used Indices, etc.

d	Space dimension.
L	Total number of multigrid levels.
l	Index indicating a certain multigrid level ($l \in \{1, \ldots, L\}$).
D	Index indicating a diffusive term or Laplacian.
C	Index indicating a convective term.
R	Index indicating a reaction term.
S	Index indicating a stabilization term.
s	Index used if we want to emphasize that some operator is scalar.

Chapter 1

Introduction

A very important set of partial differential equations in the field of computational fluid dynamics are the Navier-Stokes equations. They are capable of describing various phenomena of (in our case incompressible) Newtonian fluid flow, but give rise to many nontrivial mathematical problems despite of their relatively simple outer form. So, for example, the existence and smoothness of solutions of their non-stationary form are currently the topic of one of the prominent one-million-dollar-problems [Fef00, Dic00]. This thesis will unfortunately make no contribution to that aspect (in all probability), but to an efficient numerical solution of the equations.

After deciding which kind of nonlinear iteration to use (in our case fixed point iteration, which leads to the Oseen equations) and which discretization to choose (in our case the finite element method) one obtains an (indefinite) saddle point problem, which has to be solved. Classical iterative methods for that are variants of SIMPLE schemes (as introduced by Patankar and Spalding [PS72]) or Uzawa's algorithm [AHU58], having in common an iterative decoupling of the saddle point system into separate equations for pressure and velocity, which then can be solved with methods known for the solution of positive definite systems.

A milestone for the efficient solution of scalar, elliptic problems was set with the development of geometric multigrid (GMG) methods, for example by Federenko [Fed61], Bachvalov [Bac66], Astrachancev [Ast71], Brandt [Bra73], or Hackbusch [Hac76] to name but a few (see also the monographs e.g. by Korneev [Kor77], Hackbusch [Hac85], Wesseling [Wes92], Bramble [Bra93], or Trottenberg et.al. [TOS01]). The idea of these methods is to split the process into two parts, a smoothing of the error (i.e. a reduction of its high frequency components) and a correction step on a coarser grid.

First steps in the application of multigrid algorithms to saddle point systems were made by Verfürth [Ver84b] and Wittum [Wit89]. Further important work in this direction was done by Braess and Sarazin, who showed that it is possible to use the classical Uzawa method as smoothing iteration [BS97].

When confronted with "real life" applications with complex three dimensional geometries, a hierarchical refinement of a 'coarse' initial mesh — which is needed by geometric multigrid methods — would be impossible with respect to the limitations on computer

memory and CPU speed of today's generation of computer hardware. A solution to this problem are the algebraic multigrid (AMG) methods, where the initial mesh is used as finest level, and the coarser levels are generated using (almost) only information of the algebraic system.

A second reason for the popularity of AMG methods is their "black-box" character. In an ideal situation the user does not need to construct any hierarchy, the method operates on one single algebraic system and can therefore be used e.g. as a replacement for the direct solver on the coarsest level of a geometric multigrid algorithm.

Since the pioneering work of Ruge and Stüben [RS86] and Brandt et al. [BMR84] these methods have been applied to a wide class of linear systems arising (mostly) from scalar partial differential equations. For an overview of the technique itself and various applications we refer for example to Stüben [Stü01b].

For the application of AMG to saddle point problems one has the same two general possibilities as in the geometric multigrid case. The first is the segregated approach, i.e. to use a classical method (Uzawa, SIMPLE,...) for an outer iteration and to apply AMG to the resulting elliptic problems. This approach is described e.g. by Griebel et al. [GNR98] or Stüben [Stü01a]. Another idea in this class is to use a Krylov space method such as GMRES or BiCGstab with a special preconditioner which again decouples velocity and pressure equations. This was done for example by Silvester et al. first for the Stokes case [SW94] and later for the Navier-Stokes problem [SEKW01].

The focus of our work lies on the second possibility, on the coupled approach where an AMG method for the whole saddle point system is developed (as mentioned above for GMG methods). Work in this direction has been done for example by Webster [Web94] and Raw [Raw95] for finite volume discretizations of the Navier-Stokes equations, by Bertling [Ber02] for a finite element discretization of the Stokes equations, by Adams for contact problems in solid mechanics [Ada03], and by Bungartz for constrained optimization (with a small number of constraints) [Bun88].

This thesis is structured as follows. The second chapter contains the preliminaries which are needed for a numerical solution of the Navier-Stokes equations. We start with the problem statement, continue with the weak formulation and the finite element discretization, sketch the analysis of the associated Stokes problem, mention some problems induced by the convection, and finally discuss classical solution methods for the linear system.

In the third chapter we introduce algebraic multigrid methods. In this chapter we will apply it only to scalar equations, but the underlying ideas will be important for the saddle point case, too.

The central part of this work is chapter four, where we develop methods for the coupled application of AMG methods to saddle point systems. We provide ideas for the construction of multigrid hierarchies for different types of mixed finite elements, and we will deal with stability problems which may occur on coarse levels. Unfortunately (but not surprisingly) we were not able to construct a "black box method" capable of any saddle point problem, with whatever choice of discretization on an arbitrary mesh. All our methods depend for example on the concrete choice of the finite element.

Finally, chapter five is devoted to the presentation of numerical results. After a short

overview of the software package which was developed during the working on this thesis, we compare different aspects of the methods presented in the first three chapters for various problems, up to flows in fairly complex three dimensional geometries.

In most of the tests we observe advantages of the coupled approach, it seems as if it pays off to keep the structure of the problem on the coarser levels. Although there is still work to be done, the results we have are really promising.

Chapter 2

Preliminaries

The classical process for the numerical solution of partial differential equations (describing a physical phenomenon, in our case the Navier-Stokes equations describing the flow of an incompressible fluid, or the related Oseen or Stokes equations) is to derive a weak formulation, provide analysis, discretize the system (in our case with finite elements), and finally to solve the resulting linear algebra problems.

This first chapter contains the parts of this process, from the problem formulation to (non-multigrid) solution methods for the arising linear systems.

2.1 Navier-Stokes Equations

Our main point of investigation will be the *Navier-Stokes equations* for incompressible flow (Claude Navier, 1785–1836, and George Stokes, 1819–1903). A mathematically rigorous derivation from fundamental physical principles and conservation laws can be found in [Fei93].

We denote by $\mathbf{u}$ the velocity of the fluid, p the static pressure, ρ the density of the fluid, μ its viscosity and $\mathbf{f}$ some outer force. Then the instationary flow of incompressible Newtonian fluids in a domain $\mathcal{G}$ (where $\mathcal{G}$ is an open, connected subset of $\mathbb{R}^d$ with Lipschitz continuous boundary $\partial\mathcal{G}$) is governed by

$$\rho\frac{\partial}{\partial t}\mathbf{u} - \mu\Delta\mathbf{u} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} \tag{2.1a}$$

$$\operatorname{div}\mathbf{u} = 0. \tag{2.1b}$$

Equation (2.1a) expresses Newton's law of motion, (2.1b) the conservation of mass.

The underlying physical assumption for these equations to hold are incompressibility and Stokes' hypothesis for the stress tensor

$$\mathbf{T}(\mathbf{u}, p) = -p\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right), \tag{2.2}$$

for incompressible Newtonian fluids.

Physical similarity

With the choice of scales $\mathbf{u} = V\mathbf{u}^*$, $\mathbf{x} = L\mathbf{x}^*$, $t = L/Vt^*$, $p = V^2\rho p^*$ and $\mathbf{f} = V^2/L\mathbf{f}^*$ (with a characteristic velocity V and a characteristic length L) we get the dimensionless formulation

$$\frac{\partial}{\partial t^*}\mathbf{u}^* - \nu\Delta^*\mathbf{u}^* + (\mathbf{u}^*\cdot\nabla^*)\mathbf{u}^* + \nabla^* p^* = \mathbf{f}^*, \tag{2.3a}$$

$$\operatorname{div}^* \mathbf{u}^* = 0, \tag{2.3b}$$

where

$$\nu := \frac{1}{Re} := \frac{\mu}{\rho LV},$$

with the dimensionless Reynolds number Re (Osborne Reynolds, 1842–1912). For simplicity in notation we will omit the stars in the following.

We primarily consider two types of boundary conditions (more can be found e.g. in [Tur99]). Let $\partial\mathcal{G} = \Gamma_1 \cup \Gamma_2$. On Γ_1 we prescribe Dirichlet conditions

$$\mathbf{u}|_{\Gamma_1} = \mathbf{u}_1,$$

on Γ_2 natural outflow conditions of the form

$$(-pI + \nu\nabla\mathbf{u}^T)\cdot\mathbf{n} = \mathbf{0}.$$

In the non-stationary case we also need a pair of initial conditions

$$\mathbf{u}|_{t=0} = \mathbf{u}_0, \quad p|_{t=0} = p_0.$$

If we linearize the system by fixed point iteration we get the so called *Oseen equations* (Carl Wilhelm Oseen, 1879–1944)

$$\frac{\partial}{\partial t}\mathbf{u} - \nu\Delta\mathbf{u} + (\mathbf{w}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \tag{2.4a}$$

$$\operatorname{div}\mathbf{u} = 0, \tag{2.4b}$$

where $\mathbf{w}$ is the old approximation of the velocity, sometimes also called the wind.

Dropping the convection term leads to the *Stokes equations*

$$\frac{\partial}{\partial t}\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f}, \tag{2.5a}$$

$$\operatorname{div}\mathbf{u} = 0. \tag{2.5b}$$

Setting $\frac{\partial}{\partial t}\mathbf{u} \equiv 0$ gives the stationary versions of (2.3), (2.4), and (2.5).

2.1.0.1 Weak Formulation of the Stationary Problem

Assume for now that we want to solve the stationary problem, we will return to the time dependent problem in Section 2.3.

Assumption. There exists $\tilde{\mathbf{u}}_1 \in H^1(\mathcal{G})^d$ with

$$\operatorname{div} \tilde{\mathbf{u}}_1 = 0 \quad \text{in } \mathcal{G}, \tag{2.6}$$
$$\tilde{\mathbf{u}}_1 = \mathbf{u}_1 \quad \text{on } \Gamma_1. \tag{2.7}$$

Now let

$$\begin{aligned} \mathbf{U} &:= \left(H_0^1(\mathcal{G})\right)^d, \\ \mathbf{U}(\tilde{\mathbf{u}}_1) &:= \left\{\mathbf{v} \in H^1(\mathcal{G})^d : \mathbf{v} - \tilde{\mathbf{u}}_1 \in \mathbf{U}\right\}, \\ Q &:= \left\{q \in L^2 : \int_{\mathcal{G}} q \,\mathrm{d}\mathbf{x} = 0\right\}. \end{aligned}$$

Then one can derive the weak formulation of the Navier-Stokes equations: Find $\mathbf{u} \in \mathbf{U}(\tilde{\mathbf{u}}_1)$ and $p \in Q$ such that

$$\bar{a}(\mathbf{u}; \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle F, \mathbf{v}\rangle \quad \forall \mathbf{v} \in \mathbf{U}, \tag{2.8a}$$
$$b(\mathbf{u}, q) = 0 \quad \forall q \in Q, \tag{2.8b}$$

where

$$\bar{a}(\mathbf{w}; \mathbf{u}, \mathbf{v}) = a_D(\mathbf{u}, \mathbf{v}) + a_C(\mathbf{w}; \mathbf{u}, \mathbf{v}),$$

and

$$\begin{aligned} a_D(\mathbf{u}, \mathbf{v}) &= \nu(\nabla \mathbf{u}, \nabla \mathbf{v}), \\ a_C(\mathbf{w}; \mathbf{u}, \mathbf{v}) &= \left((\mathbf{w} \cdot \nabla)\mathbf{u}, \mathbf{v}\right), \\ b(\mathbf{u}, q) &= -(\operatorname{div} \mathbf{u}, q), \end{aligned}$$

$\tilde{\mathbf{u}}_1$ as in (2.6), (2.7) and

$$\langle F, .\rangle = (\mathbf{f}, .)_0.$$

2.1.1 Analysis of the Associated Stokes Problem — the Inf-Sup Condition

We sketch the analysis of the associated stationary Stokes problem with homogeneous Dirichlet boundary conditions, because here one can get a first impression of the importance of a major criterion for stability — the inf-sup condition — which appears again and again in the analysis and numerical solution of mixed problems. The associated Stokes problem reads as: Find $(\mathbf{u}, p) \in \mathbf{U} \times Q$ such that

$$a_D(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle F, \mathbf{v}\rangle \quad \forall \mathbf{v} \in \mathbf{U}, \tag{2.9a}$$
$$b(\mathbf{u}, q) = 0 \quad \forall q \in Q. \tag{2.9b}$$

Define

$$\mathbf{V} = \{\mathbf{v} \in \mathbf{U} : b(v, q) = 0 \text{ for all } q \in Q\}.$$

The first step is to show existence and uniqueness of solutions of the following subproblem: Find $\mathbf{u} \in \mathbf{V}$ such that for all $\mathbf{v} \in \mathbf{V}$

$$a_D(\mathbf{u}, \mathbf{v}) = \langle F, \mathbf{v} \rangle . \tag{2.10}$$

Theorem 2.1. *Problem (2.10) has a unique solution.*

proof (sketch). a_D is a bilinearform, and because one can show that a_D is $\mathbf{V}$-elliptic and continuous and that $F = (\mathbf{f}, .)_0$ is contained in the dual space of $\mathbf{V}$, the theorem of Lax and Milgram completes the proof (c.f. [BF91]). □

What remains is to find a unique $p \in Q$ solving the problem

$$b(\mathbf{v}, p) = \langle F, \mathbf{v} \rangle - a_D(\mathbf{u}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{U}, \tag{2.11}$$

where $\mathbf{u}$ is the solution of (2.10). We define $B^* : Q \to \mathbf{U}^*$, $B^*p = b(., p)$, where $\mathbf{U}^*$ denotes the dual space of $\mathbf{U}$, and rewrite (2.11) as

$$B^*p = \langle F, . \rangle - a_D(\mathbf{u}, .), \tag{2.12}$$

with the right hand side being element of the polar set

$$\mathbf{V}^0 := \{l \in \mathbf{U}^* : l(\mathbf{v}) = 0 \text{ for all } \mathbf{v} \in \mathbf{V}\}.$$

The following theorem introduces the already mentioned criterion for the solvability of (2.11) resp. (2.12).

Theorem 2.2. *The operator $B^* : Q \to \mathbf{V}^0$ is an isomorphism if and only if there exists a constant $c_0 > 0$ such that*

$$\inf_{0 \neq q \in Q} \sup_{\mathbf{0} \neq \mathbf{v} \in \mathbf{U}} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_{\mathbf{U}} \|q\|_Q} \geq c_0. \tag{2.13}$$

The proof is based on the closed-range theorem (see e.g. [Yos80]) and can be found for example in [GR86] or [Bra97]. Condition (2.13) is called *LBB condition* (after Ladyzhenskaya, Babuška, and Brezzi) or *inf-sup condition*.

For instance in [GR86] it is shown that in our concrete case $b(., .)$ fulfills the inf-sup condition, thus we can combine the theorems above to the following.

Theorem 2.3. *Problem (2.9) is uniquely solvable.*

2.2 Finite Element Discretization

We will briefly introduce the concept of mixed Finite Element Methods (FEM). Details can be found e.g. in [Pir89] or [Bra97].

We assume from now on, that $\mathcal{G}$ is a polygonal resp. polyhedral domain.

Let $\mathbf{U}_h$ and Q_h be finite-dimensional subspaces of $\mathbf{U}$ and Q, respectively, and let

$$\begin{aligned}\mathbf{U}_h(\tilde{\mathbf{u}}_1) &:= \left\{\mathbf{v} \in H^1(\mathcal{G})^d : \mathbf{v} - \tilde{\mathbf{u}}_1 \in \mathbf{U}_h\right\},\\ \mathbf{V}_h &:= \{\mathbf{v}_h \in \mathbf{U}_h : b(\mathbf{v}_h, q_h) = 0 \text{ for all } q_h \in Q_h\}.\end{aligned}$$

Now we can formulate a discrete version of problem (2.8): Find a couple $(\mathbf{u}_h, p_h) \in \mathbf{U}_h(\tilde{\mathbf{u}}_{1h}) \times Q_h$ such that

$$\bar{a}(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \langle F, \mathbf{v}_h\rangle \quad \forall \mathbf{v}_h \in \mathbf{U}_h, \tag{2.14a}$$

$$b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h, \tag{2.14b}$$

where $\tilde{\mathbf{u}}_{1h}$ is a reasonable approximation of $\tilde{\mathbf{u}}_1$.

For reasons which will become obvious later we extend problem (2.14) to

$$\begin{aligned}\bar{a}(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) &= \langle F, \mathbf{v}_h\rangle \quad \forall \mathbf{v}_h \in \mathbf{U}_h,\\ b(\mathbf{u}_h, q_h) - c(p_h, q_h) &= \langle G, q_h\rangle \quad \forall q_h \in Q_h,\end{aligned} \tag{2.15}$$

where $c(.,.)$ is a positive semi-definite bilinearform and $G \in Q^*$ (both may be identical zero).

The following theorem shows that again the inf-sup condition is of major importance (for the proof we refer to [GR86]).

Theorem 2.4. *Assume that a_D is $\mathbf{V}_h$-elliptic (with h independent ellipticity constant) and that there exists a constant $c_0 > 0$ (independent of h) such that the discrete inf-sup condition*

$$\inf_{0\neq q\in Q_h} \sup_{\mathbf{0}\neq \mathbf{v}\in \mathbf{U}_h} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_{\mathbf{U}}\|q\|_Q} \geq c_0, \tag{2.16}$$

holds.

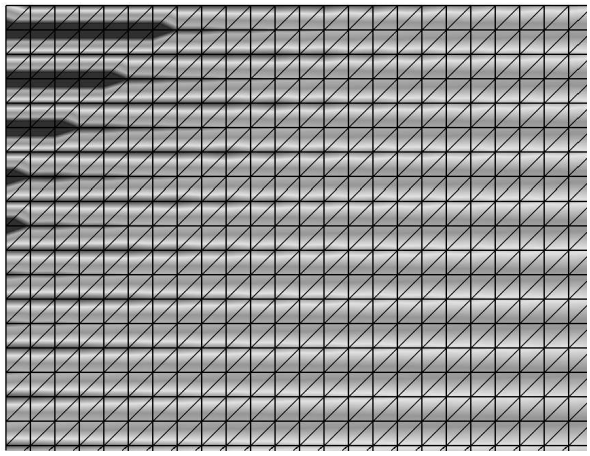
Then the associated (discretized, stationary) Stokes problem has a unique solution $(\mathbf{u}_h, p_h)$, and there exists a constant c_1 such that

$$\|\mathbf{u} - \mathbf{u}_h\|_{\mathbf{U}} + \|p - p_h\|_Q \leq c_1 \left(\inf_{\mathbf{v}_h\in\mathbf{U}_h} \|\mathbf{u} - \mathbf{v}_h\|_{\mathbf{U}} + \inf_{q_h\in Q_h} \|p - q_h\|_Q\right), \tag{2.17}$$

where $(\mathbf{u}, p)$ is the solution of (2.9).

Remark 2.5. *In literature (e.g. [GR86], [BF91], or [Bra97]) one can find prominent examples of what can go wrong with elements not fulfilling the inf-sup condition ('checkerboard'-instabilities, spurious pressure modes, etc.). The discrete solution may contain unphysical oscillations and may for $h \to 0$ not converge to the solution of the continuous problem, what is illustrated in Figure 2.1*

Figure 2.1 Part of the discrete pressure solution for a driven cavity problem discretized with an unstable element (unstabilized P_1-P_1 element, see Section 2.2.1.2). Oscillations in the pressure can be observed (light-grey indicates high pressure, dark-grey low pressure)



If a basis of Q_h is given by $\{\psi_1, \ldots, \psi_m\}$ and of $\mathbf{U}_h(\tilde{\mathbf{u}}_{1h})$ by $\{\varphi_1, \ldots, \varphi_n\}^d$ we can represent an element $q_h \in Q_h$ by

$$q_h = (\psi_1, \ldots, \psi_m) \cdot \underline{q}_h, \quad \text{with } \underline{q}_h \in \underline{Q}_h := \mathbb{R}^m$$

(where we use the notation $(\psi_1, \ldots, \psi_m) \cdot \underline{q}_h := \sum_{i=1}^m (\underline{q}_h)_i \cdot \psi_i$), and analogously the components of an element $\mathbf{v}_h = \left(v_{h_1}^T, \ldots, v_{h_d}^T\right)^T \in \mathbf{U}_h(\tilde{\mathbf{u}}_{1h})$ by

$$v_{hi} = (\varphi_1, \ldots, \varphi_n) \cdot \underline{v}_{hi}, \quad \text{with } \underline{\mathbf{v}}_h = \begin{pmatrix} \underline{v_{h_1}} \\ \vdots \\ \underline{v_{h_d}} \end{pmatrix} \in \underline{\mathbf{U}}_h := (\mathbb{R}^n)^d.$$

Then we can write (2.15) in matrix form

$$\begin{pmatrix} A(\underline{\mathbf{u}}_h) & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_h \\ \underline{p}_h \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{f}}_h \\ \underline{g}_h \end{pmatrix}. \tag{2.18}$$

Here $A(\underline{\mathbf{u}}_h)$ is defined as

$$A(\underline{\mathbf{u}}_h) = \begin{pmatrix} A(\underline{\mathbf{u}}_h)^{1,1} & A(\underline{\mathbf{u}}_h)^{1,2} \\ A(\underline{\mathbf{u}}_h)^{2,1} & A(\underline{\mathbf{u}}_h)^{2,2} \end{pmatrix}$$

in 2D resp.

$$A(\underline{\mathbf{u}}_h) = \begin{pmatrix} A(\underline{\mathbf{u}}_h)^{1,1} & A(\underline{\mathbf{u}}_h)^{1,2} & A(\underline{\mathbf{u}}_h)^{1,3} \\ A(\underline{\mathbf{u}}_h)^{2,1} & A(\underline{\mathbf{u}}_h)^{2,2} & A(\underline{\mathbf{u}}_h)^{2,3} \\ A(\underline{\mathbf{u}}_h)^{3,1} & A(\underline{\mathbf{u}}_h)^{3,2} & A(\underline{\mathbf{u}}_h)^{3,3} \end{pmatrix}$$

in 3D, with

$$A(\underline{\mathbf{u}}_h)^{r,s} = (\bar{a}(\underline{\mathbf{u}}_h; \varphi_j \cdot \mathbf{e}_r, \varphi_k \cdot \mathbf{e}_s))_{j,k},$$

where $\mathbf{e}_r$ is the r-th unity vector in $\mathbb{R}^d$. For $\bar{a}(.;.,.)$ as defined above we get

$$A(\underline{\mathbf{u}}_h)^{r,s} \equiv 0 \quad \text{if } r \neq s.$$

Analogously B is defined by the relation

$$B = \begin{pmatrix} B^1 & B^2 \end{pmatrix}$$

in 2D resp.

$$B = \begin{pmatrix} B^1 & B^2 & B^3 \end{pmatrix}$$

in 3D, with

$$B^r = b(\varphi_j \cdot \mathbf{e}_r, \psi_k))_{j,k},$$

and C by

$$C = (c(\psi_j, \psi_k))_{j,k}.$$

In the same manner we define the *mass matrix*

$$M = ((\varphi_j, \varphi_k)_0)_{j,k},$$

the *pressure mass matrix*

$$M_p = ((\psi_j, \psi_k)_0)_{j,k},$$

and the *Laplacian*

$$A_D = (a_D(\varphi_j, \varphi_k))_{j,k},$$

which we will need later in this thesis.

We denote the *FE-isomorphisms* between the discrete spaces and the spaces of coefficient vectors by $\phi_U : (\mathbb{R}^n)^d \to \mathbf{U}_h(\tilde{\mathbf{u}}_{1h})$ and $\phi_Q : \mathbb{R}^m \to Q_h$. The underline notation is used to indicate their inverses, i.e.

$$\phi_U \underline{\mathbf{v}}_h = \mathbf{v}_h, \qquad \underline{\phi_U \underline{\mathbf{v}}_h} = \underline{\mathbf{v}}_h, \tag{2.19}$$

$$\phi_Q \underline{q}_h = q_h, \qquad \underline{\phi_Q \underline{q}_h} = \underline{q}_h. \tag{2.20}$$

If it is clear from the context we omit the underlines and ϕ's and identify $\mathbf{v}_h \in \mathbf{U}_h(\tilde{\mathbf{u}}_{1h})$ and the associated $\underline{\mathbf{v}}_h \in (\mathbb{R}^n)^d$ and analogously q_h and $\underline{q}_h$.

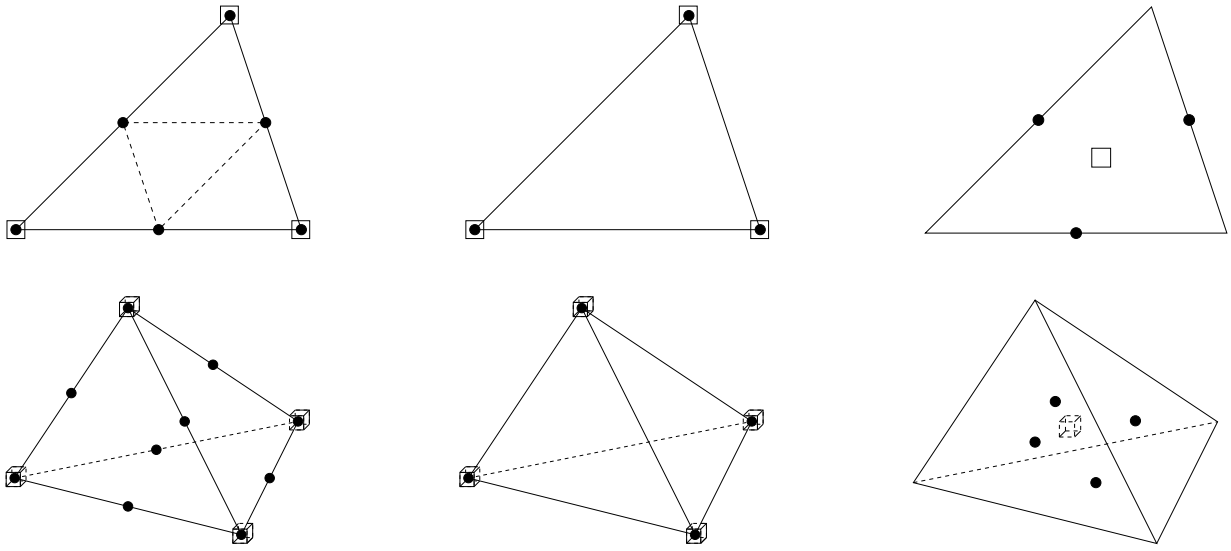
2.2.1 Examples of Mixed Elements

We present some popular choices of finite element pairs $\mathbf{U}_h \times Q_h$, in particular those we will use later for the construction of algebraic multigrid methods and in the numerical examples, all of them based on triangular resp. tetrahedral elements. Thus, we assume that some partitioning of $\mathcal{G}$ into triangles resp. tetrahedra $\mathcal{G} = \bigcup_i \tau_i$ is given, we denote the diameter of an element τ_i by h_{τ_i}, we assume that we can identify some typical diameter h (the discretization parameter) with

$$\underline{\alpha} h \leq h_{\tau_i} \leq \bar{\alpha} h \quad \text{for all } i,$$

where $\underline{\alpha}$ and $\bar{\alpha}$ are some positive constants, and we denote the set of elements by $\mathcal{T}_h = \{\tau_1, \tau_2, \ldots\}$. On each element τ_i we define the space $P_k(\tau_i)$ of polynomials of degree less than or equal k.

Figure 2.2 Some mixed finite elements for triangles (first row) and tetrahedra (second row). The circles/spheres indicate degrees of freedom for velocity-components, the boxes for pressure.



(a) Taylor-Hood (b) P_1-P_1 (c) Crouzeix-Raviart

2.2.1.1 (Modified) Taylor-Hood Element

For the Taylor-Hood element, or P_2-P_1 element, we specify

$$\begin{aligned} \mathbf{U}_h &= \{\mathbf{v}_h \in \mathbf{U} : \mathbf{v}_h|_{\tau_i} \in P_2(\tau_i)^d \text{ for all elements } \tau_i\}, \\ Q_h &= \{q_h \in Q : q_h|_{\tau_i} \in P_1(\tau_i) \text{ for all elements } \tau_i\}. \end{aligned} \tag{2.21}$$

An element $(\mathbf{v}_h, q_h)$ in $\mathbf{U}_h \times Q_h$ is uniquely determined by specifying the values of the d components of $\mathbf{v}_h$ on the nodes and on the midpoints of edges of the elements and the values of q_h on the nodes of the elements as illustrated in Figure 2.2(a).

The so called modified Taylor-Hood element, or $P_1\text{iso}P_2$-P_1, is a mixed element with the same degrees of freedom as the classical Taylor-Hood element, which is obtained the following way. We take Q_h as in (2.21), and then refine the mesh as indicated in the 2D-part of Figure 2.2(a): we divide each triangle into four subtriangles, each tetrahedron into eight subtetrahedra, and get the finer partitioning $\mathcal{G} = \bigcup_i \tilde{\tau}_i$. There we define the velocity space

$$\mathbf{U}_h = \{\mathbf{v}_h \in \mathbf{U} : \mathbf{v}_h|_{\tilde{\tau}_i} \in P_1(\tilde{\tau}_i)^d \text{ for all (sub-) elements } \tilde{\tau}_i\}.$$

Both the classical Taylor-Hood element and the modified one fulfill the discrete inf-sup condition as shown in [BF91]. Thus, their precision can be directly estimated using (2.17) and the well known approximation results for P_1 resp. P_2 elements. For the classical

element we get

$$\|\mathbf{u}-\mathbf{u}_h\|_1 + \|p-p_h\|_0 \leq Ch^t(|\mathbf{u}|_{t+1} + |p|_t), \quad \text{for } t=1 \text{ or } t=2,$$

if $(\mathbf{u},p) \in H^{t+1}(\mathcal{G})^d \times H^t(\mathcal{G})$. For the modified element, only the estimate with $t=1$ remains true.

2.2.1.2 Stabilized P_1-P_1 Element

If we use piecewise linear basis functions for both pressure and velocity components (Figure 2.2(b)) we obtain an element which is very easy to implement in a concrete computer program but unfortunately does not fulfill the discrete inf-sup condition. As mentioned in Remark 2.5, numerical solutions computed using this element often contain unphysical pressure modes which prevent convergence against the solution of the continuous problem. A possible way out is the introduction of the following stabilizing $c(.,.)$ term in (2.15)

$$c(p,q) = \alpha_S \sum_i h_{\tau_i}^2 (\nabla p, \nabla q)_{0,\tau_i}, \tag{2.22}$$

and a right hand side term to preserve consistency

$$\langle G, q\rangle = -\alpha_S \sum_i h_{\tau_i}^2 (\mathbf{f}, \nabla q)_{0,\tau_i}, \tag{2.23}$$

where α_S is a positive parameter (intensive discussion on the correct choice of this parameter and the local mesh size h_{τ_i} can be found for example in [Bec95] or [FM93]). We will refer to this stabilized element as P_1-P_1-stab.

Remark 2.6. *Another possibility of stabilizing the P_1-P_1 element leads to the so called MINI-element. Here the velocity space is extended by bubble functions, i.e. $\mathbf{u}_h$ is an element of $\tilde{\mathbf{U}}_h$ with*

$$\tilde{\mathbf{U}}_h := \{\mathbf{v} \in \mathbf{U} : \mathbf{v}|_{\tau_i} = \mathbf{w}|_{\tau_i} + b_{\tau_i}\bar{\alpha}_{\tau_i} \text{ with } \mathbf{w} \in \mathbf{U}_h \text{ and } \bar{\alpha}_{\tau_i} \in \mathbb{R}^d\},$$

where $b_{\tau_i}(\mathbf{x}) = \prod_j \lambda_j(\mathbf{x})$, and $\lambda_j(\mathbf{x})$ are the barycentric coordinates of $\mathbf{x}$ with respect to τ_i. It is possible to locally eliminate the bubble-variables, which leads to a similar problem as (2.15),(2.22),(2.23), with slightly more information on the choice of α_S, e.g. that it should be of order $\mathcal{O}(1/\nu)$.

Although this element does not fulfill the inf-sup condition, the following result holds (without proof)

Theorem 2.7. *[FS91, theorem 3.1] Suppose that the solution of the continuous Stokes problem satisfies $\mathbf{u} \in H^2(\mathcal{G})^d$ and $p \in H^2(\mathcal{G})$. Then for $\alpha_S > 0$ the problem (2.15),(2.22),(2.23) has a unique solution, satisfying*

$$\|\mathbf{u}-\mathbf{u}_h\|_1 + \|p-p_h\|_0 \leq C(h|\mathbf{u}|_2 + h^2|p|_2). \tag{2.24}$$

2.2.1.3 Crouzeix-Raviart Element

Here we drop the requirement that the finite dimensional spaces have to be subsets of the continuous spaces, insofar as the functions in $\mathbf{U}_h$ will not be continuous. We use nonconforming P_1 velocity components (P_1^{nc} for short), i.e. we define

$$\begin{aligned}\mathbf{U}_h = \{\mathbf{v}_h : \mathbf{v}_h|_{\tau_i} \in P_1(\tau_i)^d \text{ for all elements } \tau_i,\\ \mathbf{v}_h \text{ is continuous at the midpoints of all element-edges/faces (in 2D/3D)},\\ \mathbf{v}_h(b_j) = 0 \text{ for all midpoints of boundary edges/faces } b_j \text{ on } \Gamma_{1h}.\}\end{aligned} \tag{2.25}$$

and

$$Q_h = \{q_h \in Q : q_h|_{\tau_i} \in P_0(\tau_i) \text{ for all elements } \tau_i\}. \tag{2.26}$$

The elements in $\mathbf{U}_h \times Q_h$ are determined by their velocity values at the edge-/face-centers and pressure values at the element centers (see 2.2(c)).

Detailed analysis for this mixed element can be found in [CR73], for example the convergence result

$$\|\mathbf{u} - \mathbf{u}_h\|_1 \leq Ch(|\mathbf{u}|_2 + |p|_1)$$

for $(\mathbf{u}, p) \in H^2(\mathcal{G})^d \times H^1(\mathcal{G})$.

A nice property of the Crouzeix-Raviart element is the element-wise mass conservation, which is enforced by the piecewise constant pressure discretization.

Note that the term "Crouzeix-Raviart element", which we use for P_1^{nc}-P_0, is often associated to different elements, for example (scalar) P_1^{nc} or the divergence-free P_1^{nc} element (where a divergence-free basis for the velocities is constructed, and the pressure can therefore be eliminated from the equations).

In the following we will often drop the h subscripts if it is obvious from the context.

2.2.2 Multi-Element Meshes

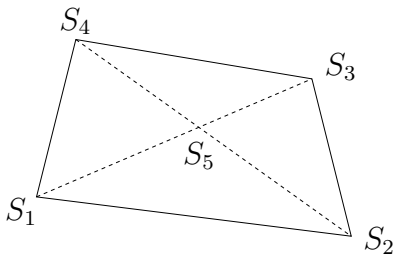
All the elements presented above are based on a mesh consisting of triangles resp. tetrahedra. We want to mention that they have counterparts for quadrilateral resp. hexahedral meshes, but we will not go into detail and refer to literature, especially [BF91] and [Tur99].

The following example describes the technique we use when we want to generate FEM matrices for more general meshes, namely *inner condensation*.

Example 2.8. Suppose we want to construct an element matrix based on a P_1 FE-discretization of a (scalar) equation for a quadrilateral $S_1S_2S_3S_4$ as in Figure 2.3. If we construct the midpoint S_5, then the discretization on the four resulting subtriangles would result in an element stiffness matrix

$$\begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & 0 & a_{25} \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & 0 & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix}.$$

Figure 2.3 A quadrilateral is discretized by decomposition into its sub-triangles.



As the midpoint has only connections to $S_1,\ldots,S_4$ — also in the fully assembled matrix — there would be a line like

$$a_{51}x_1 + a_{52}x_2 + a_{53}x_3 + a_{54}x_4 + a_{55}x_5 = f_5$$

in the system (where x is the solution vector, f the right hand side). Thus we can locally eliminate the entries for S_5 and get the resulting element matrix

$$\begin{pmatrix} a_{11} - a_{15}a_{51}/a_{55} & a_{12} - a_{15}a_{52}/a_{55} & -a_{15}a_{53}/a_{55} & a_{14} - a_{15}a_{54}/a_{55} \\ a_{21} - a_{25}a_{51}/a_{55} & a_{22} - a_{25}a_{52}/a_{55} & a_{23} - a_{25}a_{53}/a_{55} & -a_{25}a_{54}/a_{55} \\ -a_{35}a_{51}/a_{55} & a_{32} - a_{35}a_{52}/a_{55} & a_{33} - a_{35}a_{53}/a_{55} & a_{34} - a_{35}a_{54}/a_{55} \\ a_{41} - a_{45}a_{51}/a_{55} & -a_{45}a_{52}/a_{55} & a_{43} - a_{45}a_{53}/a_{55} & a_{44} - a_{45}a_{54}/a_{55} \end{pmatrix}$$

(and additional right hand side terms if $f \neq 0$).

This idea can be generalized to any cell-type (e.g. pentagons, pyramids, hexahedra, octahedra, or prisms). First, one has to split the cell into triangles resp. tetrahedra and then eliminate the auxiliary unknowns locally.

2.3 The Non-Stationary Problem

In the non-stationary case we use the method of lines for time integration. First, the weak formulation and the FEM approximation in the space variables (with time dependent coefficients) is performed as shown above to get the system

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{u}_h, \mathbf{v}_h)_0 + a_D(\mathbf{u}_h, \mathbf{v}_h) + a_C(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) &= \langle F, \mathbf{v}_h \rangle \,, \\ b(\mathbf{u}_h, q_h) &= 0 \end{aligned}$$

(plus initial conditions), a system of ordinary differential equations, where standard methods of time integration can be applied [HWN00].

To show two examples thereof, we assume that the k-th time step has length δ_k and that the right hand side is constant in time, and we search the discrete solution $(\mathbf{u}^k, p^k)$ at time $t_k = t_0 + \sum_{i=1}^{k} \delta_i$.

The first example is the *one-step-θ scheme*, which takes the matrix form

$$\begin{aligned}\left[\frac{1}{\delta_k}M+\theta A(\mathbf{u}^k)\right]\mathbf{u}^k+B^Tp^k &= \left[\frac{1}{\delta_k}M+(\theta-1)A(\mathbf{u}^{k-1})\right]\mathbf{u}^{k-1}+\mathbf{f},\\ B\mathbf{u}^k-Cp^k &= g.\end{aligned}$$

The parameter θ can be chosen in $[0,1]$, $\theta=0$ gives the *explicit Euler scheme*, $\theta=1$ the *implicit Euler scheme*, and $\theta=0.5$ the *Crank-Nicolson scheme* scheme.

As second example we present the *fractional-step-θ-scheme*, where each time step is divided into three substeps ($t_{k-1}\to t_{k-1+\theta}\to t_{k-\theta}\to t_k$):

$$\begin{aligned}
1.)\quad &\left[\frac{1}{\delta_k\theta}M+\alpha A(\mathbf{u}^{k-1+\theta})\right]\mathbf{u}^{k-1+\theta}+B^Tp^{k-1+\theta} = \left[\frac{1}{\delta_k\theta}M-\beta A(\mathbf{u}^{k-1})\right]\mathbf{u}^{k-1}+\mathbf{f},\\
&B\mathbf{u}^{k-1+\theta}-Cp^{k-1+\theta} = g,\\
2.)\quad &\left[\frac{1}{\delta_k\theta'}M+\beta A(\mathbf{u}^{k-\theta})\right]\mathbf{u}^{k-\theta}+B^Tp^{k-\theta} = \left[\frac{1}{\delta_k\theta'}M-\alpha A(\mathbf{u}^{k-1+\theta})\right]\mathbf{u}^{k-1+\theta}+\mathbf{f},\\
&B\mathbf{u}^{k-\theta}-Cp^{k-\theta} = g,\\
3.)\quad &\left[\frac{1}{\delta_k\theta}M+\alpha A(\mathbf{u}^{k})\right]\mathbf{u}^{k}+B^Tp^{k} = \left[\frac{1}{\delta_k\theta}M-\beta A(\mathbf{u}^{k-\theta})\right]\mathbf{u}^{k-\theta}+\mathbf{f},\\
&B\mathbf{u}^{k}-Cp^{k} = g,
\end{aligned}$$

with $\theta=1-\frac{\sqrt{2}}{2}$, $\theta'=1-2\theta$, $\alpha\in(\frac{1}{2},1]$ and $\beta=1-\alpha$ (where the choice $\alpha=\frac{1-2\theta}{1-\theta}$ is convenient for implementation, because then $\alpha\theta=\beta\theta'$).

In Table 2.1 we list convergence and stability properties of this schemes without going into the details and without giving any motivation for this properties (what can be found in [Ran00] or [HW02]). The terms used are described in the following definition.

Definition 2.9. Assume that the discrete solution (with constant time step length) of the test-problem

$$y'(t)=\lambda y(t),\quad y(0)=y_0,$$

with $\lambda\in\mathbb{C}$ has the form

$$y^k=R(\delta\lambda)y^{k-1},$$

where $R(z)$ is called the *stability function*. A scheme is said to be

- *A-stable* if $|R(z)|\le 1$ for all $z\in\mathbb{C}^-:=\{z\in\mathbb{C}:\operatorname{Re}z\le 0\}$,
- *strongly A-stable* if it is A-stable and $\lim_{z\to\infty}R(z)<1$, and
- *L-stable* if it is A-stable and $\lim_{z\to\infty}R(z)=0$.

Table 2.1 Stability and accuracy of some time stepping methods.

	A-stab.	str. A-stab.	L-stab.	accuracy
explicit Euler	no	no	no	1st order
implicit Euler	yes	yes	yes	1st order
Crank-Nicolson	yes	no	no	2nd order
frac.-step-θ	yes	yes	no	2nd order

2.4 The Convective Term

The convective term $(\mathbf{u}\nabla)\mathbf{u}$ resp. $(\mathbf{w}\nabla)\mathbf{u}$ causes two problems we have to deal with. Firstly instabilities may occur because of it, secondly we have to cope with its nonlinearity.

2.4.1 Instability

The unstable behavior can already be observed in the following 1D model problem.

Example 2.10. Assume that we want to solve the following scalar convection diffusion equation for u:

$$-\nu u''(x) + w u'(x) = f \quad \text{for } x \in (0,1),$$

$u(0) = u(1) = 0$, w, f and ν constant on $[0,1]$. A linear finite elements discretization on a regular grid with mesh-width h leads to the system

$$\underbrace{\begin{pmatrix} \frac{2\nu}{h} & -\frac{\nu}{h}+\frac{w}{2} & & \\ -\frac{\nu}{h}-\frac{w}{2} & \frac{2\nu}{h} & -\frac{\nu}{h}+\frac{w}{2} & \\ \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ & & -\frac{\nu}{h}-\frac{w}{2} & \frac{2\nu}{h} \end{pmatrix}}_{=:A} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} fh \\ fh \\ \vdots \\ \vdots \\ fh \end{pmatrix}.$$

The corresponding eigenvalue problem reads row-wise

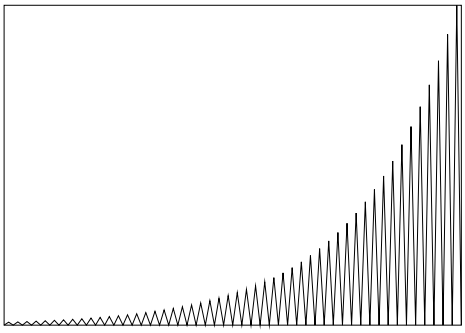
$$\left(\frac{w}{2} - \frac{\nu}{h}\right) u_{i+1} + \left(\frac{2\nu}{h} - \lambda\right) u_i + \left(-\frac{w}{2} - \frac{\nu}{h}\right) u_{i-1} = 0, \text{ for } i = 1, \ldots, n,$$
$$u_0 = u_{n+1} = 0,$$

where λ is the eigenvalue we are searching for. Assume that n is odd and $wh \neq 2\nu$, then one solution can easily be found as $\lambda = \frac{2\nu}{h}$, $u_{2k} = 0$, $u_{2k+1} = \left(\frac{wh+2\nu}{wh-2\nu}\right)^k$, for $k = 0, \ldots, \frac{n-1}{2}$.

Thus, for small ν this eigenvalue tends to zero and the very oscillatory eigenvector (Figure 2.4) is amplified in the solution if h is not small enough.

A solution of this problem is to use a less centered discretization, test-functions with more weight upstream than downstream. In the *Streamline Upwinding Petrov Galerkin*

Figure 2.4 Unphysical eigenmode caused by an unstable convection term.



(SUPG) scheme (details e.g. in [Pir89]) this is realized by applying the test-functions $\mathbf{v}_h + \beta_h(\mathbf{w}_h\nabla)\mathbf{v}_h$ instead of $\mathbf{v}_h$ for the momentum equation, where β_h is a parameter of magnitude $\mathcal{O}(h)$ and $\mathbf{w}_h$ is a "good guess" for the velocity $\mathbf{u}_h$, e.g. the solution at the previous time-step or the latest iterate of the nonlinear iteration (see the following Section 2.4.2). Amongst the terms introduced this way, only

$$\beta_h \left((\mathbf{w}_h \cdot \nabla)\mathbf{u}_h, (\mathbf{w}_h \cdot \nabla)\mathbf{v}_h\right) \tag{2.27}$$

is of importance for the increase of stability, thus we want to add it to $\bar{a}(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h)$. Just doing this solves the stability problems, but results in a loss of order of accuracy because the equation is no longer consistent. For example in the stationary case with a modified Taylor-Hood discretization we could repair this by adding appropriate terms to the momentum equation, i.e. by using

$$\begin{aligned}\nu(\nabla\mathbf{u}_h, \nabla\mathbf{v}_h) + ((\mathbf{w}_h \cdot \nabla)\mathbf{u}_h, \mathbf{v}_h) - (p_h, \operatorname{div}\mathbf{v}_h) \\ + \sum_{\tau_i} \beta_h \left[((\mathbf{w}_h \cdot \nabla)\mathbf{u}_h, (\mathbf{w}_h \cdot \nabla)\mathbf{v}_h)_{\tau_i} + (\nabla p_h, (\mathbf{w}_h \cdot \nabla)\mathbf{v}_h)_{\tau_i}\right] \\ = \langle F, \mathbf{v}_h\rangle + \sum_{\tau_i} \beta_h \left\langle F, (\mathbf{w}_h \cdot \nabla)\mathbf{v}_h\right\rangle_{\tau_i}. \end{aligned} \tag{2.28}$$

2.4.2 Nonlinearity

Because of its super-linear convergence Newton's method is a frequently used algorithm for solving nonlinear equations. As the (Gateaux-) derivative of the convective term calculates as

$$\left((\mathbf{u} \cdot \nabla)\mathbf{u}\right)' \mathbf{v} = (\mathbf{u} \cdot \nabla)\mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{u},$$

this would lead to equations of the following form (in the stationary case, ignoring for the moment the stabilizing terms introduced in the previous section)

$$\begin{aligned}(A_D + A_C(\mathbf{u}_k) + A_R(\mathbf{u}_k))(\mathbf{u}_{k+1} - \mathbf{u}_k) + B^T(p_{k+1} - p_k) &= \mathbf{d}_k, \\ B(\mathbf{u}_{k+1} - \mathbf{u}_k) \;\; - C(p_{k+1} - p_k) &= e_k,\end{aligned}$$

where $A_R(\mathbf{w})\mathbf{u}$ is the discretization of $(\mathbf{u}\cdot\nabla)\mathbf{w}$, $(\mathbf{u}_k, p_k)$ are the Newton iterates and $(\mathbf{d}_k, e_k)$ are some defect right hand sides.

Unfortunately the zero order reaction term A_R poses two problems. Firstly it adds block-off-diagonal entries to matrix A which increase its computational complexity, secondly it has an uncontrollable effect on the diagonal of A and could cause divergence. Thus it is common practice to drop this term. This leads to the fixed point method, where in each iteration step the Oseen equations have to be solved.

A third possibility would be to use (few steps of) an Oseen-preconditioned Richardson-iteration for the linear problem in each Newton-step, which avoids the reaction term in the system matrix but puts it to the right hand side.

In the case of strongly dominant convection and stationary equations, the nonlinear iteration is often hard to control. As this is less the case when solving the instationary problem, we introduce a pseudo time term, i.e. we obtain an iterative process where $\mathbf{u}_{k+1}$ and p_{k+1} satisfy

$$\begin{pmatrix} A(\mathbf{u}_k) + \alpha\bar{M} & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u}_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f} + \alpha\bar{M}\mathbf{u}_k \\ g \end{pmatrix},$$

where $\bar{M}$ is the mass matrix or (as we are not interested in the correct reconstruction of a non-stationary process) a lumped mass matrix and α a (small) parameter.

Besides the stabilization of the nonlinear process, this method has the nice property of increasing the symmetry of the linear systems.

Summing up, the resulting linear saddle point system which has to be solved (once or for every nonlinear iteration step and/or for every time step) has the general form (where we denote the block matrix with K, the solution block vector with $\mathbf{x}$, and the right hand side block vector with $\mathbf{b}$)

$$K\mathbf{x} := \begin{pmatrix} A(\mathbf{w}) & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ g \end{pmatrix} =: \mathbf{b}, \tag{2.29}$$

with

$$A(\mathbf{w}) = c_1 M + A_D + c_2 A_C(\mathbf{w}) + c_3 A_S(\mathbf{w}) + c_4 A_R(\mathbf{w}), \tag{2.30}$$

with mass matrix M, symmetric positive definite Laplacian A_D, non-symmetric convection A_C and reaction A_R, symmetric positive semi-definite convection stabilization A_S and constants c_1,...,c_4 which may be zero, and symmetric positive semi-definite (or zero) element stabilization C.

Because of (2.28) it may occur that we have no symmetry in the off-diagonal blocks, i.e.

$$\begin{pmatrix} A(\mathbf{w}) & B_1^T \\ B_2 & -C \end{pmatrix},$$

with $B_1 \neq B_2$. We will not deal with this situation separately in the remaining of this thesis, but assume the form (2.29). Note that the case $B_1 \neq B_2$ would not cause any additional problems, because then a dominating non-symmetry is already found in $A(\mathbf{w})$. Therefore we have to deal with a (substantial) non-symmetric system matrix anyway.

2.5 Iterative Solvers

In this section we give a brief overview over (non-multigrid) iterative solvers which are applicable to the saddle point system (2.29). Of course, there is a great variety of possible methods, we have only chosen some prominent examples.

2.5.1 Krylov Space Methods

A first possible family of solvers are those (preconditioned) Krylov space methods which are capable of solving indefinite and (in the non-Stokes case) non-symmetric problems. Examples thereof are GMRES and the BiCGstab. An overview of more Krylov space methods can be found for example in [Vos93] or [Meu99].

2.5.1.1 GMRES

The *generalized minimal residual method* (GMRES), introduced in [SS86], is a generalization of the MINRES method to the non-symmetric case. The idea is to solve in the k-th iteration step the least squares problem: Find $\mathbf{y} \in \mathbb{R}^k$ such that

$$\|\mathbf{b} - K(\mathbf{x}^0 + Q_k\mathbf{y})\|_{\ell_2} \to \min,$$

where the column vectors of Q_k build an orthonormal (w.r.t. the ℓ_2-scalar product) basis of the k-th Krylov space

$$\mathcal{K}_k(\mathbf{b}, K) = \text{span}\{\mathbf{b}, K\mathbf{b}, \ldots, K^{k-1}\mathbf{b}\}.$$

Thus, it could be seen as an exact method, which stops at the solution after finitely many steps, but which uses an increasing amount of memory in each step. Therefore in practice we use the GMRES(m) method, i.e. GMRES restarted periodically after m steps.

Algorithm 2.11. Preconditioned GMRES(m). Iterative Solution of $K\mathbf{x} = \mathbf{b}$, with preconditioner $\hat{K}$.

Choose starting solution $\mathbf{x}_0$;
$\mathbf{q}_1 = \hat{K}^{-1}(\mathbf{b} - K\mathbf{x}_0)$;
$z_1 = \|\mathbf{q}_1\|$;
$\mathbf{q}_1 = (1/z_1) \cdot \mathbf{q}_1$;
repeat
  begin
    for $k = 1$ **to** m **do**
      begin
        $\mathbf{q}_{k+1} = \hat{K}^{-1}K\mathbf{q}_k$;
        for $i = 1$ **to** k **do**
          begin
            $h_{ik} = \mathbf{q}_i \cdot \mathbf{q}_{k+1}$; $\mathbf{q}_{k+1} = \mathbf{q}_{k+1} - h_{ik}\mathbf{q}_i$;

end
$h_{k+1,k} = \|\mathbf{q}_{k+1}\|$; $\mathbf{q}_{k+1} = \mathbf{q}_{k+1}/h_{k+1,k}$;
end
for $k = 1$ **to** m **do**
begin
$cc = \sqrt{h_{kk}^2 + h_{k+1,k}^2}$;
$c = h_{kk}/cc$; $s = h_{k+1,k}/cc$; $h_{kk} = cc$;
for $i = k+1$ **to** m **do**
$\begin{pmatrix} h_{k,i} \\ h_{k+1,i} \end{pmatrix} = \begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} h_{k,i} \\ h_{k+1,i} \end{pmatrix}$;
$\begin{pmatrix} z_k \\ z_{k+1} \end{pmatrix} = \begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} z_k \\ 0 \end{pmatrix}$;
end
$y_m = z_m/h_{mm}$;
for $i = m$ **down-to** 1 **do**
$y_i = \left(z_i - \sum_{j=i+1}^{m} h_{ij} y_j\right) / h_{ii}$;
$\mathbf{x}_m = \mathbf{x}_0 + \sum_{i=1}^{m} y_i \mathbf{q}_i$;
$\mathbf{r}_m = \hat{K}^{-1}(\mathbf{b} - K\mathbf{x}_m)$;
$\mathbf{x}_0 = \mathbf{x}_m$; $\mathbf{r}_0 = \mathbf{r}_m$;
$z_1 = \|\mathbf{r}_0\|$; $\mathbf{q}_1 = (1/z_1) \cdot \mathbf{r}_0$;
end
until $|z_1| <$ tolerance

2.5.1.2 BiCGstab

The *stabilized bi-conjugate gradient method* (BiCGstab) was introduced in [VdV92] (with slight modifications in [SVdV94]). It is not optimal in each step, i.e. it solves the minimization problem only approximately, but as it uses a short range recurrence for the construction of the orthonormal basis of the Krylov space, it consumes considerably less computer memory as GMRES.

Algorithm 2.12. BiCGstab. Iterative Solution of $K\mathbf{x} = \mathbf{b}$, with preconditioner $\hat{K}$.

Choose starting solution $\mathbf{x}_0$;
$\mathbf{r}_0 = \hat{K}^{-1}(\mathbf{b} - K\mathbf{x}_0)$;
Choose arbitrary $\hat{\mathbf{r}}_0$, such that $\hat{\mathbf{r}}_0 \cdot \mathbf{r}_0 \neq 0$, e.g. $\hat{\mathbf{r}}_0 = \mathbf{r}_0$;
$\rho_0 = \alpha = \omega_0 = 1$;
$\mathbf{v}_0 = \mathbf{p}_0 = \mathbf{0}$;
$i \leftarrow 1$;
repeat
begin
$\rho_i = \hat{\mathbf{r}}_0 \cdot \mathbf{r}_{i-1}$; $\beta = (\rho_i/\rho_{i-1})(\alpha/\omega_{i-1})$;
$\mathbf{p}_i = \mathbf{r}_{i-1} + \beta(\mathbf{p}_{i-1} - \omega_{i-1}\mathbf{v}_{i-1})$;

$\mathbf{y} = K\mathbf{p}_i$; $\mathbf{v}_i = \hat{K}^{-1}\mathbf{y}$;
$\alpha = \rho_i/(\hat{\mathbf{r}}_0 \cdot \mathbf{v}_i)$;
$\mathbf{s} = \mathbf{r}_{i-1} - \alpha\mathbf{v}_i$;
$\mathbf{y} = K\mathbf{s}$; $\mathbf{t} = \hat{K}^{-1}\mathbf{y}$;
$\omega_i = (\mathbf{t} \cdot \mathbf{s})/(\mathbf{t} \cdot \mathbf{t})$;
$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha\mathbf{p}_i + \omega_i\mathbf{s}$;
if $\mathbf{x}_i$ is accurate enough **then quit**;
$\mathbf{r}_i = \hat{K}^{-1}(\mathbf{b} - K\mathbf{x}_i)$;
$i \leftarrow i + 1$;
end

2.5.2 SIMPLE

If linear solvers for scalar elliptic equations are available, a very popular method is *SIMPLE* (Semi-Implicit Method for Pressure-Linked Equations), developed by Patankar and Spalding [PS72, Pat80], which iteratively decouples the system to equations for pressure and for velocity (even for velocity-components in the Oseen or in the Stokes case as then A is block-diagonal).

We start with the factorization

$$K = \begin{pmatrix} A & 0 \\ B & S \end{pmatrix} \begin{pmatrix} I & A^{-1}B^T \\ 0 & -I \end{pmatrix}, \tag{2.31}$$

with the Schur complement $S = C + BA^{-1}B^T$, and then introduce preconditioners $\hat{A}$ for A in the first factor, $\hat{\hat{A}}$ for A in the second factor and $\hat{S}$ for S. Using this in a preconditioned Richardson method leads to the scheme

$$\hat{A}(\hat{\mathbf{u}}_{k+1} - \mathbf{u}_k) = \mathbf{f} - A\mathbf{u}_k - B^T p_k, \tag{2.32a}$$
$$\hat{S}(p_{k+1} - p_k) = B\hat{\mathbf{u}}_{k+1} - Cp_k - g, \tag{2.32b}$$
$$\hat{\hat{A}}(\mathbf{u}_{k+1} - \hat{\mathbf{u}}_{k+1}) = -B^T(p_{k+1} - p_k), \tag{2.32c}$$

where $\hat{\mathbf{u}}_{k+1}$ is some auxiliary vector. Now in (2.32a) $A\mathbf{u}_k$ is replaced by $\hat{A}\mathbf{u}_k$, leading to

$$\hat{A}\hat{\mathbf{u}}_{k+1} = \mathbf{f} - B^T p_k.$$

In the classical SIMPLE algorithm $\hat{S}$ is a preconditioner for the modified Schur complement $C + B\hat{\hat{A}}^{-1}B^T$, $\hat{\hat{A}}$ is the diagonal of A, denoted by D, and the pressure update is damped. This leads to the following algorithm.

Algorithm 2.13. SIMPLE
Choose preconditioners $\hat{A}$ for A and $\hat{S}$ for the modified Schur complement $C + BD^{-1}B^T$;

choose a starting solution $(\mathbf{u}_0, p_0)$; $k \leftarrow 0$;
repeat until convergence
begin
solve $\hat{A}\hat{\mathbf{u}} = \mathbf{f} - B^T p_k$;
solve $\hat{S}\tilde{p} = B\hat{\mathbf{u}} - Cp_k - g$;
choose a damping parameter γ;
$p_{k+1} = p_k + \gamma\tilde{p}$;
$\mathbf{u}_{k+1} = \hat{\mathbf{u}} - D^{-1}B^T\tilde{p}$;
$k \leftarrow k + 1$;
end

In all our numerical tests we use AMG methods for $\hat{A}$ and $\hat{S}$, therefore we will sometimes call the method *AMG-SIMPLE*.

Remark 2.14. *Algorithm 2.13 represents a simple version of this class of algorithms. One can find many variants in literature, examples are SIMPLER, SIMPLEC or SIMPLEV. Often the nonlinear iteration in the Navier-Stokes case is also embedded in the SIMPLE-scheme. For details see e.g. the references above, [GNR98], [AB01] and the references therein.*

2.5.3 Inexact Uzawa Methods

Like the SIMPLE algorithm the methods in this section decouple velocity and pressure equations iteratively. They are based on the following factorization of the inverse of the system matrix K,

$$K^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & -S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -BA^{-1} & I \end{pmatrix}. \tag{2.33}$$

An inexact inverse is built by replacing A^{-1} and S^{-1} in (2.33) by preconditioners $\hat{A}^{-1}$ and $\hat{S}^{-1}$:

$$\begin{pmatrix} \hat{A}^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & -\hat{S}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -B\hat{A}^{-1} & I \end{pmatrix}. \tag{2.34}$$

Now different combinations of these four factors are used to construct a preconditioner $\hat{K}^{-1}$.

Using all four factors — which would lead to

$$\hat{K} = \begin{pmatrix} \hat{A} & B^T \\ B & B\hat{A}^{-1}B^T - \hat{S} \end{pmatrix} \tag{2.35}$$

— for a preconditioned Richardson iteration we get the *inexact symmetric Uzawa algorithm*

$$\hat{A}(\hat{\mathbf{u}}_{k+1} - \mathbf{u}_k) = \mathbf{f} - A\mathbf{u}_k - B^T p_k, \tag{2.36a}$$
$$\hat{S}(p_{k+1} - p_k) = B\hat{\mathbf{u}}_{k+1} - Cp_k - g, \tag{2.36b}$$
$$\hat{A}(\mathbf{u}_{k+1} - \hat{\mathbf{u}}_{k+1}) = -B^T(p_{k+1} - p_k). \tag{2.36c}$$

Factors one, three and four and preconditioned Richardson result in the *inexact Uzawa algorithm*

$$\hat{A}(\mathbf{u}_{k+1} - \mathbf{u}_k) = \mathbf{f} - A\mathbf{u}_k - B^T p_k, \tag{2.37a}$$
$$\hat{S}(p_{k+1} - p_k) = B\mathbf{u}_{k+1} - Cp_k - g, \tag{2.37b}$$

which leads for $\hat{A} = A$, $\hat{S} = \sigma I$ to the *classical Uzawa algorithm*, for $\hat{A} = \alpha I$, $\hat{S} = \sigma I$ to the *classical Arrow-Hurwicz algorithm* [AHU58, BF91].

Details to these two methods can be found e.g. in [LQ87], [BWY90], or [Zul02] .

The combination of factors one, two and three

$$\hat{K}^{-1} = \begin{pmatrix} \hat{A}^{-1} & \hat{A}^{-1}B^T\hat{S}^{-1} \\ 0 & -\hat{S}^{-1} \end{pmatrix} \tag{2.38}$$

is studied e.g. in [SEKW01] and [MGW00], the use of the preconditioner

$$\hat{K}^{-1} = \begin{pmatrix} \hat{A}^{-1} & 0 \\ 0 & \hat{S}^{-1} \end{pmatrix} \tag{2.39}$$

in [SW94] and [IRT93].

The preconditioner (2.39) can also be motivated by the following observation for its exact version

$$\hat{K} = \begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix}.$$

We want to have an h-independent upper bound for the condition number of the preconditioned system

$$\mathrm{cond}_{\hat{K}}\left(\hat{K}^{-1}K\right),$$

where the condition number for a matrix Y with respect to a matrix norm $\|.\|_\alpha$ is defined as

$$\mathrm{cond}_\alpha(Y) := \|Y\|_\alpha \cdot \|Y^{-1}\|_\alpha.$$

For this purpose we need the following result.

Lemma 2.15. *Assume that A is positive definite, C positive semi-definite, and that if B does not possess full row-rank then C is positive definite. Then the generalized eigenvalues λ of*

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \lambda \begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}$$

are contained in the set

$$\left[-1;\, \frac{1-\sqrt{5}}{2}\right] \cup \left[1;\, \frac{1+\sqrt{5}}{2}\right].$$

Proof. To solve the problem

$$\lambda A\mathbf{u} = A\mathbf{u} + B^T p \tag{2.40a}$$

$$\lambda S p = B\mathbf{u} - Cp \tag{2.40b}$$

we distinguish two cases.

First if $\lambda = 1$ then (2.40a) can be fulfilled with $p = 0$ and (2.40a) by any $\mathbf{u}$ with $B\mathbf{u} = 0$.

Now for $\lambda \neq 1$ we get $\mathbf{u} = \frac{1}{\lambda - 1} A^{-1} B^T p$ from (2.40a), insert this in (2.40b) and get

$$\left(\lambda - \frac{1+\sqrt{5}}{2}\right)\left(\lambda - \frac{1-\sqrt{5}}{2}\right) BA^{-1}B^T p + (\lambda+1)(\lambda-1)Cp = 0. \tag{2.41}$$

As both $BA^{-1}B^T$ and C are positive semi-definite and at least one of them is positive definite, (2.41) can only be fulfilled for

$$\lambda \in \left[-1;\, \frac{1-\sqrt{5}}{2}\right] \cup \left(1;\, \frac{1+\sqrt{5}}{2}\right],$$

which completes the proof. □

Remark 2.16. *If $C \equiv 0$ then one can even show that $\lambda \in \{1, (1 \pm \sqrt{5}/2\}$. Similar and more general results can be found e.g. in [RW92], [IRT93], or [SW94].*

Now we can easily calculate that

$$\|\hat{K}^{-1}K\|_{\hat{K}} = \|K^{1/2}\hat{K}^{-1}K^{1/2}\|_{\ell_2}$$

and

$$\|K^{-1}\hat{K}\|_{\hat{K}} = \|\hat{K}^{1/2}K^{-1}\hat{K}^{1/2}\|_{\ell_2},$$

and deduce from Lemma 2.15 that

$$\|K^{1/2}\hat{K}^{-1}K^{1/2}\|_{\ell_2} \leq \frac{1+\sqrt{5}}{2},$$

and

$$\|\hat{K}^{1/2}K^{-1}\hat{K}^{1/2}\|_{\ell_2} \leq \frac{2}{\sqrt{5}-1}.$$

Thus we can estimate

$$\mathrm{cond}_{\hat{K}}\left(\hat{K}^{-1}K\right) \leq \frac{3+\sqrt{5}}{2},$$

where the upper bound is clearly independent of h.

2.5.3.1 "Black-Box"-Preconditioners

This class of preconditioners — it has been introduced in [SEKW01], [LW02] and [ESW02] for the Oseen problem and earlier for example in [SW94] for the Stokes problem — is based on two ideas.

First, one of the preconditioners (2.38), (2.39) is used for a preconditioned Krylov space method (Section 2.5.1).

Remark 2.17. *If we consider the stationary Stokes problem and the exact version of (2.38) (i.e. $\hat{A} = A$, $\hat{S} = S$) then the eigenvalues of $\hat{K}^{-1}K$ are contained in the set $\{-1, 1\}$, which would cause a Krylov space method to converge to the exact solution in two steps. We have already mentioned in Remark 2.16 that a similar result applies to the exact version of (2.39) if $C \equiv 0$, here the set is $\left\{1, (1 \pm \sqrt{5})/2\right\}$ and the solution is reached in three steps (see [MGW00, Ips01]).*

The second key point (in the Oseen case and for $C \equiv 0$, i.e. LBB-stable elements) is the following heuristic commutativity relation

$$\nabla(-\nu\Delta + (\mathbf{w} \cdot \nabla))_s \approx (-\nu\Delta + (\mathbf{w} \cdot \nabla))\nabla, \tag{2.42}$$

where the s-index of the convection diffusion operator indicates its scalar version. Thereof we can deduce

$$(-\nu\Delta + (\mathbf{w} \cdot \nabla))^{-1}\nabla \approx \nabla(-\nu\Delta + (\mathbf{w} \cdot \nabla))_s^{-1}.$$

Applying the divergence on both sides leads to

$$\mathrm{div}(-\nu\Delta + (\mathbf{w} \cdot \nabla))^{-1}\nabla \approx \Delta_s(-\nu\Delta + (\mathbf{w} \cdot \nabla))_s^{-1},$$

and inverting gives

$$\left[\mathrm{div}(-\nu\Delta + (\mathbf{w} \cdot \nabla))^{-1}\nabla\right]^{-1} \approx (-\nu\Delta + (\mathbf{w} \cdot \nabla))_s\Delta_s^{-1}, \tag{2.43}$$

i.e. the inverse of the Schur complement could be approximated by an inverse (pressure) Laplacian and a (pressure!) convection diffusion operator.

For the Stokes problem the scaled pressure mass matrix $\frac{1}{\nu}M_p$ is an optimal Schur-complement-preconditioner, i.e. γ_1 and γ_2 in

$$\gamma_1 \leq \frac{\nu q^T S q}{q^T M_p q} \leq \gamma_2 \quad \forall q \in \mathbb{R}^m$$

are h-independent, where the lower bound is valid because of the inf-sup-condition (2.16), the upper bound because of the continuity of $b(.,.)$

$$|b(\mathbf{v}, q)| \leq \sqrt{\gamma_2}\|\mathbf{v}\|_A\|q\|_{M_p}$$

(see also [LQ86, LQ87]). Therefore also in the case of Oseen equations it is suggested not to use just the two factors indicated by (2.43) but

$$\hat{S}^{-1} = \hat{M}_p^{-1}A(\mathbf{w})_s\hat{A}_{D_s}^{-1}, \tag{2.44}$$

where $\hat{M}_p$ is a preconditioner for the pressure mass matrix (e.g. the lumped mass matrix), $A(\mathbf{w})_s$ the scalar variant of $A(\mathbf{w})$ and $\hat{A}_{D_s}$ a preconditioner for the (pressure) Laplacian.

Remark 2.18. *Two possible problems of this preconditioner can arise of the assumption (2.42). First, the commutativity*

$$\nabla[(\mathbf{w} \cdot \nabla)] \approx (\mathbf{w} \cdot \nabla)\nabla,$$

is not fulfilled in general (except for some special situation, e.g. constant $\mathbf{w}$*), what poses problems if this term dominates (for small* ν*'s).*

Second, for non-constant ν *(e.g. due to a* k*-*ϵ *turbulence model, c.f. [MP94] or [RW99]) even the first part of (2.42) would be violated, as in this case*

$$\nabla\nu\Delta \neq \nu\Delta\nabla.$$

Chapter 3

Multigrid Methods

In the previous chapter we have introduced some iterative methods for the solution of saddle point systems, most of them having in common that (without preconditioning) they are not optimal, i.e. the number of arithmetical operations $\mathcal{Q}(\varepsilon)$ for a reduction of the residual by a factor ε is considerably larger than $\mathcal{O}(n)$, where n is the number of unknowns of the system.

Multigrid methods, which will be the main topic in the remaining of this thesis, possess this optimality-property $\mathcal{Q}(\varepsilon) = \mathcal{O}(n)$ (at least geometric multigrid methods), therefore we want to apply them as solvers (or preconditioners) for our system.

First we will describe a general algebraic multigrid (AMG) method, introduce the notation and pinpoint some differences to geometric multigrid (GMG) methods. Then we will give some concrete examples of methods for scalar elliptic equations.

3.1 A General Algebraic Multigrid Method

We want to construct a general AMG method for a set of linear equations

$$K_1 x = b_1,$$

where K_1 is a regular $n_1 \times n_1$ matrix. The index indicates the level, 1 is the finest level, L will be the coarsest. For AMG methods, which will be the main focus of this thesis, this numbering is natural, but note that it is the reverse of the natural numbering for GMG methods.

The first step in this method is to create a full rank *prolongation matrix* P_2^1 based on some coarsening (see later, Section 3.2.2), with $P_2^1 : \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$ and $n_2 < n_1$. For this purpose (almost) only information from some auxiliary matrix H_1 is used. Normally one uses the information from the matrix K_1, but the utilization of an auxiliary matrix (which is suggested for example in [Rei01]) enhances the flexibility of the method. In AMG methods the size of the (negative) matrix entries is related to the strength of the coupling of two unknowns, thus different notions of 'strength' can be introduced for different choices

of H_1. One could use e.g.

$$(H_1)_{i,j} = \begin{cases} -1/\|e_{i,j}\| & \text{if } i \neq j \text{ and vertex } i \text{ and } j \text{ are connected,} \\ \sum_{k \neq i} 1/\|e_{i,k}\| & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

where $\|e_{i,j}\|$ is the length of the edge connecting the nodes i and j, to represent a virtual FE-mesh. For convection diffusion equations, this could be modified for regions with dominating convection, which causes a faster "transport of information". More choices seem conceivable, but will not be dealt with in this thesis.

We also need a *restriction matrix* $R_1^2 : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$, for which we use $R_1^2 = (P_2^1)^T$. Now we can build the Galerkin projected matrix

$$K_2 = R_1^2 K_1 P_2^1,$$

and the auxiliary matrix on this level

$$H_2 = R_1^2 H_1 P_2^1.$$

Repeating this step we end up with a set of prolongation matrices P_{l+1}^l, $l = 1, \ldots, L-1$, where $P_{l+1}^l : \mathbb{R}^{n_{l+1}} \to \mathbb{R}^{n_l}$, $n_1 > n_2 > \ldots > n_L$, a set of restriction matrices R_l^{l+1}, and a set of coarse level matrices K_l and auxiliary matrices H_l with

$$K_{l+1} = R_l^{l+1} K_l P_{l+1}^l \tag{3.2}$$

and

$$H_{l+1} = R_l^{l+1} H_l P_{l+1}^l.$$

Completing the AMG method we need on each level $l = 1, \ldots, L-1$ an iterative method for the problem $K_l x_l = b_l$,

$$x_l^{j+1} = \mathcal{S}_l(x_l^j, b_l),$$

the *smoothing operator*.

Algorithm 3.1. Basic multigrid iteration for the system $K_l x_l = b_l$.
Let m_{pre} be the number of presmoothing steps, m_{post} of postsmoothing steps. Suppose we have chosen a starting solution x_l^0 on level l.

for $k \leftarrow 1$ **to** m_{pre} **do** $x_l^k = \mathcal{S}_l(x_l^{k-1}, b_l)$; (presmoothing)
$b_{l+1} \leftarrow R_l^{l+1}(b_l - K_l x_l^{m_{\text{pre}}})$; (restriction)
if $l + 1 = L$
  Compute the exact solution $\bar{x}_L$ of $K_L \bar{x}_L = b_L$;
else
  begin
    Apply Algorithm 3.1 (μ times) on
      $K_{l+1} x_{l+1} = b_{l+1}$
      (with starting solution $x_{l+1}^0 = 0$)
      and get $\bar{x}_{l+1}$;
  end
$x_l^{m_{\text{pre}}+1} \leftarrow x_l^{m_{\text{pre}}} + P_{l+1}^l \bar{x}_{l+1}$; (prolongation and correction)
for $k \leftarrow 1$ **to** m_{post} **do** $x_l^{m_{\text{pre}}+k+1} = \mathcal{S}_l(x_l^{\text{pre}+k}, b_l)$; (postsmoothing)
return $\bar{x}_l \leftarrow x_l^{m_{\text{pre}}+m_{\text{post}}+1}$;

The part from (restriction) to (prolongation and correction) will be referred to as "*coarse grid correction*".

Repeated application of this algorithm until fulfillment of some convergence criterion yields a basic AMG method. For $\mu = 1$ the iteration is called a 'V-cycle', for $\mu = 2$ 'W-cycle'. We use the abbreviations V-m_{pre}-m_{post} resp. W-m_{pre}-m_{post} for a V- resp. W-cycle with m_{pre} presmoothing and m_{post} postsmoothing steps.

Geometric Multigrid. The base for GMG methods is a hierarchical sequence of finer and finer meshes. Each level has an associated grid, thus P_{l+1}^l and R_l^{l+1} can be constructed using geometric information of two consecutive meshes, the auxiliary matrices H_l are not needed.

The coarse system matrices need not be built using the Galerkin approach (3.2), direct discretization of the differential operator on the specific mesh can be performed. For non-nested FE spaces (e.g. velocity components of the Crouzeix-Raviart element in Section 2.2.1.3) these two approaches differ, the direct discretization seems to be more natural.

3.1.1 Basic Convergence Analysis

The common denominator and key point of all multigrid methods is the splitting of the error components in two classes. One that can be reproduced on coarser levels/grids and therefore can be reduced by the coarse grid correction and one that has to be reduced by the smoother. For the geometric multigrid method the first group consists typically of low frequency parts the second of high frequency parts of the error. The ability to cope with

the first group is called *approximation property*, with the second *smoothing property*. The (optimal) convergence of the multigrid method is the consequence of their combination.

The Geometric Multigrid Case. Two classical techniques of proofing the convergence of geometric multigrid methods assure two-grid convergence (which can be shown to imply W-cycle convergence) by different splittings of the two-grid iteration operator (without postsmoothing)

$$\mathcal{M}_l^{l+1} := (I - P_{l+1}^i K_{l+1}^{-1} R_l^{l+1} K_l)\mathcal{S}_l^m,$$

where m is the number of smoothing steps (i.e. $m = m_{\text{pre}}$ in Algorithm 3.1) and $\mathcal{S}_l$ the iteration matrix of the smoother (for a preconditioned Richardson iteration with preconditioner $\hat{K}_l$ we have $\mathcal{S}_l = I - \hat{K}_l^{-1} K_l$).

The first technique, which was mainly developed by the Russian school [Bac66, Ast71, Kor77, Lan82] is based on a sum splitting. Here the projections P^{low} and P^{high} (which project on the subspaces spanned by the low and high frequency eigenvectors of the system matrix) are introduced, and the identity, decomposed into $I = P^{\text{low}} \oplus P^{\text{high}}$, is inserted into $\mathcal{M}_l^{l+1}$ (left of $\mathcal{S}_l^m$) to get the estimate

$$\begin{aligned}\|\mathcal{M}_l^{l+1}\| \leq\ & \|(I - P_{l+1}^i K_{l+1}^{-1} R_l^{l+1} K_l) P^{\text{low}}\| \|\mathcal{S}_l\|^m \\ & + \|(I - P_{l+1}^i K_{l+1}^{-1} R_l^{l+1} K_l)\| \|P^{\text{high}} \mathcal{S}_l^m\|.\end{aligned}$$

Two-grid convergence is then proven by showing that the term $\|(I - P_{l+1}^i K_{l+1}^{-1} R_l^{l+1} K_l) P^{\text{low}}\|$ is small (the *approximation property*) and that $\|P^{\text{high}} \mathcal{S}_l^m\|$ is arbitrary small for sufficiently many smoothing steps m (the *smoothing property*).

The other classical technique can be found in Hackbusch [Hac85]. Here a product splitting is constructed after inserting the identity $I = K_l^{-1} K_l$ into $\mathcal{M}_l^{l+1}$ (left of $\mathcal{S}_l^m$). Then again two properties have to be shown. One is the *approximation property* which here reads as

$$\|K_l^{-1} - P_{l+1}^l K_{l+1}^{-1} R_l^{l+1}\|_{\ell_2} \leq c / \|K_l\|_{\ell_2} \tag{3.3}$$

the other the *smoothing property*

$$\|K_l \mathcal{S}_l^m\|_{\ell_2} \leq \eta(m) \|K_l\|_{\ell_2} \tag{3.4}$$

where c is a positive constant and $\eta(m)$ the so called *smoothing rate* (independent of level l) with

$$\eta(m) \to 0 \quad \text{for} \quad m \to \infty.$$

They together imply the convergence of the two-grid method if m is large enough.

Remark 3.2. *Instead of (3.3) and (3.4) the approximation and smoothing properties are often formulated directly using the h-scaling induced by K_l, i.e.*

$$\|K_l^{-1} - P_{l+1}^l K_{l+1}^{-1} R_l^{l+1}\| \leq c h_l^{\delta}$$

and

$$\|K_l \mathcal{S}_l^m\| \leq \eta(m) h_l^{-\delta},$$

with appropriate δ and $\|.\|$.

The Algebraic Multigrid Case. Convergence analysis for algebraic multigrid methods has been mostly restricted to the symmetric positive definite case up to now. For a symmetric positive definite system matrix K_l the *smoothing property* (according to Ruge and Stüben [RS86]) takes the form

$$\|\mathcal{S}_l e\|_{K_l}^2 \leq \|e\|_{K_l}^2 - c_1 \|K_l e\|_{D_l^{-1}}^2 \quad \text{for all } e, \tag{3.5}$$

with $c_1 > 0$ independent of e and D_l the diagonal of K_l. This implies that the smoother has to reduce an error e strongly if $\|K_l e\|_{D_l^{-1}}$ is large (compared to $\|e\|_{K_l}$). If additionally the *approximation property*

$$\|(I - P_{l+1}^l K_{l+1}^{-1} R_l^{l+1} K_l) e\|_{K_l}^2 \leq c_2 \|K_l (I - P_{l+1}^l K_{l+1}^{-1} R_l^{l+1} K_l) e\|_{D_l^{-1}}^2 \tag{3.6}$$

is fulfilled with c_2 independent of e then

$$\begin{aligned} \|\mathcal{S}_l (I - P_{l+1}^i K_{l+1}^{-1} R_i^{l+1} K_l) e\|_{K_l}^2 \leq & \|(I - P_{l+1}^i K_{l+1}^{-1} R_l^{l+1} K_l) e\|_{K_l}^2 \\ & - c_1 \|K_l (I - P_{l+1}^l K_{l+1}^{-1} R_l^{l+1} K_l) e\|_{D_l^{-1}} \\ \leq & (1 - \frac{c_1}{c_2}) \|(I - P_{l+1}^l K_{l+1}^{-1} R_l^{l+1} K_l) e\|_{K_l}^2 \\ \leq & (1 - \frac{c_1}{c_2}) \|e\|_{K_l}^2, \end{aligned}$$

i.e. the two-grid algorithm with one postsmoothing step converges.

3.2 Examples in the Scalar Elliptic Case

Assume for this section, that the system we want to solve results from a FE approximation of the scalar elliptic model problem: Find $u : \mathcal{G} \to \mathbb{R}$ such that

$$\begin{aligned} \Delta u(\mathbf{x}) &= b(\mathbf{x}) \text{ for } \mathbf{x} \in \mathcal{G}, \\ u(\mathbf{x}) &= 0 \text{ for } \mathbf{x} \in \partial\mathcal{G}, \end{aligned}$$

which leads to the linear system

$$K_1 u = b_1.$$

3.2.1 Geometric Multigrid

Basic point wise iterations like the ω-Jacobi or the Gauss-Seidel method can be used as smoothers in the case of our model problem.

For nested FE spaces $X_1 \supset X_2 \supset \ldots$ (again the index indicates the level) the construction of the prolongation is straight-forward, it only has to reproduce identity.

In the case of non-nested spaces different strategies are needed. For the example of P_1^{nc} finite elements (e.g. the velocity components of the Crouzeix-Raviart element) one could use

$$\left(P_{l+1}^{l}\underline{u}_h\right)_e = \begin{cases} u_h(e) & \text{if } e \text{ is a fine grid edge inside a coarse grid element,} \\ \frac{1}{2}\left[u_h|_{\tau_1}(e) + u_h|_{\tau_2}(e)\right] & \text{if } e = \tau_1 \cap \tau_2 \text{ for two coarse grid elements } \tau_1,\, \tau_2 \end{cases} \tag{3.7}$$

(c.f. [BV90] or [Bre93]).

3.2.2 Algebraic Multigrid

We assume now that the discretization of our model problem is nodal based, i.e. each unknown is associated with a unique mesh node.

The smoother can again consist of ω-Jacobi or Gauss-Seidel iterations. For the construction of the coarse levels, i.e. the assembling of the prolongation matrices there are various possibilities, we will describe those which will be important later in this theses.

3.2.2.1 AMG Based on C/F-Splitting

The classical AMG methods use a splitting of the set of nodes into a set of coarse nodes (C) which will also be used on the coarse level, and a set of fine nodes (F) which 'live' only on the fine level, details can be found in [BMR84], [RS86] or [Stü01a].

Suppose that — after such a splitting has been chosen — the unknowns are sorted F-unknowns (living on F-nodes) first, then C-unknowns (living on C-nodes). This induces a block structuring of the linear system

$$K_l u = \begin{pmatrix} K_{FF}^l & K_{FC}^l \\ K_{CF}^l & K_{CC}^l \end{pmatrix} \begin{pmatrix} u_F \\ u_C \end{pmatrix} = \begin{pmatrix} b_F \\ b_C \end{pmatrix} = b$$

(and the same structuring for H_l and P_{l+1}^l). Now for the prolongation it is obviously a good choice to leave the C-unknowns unchanged, i.e. to use

$$P_{l+1}^{l} = \begin{pmatrix} P_C^F \\ I \end{pmatrix} \tag{3.8}$$

(omitting the level index l in P_C^F), where again there are many variants for P_C^F, some of them will be described in what follows. All of them have in common that each coarse node prolongates only to a very restricted set of fine nodes to prevent fill-in in the coarse level matrices and a resulting explosion of complexity.

One possibility is to do averaging on the F nodes, i.e. we could define

$$(P_C^F)_{j,k} = \begin{cases} \frac{1}{m_j} & \text{if } k \text{ is a neighboring C node of a F node } j, \\ 0 & \text{otherwise,} \end{cases} \tag{3.9}$$

where m_j is the number of neighboring C nodes of the F node j, and the neighbor-relation is induced by non-zero entries in the auxiliary matrix H_l.

Remark 3.3. *If the fine level mesh was constructed by a hierarchical refinement of a coarse mesh, the C and F nodes are chosen accordingly, and the underlying discretization is the P_1 element, then this strategy reproduces the geometric multigrid method.*

A more sophisticated prolongation can be found in [Stü01a]. Before presenting it we need the concepts of M-matrices and of essentially positive type matrices.

Definition 3.4. A matrix $H = (h_{ij})$ is called *M-matrix* if

1. $h_{ii} > 0$ for all i,
2. $h_{ij} \leq 0$ for all $i \neq j$,
3. H is regular, and $H^{-1} \geq 0$ (where here '$\geq$' is meant component-wise).

(One can skip the first requirement because it is a consequence of 2. and 3., see e.g. [Hac93].)

Definition 3.5. A matrix $H = (h_{ij})$ is said to be of *essentially positive type* if it is positive definite and there exists a constant $\omega > 0$ such that for all e,

$$\sum_{i,j}(-h_{ij})(e_i - e_j)^2 \geq \omega \sum_{i,j}(-h_{ij}^-)(e_i - e_j)^2, \tag{3.10}$$

with

$$h_{ij}^- = \begin{cases} h_{ij} & \text{if } h_{ij} < 0, \\ 0 & \text{otherwise} \end{cases}$$

(and $h_{ij}^+ = h_{ij} - h_{ij}^-$).

Remark 3.6. *If H is a M-matrix than condition (3.10) is fulfilled with $\omega = 1$. The class of essentially positive type matrices was introduced to capture "almost M"-matrices with small positive off-diagonal entries which can be 'repaired' (see [Bra86]).*

Lemma 3.7. *If a matrix H is of essentially positive type with ω as in (3.10) then for all e*

$$\frac{2}{\omega} e^T D_H e \geq e^T H e, \tag{3.11}$$

where D_H is the diagonal of H.

Proof. It is easy to check that

$$\frac{2}{\omega} e^T D_H e \geq e^T H e - \frac{1}{2} \underbrace{\sum_j \sum_{k \neq j} h_{jk} \left[\left(\frac{2}{\omega} - 1 \right) e_j^2 + 2 e_j e_k + \left(\frac{2}{\omega} - 1 \right) e_k^2 \right]}_{=:\Lambda}$$

and that

$$\left(\frac{2}{\omega} - 2 \right) (e_j^2 + e_k^2) \leq \left(\frac{2}{\omega} - 1 \right) e_j^2 + 2 e_j e_k + \left(\frac{2}{\omega} - 1 \right) e_k^2 \leq \frac{2}{\omega} (e_j^2 + e_k^2).$$

Thus

$$\Lambda \leq \frac{2}{\omega} \sum_{\substack{j,k \\ j \neq k}} h_{jk}(e_j^2 + e_k^2) - 2 \sum_{\substack{j,k \\ j \neq k}} h_{jk}^-(e_j^2 + e_k^2). \tag{3.12}$$

With $e = (0, \ldots, 0, 1, 0, \ldots, 0)^T$, the i-th unit vector, we get from (3.10)

$$\sum_{\substack{j \\ j \neq i}} h_{ij} \leq \omega \sum_{\substack{j \\ j \neq i}} h_{ij}^-,$$

which together with (3.12) gives

$$\Lambda \leq 0$$

and therefore completes the proof. □

From now on we will write

$$A \geq B$$

for two matrices A and B if $A - B$ is positive semi-definite (or $A > B$ if it is positive definite), e.g. we can express (3.11) as $\frac{2}{\omega} D_H \geq H$.

We shortly sketch the construction of a reasonable P_C^F for an essentially positive type matrix $H_l = (h_{ij})_{ij}$ according to [Stü01a]. The construction is done in a way that for a coarse level vector e_C the interpolation $P_C^F e_C$ "fits smoothly" to e_C, i.e. that if we set

$$e = \begin{pmatrix} P_C^F \\ I \end{pmatrix} e_C$$

then

$$h_{ii} e_i + \sum_{j \in N_i} h_{ij} e_j \approx 0, \quad \text{for } i \in F, \tag{3.13}$$

where N_i is the set of neighboring F- and C-nodes of F-node i, i.e.

$$N_i := \{j : \ j \neq i, \ h_{ij} \neq 0\}$$

the *direct neighborhood*. We will denote the subset of N_i with negative matrix connections with N_i^-, and $P_i \subseteq C \cap N_i^-$ will be the set of interpolatory nodes, i.e. the set of C-nodes which prolongate to F-node i. If we assume that for smooth error e

$$\frac{1}{\sum_{j \in P_i} h_{ij}} \sum_{j \in P_i} h_{ij} e_j \approx \frac{1}{\sum_{j \in N_i} h_{ij}} \sum_{j \in N_i} h_{ij} e_j$$

we could approximate (3.13) by

$$h_{ii} e_i + \kappa_i \sum_{j \in P_i} h_{ij} e_j = 0,$$

with

$$\kappa_i = \frac{\sum_{j\in N_i} h_{ij}}{\sum_{j\in P_i} h_{ij}}.$$

But for practical reasons (details can be found in [Stü01a]) we add all positive entries to the diagonal, i.e. we use

$$\tilde{h}_{ii}e_i + \tilde{\kappa}_i \sum_{j\in P_i} h_{ij}^- e_j = 0,$$

with

$$\tilde{h}_{ii} = h_{ii} + \sum_{j\in N_i} h_{ij}^+$$

and

$$\tilde{\kappa}_i = \frac{\sum_{j\in N_i} h_{ij}^-}{\sum_{j\in P_i} h_{ij}^-}.$$

Thus we set

$$(P_C^F)_{j,l} = -\kappa_j h_{jl}^- / \tilde{h}_{jj}. \tag{3.14}$$

If P_{l+1}^l is of the form (3.8) a sufficient condition for the approximation property (3.6) is given by the following theorem (without proof).

Theorem 3.8. *[RS86] If for all* $e = \begin{pmatrix} e_F \\ e_C \end{pmatrix}$

$$\|e_F - P_C^F e_C\|_{D_l,F}^2 \le c\|e\|_{K_l}^2, \tag{3.15}$$

where c is independent of e, and D_l *denotes the diagonal of* K_l*, then (3.6) is satisfied (here* $\|.\|_{D_l,F}$ *denotes the 'F-part' of the norm).*

For the prolongation (3.14) with $H_l \equiv K_l$ we can find another sufficient condition, again without proof.

Theorem 3.9. *[Stü01a, Theorem A.4.5] Let* $K_l = (h_{j,k})_{j,k} > 0$ *be of essentially positive-type with* $\sum_k h_{jk} \ge 0$ *for all k. With fixed* $\sigma \ge 1$*, select a C/F-splitting such that, for each* $j \in F$*, there is a set* $P_j \subseteq C \cap N_j^-$ *satisfying*

$$\sum_{k\in P_j} |h_{jk}^-| \ge \frac{1}{\sigma} \sum_{k\in N_j} |h_{jk}^-|. \tag{3.16}$$

Then the interpolation (3.14) satisfies for all e

$$\|e_F - P_C^F e_C\|_{D_l,F}^2 \le \frac{\sigma}{\omega}\|e\|_{K_l}^2, \tag{3.17}$$

with ω *from (3.10).*

Choice of C nodes. What still has to be fixed is a concrete C/F-splitting. A very easy to implement algorithm for this purpose is the *red-black-coloring method* [Kic98].

Algorithm 3.10. Red-Black Coloring

repeat until all the nodes are colored
begin
step 1: choose an uncolored node (e.g. with minimal node number);
step 2: this node is colored black;
step 3: all uncolored neighbors are colored red;
end

The black nodes are then used as C nodes.

A first variation of this algorithm is to use a different notion of 'neighboring' in step 3, to color only the *strongly negatively coupled* (snc) nodes, where a node j is said to be snc to a node k if

$$-h_{jk} \geq \varepsilon_{\text{str}} \max_i |h_{ji}^-|, \tag{3.18}$$

with fixed parameter $\varepsilon_{\text{str}} \in (0, 1]$ (typically $\varepsilon_{\text{str}} = 0.25$). We denote the set of strongly negative couplings of a node j by

$$S_j = \{k \in N_j : \ j \text{ is snc to } k\} \tag{3.19}$$

and the set of transposed strongly negative couplings by

$$S_j^T = \{k : \ j \in S_k\}. \tag{3.20}$$

Now step 3 can be replaced by

"all uncolored nodes which are snc to the black node are colored red". (3.21)

Another variant concerns step 1. The order in which the C nodes are chosen may be crucial if we want to obtain a uniform distribution of C and F nodes. One suggestion in this direction in [RS86] is to introduce a "measure of importance" λ_j for each node j in the set of 'undecided' nodes U, and to choose a node with maximal λ_j as next C node. One possibility for this measure is

$$\lambda_j = |S_j^T \cap U| + 2|S_j^T \cap F|, \tag{3.22}$$

which can be evaluated for all nodes in a preprocessing step and updated locally after each iteration step.

3.2.2.2 Element Agglomerating AMGe

The algorithms in the previous section were based on heuristics for M-matrices (or 'almost' M-matrices, like the class of essentially positive type matrices). AMGe (AMG using element stiffness matrices) was developed to obtain more general methods for FEM systems, where it takes advantage of the element matrices. In [BCF$^+$00] AMGe was introduced, the coarse grid construction, i.e. the C/F-splitting was adopted from standard AMG, the interpolation was built using the new technique.

An approach which combined the AMGe idea with a method yielding detailed topologic information on coarse levels (i.e. elements, faces, edges, nodes) was presented in [JV01], we will show the basic ideas and algorithms.

Assume that on one level (e.g. on the discretization level) we know the element-to-node connectivity, i.e. which nodes are part of a given element. Assume further that a method for the agglomeration of elements is known, satisfying the requirements that each element is part of one unique agglomerate and that each agglomerate is a connected set, meaning that for any two elements part of the the same agglomerate there exists a connected path of elements of this agglomerate connecting the two elements. Then we can apply the following algorithm for the creation of the coarse level topology.

Algorithm 3.11. [JV01] **AMGe coarse level topology**

1. Agglomerate the fine elements to coarse elements E_j (with the above properties).
2. Consider all intersections $E_j \cap E_k$ for all pairs of different agglomerated elements E_j and E_k. If such an intersection is maximal, i.e. is not contained in any other intersection, then it is called a face.
3. Consider the faces as sets of nodes. For each node n compute the intersection $\bigcap\{\text{all faces which contain } n\}$. Now the set of minimal, nonempty intersections defines the vertices.

We have formulated the algorithm for the 3D case, but it can be directly applied to 2D problems (then the 'faces' correspond to edges). If (in the 3D case) one additionally wants to construct edges, then this can be done in step 3 using the set of minimal, nonempty intersections which are not already vertices.

For the construction of the interpolation we first define the neighborhood of a (fine-level) node n by

$$\Omega(n) := \bigcup\{\text{all agglomerated elements that contain } n\}$$

and the minimal set

$$\Lambda(n) := \bigcap\{\text{all agglomerated elements that contain } n\}$$

($\Lambda(n)$ can be a node, edge, face or element). For the coarse-level nodes we use again identity prolongation (they are the C-nodes of standard AMG) and for the edges, faces,

cells we proceed recursively as follows. Assume for a set $\Lambda(n)$ that the interpolation on the unknowns in $\partial\Lambda(n)$ has been fixed,[1] and we want to calculate the interpolation on the nodes in $\Lambda(n) \setminus \partial\Lambda(n)$. For that we build the local stiffness matrix of $\Omega(n)$ (consisting of the element stiffness matrices of elements in $\Omega(n)$) with the underlying partitioning $(\Omega(n) \setminus \partial\Lambda(n)) \cup \partial\Lambda(n)$

$$K_{\Omega(n)} = \begin{pmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{pmatrix} \begin{matrix} \} & \Omega(n) \setminus \partial\Lambda(n) \\ \} & \partial\Lambda(n) \end{matrix}$$

(i stands for interior, b for boundary) and perform local energy minimization:

$$\text{find } u_i \text{ such that } (u_i^T u_b^T) \begin{pmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{pmatrix} \begin{pmatrix} u_i \\ u_b \end{pmatrix} \text{ is minimized, with given } u_b,$$

with the result (for symmetric, positive definite $K_{\Omega(n)}$)

$$u_i = -K_{ii}^{-1} K_{ib} u_b.$$

Now we can set

$$(P_C^F)_{j,k} = \left[-K_{ii}^{-1} K_{ib} \left(\tilde{P}_j \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{matrix} \leftarrow \text{ vertices of } \Lambda(j) \setminus \{k\} \\ \leftarrow k \end{matrix} \right)_{\partial\Lambda(j)} \right]_j,$$

where $\tilde{P}_j$ is the localized version of P_{l+1}^l.

Remark 3.12. *The drawback of this method is a possibly expensive set-up phase, as many local minimization problems have to be solved and one has to save all the element stiffness matrices. Approaches to overcome this difficulty can be found in [HV01].*

Remark 3.13. *A very nice property of element agglomerating AMGe is the complete information about grid topology on the coarse levels. This could be utilized in various ways, so e.g. stability analysis for saddle point problems can be performed nearly as in a geometric context (c.f. Section 4.1.3) or as another example one could use the information to construct some FAS-like schemes*[2] *for nonlinear problems, which is done in [JVW02].*

What we have not specified yet is how to construct the coarse agglomerates. One possibility for that is the following algorithm.

[1] The 'boundary' $\partial\Lambda(n)$ is defined straightforward: if $\Lambda(n)$ is a face then $\partial\Lambda(n)$ are those nodes of $\Lambda(n)$ which belong to more than one face, if $\Lambda(n)$ is an agglomerated element then $\partial\Lambda(n)$ is the union of faces of this element.

[2] FAS ... *full approximation storage*, a multigrid method which is capable of solving nonlinear problems, developed by Brandt [Bra77].

Algorithm 3.14. [JV01] **Jones-Vassilevski element agglomeration**
Assume we have a set of finer level elements $\{e_j\}$ and faces $\{f_j\}$, and introduce an integer weight $w(f_j)$ for each face f_j.

- **initiate.** Set $w(f) \leftarrow 0$ for all faces f;
- **global search.** Find a face f with maximal $w(f)$, if $w(f) = -1$ we are done; set $E \leftarrow \emptyset$;
  1. Set $E \leftarrow E \cup e_1 \cup e_2$, where $e_1 \cap e_2 = f$ and set $w_{\max} \leftarrow w(f)$, $w(f) \leftarrow -1$
  2. Increment $w(f_1) \leftarrow w(f_1) + 1$ for all faces f_1 such that $w(f_1) \neq -1$ and f_1 is a neighbor of f;
  3. Increment $w(f_2) \leftarrow w(f_2) + 1$ for all faces f_2 such that $w(f_2) \neq -1$, f_2 is a neighbor of f, and f_2 and f are faces of a common element;
  4. From the neighbors of f, choose a face g with maximal $w(g)$; if $w(g) \geq w_{\max}$ set $f \leftarrow g$ and go to step 1.;
  5. If all neighbors of f have smaller weight than $w_{\max}$, the agglomerated element E is complete; set $w(g) \leftarrow -1$ for all faces of the elements e contained in E; go to **global search**;

Remark 3.15. *In [JV01] also modifications to this algorithm are presented which allow some kind of semi-coarsening, i.e. coarsening with the focus in one specific direction (for example determined by convection).*

For the 2D case this algorithm mostly produces nice agglomerated elements, in the 3D case some strange shapes may occur, therefore some adjustments of the algorithm seem to be necessary.

A second method which produces good agglomerates, but often leads to a too strong coarsening (what has a disadvantageous influence on the h-independence) is the following.

Algorithm 3.16. Red-grey-black element agglomeration

repeat until all elements are colored
begin
choose an uncolored element, this is colored black;
color all uncolored or grey neighboring elements red
(where 'neighboring' could be induced by faces, edges or nodes);
color all uncolored elements neighbored to red elements grey;
end
the black elements plus surrounding red elements build the agglomerated elements;
each grey element is appended to the agglomerate where it "fits best"
(e.g. to the agglomerate it shares the largest face with);

Chapter 4

AMG Methods for the Mixed Problem

We now come back to the original problem, i.e. to solve the saddle point system (2.29)

$$\begin{pmatrix} A(\mathbf{w}) & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ g \end{pmatrix},$$

using algebraic multigrid methods.

The Segregated Approach. A first possibility is the *segregated approach*. One could use an outer solver which iteratively decouples the equations — for example SIMPLE (Section 2.5.2), Uzawa (Section 2.5.3), or the preconditioner in Section 2.5.3.1 combined with a Krylov space method (section 2.5.1) — and then use an AMG method for the resulting scalar elliptic problems. This direction is followed e.g. in [GNR98], [Stü01a] or [SEKW01] and related publications.

A very nice property of this approach is the simplicity of its realization in a concrete computer program. If components like scalar solvers are available they are easily assembled to a full solver for saddle point problems. One important subproblem here is the development of a fast and robust (multigrid-) solver for convection diffusion problems. We will not go into detail about that but refer to the literature, for example (without claiming completeness) [Reu95, BW97, OGWW98, YVB98, PTV00, DMS01, Reu02].

Unfortunately, sometimes (especially for Navier-Stokes resp. Oseen equations) the segregated approach has the drawback of a loss of optimality, i.e. an asymptotic complexity considerably larger than $\mathcal{O}(n)$ (n being the number of unknowns).

This is one of the reasons why we want to follow a different way. We will develop AMG methods for the coupled problem. As the techniques of the previous chapter mostly require positive definite systems, we have to adapt the ingredients to fit to our problem and develop some new methods.

4.1 Construction of the Coarse Level Systems

The first important part of our strategy is to avoid a mixture of velocity components and pressure on the coarse levels (which could occur if one just applies some "off-the-shelf" solver to the whole saddle point system), thus we choose a prolongation

$$P_{l+1}^l = \begin{pmatrix} \tilde{I}_{l+1}^l & \\ & J_{l+1}^l \end{pmatrix},$$

with

$$\tilde{I}_{l+1}^l = \begin{pmatrix} I_{l+1}^l & \\ & I_{l+1}^l \end{pmatrix} \text{ in 2D}$$

resp.

$$\tilde{I}_{l+1}^l = \begin{pmatrix} I_{l+1}^l & & \\ & I_{l+1}^l & \\ & & I_{l+1}^l \end{pmatrix} \text{ in 3D,}$$

where

$$I_{l+1}^l : \mathbb{R}^{n_{l+1}} \to \mathbb{R}^{n_l}$$

is the prolongation matrix for one velocity component,

$$J_{l+1}^l : \mathbb{R}^{m_{l+1}} \to \mathbb{R}^{m_l}$$

for pressure. We denote the corresponding restriction matrices by $\tilde{I}_l^{l+1}$, I_l^{l+1}, J_l^{l+1} and use

$$\tilde{I}_l^{l+1} = (\tilde{I}_{l+1}^l)^T, \qquad I_l^{l+1} = (I_{l+1}^l)^T, \qquad J_l^{l+1} = (J_{l+1}^l)^T.$$

The system matrix on level l is denoted by

$$\begin{pmatrix} A_l & B_l^T \\ B_l & -C_l \end{pmatrix}.$$

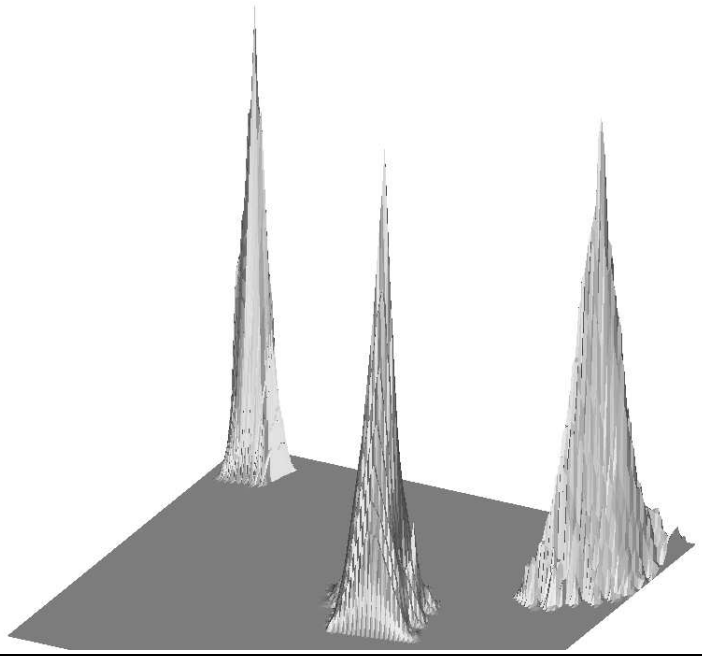
the spaces for velocity and pressure unknown vectors by $\underline{\mathbf{U}}_l := (\mathbb{R}^{n_l})^d$ and $\underline{Q}_l := \mathbb{R}^{m_l}$ and the coarse function spaces by

$$\mathbf{U}_l := \left\{ \mathbf{v} : \ \exists \underline{\mathbf{w}} \in \underline{\mathbf{U}}_l \text{ such that } \underline{\mathbf{v}} = \tilde{I}_2^1 \tilde{I}_3^2 \dots \tilde{I}_l^{l-1} \underline{\mathbf{w}} \right\},$$
$$Q_l := \left\{ p : \ \exists \underline{q} \in \underline{Q}_l \text{ such that } \underline{p} = J_2^1 J_3^2 \dots J_l^{l-1} \underline{q} \right\}.$$

Analogous to (2.19) we introduce the *FE-AMG-isomorphisms* $\phi_U^l : \underline{\mathbf{U}}_l \to \mathbf{U}_l$ and $\phi_Q^l : \underline{Q}_l \to Q_l$, and we will often identify elements of $\underline{\mathbf{U}}_l$ and $\mathbf{U}_l$, and $\underline{Q}_l$ and Q_l (see also Figure 4.1). In situations where it is not obvious which level the underline notation refers to, we will use the operators ${\phi_U^l}^{-1}$ and ${\phi_Q^l}^{-1}$ explicitly.

The system matrix on level l is constructed by Galerkin projection from the finer level $l-1$ as described in Section 3.1, we only have to take extra care of the stabilizing terms for unstable elements as in Section 2.2.1.2 or convection as in Section 2.4.1.

Figure 4.1 Using the FE-AMG-isomorphism we can associate coarse basis functions with basis vectors of $\mathbb{R}^{m_l}$. Here we have three basis functions for a certain Q_l.



Both are especially delicate in the multigrid setting, because if the modes illustrated in Figures 2.1 and 2.4 occur on a coarse level, then the smoother on the finer level might not damp them (they have lower frequency than the modes the smoother is intended to reduce) and the whole iteration might fail. As both terms have a non-standard h-dependence we try to reproduce this on coarser levels to avoid a 'flattening' of the stabilization. Numerical tests show that for the SUPG term a_S in (2.27) it is sufficient to do a simple scaling, i.e.

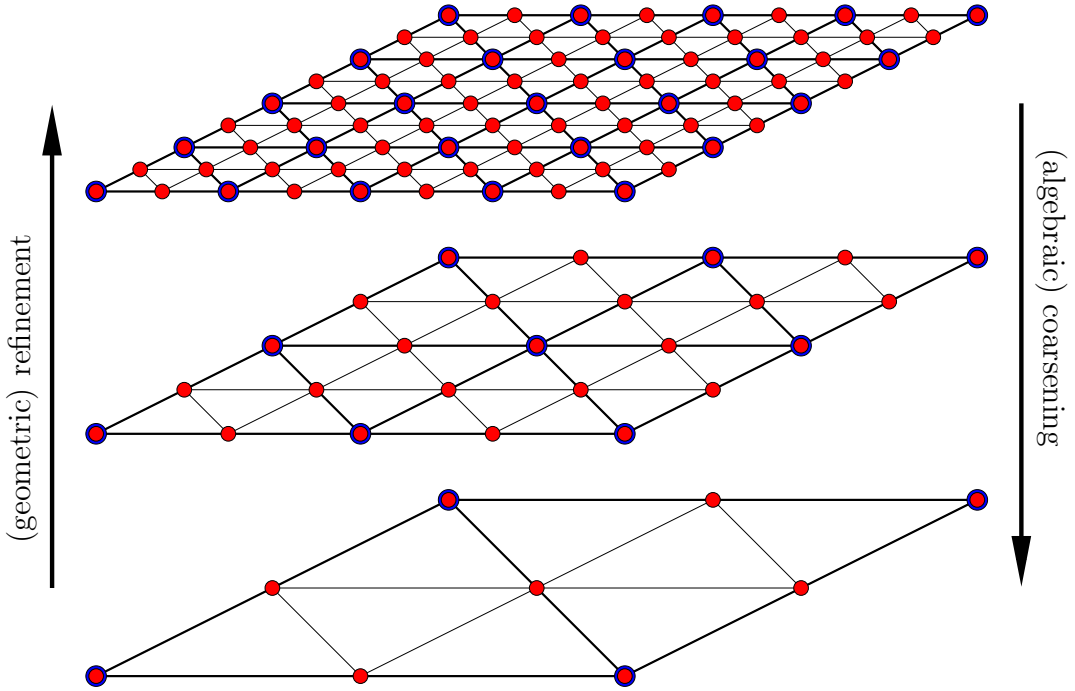
$$A_{S_{l+1}} = \sqrt[d]{\frac{n_l}{n_{l+1}}}\tilde{I}_l^{l+1} A_{S_i} \tilde{I}_{l+1}^l. \tag{4.1}$$

The scaling of the element stabilization will be dealt with later (Section 4.1.2).

Another major part of our strategy is to somehow project the relation of the velocity and pressure unknowns, which is indicated by the specific finite element, to the coarser levels. This makes it obvious that we will not construct a "black box" method, i.e. a method where the user just has to feed in the matrix, and the solution is found in optimal computation time. We try to exploit more information and hope that this will pay off.

We will now construct coarse level systems, which comply with this strategy, for the conforming linear elements of Section 2.2.1, namely the modified Taylor-Hood element $P_1\text{iso}P_2$-P_1, the P_1-P_1-stab element, and the Crouzeix-Raviart element P_1^{nc}-P_0.

Figure 4.2 Motivation for the construction of the coarse level hierarchy for the modified Taylor-Hood element. Red dots indicate velocity nodes, blue dots (partially behind the red dots) pressure nodes.



4.1.1 The P_1isoP_2-P_1 Hierarchy

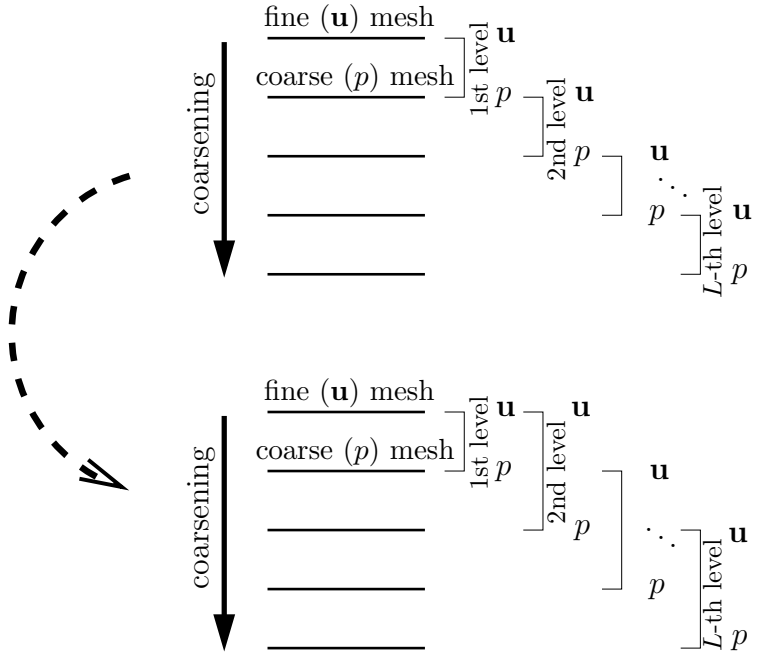
The motivation for the construction of the hierarchy is based on the GMG method for this element. If we look at Figure 4.2 we see that the velocity nodes on one level are exactly the pressure nodes of the next finer level. In the AMG case we use this observation for reversing the construction.

We start with the pressure unknowns on a given mesh. We take a method of Chapter 3 with auxiliary matrices as in (3.1) or equal to a (pressure) Laplacian, and construct a hierarchy with prolongation matrices J_{l+1}^l.

The given "velocity mesh" for the P_1isoP_2-P_1 element is the once refined (pressure) mesh. Thus, the first coarsening step for the velocities can be performed purely geometrically, the prolongation is simply the interpolation from one (pressure) grid to the once refined (velocity) grid. For the coarser part of the velocity hierarchy we then take the shifted pressure hierarchy, i.e. $I_{l+1}^l = J_l^{l-1}$ for $l \geq 2$.

Remark 4.1. *A discretization using the P_1isoP_2-P_1 element requires a refinement of a given mesh (for the velocities). If we want to avoid this (e.g. because of limitations of computer memory), then we could use the given mesh as "velocity mesh", construct the hierarchy based on the velocity nodes, and take the first coarsened level as first pressure 'mesh'.*

Figure 4.3 In the original approach, which is sketched in the upper part of the figure, the pressure nodes of one level are the velocity nodes of the coarser nodes. In the 2-shift strategy (lower part of the figure) there is a gap in between (except for the finest level).



Note that in this case the discretization and the solution process are no longer fully separated, the pressure unknowns have no direct interpretation in a finite element context, only their interpolation to the velocity mesh.

A problem which now turns up again is the fulfillment of the inf-sup condition. If the coarse level systems get unstable then this will influence the approximation property badly, and that they can get unstable can easily be seen by a (we admit pathological) example.

Take at one coarsening step $J_{l+1}^l = I$, the identity matrix. Then the velocity and pressure unknowns will collapse to an (unstabilized!) P_1-P_1 situation.

It is clear that this example is too extreme, nevertheless one observes (especially in the 3D case) problems in the numerical tests. It seems as if some methods for the construction of the hierarchy are less prone to these stability problems than others, but up to now we have found no general criterion, which could for example guarantee the inf-sup condition on the coarser levels.

A first (purely heuristical) way out is the following. The inf-sup condition requires roughly spoken, that there are enough velocity unknowns per pressure unknown to get a big enough quotient. We could satisfy this by a larger shift between velocity and pressure nodes, for example a 2-shift (illustrated in Figure 4.3), i.e. $I_{l+1}^l = J_{l-1}^{l-2}$.

A better way (with analytical background) was found for the P_1-P_1-stab element.

4.1.2 The P_1-P_1-stab Hierarchy

Here the unknowns for the velocity components and for the pressure unknowns 'live' on the same positions of the element, on the nodes. Thus it seems to be a good choice to take an arbitrary strategy from Chapter 3 for the construction of say I_{l+1}^l and use $J_{l+1}^l = I_{l+1}^l$ for the pressure.

With some weak assumptions on I_{l+1}^l it will be possible to show stability of the coarser levels, what has to be specified first is the construction of the coarse level stabilizing matrices C_l. For reasons which will become apparent in the proof of Lemma 4.3 we propose the following. Set

$$\tilde{C}_1 = C_1, \quad \tilde{C}_{l+1} = J_l^{l+1} \tilde{C}_l J_{l+1}^l, \text{ for } l \geq 1 \tag{4.2}$$

and

$$C_{l+1} = \frac{\lambda_{\max}(D_l^{-1} M_l)}{h^2} \tilde{C}_{l+1}, \text{ for } l \geq 1, \tag{4.3}$$

where h is the discretization parameter of the finest level, M_l the Galerkin projection of the mass matrix M_1 to level l, D_l the diagonal of one (component-) block of the Galerkin projection of the vector-Laplacian A_{D_l}, and $\lambda_{\max}(D_l^{-1} M_l)$ denotes the largest eigenvalue of $D_l^{-1} M_l$. For practical computation we will use very rough estimates for $\lambda_{\max}(D_l^{-1} M_l)$.

Now we will show stability of the coarse level systems in the form

$$\sup_{\substack{\mathbf{0} \neq \mathbf{v} \in \underline{\mathbf{U}}_l \\ 0 \neq q \in \underline{Q}_l}} \frac{\mathcal{B}_l(\mathbf{u}, p; \mathbf{v}, q)}{\|\mathbf{v}\|_{A_{Dl}} + \|q\|_{M_l}} \geq \zeta \left(\|\mathbf{u}\|_{A_{Dl}} + \|p\|_{M_l} \right) \quad \forall (\mathbf{u}, p) \in \underline{\mathbf{U}}_l \times \underline{Q}_l, \tag{4.4}$$

with a constant ζ and

$$\mathcal{B}_l(\mathbf{u}, p; \mathbf{v}, q) = \mathbf{u}^T A_{D_l} \mathbf{v} + p^T B_l \mathbf{v} + \mathbf{u}^T B_l^T q - p^T C_l q.$$

For this we use the ideas given in [FS91, FHS93] and just 'translate' them to our algebraic setting.

Remark 4.2. *Condition (4.4) is the inf-sup condition needed by the theorem of Babuška and Aziz [BA72], [Bra97, theorem 3.6]. It would be a consequence of the LBB condition needed by Theorem 2.2, the reverse does obviously not hold.*

The main point of the stability analysis will be the following lemma, which has been proven in [Ver84a] for the geometric case.

Lemma 4.3. *Assume that for all elements $\tau_j \in \mathcal{T}_h$ the diameter h_{τ_j} fulfills*

$$\underline{\alpha} h \leq h_{\tau_j} \leq \bar{\alpha} h, \tag{4.5}$$

with positive constants $\underline{\alpha}$ and $\bar{\alpha}$ and the discretization parameter h, and assume further that A_{D_l} is symmetric and of essentially positive type (see Definition 3.5) and that for all $\mathbf{v}_l \in \underline{\mathbf{U}}_l$ we can find $\underline{\Pi}_l^{l+1}\mathbf{v}_l \in \underline{\mathbf{U}}_{l+1}$ such that

$$\|\mathbf{v}_l - \tilde{I}_{l+1}^l \underline{\Pi}_l^{l+1}\mathbf{v}_l\|_{D_l}^2 \leq \beta_1 \|\mathbf{v}_l\|_{A_{D_l}}^2, \tag{4.6}$$

with some constant β_1.

Then for all levels $l \in \{1, \ldots, L\}$ there exist positive c_l and d_l such that

$$\sup_{\mathbf{0}\neq\mathbf{v}\in\underline{\mathbf{U}}_l} \frac{\mathbf{v}B_l^T p}{\|\mathbf{v}\|_{A_{D_l}}} \geq c_l \|p\|_{M_l} - d_l \left(p^T C_l p\right)^{\frac{1}{2}} \quad \forall p \in Q_l. \tag{4.7}$$

Proof. Since in the course of this proof we will have to distinguish between the elements of $\underline{\mathbf{U}}_l$, $\underline{Q}_l$ and their representatives in $\mathbf{U}_l$, Q_l we will again use the "underline-notation".

Obviously

$$\|\underline{\mathbf{x}}_l\|_{M_l}^2 \leq \lambda_{\max}(D_l^{-1}M_l)\|\underline{\mathbf{x}}_l\|_{D_l}^2$$

for any $\mathbf{x}_l$, hence

$$\|\underline{\mathbf{v}}_l - \tilde{I}_{l+1}^l \underline{\Pi}_l^{l+1}\underline{\mathbf{v}}_l\|_{M_l}^2 \leq \lambda_{\max}(D_l^{-1}M_l)\beta_1\|\underline{\mathbf{v}}_l\|_{A_{D_l}}^2,$$

therefore there exists $\Pi_l^{l+1}\mathbf{v}_l \in \mathbf{U}_{l+1}$, such that

$$\|\mathbf{v}_l - \Pi_l^{l+1}\mathbf{v}_l\|_0^2 \leq \lambda_{\max}(D_l^{-1}M_l)\beta_1\|\mathbf{v}_l\|_1^2. \tag{4.8}$$

Because A_{D_l} is essentially positive definite we know from Lemma 3.7 that

$$\frac{2}{\omega}\underline{\mathbf{x}}^T D_l \underline{\mathbf{x}} \geq \underline{\mathbf{x}}^T A_{D_l}\underline{\mathbf{x}}. \tag{4.9}$$

Because of

$$\begin{aligned}
\|\tilde{I}_{l+1}^l\underline{\Pi}_l^{l+1}\underline{\mathbf{v}}_l\|_{A_{D_l}} - \|\underline{\mathbf{v}}_l\|_{A_{D_l}} &\leq \|\tilde{I}_{l+1}^l\underline{\Pi}_l^{l+1}\underline{\mathbf{v}}_l - \underline{\mathbf{v}}_l\|_{A_{D_l}} \\
&\leq \sqrt{\frac{2}{\omega}}\|\tilde{I}_{l+1}^l\underline{\Pi}_l^{l+1}\underline{\mathbf{v}}_l - \underline{\mathbf{v}}_l\|_{D_l} \\
&\leq \sqrt{\frac{2\beta_1}{\omega}}\|\underline{\mathbf{v}}_l\|_{A_{D_l}}
\end{aligned}$$

we see that

$$\|\tilde{I}_{l+1}^l\underline{\Pi}_l^{l+1}\underline{\mathbf{v}}_l\|_{A_{D_l}} \leq \underbrace{\left(1 + \sqrt{\frac{2\beta_1}{\omega}}\right)}_{=:\beta_2}\|\underline{\mathbf{v}}_l\|_{A_{D_l}}. \tag{4.10}$$

We will now show by induction that on all levels l we can estimate

$$(\operatorname{div} \mathbf{w}_l, p_l) \geq c_l \|\mathbf{w}_l\|_1 \|p_l\|_0 - d_l \|\mathbf{w}_l\|_1 \left(\underline{p}_l^T C_l \underline{p}_l\right)^{1/2} \quad \text{for all } p_l \in Q_l. \tag{4.11}$$

For that we assume that (4.11) is valid on some level l (for $l = 1$ it is valid because there we are in the geometric case). Set $\mathbf{w}_{l+1} = \Pi_l^{l+1} \mathbf{w}_l$. Then

$$\begin{aligned}
(\operatorname{div} \mathbf{w}_{l+1}, p_{l+1}) &= (\operatorname{div}(\mathbf{w}_{l+1} - \mathbf{w}_l), p_{l+1}) + (\operatorname{div} \mathbf{w}_l, p_{l+1}) \\
&= (\mathbf{w}_l - \mathbf{w}_{l+1}, \nabla p_{l+1}) + (\operatorname{div} \mathbf{w}_l, p_{l+1}) \\
&\geq -\left(\sum_j h_{\tau_j}^{-2} \|\mathbf{w}_l - \mathbf{w}_{l+1}\|_{0,\tau_j}^2\right)^{1/2} \cdot \left(\sum_j h_{\tau_j}^2 \|\nabla p_{l+1}\|_{0,\tau_j}^2\right)^{1/2} \\
&\quad + (\operatorname{div} \mathbf{w}_l, p_{l+1}).
\end{aligned} \tag{4.12}$$

We will now derive estimates for the terms in the last inequality. Because of (4.5) and (4.8) we know that

$$\begin{aligned}
\sum_j h_{\tau_j}^{-2} \|\mathbf{w}_l - \mathbf{w}_{l+1}\|_{0,\tau_j}^2 &\leq (\underline{\alpha} h)^{-2} \|\mathbf{w}_l - \mathbf{w}_{l+1}\|_0^2 \\
&\leq \frac{\beta_1}{\underline{\alpha}^2} \frac{\lambda_{\max}(D_l^{-1} M_l)}{h^2} \|\mathbf{w}_l\|_1^2,
\end{aligned}$$

with the definition of C_1, $\tilde{C}_{l+1}$, and C_{l+1} we get

$$\sum_j h_{\tau_j}^2 \|\nabla p_{l+1}\|_{0,\tau_j}^2 = \underline{p}_{l+1}^T \tilde{C}_{l+1} \underline{p}_{l+1} = \frac{h^2}{\lambda_{\max}(D_l^{-1} M_l)} \underline{p}_{l+1}^T C_{l+1} \underline{p}_{l+1},$$

and because of (4.11) we can derive that

$$\begin{aligned}
(\operatorname{div} \mathbf{w}_l, p_{l+1}) &\geq c_l \|\mathbf{w}_l\|_1 \|p_{l+1}\|_0 - d_l \|\mathbf{w}_l\|_1 \left(\underline{p}_{l+1}^T J_l^{l+1} C_l J_{l+1}^l \underline{p}_{l+1}\right)^{1/2} \\
&= c_l \|\mathbf{w}_l\|_1 \|p_{l+1}\|_0 - d_l \sqrt{\frac{\lambda_{\max}(D_l^{-1} M_l)}{\lambda_{\max}(D_{l-1}^{-1} M_{l-1})}} \|\mathbf{w}_l\|_1 \left(\underline{p}_{l+1}^T C_{l+1} \underline{p}_{l+1}\right)^{1/2}.
\end{aligned}$$

Combining these results with (4.12) gives

$$\begin{aligned}
(\operatorname{div} \mathbf{w}_{l+1}, p_{l+1}) &\geq -\frac{\sqrt{\beta_1}}{\underline{\alpha}} \|\mathbf{w}_l\|_1 \left(\underline{p}_{l+1}^T C_{l+1} \underline{p}_{l+1}\right)^{1/2} + (\operatorname{div} \mathbf{w}_l, p_{l+1}) \\
&\geq -\left(\frac{\sqrt{\beta_1}}{\underline{\alpha}} + d_l \sqrt{\frac{\lambda_{\max}(D_l^{-1} M_l)}{\lambda_{\max}(D_{l-1}^{-1} M_{l-1})}}\right) \cdot \|\mathbf{w}_l\|_1 \left(\underline{p}_{l+1}^T C_{l+1} \underline{p}_{l+1}\right)^{1/2} \\
&\quad + c_l \|p_{l+1}\|_0 \|\mathbf{w}_l\|_1.
\end{aligned}$$

With (4.10) we get

$$\begin{aligned}\frac{(\operatorname{div}\mathbf{w}_{l+1},p_{l+1})}{\|\mathbf{w}_{l+1}\|_1} &\geq \beta_2^{-1}\frac{(\operatorname{div}\mathbf{w}_{l+1},p_{l+1})}{\|\mathbf{w}_l\|_1}\\ &\geq c_l/\beta_2\|p_{l+1}\|_0\\ &\quad-\left(\frac{\sqrt{\beta_1}}{\beta_2\underline{\alpha}}+\frac{d_l}{\beta_2}\sqrt{\frac{\lambda_{\max}(D_l^{-1}M_l)}{\lambda_{\max}(D_{l-1}^{-1}M_{l-1})}}\right)\cdot\left(\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}\right)^{1/2},\end{aligned}$$

hence, with

$$\begin{aligned}c_{l+1} &:= c_l/\beta_2 \quad\text{and}\\ d_{l+1} &:= \frac{\sqrt{\beta_1}}{\beta_2\underline{\alpha}}+\frac{d_l}{\beta_2}\sqrt{\frac{\lambda_{\max}(D_l^{-1}M_l)}{\lambda_{\max}(D_{l-1}^{-1}M_{l-1})}}\end{aligned}$$

we complete the proof. □

We are now ready to prove stability.

Theorem 4.4. *Suppose that the assumptions of Lemma 4.3 hold. Then*

$$\sup_{\substack{\mathbf{0}\neq\mathbf{v}\in\underline{\mathbf{U}}_l\\ 0\neq q\in\underline{Q}_l}}\frac{\mathcal{B}_l(\mathbf{u},p;\mathbf{v},q)}{\|\mathbf{v}\|_{A_{D_l}}+\|q\|_{M_l}}\geq\zeta\left(\|\mathbf{u}\|_{A_{D_l}}+\|p\|_{M_l}\right)\quad\forall(\mathbf{u},p)\in\underline{\mathbf{U}}_l\times\underline{Q}_l,\tag{4.13}$$

with some $\zeta>0$ (where ζ may depend on l).

Proof. Choose (for given $\mathbf{u}$ and p) $\mathbf{w}\in\underline{\mathbf{U}}_l$ such that the supremum in Lemma 4.3 is attained and that $\|\mathbf{w}\|_{A_{D_l}}=\|p\|_{M_l}$. Now

$$\begin{aligned}\mathcal{B}_l(\mathbf{u},p;\mathbf{w},0) &= \mathbf{u}^T A_{D_l}\mathbf{w}+p^T B_l\mathbf{w}\\ &\overset{\text{Lemma 4.3}}{\geq} -\|\mathbf{u}\|_{A_{D_l}}\|\mathbf{w}\|_{A_{D_l}}+c_l\|\mathbf{w}\|_{A_{D_l}}\|p\|_{M_l}-d_l\|\mathbf{w}\|_{A_{D_l}}\left(p^TC_lp\right)^{1/2}\\ &= -\|\mathbf{u}\|_{A_{D_l}}\|p\|_{M_l}+c_l\|p\|_{M_l}^2-d_l\|p\|_{M_l}\left(p^TC_lp\right)^{1/2}\\ &\overset{xy\leq\frac{x^2}{\varepsilon}+\varepsilon y^2}{\geq} -\frac{1}{2\varepsilon}\|\mathbf{u}\|_{A_{D_l}}^2-\frac{\varepsilon}{2}\|p\|_{M_l}^2+c_l\|p\|_{M_l}^2-\frac{d_l}{2\varepsilon}p^TC_lp-\frac{d_l\varepsilon}{2}\|p\|_{M_l}^2\\ &= -\theta_1\|\mathbf{u}\|_{A_{D_l}}^2+\theta_2\|p\|_{M_l}^2-\theta_3p^TC_lp,\end{aligned}$$

where c_l and d_l are the constants given by Lemma 4.3 and $\theta_1:=\frac{1}{2\epsilon}$, $\theta_2:=c_l-\frac{\epsilon}{2}(1+d_l)$, and $\theta_3:=\frac{d_l}{2\epsilon}$ are positive constants if we choose $0<\varepsilon<\frac{2c_l}{1+d_l}$.

We now take $(\mathbf{v},q)=(\mathbf{u}+\vartheta\mathbf{w},-p)$ with a parameter ϑ and get

$$\begin{aligned}\mathcal{B}_l(\mathbf{u},p;\mathbf{v},q) &= \mathcal{B}_l(\mathbf{u},p;\mathbf{u}+\vartheta\mathbf{w},-p)\\ &= \mathcal{B}_l(\mathbf{u},p;\mathbf{u},-p)+\vartheta\mathcal{B}_l(\mathbf{u},p;\mathbf{w},0)\\ &\geq \|\mathbf{u}\|_{A_{D_l}}^2+p^TC_lp-\vartheta\theta_1\|\mathbf{u}\|_{A_{D_l}}^2+\vartheta\theta_2\|p\|_{M_l}^2-\vartheta\theta_3p^TC_lp.\end{aligned}$$

We choose $0 < \vartheta < \min\left(\frac{1}{\theta_1}, \frac{1}{\theta_3}\right)$, resulting in

$$\mathcal{B}_l(\mathbf{u}, p; \mathbf{v}, q) \geq \theta_4 \left(\|\mathbf{u}\|^2_{A_{D_l}} + \|p\|^2_{M_l} \right),$$

with some appropriate constant θ_4. Since

$$\begin{aligned} \|\mathbf{v}\|_{A_{D_l}} + \|q\|_{M_l} &= \|\mathbf{u} + \vartheta\mathbf{w}\|_{A_{D_l}} + \|p\|_{M_l} \\ &\leq \|\mathbf{u}\|_{A_{D_l}} + \vartheta\|\mathbf{w}\|_{A_{D_l}} + \|p\|_{M_l} \\ &\leq (1+\vartheta)\left(\|\mathbf{u}\|_{A_{D_l}} + \|p\|_{M_l} \right) \end{aligned}$$

we can sum up

$$\frac{\mathcal{B}_l(\mathbf{u}, p; \mathbf{v}, q)}{\|\mathbf{v}\|_{A_{D_l}} + \|q\|_{M_l}} \geq \theta_5 \frac{\|\mathbf{u}\|^2_{A_{D_l}} + \|p\|^2_{M_l}}{\|\mathbf{u}\|_{A_{D_l}} + \|p\|_{M_l}} \geq \frac{\theta_5}{2}\left(\|\mathbf{u}\|_{A_{D_l}} + \|p\|_{M_l} \right)$$

(where $\theta_5 = \theta_4/(1+\vartheta)$). □

Now the question remains, how restrictive assumption (4.6) is. For the Ruge-Stüben prolongation (3.14) we have discussed a similar property in Theorems 3.8 and 3.9. Since property (3.16) in Theorem 3.9 is fulfilled if every F node is strongly coupled to a C node (which is the case for Algorithm 3.10 with the modifications concerning the strongly negative couplings), we get that (3.17) holds, i.e.

$$\|e_F - P_C^F e_C\|^2_{D_l,F} \leq c\|e\|^2_{A_{D_l}}.$$

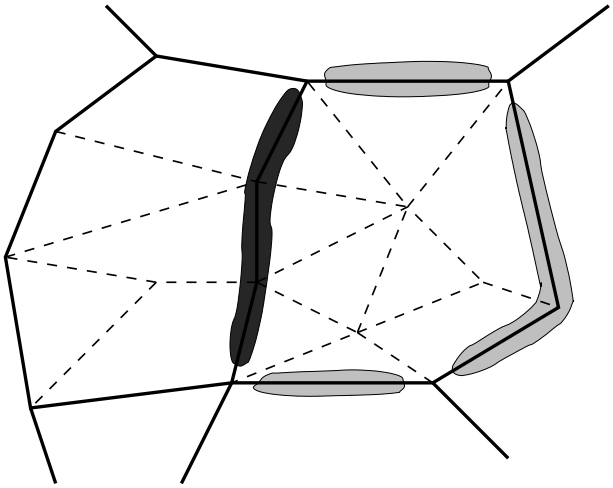
Thus, (4.6) is fulfilled for

$$\underline{\Pi}_l^{l+1}\mathbf{v}_l = \underline{\Pi}_l^{l+1}\begin{pmatrix}\mathbf{v}_F \\ \mathbf{v}_C\end{pmatrix} = \mathbf{v}_C.$$

As the methods presented in Sections 4.1.1 and 4.1.2 were more or less tailor-made for mixed elements constructed from linear elements, the question on generalizations could arise. One possibility could be the strategy developed by John, Knobloch, Matthies and Tobiska in [JKMT02] for geometric multigrid methods. There the first fine-to-coarse transfer is an element-type transfer, in their case from higher-order elements (which possess preferable discretization properties) to the P_1^{nc}-P_0 element (which possesses preferable properties in connection with the linear solver). Adapting this idea to our needs, we could discretize using an arbitrary mixed-element, transfer to the P_1-P_1-stab element and then use the hierarchy presented above. To us this seems to be a good possibility, however we have not made any numerical experiences in this direction up to now.

Another general method is the application of the element agglomerating AMGe technique (Section 3.2.2.2) on mixed elements. We exemplarily present this for the Crouzeix-Raviart element in the following section.

Figure 4.4 When constructing the coarse level shape function associated to the dark-grey edge, we set the fine (edge-) unknowns which are part of this coarse edge to 1, the fine (edge-) unknowns which are part of the light-grey edges to 0 and perform energy minimization in the interior.



4.1.3 The P_1^{nc}-P_0 Hierarchy — Applying AMGe

In the previous sections the AMG prolongation was always motivated by the prolongation which would have been constructed in the GMG case. For the P_1^{nc}-P_0 element the geometric hierarchy is non-nested, a property which cannot be achieved by AMG methods (at least not in a straight-forward manner), as the Galerkin approach here implies that the constructed hierarchy is nested. Nevertheless we try to construct a "geometric reasonable" hierarchy.

Using the element agglomeration approach of Section 3.2.2.2 we obtain on each coarse level a topology of elements, faces, edges and nodes. Now like in the geometric case we associate each face with a velocity 'node', i.e. two resp. three velocity component unknowns, and each element with a pressure unknown.

The interpolation for pressure is trivial, we do identity prolongation for all fine-level elements which are part of one coarse-level element. We propose to use identity prolongation also from each coarse-level face to the fine-level faces which are part of it and use the energy minimization approach in the interior of the element, as illustrated in Figure 4.4.

Remark 4.5. *The set-up process for this element is computationally cheaper as in the case described in Section 3.2.2.2, because we only have to perform energy minimization in the interior of each element, not for each element pairing connected by a face, and we do not have to save the element stiffness matrices separately (we do not need any face-to-face entries, and the interior-to-face and interior-to-interior entries are found explicitly in the global matrix).*

Again we can show the inf-sup condition on all levels by applying the following lemma inductively.

Lemma 4.6. *Assume that there exists a linear Operator* $\Pi_{l-1}^{l} : \mathbf{U}_{l-1} \to \mathbf{U}_l$ *with*

$$b(\Pi_{l-1}^{l}\mathbf{v}_{l-1}, q_l) = b(\mathbf{v}_{l-1}, J_l^{l-1}q_l) \quad \textit{for all } q_l \in Q_l \textit{ and } \mathbf{v}_{l-1} \in \mathbf{U}_{l-1} \tag{4.14}$$

and that

$$\|\Pi_{l-1}^{l}\mathbf{v}_{l-1}\|_1 \leq \delta \|\mathbf{v}_{l-1}\|_1 \quad \textit{for all } \mathbf{v}_{l-1} \in \mathbf{U}_{l-1}, \tag{4.15}$$

with δ *independent of* h *and* l.

Then the inf-sup condition in $\mathbf{U}_{l-1} \times Q_{l-1}$ *implies the inf-sup condition in* $\mathbf{U}_l \times Q_l$.

Proof.

$$\begin{aligned}\inf_{0\neq q_l \in Q_l} \sup_{\mathbf{0}\neq \mathbf{v}_l \in \mathbf{U}_l} \frac{b(\mathbf{v}_l, q_l)}{\|v_l\|_1 \|q_l\|_0} &\geq \inf_{0\neq q_l \in Q_l} \sup_{\mathbf{0}\neq \mathbf{v}_{l-1} \in \mathbf{U}_{l-1}} \frac{b(\Pi_{l-1}^{l}\mathbf{v}_{l-1}, q_l)}{\|\Pi_{l-1}^{l}\mathbf{v}_l\|_1 \|q_l\|_0} \\ &= \inf_{0\neq q_l \in Q_l} \sup_{\mathbf{0}\neq \mathbf{v}_{l-1} \in \mathbf{U}_{l-1}} \frac{b(\mathbf{v}_{l-1}, J_l^{l-1}q_l)}{\|\mathbf{v}_{l-1}\|_1 \|q_l\|_0} \cdot \frac{\|\mathbf{v}_{l-1}\|_1}{\|\Pi_{l-1}^{l}\mathbf{v}_{l-1}\|_1} \geq \tilde{\delta}\end{aligned}$$

□

The proof of the following theorem is rather technical as we often have to switch between two consecutive levels and the finest level.

Theorem 4.7. *Assume that*

$$\frac{h_{max}}{h_{min}} \leq \gamma, \tag{4.16}$$

where h_{max} *is the maximal element diameter and* h_{min} *the minimal diameter (at the finest level), and* γ *is a positive constant, and assume that the coarse levels are built as described above.*

Then the inf-sup condition holds on all levels.

Proof. For the proof we will construct an operator Π_{l-1}^{l} with properties (4.14), (4.15).

We consider the 2D case first (illustrated in Figure 4.5). Define on level l the index sets $\mathcal{E}_j^l$ of all $(l-1)$-level edges which are part of l-level edge j. We define the length of a l-level edge recursively by

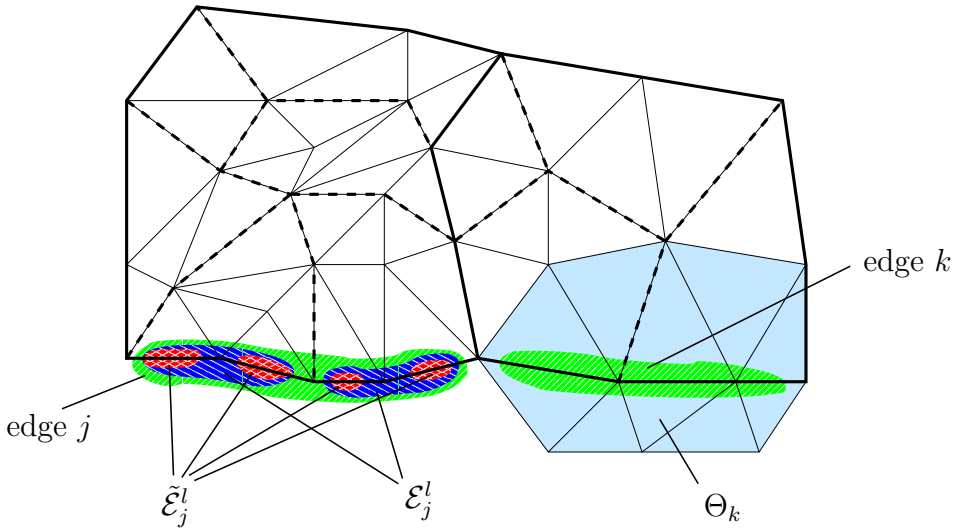
$$e_j^l := \sum_{k\in\mathcal{E}_j^l} e_k^{l-1} \quad (\text{for } l > 1),$$

for $l = 1$ it is determined by the mesh.

We now construct Π_{l-1}^{l} as follows. For some $(l-1)$-level function v_{l-1} the l-level function $\Pi_{l-1}^{l}v_{l-1}$ is determined by its values on the (l-level) edges. We set the value on a certain l-level edge to the weighted mean of the values of v_{l-1} on the $(l-1)$-level edges which are part of the edge, i.e.

$$\left(\underline{\Pi_{l-1}^{l}v_{l-1}}\right)_j = \frac{1}{e_j^l}\sum_{k\in\mathcal{E}_j^l} e_k^{l-1}\left(\underline{v}_{l-1}\right)_k.$$

Figure 4.5 The solid thick black lines describe two level l elements, the dashed lines level $l-1$ elements and the solid thin lines the finest level elements. For edge j this figure shows the sets $\mathcal{E}_j^l$ and $\tilde{\mathcal{E}}_j^l$, for edge k the tube Θ_k.



For a vector valued function $\mathbf{v}_{l-1}$ the term $\Pi_{l-1}^l \mathbf{v}_{l-1}$ will denote the application of Π_{l-1}^l to the components.

The fact that Π_{l-1}^l fulfills (4.14) is seen as follows. We want to show that

$$\sum_j \int_{\tau_j} \operatorname{div}(\Pi_{l-1}^l \mathbf{v}_{l-1}) \cdot q_l \, \mathrm{d}\mathbf{x} = \sum_j \int_{\tau_j} \operatorname{div} \mathbf{v}_{l-1} \cdot (J_l^{l-1} q_l) \, \mathrm{d}\mathbf{x}.$$

Because $q_l = J_l^{l-1} q_l$ (in functional notation), because q_l is piecewise constant on the l-level agglomerates E_j, and because both $\mathbf{v}_{l-1}$ and $\Pi_{l-1}^l \mathbf{v}_{l-1}$ are piecewise linear on the (finest level) elements τ_j and continuous at the midpoints of their edges, we can use partial integration to derive

$$\sum_j \int_{\tau_j} \operatorname{div}(\Pi_{l-1}^l \mathbf{v}_{l-1}) \cdot q_l \, \mathrm{d}\mathbf{x} = \sum_j q_l(E_j) \int_{\partial E_j} (\Pi_{l-1}^l \mathbf{v}_{l-1}) \cdot \mathbf{n}$$

and

$$\sum_j \int_{\tau_j} \operatorname{div} \mathbf{v}_{l-1} \cdot (J_l^{l-1} q_l) \, \mathrm{d}\mathbf{x} = \sum_j q_l(E_j) \int_{\partial E_j} \mathbf{v}_{l-1} \cdot \mathbf{n}.$$

By the definition of Π_{l-1}^l we see that

$$\int_{\partial E_j} (\Pi_{l-1}^l \mathbf{v}_{l-1}) \cdot \mathbf{n} = \int_{\partial E_j} \mathbf{v}_{l-1} \cdot \mathbf{n} \quad \text{for all agglomerates } E_j,$$

therefore (4.14) is shown to be true.

What remains to show is (4.15). This will be done by the introduction of an auxiliary operator $\tilde{\Pi}^l_{l-1}$ on the finer level $\mathbf{U}_{l-1}$, which fulfills (4.15) and which is identical to Π^l_{l-1} on the coarse edges. Because we use energy minimization for the interpolation in the interior of agglomerates we will then be able to estimate Π^l_{l-1} by $\tilde{\Pi}^l_{l-1}$ which will complete the proof.

We define $\tilde{\Pi}^l_{l-1} : \mathbf{U}_{l-1} \to \mathbf{U}_{l-1}$ by

$$\underline{\tilde{\Pi}}^l_{l-1}(\underline{v}_{l-1})_j := \begin{cases} (\underline{v}_{l-1})_j & \text{if } j \notin \mathcal{E}^l_k \text{ for all } k, \\ \left(\underline{\Pi^l_{l-1} v_{l-1}}\right)_k & \text{if } j \in \mathcal{E}^l_k \text{ for a certain } k. \end{cases}$$

Note that $\tilde{\Pi}^l_{l-1} v_{l-1}$ still 'lives' on level $l-1$, only the values at the l-level faces are averaged.

We try to find an upper bound for $\left| v_{l-1} - \tilde{\Pi}^l_{l-1} v_{l-1} \right|_1$. Define $\tilde{\mathcal{E}}^l_j$ the index set of all *finest*-level edges which lie on coarse edge j. Set $\tilde{v}_1$ (component of finest-level function $\tilde{\mathbf{v}}_1 \in \mathbf{U}_1$) equal to $v_{l-1} - \tilde{\Pi}^l_{l-1} v_{l-1}$ on all (finest level) degrees of freedom in $\bigcup_j \tilde{\mathcal{E}}^l_j$ and zero on all other (finest level) degrees of freedom. Then because of the energy minimization in the prolongation

$$\begin{aligned} \int_{\mathcal{G}} \nabla \left(v_{l-1} - \tilde{\Pi}^l_{l-1} v_{l-1} \right) \nabla \left(v_{l-1} - \tilde{\Pi}^l_{l-1} v_{l-1} \right) \mathrm{d}\mathbf{x} &\leq \int_{\mathcal{G}} \nabla \tilde{v}_1 \nabla \tilde{v}_1 \, \mathrm{d}\mathbf{x} \\ &\leq \sum_{\substack{l\text{-level} \\ \text{edges } j}} \int_{\Theta_j} \nabla \tilde{v}_1 \nabla \tilde{v}_1 \, \mathrm{d}\mathbf{x}, \end{aligned} \tag{4.17}$$

where Θ_j is the tube of (finest level) elements which share a point or edge with l-level edge number j.

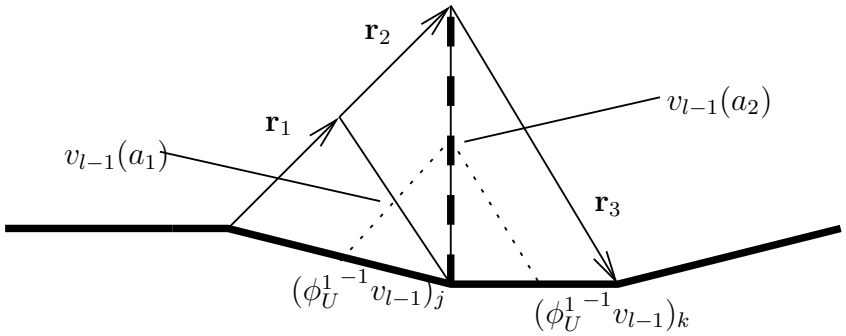
For a (finest level) triangle PQR and the basis function φ_{PQ}, which is equal to 1 at the midpoint of PQ and zero at the midpoints of QR and RP one can easily calculate

$$\int_{PQR} \nabla \varphi_{PQ} \nabla \varphi_{PQ} \, \mathrm{d}\mathbf{x} = \frac{|PQ|^2}{A(PQR)},$$

where $A(PQR)$ denotes the area of the triangle PQR. Now with

$$c_1 := \max_{\tau_j} \frac{(\text{length of longest edge of } \tau_j)^2}{A(\tau_j)}$$

Figure 4.6 Detail of Figure 4.5, the set $\mathcal{S}_m^{l-1}$.



we get

$$
\begin{aligned}
\int_{\Theta_j} \nabla\tilde{v}_1 \nabla\tilde{v}_1 \,\mathrm{d}\mathbf{x} &\le c_1 \sum_{s\in\tilde{\mathcal{E}}_j^l} \left((\underline{\tilde{v}}_1)_s\right)^2 \\
&= c_1 \sum_{s\in\tilde{\mathcal{E}}_j^l} \left[\frac{1}{\sum_{k\in\tilde{\mathcal{E}}_j^l} e_k} \sum_{k\in\tilde{\mathcal{E}}_j^l} e_k \left(\left(\phi_U^{1\,-1} v_{l-1}\right)_s - \left(\phi_U^{1\,-1} v_{l-1}\right)_k \right) \right]^2 \\
&\le c_1 \frac{\sum_{k\in\tilde{\mathcal{E}}_j^l} e_k^2}{\left(\sum_{k\in\tilde{\mathcal{E}}_j^l} e_k\right)^2} \sum_{\substack{s\in\tilde{\mathcal{E}}_j^l,\\ k\in\tilde{\mathcal{E}}_j^l}} \left(\left(\phi_U^{1\,-1} v_{l-1}\right)_s - \left(\phi_U^{1\,-1} v_{l-1}\right)_k \right)^2 \\
&\le c_1 \sum_{\substack{s\in\tilde{\mathcal{E}}_j^l,\\ k\in\tilde{\mathcal{E}}_j^l}} \left(\left(\phi_U^{1\,-1} v_{l-1}\right)_s - \left(\phi_U^{1\,-1} v_{l-1}\right)_k \right)^2 \\
&\le \bar{c}_1 \sum_{\substack{s\in\mathcal{E}_j^l,\\ k\in\mathcal{E}_j^l}} \left(\left(\underline{v}_{l-1}\right)_s - \left(\underline{v}_{l-1}\right)_k \right)^2,
\end{aligned}
\tag{4.18}
$$

where $\phi_U^{1\,-1} v$ is the representation of a coarse function v on $\underline{\mathbf{U}}_1$ as in Section 4.1.

We note that ∇v_{l-1} is constant on each finest level element. Therefore we can derive the following estimate (illustrated in Figure 4.6). Assume that the $(l-1)$-level edges j and k share the node m. We denote the set of all finest level elements which share the node m with $\mathcal{S}_m^{l-1}$, its index set with $\underline{\mathcal{S}}_m^{l-1}$ (where we assume w.l.o.g. $\underline{\mathcal{S}}_m^{l-1} = \{1, 2, \ldots, i+1\}$). For each element τ_e in $\mathcal{S}_m^{l-1}$ we denote the edge-vector of the edge not connected to m ("in direction" $j \to k$) with $\mathbf{r}_e$.

Then

$$\begin{aligned}(\underline{v}_{l-1})_k - (\underline{v}_{l-1})_j &= ((\underline{v}_{l-1})_k - v_{l-1}(a_1)) + (v_{l-1}(a_1) - v_{l-1}(a_2)) + \ldots + (v_{l-1}(a_i) - (\underline{v}_{l-1})_j) \\ &= \frac{1}{2}\left(\nabla v_{l-1}|_{\tau_1} \cdot \mathbf{r}_1 + \ldots + \nabla v_{l-1}|_{\tau_{i+1}} \cdot \mathbf{r}_{i+1}\right) \\ &\leq \frac{1}{2}\left(\sqrt{\nabla v_{l-1}|_{\tau_1} \cdot \nabla v_{l-1}|_{\tau_1}}|\mathbf{r}_1| + \ldots + \sqrt{\nabla v_{l-1}|_{\tau_{i+1}} \cdot \nabla v_{l-1}|_{\tau_{i+1}}}|\mathbf{r}_{i+1}|\right),\end{aligned}$$

where $a_1,\ldots,a_i$ are finest level edge midpoints as in Figure 4.6, thus (using the algebraic-geometric mean inequality)

$$\begin{aligned}\left((\underline{v}_{l-1})_k - (\underline{v}_{l-1})_j\right)^2 &\leq \frac{i+1}{4}\left[|\mathbf{r}_1|^2\left(\nabla v_{l-1}|_{\tau_1}\right)^2 + \ldots + |\mathbf{r}_{i+1}|^2\left(\nabla v_{l-1}|_{\tau_{i+1}}\right)^2\right] \\ &\leq c_2 \int_{\mathcal{S}_m^{l-1}} \nabla v_{l-1} \nabla v_{l-1}\,\mathrm{d}\mathbf{x}.\end{aligned} \tag{4.19}$$

We apply this estimate to the last term in (4.18), which is done directly for those $(l-1)$-level edges s and k which share a node. For all others we have to build a chain of connecting edges.

This leads to

$$\int_{\Theta_l} \nabla \tilde{v}_1 \nabla \tilde{v}_1\,\mathrm{d}\mathbf{x} \leq c_3 \int_{\Theta_l} \nabla v_{l-1} \nabla v_{l-1}\,\mathrm{d}\mathbf{x}.$$

Now because of (4.17) we get

$$\left\| v_{l-1} - \tilde{\Pi}_{l-1}^l v_{l-1} \right\|_1 \leq c_4 \|v_{l-1}\|_1,$$

thus

$$\left\| \tilde{\Pi}_{l-1}^l v_{l-1} \right\|_1 \leq (1+c_4)\|v_{l-1}\|_1.$$

Because we use energy minimization for the interpolation in the interior of coarse agglomerates, $\tilde{I}_l^{l-1}\overline{\Pi_{l-1}^l v_{l-1}}$ has minimal energy amongst all $l-1$-level functions which are identical to it on the $\overline{l\text{-level edges}}$, therefore

$$\left\|\Pi_{l-1}^l v_{l-1}\right\|_1 \leq c_5 \left\|\tilde{\Pi}_{l-1}^l v_{l-1}\right\|_1 \leq c_5(1+c_4)\|v_{l-1}\|_1. \tag{4.20}$$

3D case. For 3D tetrahedral elements we replace c_1 in (4.18) by $c_1 h_{j,\max}$, where $h_{j,\max}$ is the maximal element height in tube Θ_j, and c_2 in (4.19) by $c_2/h_{j,\min}$, where $h_{j,\min}$ is the minimal element height in this tube. Then because of (4.16) the argumentation remains unchanged, only the scaling argument is based on the (finest level) tetrahedron $PQRS$

$$\int_{PQRS} \nabla \varphi_{PQR} \nabla \varphi_{PQR}\,\mathrm{d}\mathbf{x} = \frac{A(PQR)^2}{V(PQRS)},$$

where $V(PQRS)$ is the volume of the tetrahedron. □

4.2 Smoothers

Now that the coarse levels are constructed we need smoothers for the coupled systems on all levels. We will give a short overview of some possibilities thereof, and then go into detail for the two methods we primarily use, namely Braess-Sarazin and Vanka smoothers.

To the author's knowledge there is no smoothing-theory for algebraic multigrid methods for fundamentally indefinite systems, i.e. not just disturbed positive definite systems. We admit that we have no contributions in this direction either, thus we will apply the following heuristic.

Hypothesis 4.8. If a smoother for a saddle point problem performs 'well' in the geometric multigrid situation, then it will also do so in the algebraic multigrid situation.

In the light of this heuristic an important quality factor for the smoothers is the *smoothing rate*, which we introduced in Section 3.1.1.

Note that the parts of the following sections which concern the analysis of the smoothers all base on geometric multigrid and on a symmetric system matrix, i.e. the Stokes case. As the smoothers (mostly) operate on one single level, we will drop the level index in the following sections.

4.2.1 Standard Smoothers for the Squared System

As the system we are interested in is indefinite, it is not possible to apply standard smoothing methods (e.g. Richardson, Jacobi, Gauss-Seidel) at first. This could be overcome by applying those smoothers to the squared (and thus symmetric, positive definite) system

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}^T \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}^T \begin{pmatrix} \mathbf{f} \\ g \end{pmatrix}.$$

This idea is used for example in [Ver84b] (here the second block row is scaled by $1/h^2$) or [Bre93], both use Richardson iterations on the squared system which can be shown to lead to a (geometric) smoothing rate $\eta(m)$ of order $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$.

4.2.2 Transforming Smoothers

This class of smoothers was introduced by Wittum in [Wit89, Wit90] and is based on a generalization of the factorization for the SIMPLE scheme (2.31), namely

$$K = \underbrace{\begin{pmatrix} A & 0 \\ B & E \end{pmatrix}}_{=:K_1} \underbrace{\begin{pmatrix} I & A^{-1}B^T \\ 0 & -E^{-1}S \end{pmatrix}}_{=:K_2}$$

with an arbitrary positive definite matrix E (in (2.31) we had $E = S$). The idea is to transform K by a multiplication from the right with K_2^{-1} to K_1, and then to find

smoothers for the block-triangular K_1 (in [Wit89] it is shown that they only have to fulfill the smoothing property for the diagonal blocks A and E). Suggested choices for E are S or A_s (the scalar, i.e. pressure variant of A). The latter does not lead to a practicable method at first, (heuristic) considerations about commuting operators (similarly as in Section 2.5.3.1) are needed, but we will not go into detail here.

In [Wit90] damped Jacobi, Gauss-Seidel and ILU smoothers are used for the transformed system. For damped Jacobi a smoothing rate of order $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ is shown, for the other two only $\mathcal{O}\left(\frac{\ln m}{\sqrt{m}}\right)$, but it is stated that in numerical practice a rate of order $\mathcal{O}\left(\frac{1}{m}\right)$ is observed for the ILU transforming smoother.

4.2.3 Braess-Sarazin Smoother

This smoother consists of the application of the *inexact symmetric Uzawa algorithm* (2.36), which we repeat here for convenience of reading:

$$\hat{A}(\hat{\mathbf{u}}_{k+1} - \mathbf{u}_k) = \mathbf{f} - A\mathbf{u}_k - B^T p_k, \tag{4.21a}$$

$$\hat{S}(p_{k+1} - p_k) = B\hat{\mathbf{u}}_{k+1} - Cp_k - g, \tag{4.21b}$$

$$\hat{A}(\mathbf{u}_{k+1} - \hat{\mathbf{u}}_{k+1}) = -B^T(p_{k+1} - p_k), \tag{4.21c}$$

where now $\hat{S}$ is a preconditioner for the inexact Schur-complement $C + B\hat{A}^{-1}B^T$.

The smoothing property with a rate of $\mathcal{O}\left(\frac{1}{m}\right)$ for this method was shown in [BS97] under the assumption that the pressure update (4.21b) is done (almost) exactly, i.e. $\hat{S} \equiv C + B\hat{A}^{-1}B^T$. In [Zul00] we find that the same behavior can be obtained under weaker assumptions.

Theorem 4.9. *[Zul00, theorem 2] Let A and C be symmetric positive semi-definite matrices, $\hat{A}$ and $\hat{S}$ symmetric positive definite matrices, satisfying*

$$\hat{A} \geq A, \tag{4.22a}$$

$$\hat{S} \leq C + B\hat{A}^{-1}B^T, \tag{4.22b}$$

$$C + B\hat{A}^{-1}B^T \leq (1+\beta)\hat{S}, \tag{4.22c}$$

and

$$\|\hat{A}\|_{\ell_2} \leq c_1 h^{-2}, \;\; \|\hat{A}^{-1}\|_{\ell_2} \leq c_2 h^2, \;\; \|B\|_{\ell_2} \leq c_3 h^{-2}, \;\; \|C\|_{\ell_2} \leq c_4 h^{-2}.$$

Then the smoothing property

$$\|K\mathcal{S}^m\|_{\ell_2} \leq \eta(m) h^{-2}$$

is satisfied with

$$\eta(m) = \max\left[(1+\rho)\rho^{m-1}, \frac{(m-1)^{m-1}}{m^m}\right], \quad \textit{with} \;\; \rho = \beta + \sqrt{\beta^2 + \beta}.$$

For $\beta < 1/3$ we get $\rho < 1$, thus the second term in the maximum will dominate for m large enough. As this term can be bounded by

$$\frac{(m-1)^{m-1}}{m^m} = \frac{\left(1-\frac{1}{m}\right)^m}{(m-1)} \leq \frac{1}{e(m-1)} \quad \text{for } m > 1$$

we get the smoothing rate $\mathcal{O}(\frac{1}{m})$ mentioned above.

We use an (inner) AMG method for $C + B\hat{\hat{A}}^{-1}B^T$ as $\hat{S}^{-1}$, where $\hat{\hat{A}} = \alpha \cdot D$, with the diagonal D of A, (damped Jacobi iteration; thus we are able to explicitly construct the matrix, what is needed for the AMG method). For $\hat{A}$ we want to use some damped SSOR (or SSUR) iteration, i.e.

$$\hat{A} = \delta(D + \gamma E)D^{-1}(D + \gamma F),$$

where E is the lower left triangular part (with zero diagonal), and F the upper right triangular part (with zero diagonal) of A, and δ and γ are some parameters. The following corollary shows the impact of (4.22) on the choice of the parameters α, δ and γ, under the assumption that A is an M-matrix. For essentially positive type matrices similar results as the following could be shown, for simplicity of presentation we will not go into the details but restrict ourselves to the M-matrix case.

Corollary 4.10. *Assume that*

- *A is an M-matrix, that*
- *$\hat{A}$ and $\hat{S}$ are constructed as indicate above, and that*
- *the estimate*

$$(1-\eta^k)\hat{S} \leq C + \frac{1}{\alpha}BD^{-1}B^T \leq \hat{S}, \tag{4.23}$$

  where $\eta < 1$ is the convergence rate of the multigrid method and k the number of iterations, is valid,

and define

$$\omega := \sup_{0 \neq p \in \mathbb{R}^m} \frac{p^T C p}{p^T B D^{-1} B^T p}.$$

Then (4.22) is fulfilled if

$$\delta(1-\gamma) \geq 2, \tag{4.24a}$$

$$\delta(1+\gamma+\rho^2\gamma^2) \leq \alpha\frac{1-\eta^k}{1+\omega\eta^k}, \quad \textit{and} \tag{4.24b}$$

$$(1+\beta)\delta(1-\gamma) \geq \alpha \tag{4.24c}$$

hold, where

$$\rho := \frac{\max_i a_{ii}}{\min_i a_{ii}}.$$

Proof. It is easily seen that

$$\alpha D \leq (1+\beta)\delta(D+\gamma F+\gamma E+\gamma^2 E D^{-1}F)$$

is a sufficient condition for (4.22c), which is certainly fulfilled for (4.24c).

Analogously one can show that (4.24a) is sufficient for (4.22a).

What remains is (4.22b), a sufficient condition would be

$$\delta(D+\gamma E+\gamma F+\gamma^2 E D^{-1}F) \leq \alpha\frac{1-\eta^k}{1+\omega\eta^k}D.$$

To estimate the last term on the left hand side we split as follows

$$\frac{\mathbf{x}^T E D^{-1}F\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \leq \|ED^{-1/2}\|_\infty \|D^{-1/2}F\|_\infty$$

and calculate

$$\|ED^{-1/2}\|_\infty = \max_i\left(\sum_{j=1}^{i-1}\frac{|a_{ij}|}{\sqrt{a_{jj}}}\right) \leq \rho\max_i\left(\sum_{j=1}^{i-1}\frac{|a_{ij}|}{\sqrt{a_{ii}}}\right) \leq \rho\|D^{1/2}\|_\infty$$

and similarly

$$\|D^{-1/2}F\|_\infty \leq \|D^{1/2}\|_\infty.$$

Now we get for all $\mathbf{x}$

$$\frac{\mathbf{x}^T E D^{-1}F\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \leq \rho\|D\|_\infty \leq \rho^2\frac{\mathbf{x}^T D\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$$

and summing up

$$\begin{aligned} D+\gamma E+\gamma F+\gamma^2 E D^{-1}F &= (1-\gamma)D+\gamma A+\gamma^2 E D^{-1}F \\ &\leq (1+\gamma+\rho^2\gamma^2)D. \end{aligned}$$

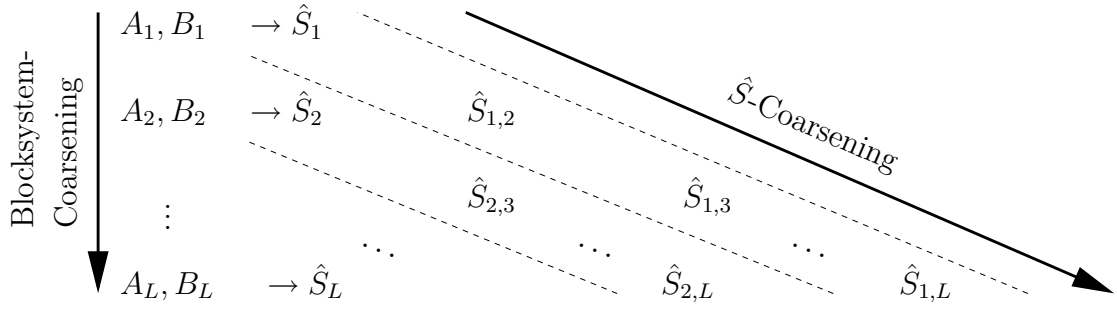
Thus we get the sufficient condition for (4.22b)

$$\delta(1+\gamma+\rho^2\gamma^2) \leq \alpha\frac{1-\eta^k}{1+\omega\eta^k}.$$

□

Remark 4.11. *Condition (4.23) is fulfilled e.g. if $\hat{S}$ originates in a convergent, symmetric multigrid method and some additional (but weak) assumption hold (for example that the coarse level systems are constructed with the Galerkin approach), for details we refer e.g. to [Hac93].*

Figure 4.7 If we use Galerkin projections for the inner AMG, we have to build and keep all the matrices $\hat{S}_{i,j}$



Equations (4.24a)–(4.24c) provide a set of sufficient conditions for (4.22) but are for $\gamma > 0$ very pessimistic as we will show in the numeric results. If we assume for example that k is large enough and therefore

$$\frac{1-\eta^k}{1+\omega\eta^k} \approx 1,$$

and further that $\rho = 1$, which corresponds to a uniform mesh, then we can only calculate a maximal admissible $\gamma = 0.135$ and $\delta \geq 2.32$, $\alpha \geq 2.68$. For $\gamma = 0$, i.e. Jacobi iteration for $\hat{A}$, we get a minimal allowed $\alpha = 2$ and then $\delta = 2$, which is also the choice for the numerical tests in [Zul00].

Remark 4.12. *It seems to be a good idea (at least from the point of view of computer-memory consumption) to build the coarse matrices for the (inner) $\hat{S}$-AMG method using the coarse level versions of the matrices C, B and A. Instinctively one may think that it does not make a big difference that this does not correspond to the Galerkin approach. But especially for complex 3D problems we run into convergence problems, which we do not have if we use Galerkin projected matrices.*

Unfortunately the plain Galerkin approach causes an increase of memory-usage because we have to perform the coarsenings for $\hat{S}$ on each level, illustrated in Figure 4.7.

4.2.4 Vanka Smoothers

The discretization of the Navier-Stokes equations which was used by Vanka when he introduced this method [Van86] was a finite volume method on a staggered grid with pressure nodes at the cell-centers and velocity nodes at the cell-faces. Small subproblems are set up cell by cell — i.e. with one pressure degree of freedom and the connected velocity unknowns — and the solutions are combined using a multiplicative Schwarz iteration. The smoothing property of this method in a finite element context was analyzed (for the additive Schwarz case) in [SZ03]. We will sketch shortly the prerequisites needed therein.

The local sub-problems are set up using the (local) prolongators (on a fixed multigrid-level l)

$$\pi_j : \mathbb{R}^{n_{l,j}} \to \mathbb{R}^{n_l}, \quad \sigma_j : \mathbb{R}^{m_{l,j}} \to \mathbb{R}^{m_l},$$

where j is the index of the sub-problem and $n_{l,j}$ and $m_{l,j}$ determine its dimension, and we assume

$$\sum_j \pi_j \pi_j^T = I \tag{4.25}$$

and that

$$\sum_j \sigma_j \sigma_j^T \text{ is nonsingular.} \tag{4.26}$$

The small problems are now constructed as

$$\begin{pmatrix} \hat{A}_j & B_j^T \\ B_j & \frac{1}{\beta}\left[(\beta-1)B_j\hat{A}_j^{-1}B_j^T - C_j\right] \end{pmatrix} \begin{pmatrix} \mathbf{v}_j \\ q_j \end{pmatrix} = \begin{pmatrix} \pi_j^T \mathbf{r} \\ \sigma_j^T s \end{pmatrix},$$

where $\mathbf{r}$ and s are the global residuals, β some relaxation parameter, and the following relations have to be fulfilled for all j

$$\pi_j^T \hat{A} = \hat{A}_j \pi_j^T, \tag{4.27a}$$
$$\sigma_j^T B = B_j \pi_j^T, \tag{4.27b}$$
$$C_j = \sigma_j^T C \sigma_j, \tag{4.27c}$$

where $\hat{A}$ is a preconditioner for A.

Remark 4.13. *In [SZ03, theorem 1] we find that for the additive Schwarz case the method can again be represented as preconditioned Richardson iteration with preconditioner*

$$\hat{K} = \begin{pmatrix} \hat{A} & B^T \\ B & B\hat{A}^{-1}B^T - \hat{S} \end{pmatrix}, \tag{4.28}$$

with $\hat{S} = \left(\sum_j \sigma_j \hat{S}_j^{-1} \sigma_j^T\right)^{-1}$ *and* $\hat{S}_j = \frac{1}{\beta}(C_j + B_j \hat{A}_j^{-1} B_j^T)$*, and is therefore contained in the class of inexact symmetric Uzawa algorithms (compare with (2.35)).*

The smoothing result obtained in this situation is summarized in the following theorem (without proof).

Theorem 4.14. *[SZ03, theorem 4] For* $\hat{K}$ *as in (4.28) with* $\hat{A}$ *and* $\hat{S}$ *symmetric positive definite,*

$$\hat{A} \geq A, \tag{4.29}$$
$$\hat{S} \geq C + B\hat{A}^{-1}B^T, \tag{4.30}$$

and

$$\|\hat{K} - K\|_{\ell_2} \leq c\|K\|_{\ell_2}$$

we get that the smoothing property

$$\|K\mathcal{S}^m\|_{\ell_2} \leq \eta(m)\|K\|_{\ell_2}$$

is satisfied with $\eta(m) = \mathcal{O}(1/\sqrt{m})$.

We will now present one possibility of constructing the small systems meeting the requirements (4.25), (4.26) and (4.27) on a specific level. Choose a partitioning of the set of pressure unknowns $\bigcup_j \mathcal{P}_j$ (with $\mathcal{P}_j \cap \mathcal{P}_k = \emptyset$ for $j \neq k$) and build corresponding sets of velocity nodes $\mathcal{V}_j$, where a velocity node is contained in $\mathcal{V}_j$ if and only if it is connected to a pressure node in $\mathcal{P}_j$ via an entry in the matrix B.

The matrix $\hat{A}$ is now constructed by setting those entries a_{ij} of A to zero where $i \neq j$ and $i \in \mathcal{V}_{\hat{\imath}}$, $j \in \mathcal{V}_{\hat{\jmath}}$ for some $\hat{\imath}$, $\hat{\jmath}$ with $\hat{\imath} \neq \hat{\jmath}$. This can be interpreted as the application of a Jacobi method on the boundary of the sets $\mathcal{V}_j$ and using the full matrix information in the interior. Some scaling may be needed because of (4.29).

Then σ_j can be chosen as the canonical embedding from $\mathbb{R}^{m_{l,j}}$ into $\mathbb{R}^{m_l}$. For π_j we start with the canonical embedding from $\mathbb{R}^{n_{l,j}}$ into $\mathbb{R}^{n_l}$, which we denote with $\hat{\pi}_j$ and then scale each row with $1/\sqrt{s}$, where s is the number of sets $\mathcal{V}_k$ that the node associated to the row is part of, to fulfill (4.25).

Now $\hat{A}_j$ is extracted from the matrix $\hat{A}$ as one might expect, with the full block for the interior unknowns and the diagonal for the boundaries of the patch. The local matrix B_j is directly extracted directly from B but with a scaling according to the above construction of π_j and (4.27b). Finally β has to be chosen such that (4.30) is fulfilled.

Computationally cheaper versions of $\hat{A}$ can be built by using only the diagonal of the interior-unknown-blocks, the upper triangle (corresponds to Gauss-Seidel), or by applying some sort of lumping.

Remark 4.15. *If one wants to use information of the whole patch blocks of A but wants the matrices $\hat{A}_j$ to be diagonal (because then the small problems can be solved faster), numerical experiments have shown that they should be constructed based on the following heuristics.*

We first build the full local matrix $\tilde{A}_j = (\tilde{a}_{kl})$ which we want to approximate by a diagonal matrix $\hat{A}_j$. Assume that for given $\mathbf{f}_j$ the vectors $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ are the solutions of

$$\tilde{A}_j\tilde{\mathbf{x}} = \mathbf{f}_j \quad \textit{and} \quad \hat{A}_j\hat{\mathbf{x}} = \mathbf{f}_j.$$

Now we want $\hat{\mathbf{x}}$ to fulfill

$$\tilde{A}_j\hat{\mathbf{x}} \approx \mathbf{f}_j,$$

thus we try to minimize

$$\|\tilde{A}_j\hat{\mathbf{x}} - \tilde{A}_j\tilde{\mathbf{x}}\|_{\ell_2}.$$

Now

$$\begin{aligned}\|\tilde{A}_j\hat{\mathbf{x}} - \tilde{A}_j\tilde{\mathbf{x}}\|_{\ell_2} &= \|(I - \tilde{A}_j\hat{A}_j^{-1})\mathbf{f}_j\|_{\ell_2} \\ &\leq \|(I - \tilde{A}_j\hat{A}_j^{-1})\|_F\|\mathbf{f}_j\|_{\ell_2},\end{aligned}$$

where $\|.\|_F$ is the Frobenius norm. If we determine $\hat{A} = \operatorname{diag}\{\hat{a}_1, \ldots, \hat{a}_{n_{l,j}}\}$ such that $\|(I - \tilde{A}_j\hat{A}_j^{-1})\|_F$ is minimal, then this leads to

$$\hat{a}_k = \frac{1}{\tilde{a}_{kk}} \sum_l \tilde{a}_{lk}^2.$$

Chapter 5

Software and Numerical Studies

Now we fill the methods of the previous chapters with life, which means that we apply them to problems with various levels of complexity. But before that, we give a short overview of the software developed in the course of the working on this thesis.

5.1 The Software Package AMuSE

All the numerical tests in this thesis were performed using the software package AMuSE — Algebraic Multigrid for Stokes-type Equations — which was developed by the author. It is based on the mesh generator and AMG solver for potential equation and plain strain elasticity problems NAOMI by Ferdinand Kickinger[1] [Kic96, Kic97a, Kic97b, Kic98] and was also contributed to by Christoph Reisinger[2].

Apart from this thesis AMuSE or its earlier incarnation NAOMI have been used as solvers for mixed problems in several projects [RW99, Him02, Bec02, Pönng].

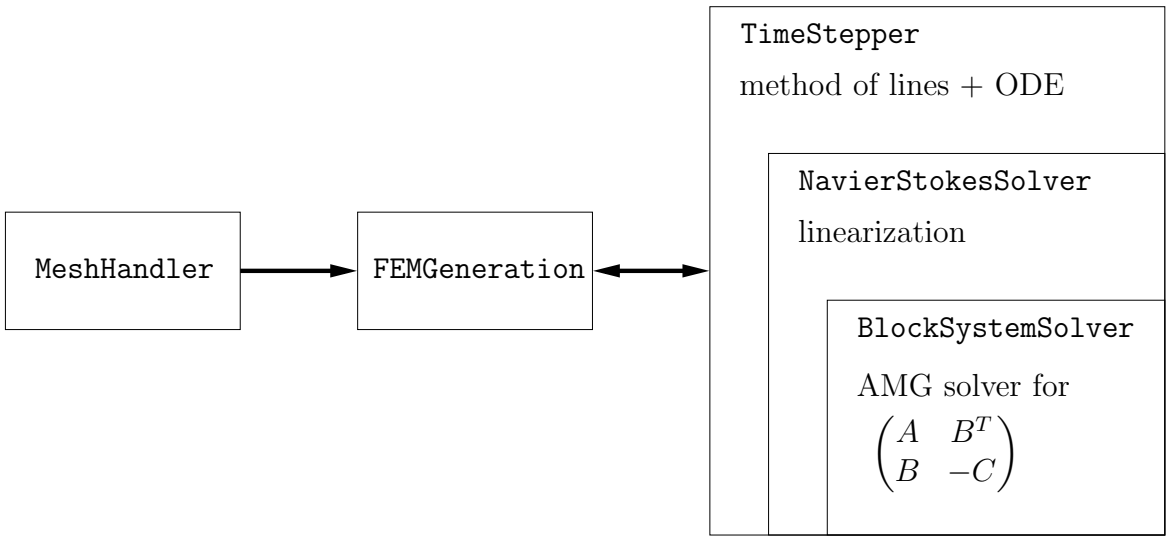
It is mostly written in C++ and takes advantage of the object oriented capabilities of this programming language (we will not go into detail about that but refer to the standard literature, e.g. [Str97]). We want to emphasize that there may be faster codes than ours for some of the methods mentioned in this thesis (in fact, the author is very sure that there are), many modern and efficient programming techniques (e.g. expression templates, cache aware programming, etc.) or parallelization were not applied. But the aim was not to develop a code which is the fastest for one method, but to have a tool to compare various methods using the same basic programming environment. Thus, we do not to compare different implementations but really different methods. Therefore it may not be sensible to look at the absolute timings presented in the sections with numerical results, but only to compare different timings for different methods.

We will now shortly sketch the structure of AMuSE, where we will use `typewriter` font for expressions directly related to the source code of the package.

[1] now at AVL List GmbH, Graz, Austria

[2] now at University of Heidelberg, Germany

Figure 5.1 The general structure of AMuSE. More complex dependencies (e.g. due to moving meshes, etc.) were omitted in this figure.



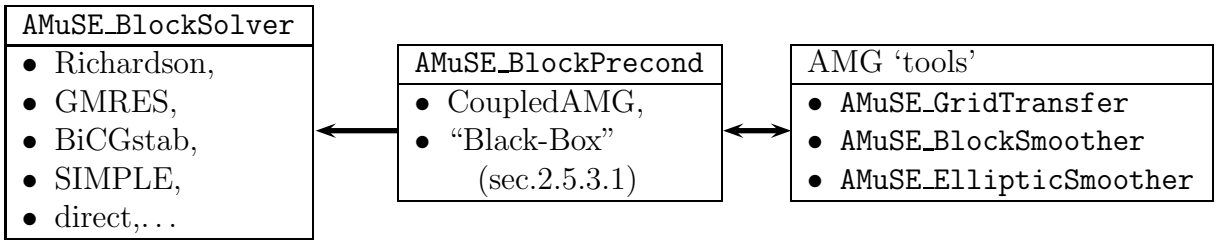
5.1.1 Structure

The structure of the program is sketched in Figure 5.1. The central part (at least concerning this thesis) is the block-system solver, its components are described in Figure 5.2.

The following external libraries and packages are used in parts of the program:

- *Template Numerical Toolkit (TNT)*, Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Gaithersburg, MD USA. Used for the direct solution of the coarse level systems.
- *General Mesh Viewer (gmv)*, Applied Physics Division, Los Alamos National Laboratory, CA USA. Used for the visualization of the numerical results.

Figure 5.2 The block-system-solvers are implemented as derived classes of the base `AMuSE_BlockSolver`. Each one (except the direct solvers) can use a preconditioner, i.e. an object which is derived from `AMuSE_BlockPrecond`. And the preconditioners (which use some AMG solver) can use and manipulate the structures needed for an AMG method.



- *gzstream*, Deepak Bandyopadhay and Lutz Kettner, and *zlib*, Jean-Loup Gailly and Mark Adler. Used for the compressed output of solutions.
- *Linuxthreads*, Xavier Leroy. Used for the shared memory parallelization of the FEM matrix generation.

5.1.2 Matrices

AMuSE provides several sparse matrix classes which are substantial for the components described above. As some non-standard ideas are used for their construction, we will sketch them in this section.

The basic (templated) class is `AMuSE_SparseMatrixData< T >`, where elements of type `T` are stored in a similar way as compressed row storage (CRS) format, the only difference is, that we use separate arrays for the element and index data of each single row, not one long array for all elements. The template parameter `T` could be a scalar type like `float` or `double` or again a small matrix (class `AMuSE_SmallMatrix< T, m, n >`, where `T` is the type of the entries and `m` and `n` are the row and column dimensions). This is used to store block matrices like

$$\begin{pmatrix} A^{1,1} & A^{1,2} & \dots & A^{1,j} \\ A^{2,1} & A^{2,2} & \dots & A^{2,j} \\ \dots & \dots & \dots & \dots \\ A^{j,1} & A^{j,2} & \dots & A^{j,j} \end{pmatrix},$$

with $k \times k$ blocks $A^{l,m}$ ($k \gg j$) with similar sparsity pattern, efficiently as

$$\begin{pmatrix} a^{1,1} & a^{1,2} & \dots & a^{1,k} \\ a^{2,1} & a^{2,2} & \dots & a^{2,k} \\ \dots & \dots & \dots & \dots \\ a^{k,1} & a^{k,2} & \dots & a^{k,k} \end{pmatrix},$$

with small $j \times j$ matrices $a^{l,m}$, where

$$(a^{l,m})_{a,b} = (A^{a,b})_{l,m}.$$

The `AMuSE_SparseMatrixData< T >` class is only used for the storage of sparse matrices. The 'mathematical' objects (which can be multiplied with vectors, 'inverted', etc.) are of type `AMuSE_SparseMatrix< T >`.

The next generalization `AMuSE_MaskedSparseMatrix< T >` can be understood in the following way. Assume we want to store the matrix A of the Oseen linearized problem, which for example in 3D in general has the form

$$\begin{pmatrix} \bar{A} & & \\ & \bar{A} & \\ & & \bar{A} \end{pmatrix}. \tag{5.1}$$

Saving it as `AMuSE_SparseMatrix< AMuSE_SmallMatrix< double, 3, 3 > >` would be a waste of computer memory, thus it is saved as `AMuSE_SparseMatrix< double >`. For certain types of boundary conditions, e.g. symmetry planes, matrix A does not totally fit structure (5.1), some entries differ in the three diagonal blocks but most do not. Using for example `AMuSE_MaskedSparseMatrix< AMuSE_SmallMatrix< double, 3, 3 > >` hides this problem from the user. From outside it behaves like `AMuSE_SparseMatrix< AMuSE_SmallMatrix< double, 3, 3 > >`, but internally it tries to use and save only `double` instead of `AMuSE_SmallMatrix< double, 3, 3 >` elements.

5.2 Numerical results

All the rates which will be stated in this section are based on the ℓ_2 norm of the residual after the i-th iteration step, i.e. on

$$\|\mathbf{r}_i\|_{\ell_2} = \|\mathbf{b} - K\mathbf{x}_i\|_{\ell_2}.$$

Because we compare methods with different costs per iteration step we prefer the following two measures for efficiency:

- the average reduction of the norm of the residual per minute CPU time (which will be abbreviated by "red./min." in the tables below, "asympt. red./min." will be the average reduction per minute in an asymptotic region of the convergence history) and
- the measure

$$T_{0.1} := \frac{\begin{pmatrix}\text{average CPU time in minutes for the reduction}\\ \text{of the norm of the residual by a factor of 0.1}\end{pmatrix}}{\text{number of unknowns}}.$$

  This number would be constant for different levels of refinement if we had an optimal method, i.e. if the work for a given reduction of the residual is $\mathcal{O}(n)$, where n is the number of unknowns.

Not all tests have been carried out on the same computers, thus a cross-comparison of those values for different tests may not be sensible. In all cases we used standard Linux-PCs.

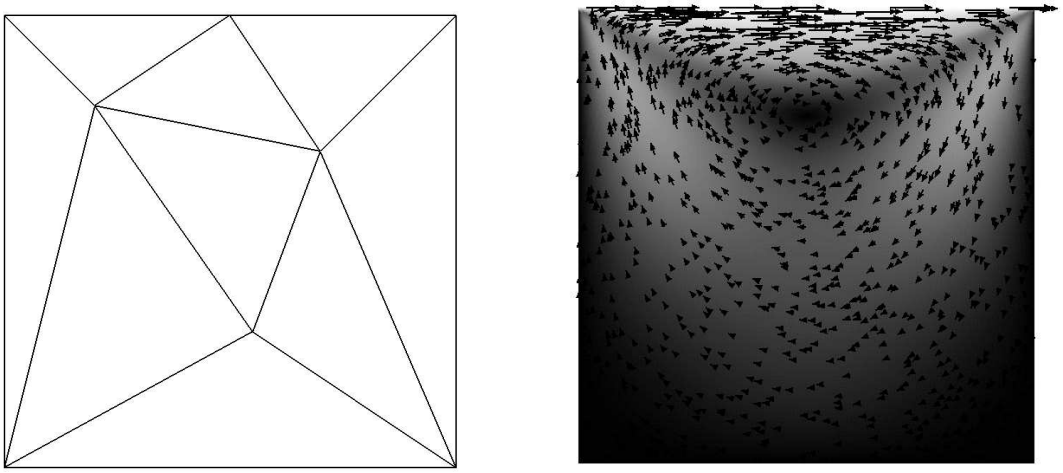
If not stated differently the results for the Oseen problems are always based on the linear problem with $\mathbf{w}$ near the solution of the corresponding Navier-Stokes equations.

The geometries in Figures 5.4, 5.11, and 5.12 were provided by AVL List GmbH, Graz, Austria.

5.2.1 2D Test Cases

We have stated to show how the methods behave for complex 3D problems, nevertheless we start with 2D problems with moderately complex geometry. The reason is, that here

Figure 5.3 The initial mesh and the numerical solution of the driven cavity Stokes problem (lighter grey indicates higher velocities)



we can carry out parameter-studies for various methods on different levels in a reasonable period of time, which would not be possible otherwise.

Our two model problems are the following:

- **Driven cavity.** The initial grid (and the numerical solution of the Stokes problem) can be found in Figure 5.3. The problem is easily described, on a unit square we pose a Dirichlet condition with velocity $(1, 0)$ on the upper boundary and homogeneous Dirichlet conditions on the rest of the boundary (the walls). The finer levels were generated by a hierarchical refinement of the coarse grid.

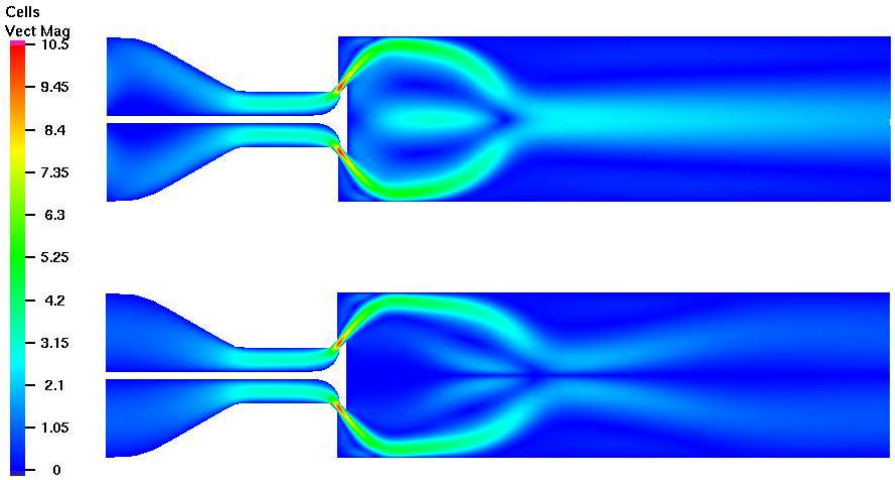
- **2D valve.** Here the geometry is one half of the region round a valve with inlet-size 0.03 and the distance between the walls at the narrowest part 0.003. The meshes were generated using Ferdinand Kickinger's NAOMI.

  We pose two problems, one with symmetry boundary conditions along the symmetry plane, the other one with homogeneous Dirichlet conditions there. In both cases we set Dirichlet condition with quadratic profile (with maximum velocity 1) on the upper boundary, natural outflow conditions on the lower boundary. The geometry and the solutions for the two problems (Navier-Stokes, $\nu = 5 \cdot 10^{-4}$) are illustrated in Figure 5.4.

Dependence on Mesh Width

In the first set of tests we want to check the "h-independence" of the methods, i.e. we solve the same Stokes problem on different levels of refinement and compare the efficiencies. We solve the driven cavity problem with modified Taylor-Hood-discretization and the red-black coloring algorithm with averaging for the coarse level construction. Just doing this

Figure 5.4 Valve problems. The upper part shows the absolute values of the velocity-solution of the problem with symmetry boundary condition, the lower part of the problem with homogeneous Dirichlet conditions at the symmetry plane (in both cases only the problem on one half of the geometry shown was solved).



on the fine level leads to a geometric multigrid method (as mentioned in Remark 3.3), the results can be found in Table 5.1 and Figure 5.5(a) (there 'BBPre' denotes the "black-box" preconditioner of Section 2.5.3.1 and 'MSM' the Vanka smoother of Section 4.2.4, where the patches consist of one pressure unknown and the connected velocity degrees of freedom). If we randomly mix the numbering of the fine level nodes, then the red-black algorithm is not able to reproduce the hierarchy, thus we have a real AMG method, for which we show the results in Table 5.2 and Figure 5.5(b). One can also apply the idea of Remark 4.1, i.e. to use the given mesh as "velocity-mesh" and to do one coarsening step to get the first pressure level, which leads to the results in Figure 5.5(c). For all tests the linear solver was stopped after a reduction of the residual by a factor of 10^{-5}.

The corresponding results for different AMG methods for the P_1-P_1-stab-discretized driven cavity problem can be found in Table 5.3 and Figure 5.6.

In both cases, for the P_1isoP_2-P_1 and the P_1-P_1-stab element, we see common behaviors of the solvers. First, for the GMG situation $T_{0.1}$ is (nearly) constant for the coupled methods, as predicted by theory. For the AMG situation this gets a little worse, but is still acceptable. The efficiency of the AMG-SIMPLE method suffers if h gets small, which is not surprising. The "black-box" preconditioner for BiCGstab performs best, which also is not surprising as it is cheap and at the same time optimal (i.e. the rates are h-independent) for the Stokes case, according to theory (c.f. Section 2.5.3.1).

Dependence on Convection

Now we take a single mesh and check how the methods perform when we want to solve the Oseen problem for different intensities of convection. For each Oseen problem we take the

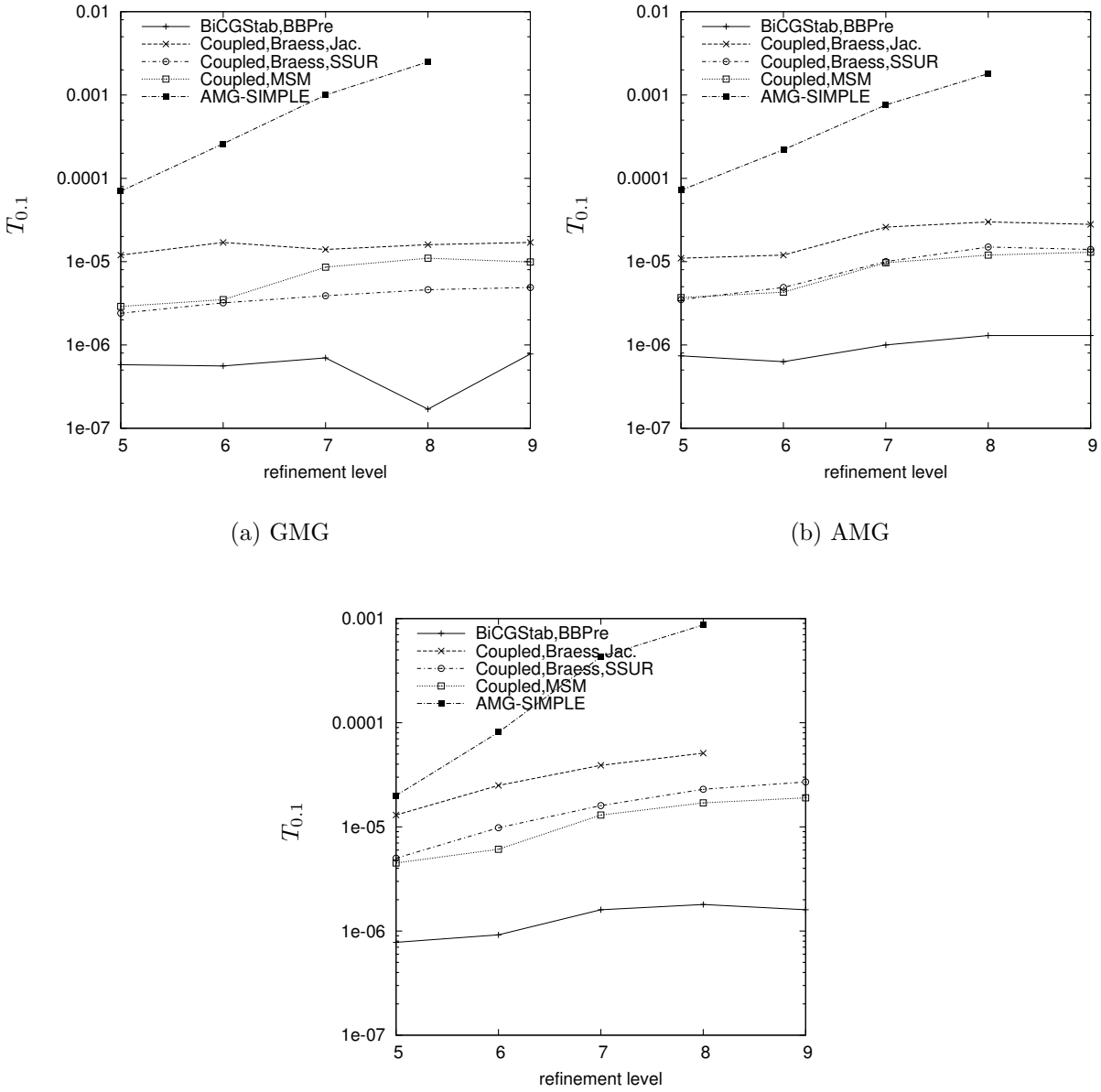
Table 5.1 The results for driven cavity Stokes flow with hierarchically refined grid (→ GMG) and modified Taylor-Hood discretization.

refinement-level		5	6	7	8	9
total number of unknowns on finest level		10,571	52,446	166,691	665,155	2,657,411
Coupled, Braess SSUR 0.8, W-6-6	red./min	8.6e-41	4.0e-8	0.03	0.47	0.84
	$T_{0.1}$	2.4e-6	3.2e-6	3.9e-6	4.6e-6	4.9e-6
Coupled, Braess Jacobi, W-12-12	red./min	1.6e-8	0.043	0.36	0.80	0.95
	$T_{0.1}$	1.2e-5	1.7e-5	1.4e-5	1.6e-5	1.7e-5
Coupled, MSM W-11-11	red./min	2.8e-33	3.3e-6	0.2	0.72	0.92
	$T_{0.1}$	2.9e-6	3.5e-6	8.6e-6	1.1e-5	9.9e-6
BiCGstab + Black-Box Prec.	red./min	1.3e-163	5.1e-35	2.4e-9	6.6e-3	0.33
	$T_{0.1}$	5.8e-7	5.6e-7	7.0e-7	1.7e-7	7.8e-7
AMG-SIMPLE	red./min	0.044	0.842	0.9869	0.9986	—
	$T_{0.1}$	7.0e-5	2.6e-4	1e-3	2.5e-3	—

Table 5.2 The results for driven cavity Stokes flow with hierarchically refined grid and renumbering (→ AMG) and modified Taylor-Hood discretization.

refinement-level		5	6	7	8	9
total number of unknowns on finest level		10,571	52,446	166,691	665,155	2,657,411
Coupled, Braess SSUR 0.8, W-6-6	red./min	1.8e-27	1.2e-4	0.26	0.79	0.94
	$T_{0.1}$	3.5e-6	4.9e-6	1.0e-5	1.5e-5	1.4e-5
Coupled, Braess Jacobi, W-12-12	red./min	4.6e-9	0.028	0.59	0.89	0.97
	$T_{0.1}$	1.1e-5	1.2e-5	2.6e-5	3.0e-5	2.8e-5
Coupled, MSM W-15-15	red./min	2.3e-26	3.7e-5	0.24	0.76	0.94
	$T_{0.1}$	3.7e-6	4.3e-6	9.7e-6	1.2e-5	1.3e-5
BiCGstab + Black-Box Prec.	red./min	6.1e-128	7.1e-31	1.6e-6	0.068	0.51
	$T_{0.1}$	7.4e-7	6.3e-7	1.0e-6	1.3e-6	1.3e-6
AMG-SIMPLE	red./min	0.049	0.82	0.98	0.998	—
	$T_{0.1}$	7.2e-5	2.2e-4	7.6e-4	1.8e-3	—

Figure 5.5 The efficiencies of the methods for driven cavity Stokes flow with modified Taylor-Hood discretization.



(a) GMG

(b) AMG

(c) AMG, 'artificial' pressure

Table 5.3 The results for driven cavity Stokes flow with hierarchically refined grid and renumbering (→ AMG) and P_1-P_1-stab discretization. (mp. . . memory problems, i.e. the solver ran out of computer memory)

refinement-level		5	6	7	8	9
total number of unknowns on finest level		14,067	55,779	222,147	886,659	3,542,787
Coupled, Braess SSUR 0.8, W-6-6	red./min	0.048	0.55	0.89	0.97	mp
	$T_{0.1}$	5.4e-5	6.9e-5	8.9e-5	8.4e-5	mp
Coupled, Braess Jacobi, W-12-12	red./min	0.084	0.70	0.94	0.99	mp
	$T_{0.1}$	6.6e-5	1.2e-4	1.7e-4	2.0e-4	mp
Coupled, MSM W-15-15	red./min	1.6e-11	0.021	0.56	0.90	0.96
	$T_{0.1}$	6.6e-6	1.1e-5	1.8e-5	2.4e-5	2.6e-5
BiCGstab + Black-Box Prec.	red./min	1.0e-100	2.3e-10	0.0016	0.17	0.67
	$T_{0.1}$	7.1e-7	1.9e-6	1.6e-6	1.5e-5	1.6e-6
AMG-SIMPLE	red./min	0.089	0.88	0.99	1.0	—
	$T_{0.1}$	6.8e-5	3.1e-4	1.3e-3	2.7e-3	—

Figure 5.6 The efficiencies of the (AMG) methods for driven cavity Stokes flow with P_1-P_1-stab discretization.

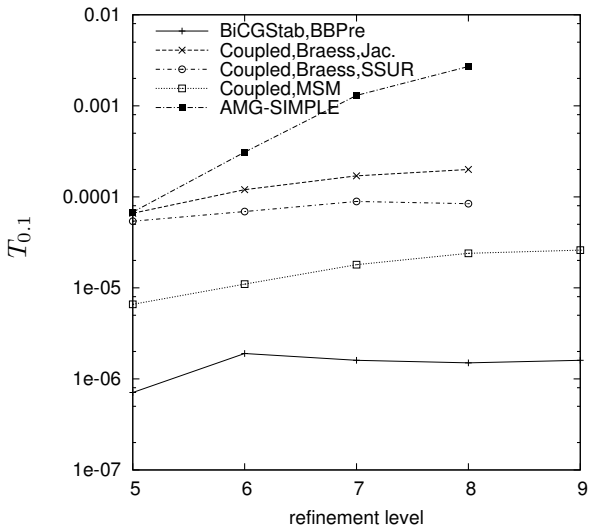


Table 5.4 Dependence of the efficiency of the methods on varying strength of convection for the two-dimensional valve problem.

ν		1	0.1	3.2e-3	1e-3	7e-4	5e-4	4.2e-4
Coupled, Braess	red./min	3.8e-8	1.4e-3	1.9e-3	5.1e-3	0.019	0.06	0.087
SSUR 0.8, W-4-4	$T_{0.1}$	1.3e-6	3.4e-6	3.6e-6	4.2e-6	5.6e-6	7.9e-6	9.1e-6
Coupled, Braess	red./min	2.8e-5	0.19	0.19	0.21	0.28	0.40	0.48
Jacobi, W-10-10	$T_{0.1}$	2.1e-6	1.3e-5	1.3e-5	1.4e-5	1.8e-5	2.4e-5	3.0e-5
Coupled, MSM	red./min	3.1e-6	1.8e-5	0.080	0.042	5.8e-4	4.7e-4	1.4e-3
W-8-8	$T_{0.1}$	1.8e-6	2.0e-6	8.8e-6	7.0e-6	3.0e-6	2.9e-6	3.4e-6
BiCGstab+	red./min	8.4e-7	3.1e-4	8.3e-4	4.8e-4	2.8e-3	0.023	0.042
Black-Box Prec.	$T_{0.1}$	1.6e-6	2.8e-6	3.1e-6	2.9e-6	3.8e-6	5.9e-6	7.0e-6
AMG-SIMPLE	red./min	0.51	0.51	0.39	0.38	0.38	0.37	0.39
	$T_{0.1}$	3.3e-5	3.3e-5	2.4e-5	2.3e-5	2.3e-5	2.2e-5	2.4e-5

convection speed $\mathbf{w}$ near the solution of the Navier-Stokes problem with given ν (therefore the dependence of the linear problem on ν is twofold, via ν itself and via $\mathbf{w}(\nu)$), and we stop the linear iteration after a reduction of the residual by a factor of 10^{-3}.

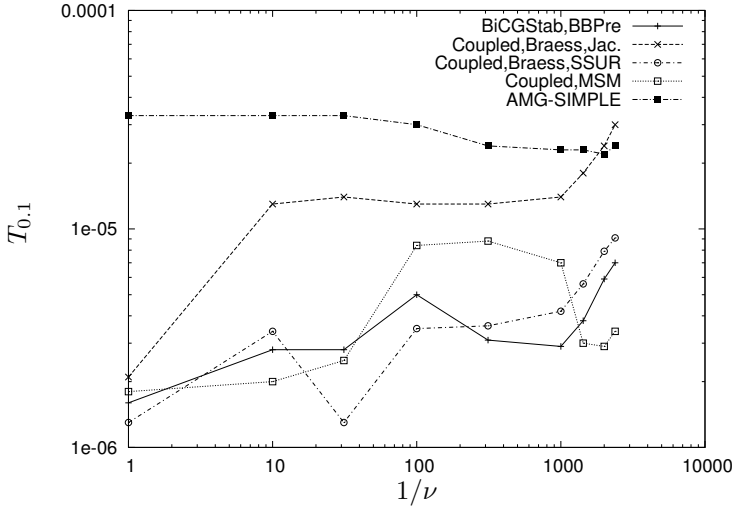
The first geometry here is the two-dimensional valve, it is discretized with the modified Taylor-Hood element with 103,351 unknowns (in total) on the discretization level. The results can be found in Figure 5.7(a) and Table 5.4. In Figure 5.8 we plot residuals vs. CPU-time for the nonlinear iteration for $\nu = 8 \cdot 10^{-4}$, where we use different methods for the solution of the linear problems, and the linear iterations are stopped after a reduction of the residual by a factor of 10^{-2}. There, we also put a comparison of the Oseen iterations and this variant of Newton's method, where the linear problems are solved by (in this case three steps of) a "Oseen-preconditioned" Richardson iteration, as suggested in Section 2.4.2.

The same test (Oseen problem, fixed h, varying ν and $\mathbf{w}(\nu)$) was carried out for the driven cavity problem, again with modified Taylor-Hood discretization and 166,691 unknowns. The dependence on ν is plotted in Figure 5.7(b).

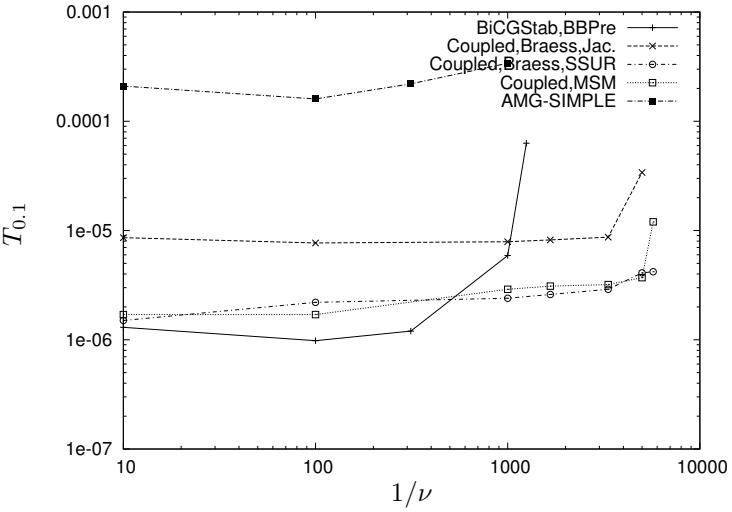
We see that for moderate convection again the "black-box" preconditioner performs well, although it is not as cheap as in the Stokes case (an additional pressure-Laplace-AMG has to be performed). For smaller ν the coupled method (especially with the local smoother) is preferable (at least in these examples). It is remarkable that the AMG-SIMPLE method is robust in ν over a large interval, in the valve-example its rates start to get worse only shortly before the whole nonlinear iteration breaks down. In the driven cavity example the AMG method (with red-black coarsening and simple averaging as interpolation) for the A-problem was divergent for $\nu < 8 \cdot 10^{-4}$, thus we have no results for the AMG-SIMPLE method and the "black-box" preconditioner for smaller ν.

The comparison of the Oseen iteration with the "Oseen-preconditioned Richardson"-Newton iteration gives no clear result. Asymptotically the second method performs better — the asymptotic reduction of the residual per minute is 0.42 compared to 0.54 for the

Figure 5.7 $T_{0.1}$ for the Oseen problem on one fixed grid with different ν and with $\mathbf{w}$ near the solution of the Navier-Stokes problem, for different methods, for the 2D valve and the driven cavity problem.



(a) Valve



(b) Driven cavity

Figure 5.8 Residual-histories for the nonlinear iteration. The linear iterations were solved with the different methods until a mild reduction (factor 10^{-2}) of the (linear) residual was reached. The second BiCGstab result ("hi.-pr") was reached with a stronger reduction (factor 10^{-3}) of the (linear) residual.

The second picture compares the Oseen iterations ("Coupled,MSM" from the first picture) with the Newton method, where the linear problem is solved with Oseen-preconditioned Richardson iterations.

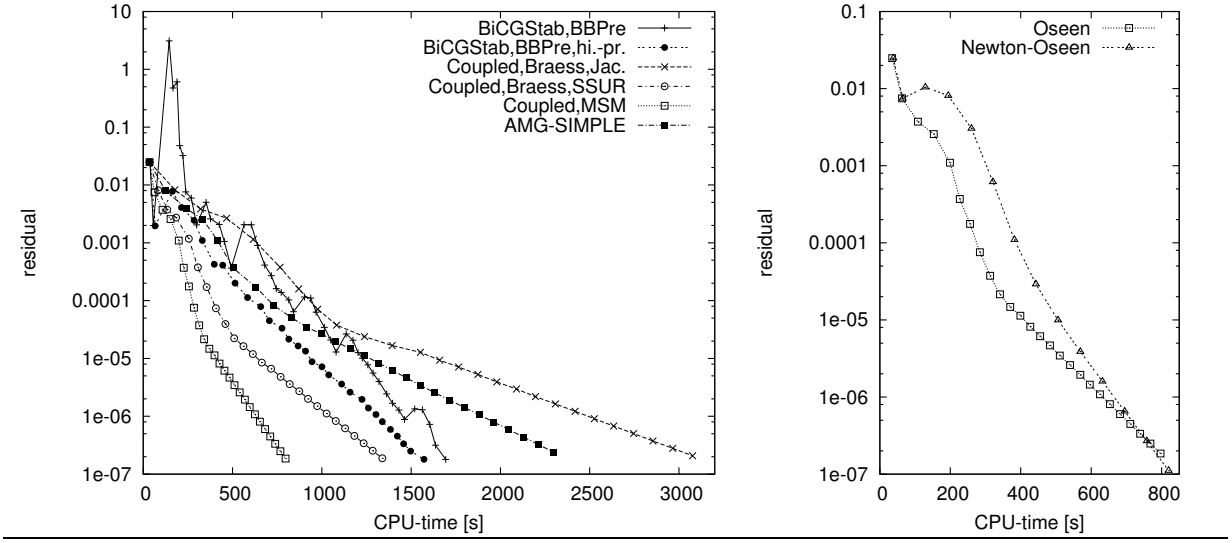


Table 5.5 Different methods for the coarse level construction in the coupled method (with local smoother) for the P_1-P_1-stab discretized valve problem. "Ruge-Stüben splitting" means the modification (3.21) of the red-black algorithm, "Ruge-Stüben interpolation" the method (3.14).

	asympt. reduction/step	red./min	$T_{0.1}$
system matrix, red-black splitting, averaging	0.24	0.00286	1.1e-5
system matrix, Ruge-Stüben splitting, averaging	0.13	0.127	3.2e-5
system matrix, Ruge-Stüben splitting and interpolation	0.18	0.0453	2.1e-5
distance matrix, Ruge-Stüben splitting, averaging	0.15	0.0475	2.2e-5
distance matrix, Ruge-Stüben splitting and interpolation	0.13	0.00975	1.4e-5

Oseen iteration. But in practice one is not interested in a solution of the problem up to a very strong reduction of the residual, the first three or four powers of ten are more important, and in this initial phase the standard Oseen iteration is faster.

Influence of C/F-Splitting and Interpolation

In Section 3.2.2.1 we have presented different possibilities for the C/F-splitting and for the prolongator. In Table 5.5 we compare the standard red-black coloring Algorithm 3.10, the modification (3.21), and the interpolation by averaging and (3.14), all applied to the system matrix and to the distance matrix (3.1). Theses methods are used for the coarse level construction of a coupled AMG method with the local smoother (W-15-15) for the Oseen problem ($\nu = 5 \cdot 10^{-4}$) on the P_1-P_1-stab discretized valve (with 34,863 unknowns).

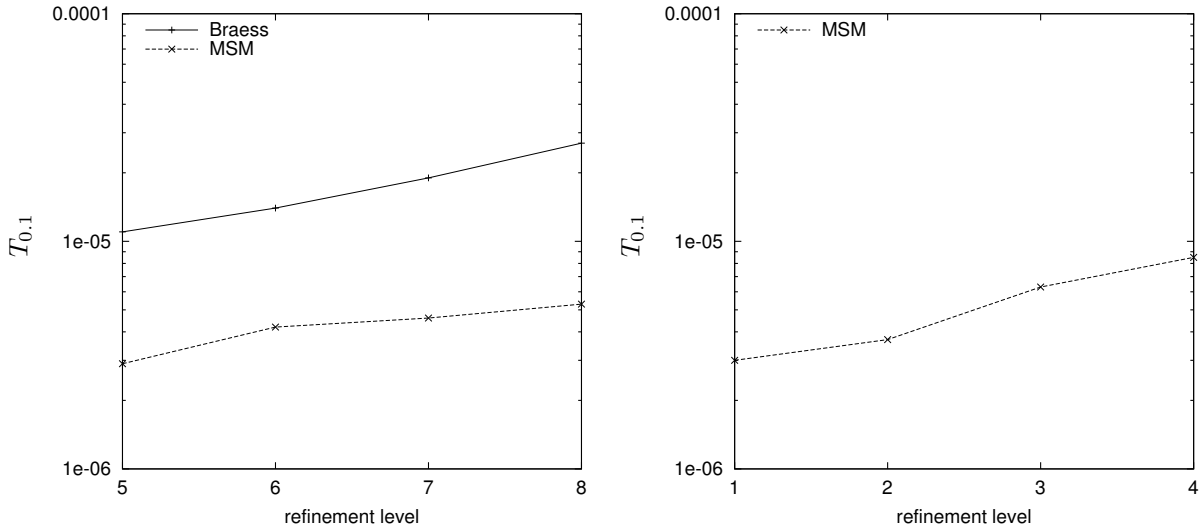
It seems clear that the advanced methods result in better convergence rates than the simple red-black coloring with averaging. What is a bit surprising at first glance is that the situation is the other way round when we look at the efficiencies. The reason thereof are slightly denser coarse level matrices generated by the modified red-black splitting, but this could be repaired in various ways (some can be found e.g. in [Stü01a]). Thus, the better rates indicate that some more thoughts in this direction could pay off.

AMGe for the Crouzeix-Raviart Element

AMuSE is not yet capable of solving Crouzeix-Raviart-discretized problems in any other way than with the coupled AMGe method, presented in Section 4.1.3. Therefore, we are not able to present comparisons of different solvers, but only to show how the method

Figure 5.9 The agglomerates of one coarse level of the valve problem.

Figure 5.10 The efficiency of the coupled AMGe method with Braess-Sarazin and local smoother for different levels of refinement. The figure on the left comes from the driven cavity problem, on the right from the 2D valve.



performs for the two test problems. There are also no 3D tests for AMGe in this thesis, as our experiences there are still at a very basic level.

In Figure 5.9 we illustrate the formation of agglomerates on a coarse level for the valve problem. The results for the solution of the driven cavity and the valve problem, both with Stokes flow, for increasingly finer grids can be found in Table 5.6 and Figure 5.10. For both smoothers we applied, the level dependence of our version of the AMGe method is surely improvable, especially the Braess-Sarazin smoother seems to barely fit to the rest of the algorithm. The local smoother performs all right for the driven cavity problem, but also deteriorates for the valve. What we have observed is, that up to a certain number of coarse levels the method behaves nicely and then suddenly gets worse (note for example the jump in the convergence rates of the valve problem from two to three refinement levels), which we assume is related to ill shaped agglomerates on the coarser levels. Thus, we think that improvement is possible if a more sophisticated agglomeration algorithm is used.

Table 5.6 The efficiency of the coupled method with Braess-Sarazin and local smoother for different levels of refinement.

Driven Cavity					
refinement-level		5	6	7	8
total number of unknowns on finest level		37,024	147,776	590,464	2,360,576
Coupled, Braess SSUR 0.8, W-6-6	asympt.red./step	0.39	0.45	0.43	0.52
	$T_{0.1}$	1.1e-5	1.4e-5	1.9e-5	2.7e-5
Coupled, MSM W-10-10	asympt.red./step	0.11	0.26	0.23	0.29
	$T_{0.1}$	2.9e-6	4.2e-6	4.6e-6	5.3e-6
2D Valve					
refinement-level		1	2	3	4
total number of unknowns on finest level		22,932	91,112	363,216	1,450,400
Coupled, MSM W-10-10	asympt.red./step	0.046	0.098	0.30	0.40
	$T_{0.1}$	3.0e-6	3.7e-6	6.3e-6	8.5e-6

5.2.2 3D Problems

The two three-dimensional geometries we use in this thesis are the following:

- **3D valves.** In Figure 5.11 we show this geometry with two valves. We prescribe a velocity of 0.5 at the inlets, the distance between the walls at the narrowest part is 0.03. The mesh was generated with Joachim Schöberl's 'netgen' [Sch97].

- For the so called **rotax** (illustrated in Figure 5.12) a multi-element mesh is used, which was provided by the AVL List GmbH and which consists of 302 tetrahedra, 142,339 hexahedra, 5095 pyramids, and 10019 prisms with triangular basis. Thus, we apply the strategy explained in Section 2.2.2.

  We prescribe a velocity of 0.05 at the inlets, the outlets have a diameter of 0.045.

Modified Taylor-Hood Element

We want to solve the problem obtained by a modified Taylor-Hood discretization of the valve, which has a total number of unknowns of 2,092,418. For the coupled method with Braess smoother and standard, single-shifted red-black coarsening we observe a poor performance (see Figure 5.13) as neither for the Stokes problem, nor for the Oseen problem it is clearly faster than the AMG-SIMPLE method. If we use the local smoother the situation is even worse, it is hardly possible to obtain a converging method with a reasonable number of smoothing steps. Here the stability problems mentioned in Section 4.1.1 strike, therefore we apply the 2-shift strategy and get slightly better (at least better than with AMG-SIMPLE) but not satisfactory results.

Figure 5.11 The 3D valve problem. The fluid enters at the two inlets on the right, passes the valves, enters the wider area and leaves it again via two holes on the left (which can not be recognized in this picture).

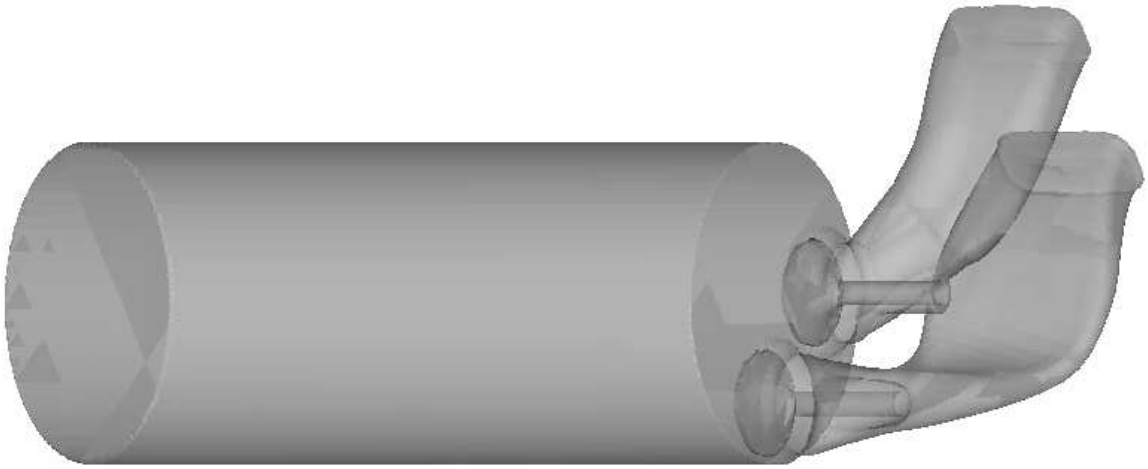


Figure 5.12 The so called rotax. It has two inlet ports in an angle of $\pi/2$ and two smaller outlets. The lower part of the figure provides a view inside the geometry, with a viewpoint indicated by the arrow.

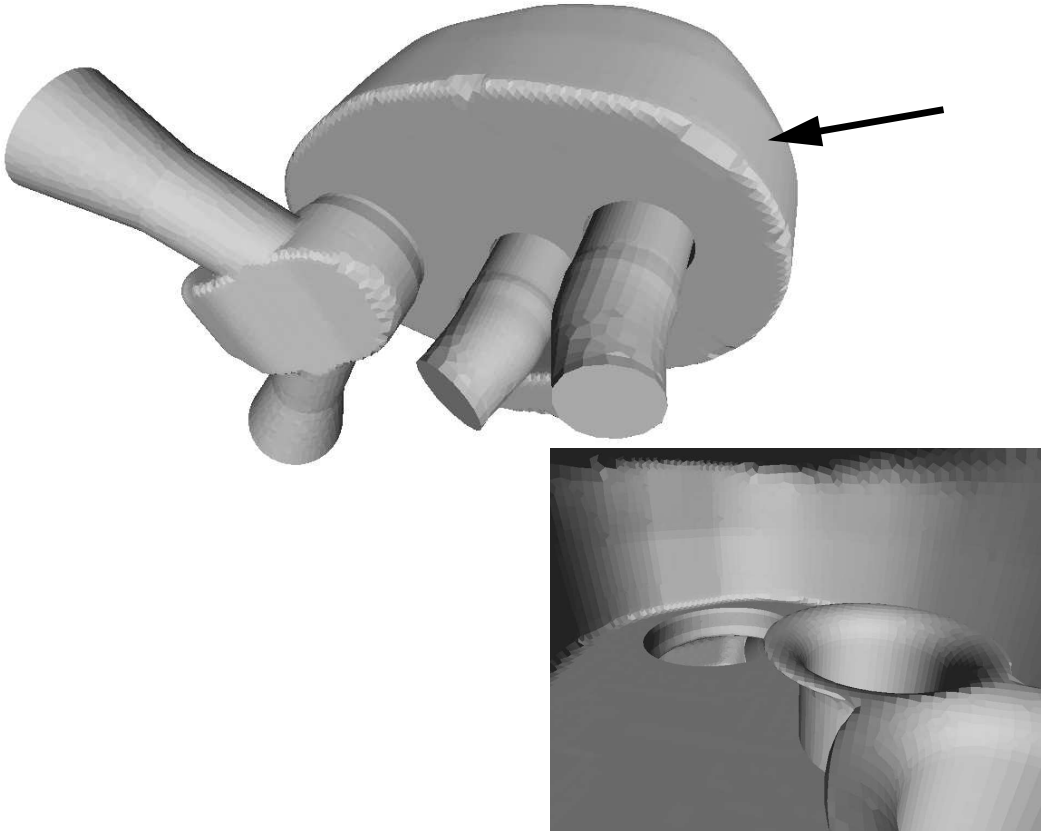
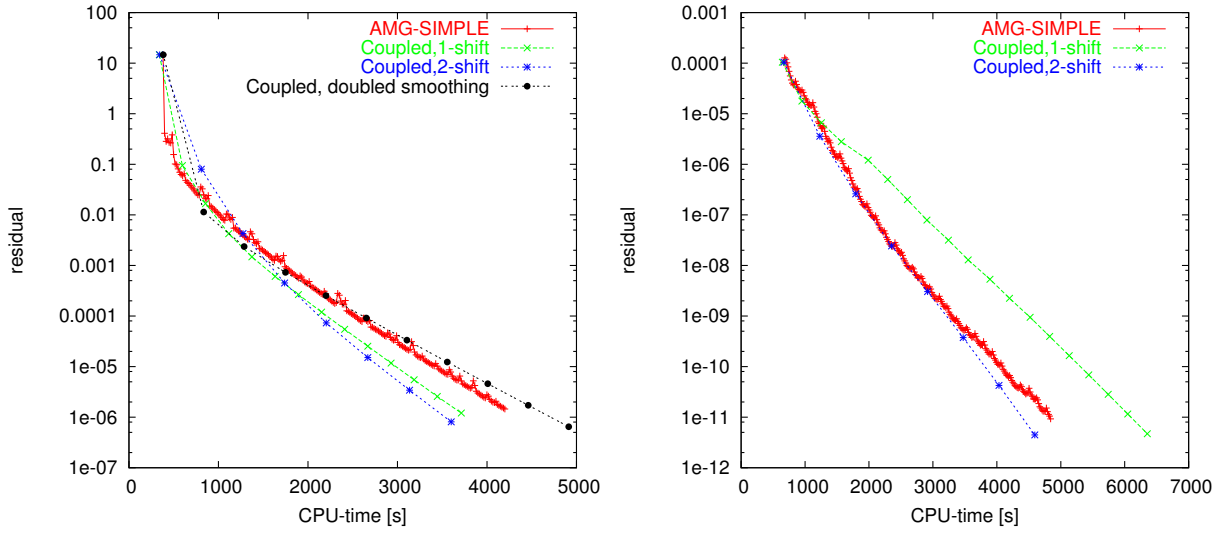


Figure 5.13 Convergence behavior of the Stokes (on the left) and the Oseen problem with $\nu = 10^{-3}$ (on the right) for the P_1isoP_2-P_1 discretized 3D valves.



One could suppose that the better performance of the 2-shift strategy originates in the increase of smoothing steps on the fine (velocity-) level. Therefore we have included the results for the 1-shift method with a doubled number of smoothing steps on the finest level, to show that this is not the case.

An illustration of the solution of the Navier-Stokes problem can be found in Figure 5.14.

P_1-P_1-stab Element

As indicated by the results of Section 4.1.2 the results for the P_1-P_1-stab discretization are more promising. We use it on the rotax problem and get a total number of 658,528 (visible) unknowns. Because of the strategy of Section 2.2.2 our problem has implicitly more degrees of freedom, but they are locally eliminated (which results in a denser matrix). For example for a hexahedron we need seven auxiliary nodes which nearly doubles the number of the eight 'real' nodes.

In Figure 5.15 and Table 5.8 we compare the efficiencies of the AMG-SIMPLE method and the coupled AMG method (with Braess-Sarazin smoother) and see that in this situation again the coupled method has to be preferred.

In Figure 5.16 we show the solution of the Navier-Stokes problem with $\nu = 5 \cdot 10^{-4}$.

Table 5.7 Efficiencies for the Stokes and the Oseen problem with $\nu = 10^{-3}$ for the $P_1 \text{iso} P_2$-P_1 discretized 3D valves. For the coupled method for the Stokes problem we used the Braess smoother and a W-11-11 cycle, for the Oseen problem a W-14-13 cycle.

Stokes problem

	asympt. red./min.	asympt. $T_{0.1}$
AMG-SIMPLE	0.85	6.9e-6
Coupled, standard red-black	0.84	6.2e-6
Coupled, 2-shift	0.82	5.7e-6
Coupled, doubled number of smooth. steps on finest lev.	0.88	8.3e-6

Oseen problem

	asympt. red./min.	asympt. $T_{0.1}$
AMG-SIMPLE	0.82	5.5e-6
Coupled, standard red-black	0.84	6.4e-6
Coupled, 2-shift	0.79	4.8e-6

Figure 5.14 Pressure distribution on the boundary and flow around the valves for the 3D valve problem with $\nu = 10^{-3}$.

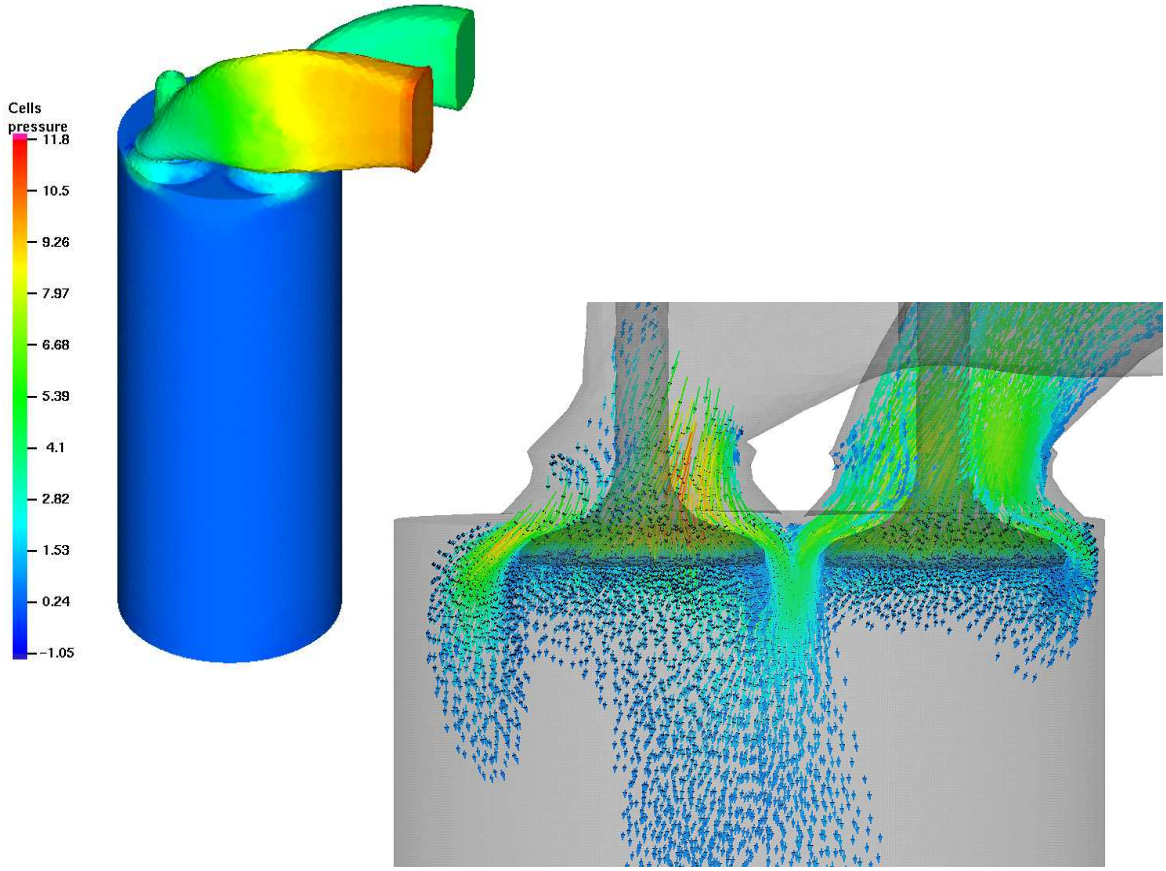


Figure 5.15 Comparison of the AMG-SIMPLE method and the coupled approach for the P_1-P_1-stab discretization of the rotax (Oseen problem, $\nu = 5 \cdot 10^{-4}$).

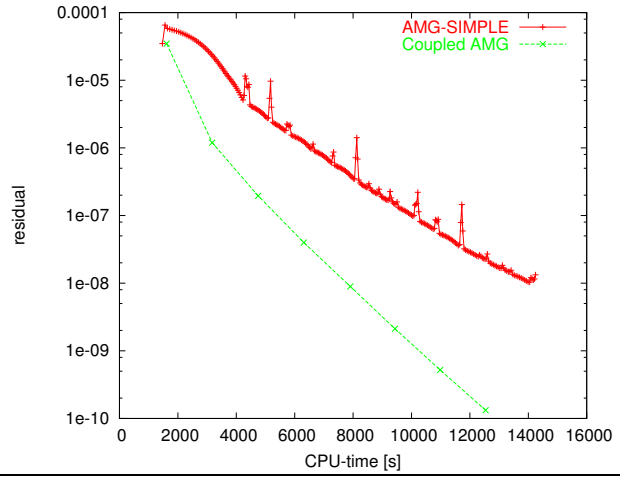


Table 5.8 Efficiencies of the AMG-SIMPLE method and the coupled approach for the P_1-P_1-stab discretization of the rotax (Oseen problem, $\nu = 5 \cdot 10^{-4}$).

	asympt. red./min.	asympt. $T_{0.1}$
AMG-SIMPLE	0.97	1.0e-4
Coupled	0.95	6.4e-4

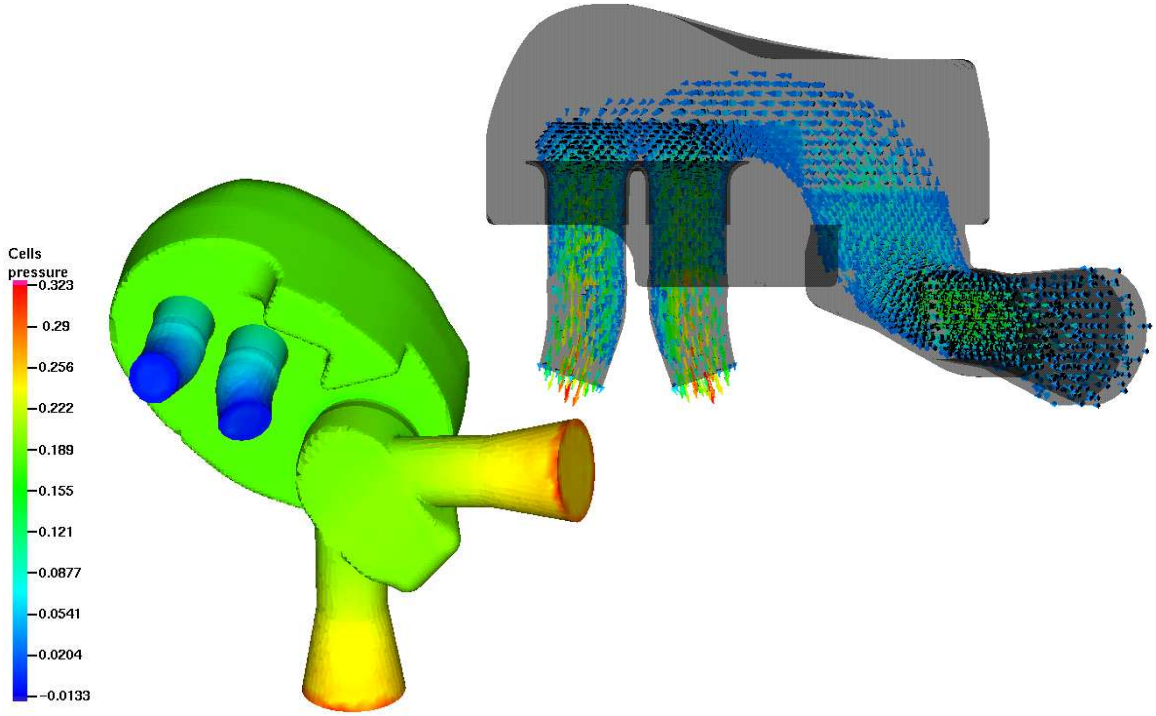
5.3 Conclusions and Outlook

In this thesis we have investigated the application of several possible components for the AMG solution of the saddle point problem arising in the finite element discretization of the Oseen equations.

Our main achievements lie in the development of coupled algebraic multigrid solvers for such problems, to be concrete

- we have found a technique for the construction of coarse grid hierarchies for problems discretized with the *modified Taylor-Hood element*, and we have given some heuristics for their stabilization,
- we have developed an AMG method for the P_1-P_1-*stab element* and have proven that the coarse level systems there are *stable*,
- we have made first steps in the application of *AMGe* to the coupled problem and have shown exemplarily, that if it is used in a certain way for the *Crouzeix-Raviart element*, then again one obtains *stable* coarse level systems,
- we have shown how to apply *smoothers* known from geometric multigrid methods in a *purely algebraic* context, and

Figure 5.16 Pressure on the surface and main flow of the solution of the rotax problem with $\nu = 5 \cdot 10^{-4}$.



- we have developed a *software package* which is capable of most of the techniques described in this thesis (and some more) and which has provided us with the possibility of
- applying the methods to *"real-life" industrial problems* and of
- *comparing* our approach with methods using the *segregated approach* (e.g. AMG-SIMPE or the "black-box" preconditioner).

We have seen that if AMG is applied using the segregated approach, then this has clear advantages with respect to the simplicity of development. One just needs "off the shelf" AMG solvers for elliptic problems and can plug them in some relatively easy to implement method, and that is it.

Most of this "black-box" character is lost for the coupled method on which we have focused, as the behavior of the solver depends for example strongly on the chosen finite element pairing and its stability properties. But, as the numerical experiments indicate, if the coarse grid is constructed carefully and if the smoother fits, then the method can become powerful and can outperform solvers using the segregated approach.

Of course, much more work can be invested in different aspects of these methods, on the practical and on the theoretical side. One question is the possibility of generalizations to arbitrary finite elements. We have shortly discussed this in Chapter 4, but our numerical experiences in this direction are still very rudimentary.

Other questions arise on the analytic side. We have proven (in some cases) the stability of the coarse level system, but what about convergence? What about stability results for a more general class of elements?

If the methods presented here should be used in an industrial context, then some more thought should also be spent on an efficient implementation, on modern programming techniques, on parallelization aspects, etc.

Nevertheless, we have developed a coupled method which can compete with classical approaches, and which has an area of application beyond unit square problems with weak convection.

Bibliography

[AB01] V.A.O. Anjorin and I.E. Barton. Removal of temporal and under-relaxation terms from the pressure-correction equation of the SIMPLE algorithm. *Int. J. Fluid Dynamics*, 5:59–75, 2001.

[Ada03] M.F. Adams. Algebraic multrigrid methods for constrained linear systems with applications to contact problems in solid mechanics. preprint available at `http://www.cs.berkeley.edu/~madams/`, 2003. submitted to Numerical Linear Algebra with Applications.

[AHU58] K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in nonlinear programming.* Stanford University Press, Stanford, CA, 1958.

[Ast71] G.P. Astrachancev. An iterative method for solving elliptic net problems. *USSR Comput. Math. Math. Phys.*, 11(2):171–182, 1971.

[BA72] I. Babuška and A.K. Aziz. Survey lectures on the mathematical foundations of the finite element method. With the collaboration of G. Fix and R.B. Kellogg. In A.K. Aziz, editor, *The Mathematical Foundation of the Finite Element Method with Applications to Partial Differential Equations, Sympos. Univ. Maryland, Baltimore 1972*, pages 3–363. Academic Press, New York - London, 1972.

[Bac66] N.S. Bachvalov. On the convergence of a relaxation methods with natural constraints on the elliptic operator. *USSR Comput. Math. Math. Phys.*, 6(5):101–135, 1966.

[BCF+00] M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteufel, S.F. McCormick, and J.W. Ruge. Algebraic multigrid based on element interpolation (AMGe). *SIAM J. Sci. Comput.*, 22(5):1570–1592, 2000.

[Bec95] R. Becker. An adaptive finite element method for the incompressible Navier-Stokes equations on time-dependent domains. Preprint 95-44, SFB 359, University Heidelberg, 1995.

[Bec02] A. Becirovic. Berechnung von Gebietsableitungen für das Stokes-Problem. Master's thesis, Johannes Kepler University Linz, 2002.

[Ber02] S. Bertling. Ein Algebraisches Mehrgitterverfahren zur Lösung der Stokes-Gleichungen. Master's thesis, RWTH Aachen, 2002.

[BF91] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer series in computational mathematics. Springer, New York, 1991.

[BMR84] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D.J. Evans, editor, *Sparsity and its applications*, pages 257–284. Cambridge University Press, Cambridge, 1984.

[Bra73] A. Brandt. Multi-level adaptive techniques (mlat) for fast numerical solution to boundary value problems. In *Proc. 3rd Internat. Conf. on Numerical Methods in Fluid Mechanics, held in Paris, 1972*, Lecture Notes in Physics, 1973.

[Bra77] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, 31:333–390, 1977.

[Bra86] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comp.*, 19:23–56, 1986.

[Bra93] J.H. Bramble. *Multigrid methods*. Longman Scientific & Technical, Harlow, 1993.

[Bra97] D. Braess. *Finite Elemente*. Springer, Berlin, Heidelberg, New York, 2nd edition, 1997.

[Bre93] S.C. Brenner. A nonconforming mixed multigrid method for the pure displacement problem in planar linear elasticity. *SIAM J. Numer. Anal.*, 30(1):116–135, 1993.

[BS97] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23:3–20, 1997.

[Bun88] H. Bungartz. Algebraische Mehrgitterverfahren zur Optimierung unter Nebenbedingungen. Master's thesis, TU Munich, 1988.

[BV90] D. Braess and R. Verfürth. Multigrid methods for nonconforming finite element methods. *SIAM J. Numer. Anal.*, 27(4):979–986, 1990.

[BW97] J. Bey and G. Wittum. Downwind numbering: Robust multigrid for convection-diffusion problems. *Appl. Numer. Math.*, 23(1):177–192, 1997.

[BWY90] R.E. Bank, B.D. Welfert, and H. Yserentant. A class of iterative methods for solving saddle point problems. *Num. Math.*, 56:645–666, 1990.

[CR73] M. Crouzeix and P.A. Raviart. Conforming and nonconforming finite element methods for solving the stationary Stokes equations I. *RAIRO Anal. Numér.*, 7:33–76, 1973.

[Dic00] D. Dickson. Mathematicians chase the seven million-dollar proofs. *Nature*, 405:383, 25 May 2000.

[DMS01] W. Dahmen, S. Müller, and T. Schlinkmann. On an adaptive multigrid solver for convection-dominated problems. *SIAM J. Sci. Comput.*, 23(3):781–804, 2001.

[ESW02] H.C. Elman, D.J. Silvester, and A.J. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numer. Math.*, 90:665–688, 2002.

[Fed61] R.P. Federenko. A relaxation method for elliptic difference equations. *USSR Comput. Math. Math. Phys.*, 1(5):1092–1069, 1961.

[Fef00] C.L. Fefferman. Existence & smoothness of the Navier-Stokes equation. available at `http://www.claymath.org`, 2000. Official problem description of the "Millenium Price Problem".

[Fei93] M. Feistauer. *Mathematical Methods in Fluid Dynamics*. Number 67 in Pitman Monographs and Surveys in Pure and Applied Mathematics. Longman Scientific & Technical, London, 1993.

[FHS93] L.P. Franca, T.J.R. Hughes, and R. Stenberg. Stabilized finite element methods for the Stokes problem. In M. Gunzburger and R.A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, chapter 4, pages 87–107. Cambridge University Press, 1993.

[FM93] L.P. Franca and A.L. Madureira. Element diameter free stability parameters for stabilized methods applied to fluids. *Comp. Meth. Appl. Mech. Eng.*, 105:395–403, 1993.

[FS91] L.P. Franca and R. Stenberg. Error analysis of some Galerkin least squares methods for the elasticity equations. *SIAM J. Numer. Anal.*, 28(6):1680–1697, 1991.

[GNR98] M. Griebel, T. Neunhoeffer, and H. Regler. Algebraic multigrid methods for the solution of the Navier-Stokes equations in complicated geometries. *Int. J. Numer. Methods Fluids*, 26:281–301, 1998.

[GR86] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer, Berlin, Heidelberg, 1986.

[Hac76] W. Hackbusch. Implementation of the multi-grid method for solving partial differential equations. Technical report RA 82, IBM T.J. Watson Research Center, 1976.

[Hac85] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, Berlin, 1985.

[Hac93] W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, Stuttgart, 2nd edition, 1993.

[Him02] J. Himmelbauer. Numerische Lösung der Navier-Stokes-Gleichung bei zeitabhängiger Geometrie. Master's thesis, Johannes Kepler University Linz, 2002.

[HV01] V.E. Henson and P.S. Vassilevski. Element-free AMGe: General algorithms for computing interpolation weights in AMG. *SIAM J. Sci. Comput.*, 23(2):629–650, 2001.

[HW02] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer, Heidelberg, 2nd, rev. edition, 2002.

[HWN00] E. Hairer, G. Wanner, and S.P. Nørsett. *Solving Ordinary Differential Equations I, Nonstiff Problems*. Springer, Heidelberg, 2nd, rev. edition, 2000.

[Ips01] I.C.F. Ipsen. A note on preconditioning nonsymmetric matrices. *SIAM J. Sci. Comput.*, 23(3):1050–1051, 2001.

[IRT93] J. Iliash, T. Rossi, and J. Toivanen. Two iterative methods for solving the Stokes problem. Report 2/1993, Department of Mathematics, University of Jyväskylä, 1993.

[JKMT02] V. John, P. Knobloch, G. Matthies, and L. Tobiska. Non-nested multilevel solvers for finite element discretizations of mixed problems. *Computing*, 68:313–341, 2002.

[JV01] J.E. Jones and P.S. Vassilevski. AMGe based on element agglomeration. *SIAM J. Sci. Comput.*, 23(1):109–133, 2001.

[JVW02] J.E. Jones, P.S. Vassilevski, and C.S. Woodward. Nonlinear Schwarz-FAS methods for unstructured finite element problems. Technical Report UCRL-JC-150427, Lawrence Livermore National Laboratory, 2002. submitted to the Second M.I.T. Conference on Computational Fluid and Solid Mechanics, held in Cambridge, MA, June 17–20, 2003.

[Kic96] F. Kickinger. Algebraic multigrid for discrete elliptic second-order problems — a program description. Technical Report No. 96-5, Department of Mathematics, Johannes Kepler University, Linz, 1996.

[Kic97a] F. Kickinger. Automatic microscaling mesh generation. Institusbericht No. 525, Department of Mathematics, Johannes Kepler University, Linz, 1997.

[Kic97b] F. Kickinger. Programmbeschreibung NAOMI 2D und Algebraic Multigrid. Technical Report No. 97-4, Department of Mathematics, Johannes Kepler University, Linz, 1997.

[Kic98] F. Kickinger. Algebraic multigrid for discrete elliptic second-order problems. In W. Hackbusch, G. Wittum, and R. Nieminen, editors, *Multigrid Methods V, Proceedings of the 5th European Multigrid conference, held in Stuttgart, Germany, 1996*, volume 3 of *Lect. Notes Comput. Sci. Eng.*, pages 157–172, Berlin, 1998. Springer.

[Kor77] V.G. Korneev. *Finite element schemes of higher order of accuracy (in Russian).* Leningrad University Press, Leningrad, 1977.

[Lan82] U. Langer. On the choice of iterative parameters in the relaxation method on a sequence of meshes. *USSR Comput. Math. Math. Phys.*, 22(5):98–114, 1982.

[LQ86] U. Langer and W. Queck. On the convergence factor of Uzawa's algorithm. *J. Comput. Appl. Math.*, 15:191–202, 1986.

[LQ87] U. Langer and W. Queck. Preconditioned Uzawa-type iterative methods for solving mixed finite element equations. Wissenschaftliche Schriftenreihe der Technischen Universität Karl-Marx-Stadt, Karl-Marx-Stadt (Chemnitz), 1987.

[LW02] D. Loghin and A.J. Wathen. Schur complement preconditioning for elliptic systems of partial differential equations. Technical Report 01/20, Oxford University Computing Laboratory, Numerical Analysis Group, 2002.

[Meu99] G. Meurant. *Computer solution of large linear systems*, volume 28 of *Studies in Mathematics and its Applications*. Elsevier, Amsterdam, 1999.

[MGW00] M.F. Murphy, G.H. Golub, and A.J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, 21(6):1969–1972, 2000.

[MP94] B. Mohammadi and O. Pironneau. *Analysis of the K-Epsilon TURBULENCE MODEL.* John Wiley & Sons, Chichester, 1994.

[OGWW98] C.W. Oosterlee, F.J. Gaspar, T. Washio, and R. Wienands. Fast multigrid solvers for higher order upwind discretizations of convection-dominated problems. In W. Hackbusch, G. Wittum, and R. Nieminen, editors, *Multigrid methods V. Proceedings of the 5th European multigrid conference, held in Stuttgart, Germany, 1996*, volume 3 of *Lect. Notes Comput. Sci. Eng.*, pages 212–224, Berlin, 1998. Springer.

[Pat80] S. Patankar. *Numerical heat transfer and fluid flow*. Series in Computational Methods in Mechanics and Thermal Sciences. McGraw-Hill, New York, 1980.

[Pir89] O. Pironneau. *Finite Element Methods for Fluids*. John Wiley & Sons, Chichester, 1989.

[Pönng] A. Pönitz. Shape optimization for the Navier-Stokes equations. Master's thesis, Johannes Kepler University Linz, ongoing.

[PS72] S. Patankar and D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15:1787–1806, 1972.

[PTV00] S.I. Petrova, L. Tobiska, and P.S. Vassilevski. Multigrid methods based on matrix-dependent coarse spaces for nonconforming streamline-diffusion finite element discretization of convection-diffusion problems. *East-West J. Numer. Math.*, 8(3):223–242, 2000.

[Ran00] R. Rannacher. Finite Element Methods for the Incompressible Navier-Stokes Equations. In Giovanni Galdi et al., editors, *Fundamental directions in mathematical fluid mechanics*, pages 191–293. Birkhäuser, Basel, 2000.

[Raw95] M.J. Raw. A coupled algebraic multigrid method for the 3D Navier-Stokes equations. In W. Hackbusch and G. Wittum, editors, *Fast solvers for flow problems. Proceedings of the tenth GAMM-Seminar, held in Kiel, Germany, 1994*, volume 49 of *Notes Numer. Fluid Mech.*, pages 204–215, Wiesbaden, 1995. Vieweg.

[Rei01] S. Reitzinger. *Algebraic Multigrid Methods for Large Scale Finite Element Equations*. PhD thesis, Johannes Kepler University, Linz, 2001.

[Reu95] A. Reusken. A new robust multigrid method for 2D convection diffusion problems. In W. Hackbusch and G. Wittum, editors, *Fast solvers for flow problems. Proceedings of the tenth GAMM-Seminar, held in Kiel, Germany, 1994*, volume 49 of *Notes Numer. Fluid Mech.*, pages 229–240, Wiesbaden, 1995. Vieweg.

[Reu02] A. Reusken. Convergence analysis of a multigrid method for convection-diffusion equations. *Numer. Math.*, 91(2):323–349, 2002.

[RS86] J.W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. McCormick, editor, *Multigrid Methods*, volume 5 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, 1986.

[RW92] T. Rusten and R. Winther. A preconditioned iterative method for saddlepoint problems. *SIAM J. Matrix Anal. Appl.*, 13(3):887–904, 1992.

[RW99] C. Reisinger and M. Wabro. Analysis and numerical solution of the k-ϵ turbulence model with non-standard boundary conditions. Master's thesis, Johannes Kepler University Linz, 1999.

[Sch97] J. Schöberl. Netgen — an advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997. Netgen homepage: `www.hpfem.jku.at/netgen/`.

[SEKW01] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *Journal of Computational and Applied Mathematics*, 128:261–279, 2001.

[SS86] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[Str97] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, Boston, 3rd edition, 1997.

[Stü01a] K. Stüben. An introduction to algebraic multigrid. In U. Trottenberg, C. Oosterlee, and A. Schüller, editors, *Multigrid*, pages 413–532. Academic Press, New York - London, 2001.

[Stü01b] K. Stüben. A review of algebraic multigrid. *Computational and Applied Mathematics*, 128:281–309, 2001.

[SVdV94] G.L.G. Sleijpen and H.A. Van der Vorst. Maintaining convergence properties of BiCGstab methods in finite precision arithmetic. Preprint 861, Department of Mathematics, University Utrecht, 1994.

[SW94] D.J. Silvester and A.J. Wathen. Fast iterative solution of stabilised Stokes systems. II. Using general block preconditioners. *SIAM J. Num. Anal.*, 31:1352–1367, 1994.

[SZ03] J. Schöberl and W. Zulehner. On Schwarz-type smoothers for saddle point problems. *Numer. Math.*, 95(2):377–399, 2003.

[TOS01] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, New York - London, 2001.

[Tur99] S. Turek. *Efficient Solvers for Incompressible Flow Problems*. Lecture Notes in Computational Science and Engineering. Springer, Heidelberg, 1999.

[Van86] S. Vanka. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Comp. Meth. Appl. Mech. Eng.*, 59(1):29–48, 1986.

[VdV92] H.A. Van der Vorst. Bi-CGstab: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, 1992.

[Ver84a] R. Verfürth. Error estimates for a mixed finite element approximation of the Stokes equations. *R.A.I.R.O. Num. Anal.*, 18(2):175–182, 1984.

[Ver84b] R. Verfürth. A multilevel algorithm for mixed problems. *SIAM J. Numer. Anal.*, 21:264–271, 1984.

[Vos93] H. Voss. Iterative methods for linear systems of equations. Textbook of the 3rd International Summerschool Jyväskylä, Finland. Hamburger Beiträge zur Angewandten Mathematik, Reihe B: Berichte, Hamburg, 1993.

[Web94] R. Webster. An algebraic multigrid solver for Navier-Stokes problems. *Int. J. Numer. Meth. Fluids*, 18:761–780, 1994.

[Wes92] P. Wesseling. *An introduction to multigrid methods.* John Wiley & Sons Ltd., Chichester, 1992.

[Wit89] G. Wittum. Multi-grid methods for Stokes and Navier-Stokes equations. *Numer. Math.*, 54:543–563, 1989.

[Wit90] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. *Numer. Math.*, 57:15–38, 1990.

[Yos80] K. Yosida. *Functional analysis.* Springer, Berlin-Heidelberg-New York, 6th edition, 1980.

[YVB98] I. Yavneh, C.H. Venner, and A. Brandt. Fast multigrid solution of the advection problem with closed characteristics. *SIAM J. Sci. Comput.*, 19(1):111–125, 1998.

[Zul00] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.

[Zul02] W. Zulehner. Analysis of iterative methods for saddle point problems: a unified approach. *Mathematics of Computation*, 71(238):479–505, 2002.

Index

Lebenslauf

Dipl.-Ing. Markus Wabro
Am Nordsaum 29
A-4050 Traun

Geboren: 24.11.1975 in Linz, Österreich
Eltern: Hermann Wabro, Adelheid Puchner, geborene Schabetsberger
Bruder: Michael Wabro
Verheiratet mit Karolin Wabro, geborene Hagenbuchner

Ausbildung

1982 – 1986:	Besuch der Volksschule 18 in Linz.
1986 – 1994:	Besuch des Bundesrealgymnasiums Linz, Landwiedstraße.
06.1994:	Ablegung der Matura mit Auszeichnung.
1994–1999:	Studium der Technischen Mathematik an der Johannes Kepler Universität in Linz.
06.1999:	Ablegung der Diplomprüfung mit Auszeichnung.

Berufliche Laufbahn

1994 – 1999:	Abhaltung diverser Kurse in Mathematik bzw. EDV an der *Volkshochschule*, dem *Wirtschaftsförderungsinsitut*, dem *Arbeitsmarktservice* und dem *Studienkreis*.
07.1999 – 05.2000:	Mitarbeiter an der *J.K. Universität* in Linz, an der Abteilung für Numerische Mathematik und Optimierung, im FFF-Projekt "Automatischer Multi-Element Netzgenerator und Strömungssimulator AUTOGEN".
06.2000 – 05.2001:	Zivildienst beim *Roten Kreuz* in Traun.
06.2001 – 09.2001:	Mitarbeiter an der *J.K. Universität* in Linz, am Institut für Numerische Mathematik. Arbeit an einem Industrieprojekt in Kooperation mit der AVL List GmbH, Graz.
seit 10.2001:	Mitarbeiter an der *J.K. Universität*, Institut für Numerische Mathematik. Arbeit am FWF-Projekt "Robust Algebraic Multigrid Methods and their Parallelization".