A numerical study of iterative methods for the solution of convection-diffusion problems

Jürgen Witzel*

Abstract

We consider different iterative methods (Krylov subspace algorithms, multigrid methods) for the solution of linear systems which arise from a nonconforming Petrov-Galerkin discretization of convection-diffusion problems. This discretization technique is based on inhomogeneous exponentially fitted splines. A numerical examination and comparison of these solution methods gives an answer to the question which methods may be preferred in practice.

AMS Subject Classification: 65F10, 65F50, 65M20, 65M55, 76D30

Key words: convection-diffusion equation, singularly perturbed problems, Krylov subspace methods, multigrid methods, preconditioning.

1 Introduction

The aim of this article is to present an overview of iterative solution techniques for linear systems which arise from a new Petrov-Galerkin discretization of convection-diffusion problems and to give an answer to the fundamental question which method is to be preferred.

First we derive a discretization method for instationary convection-diffusion problems which is based on the method of lines, i.e. a nonconforming Petrov-Galerkin discretization in space followed by a time integration. The main aspect of the Petrov-Galerkin discretization is that we use exponentially fitted trial and test functions.

The algebraic systems of our discretization are nonsymmetric positive definite. Thus we have to use solution methods which are well-suited for such systems. We give an overview on current Krylov subspace methods and present some preconditioning techniques. Subsequently we derive a special multigrid method, too. Since Krylov subspace methods only work well when they are equipped with a good preconditioning technique we use incomplete LU decompositions and also the multigrid method as preconditioners.

The intention of this work is not to present results of quantitative character which means that we did not measure CPU-time, for instance. Since our aim is to judge the different techniques on a qualitative basis we compare the residuals during the course of our computations, mostly.

*Technische Universität Darmstadt, Fachbereich Mathematik, AG Numerische Mathematik, Schloßgartenstr. 7, 64289 Darmstadt, GERMANY, e-mail: witzel@mathematik.tu-darmstadt.de

2 The convection-diffusion problem

Convection-diffusion problems appear in several fields of applications. They are of particular interest in fluids dynamics where they serve as model problems for the derivation of new discretization techniques [28]. If the diffusion coefficient tends to zero the solutions of convection-diffusion problems which are formally of parabolic type behave more and more hyperbolically. On the other hand the numerical methods for the treatment of parabolic resp. hyperbolic problems differ very much. Consequently there is a significant demand for robust discretization techniques which work well for all values of the diffusion parameter.

Here we consider the unsteady linear convection-diffusion equation of the form

$$u_t + Lu = q \quad \text{with} \quad Lu = \boldsymbol{\nabla} \cdot (\boldsymbol{\beta} u - \epsilon \boldsymbol{\nabla} u) \tag{1}$$

on a bounded domain $\Omega \subseteq \mathbb{R}^2$ with constant diffusion coefficient $\epsilon > 0$ and a constant velocity field $\boldsymbol{\beta}$ in $\mathbb{R}^2$. The initial and boundary conditions are given by

$$\begin{aligned} u(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}) && \text{on} \quad \Omega \\ u(\boldsymbol{x}, t) &= u_D(\boldsymbol{x}) && \text{on} \quad \Gamma_D \supseteq \Gamma_{\text{in}} := \{\boldsymbol{x} \in \partial\Omega\,,\ \boldsymbol{\beta} \cdot \boldsymbol{n} \leq 0\} \\ \partial_n u(\boldsymbol{x}, t) &= 0 && \text{on} \quad \Gamma_N \subseteq \partial\Omega \setminus \Gamma_D\,. \end{aligned} \tag{2}$$

The extension of the method, which is presented in the subsequent section, to the case of an arbitrary divergence free velocity field is treated in [27, 46].

For convection-diffusion problems standard Galerkin methods, whether conforming or not, are perfectly satisfactory from the point of convergence, but in practice the mesh has to be unrealistically fine to meet the theoretical requirements. Typically the grid size h should not exceed some fixed multiple of the global Péclet number Pe. When h exceeds this threshold the numerical solution shows spurious oscillations and is of little or no value.

We expect an acceptable numerical method to give an error that should decrease when the mesh is refined. Unfortunately a careful examination of simple upwind schemes show that for fixed ε the maximum pointwise error usually increases as the mesh is refined until the grid parameter h and the perturbation parameter have the same order of magnitude. This statement holds true for standard Galerkin methods, too. Here, we consider the concept of uniform convergence which has great influence in practice.

A discretization method is called *uniformly convergent* of order $k > 0$ with respect to ε in the norm $\|\cdot\|$, if there exists a constant C which is independent of ε, such that for all sufficiently small h (independently of ε),

$$\|u - u_h\| \leq Ch^k\,.$$

For uniformly convergent methods the error bound decreases as the mesh is refined, regardless of the ratio of h and ε.

In order to develop practicable techniques for small values of ε, the Galerkin approach has to be generalized by allowing different trial and test spaces, i.e. we apply a Petrov-Galerkin method.

Since it is possible to show that some fitted upwind schemes can be generated using a Petrov-Galerkin method with linear trial and quadratic test functions [6], we can state that also Petrov-Galerkin methods based on polynomial trial and test spaces are not practicable. Moreover, it is known from [16] that it is even not possible to construct a uniformly convergent method with polynomial spline functions on an equidistant grid. Consequently we summarize and state the numerical results for Petrov-Galerkin methods with polynomial trial and test spaces are not convincing.

Thus, we have to use exponential fitted Petrov-Galerkin methods. That means that the trial and – probably – the test functions are piecewise solutions of a simplified differential equation which is related to the original one in some sense.

In one space dimension we consider the differential equation

$$Lu := -\varepsilon u'' + \beta u' + \gamma u = f(x) .$$

A *homogeneous L-spline* is now constructed by using the condition that a basis function ϕ_i should be a solution of

$$\bar{L}\phi_i = 0 \quad \text{on every open subinterval.}$$

Analogously, we obtain an *inhomogeneous L-spline* by postulating that a basis function ϕ_i should fulfill

$$\bar{L}\phi_i = \bar{f} \quad \text{on every open subinterval.}$$

Here, we approximate the differential operator L by $\bar{L}u := -\varepsilon u'' + \beta u' + \gamma(x)u = \bar{f}(x)$ with $\bar{f}$ as a piecewise constant approximation to f [37].

This article does not intend to present convergence results for the Petrov-Galerkin discretization of the following section. Results on uniformly convergence for our method are part of [11].

3 A nonconforming Petrov-Galerkin method

For the presentation of the Petrov-Galerkin method we restrict ourselves to the case of homogeneous Dirichlet boundary conditions. The well-known extension to the general inhomogeneous case can be found in [8, 15].

With the aim to obtain a weak formulation for the original differential equation we multiply (1) with a function $w = w(\boldsymbol{x}) \in \mathbb{W}$ and integrate over the domain Ω. We choose $H_0^1 \subset \mathbb{V} := \{v \in H^1 | v = 0 \text{ on } \Gamma_D\} \subset H^1(\Omega)$ and $\mathbb{V} = \mathbb{W}$. Using Gauss' Theorem we observe that $u(t) \in \mathbb{V}$ is now the solution of the weak formulation

$$\frac{d}{dt}\int_\Omega u\, w\, d\Omega + \int_\Omega (\boldsymbol{\beta}\cdot\boldsymbol{\nabla}u\, w + \varepsilon\boldsymbol{\nabla}u\cdot\boldsymbol{\nabla}w)\; d\Omega \;\; = \;\; \int_\Omega qw\, d\Omega \quad \forall w \in \mathbb{W}.$$

This equation can be written shortly as

$$\frac{d}{dt}(u,w) + a(u,w) = (q,w) \quad \forall w \in \mathbb{W}$$

where the nonsymmetric bilinear form $a(\cdot,\cdot)$ is given by

$$a(u,w) \;\; := \;\; \int_\Omega \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u \, w + \varepsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} w \, d\Omega$$

and $(\cdot,\cdot)$ denotes the L_2-scalar product.

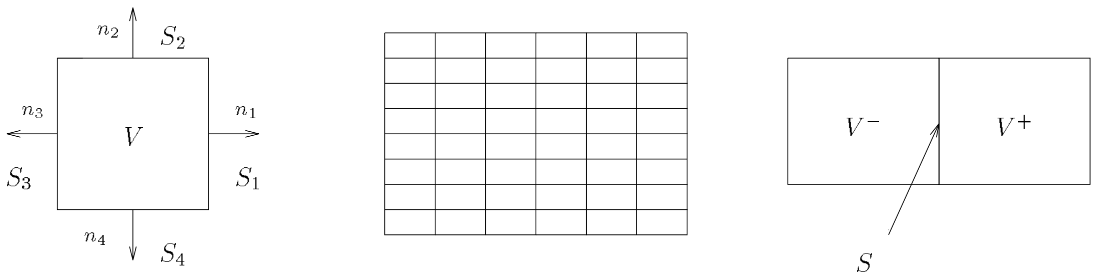

Figure 1: local cell, grid, and neighbouring cells

To obtain a Petrov-Galerkin method we replace the spaces $\mathbb{V}$ and $\mathbb{W}$ by finite spaces $\mathbb{V}_h$ and $\mathbb{W}_h$. The construction of the trial space $\mathbb{V}_h$ is based on the local spaces $\mathbb{V}_V$ defined by

$$\mathbb{V}_V = \operatorname{span}\left\{1, x_1, x_2, \exp\left(\frac{\beta_1}{\epsilon}x_1\right), \exp\left(\frac{\beta_2}{\epsilon}x_2\right)\right\}. \tag{3}$$

The space $\mathbb{V}_V$ contains the inhomogeneous exponentially fitted L-splines in both space directions.

We require that a trial function $v \in \mathbb{V}_h$ restricted to a cell V belongs to $\mathbb{V}_V$ and that the evaluation of the average for an interior edge S using these restrictions is independent of the cell under consideration. Therefore, we define a canonical basis of $\mathbb{V}$ by using the cell and edge averages as 'nodal values'.

Let $\mathcal{V}$ be the set of cells and $\mathcal{S}$ be the set of edges. Canonical trial functions ϕ^V for $V \in \mathcal{V}$ and ϕ^S for $S \in \mathcal{S}$ as a basis of $\mathbb{V}_h$ are defined by

$$\begin{aligned}
\phi^V(\tilde{V}) = \frac{1}{|\tilde{V}|}\int_{\tilde{V}} \phi^V(\boldsymbol{x})d\tilde{V} \;\; &= \;\; \delta_{V\tilde{V}} \quad \forall \tilde{V} \in \mathcal{V}\,, \\
\phi^V(\tilde{S}) = \frac{1}{|\tilde{S}|}\int_{\tilde{S}} \phi^V(\boldsymbol{x})d\tilde{S} \;\; &= \;\; 0 \qquad \forall \tilde{S} \in \mathcal{S}\,,
\end{aligned}$$

and

$$\begin{aligned}
\phi^S(\tilde{V}) = \frac{1}{|\tilde{S}|}\int_{\tilde{S}} \phi^S(\boldsymbol{x})d\tilde{S} \;\; &= \;\; \delta_{S\tilde{S}} \quad \forall \tilde{S} \in \mathcal{S}\,, \\
\phi^S(\tilde{S}) = \frac{1}{|\tilde{V}|}\int_{\tilde{V}} \phi^S(\boldsymbol{x})d\tilde{V} \;\; &= \;\; 0 \qquad \forall \tilde{V} \in \mathcal{V}\,.
\end{aligned}$$

In analogy to the definition of the trial functions we construct a test space $\mathbb{W}_h$ which is – for each $V \in \mathcal{V}$ – based on the local test space

$$\mathbb{W}_V = \operatorname{span}\Big\{1, x_1, x_2, \exp\Big(-\frac{\beta_1}{\epsilon}x_1\Big), \exp\Big(-\frac{\beta_2}{\epsilon}x_2\Big)\Big\}. \tag{4}$$

Here, $\mathbb{W}_V$ consists of inhomogeneous exponentially fitted L^*-splines where L^* denotes the adjoint operator of L.

Canonical test functions ψ^V, $V \in \mathcal{V}$ and ψ^S, $S \in \mathcal{S}$ are defined analogously to the canonical trial functions.

We observe that the trial and test functions are discontinuous at interelement boundaries. Consequently, we have to deal with nonconforming spaces and, moreover, we have to change the bilinear form $a(\cdot,\cdot)$ – by summing up over all grid cells – into

$$a_h(u,w) := \sum_{V\subset\Omega} \int_V \boldsymbol{\beta}\cdot\boldsymbol{\nabla} u\, w + \varepsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} w\, dV\,. \tag{5}$$

Thus the resulting nonconforming Petrov-Galerkin method can be described as follows.

Find $u(t) \in \mathbb{V}_h$ such that

$$\frac{d}{dt}(u,w) + a_h(u,w) = (q,w) \quad \forall w \in \mathbb{W}_h\,. \tag{6}$$

Describing $u(t)$ by its nodal values as

$$u(t) = \sum_{V\in\mathcal{V}} u_V(t)\Phi_V + \sum_{S\in\mathcal{S}} u_S(t)\Phi_S$$

and inserting the canonical test functions into (6) we obtain by setting $\boldsymbol{u}_{\mathcal{V}} := (u_V)_{V\in\mathcal{V}}$ and $\boldsymbol{u}_{\mathcal{S}} := (u_S)_{S\in\mathcal{S}}$

$$\hat{\boldsymbol{M}}\frac{d}{dt}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{u} = \boldsymbol{b} \iff \hat{\boldsymbol{M}}\frac{d}{dt}\boldsymbol{u} + \begin{pmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{pmatrix}\begin{pmatrix} \boldsymbol{u}_{\mathcal{V}} \\ \boldsymbol{u}_{\mathcal{S}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}. \tag{7}$$

If we approximate the L_2-scalar product appearing in the mass matrix $\hat{\boldsymbol{M}}$ and the right hand side $\boldsymbol{b}$ of (7) by the quadrature rule

$$(f,g) \approx \sum_{V\in\mathcal{V}} |V| f(V) g(V)$$

we achieve a system of the form

$$\boldsymbol{M}\frac{d}{dt}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{u} = \boldsymbol{b} \iff \begin{pmatrix} \boldsymbol{V} & 0 \\ 0 & 0 \end{pmatrix}\frac{d}{dt}\begin{pmatrix} \boldsymbol{u}_{\mathcal{V}} \\ \boldsymbol{u}_{\mathcal{S}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{pmatrix}\begin{pmatrix} \boldsymbol{u}_{\mathcal{V}} \\ \boldsymbol{u}_{\mathcal{S}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix} \tag{8}$$

where $\boldsymbol{V} = \operatorname{diag}(|V|)_{V\in\mathcal{V}}$.

An identical mass matrix is also obtained by using a mass lumping technique because of

$$\sum_{V\in\mathcal{V}} \Phi^V + \sum_{S\in\mathcal{S}} \Phi^S \equiv 1$$

in all cells.

The linear systems which arise are differential-algebraic systems of index 1 [5]. The stiffness matrix $\boldsymbol{B}$ in (8) can be assembled by the cell matrices

$$\begin{aligned}\boldsymbol{B}_{V,i} \;&=\; \frac{|S_i|}{2}\Biggl\{\mathrm{cof}\Bigl(\beta, \frac{2\varepsilon}{|S_{i+1}|}\Bigr)\begin{pmatrix} 4 & -2 & -2 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{pmatrix} + \frac{2\varepsilon}{|S_{i+1}|}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \\ &\qquad + \beta_i \begin{pmatrix} 0 & -2 & 2 \\ 2 & -1 & -1 \\ -2 & 1 & 1 \end{pmatrix}\Biggr\} \text{ for } V\in\mathcal{V},\, i=1,2.\end{aligned} \tag{9}$$

The cell matrices $\boldsymbol{B}_{V,i}$ are connected to the cell vectors $\boldsymbol{u}_{V,i} := (u_V, u_{S_{i+2}}, u_{S_i})^T$ ignoring rows and columns belonging to Dirichlet data u_S for $S \in \partial\mathcal{S}_D$. The scalar B_S is associated with u_S for Neumann boundaries $S \in \mathcal{S}_N$.

The function cof in (9) is defined by

$$\mathrm{cof}(\xi,\eta) \;=\; \begin{cases} \xi\Bigl(\coth(\frac{\xi}{\eta}) - \frac{\eta}{\xi}\Bigr)^{-1}, & \xi,\eta \neq 0 \\ |\xi|\,, & \eta = 0 \\ 3\eta\,, & \xi = 0 \end{cases}.$$

At this point we mention that the finite volume and also the Petrov-Galerkin approach can be extended to the case of an arbitrary, divergence free velocity field $\boldsymbol{\beta}$ (see [27, 46]).

For the application of an iterative solution method we state the definiteness of the stiffness matrix $\boldsymbol{B}$, the matrix $\boldsymbol{A}$ in

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} \quad \text{with} \quad \boldsymbol{A} = \frac{\gamma}{\Delta t}\boldsymbol{M} + \boldsymbol{B} \tag{10}$$

for constant time steps Δt which arises during time integration, and the corresponding Schur complement. For a proof see [29, 46].

Theorem 3.1
The stiffness matrix $\boldsymbol{B}$ is positive definite uniformly in h^2, i.e.

$$\boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} \geq \varepsilon C h^2 \boldsymbol{x}^T\boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N \tag{11}$$

with a constant $C > 0$ independent of h.

Corollary 3.2
The matrix $\boldsymbol{A}$ of (10) and its Schur complement $[\boldsymbol{A}/\boldsymbol{A}_{22}]$ are positive definite uniformly in h^2, i.e.

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geq \varepsilon\, C h^2 \boldsymbol{x}^T\boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N \tag{12}$$

and

$$\boldsymbol{x}_{\mathcal{V}}^T[\boldsymbol{A}/\boldsymbol{A}_{22}]\boldsymbol{x}_{\mathcal{V}} \geq \left(\varepsilon\, C + \frac{\gamma}{\Delta t} C_h^2\right) h^2 \boldsymbol{x}_{\mathcal{V}}^T\boldsymbol{x}_{\mathcal{V}} \quad \forall \boldsymbol{x}_{\mathcal{V}} \in \mathbb{R}^{N_1}. \tag{13}$$

Here, $C > 0$ is the constant from (11).

4 Krylov subspace methods

A large class of practical methods for the solution of sparse linear systems uses in some kind or another a projection process. The first method of this class, the conjugate gradient algorithm, was developed by Hestenes, Stiefel [26], and Lanczos [30], independently. First, it was seen as a direct method. In practice, there was no convergence in at most N steps observed, as it was predicted by theory. This effect is due to the loss of orthogonality during the runtime of the cg-algorithm. At the beginning of the 1970's the algorithm was interpreted as an iterative method for solving large sparse linear systems [36]. During the last two decades a great variety of conjugate-gradient-like methods was introduced [14]. Here, we present a selection of the most recent methods. For a detailled description we refer to [38, 34, 46], for instance.

The main idea of projection methods, in general, is to extract an approximation from a space of m candidate approximants denoted as $\mathcal{K}_m$ by imposing m constraints. To describe these constraints we specify m orthogonality conditions. In our case we postulate that the residual vector $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ should be orthogonal to m linearly independent vectors which form the so-called left subspace $\mathcal{L}_m$. Krylov subspace methods are based on orthogonal or oblique projection processes onto Krylov subspaces.

Let $\mathcal{K}_m$ and $\mathcal{L}_m$ denote two m-dimensional subspaces of $\mathbb{R}^N$. Using a projection onto $\mathcal{K}_m$ and a Petrov-Galerkin condition we construct an approximate solution $\tilde{\boldsymbol{x}}$ to the solution $\boldsymbol{x}$ of a linear system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

with a regular matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^N$.

As mentioned above, we like to find an approximate solution $\tilde{\boldsymbol{x}}$ from $\mathcal{K}_m$ — or more generally from $\boldsymbol{x}^{(0)} + \mathcal{K}_m$ with an arbitrary initial guess $\boldsymbol{x}^{(0)}$ — by imposing that the residual vectors should be orthogonal to $\mathcal{L}_m$. Our general problem now reads

Problem 4.1 (General Projection Method)
Find an approximate solution $\tilde{\boldsymbol{x}}$ from the affine subspace $\boldsymbol{x}^{(0)} + \mathcal{K}_m$ such that

$$\left(\boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}\right) \perp \mathcal{L}_m. \tag{14}$$

□

Here $\boldsymbol{x}^{(0)}$ is an arbitrary initial guess. We call $\mathcal{K}_m$ the *subspace of candidate approximants* and $\mathcal{L}_m$ the *subspace of constraints*. The orthogonality relations are defined via $\boldsymbol{x} \perp \boldsymbol{y} \iff (\boldsymbol{x}, \boldsymbol{y})_2 = 0$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$.

A *Krylov subspace* is defined by

$$K_m(\boldsymbol{A}, \boldsymbol{x}) := \operatorname{span}\left\{\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{x}\right\},$$

and the choice of $\mathcal{K}_m = K_m(\boldsymbol{A}, \boldsymbol{r}^{(0)})$ leads to a *Krylov subspace method*. Here $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$ denotes the initial residual.

Different versions of Krylov subspace methods arise from different choices of $\mathcal{L}_m$. For $\mathcal{K}_m = \mathcal{L}_m$ we get an *orthogonal projection method* and (14) represents a Galerkin condition. If $\mathcal{K}_m \neq \mathcal{L}_m$ we obtain an *oblique projection method* which is defined via the Petrov-Galerkin condition (14).

Here, we restrict ourselves to the most important Krylov subspace methods, namely the Generalized Minimal Residual (GMRES) [39], the Conjugate Gradient Squared (CGS) [40], the Transpose-free Quasi Minimal Residual (TFQMR) [12], the Bi-conjugate Gradient Stabilized (BiCGSTAB) [42], and the Quasi Minimal Residual Bi-conjugate Gradient Stabilized (QMRCGSTAB) [7] method.

The GMRES method represents an oblique projection method onto $\boldsymbol{x}^{(0)} + K_m$ orthogonal to $\mathcal{L}_m = \boldsymbol{A}K_m$. Since the storage amount of the GMRES method grows with m a GMRES version with restart is used in practice.

The CGS algorithm is a variant of the Bi-conjugate Gradient (BiCG) algorithm [10] which is an oblique projection method onto $\boldsymbol{x}^{(0)} + K_m$ orthogonal to the subspace $\mathcal{L}_m = \operatorname{span}\left\{\hat{\boldsymbol{r}}^{(0)}, \boldsymbol{A}^T\hat{\boldsymbol{r}}^{(0)}, \ldots, (\boldsymbol{A}^T)^{m-1}\hat{\boldsymbol{r}}^{(0)}\right\}$. Mostly $\hat{\boldsymbol{r}}^{(0)} = \boldsymbol{r}^{(0)}$ is chosen. In contrast to the BiCG method the CGS algorithm uses a slight modification to avoid multiplications with $\boldsymbol{A}^T$. It replaces the representation of the residual of the BiCG which is of the form

$$\boldsymbol{r}^{(m)} = p_m(\boldsymbol{A})\boldsymbol{r}^{(0)}$$

by

$$\boldsymbol{r}^{(m)} = p_m^2(\boldsymbol{A})\boldsymbol{r}^{(0)} \tag{15}$$

where p_m is a polynomial of degree m with $p_m(0) = 1$.

The BiCG and the CGS method have the advantageous property that the storage does not increase with m and that they are based on three-term recurrences. On the other hand they suffer from different breakdown possibilities. Additionally, the CGS method tends to oscillations which is due to the squaring of residual polynomials used.

With the aim to avoid the oscillating behaviour of the CGS algorithm the BiCGSTAB method replaces the representation of the residual from (15) by

$$\boldsymbol{r}^{(m)} = p_m(\boldsymbol{A})q_m(\boldsymbol{A})\boldsymbol{r}^{(0)}$$

with a certain polynomial q_m of degree m. Afterwards this additional degree of freedom is used for the minimization of the residual.

In order to avoid breakdowns the BiCG approach was equipped with a minimization strategy in [13]. The resulting QMR algorithm does not minimize the norm of the residual over the affine Krylov space $\boldsymbol{x}^{(0)} + K_m$ like in the case of the GMRES algorithm. Instead of the residual a so-called quasi-residual – which is easily computable – is used. A modification of the CGS method in an analogous way yields the TFQMR algorithm and the quasi-minimal version of the BiCGSTAB algorithm is the QMRCGSTAB algorithm.

5 Preconditioning of linear systems

The convergence rate of an iterative method for the solution of

$$\boldsymbol{A}\boldsymbol{x} \;=\; \boldsymbol{b} \tag{16}$$

strongly depends on the condition number $\operatorname{cond}(\boldsymbol{A}) := \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$.

The consequence of a large condition number is that the solution of the system is very sensitive to disturbances of the matrix and the right hand side. In general we state that the systems which arise from the discretization of partial differential equations are ill-conditioned, i.e. $\operatorname{cond}(\boldsymbol{A}) \gg 1$.

With the intention to improve the convergence of an iterative method we construct an equivalent system to (16) by multiplying with the inverse of a regular matrix $\boldsymbol{T}_1$ and substituting $\boldsymbol{x}$ by $\boldsymbol{T}_2^{-1}\boldsymbol{y}$. (16) now transforms into

$$\boldsymbol{T}_1^{-1}\boldsymbol{A}\boldsymbol{T}_2^{-1}\boldsymbol{y} \;=\; \boldsymbol{T}_1^{-1}\boldsymbol{b}. \tag{17}$$

Afterwards we apply an arbitrary iterative solution method.

The advantage of (17) is that the matrix of the transformed system remains symmetric if $\boldsymbol{A}$ is symmetric and if we have $\boldsymbol{T}_1 = \boldsymbol{T}_2^T$. This is the case when we use an incomplete Cholesky decomposition [1, 43], for instance.

The main problem is now to find preconditioning matrices which allow an efficient solution of the system. If we consider, for example, a left preconditioned system, i.e. $\boldsymbol{T}_2 = \boldsymbol{I}$ we can state that the matrix $\boldsymbol{T}_1$ should be a good approximation to the system matrix $\boldsymbol{A}$, but $\boldsymbol{T}_1$ should be easy to invert, additionally. With the aim to fulfill both requirements we now suggest some possibilities for the selection of preconditioning matrices.

We note that a preconditioned system may be a full system. Even though the preconditioning matrix may be sparse, its inverse does not necessarily need to be sparse, too. As a consequence the number of techniques we can choose for the solution of a preconditioned system is limited. We use methods which require matrix-vector products only. E.g. for an operation of the form $\boldsymbol{w} = \boldsymbol{T}_1^{-1}\boldsymbol{A}\boldsymbol{v}$ we proceed in two steps. First, we compute $\boldsymbol{r} := \boldsymbol{A}\boldsymbol{v}$ and, subsequently, we solve $\boldsymbol{w} = \boldsymbol{T}_1^{-1}\boldsymbol{r}$.

First we consider the simplest way of preconditioning. A preconditioning with $\boldsymbol{T}_1 := \boldsymbol{D} = \operatorname{diag}(d_{11}, \ldots, d_{NN})$ with a diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is called *scaling* of the system (16). The simplest choice is $d_{ii} = a_{ii}$ for $i = 1, \ldots, N$ where we suppose that $a_{ii} \neq 0$.

This very simple choice of a preconditioner leads often to poor improvements, especially for discretization matrices. Consequently, we have to take a more sophisticated way. Stationary iterative methods were used as iterative solution methods up to the 1960's. Nowadays, they play a role as smoothing iterations in a multigrid code (see Chapter 6), or as preconditioners.

Here we give an insight into the relation between classical iterative methods such as the Jacobi iteration and the solution of certain preconditioned systems. Therefore, we recall that classical iterative methods are of the general form

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{W}\boldsymbol{x}^{(m)} + \boldsymbol{f}\,. \tag{18}$$

They are based on a splitting of the system matrix $\boldsymbol{A}$ of the form

$$\boldsymbol{A} = \boldsymbol{T_1} - \boldsymbol{N}$$

with a regular matrix $\boldsymbol{T_1}$.

We define a fixed point iteration for our original system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by the recursion (18) rewritten as

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{T}_1^{-1}(\boldsymbol{N}\boldsymbol{x}^{(m)} + \boldsymbol{b})\,.$$

Now we are able to view the iteration (18) as a technique for attempting to solve $(\boldsymbol{I} - \boldsymbol{W})\boldsymbol{x} = \boldsymbol{f}$ with $\boldsymbol{f} := \boldsymbol{T}_1^{-1}\boldsymbol{b}$. Since $\boldsymbol{W}$ is of the form $\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{T}_1^{-1}\boldsymbol{A}$ this system can be written as

$$\boldsymbol{T}_1^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{T}_1^{-1}\boldsymbol{b}\,.$$

This system is a left-preconditioned system, and we can consider a relaxation scheme of the form (18) for the original system as a fixed point iteration on a preconditioned system.

Classical schemes are based on an additive decomposition of the form $\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{E} - \boldsymbol{F}$. Here $\boldsymbol{D}$ is the diagonal of $\boldsymbol{A}$, $\boldsymbol{E}$ is the strict lower and $\boldsymbol{F}$ the strict upper part of $\boldsymbol{A}$.

The most common way of deriving a preconditioner for the system (16) is to approximate the matrix $\boldsymbol{A}$ by a decomposition of the form

$$\boldsymbol{A} \;=\; \boldsymbol{L}\boldsymbol{U} - \tilde{\boldsymbol{R}} \tag{19}$$

where $\boldsymbol{L}$ is a lower and $\boldsymbol{U}$ an upper triangular matrix. This means, in practice, that we determine an approximate LU decomposition of $\boldsymbol{A}$.

Definition 5.1 (Graph of a matrix)
Let $\mathcal{I} := \{1, \ldots N\}$ and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$. The subset

$$\mathcal{G}(\boldsymbol{A}) \;=\; \{(i,j) \in \mathcal{I} \times \mathcal{I} : a_{ij} \neq 0\}$$

of all elements of $\mathcal{I} \times \mathcal{I}$ is denoted as the *graph* of $\boldsymbol{A}$. □

For the construction of an incomplete decomposition we postulate that

$$l_{ij} = u_{ij} = 0 \text{ for all } (i,j) \notin \mathcal{P}$$

where the pattern $\mathcal{P}$ is a subset of $\mathcal{I} \times \mathcal{I}$. We always require $(i,i) \in \mathcal{P}$ for all $i \in \mathcal{I}$, and in general one should choose $\mathcal{P}$ large enough, i.e.

$$\mathcal{P} \supset \mathcal{G}(\boldsymbol{A}) .$$

The algorithm for the determination of an ILU decomposition, which fulfills

$$\sum_{k=1}^{n} l_{ik} u_{kj} = a_{ij} \text{ for all } (i,j) \in \mathcal{P} , \tag{20}$$

can be found in [19].

For our computations we set $\mathcal{P} = \mathcal{G}(\boldsymbol{A})$, but we point out that for specific problems the choice of a larger pattern $\mathcal{P}$ can lead to better convergence rates [45].

There are several modifications and extensions to define incomplete LU decompositions, mainly with the aim to derive stable versions of the solution algorithm [23, 38]. Concerning the existence of ILU decompositions we refer to [19, 31, 33].

6 A linear multigrid method

Another class of solution methods for linear (and also nonlinear) systems are multigrid methods. For an introduction see [18, 32, 44]. The first multigrid method (MGM) was formulated by Fedorenko in 1964 [9]. This was a multigrid algorithm for the standard 5-point finite difference discretization of Poisson's equation on a square. In 1966 Bachvalov [2] proved the optimal $\mathcal{O}(N)$-order of complexity for the more difficult case of a difference scheme for a general elliptic partial differential equation on the unit square. The first practical results and the efficiency of multigrid algorithms were reported by Brandt in 1972 [3]. Independently the multigrid method was discovered by Hackbusch in 1976 [17] who laid the firm mathematical foundations. Articles on advanced topics can be found in the collections of Hackbusch and Trottenberg [20, 21, 22], and Hemker and Wesseling [24, 25].

In practice the solution of advanced problems often exceeds the capacity of the computers or requires to much time. Multigrid methods offer the possibility of solving problems with N unknowns with $\mathcal{O}(N)$ work and storage for a large class of problems. Consequently, there is a big significance of multigrid methods for scientific computing. A second, very important additional feature is a convergence rate which is bounded away from 1 for decreasing mesh size.

We observe that basic iterative methods reduce the components of the error belonging to so-called high frequencies rapidly and, moreover, that the slow convergence is caused by the lower frequencies. The essential idea of multigrid is now to approximate the smooth parts of the error on coarser grids and to reduce the rough parts with a small number of iterations, which is independent of h, using a basic iterative method on the fine grid.

We consider a sequence of nested grids (see Figure 2)

$$\mathcal{G}_0 \subset \mathcal{G}_1 \subset \ldots \subset \mathcal{G}_{l_{max}}\,.$$

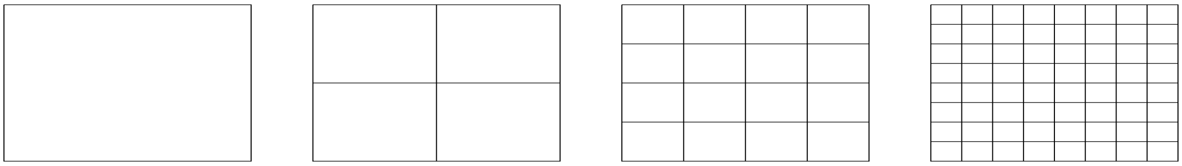

Figure 2: A sequence of nested grids

Let $\boldsymbol{u}^l \in \mathbb{R}^{N_l}$, $l = 0, \ldots, l_{max}$ be grid functions and, furthermore, let $\boldsymbol{S}^l : \mathbb{R}^{N_l} \to \mathbb{R}^{N_l}$ be a smoothing iteration, $\boldsymbol{P}^l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ a prolongation, and $\boldsymbol{R}^l : \mathbb{R}^{N_l} \to \mathbb{R}^{N_{l-1}}$ a restriction operator for $l = 1, \ldots, l_{max}$.

For the derivation of a linear algorithm we consider a sequence of matrices

$$\boldsymbol{A}^0, \boldsymbol{A}^1, \ldots \boldsymbol{A}^{l_{\max}} \quad \text{with} \quad \boldsymbol{A}^l \in \mathbb{R}^{N_l \times N_l}\,.$$

The problem to be solved on grid $\mathcal{G}_l$ reads

$$\boldsymbol{A}^l \boldsymbol{u}^l = \boldsymbol{f}^l\,. \tag{21}$$

Concerning the computation of these matrices we make the following remark. The two-grid algorithm, for instance, requires an approximation of the fine grid matrix $\boldsymbol{A} := \boldsymbol{A}^1$ by a matrix $\bar{\boldsymbol{A}} := \boldsymbol{A}^0$ on the coarse grid. Basically there are two ways to compute $\bar{\boldsymbol{A}}$. The *discretization coarse grid approximation (DCA)* computes the matrix $\bar{\boldsymbol{A}}$ as a discretization matrix of the partial differential equation on the coarse grid. And the *Galerkin coarse grid approximation (GCA)* [47] chooses $\bar{\boldsymbol{A}}$ as $\bar{\boldsymbol{A}} = \boldsymbol{RAP}$ with a restriction operator $\boldsymbol{R}$ and a prolongation operator $\boldsymbol{P}$. The advantages of GCA in comparison with DCA are that the coarsest grids may be very coarse. On such grids DCA may be unreliable if the coefficients are variable, because these coefficients are sampled in very few points. A remedy is to replace the pointwise sampling of the coefficients by a suitable averaging. GCA does this accurately and automatically. Also GCA has a pure algebraic nature and makes no use of the underlying partial differential equation. Consequently, it is well-suited for the use in 'black box' multigrid codes. The disadvantages of GCA are that for nonlinear problems and systems of differential equations there is no general way to implement GCA. For an example see [47]. Anyway both alternatives are in widespread use. For our computations we use a DCA representation.

A multigrid method for the solution of our linear systems can be written as

Algorithm 6.1

$\mathbf{MGM}(l, \boldsymbol{u}^l, \boldsymbol{f}^l)$

$$
\begin{array}{lll}
\text{if} \quad (l = 0) & & \\
\quad \boldsymbol{u}^0 & := & (\boldsymbol{A}^0)^{-1}\boldsymbol{f}^0 \qquad \textit{(exact solution on coarsest grid)} \\
\text{else} & & \\
\quad \boldsymbol{u}^l & := & \boldsymbol{S}^l(\boldsymbol{u}^l, \boldsymbol{f}^l; \nu_1) \qquad \textit{(presmoothing)} \\
\quad \boldsymbol{d}^{l-1} & := & \boldsymbol{R}^l(\boldsymbol{f}^l - \boldsymbol{A}^l\boldsymbol{u}^l) \qquad \textit{(restriction of defect)} \\
\quad \tilde{\boldsymbol{u}}^{l-1} & := & \boldsymbol{0} \\
\quad \text{for} \quad j = 1, \ldots, \mu \quad \text{do } \mathbf{MGM}(l-1, \tilde{\boldsymbol{u}}^{l-1}, \boldsymbol{d}^{l-1}) & & \\
\quad \boldsymbol{u}^l & := & \boldsymbol{u}^l + \boldsymbol{P}^l\boldsymbol{u}^{l-1} \qquad \textit{(coarse grid correction)} \\
\quad \boldsymbol{u}^l & := & \boldsymbol{S}^l(\boldsymbol{u}^l, \boldsymbol{f}^l; \nu_2) \qquad \textit{(postsmoothing)} \\
\text{endif} & &
\end{array}
$$

□

We can improve the algorithm above by starting on the coarsest grid and successively computing the solution on the next level in order to obtain a better starting guess on the finer level. This approach is called *nested iteration* or *full multigrid (FMG)* [4].

The main problem for the derivation of an efficient multigrid algorithm is the determination of the different components of the algorithm, especially of the smoother and the transfer operators.

The coarsest grid consists of five equations, i.e. we have one cell at least. In practice we choose a coarse grid with a moderate number of equations. Consequently, we do not solve the coarse grid system exactly, but approximately. For the solution on this grid we use the GMRES method since this method is theoretically convergent and practically a fast convergent method for a medium-sized system of our discretization [39, 45].

Choice of a smoothing iteration

If we consider linear systems which arise from the discretization of singularly perturbed problems we cannot use any basic iterative method as a smoother. We have to construct special smoothing iterations. From [18] we quote the following criterion for the choice of $\boldsymbol{S}^l$.

Criterion 6.2
The smoothing iteration $\boldsymbol{S}^l$ should be a fast iterative (or even a direct) solver for the limit case $\varepsilon = 0$. □

We suggest two different smoothers. The first is the well-known Gauss-Seidel method [41] and the second is an iteration which is based on the ILU decomposition. This ILU iteration reads as

$$\boldsymbol{T}_1(\boldsymbol{u}^{m+1} - \boldsymbol{u}^m) = \boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}^m \tag{22}$$

with $\boldsymbol{T}_1 = \boldsymbol{LU}$ from the ILU decomposition.

Since in the limit case $\varepsilon = 0$ and with a numbering in direction of the velocity field our linear system reduces to lower triangular form we can state that the Gauss-Seidel and the ILU iteration as well are exact solvers. We obtain

Lemma 6.3
The Gauss-Seidel method and the ILU iteration are well-suited smoothing iterations for a multigrid algorithm based on our Petrov-Galerkin discretization. □

Again we consider the ILU decomposition and prove

Theorem 6.4 (Exactness of the ILU decomposition)
For the ILU decomposition with the pattern $\mathcal{P} = \mathcal{G}(\boldsymbol{A})$ we have with $\mathcal{N} = \mathcal{V} \cup \mathcal{S}$

$$a_{vn} \quad = \quad (\boldsymbol{LU})_{vn} \text{ for } v \in \mathcal{V} \text{ and } n \in \mathcal{N}\,. \tag{23}$$

Proof:

From (20) we obtain the exactness of the ILU decomposition for the pattern $\mathcal{P}$, i.e.

$$a_{vn} \quad = \quad (\boldsymbol{LU})_{vn} \text{ for } v, n \in \mathcal{P}\,. \tag{24}$$

Without loss of generality we assume that the velocity field $\boldsymbol{\beta}$ has components $\beta_1, \beta_2 > 0$.

With a numbering in the direction of the velocity field we observe that the column indices affiliated to u_{S_3}, u_V and u_{S_1} belong to the v-th row of $\boldsymbol{L}$ (see Figure 3).

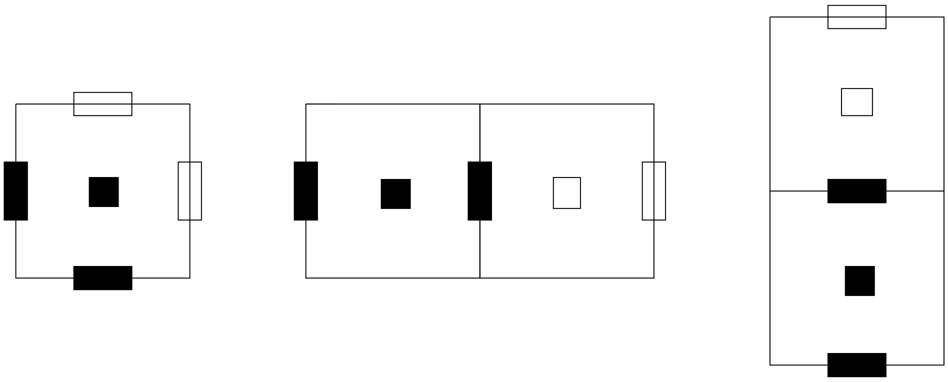

Figure 3: Pattern of $\boldsymbol{L}$ and $\boldsymbol{U}$

For the limit case $\boldsymbol{\beta} = \boldsymbol{0}$ we obtain a symmetric matrix $\boldsymbol{A}$ which has a symmetric pattern, of course. The pattern of $\boldsymbol{A}$ is independent of the choice of $\boldsymbol{\beta}$ so that the matrix $\boldsymbol{A}$ has a symmetric pattern in general.

This means that $\boldsymbol{U}^T$ has the same pattern as $\boldsymbol{L}$ and, thus, row n of $\boldsymbol{U}$ has the same structure as the column n of $\boldsymbol{L}$.

The element $(\boldsymbol{LU})_{vn}$ is the scalar product of the vth row of $\boldsymbol{L}$ with the nth column of $\boldsymbol{U}$. And this scalar product is zero if there are no common nonvanishing entries. Considering for $v \in \mathcal{V}$ the two cases $n \in \mathcal{V}$ with $n \neq v$ and $n \in \mathcal{S}$ we conclude that there are no common entries and, consequently, that the scalar product is zero. Hence, we have shown $(\boldsymbol{LU})_{vn} = 0 = a_{vn}$ for $v \in \mathcal{V}$ and $(v, n) \notin \mathcal{P}$ which concludes the proof. □

As a consequence of this result we obtain

Lemma 6.5
The ILU iteration is an exact solver for the equations belonging to the cells.

Proof:

We consider equation (22)

$$\boldsymbol{LU}(\boldsymbol{u}^{m+1} - \boldsymbol{u}^m) = \boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}^m$$

and examine the equation belonging to a cell equation v which has the form

$$\sum_{n\in\mathcal{N}} (\boldsymbol{LU})_{vn}(\boldsymbol{u}_n^{m+1} - \boldsymbol{u}_n^m) = \boldsymbol{f}_v - \sum_{n\in\mathcal{N}} a_{vn}\boldsymbol{u}_n^m .$$

Theorem 6.4 gives now

$$\sum_{n\in\mathcal{N}} a_{vn}\boldsymbol{u}_n^{m+1} = \boldsymbol{f}_v$$

which is the desired result. □

Restriction and prolongation

For the description of restriction and prolongation operators we assume that a cell V belonging to $\mathcal{G}_{l-1}$ consists of four adjacent fine grid cells $\tilde{V}_k$ on level l as shown in Figure 4. The two interpretations of our discretization lead to different versions of these operators. The index l is neglected when there is no ambiguity.

Petrov-Galerkin approach

For finite element methods the prolongation and restriction operators are naturally defined in terms of the nodal functions using the corresponding function spaces [18].

For a conforming method with nested finite element spaces $\mathbb{V}_h^0 \subset \mathbb{V}_h^1 \subset \cdots \subset \mathbb{V}_h^{l_{\max}}$ and a nodal basis $\mathbb{V}_h^l = \mathrm{span}\{\phi^{l,i}, i = 1, \ldots, N_l\}$ we can express the nodal functions on level $l-1$ by

$$\phi^{l-1,j} = \sum_{i=1}^{N_l} \phi_i^{l-1,j}\phi^{l,i}\,, j = 1, \ldots, N_{l-1}$$

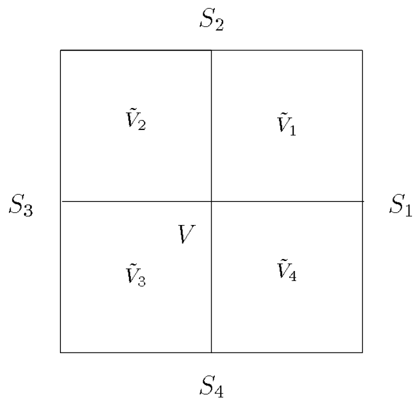

Figure 4: local notations

where ϕ_i denotes the nodal value of ϕ for the index i. The canonical prolongation $\tilde{P}^l$ on the trial spaces is defined by the inclusion

$$\mathbb{V}_h{}^l \ni \tilde{P}^l u^{l-1} = u^{l-1} = \sum_{i=1}^{N_l} \sum_{j=1}^{N_{l-1}} \phi_i^{l-1,j} u_j^{l-1} \phi^{l,i} \in \mathbb{V}_h{}^{l-1}$$

with the prolongation matrix given as

$$(\boldsymbol{P}^l)_{ij} = \phi_i^{l-1,j} \quad \text{for} \quad i = 1, \ldots, N_l, \quad j = 1, \ldots, N_{l-1}.$$

The corresponding restriction matrix $\boldsymbol{R}^l$ for the residuals tested by the nodal functions is

$$\boldsymbol{R}^l = (\boldsymbol{P}^l)^T.$$

Considering Petrov-Galerkin methods the prolongation is connected to the trial spaces $\mathbb{V}_h{}^l$ and the restriction to the test spaces $\mathbb{W}_h{}^l$. Therefore, we have for $i = 1, \ldots, N_l$ and $j = 1, \ldots, N_{l-1}$

$$(\boldsymbol{P}^l)_{ij} = \phi_i^{l-1,j}, \tag{25}$$

$$(\boldsymbol{R}^l)_{ji} = \psi_i^{l-1,j}. \tag{26}$$

In our case the nodal values are the averages indexed by $V \in \mathcal{V}$ and $S \in \mathcal{S}$ instead of i. For the one-dimensional problem the evaluation is evident. In two dimensions we have to deal with nonconforming nodal functions. Due to the discontinuity at the boundary D of $\operatorname{supp}(\phi^{l-1,j})$ and $\operatorname{supp}(\psi^{l-1,j})$ we define the edge averages for $\mathcal{S} \subset D$ to be zero as in the conforming case:

$$\phi_S^{l-1,j} := \begin{cases} \frac{1}{2}\phi^{l-1,j}{}_{|D}(S) & \text{for} \quad S \in \mathcal{S} \setminus \partial\mathcal{S} \\ \phi^{l-1,j}{}_{|D}(S) & \text{for} \quad S \in \partial\mathcal{S} \end{cases}$$

and for the restriction, analogously.

Figure 5 illustrates the prolongation and restrictions for the limit cases of the diffusion and the convection problem in form of distribution stencils for $\boldsymbol{P}^l$ and collection stencils for $\boldsymbol{R}^l$. Since $\boldsymbol{P}^l = (\boldsymbol{R}^l)^T$ holds for the diffusion case we show the distribution stencil, only.

$$
\left]\begin{array}{ccccc}
 & 0 & & 0 & \\
0 & 1 & 3/2 & 1 & 0 \\
 & 3/2 & \blacksquare & 3/2 & \\
0 & 1 & 3/2 & 1 & 0 \\
 & 0 & & 0 &
\end{array}\right[
\qquad
\left]\begin{array}{ccccccccc}
 & 0 & & 0 & & 0 & & 0 & \\
0 & -1/4 & -1/4 & 1/4 & 1 & 1/4 & -1/4 & -1/4 & 0 \\
 & -1/4 & & 1/4 & \blacksquare & 1/4 & & -1/4 & \\
0 & -1/4 & -1/4 & 1/4 & 1 & 1/4 & -1/4 & -1/4 & 0 \\
 & 0 & & 0 & & 0 & & 0 &
\end{array}\right[
$$

$$
\left]\begin{array}{ccccc}
 & 0 & & 0 & \\
0 & 1 & 3/2 & 2 & 0 \\
 & 1/2 & \blacksquare & 3/2 & \\
0 & 0 & 1/2 & 1 & 0 \\
 & 0 & & 0 &
\end{array}\right[
\qquad
\left]\begin{array}{ccccccccc}
 & 0 & & 0 & & 0 & & 0 & \\
0 & 0 & 0 & 0 & 1 & 1/2 & 0 & -1/2 & 0 \\
 & 0 & & 0 & \blacksquare & 1/2 & & -1/2 & \\
0 & 0 & 0 & 0 & 1 & 1/2 & 0 & -1/2 & 0 \\
 & 0 & & 0 & & 0 & & 0 &
\end{array}\right[
$$

$$
\left[\begin{array}{ccccc}
 & 0 & & 0 & \\
0 & 1 & 1/2 & 0 & 0 \\
 & 3/2 & \blacksquare & 1/2 & \\
0 & 2 & 3/2 & 1 & 0 \\
 & 0 & & 0 &
\end{array}\right]
\qquad
\left[\begin{array}{ccccccccc}
 & 0 & & 0 & & 0 & & 0 & \\
0 & -1/2 & 0 & 1/2 & 1 & 0 & 0 & 0 & 0 \\
 & -1/2 & & 1/2 & \blacksquare & 0 & & 0 & \\
0 & -1/2 & 0 & 1/2 & 1 & 0 & 0 & 0 & 0 \\
 & 0 & & 0 & & 0 & & 0 &
\end{array}\right]
$$

Figure 5: Prolongation and restriction

Finite volume approach

The underlying system of ordinary differential equations motivates to understand the differential part of the semidiscrete system as finite volume equations and the algebraic part as constraints. This approach is based on an alternative derivation of the linear systems (8) in the framework of finite volume methods which is presented in [27, 28]. Transferring this point of view to the linear system (10) we consider first of all the Schur complement system

$$[\boldsymbol{A}/\boldsymbol{A}_{22}]\boldsymbol{u}_{\mathcal{V}} = \boldsymbol{f}_1 - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{f}_2 = \boldsymbol{f}_{\text{fv}} \tag{27}$$

which determines the cell averages $\boldsymbol{u}_{\mathcal{V}}$ independently of the edge averages $\boldsymbol{u}_{\mathcal{S}}$. Thus, we have to derive a restriction and prolongation for the parts corresponding to cells.

The defect vector for this Schur complement system is given by $\boldsymbol{d}_{\mathcal{V}} = [\boldsymbol{A}/\boldsymbol{A}_{22}]\boldsymbol{u}_{\mathcal{V}} - \boldsymbol{f}_{\text{fv}}$. Its components d_V should be approximations to the integral of a defect function $\tilde{d}(\boldsymbol{x})$, i.e.

$$d_V \approx \int_V \tilde{d}(\boldsymbol{x})d\boldsymbol{x} \tag{28}$$

which is motivated by the finite volume approach [27, 28].

This defines a restriction $\boldsymbol{R}^l_{\mathcal{V}}$ by means of

$$(\boldsymbol{R}^l_{\mathcal{V}}\boldsymbol{d}^l_{\mathcal{V}})_V := \sum_{i=1}^{4} d^l_{\tilde{V}_i} \approx \sum_{i=1}^{4} \int_{\tilde{V}_i} \tilde{d}(\boldsymbol{x})d\boldsymbol{x} = \int_V \tilde{d}(\boldsymbol{x})d\boldsymbol{x} \approx d^{l-1}_V \text{ for all } V \in \mathcal{V}. \tag{29}$$

With respect to the scalar products $\boldsymbol{d}_{\mathcal{V}}^T \boldsymbol{V}^{-1} \boldsymbol{d}_{\mathcal{V}}$ for defect vectors and $\boldsymbol{u}_{\mathcal{V}}^T \boldsymbol{V} \boldsymbol{u}_{\mathcal{V}}$ for cell averages the corresponding prolongation $\boldsymbol{P}_{\mathcal{V}}^l$ simply reads

$$\boldsymbol{P}_{\mathcal{V}}^l = (\boldsymbol{R}_{\mathcal{V}}^l)^T. \tag{30}$$

The resulting multigrid algorithm is not really efficient since the Schur complement $[\boldsymbol{A}/\boldsymbol{A}_{22}]$ should not be computed explicitly. Hence, we have to extend the grid transfer operators including the edge components to get a comparable multigrid method for the whole system.

The matrix $\boldsymbol{A}$ can be decoupled by the block decomposition

$$\begin{aligned} \boldsymbol{A} &= \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} [\boldsymbol{A}/\boldsymbol{A}_{22}] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21} & \boldsymbol{I} \end{pmatrix} && (31) \\ &= (\boldsymbol{I}+\boldsymbol{U})\hat{\boldsymbol{A}}(\boldsymbol{I}+\boldsymbol{L}) && (32) \end{aligned}$$

where the back transformations are given by the identities $(\boldsymbol{I}+\boldsymbol{U})(\boldsymbol{I}-\boldsymbol{U}) = \boldsymbol{I}$ and $(\boldsymbol{I}+\boldsymbol{L})(\boldsymbol{I}-\boldsymbol{L}) = \boldsymbol{I}$. Using this, we are able to define a decoupled system by $\hat{\boldsymbol{A}}\hat{\boldsymbol{u}} = \hat{\boldsymbol{f}}$ with $\hat{\boldsymbol{u}} = (\boldsymbol{I}+\boldsymbol{L})\boldsymbol{u}$ and $\hat{\boldsymbol{f}} = (\boldsymbol{I}-\boldsymbol{U})\boldsymbol{f}$.

For this system the restriction and the prolongation should also be decoupled, i.e.

$$\hat{\boldsymbol{R}}^l := \begin{pmatrix} \boldsymbol{R}_{\mathcal{V}}^l & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{\mathcal{S}}^l \end{pmatrix}, \quad \hat{\boldsymbol{P}}^l := \begin{pmatrix} \boldsymbol{P}_{\mathcal{V}}^l & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{\mathcal{S}}^l \end{pmatrix} \tag{33}$$

with $\boldsymbol{R}_{\mathcal{V}}^l$ and $\boldsymbol{P}_{\mathcal{V}}^l$ from (29, 30) and analogously

$$(\boldsymbol{R}_{\mathcal{S}}^l \boldsymbol{d}_{\mathcal{S}}^l)_S := \sum_{i=1}^{2} d_{\tilde{S}_i}^l, \quad \boldsymbol{P}_{\mathcal{S}}^l = (\boldsymbol{R}_{\mathcal{S}}^l)^T \text{ for all } S \in \mathcal{S} \tag{34}$$

where $\tilde{S}_1, \tilde{S}_2$ are the edges on the fine grid belonging of S.

Assuming that the defect vector transforms like the right hand side and the correction vector like a solution vector the corresponding transfer operators of the original system are

$$\begin{aligned} \boldsymbol{R}^l &:= (\boldsymbol{I}+\boldsymbol{U}^{l-1})\hat{\boldsymbol{R}}^l(\boldsymbol{I}-\boldsymbol{U}^l), && (35) \\ \boldsymbol{P}^l &:= (\boldsymbol{I}-\boldsymbol{L}^l)\hat{\boldsymbol{P}}^l(\boldsymbol{I}+\boldsymbol{L}^{l-1}). && (36) \end{aligned}$$

Theorem 6.6
Under the assumption of an exact solution on the coarse grid the defect corrections for the cell variables according to the Schur complement system (27) and the original system (10) are identical.

Proof:

The correction vector belonging to (10) is defined by

$$\Delta \boldsymbol{u}^l \quad := \quad \boldsymbol{P}^l (\boldsymbol{A}^{l-1})^{-1} \boldsymbol{R}^l \left(\boldsymbol{f}^l - \boldsymbol{A} \boldsymbol{u}^l \right)$$

and for the Schur complement system by

$$\Delta \boldsymbol{u}^l_{\text{fv}} \quad := \quad \boldsymbol{P}^l_{\mathcal{V}} \left([\boldsymbol{A}/\boldsymbol{A}_{22}]^{l-1} \right)^{-1} \boldsymbol{R}^l_{\mathcal{V}} \left(\boldsymbol{f}^l_{\text{fv}} - [\boldsymbol{A}/\boldsymbol{A}_{22}]^l \boldsymbol{u}^l_{\mathcal{V}} \right)$$

with exact solution on the coarse level $l-1$. Hence, we have to show that

$$\left(\Delta \boldsymbol{u}^l \right)_{\mathcal{V}} = \Delta \boldsymbol{u}^l_{\text{fv}} \,. \tag{37}$$

Using the definitions (35, 36) for the restriction and the prolongation, the factorization (31), and the block diagonal structure of $\hat{\boldsymbol{P}}^l$, $\hat{\boldsymbol{R}}^l$ and $\hat{\boldsymbol{A}}^{l-1}$ we get

$$\begin{aligned}
\Delta \boldsymbol{u}^l \quad &= \quad (\boldsymbol{I} - \boldsymbol{L}^l) \hat{\boldsymbol{P}}^l (\boldsymbol{I} + \boldsymbol{L}^{l-1}) \left(\boldsymbol{A}^{l-1} \right)^{-1} (\boldsymbol{I} + \boldsymbol{U}^{l-1}) \hat{\boldsymbol{R}}^l (\boldsymbol{I} - \boldsymbol{U}^l) \left(\boldsymbol{f}^l - \boldsymbol{A}^l \boldsymbol{u}^l \right) \\
&= \quad (\boldsymbol{I} - \boldsymbol{L}^l) \hat{\boldsymbol{P}}^l \left(\hat{\boldsymbol{A}}^{l-1} \right)^{-1} \hat{\boldsymbol{R}}^l \left((\boldsymbol{I} - \boldsymbol{U}^l) \boldsymbol{f}^l - \hat{\boldsymbol{A}}^l (\boldsymbol{I} + \boldsymbol{L}^l) \boldsymbol{u}^l \right)
\end{aligned}$$

and finally

$$\Delta \boldsymbol{u}^l = (\boldsymbol{I} - \boldsymbol{L}^l) \begin{pmatrix} \hat{\boldsymbol{P}}^l_{\mathcal{V}} \left([\boldsymbol{A}/\boldsymbol{A}_{22}]^{l-1} \right)^{-1} \boldsymbol{R}^l_{\mathcal{V}} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{P}}^l_{\mathcal{S}} (\boldsymbol{A}^{l-1}_{22})^{-1} \hat{\boldsymbol{R}}^l_{\mathcal{S}} \end{pmatrix} \begin{pmatrix} \boldsymbol{f}^l_{\text{fv}} - [\boldsymbol{A}/\boldsymbol{A}_{22}]^l \boldsymbol{u}^l_{\mathcal{V}} \\ \boldsymbol{f}^l_2 - (\boldsymbol{A}^l_{21} \boldsymbol{u}^l_{\mathcal{V}} + \boldsymbol{A}^l_{22} \boldsymbol{u}^l_{\mathcal{S}}) \end{pmatrix} .$$

Evaluating the cell part of this product the assertion (37) follows, directly. □

For the implementation of the operators (33) it is important that a system with the matrix $\boldsymbol{A}_{22}$ can be solved at low costs. Fortunately we can state that the matrix $\boldsymbol{A}_{22} = \boldsymbol{B}_{22}$ of (10) is strictly diagonal dominant for $\varepsilon > 0$ and reduces to triangular form for $\varepsilon = 0$ with a numbering in convection direction. Hence the Gauss-Seidel method is a fast solver for systems involving this matrix and this statement holds also true for the ILU iteration.

At this point we mention that the statement of Theorem 6.6 is true for every restriction $\hat{\boldsymbol{R}}^l$ and prolongation $\hat{\boldsymbol{P}}^l$ which are decoupled. The proof of Theorem 6.6 does not use the special structure of $\boldsymbol{R}^l_{\mathcal{V}}$, $\boldsymbol{R}^l_{\mathcal{S}}$ and $\boldsymbol{P}^l_{\mathcal{V}}$, $\boldsymbol{P}^l_{\mathcal{S}}$. This opens additional possibilities for the definition of $\boldsymbol{R}^l_{\mathcal{V}}$ and $\boldsymbol{P}^l_{\mathcal{V}}$ for the main variables $\boldsymbol{u}_{\mathcal{V}}$. In our case we keep $\boldsymbol{R}^l_{\mathcal{V}}$ as defined in equation (29) and take $\boldsymbol{P}^l_{\mathcal{V}}$ as a bilinear interpolation, i.e. equation (30) is not valid any more. This approach provides two advantages. In practice we gain a better convergence rate as shown later. And additionally, we fulfill a necessary condition for convergence on the level of the cell variables as the main variables of the form

$$m_R + m_P > m$$

where m_R is the order of the restriction, m_P is the order of the prolongation and m is the order of the differential equation [18].

7 Numerical results

For our numerical computations we consider different test problems [27, 35, 40] for the linear convection-diffusion equation. The first two problems are steady partial differential equations, but – since we have an additional positive contribution to the diagonal of our discretization matrices in the unsteady case – the numerical solution of these problems is even more difficult.

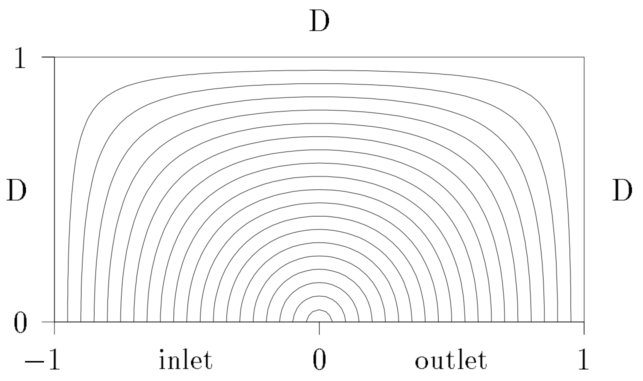

Figure 6: Streamlines and boundary conditions for problem 7.1

Problem 7.1 (Berkeley problem)

$$\boldsymbol{\beta} \cdot \boldsymbol{\nabla} u - \varepsilon \Delta u = 0 \quad \text{in } \Omega = (-1,1) \times (0,1)$$

with $\boldsymbol{\beta} = (-\psi_y, \psi_x)^T$ given by the stream function $\psi = (1-x^2)(1-y^2)$ and

$$\begin{aligned} u(x,y) &= \begin{cases} 1 + \tanh\left(10(2x+1)\right) & \text{on } \Gamma_{\text{in}} := \{(x,y) \in \Gamma \mid -1 \le x \le 0, y = 0)\}\,, \\ 0 & \text{on } \Gamma_D = \Gamma \setminus \{\Gamma_{\text{in}} \cup \Gamma_N\}\,, \end{cases} \\ \partial_n u(x,y) &= 0 \quad \text{on } \Gamma_N := \{(x,y) \in \Gamma \,|\, 0 \le x \le 1, y = 0)\}\,. \end{aligned}$$

Problem 7.2 (Convection-diffusion equation on the unit square)

$$\boldsymbol{\beta} \cdot \boldsymbol{\nabla} u - \varepsilon \Delta u = 0 \quad \text{in } \Omega = (0,1) \times (0,1)$$

with $\boldsymbol{\beta} = (\cos\alpha, \sin\alpha)^T$ and $u(x,y) = x^2 + y^2$ on Γ. The angle α is varying in multiples of $15°$.

Results for the Krylov subspace methods

The next part of the presentation of our numerical results treats the application of the Krylov subspace methods. We like to sketch the behaviour which is typical for these methods and the problems they have to face.

Figure 7 shows results for problem 7.1 on a 48×24 grid, i.e. for 3552 unknowns, and $\varepsilon = 0.1$ for the CGS and the BiCGSTAB algorithms.

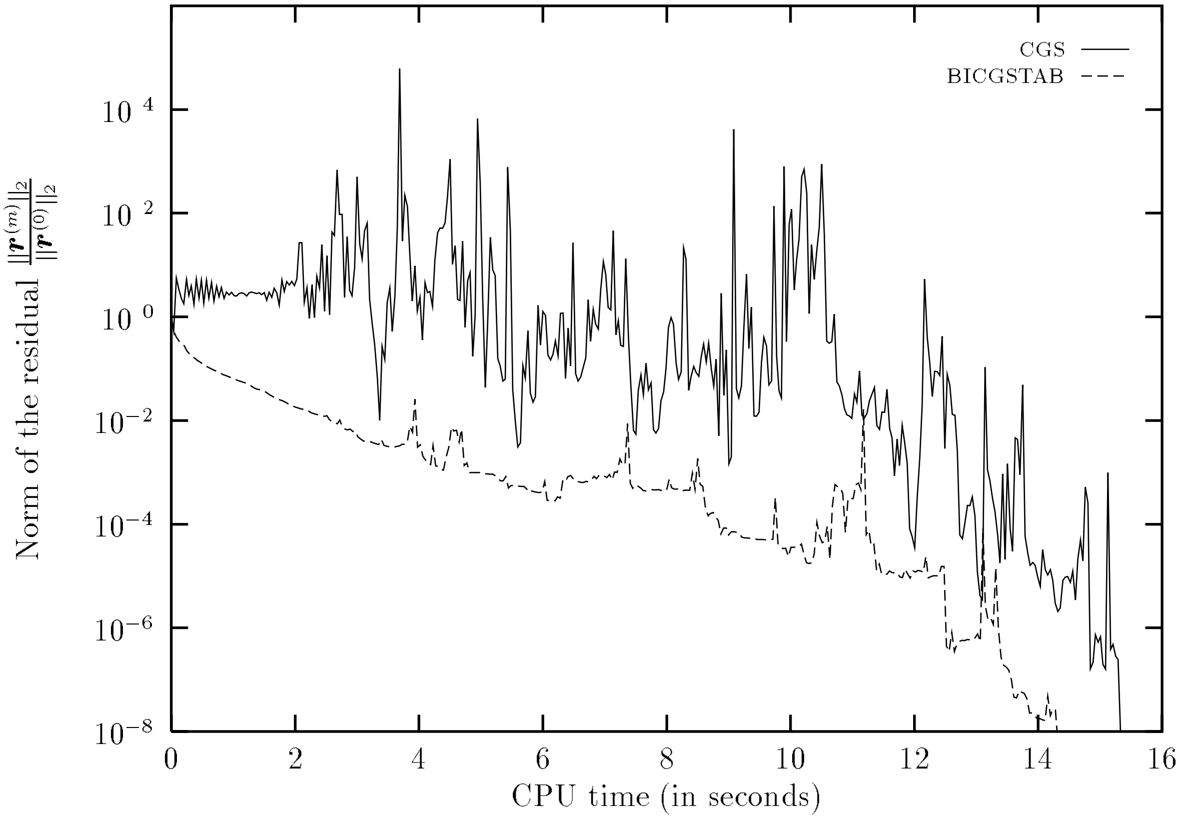

Figure 7: Comparison between the transpose-free variants of the BiCG algorithm

We observe oscillations which are typical for both methods. The residual, especially for the CGS algorithm, sometimes changes abruptly and varies by several orders of magnitude from step to step. In addition, our numerical results confirm the property that the BiCGSTAB method has a smoother convergence behaviour.

In practice the TFQMR and also the QMRCGSTAB algorithm are not using the residual norm as a stopping criteria. This is the reason for depicting the magnitude of $|\tau_m|/\|\boldsymbol{r}^{(0)}\|_2$ and – additionally – the real residual in figures 8 and 9 for these two algorithms.

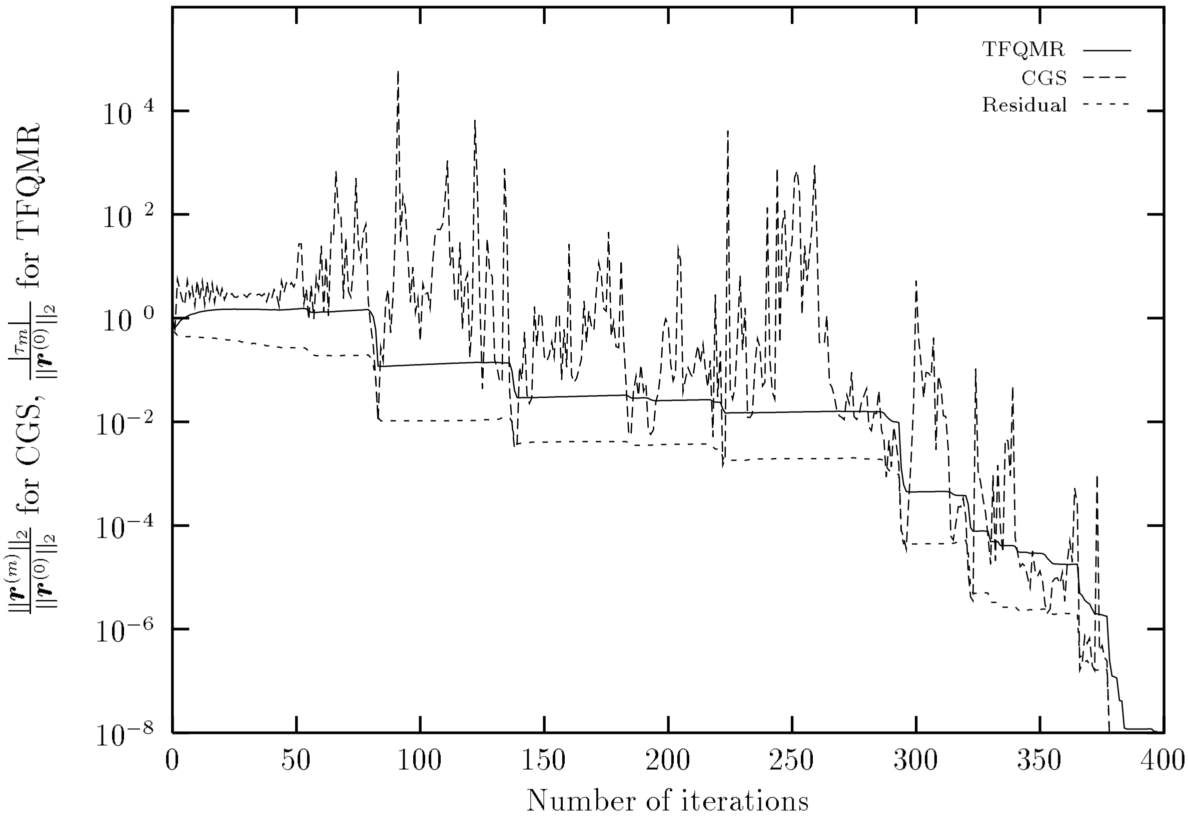

Figure 8: Convergence of CGS and TFQMR

A comparison of the CGS and the TFQMR algorithm is shown in figure 8. The predicted smoother convergence of the TFQMR method is confirmed. Additionally, we observe that the TFQMR method needs, approximately, the same number of steps.

For the QMRCGSTAB algorithm, as the quasi-minimal variant of the BiCGSTAB method, we obtain a smoother convergence, too. Figure 9 shows, clearly, that the effect of the QMRCGSTAB method is not as impressive as the one of the TFQMR method on the residuals of CGS algorithm. Since the BiCGSTAB algorithm already improves the erratic residual behaviour of the CGS method this is not surprising.

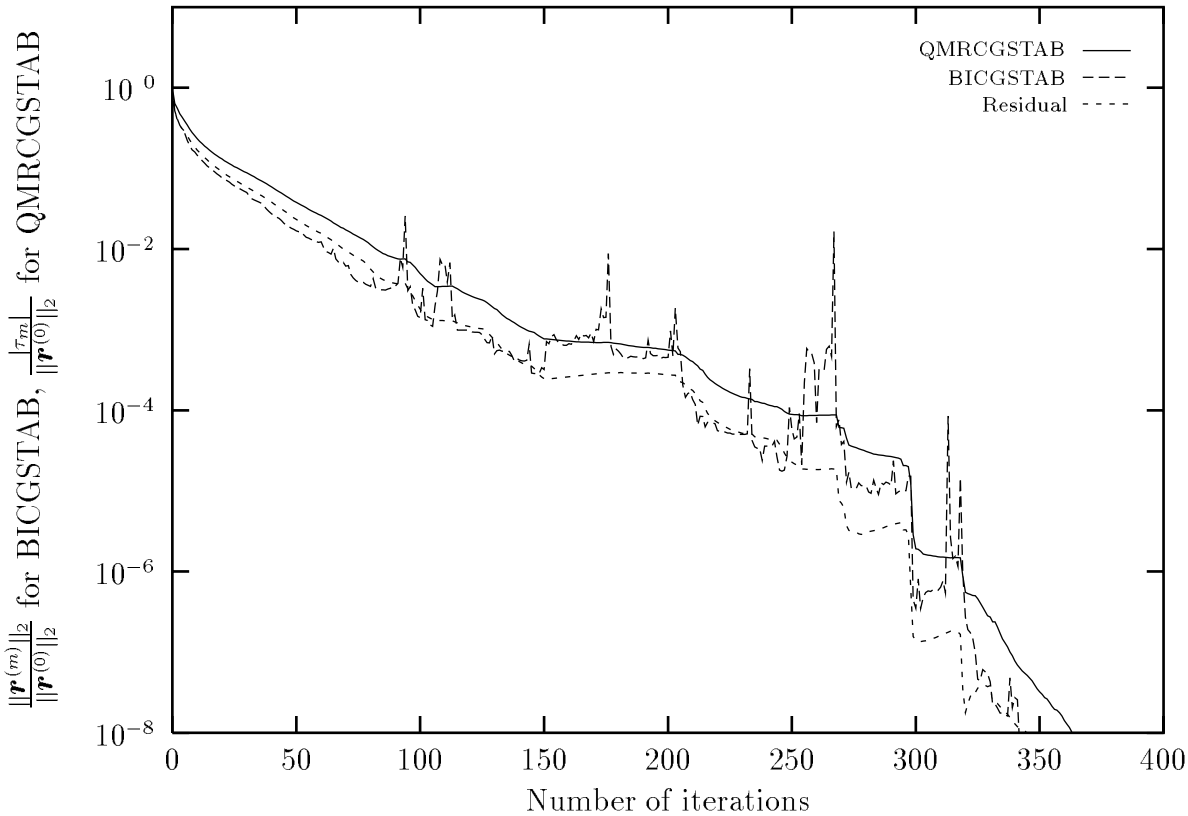

Figure 9: Convergence of BiCGSTAB and QMRCGSTAB

Since the convergence rate of the methods becomes worse with decreasing grid size we use preconditioned versions of the Krylov subspace methods in combination with an ILU decomposition for the subsequent computations.

For our numerical computations we define an averaged convergence rate via

$$\varrho_M := \left(\frac{\|\boldsymbol{r}^{(m)}\|_2}{\|\boldsymbol{r}^{(0)}\|_2} \right)^{\frac{1}{m}}$$

where m is the number of iterations.

In figure 10 results for problem 7.1 on a 96×48 grid with 13968 unknowns for the preconditioned versions of the CGS, the BiCGSTAB and the restarted GMRES method with the dimension of the Krylov subspace of 10 are depicted. We observe that the convergence rate approaches 1 with increasing diffusion coefficient. Similar behaviour is shown by the quasi-minimal variants, the TFQMR and the QMRCGSTAB algorithm.

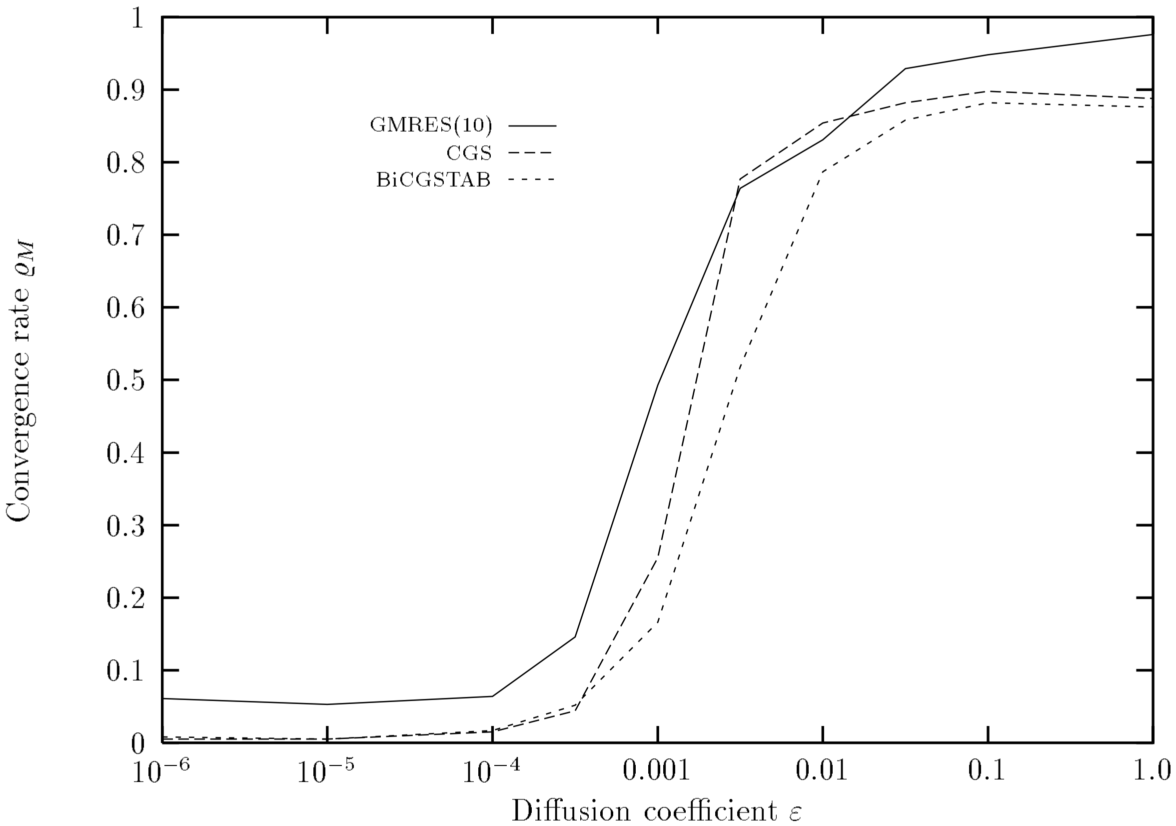

Figure 10: Convergence rate for varying diffusion coefficient

For problem 7.2 on a 64×64 grid with 12416 unknowns we show that the quality of the solution methods is independent of the direction of the flow field $\boldsymbol{\beta}$ (see table 11). Note that the results obtained for different versions of our multigrid method will be given in the subsequent section (see tables 13 and 14). The BiCGSTAB algorithm seems to be the fastest method, but only by a small margin. In general the convergence behaviour of the Krylov subspace methods is relatively bad, especially for problems with a large diffusion coefficient.

Method	$\varepsilon \setminus \alpha$	0^o	15^o	30^o	45^o	60^o	75^o	90^o	135^o	-45^o
GMRES(10)	1	.950	.955	.955	.956	.955	.955	.950	.965	.965
	0.1	.936	.945	.935	.946	.935	.945	.936	.954	.954
	0.01	.883	.870	.858	.866	.858	.870	.883	.897	.897
	0.001	.613	.498	.489	.478	.489	.498	.613	.541	.541
GMRES(15)	1	.949	.949	.949	.949	.949	.949	.949	.949	.949
	0.1	.920	.924	.927	.924	.927	.924	.920	.934	.934
	0.01	.886	.888	.883	.884	.883	.888	.886	.909	.909
	0.001	.521	.325	.316	.337	.316	.325	.521	.463	.463
CGS	1	.877	.864	.869	.867	.860	.869	.878	.865	.865
	0.1	.835	.812	.824	.834	.837	.834	.831	.844	.864
	0.01	.812	.829	.799	.845	.790	.839	.887	.826	.881
	0.001	.342	.212	.251	.295	.251	.212	.342	.278	.278
BiCGSTAB	1	.838	.838	.826	.819	.818	.829	.836	.811	.825
	0.1	.810	.810	.861	.800	.817	.803	.803	.814	.821
	0.01	.718	.752	.747	.772	.752	.748	.724	.747	.779
	0.001	.373	.233	.247	.264	.224	.311	.369	.258	.252
QMRCGSTAB	1	.853	.855	.844	.841	.836	.853	.853	.828	.837
	0.1	.842	.849	.849	.833	.849	.824	.849	.828	.825
	0.01	.761	.774	.764	.793	.762	.774	.755	.777	.756
	0.001	.450	.369	.325	.348	.333	.322	.439	.318	.311

Table 11: Comparison of preconditioned versions for problem 7.2

Of particular interest are the numerical results for an increasing number of unknowns. Since the BiCGSTAB algorithm performs best we examine its convergence behaviour for problem 7.1 with varying grid size (see figure 12). For very small gridsizes the convergence rate approaches 1 which means that the algorithm converges very slowly. Thus, it is questionable whether Krylov subspace methods with a relatively simple preconditioning, at least the selection tested here, are of any use for very large systems with more than 10^5 unknowns.

This behaviour was the main reason for the development of the multigrid method which shows a very good convergence rate – independently of the gridsize h.

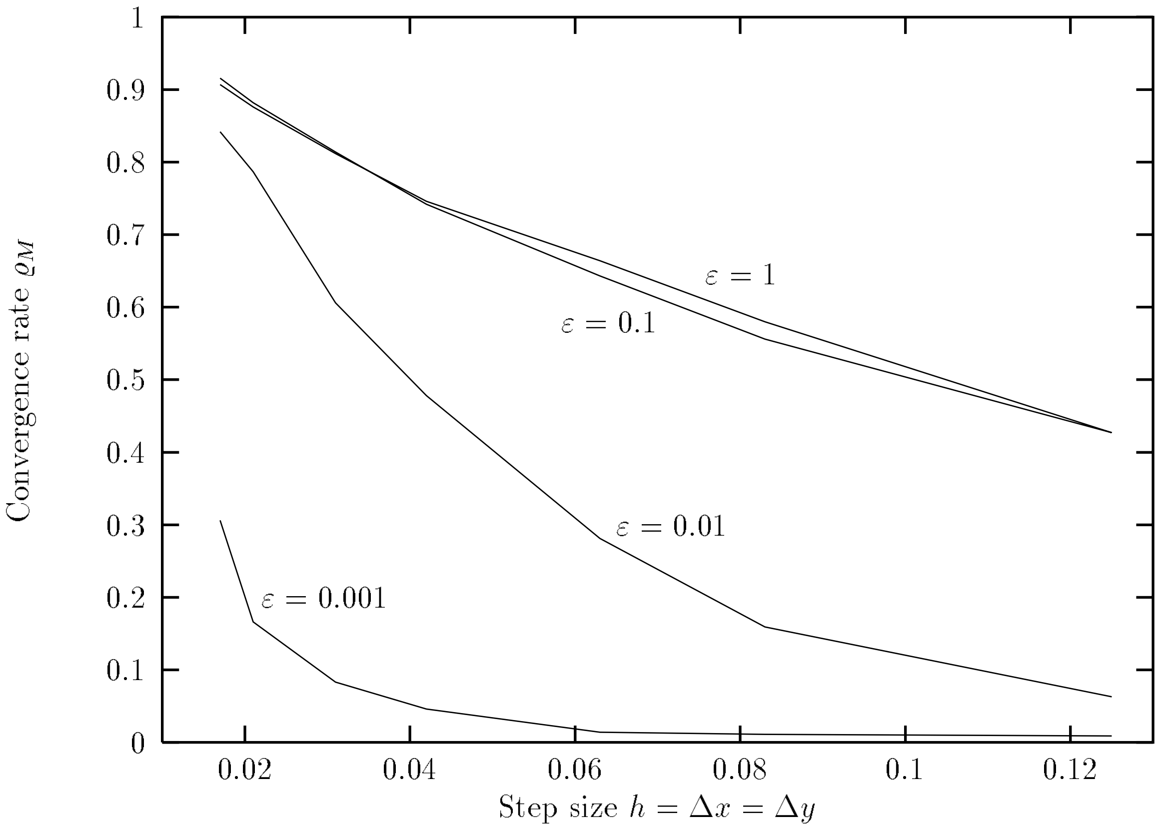

Figure 12: Convergence rate of the BiCGSTAB method for varying gridsize

Results for the multigrid algorithms

This section presents the results for different versions of the multigrid algorithm of Chapter 6. As a smoother we consider the Gauss-Seidel and ILU iteration, which are exact solvers in the convection case, with a numbering in the direction of the velocity field. This governs the excellent behaviour of the multigrid method in convection dominated cases. On the other hand the Gauss-Seidel method (GS) leads to low efficiency for diffusion dominated problems. This difficulty can be weakened by choosing a relaxation factor greater than 1, e.g. $\omega = 1.25$, in the SOR method.

The averaged reduction factor is defined by

$$\varrho_M := \left(\frac{\|\boldsymbol{r}^{(\kappa)}\|_2}{\|\boldsymbol{r}^{(0)}\|_2} \right)^{\frac{1}{\kappa}} \quad \text{with } \kappa = \text{number of multigrid cycles}\,.$$

We illustrate the behaviour of different smoothers for problem 7.2 with various parameters on a 64×64 grid with 12416 unknowns. Table 13 shows the reduction factor ϱ_M for the finite volume transfer operators and table 14 for the operators corresponding to the Petrov-Galerkin interpretation. For all computations we set the number of pre- and

smoothing iteration	$\varepsilon \setminus \alpha$	$0°$	$15°$	$30°$	$45°$	$60°$	$75°$	$90°$	$135°$	$-45°$
ILU	1	.268	.272	.272	.272	.272	.272	.268	.258	.258
	0.1	.215	.215	.221	.223	.221	.215	.215	.225	.225
	0.01	.087	.097	.085	.079	.085	.097	.087	.096	.097
	0.001	.028	.010	.005	.003	.005	.010	.028	.022	.022
GS	1	.911	.908	.906	.905	.906	.908	.911	.905	.905
	0.1	.870	.855	.845	.840	.845	.855	.869	.842	.842
	0.01	.369	.352	.324	.310	.324	.352	.369	.345	.345
	0.001	.087	.047	.025	.015	.025	.047	.087	.061	.061
SOR ($\omega = 1.25$)	1	.720	.702	.704	.701	.704	.702	.720	.700	.700
	0.1	.630	.610	.594	.586	.594	.610	.630	.556	.556
	0.01	.260	.193	.170	.162	.170	.193	.260	.161	.161

Table 13: Finite volume restriction and prolongation

smoothing iteration	$\varepsilon \setminus \alpha$	$0°$	$15°$	$30°$	$45°$	$60°$	$75°$	$90°$	$135°$	$-45°$
ILU	1	.538	.538	.538	.538	.538	.538	.538	.542	.542
	0.1	.421	.429	.445	.455	.445	.429	.421	.457	.457
	0.01	.143	.136	.152	.157	.152	.136	.143	.190	.190
	0.001	.024	.008	.006	.007	.006	.008	.024	.014	.014
GS	0.1	.675	.695	.643	.633	.643	.695	.675	.634	.634
	0.01	.407	.370	.278	.221	.278	.371	.407	.285	.285
	0.001	.076	.044	.018	.016	.018	.044	.076	.032	.032
SOR ($\omega = 1.25$)	1	.921	.917	.914	.913	.914	.917	.921	.913	.913
	0.1	.557	.559	.546	.539	.546	.559	.557	.544	.544
	0.01	.226	.136	.110	.108	.108	.136	.226	.118	.118

Table 14: Petrov-Galerkin restriction and prolongation

postsmoothing iterations to $\nu_1 = \nu_2 = 2$ in the case of ILU smoothing and $\nu_1 = \nu_2 = 4$ for the Gauss-Seidel and the SOR method as a smoother.

Comparing tables 13 and 14 we observe that the convergence for restriction and prolongation of the finite volume type is much better than in the Petrov-Galerkin case. As expected the reduction factor is very good for ε small. Anyway we have to mention that the multigrid method does not converge for the SOR method and $\varepsilon = 0.001$ as well as in the Petrov-Galerkin case for the Gauss-Seidel iteration and $\varepsilon = 1$. The crucial point for the choice of the smoothing iteration is the reduction factor for the diffusion dominated case ($\varepsilon = 0.1, 1$). With the aim of robustness we prefer the ILU method as a smoothing iteration which is used in all subsequent computations.

In addition, the different columns of these tables indicate that the results are essentially independent of the velocity direction, i.e. of the angle α.

That we can gain from using a bilinear prolongation for the cell variables is shown in table 15. The number of multigrid V-cycles is given in brackets additionally. The results have to be compared with the results for the original multigrid algorithm in table 19. For large diffusion coefficient and on very fine grids we can save up to one third of the V-cycles needed for convergence.

Method	$\varepsilon \setminus n_1$	16	32	64	128
MGM/BIL	1	.163(11)	.166(11)	.170(11)	.173(11)
	0.1	.126 (9)	.168(11)	.200(12)	.198(12)
	0.01	.010 (4)	.045 (6)	.099 (8)	.141(10)
	0.001	.001 (2)	.001 (3)	.006 (4)	.020 (5)

Table 15: Results of the multigrid method with bilinear prolongation

Note that ILU smoothing in connection with the original finite volume restriction and prolongation is used for the remaining computations of this section.

One of the main features of a multigrid method is that the convergence rate is uniformly bounded by some number smaller than 1. Our numerical experiments for problem 7.1 with varying grid size h and different values of ε shown in figure 16 are confirming this property. The largest linear systems in our computations had 221952 unknowns on a 384×192 grid. For fixed grid size h the convergence rate is increasing in ε. The gap between the convergence rate for $\varepsilon = 1$ and the other ones becomes smaller on finer grids. The uniform boundedness in h and – in the sense of robustness – even in ε is determined by the top line which seems to be bounded by 0.4.

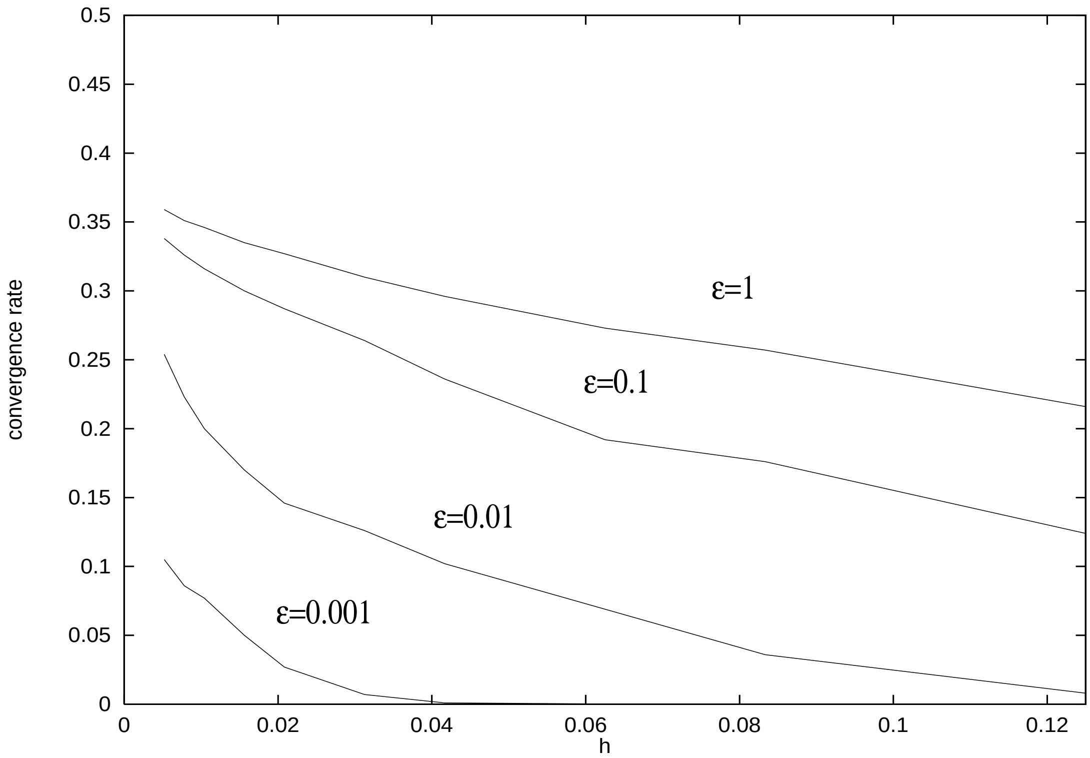

Figure 16: Convergence rates for varying diffusion parameters

Finally, we present results of the full multigrid algorithm in table 17. Again, we have to compare this with table 19. We observe that we gain from nested iteration. The reason is that a better starting vector on the finest grid is computed.

Method	$\varepsilon \setminus n_1$	16	32	64	128
FMG	1	.149(11)	.173(12)	.208(12)	.239(13)
	0.1	.097 (9)	.131(10)	.160(11)	.196(12)
	0.01	.005 (4)	.024 (5)	.055 (7)	.091 (8)
	0.001	.001 (2)	.001 (3)	.007 (4)	.008 (4)

Table 17: Results of the full multigrid algorithm

Results for the preconditioned versions

This subsection presents numerical results for problem 7.2 with a varying diffusion parameter ε.

Since the restarted GMRES algorithm is a convergent method for our discretization theoretically we examine its behaviour in combination with a multigrid algorithm. As a preconditioner we use one V-cycle with ILU smoothing and the finite volume transfer operators.

First of all we vary the dimension of the Krylov subspace for different preconditioned versions of the restarted GMRES method (see table 18). We observe that the multigrid preconditioning is very efficient in comparison to an ILU preconditioner and that we need just a small dimension of the Krylov subspace to obtain a fast convergent method. The last statement is very important since we do not have much additional storage and computational costs for the GMRES method with multigrid preconditioning in comparison to the multigrid method itself. Thus, the storage requirements are governed by the preconditioning mainly.

Method	$\varepsilon \setminus m$	2	5	7	10	12	15
GMRES/ILU	1	.997	.986	.962	.955	.952	.949
	0.1	.996	.971	.953	.935	.930	.927
	0.01	.995	.830	.857	.866	.867	.884
	0.001	.466	.541	.443	.478	.438	.337
GMRES/MGM	1	.983	.973	.387	.101	.101	.101
	0.1	.983	.972	.408	.121	.121	.121
	0.01	.954	.149	.056	.056	.056	.056
	0.001	.006	.005	.005	.005	.005	.005

Table 18: Comparison of GMRES/ILU and GMRES/MGV

Table 19 shows the results for different grid sizes and varying diffusion parameter ε. We present the results for the multigrid algorithm and the restarted GMRES method with multigrid preconditioning. In addition, we show the BiCGSTAB and QMRCGSTAB methods, each of them equipped with a multigrid preconditioning. If we look at the number of V-cycles needed for convergence we ascertain that multigrid is a good preconditioner. A combination of a restarted GMRES algorithm with a moderate dimension of the Krylov subspace leads to a very efficient solution method for our linear systems.

Method	$\varepsilon \setminus n_1$	16	32	64	128
MGM	1	.172(12)	.214(13)	.261(14)	.333(17)
	0.1	.118 (9)	.162(11)	.206(12)	.258(14)
	0.01	.007 (4)	.032 (6)	.071 (8)	.123 (9)
	0.001	.001 (2)	.001 (3)	.005 (4)	.009 (4)
GMRES/MGM	1	.072 (8)	.090 (9)	.101(10)	.127(11)
	0.1	.068 (9)	.098 (9)	.121(10)	.135(11)
	0.01	.006 (5)	.031 (7)	.056 (8)	.092 (9)
	0.001	.001 (3)	.001 (4)	.005 (4)	.007 (5)
BICGSTAB/MGM	1	.009 (8)	.035(12)	.067(14)	.099(16)
	0.1	.009 (8)	.016(10)	.068(14)	.530(60)
	0.01	.001 (4)	.001 (6)	.005 (8)	.015(10)
	0.001	.001 (2)	.001 (4)	.001 (4)	.001 (4)
QMRCGSTAB/MGM	1	.029(12)	.083(16)	.127(20)	.186(22)
	0.1	.030(12)	.069(14)	.197(24)	.586(72)
	0.01	.001 (6)	.010(10)	.043(12)	.069(14)
	0.001	.001 (4)	.001 (6)	.001 (8)	.009 (8)

Table 19: The multigrid algorithm and different preconditioned algorithms

8 Conclusion

This article presents an overview on recent solution methods for large and sparse linear systems. These methods are applied to systems which arise from a nonconforming Petrov-Galerkin discretization of convection-diffusion problems. The numerical results show clearly that – especially for large linear systems – Krylov subspace algorithms equipped with a multigrid preconditioning lead to very efficient and robust solution methods. This statement holds true in the case of the GMRES algorithm particularly.

Acknowledgements

The author would like to express his thankfulness to Prof. Dr. W. Törnig for his excellent support over the last years.

Very special thanks are due to Dr. Dietmar Hietel for years of fruitful collaboration in the field of computational fluid dynamics.

The author likes to thank also Dipl.-Math. Mario Breuer for his help in programming some of the Krylov subspace methods presented here.

References

[1] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, Cambridge 1994.

[2] N. S. Bachvalov. *On the convergence rate of a relaxation method with natural constraints on the elliptic operator.* USSR Comput. Math. and Math. Phys. **6** (1966) 101–135.

[3] A. Brandt. *Multi-level adaptive techniques (MLAT) for the fast numerical solution to boundary value problems.* In H. Cabannes, R. Temam, *Proc. 3rd Int. Conf. on Numerical Methods in Fluid Mechanics*, Lect. Notes Phys., 82–89. Academic Press, Berlin 1973.

[4] A. Brandt. *Multi-Level Adaptive Computations in Fluid Dynamics.* AIAA J. **18** (1980) 1165–1172.

[5] K. E. Brenan, S. L. Campbell, L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics In Appl. Math. 14. SIAM, Philadelphia 1996.

[6] S. Brenner, L. Scott. *The Mathematical Theory of Finite Element Methods*, Texts in Appl. Math. 15. Springer, New York 1994.

[7] T. F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, C. H. Tong. *A quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric systems.* SIAM J. Sci. Comput. **15**(2) (1994) 338–347.

[8] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam 1978.

[9] R. P. Fedorenko. *The speed of convergence of one iterative process.* USSR Comput. Math. and Math. Phys. **4** (1964) 227–235.

[10] R. Fletcher. *Conjugate gradient methods for indefinite systems.* In G. A. Watson, *Proceedings of Dundee Biennal Conference on Numerical Analysis 1974*, 73–89. Springer-Verlag, New York 1975.

[11] M. Franzke. *L-Spline Petrov-Galerkin-Verfahren für singulär gestörte Probleme.* Diplomarbeit, Technische Hochschule Darmstadt 1996.

[12] R. W. Freund. *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems.* SIAM J. Sci. Comput. **14**(2) (1993) 470–482.

[13] R. W. Freund, N. M. Nachtigal. *QMR: a quasi-minimal residual method for non-Hermitian linear systems.* Numer. Math. **60** (1991) 315–339.

[14] G. H. Golub, D. P. O'Leary. *Some history of the conjugate gradient and Lanczos algorithms: 1948-1976.* SIAM Rev. **31** (1989) 50–102.

[15] D. F. Griffiths, A. R. Mitchell. *Nonconforming elements.* In D. F. Griffiths, *The mathematical basis of finite element methods*, Oxford Science Publications. Oxford University Press, Oxford 1984.

[16] W. Guo. *Uniformly Convergent Finite Element Methods for Singular Perturbed Parabolic Partial Differential Equations.* Ph.D. thesis, University College Cork, Ireland 1993.

[17] W. Hackbusch. *Ein iteratives Verfahren zur schnellen Auflösung elliptischer Randwertprobleme.* Report 76-12, Universität Köln 1976.

[18] W. Hackbusch. *Multi-Grid Methods and Applications*, Springer Ser. Comput. Math. 4. Springer, Berlin 1985.

[19] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*, Appl. Math. Sciences 95. Springer, New York 1994.

[20] W. Hackbusch, U. Trottenberg. *Multigrid Methods I.* Lect. Notes Math. Springer, Berlin 1982.

[21] W. Hackbusch, U. Trottenberg. *Multigrid Methods II.* Lect. Notes Math. Springer, Berlin 1986.

[22] W. Hackbusch, U. Trottenberg. *Multigrid Methods III*, ISNM, Int. Ser. Numer. Math. 98. Birkhäuser, Basel 1991.

[23] W. Hackbusch, G. Wittum. *Incomplete Decompositions (ILU) - Algorithms, Theory, and Applications*, Notes on Numer. Fluid Mechanics 41. Vieweg, Braunschweig 1993.

[24] P. W. Hemker, P. Wesseling. *Contributions to Multigrid*, CWI tract 103. CWI, Amsterdam 1994.

[25] P. W. Hemker, P. Wesseling. *Multigrid Methods IV*, ISNM, Int. Ser. Numer. Math. 116. Birkhäuser, Basel 1994.

[26] M. Hestenes, E. Stiefel. *Methods of conjugate gradients for solving linear systems.* J. Res. Nat. Bur. Standards **49** (1952) 409–436.

[27] D. Hietel. *Ein neuer Typ zellorientierter Diskretisierungsverfahren zur numerischen Lösung von Konvektions-Diffusions-Problemen.* Ph.D. thesis, Technische Hochschule Darmstadt 1991.

[28] D. Hietel, J. Witzel. *A new finite-volume approach for the solution of convection-diffusion problems.* Preprint No. 1948, Fachbereich Mathematik, Technische Universität Darmstadt 1997.

[29] D. Hietel, J. Witzel. *Proving the definiteness of linear systems which arise from a nonconforming Petrov-Galerkin discretization of convection-diffusion problems.* Preprint (in preparation), Fachbereich Mathematik, Technische Universität Darmstadt 1998.

[30] C. Lanczos. *Solution of systems of linear equations by minimized iterations.* J. Res. Nat. Bur. Standards. **49** (1952) 33–53.

[31] T. A. Manteuffel. *An incomplete factorization technique for positive definite linear systems.* Math. Comput. **34** (1980) 473–479.

[32] S. F. McCormick. *Multigrid Methods*, Frontiers in Applied Mathematics 3. SIAM, Philadelphia 1987.

[33] J. Meijerink, H. Van der Vorst. *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix.* Math. Comput. **31** (1977) 148–162.

[34] A. Meister. *Zur zeitgenauen numerischen Simulation reibungsbehafteter, kompressibler, turbulender Strömungen mit einer impliziten Finite-Volumen-Methode vom Box-Typ.* Ph.D. thesis, Technische Hochschule Darmstadt 1996.

[35] O. Pironneau. *Finite Element Methods for Fluids.* John Wiley & Sons, Chichester 1989.

[36] J. K. Reid. *On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations.* In J. K. Reid, *Large Sparse Sets of Linear Equations*, 231–254. Academic Press, London 1971.

[37] H.-G. Roos, M. Stynes, L. Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations*, Springer Ser. Comput. Math. 24. Springer, Berlin 1996.

[38] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston 1996.

[39] Y. Saad, M. H. Schultz. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems.* SIAM J. Sci. Stat. Comput. **7** (1986) 856–869.

[40] P. Sonneveld. *CGS, a fast Lanczos-type solver for nonsymmetric linear systems.* SIAM J. Sci. Stat. Comput. **10** (1989) 36–52.

[41] W. Törnig, P. Spellucci. *Numerische Mathematik für Ingenieure und Physiker, Band 1.* Springer-Verlag, Berlin, Heidelberg, New York 1988.

[42] 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. **12** (1992) 631–644.

[43] H. A. van der Vorst, K. Dekker. *Conjugate gradient type methods and preconditioning.* J. Comput. Appl. Math. **24** (1988) 73–87.

[44] P. Wesseling. *An Introduction to Multigrid Methods.* John Wiley & Sons, Chichester 1991.

[45] J. Witzel. *Präkonditionierte Petrov-Galerkin-Verfahren zur Lösung linearer Gleichungssysteme bei der Kopplung Finiter Methoden.* Diplomarbeit, Technische Hochschule Darmstadt 1992.

[46] J. Witzel. *Numerical Solution of Linear Systems Arising from a New Petrov-Galerkin Discretization of Convection-Diffusion Problems* . Shaker Verlag, Aachen 1997. Ph.D. thesis.

[47] S. Zeng, P. Wesseling. *Multigrid for the Incompressible Navier-Stokes Equations on Staggered Grids in General Coordinates.* In P. W. Hemker, P. Wesseling, *Contributions to Multigrid*, CWI tract, 203–218. CWI, Amsterdam 1994.