Algebraic multigrid methods for the solution of the Navier-Stokes equations in complicated geometries

Michael Griebel

Institut für Angewandte Mathematik
Abteilung Wissenschaftliches Rechnen und Numerische Simulation
Universität Bonn
Wegelerstraße 6
53115 Bonn

Tilman Neunhoeffer, and Hans Regler

Institut für Informatik
Technische Universität München
D-80290 München

email: griebel@iam.uni-bonn.de, neunhoef@informatik.tu-muenchen.de,
regler@informatik.tu-muenchen.de

Abstract

The application of standard multigrid methods for the solution of the Navier-Stokes equations in complicated domains causes problems in two ways. First, coarsening is not possible to full extent since the geometry must be resolved by the coarsest grid used, and second, for semi-implicit time stepping schemes, robustness of the convergence rates is usually not obtained for the arising convection-diffusion problems, especially for higher Reynolds numbers.

We show that both problems can be overcome by the use of algebraic multigrid (AMG) which we apply for the solution of the pressure and momentum equations in explicit and semi-implicit time-stepping schemes.

We consider the convergence rates of AMG for several model problems and we demonstrate the robustness of the proposed scheme.

Key words: Navier-Stokes Equations, SIMPLE-Algorithm, Algebraic Multigrid Methods

1 Introduction

In this paper, we consider a fast solver for the numerical simulation of two- and three dimensional viscous, instationary, incompressible fluid flow problems in complicated geometries as they arise for example in the study of porous media flow on a micro-scale level, in multi-connected technical devices like cooling or heating systems or in a vast number of biological and medical flow simulations. Also for free boundary problems where the domain changes in time, partial differential equations in rather complicated geometries have to be solved in each time step.

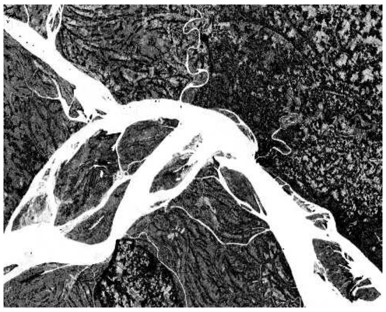
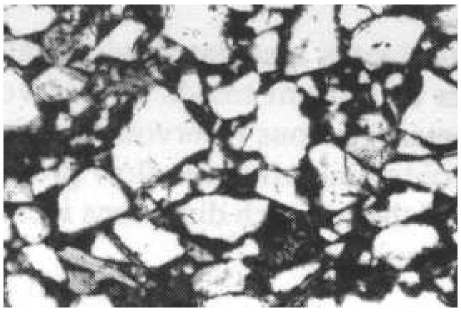

Fig. 1 Examples for flow problems in fixed complicated geometries.
Left: a river system, Right: porous media (cross section of sandstone) [Fayers & Hewett, 1992]

If we use an explicit time discretization, for example the forward Euler scheme, most of the computational effort has to be spent for the solution of the Poisson equation in a pressure correction step. Semi-implicit discretization schemes as for example the backward Euler scheme allow larger time steps. But here, besides the Poisson equation, we additionally obtain convection-diffusion equations for each component of the velocity.
Multigrid methods are often used solvers for the arising algebraic equations. However, for convection dominated convection-diffusion problems, as they appear for high Reynolds numbers, standard multigrid methods show a lack of robustness with respect to the convergence behavior. This can be overcome by some special techniques for the construction of the restriction and coarse grid operators and the use of ILU-smoothers [Kettler, 1982], [Wittum, 1989a], [Wittum, 1989b], [de Zeeuw, 1990], [Reusken, 1994]. This only works well in two spatial dimensions. The second drawback is that the geometry of the domain must be resolved on the coarsest level of discretization used in the multigrid method. Thus, the domain is not allowed to have a complicated structure. Otherwise, there would be many unknowns on the coarsest level and an iterative scheme would need too many smoothing steps on the coarsest grid to maintain good convergence rates whereas direct solvers for the coarse grid equation are too expensive. Furthermore, it is not possible to use grid transformation techniques to transform a non-rectangular physical domain into a rectangular computational domain if the domain is too complicated. For some other concepts on the application of multigrid methods to problems on complex domains, see [Bank & Xu, 1995], [Bank & Xu, 1996], [Hackbusch & Sauter, 1995], or [Kornhuber & Yserentant, 1994].
In the eigthies, algebraic multigrid methods were introduced. They do not make use of any geometric information on the grid. Here, the coarse grid points and restriction and interpolation operators are constructed by only considering the linear system and the coupling between the different unknowns. In numerical experiments, algebraic multigrid has been shown to possess advantages on conventional multigrid methods with respect to *robustness* [Ruge & Stüben, 1984], [Ruge & Stüben, 1986]. Their convergence rates are bounded by a constant $C < 1$ independent of the PDE under consideration, also for problems with strongly varying coefficient functions or singular perturbed problems, like diffusion problems or convection-diffusion problems with strong anisotropy or strong convection, respectively. Robust convergence rates are also obtained for problems on domains with *complicated geometry*, even for the additive variant of AMG (see [Grauschopf et al., 1996]).

We apply AMG to the equations arising from explicit and semi-implicit time-discretizations of the Navier-Stokes equations and we present the results of numerical experiments where we study the dependence of the convergence rates on the geometry, on the Reynolds number, and on the number of unknowns. For other concepts for the application of AMG to the Navier-Stokes equations, see [Lonsdale, 1993], [Webster, 1994], and [Raw, 1994].

2 Discretization of the Navier-Stokes Equations

The Navier-Stokes Equations

We consider the time-dependent, incompressible Navier-Stokes equations for the velocity $\vec{u}$ and the kinematic pressure p, which is defined as the real pressure divided by density, in an arbitrary bounded domain $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$

$$\begin{aligned} \vec{u}_t - \frac{1}{Re}\triangle\vec{u} + \vec{u}\cdot\nabla\vec{u} + \nabla p &= \vec{g}, && \text{(2a)} \\ \nabla\cdot\vec{u} &= 0. && \text{(2b)} \end{aligned}$$

(2a) is the momentum equation and (2b) is the continuity equation. Re denotes the Reynolds number and $\vec{g}$ is the body force, for example gravity. In addition, we need suitable initial conditions $\vec{u}|_{t=0} = \vec{u}_0$ and boundary conditions of inflow, outflow, slip or no-slip type. This means that either the velocity itself is specified at the boundary ($\vec{u}|_\Gamma = \vec{u}_\Gamma$) or its normal derivative $((\partial\vec{u}/\partial n)|_\Gamma = (\partial\vec{u}/\partial n)_\Gamma)$ where $\vec{n}$ denotes the unit outer normal vector at the boundary Γ. For a detailed description of the different boundary types, see e.g. [Hirt et al., 1975], [Griebel et al., 1995, pp 12f].

The initial condition must satisfy $\nabla\cdot\vec{u}_0 = 0$ and $\vec{u}_0|_\Gamma\cdot\vec{n} = \vec{u}_\Gamma|_{t=0}\cdot\vec{n}$ (see e.g. [Quaterpelle, 1993]). An initial velocity field $\vec{u}_0$ satisfying $\nabla\cdot\vec{u}_0 = 0$ can be obtained by solving the potential equation

$$\triangle\Phi = 0, \qquad \nabla\Phi|_\Gamma\cdot\vec{n} = \vec{u}_\Gamma|_{t=0}\cdot\vec{n} \tag{3}$$

and setting

$$\vec{u}_0 := \nabla\Phi. \tag{4}$$

Discretization in Space

For the space discretization, we use finite differences on a staggered grid with equidistant orthogonal gridlines, which was first introduced by [Harlow & Welch, 1965]. The convective parts are discretized by flux-blending, i.e. a mixture of central and upwind differences, namely the Donor-Cell scheme, such that the discretization does not suffer from instabilities [Hirt et al., 1975].[1] If Ω is non-rectangular, we approximate Ω by a domain Ω_h such that the boundary of Ω_h coincides with gridlines. Then, we imbed Ω_h in a rectangular domain $\tilde{\Omega} \supset \Omega_h$. Thus, $\tilde{\Omega}$ can be divided in the set of fluid cells representing Ω_h and a set of boundary cells (see Figure 2).

Details of the discretization in space and of the discretization of the boundary conditions can be found in [Hirt et al., 1975], [Griebel et al., 1995, chapter 3].

[1] There exist several other stable discretization schemes for convection-diffusion problems, as streamline- and upwind-diffusion methods, see [Hughes et al., 1986], [Johnson, 1987], or Petrov-Galerkin methods.

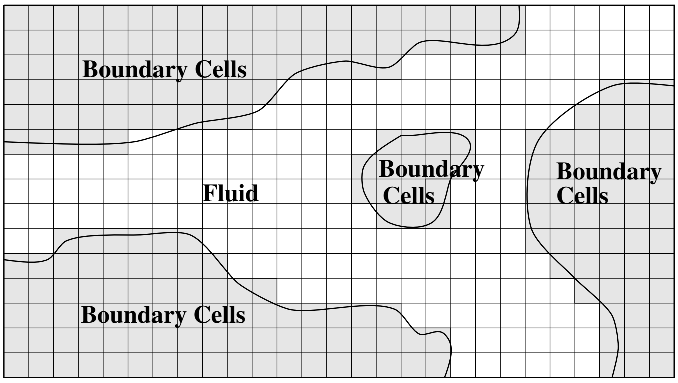

Figure 2: Imbedding of a non-rectangular domain

Discretization in Time

For time discretization, we either use the forward Euler or the backward Euler scheme.[2] The explicit forward Euler scheme with time step $\delta t := t_{n+1} - t_n$, leads to the coupled problem:

Find $\vec{u}^{\,(n+1)}, p^{(n+1)}$ such that

$$\begin{aligned} \vec{u}^{\,(n+1)} + \delta t \nabla p^{(n+1)} &= \vec{u}^{\,(n)} + \delta t \left(\frac{1}{Re} \triangle \vec{u}^{\,(n)} - \vec{u}^{\,(n)} \cdot \nabla \vec{u}^{\,(n)} + \vec{g}^{\,(n)} \right), && \text{(5a)} \\ \nabla \cdot \vec{u}^{\,(n+1)} &= 0, && \text{(5b)} \end{aligned}$$

where the index (n) denotes velocity and pressure at the time t_n.
For its solution, we first choose a tentative velocity field

$$\vec{u}^{\,\star} := \vec{u}^{\,(n)} + \delta t \left(\frac{1}{Re} \triangle \vec{u}^{\,(n)} - \vec{u}^{\,(n)} \cdot \nabla \vec{u}^{\,(n)} + \vec{g}^{\,(n)} \right) \tag{6}$$

and then we obtain $\vec{u}^{\,(n+1)}$ by adding the gradient of the pressure in the new time step

$$\vec{u}^{\,(n+1)} = \vec{u}^{\,\star} - \delta t \nabla p^{(n+1)}. \tag{7}$$

Substituting $\vec{u}^{\,(n+1)}$ by the right hand side of (7) in the continuity equation (5b) leads to a Poisson equation for the pressure

$$\triangle p^{(n+1)} = \frac{1}{\delta t} \nabla \cdot \vec{u}^{\,\star}. \tag{8}$$

Thus, we first solve (8) and then compute $\vec{u}^{\,(n+1)}$ by means of (7).
For reasons of stability of the time stepping scheme, we must observe some restrictive conditions for the step size δt, the so called Courant-Friedrichs-Levi conditions, which guarantee that no particle of the fluid can pass more than one gridline in any direction in one timestep, i.e.

[2] There also exist more elaborated time stepping schemes, even of higher order, as the Crank-Nicholson scheme or the fractional-step-θ-scheme (see e.g. [Turek, 1995]). Note that also in those schemes, basically Poisson equations and convection-diffusion equations have to be solved.

$$\max_{x \in \Omega} |u_i(x)| \, \delta t < \delta x_i \quad i = 1, 2(, 3). \tag{9}$$

Larger time steps are possible if we use an implicit time stepping scheme as the backward Euler scheme, for example. Here, we apply a semi-implicit discretization of the convection term to avoid nonlinearities in the algebraic equations. Thus, we end up with the coupled problem:

Find $\vec{u}^{\,(n+1)}, p^{(n+1)}$ such that

$$\begin{aligned} \vec{u}^{\,(n+1)} + \delta t \left(-\frac{1}{Re} \triangle \vec{u}^{\,(n+1)} + \vec{u}^{\,(n)} \cdot \nabla \vec{u}^{\,(n+1)} + \nabla p^{(n+1)} \right) &= \vec{u}^{\,(n)} + \delta t \vec{g}^{\,(n+1)}, && \text{(10a)} \\ \nabla \cdot \vec{u}^{\,(n+1)} &= 0. && \text{(10b)} \end{aligned}$$

Again, we compute a tentative velocity field, now as the solution of the convection-diffusion equation[3]

$$\vec{u}^{\,\star} + \delta t \left(-\frac{1}{Re} \triangle \vec{u}^{\,\star} + \vec{u}^{\,(n)} \cdot \nabla \vec{u}^{\,\star} + \nabla p^{(n)} \right) = \vec{u}^{\,(n)} + \delta t \vec{g}^{\,(n+1)}. \tag{11}$$

These velocities do not satisfy the continuity equation. Thus, we have to compute some correction terms $\vec{u}^{\,\prime}$ and p' such that

$$\vec{u}^{\,(n+1)} = \vec{u}^{\,\star} + \vec{u}^{\,\prime}, \qquad p^{(n+1)} = p^{(n)} + p'. \tag{12}$$

Subtracting (11) from (10a) gives us an equation for $\vec{u}^{\,\prime}$

$$\vec{u}^{\,\prime} + \delta t \left(-\frac{1}{Re} \triangle \vec{u}^{\,\prime} + \vec{u}^{\,(n)} \cdot \nabla \vec{u}^{\,\prime} + \nabla p' \right) = 0 \tag{13}$$

or, using the abbreviation

$$S(\vec{u}^{\,\prime}) := -\frac{1}{Re} \triangle \vec{u}^{\,\prime} + \vec{u}^{\,(n)} \cdot \nabla \vec{u}^{\,\prime}, \tag{14}$$

we obtain

$$\vec{u}^{\,\prime} + \delta t S(\vec{u}^{\,\prime}) + \delta t \nabla p' = 0. \tag{15}$$

Depending on the specific choice how to approximate $S(\vec{u}^{\,\prime})$, there exist different numerical methods in literature.

One is to approximate $S(\vec{u}^{\,\prime})$ by zero. This implies that the values of $\vec{u}^{\,\prime}$ change very little in space. Then we plug (15) into the continuity equation, and we obtain a Poisson equation for the pressure correction

$$\triangle p' = \frac{1}{\delta t} \nabla \cdot \vec{u}^{\,\star}. \tag{16}$$

This approach is used for example by [Nonino & del Giudice, 1985] and [Kim & Chung, 1988] who call this scheme SIMPLE as well as by [Cheng & Armfield, 1995] who derive this scheme from the SMAC method (simplified marker and cell) [Amsden & Harlow, 1970]. In the following, we call this scheme SMAC.

[3] Note that this is a system of decoupled convection-diffusion equations for each component of the tentative velocity field $\vec{u}^{\,\star}$.

In the original SIMPLE scheme introduced by [Patankar & Spalding, 1972], the space discretization of the linear operator $S(\vec{u}\,')$ is written as a product of a matrix A which depends on the velocities of the previous time step with the velocity correction $\vec{u}\,'_h$

$$S_h(\vec{u}\,'_h) = A(\vec{u}_h^{(n)})\vec{u}\,'_h \tag{17}$$

and (15) becomes

$$(I + \delta t A(\vec{u}_h^{(n)}))\vec{u}\,'_h + \delta t \nabla_h p'_h = 0. \tag{18}$$

Furthermore, $A(\vec{u}_h^{(n)})$ is substituted by the diagonal matrix $D(\vec{u}_h^{(n)}) := \mathrm{diag}(A(\vec{u}_h^{(n)}))$. Thus, $\vec{u}\,'_h$ is assumed to be small and the off-diagonal elements are neglected. (18) gives

$$\vec{u}\,'_h = -\delta t(I + \delta t D(\vec{u}_h^{(n)}))^{-1}\nabla_h p'_h. \tag{19}$$

If we substitute $\vec{u}\,'_h$ by the right hand side of (19) in the continuity equation, we end up with an equation for p'_h which depends on $\vec{u}_h^{(n)}$

$$\nabla_h \cdot (I + \delta t D(\vec{u}_h^{(n)}))^{-1}\nabla_h p'_h = \frac{1}{\delta t}\nabla \cdot \vec{u}\,^\star_h. \tag{20}$$

Here, relaxation parameters are often used for stationary problems to get better convergence results.
The SIMPLEC algorithm [Patankar, 1981] assumes that $\vec{u}\,'_h$ is nearly constant in a certain surrounding and uses lumping of the matrix $I + \delta t A(\vec{u}_h^{(n)})$ in (18). This means that we substitute the matrix $I + \delta t A(\vec{u}_h^{(n)})$ by a diagonal matrix where the diagonal elements are the sums of all elements of one row.[4]
Moreover, there exist schemes as SIMPLER [Patankar, 1980] which use an inner iteration of at most two cycles for the solution of the coupled problem (10a), (10b). One cycle consists of the computation of the tentative velocity field by (11) using the pressure computed in the previous cycle and the computation of a pressure correction by the continuity equation.
For all these different numerical schemes, we basically have to solve Poisson or Poisson-like equations for the pressure or the pressure correction and convection-diffusion equations for each component of the velocity vector. In the following, we will consider the solution of these equations by algebraic multigrid.

The Transport Equation

Besides the Navier-Stokes equations, we also consider the scalar transport equation

$$c_t - \lambda \triangle c + \vec{u} \cdot \nabla c = 0 \tag{21}$$

with the diffusion coefficient λ and the velocity field $\vec{u}$.
This describes for example the transport of a chemical substance with concentration c. Setting $c := T$, (21) is the energy equation for the temperature T. Here, for reasons of simplicity, we omit the recoupling of the concentration or temperature, respectively, on the momentum equations which can be modeled for example using the Boussinesq approximation (see [Oberbeck, 1879], [Boussinesq, 1903], [Bejan, 1984]).

[4] Note that this approach is equivalent to the SMAC approach if we use conventional upwind-discretizations instead of the Donor-Cell scheme.

Time discretization with the backward Euler scheme gives

$$c^{(n+1)} + \delta t(-\lambda \triangle c^{(n+1)} + \vec{u} \cdot \nabla c^{(n+1)}) = c^{(n)}, \tag{22}$$

where $\vec{u}$ is either an already computed stationary velocity field or $\vec{u}^{\,(n+1)}$. (22) is equivalent to the momentum equation (11) for the tentative velocity field $\vec{u}^{\,\star}$. The only differences are the right hand side and the diffusion coefficient. For the discretization in space, we use again the staggered grid and finite differences with a mixed central/upwind discretization of the convective term. For details, see [Griebel et al., 1995, pp 134ff].

3 The Algebraic Multigrid Method and its Application to the Navier-Stokes equations

Algebraic Multigrid

Algebraic multigrid methods for the solution of a linear system

$$\mathcal{A}_M u_M = f_M$$

on a fine grid level M were introduced in [Brandt et al., 1982], [Brandt, 1983], [Ruge & Stüben, 1984], and [Ruge & Stüben, 1986]. Here, first a grid is set up on the next coarser level by using algebraic information from $\mathcal{A}_L$ ($L \leq M$) and then an appropriate interpolation scheme $\mathcal{P}_{L-1}^{L}$ is defined. After computing

$$\mathcal{A}_{L-1} := (\mathcal{P}_{L-1}^{L})^T \mathcal{A}_L \mathcal{P}_{L-1}^{L} \tag{23}$$

via the Galerkin identity, the process is repeated until a sufficiently coarse level system is obtained. AMG is necessarily less efficient than highly specialized geometric multigrid solvers for elliptic problems on uniform rectangular grids. However, for more complicated cases with general domains, AMG has been shown to behave robust and thus performs quite favorably in terms of operation count and CPU time. AMG also works for problems where geometric multigrid methods are impossible to design. AMG uses no sophisticated smoother, but only standard Gauß-Seidel. The robustness of AMG is obviously the merit of the appropriately chosen grid coarsening strategy and the associated interpolations.
For algebraic multigrid, the grids should be nested as for conventional multigrid methods, but they need not to be uniform. In fact, uniformity, if given for the finest grid, is in general not maintained in the process. We will nevertheless start with fine level discretizations based on the regular grid Ω_h. In the following we will denote the set of indices of the grid points corresponding to level L by N_L and we demand that the index sets are nested

$$N_1 \subset N_2 \subset \ldots \subset N_{M-1} \subset N_M.$$

To each grid point of level L, there corresponds an unknown of the solution vector u_L with the same index.
For an AMG algorithm, the sequence of matrices $\mathcal{A}_L$ must be constructed algebraically. The $\mathcal{A}_{L-1}, L = M, \ldots, 2$ are computed successively by selecting a subset of the unknowns of the level L system by evaluating the *strength of the connections* between the unknowns in $\mathcal{A}_L$. The basis for our implementation is the AMG method described in [Ruge & Stüben, 1984, Ruge & Stüben, 1986].

According to the well-known variational principle, it is the best for a given interpolation to determine the coarse-grid discretization via Galerkin-coarsening. All error components lying in the range of the interpolation are then eliminated by a single coarse grid correction. In multigrid theory one has to take care that those error components, which are persistent to the smoother, are well represented on coarser grids.

The effect of Gauß-Seidel iterations on symmetric positive definite matrices $\mathcal{A}_M$ is well understood and can be used to guide the construction of the coarser level systems $\mathcal{A}_L$ for $L = M-1, \ldots, 1$. Gauß-Seidel smoothing is stalling whenever the error $e_L^{it} := u_L^{it} - u_L$ in iteration it is big in comparison to the residual $r_L^{it} := \mathcal{A}_L u_L^{it} - f_L$.

Because of $\mathcal{A}_L e_L = r_L$, we have $\mathcal{A}_L e_L \approx 0$ then. Or for a single unknown

$$(e_L)_i = -\frac{1}{(\mathcal{A}_L)_{ii}} \sum_{\substack{j=1 \\ j \neq i}}^{n_L} (\mathcal{A}_L)_{ij} (e_L)_j \; .$$

This sum may be splitted into the error components visible on the coarse grid (and thus eliminated by a single coarse grid correction step) and those which are not, i.e.

$$(e_L)_i = -\frac{1}{(\mathcal{A}_L)_{ii}} \left(\sum_{\substack{j \in C_L \\ j \neq i}} (\mathcal{A}_L)_{ij} (e_L)_j + \sum_{\substack{j \in F_L \\ j \neq i}} (\mathcal{A}_L)_{ij} (e_L)_j \right) \; . \tag{24}$$

Here $C_L := N_{L-1}$ and $F_L := N_L \setminus N_{L-1}$. If the second sum could be eliminated on all levels, AMG would be a direct solver. In this case, the ideal interpolation weights would be given by

$$(\mathcal{P}_{L-1}^L e_{L-1})_i = \begin{cases} (e_{L-1})_i \; , & i \in C_L \\ -\frac{1}{(\mathcal{A}_L)_{ii}} \sum_{\substack{j \in C_L \\ j \neq i}} (\mathcal{A}_L)_{ij} (e_{L-1})_j \; , & i \in F_L . \end{cases} \tag{25}$$

Unfortunately, this ideal assumption can hardly be fulfilled when we want a decrease of the number of grid points on each level. Nevertheless, we try to minimize the second sum in (24) by choosing the coarse grid points $C_L := N_{L-1}$ from N_L appropriately.

We will briefly review the coarse grid selection part of AMG, as introduced in [Ruge & Stüben, 1984, Ruge & Stüben, 1986]. For reasons of simplicity, the level index L is omitted. Here, we have to define the set of strongly coupled neighbours S_i of a point i. Let

$$d(i, I) := \frac{1}{\max\limits_{k \neq i}\{-\mathcal{A}_{ik}\}} \sum_{j \in I} -\mathcal{A}_{ij} \; ,$$

where I is any subset of N, and

$$S^i := \{j \in N | d(i, \{j\}) \geq \alpha\}, \qquad S^{i,T} := \{j \in N | i \in S^j\} \; . \tag{26}$$

The partitioning in fine and coarse grid points is performed in two phases on each level. There, we select coarse grid points in such a manner, that as many strong couplings as possible are taken into consideration.

Selection of coarse grid points:
Setup Phase I

1. Set $C = \emptyset$ and set $F = \emptyset$

2. **While** $C \cup F \neq N$ **do**

    Pick $i \in N \setminus (C \cup F)$ with maximal $|S^{i,T}| + |S^{i,T} \cap F|$
    If $|S^{i,T}| + |S^{i,T} \cap F| = 0$
        then set $F = N \setminus C$
        else set $C = C \cup \{i\}$ and set $F = F \cup (S^{i,T} \setminus C)$;
    endif

The measure $|S^{i,T}| + |S^{i,T} \cap F|$ is purely heuristical. The first term is associated to the total number of strongly coupled neighbours, the second one to the number of strongly coupled neighbours which are in F. Domains with the same discretization stencil for most nodes (typically inner nodes), tend to have the same value of the measure $|S^{i,T}| + |S^{i,T} \cap F|$ for them. Note that the action to pick an index in step 2 of the above algorithm is non-deterministic and allows different implementations, depending on the chosen underlying data structures, see also [Bungartz, 1988]. Furthermore, using dynamic data structures and incremental techniques, it is possible to implement the overall setup algorithm (i.e. phase I and II) to need a number of operations proportional to the number of fine grid unknowns. Further improvements should be possible, if one would handle nodes situated next to the boundary of the domain and inner nodes differently.
In a second phase the final C-point choice is made.

Selection of coarse grid points:
Setup Phase II

1. Set $T = \emptyset$

2. **While** $T \subset F$ **do**

    Pick $i \in F \setminus T$ and set $T = T \cup \{i\}$
    set $\tilde{C} = \emptyset$ and set $C^i = S_i \cap C$
    set $F^i = S_i \cap F$
    While $F^i \neq \emptyset$ **do**
        Pick $j \in F^i$ and set $F^i = F^i \setminus \{j\}$
        If $d(j, C^i)/d(i, \{j\}) \leq \beta$
            then if $|\tilde{C}| = 0$
                then set $\tilde{C} = \{j\}$ and set $C^i = C^i \cup \{j\}$
                else set $C = C \cup \{i\}$, set $F = F \setminus \{i\}$ and **Goto 2**
            endif
        endif
    set $C = C \cup \tilde{C}$, set $F = F \setminus \tilde{C}$

This second algorithm has to make sure, that each point in F is strongly coupled directly with points in C or at least with points in F, which are strongly coupled with points in C. Again, the strategy to force the set $\tilde{C}$ to contain at most one element is purely heuristic. The parameters α and β which control the coarsening algorithm must be given by the user.

After the points N_L where divided into the sets F_L and C_L, we could define the interpolation as given in (25). In the algorithm of Ruge and Stüben, a little more sophisticated interpolation is used, which gives better results in numerical experiments:

$$(\mathcal{P}_{L-1}^{L} e_{L-1})_i := \begin{cases} (e_{L-1})_i \ , & i \in C_L \\ -\dfrac{\sum_{j \in C_L^i} \left((\mathcal{A}_L)_{ij} + c_{ij} \right) (e_{L-1})_j}{(\mathcal{A}_L)_{ii} + c_{ii}} \ , & i \in F_L, \end{cases} \tag{27}$$

where

$$c_{ij} := \sum_{\substack{k \notin C_L^i \\ k \neq i}} \frac{(\mathcal{A}_L)_{ik} (\mathcal{A}_L)_{kj}}{(\mathcal{A}_L)_{ki} + \sum_{l \in C_L^i} (\mathcal{A}_L)_{kl}} \ .$$

Once the interpolation matrix $\mathcal{P}_{L-1}^{L}$ is constructed, the system matrix $\mathcal{A}_{L-1}$ is determined by the Galerkin identity (23). Then, the coarsening proceeds recursively until the number of remaining unknowns equals one.

Application to the Navier-Stokes Equations

In our algorithm, we apply AMG for the solution of the potential equation for the initial velocity field (3) and of the Poisson equation for the pressure (8) in the explicit code or the pressure correction p' (16) in the SMAC code, respectively. For these equations, one single setup step is sufficient which consists of the setup phases I and II in the initializing phase of the algorithm, because the equations only differ in the boundary values and the right hand side, whereas the setup depends only on the matrix of the linear system. For the SIMPLE scheme, we would need a setup step for the equation for the pressure correction (20) in each time step because this equation depends on the velocities of the previous time step. The same holds for the SIMPLEC or SIMPLER scheme.
Moreover, we apply AMG to the momentum equation for $\vec{u}^{\star}$ (11) and to the transport equation (22). Here, we have to solve convection-diffusion problems where the convection dominates for high Reynolds numbers or low diffusion coefficients. The momentum equations change from time step to time step, because the time dependent velocities $\vec{u}^{(n)}$ enter the scheme in the convective term. Thus, we would need a setup phase in each time step.
In our numerical experiments, we also tried a sort of *adaptive setup strategy* for the momentum equations. This means that we only apply the quite expensive setup step if the number of AMG V-cycles exceeds a certain given number tol_{it} to reduce the residual below ε. Otherwise, we just keep the coarse grid and the interpolation operator of the previous time step. Thus, the problems on the coarser grids are not exactly the problems related to the fine grid equation but if the velocities change not too much from one time step to the next, the coarse grid problem based on previous velocities might still produce sufficient coarse grid correction terms for the fine grid equations with the new velocities.

4 Numerical Results

In our numerical experiments, we demonstrate the robustness of the algebraic multigrid method applied to the Navier-Stokes equations, with respect to the geometry, to the number of unknowns and to the diffusion coefficient in the convection-diffusion equation.
In all our experiments we used a multigrid V-cycle with one pre- and post-smoothing step. As smoother we use Gauss-Seidel relaxation. We reduced the L^2-norm of the residuals to values below 10^{-12} and we always measured the reduction rate, that is the quotient of the L^2-norms of the residuals of two successive iterates

$$\rho_{it} := \frac{\|r_{it}\|_2}{\|r_{it-1}\|_2}$$

in the last iteration when the stopping criterion was reached. The AMG parameter β is always set to 0.35 whereas α is varied as given in the tables.

4.1 Dependence of the Convergence on the Geometry

To test the behaviour of our algorithm for complicated geometries, we consider a channel flow, where several cubic obstacles are inserted in the channel, namely 1, 2 x 2, 4 x 4, 8 x 8 and 16 x 16 cubes (see Figure 3). We used a 256 x 64 grid. The size of the cubes is chosen such that the sum of their volumes is constant for each test case.

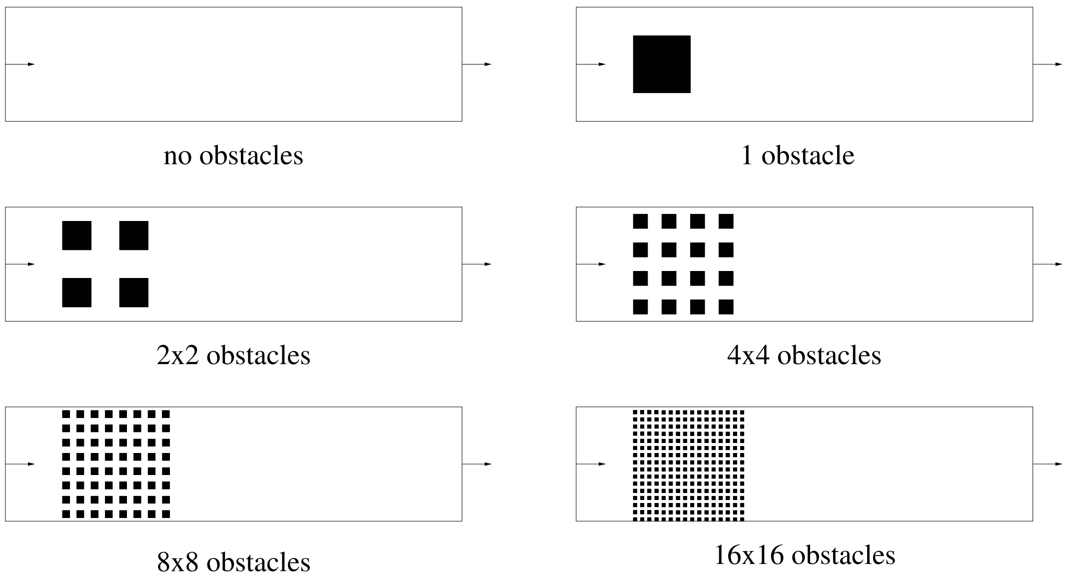

Fig. 3 Test problem for the dependence on the geometry, flow from the left to the right

First, we study the potential equation (3) for the initial velocity field. Note that the equations for the pressure (8) in the explicit time-stepping scheme and the pressure correction (16) in the SMAC scheme, respectively, are also Poisson equations with Neumann boundary conditions.

They only differ in the right hand side and in the kind of the boundary conditions (inhomogeneous/homogeneous). Thus, the convergence properties of AMG applied to those two equations are basically the same as for the potential problem (3).
In Table 1, we see that the reduction rates strongly depend on the parameter α, which determines the set of strongly coupled neighbours of a point (see (26)). The best values (underlined) are obtained for $\alpha < 0.1$. There, the reduction rates were between 0.08 and 0.18. The minimal values for each geometry are always below 0.12, but they depend in a way on α which is not yet fully understood. Thus, we can say that our algorithm is robust with respect to the geometry, but α must be chosen carefully. The number of coarse grid points is nearly the same for all cases, independent of the number of obstacles and the value of α. Thus, the time which must be spent for one V-cycle is always in the same range.

obstacles \ α	1e-6	1e-4	0.01	0.02	0.03	0.05	0.1	0.15	0.25	0.45	0.75	0.95
none	0.108	0.112	0.094	0.100	0.078	0.104	0.126	0.149	0.206	0.340	0.168	0.341
1	0.154	0.139	0.136	0.129	0.157	0.108	0.164	0.166	0.289	0.216	0.295	0.662
2x2	0.114	0.133	0.134	0.135	0.116	0.121	0.143	0.178	0.212	0.351	0.444	0.764
4x4	0.142	0.126	0.110	0.129	0.111	0.124	0.176	0.169	0.193	0.352	0.509	0.789
8x8	0.148	0.118	0.116	0.152	0.161	0.161	0.150	0.202	0.253	0.463	0.468	0.774
16x16	0.117	0.138	0.174	0.112	0.122	0.147	0.173	0.136	0.251	0.372	0.488	0.838

Table 1 Dependence of the reduction rates on the number of obstacles and on α, **2D**-potential equation with Neumann conditions

We believe that the relatively bad reduction rates for larger values of α are caused by the Neumann boundary conditions or the semidefiniteness of the linear system, respectively (see also the remark on page 9). With Dirichlet conditions we obtained much better reduction rates, also for large values of α as it can be seen in Table 2.

obstacles \ α	1e-4	0.05	0.25	0.45	0.85
0	0.081	0.079	0.099	0.101	0.035
1x1	0.088	0.066	0.095	0.061	0.035
2x2	0.095	0.067	0.101	0.089	0.033
4x4	0.090	0.080	0.100	0.058	0.033
8x8	0.080	0.079	0.100	0.081	0.149
16x16	0.082	0.079	0.098	0.128	0.091

Table 2 Dependence of the reduction rates on the number of obstacles and on α, **2D**-potential equation with Dirichlet conditions

The same results are obtained in the 3D case, where we considered a channel with 64 x 32 x 32 grid cells and between zero and 5 obstacles in each direction (see Table 3).

obstacles \ α	1e-6	1e-4	0.01	0.02	0.03	0.05	0.10	0.15	0.25	0.45	0.75	0.95
none	0.086	0.101	0.102	0.088	0.080	0.170	0.119	0.164	0.275	0.390	0.438	0.486
1	0.140	0.160	0.116	0.134	0.131	0.116	0.136	0.178	0.309	0.480	0.551	0.687
2x2x2	0.190	0.188	0.133	0.134	0.127	0.154	0.121	0.167	0.346	0.447	0.559	0.694
3x3x3	0.160	0.109	0.117	0.099	0.124	0.170	0.139	0.179	0.300	0.491	0.616	0.690
4x4x4	0.094	0.112	0.147	0.110	0.121	0.217	0.152	0.189	0.334	0.541	0.583	0.706
5x5x5	0.103	0.139	0.111	0.112	0.115	0.133	0.149	0.201	0.289	0.396	0.533	0.762

Table 3 Dependence of the reduction rates on the number of obstacles and on α, **3D**-potential equation with Neumann conditions

4.2 Dependence of the Convergence on the Grid Size

Next, we study the dependence of the reduction rates on the grid size. As test problem, we took the example of Subsection 4.1 with one obstacle and varied the number of cells. We show the reduction rates for different values of α in Table 4 (**2D** and **3D**).
We have a strong dependence on α with the best results obtained for $\alpha < 0.1$. Here, the convergence rates are slightly increasing for larger grids.

grid \ α	1e-6	1e-4	0.01	0.02	0.03	0.05	0.1	0.15	0.25	0.45	0.75	0.95
64x16	0.095	0.095	0.064	0.104	0.137	0.128	0.095	0.129	0.147	0.251	0.108	0.557
128x32	0.084	0.090	0.133	0.147	0.115	0.128	0.161	0.153	0.157	0.362	0.099	0.658
256x64	0.154	0.139	0.136	0.129	0.157	0.108	0.164	0.166	0.289	0.216	0.295	0.662
512x128	0.142	0.115	0.144	0.144	0.312	0.137	0.151	0.257	0.259	0.443	0.263	0.727
32x16x16	0.086	0.104	0.095	0.109	0.137	0.148	0.124	0.123	0.156	0.229	0.434	0.723
64x32x32	0.140	0.160	0.116	0.134	0.131	0.116	0.136	0.178	0.309	0.480	0.551	0.687
96x32x32	0.137	0.136	0.159	0.160	0.149	0.145	0.180	0.176	0.182	0.533	0.583	0.638

Table 4 Dependence of the reduction rates on the grid size and on α, potential equation with Neumann conditions, **2D** and **3D**

But if we consider the potential equation with Dirichlet conditions (Table 5), we obtain convergence rates which are in the same range, independent of the grid size. Thus, the relatively bad convergence behaviour must be caused by the Neumann conditions.

grid \ α	1e-4	0.05	0.25	0.45	0.85
64x16	0.062	0.050	0.052	0.039	0.017
128x32	0.070	0.057	0.075	0.048	0.023
256x64	0.088	0.066	0.095	0.061	0.035
512x128	0.024	0.068	0.069	0.086	0.076
32x16x16	0.0384	0.0420	0.0383	0.0437	0.0623
64x32x32	0.0518	0.0510	0.1050	0.0870	0.1622
96x32x32	0.0571	0.0510	0.0822	0.2992	0.2991

Table 5 Dependence of the reduction rates on the grid size and on α, potential equation with Dirichlet conditions, **2D** and **3D**

4.3 Dependence of the Convergence on the Diffusion Coefficient

Now, we consider the dependence of the convergence properties of the AMG algorithm for convection-diffusion equations like the momentum equations (11) as they appear in the SMAC and SIMPLE scheme and the transport equation (21). This is also a problem where standard multigrid methods fail. As test problem, we take the flow over a backward facing step with Reynolds number $Re = 500$ (see Figure 4). Here, two recirculating regions appear.
We consider the convergence behaviour of our AMG algorithm for the transport equation (21) with Dirichlet boundary conditions on the left side and Neumann (adiabatic) boundary conditions on the remaining three sides. In 2D, we employed a mesh with 300x75 cells and we used a 80x16x16 grid in 3D. As already mentioned, the time discrete transport equation (22) is of the same type as the momentum equations for each component of the tentative velocity (11). The results are shown in the Tables 6 and 7. For 2D, we also present the complexity of the coarse

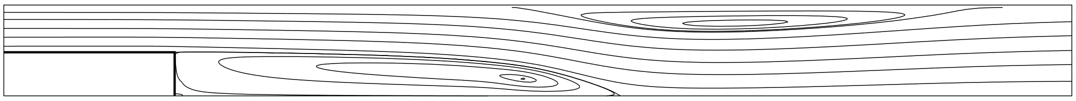

Fig. 4 Flow over a backward facing step, streamlines, $Re = 500$

grids (comp), i.e. the number of the unknowns on all levels divided by the number of unknowns on the finest level and the connectivity of the coarse grid operators (conn), i.e. the number of non-zero entries in the matrices on all levels divided by those on the finest level. Thus, both numbers indicate the work time which is necessary for one multigrid iteration.

α		1e-4	0.01	0.05	0.15	0.25	0.45	0.85
$\lambda = 1$	red	0.081	0.062	0.043	0.091	0.079	0.089	0.152
	comp	1.874	2.035	1.831	1.827	1.887	1.900	1.944
	conn	4.368	5.349	3.360	2.752	2.968	3.024	3.083
$\lambda = 10^{-2}$	red	0.025	0.040	0.032	0.045	0.066	0.105	0.406
	comp	1.884	1.924	1.995	1.967	1.950	1.969	1.960
	conn	4.483	4.515	4.688	3.735	3.265	3.358	2.841
$\lambda = 10^{-4}$	red	0.004	0.007	0.050	0.092	0.107	0.114	0.164
	comp	2.264	2.312	2.179	2.078	2.032	2.031	1.989
	conn	10.01	6.952	5.051	3.571	3.127	2.688	2.311
$\lambda = 10^{-6}$	red	0.003	0.004	0.012	0.058	0.085	0.101	0.192
	comp	2.327	2.270	2.082	2.057	2.039	2.021	1.978
	conn	9.337	5.585	4.000	3.120	2.899	2.566	2.266
$\lambda = 10^{-8}$	red	0.002	0.004	0.012	0.066	0.084	0.091	0.184
	comp	2.317	2.250	2.076	2.047	2.029	2.009	1.967
	conn	9.417	5.591	3.863	3.183	2.882	2.539	2.249

Table 6 Dependence of the reduction rates, complexity (comp) and connectivity (conn) on the diffusion parameter λ and on α, transport equation, backward facing step, **2D**

α	1e-4	0.01	0.05	0.25	0.45	0.85
$\lambda = 1$	0.106	0.097	0.125	0.135	0.470	0.724
$\lambda = 10^{-2}$	0.027	0.027	0.030	0.221	0.235	0.364
$\lambda = 10^{-4}$	0.002	0.008	0.034	0.108	0.124	0.142
$\lambda = 10^{-6}$	5e-4	0.005	0.036	0.105	0.115	0.135
$\lambda = 10^{-8}$	6e-4	0.005	0.037	0.103	0.111	0.135

Table 7 Dependence of the reduction rates on the diffusion parameter λ and on α, transport equation, backward facing step, **3D**

We see that also for the convection-diffusion equation, we get the best reduction rates ($\rho < 0.1$, for some α even below 0.01) for $\alpha \leq 0.15$, independent of λ. But for very small α and especially small λ, the connectivity is worse than for bigger values of α. Note that the recirculating regions do not affect the convergence numbers.
Moreover, we consider the convergence rates for the transport equation in dependence on the number of obstacles and the diffusion coefficient. Therefore, we choose again the domain with

the cubic obstacles as shown in Figure 3 and we set the AMG parameter $\alpha = 0.05$. As we can see in Table 8, the convergence rates do not vary very much for a fixed diffusion coefficient and different number of obstacles. Interestingly, the rates become better for smaller diffusion coefficients but even for $\lambda = 1$, they are still quite good.

obstacles \ λ	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
none	0.053	0.047	0.077	0.0008	0.0007	0.0007
1	0.077	0.046	0.065	0.025	0.027	0.027
2x2	0.086	0.030	0.063	0.051	0.053	0.057
4x4	0.097	0.038	0.054	0.036	0.027	0.023
8x8	0.133	0.049	0.071	0.033	0.027	0.027
16x16	0.147	0.048	0.060	0.057	0.031	0.031

Table 8 Dependence of the reduction rates on the number of obstacles and on λ, transport equation, **2D**, $\alpha = 0.05$, 256 x 64 cells

Equivalent results are obtained, if we consider the dependence of the reduction rates for the transport equation on the grid size and on the diffusion coefficient (see Table 9).

grid \ λ	1	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
64 x 16	0.047	0.013	0.027	0.013	0.013	0.014
128 x 32	0.069	0.038	0.022	0.025	0.025	0.025
256 x 64	0.077	0.046	0.065	0.025	0.027	0.027
512 x 128	0.077	0.067	0.079	0.079	0.033	0.033

Table 9 Dependence of the reduction rates on the grid size and on λ, transport equation, **2D**, $\alpha = 0.05$, 1 obstacle

4.4 The Full Navier Stokes Solver

Now, we compare the solution process of the Navier-Stokes equations by the explicit and the SMAC-semi-implicit scheme using AMG for the computation of the initial velocity field, for the pressure in the explicit scheme and the pressure correction as the tentative velocities in the semi-implicit scheme. Here, we also applied the above mentioned adaptive setup strategy.
As test problem, we consider the flow around an obstacle at $Re = 20$ and we use a mesh with 220 x 41 cells in 2D and with 60 x 12 x 12 cells in 3D. We run our program until $t = 15$ was reached. We stopped the iterations if the norm of the residual was below 10^{-6}. The explicit time stepping scheme becomes unstable for time step sizes $\delta t > 0.014$ in 2D and $\delta t > 0.051$ in 3D whereas the semi-implicit code still shows good results for $\delta t = 0.5$, either in 2D and 3D. However, we see in the Tables 10 and 11 that the time spent for the computation for one time step in the semi-implicit code is much larger than for the explicit code. This is due to the time which must be spent for the setup phase. Nevertheless, because of the large number of allocation and comparing steps, the time spent for this phase depends a lot on the hardware platform available.[5]
The time spent for the semi-implicit code can be reduced, if we use the adaptive setup strategy. Here, we performed a new setup step only if the number of iterations in the previous timestep

[5] We used a HP 9000/712 workstation.

was greater than or equal to the given value tol_{it}. Otherwise, the coarse grid correction computed using the coarse grid sequence of the iteration before is still good enough to reduce the error in the new timestep efficiently. As we see in the Tables 10 and 11, there are dramatic differences in the computing times for $\delta t = 0.05$ and $tol_{it} = 1$ (setup in every time step) and $tol_{it} = 4$, where only 1 setup step is needed for each momentum equation, whereas the number of iterations is not much bigger.

scheme	tol_{it}	δt	timesteps	u-Setup	v-Setup	u-Iter	v-Iter	comp. time
explicit	-:-	0.01	1500	-:-	-:-	-:-	-:-	14m5.25s
semi-implicit	4	0.01	1500	1	1	2226	1976	1h39m6.72s
semi-implicit	4	0.02	750	1	1	1230	1098	50m28.67s
semi-implicit	1	0.05	300	300	300	514	459	1h41m53.13s
semi-implicit	4	0.05	300	1	1	560	483	23m22.48s
semi-implicit	4	0.10	150	3	3	293	257	12m40.28s
semi-implicit	3	0.20	75	23	43	174	194	17m05.46s
semi-implicit	4	0.20	75	4	4	188	199	8m36.73s
semi-implicit	5	0.20	75	2	2	213	209	8m11.94s
semi-implicit	3	0.50	30	30	30	94	92	11m39.24s
semi-implicit	4	0.50	30	9	6	99	95	5m51.10s
semi-implicit	5	0.50	30	5	4	109	104	5m06.77s
semi-implicit	10	0.50	30	2	2	149	130	4m51.65s

Table 10 Number of setup steps, number of iterations and overall computation time for the Navier Stokes solver, $Re = 20$, **2D**

scheme	tol_{it}	δt	timesteps	u-Setup	v-Setup	w-Setup	u-Iter	v-Iter	w-Iter	comp. time
explicit	-:-	0.03	500	-:-	-:-	-:-	-:-	-:-	-:-	1h 0m57.40s
semi-implicit	4	0.03	500	1	1	1	900	836	724	2h17m6.40s
semi-implicit	4	0.06	250	1	1	1	497	479	423	1h16m22.69s
semi-implicit	1	0.10	150	150	150	150	308	286	242	10h16m43.39s
semi-implicit	4	0.10	150	3	2	1	316	305	279	0h55m16.78s
semi-implicit	3	0.20	75	35	30	15	182	169	142	2h 5m17.56s
semi-implicit	4	0.20	75	7	4	2	187	177	157	0h40m57.27s
semi-implicit	5	0.20	75	2	2	1	202	185	166	0h32m20.19s
semi-implicit	3	0.50	30	30	30	29	110	119	92	1h55m7.15s
semi-implicit	4	0.50	30	18	27	4	110	119	94	1h 4m36.13s
semi-implicit	5	0.50	30	4	3	2	127	124	114	0h23m1.20s
semi-implicit	10	0.50	30	2	2	2	140	137	119	0h19m45.11s

Table 11 Number of setup steps, number of iterations and overall computation time for the Navier Stokes solver, $Re = 20$, **3D**

So, the semi-implicit algorithm using adaptive setup is quite efficient compared with the explicit algorithm, because the size of the timestep is not so much restricted. We assume that the difference is more severe for finer grids, but here, we must also mention, that the memory needed for the semi-implicit algorithm is much larger, because the matrices for the momentum equations including the coarse grid operators must be stored.

4.5 Two other Problems with Complicated Domains

At last, we present some results of two other problems with complicated geometries, namely the flow through a river system, here the delta of the river Ganges in Bangladesh[6], and the flow through a porous medium on a micro-scale level.
In Figures 5 and 7, we show the geometric structure of the computational domains and the stationary velocity fields. For the Ganges-example, we use inflow conditions at the five branches at the top, for the porous medium example, we use inflow on the left and outflow on the right.
Figures 6 and 8 show the propagation of a chemical pollution modeled by the transport equation (21). In the Ganges-example, the permanent source of pollution is situated at the top of the second branch from the left, in the porous medium, the pollution is coming in in the middle of the left boundary.
The convergence properties are in the same range as for the test problems reported above.

5 Conclusions

In this paper, we considered the application of algebraic multigrid methods to the Poisson equation and the convection-diffusion equation in complicated geometries. Both equations arise in the numerical solution of the Navier-Stokes equations using explicit or semi-implicit time-discretizations.
For equations in complicated geometries and for convection-dominated problems, the convergence rates of standard multigrid methods usually deteriorate. In several numerical experiments we demonstrated that the application of AMG leads to robust and efficient algorithms, especially for a proper choice of the AMG-parameter α which controls the coarsening process. However, the dependence of the convergence rates on α is not yet fully understood. Moreover, a modification of the algorithm for Neumann boundary conditions might improve the convergence rates for the pressure equation.
Furthermore, our experiments show that an explicit time-stepping scheme, where we have a strong restriction on the time step size, can still compete with the semi-implicit algorithms, concerning the run-time of the algorithms. This is due to the large time spent for the AMG setup phases.
But we suppose, that the semi-implicit time discretization is superior for finer grids where the time step size in the explicit scheme must be reduced more and more to preserve stability. This holds especially for stationary problems, where we can apply an adaptive setup strategy in which the setup is not done in every time step but only if the convergence rate gets worse than a prescribed value.

Acknowledgement: We are grateful to Erwin Wagner, Ralph Kreissl and Markus Rykaschewski who implemented and tested the 3D version of the AMG-Navier-Stokes solver.

[6]Note that this two-dimensional calculation is not realistic at all. First, the resolution of the domain is far too coarse, second, the third dimension, i.e. the deepness of the river arms, is not involved, which influences the computed flow velocity, and third, realistic in- and outflow conditions are not known. This is only an example for complicated geometries.

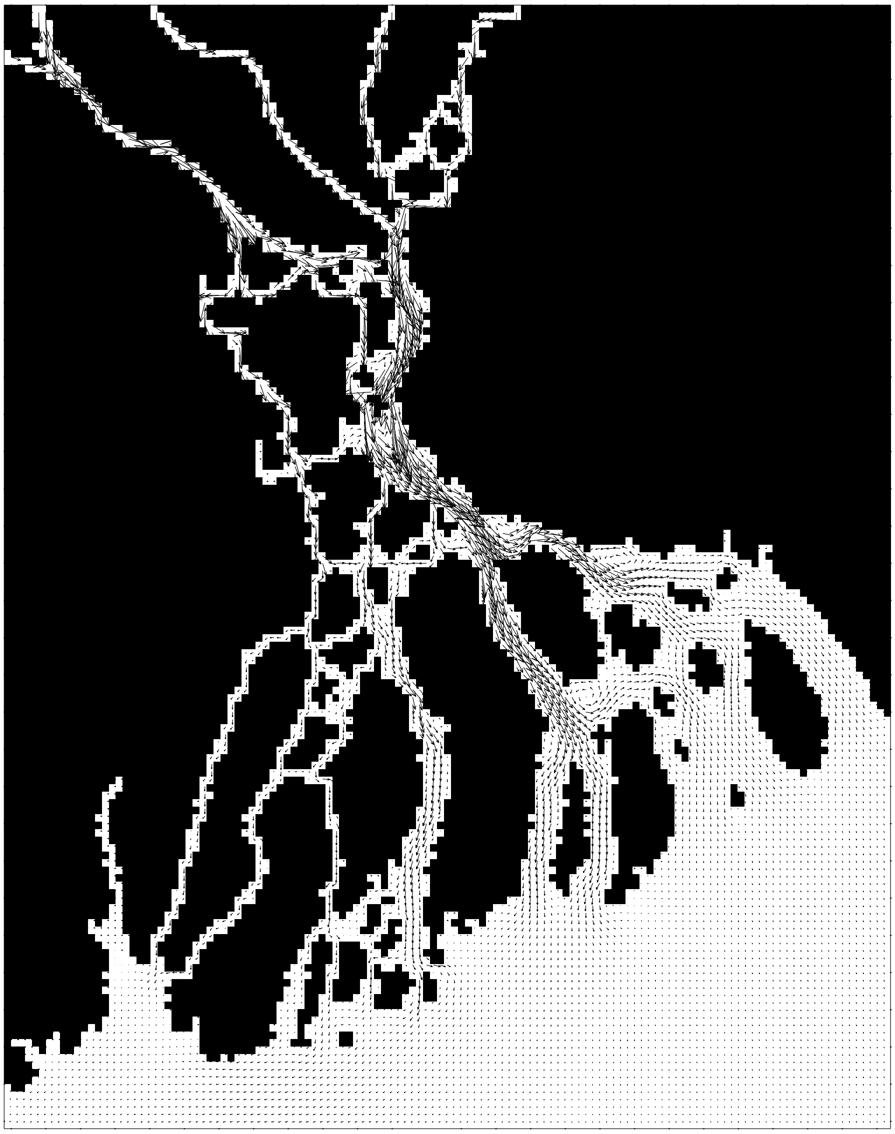

Fig. 5 Ganges Delta, velocity plot

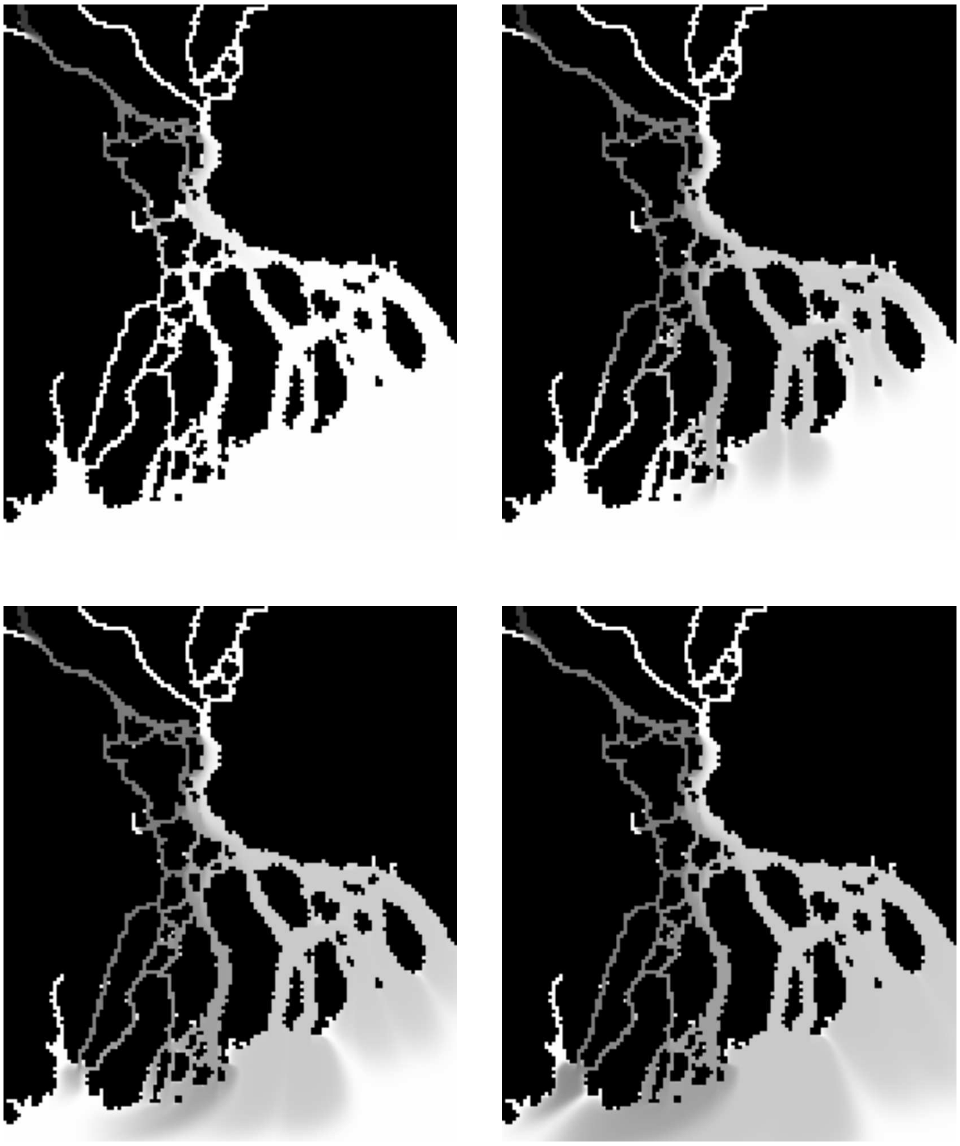

Fig. 6 Pollution transport in the Ganges Delta, $\lambda = 4.6 \cdot 10^{-11}$

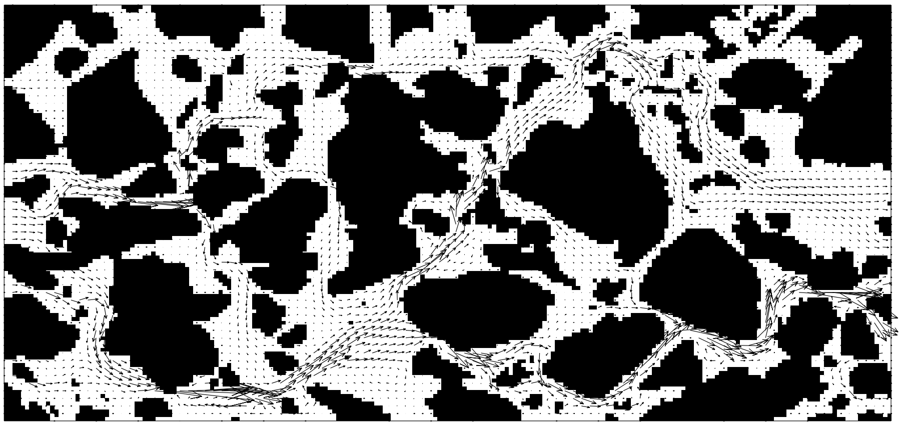

Fig. 7 Porous medium, velocity plot

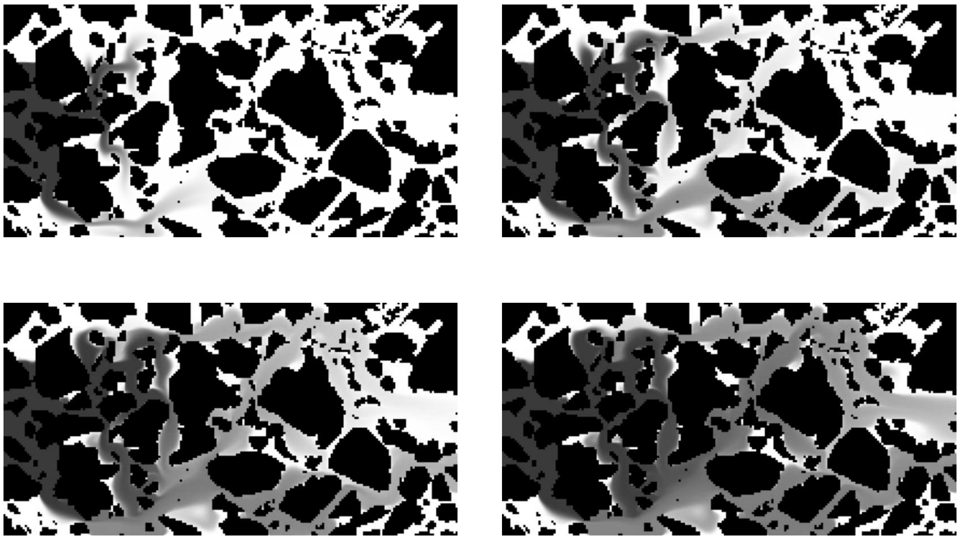

Fig. 8 Pollution transport in a porous medium, $\lambda = 10^{-4}$

References

[Amsden & Harlow, 1970] Amsden, A. & Harlow, F. (1970). *The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flow*. Technical report, Los Alamos Scientific Lab. Rep. LA-4370.

[Bank & Xu, 1995] Bank, R. & Xu, J. (1995). A hierarchical basis multi-grid method for unstructured grids. In W. Hackbusch & G. Wittum (Eds.), *Fast Solvers for Flow Problems, Proc. of the 10th GAMM Seminar*: Vieweg.

[Bank & Xu, 1996] Bank, R. & Xu, J. (1996). An algorithm for coarsening unstructured meshes. *Numer. Math.*, 73, 1–36.

[Bejan, 1984] Bejan, A. (1984). *Convection Heat Transfer*. New York: Wiley-Interscience.

[Boussinesq, 1903] Boussinesq, J. (1903). Theorie Analytique de la Chaleur. *Gauthier-Villars*, 2.

[Brandt, 1983] Brandt, A. (1983). Algebraic multigrid theory: The symmetric case. In S. McCormick & U. Trottenberg (Eds.), *Preliminary Proceedings of the International Multigrid Conference, Copper Mountain, Colorado, April 6–8, 1983*.

[Brandt et al., 1982] Brandt, A., McCormick, S. & Ruge, J. (1982). *Algebraic Multigrid (AMG) for Automatic Algorithm Design and Problem Solution*. report, Colorado State University, Ft. Collins.

[Bungartz, 1988] Bungartz, H. (1988). *Beschränkte Optimierung mit algebraischen Mehrgittermethoden*. Diplomarbeit, Institut für Informatik, Technische Universität München.

[Cheng & Armfield, 1995] Cheng, L. & Armfield, S. (1995). A simplified marker and cell method for unsteady flows on non-staggered grids. *Int. J. Num. Meth. Fluids*, 21, 15–34.

[de Zeeuw, 1990] de Zeeuw, P. (1990). Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comp. appl. Math.*, 33, 1–27.

[Fayers & Hewett, 1992] Fayers, F. & Hewett, T. (1992). A review of current trends in petroleum reservoir description and assessing the impacts on oil recovery. In T. Russell (Ed.), *Computational Methods in Water Resources IX, Vol. 2: Mathematical modeling in water resources* (pp. 3–33). Southampton: Computational Mechanics Publ.

[Grauschopf et al., 1996] Grauschopf, T., Griebel, M. & Regler, H. (1996). *Additive Multilevel-Preconditioners Based on Bilinear Interpolation, Matrix-Dependent Geometric Coarsening and Algebraic Multigrid Coarsening for Second Order Elliptic PDEs*. TU München, Institut für Informatik, SFB-Report 342/02/96 A.

[Griebel et al., 1995] Griebel, M., Dornseifer, T. & Neunhoeffer, T. (1995). *Numerische Simulation in der Strömungsmechanik – eine praxisorientierte Einführung*. Wiesbaden: Vieweg.

[Hackbusch & Sauter, 1995] Hackbusch, W. & Sauter, S. (1995). *Composite Finite Elements for the Approximation of PDEs on Domains with Complicated Micro-Structures*. Technical report, 9505, Institut fuer Informatik und Praktische Mathematik, Christian-Albrechts Universität, Kiel.

[Harlow & Welch, 1965] Harlow, F. & Welch, J. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8, 2182–2189.

[Hirt et al., 1975] Hirt, C., Nichols, B. & Romero, N. (1975). *SOLA – A Numerical Solution Algorithm for Transient Fluid Flows*. Technical report, Los Alamos Scientific Lab. Rep. LA-5852.

[Hughes et al., 1986] Hughes, T., Franca, L. & Balestra, M. (1986). A new finite element formulation for computational fluid dynamics: III. The general streamline operator for multidimensional advective-diffusi on systems. *Comp. Meth. Appl. Mech. Eng.*, 58, 305–328.

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

[Kettler, 1982] Kettler, R. (1982). Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. *Multigrid Methods, Lecture Notes in Mathematics*, 960, 502–534.

[Kim & Chung, 1988] Kim, Y. & Chung, T. (1988). Finite-element analysis of turbulent diffusion flames. *AIAA Journal*, 27, 330–339.

[Kornhuber & Yserentant, 1994] Kornhuber, R. & Yserentant, H. (1994). Multilevel-methods for elliptic problems on domains not resolved by the coarse grid. *Contemp. Mathematics*, 180, 49–60.

[Lonsdale, 1993] Lonsdale, R. (1993). An algebraic multigrid solver for the Navier-Stokes equations on unstructered meshes. *Int. J. Num. Meth. Heat Fluid Flow*, 3, 3–14.

[Nonino & del Giudice, 1985] Nonino, C. & del Giudice, S. (1985). An improved procedure for finite-element methods in laminar and turbulent flow. In C. Taylor (Ed.), *Numerical Methods in Laminar and Turbulent Flow, Part I* (pp. 597–608).: Pineridge.

[Oberbeck, 1879] Oberbeck, A. (1879). Über die Wärmeleitung der Flüssigkeiten bei Berücksichtigung der Strömungen infolge von Temperaturdifferenzen. *Ann. Phys. Chem.*, 7, 271–292.

[Patankar, 1980] Patankar, S. (1980). *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill.

[Patankar, 1981] Patankar, S. (1981). A calculation procedure for two-dimensional elliptic situations. *Num. Heat Transfer*, 4, 409–425.

[Patankar & Spalding, 1972] Patankar, S. & Spalding, D. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15, 1787–1806.

[Quaterpelle, 1993] Quaterpelle, L. (1993). *Numerical Solution of the Incompressible Navier-Stokes Equations*. Basel: Birkhäuser. ISNM 113.

[Raw, 1994] Raw, M. (1994). *A coupled algebraic multigrid method for the 3D Navier-Stokes equations*. Technical report, Advanced Scientific Computing Ltd., Waterloo Ontario, Canada.

[Reusken, 1994] Reusken, A. (1994). Multigrid with matrix-dependent transfer operators for convection-diffusion problems. In P. Hemker & P. Wesseling (Eds.), *Multigrid Methods IV* (pp. 269–280).: Birkhäuser.

[Ruge & Stüben, 1984] Ruge, J. & Stüben, K. (1984). Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG). *Arbeitspapiere der GMD*, 89.

[Ruge & Stüben, 1986] Ruge, J. & Stüben, K. (1986). Algebraic multigrid (AMG). *Arbeitspapiere der GMD*, 210.

[Turek, 1995] Turek, S. (1995). *A Comparative Study of some Time-Stepping Techniques for the Incompressible Navier-Stokes Equations*. Preprint 95-10, IWR Heidelberg.

[Webster, 1994] Webster, R. (1994). An algebraic multigrid solver for Navier-Stokes problems. *Int. J. Numer. Methods Fluids*, 18, 761–780.

[Wittum, 1989a] Wittum, G. (1989a). Linear iterations as smoothers in multigrid methods: Theory with aplications to incomplete decompositions. *Impact of Computing in Science and Engineering*, 1, 180–215.

[Wittum, 1989b] Wittum, G. (1989b). On the robustness of ILU-smoothing. *SIAM J. Sci. Stat. Comput.*, 10, 699–717.

SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

342/1/90 A	Robert Gold, Walter Vogler: Quality Criteria for Partial Order Semantics of Place/Transition-Nets, Januar 1990
342/2/90 A	Reinhard Fößmeier: Die Rolle der Lastverteilung bei der numerischen Parallelprogrammierung, Februar 1990
342/3/90 A	Klaus-Jörn Lange, Peter Rossmanith: Two Results on Unambiguous Circuits, Februar 1990
342/4/90 A	Michael Griebel: Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen Transformations- Mehrgitter-Methode
342/5/90 A	Reinhold Letz, Johann Schumann, Stephan Bayerl, Wolfgang Bibel: SETHEO: A High-Performance Theorem Prover
342/6/90 A	Johann Schumann, Reinhold Letz: PARTHEO: A High Performance Parallel Theorem Prover
342/7/90 A	Johann Schumann, Norbert Trapp, Martin van der Koelen: SETHEO/PARTHEO Users Manual
342/8/90 A	Christian Suttner, Wolfgang Ertel: Using Connectionist Networks for Guiding the Search of a Theorem Prover
342/9/90 A	Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Jaroslav Kremenek, Robert Lindhof, Thomas Ludwig, Peter Luksch, Thomas Treml: TOPSYS, Tools for Parallel Systems (Artikelsammlung)
342/10/90 A	Walter Vogler: Bisimulation and Action Refinement
342/11/90 A	Jörg Desel, Javier Esparza: Reachability in Reversible Free- Choice Systems
342/12/90 A	Rob van Glabbeek, Ursula Goltz: Equivalences and Refinement
342/13/90 A	Rob van Glabbeek: The Linear Time - Branching Time Spectrum
342/14/90 A	Johannes Bauer, Thomas Bemmerl, Thomas Treml: Leistungsanalyse von verteilten Beobachtungs- und Bewertungswerkzeugen
342/15/90 A	Peter Rossmanith: The Owner Concept for PRAMs
342/16/90 A	G. Böckle, S. Trosch: A Simulator for VLIW-Architectures
342/17/90 A	P. Slavkovsky, U. Rüde: Schnellere Berechnung klassischer Matrix-Multiplikationen
342/18/90 A	Christoph Zenger: SPARSE GRIDS
342/19/90 A	Michael Griebel, Michael Schneider, Christoph Zenger: A combination technique for the solution of sparse grid problems

Reihe A

342/20/90 A	Michael Griebel: A Parallelizable and Vectorizable Multi- Level-Algorithm on Sparse Grids
342/21/90 A	V. Diekert, E. Ochmanski, K. Reinhardt: On confluent semi- commutations-decidability and complexity results
342/22/90 A	Manfred Broy, Claus Dendorfer: Functional Modelling of Operating System Structures by Timed Higher Order Stream Processing Functions
342/23/90 A	Rob van Glabbeek, Ursula Goltz: A Deadlock-sensitive Congruence for Action Refinement
342/24/90 A	Manfred Broy: On the Design and Verification of a Simple Distributed Spanning Tree Algorithm
342/25/90 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Peter Luksch, Roland Wismüller: TOPSYS - Tools for Parallel Systems (User's Overview and User's Manuals)
342/26/90 A	Thomas Bemmerl, Arndt Bode, Thomas Ludwig, Stefan Tritscher: MMK - Multiprocessor Multitasking Kernel (User's Guide and User's Reference Manual)
342/27/90 A	Wolfgang Ertel: Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems
342/28/90 A	Rob van Glabbeek, Frits Vaandrager: Modular Specification of Process Algebras
342/29/90 A	Rob van Glabbeek, Peter Weijland: Branching Time and Abstraction in Bisimulation Semantics
342/30/90 A	Michael Griebel: Parallel Multigrid Methods on Sparse Grids
342/31/90 A	Rolf Niedermeier, Peter Rossmanith: Unambiguous Simulations of Auxiliary Pushdown Automata and Circuits
342/32/90 A	Inga Niepel, Peter Rossmanith: Uniform Circuits and Exclusive Read PRAMs
342/33/90 A	Dr. Hermann Hellwagner: A Survey of Virtually Shared Memory Schemes
342/1/91 A	Walter Vogler: Is Partial Order Semantics Necessary for Action Refinement?
342/2/91 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Rainer Weber: Characterizing the Behaviour of Reactive Systems by Trace Sets
342/3/91 A	Ulrich Furbach, Christian Suttner, Bertram Fronhöfer: Massively Parallel Inference Systems
342/4/91 A	Rudolf Bayer: Non-deterministic Computing, Transactions and Recursive Atomicity
342/5/91 A	Robert Gold: Dataflow semantics for Petri nets
342/6/91 A	A. Heise; C. Dimitrovici: Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften
342/7/91 A	Walter Vogler: Asynchronous Communication of Petri Nets and the Refinement of Transitions
342/8/91 A	Walter Vogler: Generalized OM-Bisimulation
342/9/91 A	Christoph Zenger, Klaus Hallatschek: Fouriertransformation auf dünnen Gittern mit hierarchischen Basen

Reihe A

342/10/91 A Erwin Loibl, Hans Obermaier, Markus Pawlowski: Towards Parallelism in a Relational Database System

342/11/91 A Michael Werner: Implementierung von Algorithmen zur Kompaktifizierung von Programmen für VLIW-Architekturen

342/12/91 A Reiner Müller: Implementierung von Algorithmen zur Optimierung von Schleifen mit Hilfe von Software-Pipelining Techniken

342/13/91 A Sally Baker, Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Udo Graf, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Rainer Knödlseder, Jaroslav Kremenek, Siegfried Langenbuch, Robert Lindhof, Thomas Ludwig, Peter Luksch, Roy Milner, Bernhard Ries, Thomas Treml: TOPSYS - Tools for Parallel Systems (Artikelsammlung); 2., erweiterte Auflage

342/14/91 A Michael Griebel: The combination technique for the sparse grid solution of PDE's on multiprocessor machines

342/15/91 A Thomas F. Gritzner, Manfred Broy: A Link Between Process Algebras and Abstract Relation Algebras?

342/16/91 A Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Thomas Treml, Roland Wismüller: The Design and Implementation of TOPSYS

342/17/91 A Ulrich Furbach: Answers for disjunctive logic programs

342/18/91 A Ulrich Furbach: Splitting as a source of parallelism in disjunctive logic programs

342/19/91 A Gerhard W. Zumbusch: Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme

342/20/91 A M. Jobmann, J. Schumann: Modelling and Performance Analysis of a Parallel Theorem Prover

342/21/91 A Hans-Joachim Bungartz: An Adaptive Poisson Solver Using Hierarchical Bases and Sparse Grids

342/22/91 A Wolfgang Ertel, Theodor Gemenis, Johann M. Ph. Schumann, Christian B. Suttner, Rainer Weber, Zongyan Qiu: Formalisms and Languages for Specifying Parallel Inference Systems

342/23/91 A Astrid Kiehn: Local and Global Causes

342/24/91 A Johann M.Ph. Schumann: Parallelization of Inference Systems by using an Abstract Machine

342/25/91 A Eike Jessen: Speedup Analysis by Hierarchical Load Decomposition

342/26/91 A Thomas F. Gritzner: A Simple Toy Example of a Distributed System: On the Design of a Connecting Switch

342/27/91 A Thomas Schnekenburger, Andreas Weininger, Michael Friedrich: Introduction to the Parallel and Distributed Programming Language ParMod-C

342/28/91 A Claus Dendorfer: Funktionale Modellierung eines Postsystems

342/29/91 A Michael Griebel: Multilevel algorithms considered as iterative methods on indefinite systems

Reihe A

342/30/91 A	W. Reisig: Parallel Composition of Liveness
342/31/91 A	Thomas Bemmerl, Christian Kasperbauer, Martin Mairandres, Bernhard Ries: Programming Tools for Distributed Multiprocessor Computing Environments
342/32/91 A	Frank Leßke: On constructive specifications of abstract data types using temporal logic
342/1/92 A	L. Kanal, C.B. Suttner (Editors): Informal Proceedings of the Workshop on Parallel Processing for AI
342/2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS
342/2-2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS - Revised Version (erschienen im Januar 1993)
342/3/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/4/92 A	Claus Dendorfer, Rainer Weber: Development and Implementation of a Communication Protocol - An Exercise in FOCUS
342/5/92 A	Michael Friedrich: Sprachmittel und Werkzeuge zur Unterstüt- zung paralleler und verteilter Programmierung
342/6/92 A	Thomas F. Gritzner: The Action Graph Model as a Link between Abstract Relation Algebras and Process-Algebraic Specifications
342/7/92 A	Sergei Gorlatch: Parallel Program Development for a Recursive Numerical Algorithm: a Case Study
342/8/92 A	Henning Spruth, Georg Sigl, Frank Johannes: Parallel Algorithms for Slicing Based Final Placement
342/9/92 A	Herbert Bauer, Christian Sporrer, Thomas Krodel: On Distributed Logic Simulation Using Time Warp
342/10/92 A	H. Bungartz, M. Griebel, U. Rüde: Extrapolation, Combination and Sparse Grid Techniques for Elliptic Boundary Value Problems
342/11/92 A	M. Griebel, W. Huber, U. Rüde, T. Störtkuhl: The Combination Technique for Parallel Sparse-Grid-Preconditioning and -Solution of PDEs on Multiprocessor Machines and Workstation Networks
342/12/92 A	Rolf Niedermeier, Peter Rossmanith: Optimal Parallel Algorithms for Computing Recursively Defined Functions
342/13/92 A	Rainer Weber: Eine Methodik für die formale Anforderungsspezifkation verteilter Systeme
342/14/92 A	Michael Griebel: Grid– and point–oriented multilevel algorithms
342/15/92 A	M. Griebel, C. Zenger, S. Zimmer: Improved multilevel algorithms for full and sparse grid problems
342/16/92 A	J. Desel, D. Gomm, E. Kindler, B. Paech, R. Walter: Bausteine eines kompositionalen Beweiskalküls für netzmodellierte Systeme

Reihe A

342/17/92 A	Frank Dederichs: Transformation verteilter Systeme: Von applikativen zu prozeduralen Darstellungen
342/18/92 A	Andreas Listl, Markus Pawlowski: Parallel Cache Management of a RDBMS
342/19/92 A	Erwin Loibl, Markus Pawlowski, Christian Roth: PART: A Parallel Relational Toolbox as Basis for the Optimization and Interpretation of Parallel Queries
342/20/92 A	Jörg Desel, Wolfgang Reisig: The Synthesis Problem of Petri Nets
342/21/92 A	Robert Balder, Christoph Zenger: The d-dimensional Helmholtz equation on sparse Grids
342/22/92 A	Ilko Michler: Neuronale Netzwerk-Paradigmen zum Erlernen von Heuristiken
342/23/92 A	Wolfgang Reisig: Elements of a Temporal Logic. Coping with Concurrency
342/24/92 A	T. Störtkuhl, Chr. Zenger, S. Zimmer: An asymptotic solution for the singularity at the angular point of the lid driven cavity
342/25/92 A	Ekkart Kindler: Invariants, Compositionality and Substitution
342/26/92 A	Thomas Bonk, Ulrich Rüde: Performance Analysis and Optimization of Numerically Intensive Programs
342/1/93 A	M. Griebel, V. Thurner: The Efficient Solution of Fluid Dynamics Problems by the Combination Technique
342/2/93 A	Ketil Stølen, Frank Dederichs, Rainer Weber: Assumption / Commitment Rules for Networks of Asynchronously Communicating Agents
342/3/93 A	Thomas Schnekenburger: A Definition of Efficiency of Parallel Programs in Multi-Tasking Environments
342/4/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: A Proof of Convergence for the Combination Technique for the Laplace Equation Using Tools of Symbolic Computation
342/5/93 A	Manfred Kunde, Rolf Niedermeier, Peter Rossmanith: Faster Sorting and Routing on Grids with Diagonals
342/6/93 A	Michael Griebel, Peter Oswald: Remarks on the Abstract Theory of Additive and Multiplicative Schwarz Algorithms
342/7/93 A	Christian Sporrer, Herbert Bauer: Corolla Partitioning for Distributed Logic Simulation of VLSI Circuits
342/8/93 A	Herbert Bauer, Christian Sporrer: Reducing Rollback Overhead in Time-Warp Based Distributed Simulation with Optimized Incremental State Saving
342/9/93 A	Peter Slavkovsky: The Visibility Problem for Single-Valued Surface (z = f(x,y)): The Analysis and the Parallelization of Algorithms
342/10/93 A	Ulrich Rüde: Multilevel, Extrapolation, and Sparse Grid Methods
342/11/93 A	Hans Regler, Ulrich Rüde: Layout Optimization with Algebraic Multigrid Methods
342/12/93 A	Dieter Barnard, Angelika Mader: Model Checking for the Modal Mu-Calculus using GaußElimination
342/13/93 A	Christoph Pflaum, Ulrich Rüde: Gauß' Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids

Reihe A

342/14/93 A	Christoph Pflaum: Convergence of the Combination Technique for the Finite Element Solution of Poisson's Equation
342/15/93 A	Michael Luby, Wolfgang Ertel: Optimal Parallelization of Las Vegas Algorithms
342/16/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: Pointwise Convergence of the Combination Technique for Laplace's Equation
342/17/93 A	Georg Stellner, Matthias Schumann, Stefan Lamberts, Thomas Ludwig, Arndt Bode, Martin Kiehl und Rainer Mehlhorn: Developing Multicomputer Applications on Networks of Workstations Using NXLib
342/18/93 A	Max Fuchs, Ketil Stølen: Development of a Distributed Min/Max Component
342/19/93 A	Johann K. Obermaier: Recovery and Transaction Management in Write-optimized Database Systems
342/20/93 A	Sergej Gorlatch: Deriving Efficient Parallel Programs by Systemating Coarsing Specification Parallelism
342/01/94 A	Reiner Hüttl, Michael Schneider: Parallel Adaptive Numerical Simulation
342/02/94 A	Henning Spruth, Frank Johannes: Parallel Routing of VLSI Circuits Based on Net Independency
342/03/94 A	Henning Spruth, Frank Johannes, Kurt Antreich: PHIroute: A Parallel Hierarchical Sea-of-Gates Router
342/04/94 A	Martin Kiehl, Rainer Mehlhorn, Matthias Schumann: Parallel Multiple Shooting for Optimal Control Problems Under NX/2
342/05/94 A	Christian Suttner, Christoph Goller, Peter Krauss, Klaus-Jörn Lange, Ludwig Thomas, Thomas Schnekenburger: Heuristic Optimization of Parallel Computations
342/06/94 A	Andreas Listl: Using Subpages for Cache Coherency Control in Parallel Database Systems
342/07/94 A	Manfred Broy, Ketil Stølen: Specification and Refinement of Finite Dataflow Networks - a Relational Approach
342/08/94 A	Katharina Spies: Funktionale Spezifikation eines Kommunikationsprotokolls
342/09/94 A	Peter A. Krauss: Applying a New Search Space Partitioning Method to Parallel Test Generation for Sequential Circuits
342/10/94 A	Manfred Broy: A Functional Rephrasing of the Assumption/Commitment Specification Style
342/11/94 A	Eckhardt Holz, Ketil Stølen: An Attempt to Embed a Restricted Version of SDL as a Target Language in Focus
342/12/94 A	Christoph Pflaum: A Multi-Level-Algorithm for the Finite-Element-Solution of General Second Order Elliptic Differential Equations on Adaptive Sparse Grids
342/13/94 A	Manfred Broy, Max Fuchs, Thomas F. Gritzner, Bernhard Schätz, Katharina Spies, Ketil Stølen: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems

Reihe A

342/14/94 A	Maximilian Fuchs: Technologieabhängigkeit von Spezifikationen digitaler Hardware
342/15/94 A	M. Griebel, P. Oswald: Tensor Product Type Subspace Splittings And Multilevel Iterative Methods For Anisotropic Problems
342/16/94 A	Gheorghe Ştefănescu: Algebra of Flownomials
342/17/94 A	Ketil Stølen: A Refinement Relation Supporting the Transition from Unbounded to Bounded Communication Buffers
342/18/94 A	Michael Griebel, Tilman Neuhoeffer: A Domain-Oriented Multilevel Algorithm-Implementation and Parallelization
342/19/94 A	Michael Griebel, Walter Huber: Turbulence Simulation on Sparse Grids Using the Combination Method
342/20/94 A	Johann Schumann: Using the Theorem Prover SETHEO for verifying the development of a Communication Protocol in FOCUS - A Case Study -
342/01/95 A	Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A	Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law
342/03/95 A	Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space
342/04/95 A	Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes
342/05/95 A	Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems
342/06/95 A	Maximilian Fuchs: Formal Design of a Model-N Counter
342/07/95 A	Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology
342/08/95 A	Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies
342/09/95 A	Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness
342/10/95 A	Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design
342/11/95 A	Thomas Schnekenburger: The ALDY Load Distribution System
342/12/95 A	Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm
342/13/95 A	Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming
342/14/95 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks
342/15/95 A	Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs
342/16/95 A	Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik

Reihe A

342/17/95 A	Georg Stellner: Using CoCheck on a Network of Workstations
342/18/95 A	Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications
342/19/95 A	Thomas Schnekenburger: Integration of Load Distribution into ParMod-C
342/20/95 A	Ketil Stølen: Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication
342/21/95 A	Andreas Listl, Giannis Bozas: Performance Gains Using Subpages for Cache Coherency Control
342/22/95 A	Volker Heun, Ernst W. Mayr: Embedding Graphs with Bounded Treewidth into Optimal Hypercubes
342/23/95 A	Petr Jančar, Javier Esparza: Deciding Finiteness of Petri Nets up to Bisimulation
342/24/95 A	M. Jung, U. Rüde: Implicit Extrapolation Methods for Variable Coefficient Problems
342/01/96 A	Michael Griebel, Tilman Neunhoeffer, Hans Regler: Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometrics

SFB 342 : Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B	Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B	Jörg Desel: On Abstraction of Nets
342/3/90 B	Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B	Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B	Barbara Paech1: Concurrency as a Modality
342/2/91 B	Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier- Toolbox - Anwenderbeschreibung
342/3/91 B	Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B	Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B	Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B	Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B	W. Reisig: Concurrent Temporal Logic
342/1/92 B	Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B	Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B	Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung
342/2/93 B	Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B	Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS