

---

# Algebraic Flux Correction III. Incompressible Flow Problems

Stefan Turek<sup>1</sup> and Dmitri Kuzmin<sup>2</sup>

<sup>1</sup> Institute of Applied Mathematics (LS III), University of Dortmund  
Vogelpothsweg 87, D-44227, Dortmund, Germany

Stefan.Turek@mathematik.uni-dortmund.de

<sup>2</sup> kuzmin@math.uni-dortmund.de

**Summary.** Algebraic FEM-FCT and FEM-TVD schemes are integrated into incompressible flow solvers based on the ‘Multilevel Pressure Schur Complement’ (MPSC) approach. It is shown that algebraic flux correction is feasible for nonconforming (rotated bilinear) finite element approximations on unstructured meshes. Both (approximate) operator-splitting and fully coupled solution strategies are introduced for the discretized Navier-Stokes equations. The need for development of robust and efficient iterative solvers (outer Newton-like schemes, linear multigrid techniques, optimal smoothers/preconditioners) for implicit high-resolution schemes is emphasized. Numerical treatment of extensions (Boussinesq approximation,  $k - \varepsilon$  turbulence model) is addressed and pertinent implementation details are given. Simulation results are presented for three-dimensional benchmark problems as well as for prototypical applications including multiphase and granular flows.

## 1 Introduction

For single-phase Newtonian fluids occupying a domain  $\Omega \subset \mathbf{R}^d$  ( $d = 2, 3$ ) during the time interval  $(t_0, t_0 + T]$ , the incompressible Navier-Stokes equations

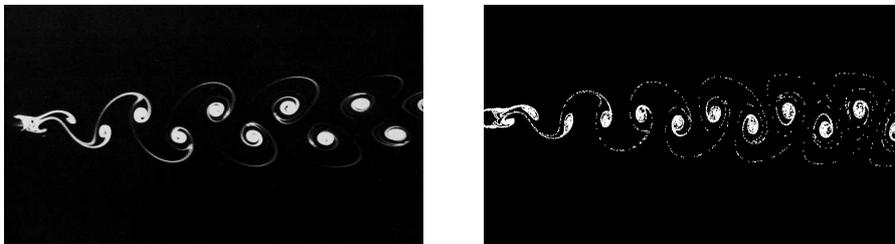
$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad , \quad (1)$$
$$\nabla \cdot \mathbf{u} = 0$$

describe the laminar flow motion which depends on the physical properties of the fluid (viscosity  $\nu$ ) and, possibly, on some external forces  $\mathbf{f}$  like buoyancy. The constant density  $\rho$  is “hidden” in the pressure  $p(x_1, \dots, x_d, t)$  which adjusts itself instantaneously so as to render the time-dependent velocity field  $\mathbf{u}(x_1, \dots, x_d, t)$  divergence-free. The problem statement is completed by specifying the initial and boundary values for each particular application.

Although these equations seem to have a quite simple structure, they constitute a ‘grand challenge’ problem for mathematicians, physicists, and engineers alike, and they are (still) object of intensive research activities in the field of *Computational Fluid Dynamics* (CFD). Incompressible flow problems are especially interesting from the viewpoint of applied mathematics and scientific computing, since they embody the whole range of difficulties which typically arise in the numerical treatment of partial differential equations. Therefore, they provide a perfect starting point for the development of reliable numerical algorithms and efficient software for CFD simulations.

Specifically, the problems which scientists and engineers are frequently confronted with, concern the following aspects:

- time-dependent partial differential equations in complex domains
- strongly nonlinear and stiff systems of intricately coupled equations
- convection-dominated transport at high Reynolds numbers ( $Re \approx \frac{1}{\nu}$ )
- saddle-point problems due to the incompressibility constraint
- local changes of the problem character in space and time



**Fig. 1.** Experiment (source: Van Dyke’s ‘Album of Fluid Motion’) vs. numerical simulation (source: ‘Virtual Album of Fluid Motion’) for flow around a cylinder.

These peculiarities of our model problem impose stringent requirements on virtually all stages of algorithm design: discretization, solver, and software engineering. In particular, the following difficulties must be reckoned with

- nonlinear systems for millions of unknowns (large but sparse matrices)
- conditional stability (explicit schemes) and/or proper time step control
- anisotropic/unstructured meshes (boundary layers, complex geometries)

Active research aimed at the development of improved numerical methods for the incompressible Navier-Stokes equations has been going on for more than three decades. The number of publications on this topic is enormous (see the book by Gresho et al. [14] for a comprehensive overview). However, in many cases the computational results produced by the available CFD tools are only *qualitatively* correct. A *quantitatively* precise flow prediction for real-life problems requires that the accuracy of discretization schemes be enhanced and/or the solvers become more efficient. This can be easily demonstrated by benchmark computations [40], especially for nonstationary flows.

Moreover, a current trend in CFD is to combine the ‘basic’ Navier-Stokes equations (2) with more or less sophisticated engineering models from physics and chemistry which describe industrial applications involving turbulence, multiphase flow, nonlinear fluids, combustion/detonation, free and moving boundaries, fluid-structure interaction, weakly compressible effects, etc. These extensions, some which will be discussed in the present chapter, have one thing in common: they require highly accurate and robust discretization techniques as well as efficient solution algorithms for generalized Navier–Stokes–like systems. In order to design and implement such powerful numerical methods for real-life problems, many additional aspects need to be taken into account.

The main ingredients of an ‘ultimate’ CFD code are as follows

- advanced mathematical methods for PDEs ( $\rightarrow$  *discretization*)
- efficient solution techniques for algebraic systems ( $\rightarrow$  *solver*)
- reliable and hardware-optimized software ( $\rightarrow$  *implementation*)

If all of these components were available, the number of unknowns could be significantly reduced (e.g., via adaptivity or a high-order approximation) and, moreover, discrete problems of the same size could be solved more efficiently. Hence, the marriage of optimal numerical methods and fast iterative solvers would make it possible to exploit the potential of modern computers to the full extent and enhance the performance of incompressible flow solvers (improve the MFLOP/s rates) by *orders of magnitude*. This is why these algorithmic aspects play an increasingly important role in contemporary CFD research.

In this contribution, we briefly review the Multilevel Pressure Schur Complement (MPSC) approach to solution of the incompressible Navier-Stokes equations and combine it with a FEM-FCT or FEM-TVD discretization of the (nonlinear) convective terms. We will explain the ramifications of these new (for the FEM community) algebraic high-resolution schemes as applied to incompressible flow problems and discuss a number of computational details regarding the efficient numerical solution of the resulting nonlinear and linear algebraic systems. Furthermore, we will examine different coupling mechanisms between the ‘basic’ flow model (standard Navier-Stokes equations for the velocity and pressure) and additional scalar or vector-valued transport equations. The primary goal is numerical simulation of high Reynolds number flows which require special stabilization techniques and/or advanced turbulence models in order to capture the relevant physical effects.

On the other hand, we also consider incompressible flows at intermediate and low Reynolds numbers. They may call for a different solution strategy but the evolution of scalar variables (temperatures, concentrations, probability densities, volume fractions, level set functions etc.) is still dominated by transport operators. Moreover, the transported quantities are inherently nonnegative in many cases. Therefore, standard discretization techniques may fail, whereas the positivity-preserving FCT/TVD schemes persevere and yield excellent results as the numerical examples in this chapter will illustrate.

## 2 Discretization of the Navier-Stokes Equations

Let us discretize the Navier–Stokes equations (2) in time by a standard method for numerical solution of ODEs. For instance, an implicit  $\theta$ -scheme (backward Euler or Crank–Nicolson) or its three-step counterpart proposed by Glowinski yields a sequence of boundary value problems of the form [47]

Given  $\mathbf{u}(t_n)$ , compute  $\mathbf{u} = \mathbf{u}(t_{n+1})$  and  $p = p(t_{n+1})$  by solving

$$\begin{aligned} [I + \theta \Delta t(\mathbf{u} \cdot \nabla - \nu \Delta)]\mathbf{u} + \Delta t \nabla p = [I - \theta_1 \Delta t(\mathbf{u}(t_n) \cdot \nabla - \nu \Delta)]\mathbf{u}(t_n) \\ + \theta_2 \Delta t \mathbf{f}(t_{n+1}) + \theta_3 \Delta t \mathbf{f}(t_n) \end{aligned} \quad (2)$$

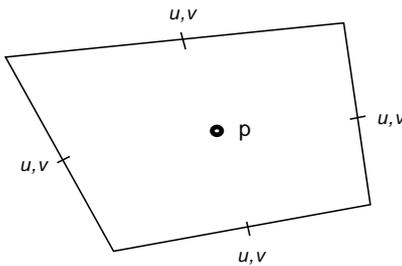
subject to the incompressibility constraint  $\nabla \cdot \mathbf{u} = 0$ .

For the spatial discretization, we choose a finite element approach. However finite volumes, finite differences or spectral methods are possible, too. A finite element model of the Navier–Stokes equations is based on a suitable variational formulation. On the finite mesh  $\mathcal{T}_h$  (triangles, quadrilaterals or their analogues in 3D) covering the domain  $\Omega$  with local mesh size  $h$ , one defines polynomial trial functions for velocity and pressure. These spaces  $H_h$  and  $L_h$  should lead to numerically stable approximations as  $h \rightarrow 0$ , i.e., they should satisfy the so-called *Babuška–Brezzi* (BB) condition [12]

$$\min_{q_h \in L_h} \max_{\mathbf{v}_h \in H_h} \frac{(q_h, \nabla \cdot \mathbf{v}_h)}{\|q_h\|_0 \|\nabla \mathbf{v}_h\|_0} \geq \gamma > 0 \quad (3)$$

with a mesh-independent constant  $\gamma$ . On the other hand, equal order interpolations for velocity and pressure are also admissible provided that an a priori unstable discretization is stabilized in an appropriate way (see, e.g., [19]).

In what follows, we employ the stable  $\tilde{Q}_1/Q_0$  finite element pair (*rotated bilinear/trilinear* shape functions for the velocities, and a piecewise constant pressure approximation). In the two-dimensional case, the nodal values are the mean values of the velocity vector over the element edges, and the mean values of the pressure over the elements (see Fig. 2).



**Fig. 2.** Nodal points of the nonconforming finite element pair  $\tilde{Q}_1/Q_0$ .

This **nonconforming** finite element is a quadrilateral counterpart of the well-known triangular Stokes element of Crouzeix–Raviart [6] and can easily be defined in three space dimensions. A convergence analysis is given in [39] and computational results are reported in [42] and [43]. An important advantage of this finite element pair is the availability of efficient multigrid solvers which are sufficiently robust in the whole range of Reynolds numbers even on nonuniform and highly anisotropic meshes [41],[47].

Using the notation  $\mathbf{u}$  and  $p$  also for the coefficient vectors in the representation of the approximate solution, the discrete version of problem (2) may be written as a coupled (nonlinear) algebraic system of the form:

Given  $\mathbf{u}^n$  and  $\mathbf{g}$ , compute  $\mathbf{u} = \mathbf{u}^{n+1}$  and  $p = p^{n+1}$  by solving

$$\mathbf{A}\mathbf{u} + \Delta t B p = \mathbf{g} \quad , \quad B^T \mathbf{u} = 0, \quad \text{where} \quad (4)$$

$$\mathbf{g} = [M - \theta_1 \Delta t N(\mathbf{u}^n)]\mathbf{u}^n + \theta_2 \Delta t \mathbf{f}^{n+1} + \theta_3 \Delta t \mathbf{f}^n. \quad (5)$$

Here  $M$  is the (consistent or lumped) mass matrix,  $B$  is the discrete gradient operator, and  $-B^T$  is the associated divergence operator. Furthermore,

$$\mathbf{A}\mathbf{u} = [M - \theta \Delta t N(\mathbf{u})]\mathbf{u}, \quad N(\mathbf{u}) = K(\mathbf{u}) + \nu L, \quad (6)$$

where  $L$  is the discrete Laplacian and  $K(\mathbf{u})$  is the nonlinear transport operator incorporating a certain amount of artificial diffusion. In the sequel, we assume that it corresponds to a FEM-TVD discretization of the convective term, although algebraic flux correction of FCT type or conventional stabilization techniques (upwinding, streamline diffusion) are also feasible.

The solution of nonlinear algebraic systems like (4) is a rather difficult task and many aspects need to be taken into account:

- **treatment of the nonlinearity:** fully nonlinear solution by Newton-like methods or iterative defect correction, explicit or implicit underrelaxation, monitoring of convergence rates, choice of stopping criteria etc.
- **treatment of the incompressibility:** strongly coupled approach (simultaneous solution for  $\mathbf{u}$  and  $p$ ) vs. segregated algorithms based on operator splitting at the continuous or discrete level (classical projection schemes [3],[56] and pressure correction methods like SIMPLE [8],[34]).
- **complete outer control:** problem-dependent degree of coupling and/or implicitness, optimal choice of linear algebra tools (iterative solvers and underlying smoothers/preconditioners), automatic time step control etc.

This abundance of choices leads to a great variety of incompressible flow solvers which are closely related to one another but exhibit considerable differences in terms of their stability, convergence, and efficiency. The Multilevel Pressure Schur Complement (MPSC) approach outlined below makes it possible to put many existing solution techniques into a common framework and combine their advantages so as to obtain better run-time characteristics. For a detailed presentation and a numerical study of the resulting schemes, the interested reader is referred to the monograph by Turek [47].

### 3 Pressure Schur Complement Solvers

The fully discretized Navier-Stokes equations (4) as well as the linear sub-problems to be solved within the outer iteration loop for a fixed-point defect correction or a Newton-like method admit the following representation

$$\begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}. \quad (7)$$

This is a saddle point problem in which the pressure acts as the Lagrange multiplier for the incompressibility constraint. In general, we have

$$A = \alpha M + \beta N(\mathbf{u}), \quad \text{where } \beta = -\theta \Delta t. \quad (8)$$

For time-dependent problems, the parameter  $\alpha$  is set equal to unity, whereas the steady-state formulation is recovered for  $\alpha := 0$  and  $\beta := -1$ .

If the operator  $A$  is nonsingular, the velocity can be formally expressed as

$$\mathbf{u} = A^{-1}(\mathbf{g} - \Delta t B p) \quad (9)$$

and plugged into the discretized continuity equation

$$B^T \mathbf{u} = 0 \quad (10)$$

which gives a scalar *Schur complement* equation for the pressure

$$B^T A^{-1} B p = \frac{1}{\Delta t} B^T A^{-1} \mathbf{g}. \quad (11)$$

Thus, the coupled system (7) can be handled as follows

1. Solve the Pressure Schur Complement (PSC) equation (11) for  $p$ .
2. Substitute  $p$  into relation (9) and compute the velocity  $\mathbf{u}$ .

It is worth mentioning that the matrix  $A^{-1}$  is full and should not be assembled explicitly. Instead, an auxiliary problem is to be solved by a direct method or by inner iterations. For instance, the velocity update (9) is equivalent to the solution of the discretized momentum equation  $A\mathbf{u} = \mathbf{g} - \Delta t B p$ .

Likewise, the matrix  $S := B^T A^{-1} B$  is never generated in practice. Doing so would be prohibitively expensive in terms of CPU time and memory requirements. It is instructive to consider a preconditioned Richardson method which yields the following **basic iteration** for the PSC equation

$$p^{(l+1)} = p^{(l)} - C^{-1} \left[ S p^{(l)} - \frac{1}{\Delta t} B^T A^{-1} \mathbf{g} \right], \quad l = 0, \dots, L-1. \quad (12)$$

Here  $C$  is a suitable preconditioner which is supposed to be a reasonable approximation to  $S$  but be easier to ‘invert’ in an iterative way.

The number of PSC cycles  $L$  can be fixed or chosen adaptively so as to achieve a prescribed tolerance for the residual. The choice  $C := S$  and  $L = 1$  is equivalent to the coupled solution of the original saddle-point problem (7). In principle, this challenging task can be accomplished by a properly configured multigrid method. However, the computational cost per iteration is very high and severe problems are sometimes observed on anisotropic grids that contain cells with high aspect ratios. Moreover, the convergence rates do not improve as the time step  $\Delta t$  is refined. Indeed, note that

$$A = M - \theta \Delta t N(\mathbf{u}) \approx M + \mathcal{O}(\Delta t) \quad (13)$$

for sufficiently small time steps. In this case,  $A$  can be interpreted as a nonsymmetric (and nonlinear) but well conditioned perturbation of the mass matrix  $M$ . On the other hand, the PSC equation (11) reveals that

$$\text{cond}(S) = \text{cond}(B^T [M + \mathcal{O}(\Delta t)]^{-1} B) \approx \text{cond}(L) = \mathcal{O}(h^{-2}). \quad (14)$$

It follows that the condition number of the coupled system (7) is bounded from below by  $\mathcal{O}(h^{-2})$  regardless of the time step. The nonsymmetric matrix  $A$  acting on the velocity components ‘improves’ for small  $\Delta t$  but, unfortunately, the overall convergence rates of coupled solvers depend also on the elliptic part  $B^T A^{-1} B \approx B^T M^{-1} B$  which is (almost) time step invariant.

In light of the above, the coupled solution strategy is inappropriate for numerical simulation of nonstationary flows which are dominated by convection and call for the use of small time steps. Hence, the preconditioner  $C$  for the Schur complement operator should be designed so as to take relation (13) into account. Let us consider ‘crude’ approximations of the form

$$C := B^T \tilde{A}^{-1} B, \quad (15)$$

where the matrix  $\tilde{A}$  should be readily ‘invertible’ but stay close to  $A$  at least in the limit  $\Delta t \rightarrow 0$ . Some typical choices are as follows

$$\tilde{A} := \text{diag}(A), \quad \tilde{A} := M_L, \quad \text{and} \quad \tilde{A} := M - \theta \Delta t \nu L.$$

Incompressible flow solvers based on the Richardson iteration (12) with this sort of preconditioning comprise fractional-step projection methods [7],[15],[36],[46], various modifications of the SIMPLE algorithm (for an overview, see Engelman [8] and the literature cited therein) as well as Uzawa-like iterations. They can be classified as *global pressure Schur complement* schemes due to the fact that  $C$  is an approximation to the global matrix  $S = B^T A^{-1} B$ .

On the other hand, coupled solution techniques are to be recommended for the treatment of (quasi-) stationary flows and Navier-Stokes equations combined with RANS turbulence models and/or convection-diffusion equations for other scalar quantities (temperatures, concentrations etc). In this case, it is worthwhile to approximate the pressure Schur complement operator *locally*

via direct inversion of small matrix blocks associated with subdomains  $\Omega_i$  of the domain  $\Omega$  or, in general, with a subset of the unknowns to be solved for. The resulting *local pressure Schur complement* techniques correspond to

$$C^{-1} := \sum_i (B_{|\Omega_i}^T A_{|\Omega_i}^{-1} B_{|\Omega_i})^{-1}, \quad (16)$$

whereby the *patches*  $\Omega_i$  are usually related to the underlying mesh and can consist of single elements or element clusters. The global relaxation scheme is obtained by embedding these “local solvers” into an outer iteration loop of Jacobi or Gauss–Seidel type. This strategy has a lot in common with *domain decomposition* methods, but is more flexible when it comes to the treatment of boundary conditions. A typical representative of local PSC schemes is the Vanka smoother [55] which is widely used in the multigrid community.

Furthermore, it is possible to combine incompressible flow solvers based on global PSC (“operator splitting”) and local PSC (“domain decomposition”) methods in a general-purpose CFD code. Indeed, all of these seemingly different solution techniques utilize *additive preconditioners* of the form

$$C^{-1} := \sum_i \alpha_i C_i^{-1}.$$

In the next two sections, we briefly discuss the design of such preconditioners and present the resulting basic iteration schemes which can be used as

- preconditioners for Krylov space methods (CG, BiCGSTAB, GMRES)
- *Multilevel pressure Schur complement* (MPSC) smoothers for multigrid

The multigrid approach is usually more efficient, as demonstrated by benchmark computations in [40] (see also the numerical examples below).

## 4 Global MPSC Approach

The basic idea behind the family of global MPSC schemes is the construction of globally defined additive preconditioners for the Schur complement operator  $S = B^T A^{-1} B$ . Recall that the matrix  $A$  has the following structure

$$A := \alpha M + \beta K(\mathbf{u}) + \gamma L, \quad (17)$$

where  $\beta = -\theta \Delta t$  and  $\gamma = \nu \beta$ . Unfortunately, it is hardly possible to construct a matrix  $\tilde{A}$  and a preconditioner  $C = B^T \tilde{A}^{-1} B$  that would be a sufficiently good approximation to all three components of  $A$  and  $S$ , respectively. Therefore, one can start with developing individual preconditioners for the reactive ( $M$ ), convective ( $K$ ), and diffusive ( $L$ ) part. In other words, the original problem can be decomposed into simpler tasks by resorting to operator splitting.

Let the inverse of  $C$  be composed from those of ‘optimal’ preconditioners for the limiting cases of a divergence-free  $L_2$ -projection (for small time steps), incompressible Euler equations, and a diffusion-dominated Stokes problem

$$C^{-1} = \alpha' C_M^{-1} + \beta' C_K^{-1} + \gamma' C_L^{-1} \approx S^{-1}, \quad (18)$$

where the user-defined parameters  $(\alpha', \beta', \gamma')$  may toggle between  $(\alpha, \beta, \gamma)$  and zero depending on the flow regime. Furthermore, it is implied that

$C_M$  is an ‘optimal’ approximation of the reactive part  $B^T M^{-1} B$ ,

$C_K$  is an ‘optimal’ approximation of the convective part  $B^T K^{-1} B$ ,

$C_L$  is an ‘optimal’ approximation of the diffusive part  $B^T L^{-1} B$ .

The meaning of ‘optimality’ has to be defined more precisely. Ideally, partial preconditioners should be direct solvers with respect to the underlying subproblem. In fact, this may even be true for the fully ‘reactive’ case  $S = B^T M^{-1} B$ . However, if these preconditioners are applied as smoothers in a multigrid context and the convergence rates are largely independent of outer parameter settings as well as of the underlying mesh, then this is already sufficient for optimality of the global MPSC solver. Preconditioners  $C_M$ ,  $C_K$ , and  $C_L$  satisfying this criterion are introduced and analyzed in [47].

At high Reynolds numbers, the time steps must remain small due to the physical scales of flow motion. Therefore, the lumped mass matrix  $M_L$  proves to be a reasonable approximation to the complete operator  $A$ . In this case, our basic iteration (12) for the pressure Schur complement equation

$$p^{(l+1)} = p^{(l)} + [B^T M_L^{-1} B]^{-1} \frac{1}{\Delta t} B^T A^{-1} [\mathbf{g} - \Delta t B p^{(l)}] \quad (19)$$

can be interpreted and implemented as a *discrete projection scheme* such as those proposed in [7],[15],[36]. The main algorithmic steps are as follows [46]

1. Solve the ‘viscous Burgers’ equation for  $\tilde{\mathbf{u}}$

$$A \tilde{\mathbf{u}} = \mathbf{g} - \Delta t B p^{(l)}.$$

2. Solve the discrete ‘Pressure-Poisson’ problem

$$B^T M_L^{-1} B q = \frac{1}{\Delta t} B^T \tilde{\mathbf{u}}.$$

3. Correct the pressure and the velocity

$$p^{(l+1)} = p^{(l)} + q, \quad \mathbf{u} = \tilde{\mathbf{u}} - \Delta t M_L^{-1} B q.$$

In essence, the right-hand side of the momentum equation is assembled using the old pressure iterate and the intermediate velocity  $\tilde{\mathbf{u}}$  is projected onto the subspace of solenoidal functions so as to satisfy the constraint  $B^T \mathbf{u} = 0$ .

The matrix  $B^T M_L^{-1} B$  corresponds to a mixed discretization of the Laplacian operator [15] so that this method is a discrete analogue of the classical projection schemes derived by Chorin ( $p^{(0)} = 0$ ) and Van Kan ( $p^{(0)} = p(t_n)$ ) via operator splitting for the continuous problem [3],[56]. For an in-depth presentation of continuous projection schemes we refer to [35],[36]. Our discrete approach offers a number of important advantages including

- applicability to discontinuous pressure approximations
- consistent treatment (no splitting) of boundary conditions
- alleviation of spurious boundary layers for the pressure
- convergence to the fully coupled solution as  $l$  increases
- remarkable efficiency for nonstationary flow problems

On the other hand, discrete projection methods lack the inherent stabilization mechanisms that make their continuous counterparts applicable to equal-order interpolations provided that the time step is not too small [35].

In our experience, it is often sufficient to perform exactly  $L = 1$  pressure Schur complement iteration with just one multigrid sweep. Due to the fact that the numerical effort for solving the linear subproblems is insignificant, global MPSC methods are much more efficient than coupled solvers in the high Reynolds number regime. However, they perform so well only for relatively small time steps, so that the more robust local MPSC schemes are to be recommended for low Reynolds number flows.

## 5 Local MPSC Approach

The local pressure Schur complement approach is tailored to solving ‘small’ problems so as to exploit the fast cache of modern processors, in contrast to the readily vectorizable global MPSC schemes. As already mentioned above, the basic idea is to subdivide the complete set of unknowns into patches  $\Omega_i$  and solve the local subproblems **exactly** within an outer block-Gauss-Seidel/Jacobi iteration loop. Typically, every patch (macroelement) for this ‘domain decomposition’ method consists of one or several neighboring mesh cells and the corresponding local ‘stiffness matrix’  $C_i$  is given by

$$C_i := \begin{bmatrix} \tilde{A}|_{\Omega_i} & \Delta t B|_{\Omega_i} \\ B|_{\Omega_i}^T & 0 \end{bmatrix}. \quad (20)$$

Its coefficients (and hence the corresponding ‘boundary conditions’ for the subdomains) are taken from the global matrices, whereby  $\tilde{A}$  may represent either the complete velocity matrix  $A$  or some approximation of it, for instance, the diagonal part  $\text{diag}(A)$ . The local subproblems at hand are so small that they can be solved directly by Gaussian elimination. This is equivalent to applying the inverse of  $C_i$  to a portion of the global defect vector.

The elimination process leads to a fill-in of the matrix, which increases the storage requirements dramatically. Thus, it is advisable to solve the equivalent local pressure Schur complement problem with the compact matrix

$$S_i := B_{|\Omega_i}^T \tilde{A}_{|\Omega_i}^{-1} B_{|\Omega_i}. \quad (21)$$

In general,  $S_i$  is a full matrix but it is much smaller than  $C_i$ , since only the pressure values are solved for. If the patch  $\Omega_i$  contains just a moderate number of elements, the pressure Schur complement matrix is likely to fit into the processor cache. Having solved the local PSC subproblem, one can recover the corresponding velocity field as described in the previous section.

In any case, the basic iteration for a local MPSC method reads

$$\begin{bmatrix} \mathbf{u}^{(l+1)} \\ p^{(l+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{(l)} \\ p^{(l)} \end{bmatrix} - \omega^{(l+1)} \sum_{i=1}^{N_p} \begin{bmatrix} \tilde{A}_{|\Omega_i} & \Delta t B_{|\Omega_i} \\ B_{|\Omega_i}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \delta \mathbf{u}_i^{(l)} \\ \delta p_i^{(l)} \end{bmatrix}, \quad (22)$$

where  $N_p$  denotes the total number of patches,  $\omega^{(l+1)}$  is a relaxation parameter, and the global defect vector restricted to a single patch  $\Omega_i$  is given by

$$\begin{bmatrix} \delta \mathbf{u}_i^{(l)} \\ \delta p_i^{(l)} \end{bmatrix} = \left( \begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(l)} \\ p^{(l)} \end{bmatrix} - \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \right)_{|\Omega_i}. \quad (23)$$

In practice, we solve the corresponding auxiliary problem

$$\begin{bmatrix} \tilde{A}_{|\Omega_i} & \Delta t B_{|\Omega_i} \\ B_{|\Omega_i}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_i^{(l+1)} \\ q_i^{(l+1)} \end{bmatrix} = \begin{bmatrix} \delta \mathbf{u}_i^{(l)} \\ \delta p_i^{(l)} \end{bmatrix} \quad (24)$$

and compute the new iterates  $\mathbf{u}_{|\Omega_i}^{(l+1)}$  and  $p_{|\Omega_i}^{(l+1)}$  as follows

$$\begin{bmatrix} \mathbf{u}_{|\Omega_i}^{(l+1)} \\ p_{|\Omega_i}^{(l+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{|\Omega_i}^{(l)} \\ p_{|\Omega_i}^{(l)} \end{bmatrix} - \omega^{(l+1)} \begin{bmatrix} \mathbf{v}_i^{(l+1)} \\ q_i^{(l+1)} \end{bmatrix}. \quad (25)$$

This two-step relaxation procedure is applied to each patch, so some velocity or pressure components may end up being updated several times. The easiest way to obtain globally defined solution values at subdomain boundaries is to overwrite the contributions of previously processed patches or to calculate an average over all patch contributions to the computational node.

The resulting local MPSC method corresponds to a simple block-Jacobi iteration for the mixed problem (4). Its robustness and efficiency can be easily enhanced by computing the local defect vector (23) using the possibly updated solution values rather than the old iterates  $\mathbf{u}_{|\Omega_i}^{(l)}$  and  $p_{|\Omega_i}^{(l)}$  for the degrees of freedom shared with other patches. This strategy is known as the block-Gauss-Seidel method. Its performance is superior to that of the block-Jacobi scheme, while the numerical effort is approximately the same (for a sequential code).

It is common knowledge that block-iterative methods of Jacobi and Gauss-Seidel type do a very good job as long as there are no strong mesh anisotropies. However, the convergence rates deteriorate dramatically for irregular triangulations which contain elements with high aspect ratios (for example, stretched cells needed to resolve a boundary layer) and/or large differences between the size of two neighboring elements. The use of ILU techniques alleviates this problem but is impractical for strongly coupled systems of equations. A much better remedy is to combine the mesh elements so as to ‘hide’ the detrimental anisotropies inside of the patches which are supposed to have approximately the same shape and size. Several adaptive blocking strategies for generation of such isotropic subdomains are described in [41],[47].

The global convergence behavior will be satisfactory because only the local subproblems are ill-conditioned. Moreover, the size of these local problems is usually very small. Thus, the complete inverse of the matrix fits into RAM and sometimes even into the cache so that the use of fast direct solvers is feasible. Consequently, the convergence rates should be independent of grid distortions and approach those for very regular structured meshes. If hardware-optimized routines such as the BLAS libraries are employed, then the solution of small subproblems can be performed very efficiently. Excellent convergence rates and a high overall performance can be achieved if the code is properly tuned and adapted to the processor architecture in each particular case.

## 6 Multilevel Solution Strategy

Pressure Schur complement schemes constitute viable solution techniques as such but they are particularly useful as smoothers for a *multilevel* algorithm, e.g., a geometric multigrid method. Let us start with explaining the typical implementation of such a solver for an abstract linear system of the form

$$A_N u_N = f_N. \quad (26)$$

It is assumed that there exists a hierarchy of levels  $k = 1, \dots, N$  which may be characterized, for instance, by the mesh size  $h_k$ . On each of these levels, one needs to assemble the matrix  $A_k$  and the right-hand side  $f_k$  for the discrete problem. We remark that only  $f_N$  is available a priori, while the sequence of residual vectors  $\{f_k\}$  for  $k < N$  is generated during the multigrid run.

The main ingredients of a (linear) multigrid algorithm are

- matrix-vector multiplication routines for the operators  $A_k$ ,  $k \leq N$
- an efficient *smoother* (basic iteration scheme) and a *coarse grid solver*
- *prolongation*  $I_{k-1}^k$  and *restriction*  $I_k^{k-1}$  operators for grid transfer

Each  $k$ -level iteration  $MPSC(k, u_k^0, f_k)$  with initial guess  $u_k^0$  represents a *multigrid cycle* which yields an (approximate) solution of the linear system  $A_k u_k = f_k$ . On the first level, the number of unknowns is typically so small that the auxiliary problem can be solved directly:  $MPSC(1, u_1^0, f_1) = A_1^{-1} f_1$ .

For all other levels ( $k > 1$ ), the following algorithm is adopted [47]

**Step 1.** Presmoothing

Apply  $m$  smoothing steps (PSC iterations) to  $u_k^0$  to obtain  $u_k^m$ .

**Step 2.** Coarse grid correction

Calculate  $f_{k-1}$  using the restriction operator  $I_k^{k-1}$  via

$$f_{k-1} = I_k^{k-1}(f_k - A_k u_k^m)$$

and let  $u_{k-1}^i$  ( $1 \leq i \leq p$ ) be defined recursively by

$$u_{k-1}^i = MPSC(k-1, u_{k-1}^{i-1}, f_{k-1}), \quad u_{k-1}^0 = 0.$$

**Step 3.** Relaxation and update

Calculate  $u_k^{m+1}$  using the prolongation operator  $I_{k-1}^k$  via

$$u_k^{m+1} = u_k^m + \alpha_k I_{k-1}^k u_{k-1}^p, \quad (27)$$

where the relaxation parameter  $\alpha_k$  may be fixed or chosen adaptively so as to minimize the error  $u_k^{m+1} - u_k$  in an appropriate norm, for instance, in the discrete energy norm

$$\alpha_k = \frac{(f_k - A_k u_k^m, I_{k-1}^k u_{k-1}^p)_k}{(A_k I_{k-1}^k u_{k-1}^p, I_{k-1}^k u_{k-1}^p)_k}.$$

**Step 4.** Postsmoothing

Apply  $n$  smoothing steps (PSC iterations) to  $u_k^{m+1}$  to obtain  $u_k^{m+n+1}$ .

After sufficiently many cycles on level  $N$ , the desired solution  $u_N$  of the generic problem (26) is recovered. In the framework of our multilevel pressure Schur complement schemes, there are (at least) two possible scenarios:

**Global MPSC approach** Solve the discrete problem (26) with

$$A_N := B^T A^{-1} B, \quad u_N := p, \quad f_N := \frac{1}{\Delta t} B^T A^{-1} \mathbf{g}.$$

The basic iteration is given by (19) and equivalent to a discrete projection cycle, whereby the velocity field  $\mathbf{u}$  is updated in a parallel manner (see above). The bulk of CPU time is spent on matrix-vector multiplications with the Schur complement operator  $S = B^T A^{-1} B$  which is needed for smoothing, defect calculation, and adaptive coarse grid correction. Unlike standard multi-grid methods for scalar problems, global MPSC schemes involve solutions of

a viscous Burgers equation and a Poisson-like problem in each matrix-vector multiplication step (to avoid matrix inversion). In the case of highly nonstationary flows, the overhead cost is insignificant but it becomes appreciable as the Reynolds number decreases. Nevertheless, numerical tests indicate that the resulting multigrid solvers are optimal in the sense that the convergence rates are excellent and largely independent of mesh anisotropies.

**Local MPSC approach** Solve the discrete problem (26) with

$$A_N := \begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix}, \quad u_N := \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix}, \quad f_N := \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}.$$

The basic iteration is given by (22) which corresponds to the block-Gauss-Seidel/Jacobi method. The cost-intensive part is the smoothing step, as in the case of standard multigrid techniques for convection-diffusion equations and Poisson-like problems. Local MPSC schemes lead to very robust solvers for coupled problems. This multilevel solution strategy is to be recommended for incompressible flows at low and intermediate Reynolds numbers.

Further algorithmic details (adaptive coarse grid correction, grid transfer operators, nonlinear iteration techniques, time step control, implementation of boundary conditions) and a description of the high-performance software package FEATFLOW based on MPSC solvers can be found in [47],[48]. Some programming strategies, data structures, and guidelines for the development of a hardware-oriented parallel code are presented in [49],[50],[51].

## 7 Coupling with Scalar Equations

Both global and local MPSC schemes are readily applicable to the Navier-Stokes equations coupled with various turbulence models and/or scalar conservation laws for temperatures, concentrations, volume fractions, and other scalar variables. In many cases, the quantities of interest must remain strictly nonnegative for physical reasons, and the failure to enforce the positivity constraint for the *numerical* solution may have disastrous consequences. Therefore, a positivity-preserving discretization of convective terms is indispensable for such applications. This prerequisite is clearly satisfied by the algebraic FEM-FCT and FEM-TVD schemes introduced in the previous chapters.

As a representative example of a two-way coupling between (2) and a scalar transport equation, we consider the well-known Boussinesq approximation for natural convection problems. The nondimensional form of the governing equations for a buoyancy-driven incompressible flow reads [4]

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \nu \Delta \mathbf{u} + T \mathbf{e}_g, \quad (28)$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = d \Delta T, \quad \nabla \cdot \mathbf{u} = 0, \quad (29)$$

where  $\mathbf{u}$  is the velocity,  $p$  is the deviation from hydrostatic pressure and  $T$  is the temperature. The unit vector  $\mathbf{e}_g$  is directed ‘upward’ (opposite to the gravitational force) and the nondimensional diffusion coefficients

$$\nu = \sqrt{\frac{Pr}{Ra}}, \quad d = \sqrt{\frac{1}{Ra Pr}}$$

depend on the Rayleigh number  $Ra$  and the Prandtl number  $Pr$ . Details of this model and parameter settings for the *MIT benchmark problem* (natural convection in a differentially heated enclosure) can be found in [4].

### 7.1 Finite element discretization

After the discretization in space and time, we obtain a system of nonlinear algebraic equations which can be written in matrix form as follows

$$A_u(\mathbf{u}^{n+1})\mathbf{u}^{n+1} + \Delta t M_T T^{n+1} + \Delta t B p^{n+1} = \mathbf{f}_u, \quad (30)$$

$$A_T(\mathbf{u}^{n+1})T^{n+1} = f_T, \quad B^T \mathbf{u}^{n+1} = 0. \quad (31)$$

Here and below the superscript  $n+1$  refers to the time level, while subscripts identify the origin of discrete operators ( $u$  for the momentum equation and  $T$  for the heat conduction equation). Furthermore, the matrices  $A_u$  and  $A_T$  can be decomposed into a reactive, convective, and diffusive part

$$A_u(\mathbf{v}) = \alpha_u M_u + \beta_u K_u(\mathbf{v}) + \gamma_u L_u, \quad (32)$$

$$A_T(\mathbf{v}) = \alpha_T M_T + \beta_T K_T(\mathbf{v}) + \gamma_T L_T. \quad (33)$$

Note that we have the freedom of using different finite element approximations and discretization schemes for the velocity  $\mathbf{u}$  and temperature  $T$ .

The discrete problem (30)–(31) admits the following representation

$$\begin{bmatrix} A_u(\mathbf{u}^{n+1}) & \Delta t M_T & \Delta t B \\ 0 & A_T(\mathbf{u}^{n+1}) & 0 \\ B^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ T^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ f_T \\ 0 \end{bmatrix} \quad (34)$$

and can be solved in the framework of a global or local MPSC method.

### 7.2 Global MPSC algorithm

Nonstationary flow configurations call for the use of operator splitting tools for the coupled system (34). This straightforward approach consists in solving the Navier-Stokes equations for  $(\mathbf{u}, p)$  and the energy equation for  $T$  in a segregated manner. The decoupled subproblems are embedded into an outer iteration loop and solved sequentially by a global MPSC method (discrete projection) and an algebraic FCT/TVD scheme, respectively. For relatively small time steps, this strategy works very well, and simulation software can be developed in a modular way making use of optimized multigrid solvers. Moreover, it is possible to choose the time step individually for each subproblem.

In the simplest case (just one outer iteration per time step), the sequence of algorithmic steps to be performed is as follows [52]

<p>1. Compute <math>\tilde{\mathbf{u}}</math> from the momentum equation</p> $A_u(\tilde{\mathbf{u}})\tilde{\mathbf{u}} = \mathbf{f}_u - \Delta t M_T T^n - \Delta t B p^n.$ <p>2. Solve the discrete Pressure-Poisson problem</p> $B^T M_L^{-1} B q = \frac{1}{\Delta t} B^T \tilde{\mathbf{u}}.$ <p>3. Correct the pressure and the velocity</p> $p^{n+1} = p^n + q, \quad \mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t M_L^{-1} B q.$ <p>4. Solve the convection-diffusion equation for <math>T</math></p> $A_T(\mathbf{u}^{n+1}) T^{n+1} = f_T.$
--

Due to the nonlinearity of the discretized convective terms, iterative defect correction or a Newton-like method must be invoked in steps 1 and 4. This algorithm combined with the nonconforming FEM discretization appears to provide an ‘optimal’ flow solver for unsteady natural convection problems.

### 7.3 Local MPSC algorithm

Alternatively, a fully coupled solution of the problem at hand can be obtained following the local MPSC approach. To this end, a multigrid solver is applied to the suitably linearized **coupled** system (34). Each outer iteration for the nonlinearity corresponds to the following solution update [41],[52]

$$\begin{bmatrix} \mathbf{u}^{(l+1)} \\ T^{(l+1)} \\ p^{(l+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{(l)} \\ T^{(l)} \\ p^{(l)} \end{bmatrix} - \omega^{(l+1)} [F(\sigma, l)]^{-1} \begin{bmatrix} \delta \mathbf{u}^{(l)} \\ \delta T^{(l)} \\ \delta p^{(l)} \end{bmatrix}, \quad (35)$$

where the global defect vector is given by the relation

$$\begin{bmatrix} \delta \mathbf{u}^{(l)} \\ \delta T^{(l)} \\ \delta p^{(l)} \end{bmatrix} = \begin{bmatrix} A_u(\mathbf{u}^{(l)}) & \Delta t M_T & \Delta t B \\ 0 & A_T(\mathbf{u}^{(l)}) & 0 \\ B^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(l)} \\ T^{(l)} \\ p^{(l)} \end{bmatrix} - \begin{bmatrix} \mathbf{f}_u \\ f_T \\ 0 \end{bmatrix} \quad (36)$$

and the matrix to be inverted corresponds to the (approximate) Fréchet derivative of the underlying PDE system such that [41],[47]

$$F(\sigma, l) = \begin{bmatrix} A_u(\mathbf{u}^{(l)}) + \sigma R(\mathbf{u}^{(l)}) & \Delta t M_T & \Delta t B \\ \sigma R(T^{(l)}) & A_T(\mathbf{u}^{(l)}) & 0 \\ B^T & 0 & 0 \end{bmatrix}. \quad (37)$$

The nonlinearity of the governing equations gives rise to the ‘reactive’ contribution  $R$  which represents a solution-dependent mass matrix and may cause severe convergence problems. This is why it is multiplied by the adjustable parameter  $\sigma$ . The Newton method is recovered for  $\sigma = 1$ , while the value  $\sigma = 0$  yields the fixed-point defect correction scheme. In either case, the linearized problem is solved by a fully coupled multigrid solver equipped with a local MPSC smoother of ‘Vanka’ type [41]. As before, the matrix  $F(\sigma, l)$  is decomposed into small blocks  $C_i$  associated with individual patches  $\Omega_i$ . The smoothing of the global residual vector is performed patchwise by solving the corresponding local subproblems.

The size of the matrices to be inverted can be further reduced by resorting to the Schur complement approach. For simplicity, consider the case  $\sigma = 0$  (extension to  $\sigma > 0$  is straightforward). It follows from (30)–(31) that

$$T^{n+1} = A_T^{-1} f_T, \quad \mathbf{u}^{n+1} = A_u^{-1} [\mathbf{f}_u - \Delta t M_T T^{n+1} - \Delta t B p^{n+1}] \quad (38)$$

and the discretized continuity equation can be cast into the form

$$B^T \mathbf{u}^{n+1} = B^T A_u^{-1} [\mathbf{f}_u - \Delta t M_T A_T^{-1} f_T - \Delta t B p^{n+1}] = 0 \quad (39)$$

which corresponds to the pressure Schur complement equation

$$B^T A_u^{-1} B p^{n+1} = B^T A_u^{-1} \left[ \frac{1}{\Delta t} \mathbf{f}_u - M_T A_T^{-1} f_T \right]. \quad (40)$$

Thus, highly efficient local preconditioners of the form (21) can be employed instead of  $C_i$ . The converged solution  $p^{n+1}$  to the scalar subproblem (40) is plugged into (38) to obtain the velocity  $\mathbf{u}^{n+1}$  and the temperature  $T^{n+1}$ .

The advantages of the seemingly complicated local MPSC strategy are as follows. First of all, steady-state solutions can be obtained without resorting to pseudo-time-stepping. Moreover, the fully coupled treatment of dynamic flows makes it possible to use large time steps without any loss of robustness. On the other hand, the convergence behavior of multigrid solvers for the Newton linearization may turn out to be unsatisfactory and the computational cost per outer iteration is rather high as compared to the global MPSC algorithm. The performance of both solution techniques as applied to the MIT benchmark problem is illustrated by the numerical results reported in [52].

## 8 Implementation of the $k - \varepsilon$ Model

High-resolution schemes like FCT and TVD play an increasingly important role in simulation of turbulent flows. Flow structures that cannot be resolved on the computational mesh activate the flux limiter which curtails the raw antidiffusion so as to filter out the small-scale fluctuations. Interestingly enough, the residual artificial viscosity provides an excellent subgrid scale model for *monotonically integrated* Large Eddy Simulation (MILES), see [2],[16].

In spite of recent advances in the field of LES and DNS (direct numerical simulation), simpler turbulence models based on Reynolds averaging (RANS) still prevail in CFD software for simulation of industrial processes. In particular, the evolution of the turbulent kinetic energy  $k$  and of its dissipation rate  $\varepsilon$  is governed by two convection-dominated transport equations

$$\frac{\partial k}{\partial t} + \nabla \cdot \left( k \mathbf{u} - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k - \varepsilon, \quad (41)$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left( \varepsilon \mathbf{u} - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k} (C_1 P_k - C_2 \varepsilon), \quad (42)$$

where  $\mathbf{u}$  denotes the averaged velocity,  $\nu_T = C_\mu k^2 / \varepsilon$  is the turbulent eddy viscosity and  $P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2$  is the production term. For the standard  $k - \varepsilon$  model, the default values of the involved parameters are as follows

$$C_\mu = 0.09, \quad C_1 = 1.44, \quad C_2 = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3.$$

The velocity field  $\mathbf{u}$  is obtained from the incompressible Navier-Stokes equations with  $\nabla \cdot (\nu + \nu_T)[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$  instead of  $\nu \Delta \mathbf{u}$ .

We remark that the transport equations for  $k$  and  $\varepsilon$  are strongly coupled and nonlinear so that their numerical solution is a very challenging task. Moreover, the discretization scheme **must** be positivity-preserving because negative values of the eddy viscosity are totally unacceptable. Unfortunately, implementation details and employed ‘tricks’ are rarely reported in the literature, so that a novice to this area of CFD research often needs to reinvent the wheel. Therefore, we deem it appropriate to discuss the implementation of a FEM-TVD algorithm for the  $k - \varepsilon$  model in some detail.

### 8.1 Positivity-preserving linearization

The block-iterative algorithm proposed in [27],[28] consists of nested loops so that the coupled PDE system is replaced by a sequence of linear sub-problems. The solution-dependent coefficients are ‘frozen’ during each outer iteration and updated as new values become available. The quasi-linear transport equations can be solved by an implicit FEM-FCT or FEM-TVD scheme but the linearization procedure must be tailored to the need to preserve the positivity of  $k$  and  $\varepsilon$  in a numerical simulation. Due to the presence of sink terms in the right-hand side of both equations, the positivity constraint may be violated even if a high-resolution scheme is employed for the discretization of convective terms. It can be proved that the exact solution to the  $k - \varepsilon$  model remains nonnegative for positive initial data [32],[33] and it is essential to guarantee that the numerical scheme will also possess this property.

Let us consider the following representation of the equations at hand [29]

$$\frac{\partial k}{\partial t} + \nabla \cdot (k \mathbf{u} - d_k \nabla k) + \gamma k = P_k, \quad (43)$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon \mathbf{u} - d_\varepsilon \nabla \varepsilon) + C_2 \gamma \varepsilon = C_1 P_k, \quad (44)$$

where the parameter  $\gamma = \frac{\varepsilon}{k}$  is proportional to the specific dissipation rate ( $\gamma = C_\mu \omega$ ). The turbulent dispersion coefficients are given by  $d_k = \frac{\nu_T}{\sigma_k}$  and  $d_\varepsilon = \frac{\nu_T}{\sigma_\varepsilon}$ . By definition, the source terms in the right-hand side are nonnegative. Furthermore, the parameters  $\nu_T$  and  $\gamma$  must also be nonnegative for the solution of the convection-reaction-diffusion equations to be well-behaved [5]. In our numerical algorithm, their values are taken from the previous iteration and their positivity is secured as explained below. This linearization technique was proposed by Lew *et al.* [29] who noticed that the positivity of the lagged coefficients is even more important than that of the transported quantities and can be readily enforced without violating the discrete conservation principle.

Applying an implicit FCT/TVD scheme to the above equations, we obtain two nonlinear algebraic systems which can be written in the generic form

$$A(u^{(l+1)})u^{(l+1)} = B(u^{(l)})u^{(l)} + q^{(k)}, \quad l = 0, 1, 2, \dots \quad (45)$$

Here  $k$  is the index of the outermost loop in which the velocity  $\mathbf{u}$  and the source term  $P_k$  are updated. The index  $l$  refers to the outer iteration for the  $k - \varepsilon$  model, while the index  $m$  is reserved for inner flux/defect correction loops. The structure of the matrices  $A$  and  $B$  is as follows:

$$A(u) = M_L - \theta \Delta t (K^*(u) + T), \quad (46)$$

$$B(u) = M_L + (1 - \theta) \Delta t (K^*(u) + T), \quad (47)$$

where  $K^*(u)$  is the LED transport operator incorporating nonlinear anti-diffusion and  $T$  denotes the standard reaction-diffusion operator which is a symmetric positive-definite matrix with nonnegative off-diagonal entries. It is obvious that the discretized production terms  $q^{(k)}$  are also nonnegative. Thus, the positivity of  $u^{(l)}$  is inherited by the new iterate  $u^{(l+1)} = A^{-1}(Bu^{(l)} + q^{(k)})$  provided that  $\theta = 1$  (backward Euler) or the time step is sufficiently small.

## 8.2 Positivity of coefficients

The predicted values  $k^{(l+1)}$  and  $\varepsilon^{(l+1)}$  are used to recompute the parameter  $\gamma^{(l+1)}$  for the next outer iteration (if any). The turbulent eddy viscosity  $\nu_T^{(k)}$  is updated in the outermost loop. In the turbulent flow regime  $\nu_T \gg \nu$  and the laminar viscosity  $\nu$  can be neglected. Hence, we set  $\nu_{\text{eff}} = \nu_T$ , where the eddy viscosity  $\nu_T$  is bounded from below by  $\nu$  and from above by the maximum admissible mixing length  $l_{\text{max}}$  (e.g. the width of the computational domain). Specifically, we define the limited mixing length  $l_*$  as

$$l_* = \begin{cases} \frac{\alpha}{\varepsilon} & \text{if } \varepsilon > \frac{\alpha}{l_{\text{max}}} \\ l_{\text{max}} & \text{otherwise} \end{cases}, \quad \text{where } \alpha = C_\mu k^{3/2} \quad (48)$$

and use it to update the turbulent eddy viscosity  $\nu_T$  in the outermost loop:

$$\nu_T = \max\{\nu, l_* \sqrt{k}\} \quad (49)$$

as well as the parameter  $\gamma$  in each outer iteration for the  $k - \varepsilon$  model:

$$\gamma = C_\mu \frac{k}{\nu_*}, \quad \text{where } \nu_* = \max\{\nu, l_* \sqrt{k}\}. \quad (50)$$

In the case of a FEM-TVD method, the positivity proof is only valid for the converged solution to (45) while intermediate solution values may be negative. Since it is impractical to perform many defect correction steps in each outer iteration, it is worthwhile to substitute  $k_* = \max\{0, k\}$  for  $k$  in formulae (48)–(50) so as to prevent taking the square root of a negative number. Upon convergence, this safeguard will not make any difference, since  $k$  will be nonnegative from the outset. The above representation of  $\nu_T$  and  $\gamma$  makes it possible to preclude division by zero and obtain bounded coefficients without making any *ad hoc* assumptions and affecting the actual values of  $k$  and  $\varepsilon$ .

### 8.3 Initial conditions

Another important issue which is seldom addressed in the CFD literature is the initialization of data for the  $k - \varepsilon$  model. As a rule, it is rather difficult to devise a reasonable initial guess for a steady-state simulation or proper initial conditions for a dynamic one. The laminar Navier-Stokes equations (2) remain valid until the flow gains enough momentum for the turbulent effects to become pronounced. Therefore, the  $k - \varepsilon$  model should be activated at a certain time  $t_* > 0$  after the startup.

During the ‘laminar’ initial phase ( $t \leq t_*$ ), a constant effective viscosity  $\nu_0$  is prescribed. The values to be assigned to  $k$  and  $\varepsilon$  at  $t = t_*$  are uniquely defined by the choice of  $\nu_0$  and of the default mixing length  $l_0 \in [l_{\min}, l_{\max}]$  where  $l_{\min}$  corresponds to the size of the smallest admissible eddies:

$$k_0 = \left(\frac{\nu_0}{l_0}\right)^2, \quad \varepsilon_0 = C_\mu \frac{k_0^{3/2}}{l_0} \quad \text{at } t \leq t_*. \quad (51)$$

This strategy was adopted because the effective viscosity  $\nu_0$  and the mixing length  $l_0$  are somewhat easier to estimate (at least for a CFD practitioner) than  $k_0$  and  $\varepsilon_0$ . In any case, long-term simulation results are typically not very sensitive to the choice of initial data.

### 8.4 Boundary conditions

At the inlet  $\Gamma_{\text{in}}$ , all velocity components and the values of  $k$  and  $\varepsilon$  are given:

$$\mathbf{u} = \mathbf{g}, \quad k = c_{bc} |\mathbf{u}|^2, \quad \varepsilon = C_\mu \frac{k^{3/2}}{l_0} \quad \text{on } \Gamma_{\text{in}}, \quad (52)$$

where  $c_{bc} \in [0.001, 0.01]$  is an empirical constant [5] and  $|\mathbf{u}| = \sqrt{\mathbf{u} \cdot \mathbf{u}}$  is the Euclidean norm of the velocity. At the outlet  $\Gamma_{\text{out}}$ , the normal gradients of

all scalar variables are required to vanish, and the ‘do-nothing’ [47] boundary conditions are prescribed:

$$\mathbf{n} \cdot \mathcal{S}(\mathbf{u}) = \mathbf{0}, \quad \mathbf{n} \cdot \nabla k = 0, \quad \mathbf{n} \cdot \nabla \varepsilon = 0 \quad \text{on } \Gamma_{\text{out}}. \quad (53)$$

Here  $\mathcal{S}(\mathbf{u}) = -(p + \frac{2}{3}k)\mathcal{I} + (\nu + \nu_T)[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]$  denotes the effective stress tensor. The numerical treatment of inflow and outflow boundary conditions does not present any difficulty. In the finite element framework, relations (53) imply that the surface integrals resulting from integration by parts vanish and do not need to be assembled.

At an impervious solid wall  $\Gamma_w$ , the normal component of the velocity must vanish, whereas tangential slip is permitted in turbulent flow simulations. The implementation of the no-penetration (free slip) boundary condition

$$\mathbf{n} \cdot \mathbf{u} = 0 \quad \text{on } \Gamma_w \quad (54)$$

is nontrivial if the boundary of the computational domain is not aligned with the axes of the Cartesian coordinate system. In this case, condition (54) is imposed on a linear combination of several velocity components whereas their boundary values are unknown. Therefore, standard implementation techniques for Dirichlet boundary conditions based on a modification of the corresponding matrix rows [47] cannot be used.

In order to set the normal velocity component equal to zero, we nullify the off-diagonal entries of the preconditioner  $A(\mathbf{u}^{(m)}) = \{a_{ij}^{(m)}\}$  in the defect correction loop. This enables us to compute the boundary values of  $\mathbf{u}$  explicitly before solving a sequence of linear systems for the velocity components:

$$a_{ij}^{(m)} := 0, \quad \forall j \neq i, \quad \mathbf{u}_i^* := \mathbf{u}_i^{(m)} + \mathbf{r}_i^{(m)}/a_{ii}^{(m)} \quad \text{for } \mathbf{x}_i \in \Gamma_w. \quad (55)$$

Next, we project the predicted values  $\mathbf{u}_i^*$  onto the tangent vector/plane and constrain the corresponding entry of the defect vector  $\mathbf{r}_i^{(m)}$  to be zero

$$\mathbf{u}_i^{(m)} := \mathbf{u}_i^* - (\mathbf{n}_i \cdot \mathbf{u}_i^*)\mathbf{n}_i, \quad \mathbf{r}_i^{(m)} := 0 \quad \text{for } \mathbf{x}_i \in \Gamma_w. \quad (56)$$

After this manipulation, the corrected values  $\mathbf{u}_i^{(m)}$  act as Dirichlet boundary conditions for the solution  $\mathbf{u}_i^{(m+1)}$  at the end of the defect correction step.

As an alternative to the implementation technique of predictor-corrector type, the projection can be applied to the residual vector rather than to the nodal values of the velocity:

$$a_{ij}^{(m)} := 0, \quad \forall j \neq i, \quad \mathbf{r}_i^{(m)} := \mathbf{r}_i^{(m)} - (\mathbf{n}_i \cdot \mathbf{r}_i^{(m)})\mathbf{n}_i \quad \text{for } \mathbf{x}_i \in \Gamma_w. \quad (57)$$

For Cartesian geometries, the algebraic manipulations to be performed affect just the normal velocity component. Note that virtually no extra programming effort is required, which is a significant advantage as compared to another feasible implementation based on local coordinate transformations during the element-by-element matrix assembly [9].

### 8.5 Wall functions

To complete the problem statement, we still need to prescribe the tangential stress as well as the boundary values of  $k$  and  $\varepsilon$  on  $\Gamma_w$ . Note that the equations of the  $k - \varepsilon$  model are invalid in the vicinity of the wall where the Reynolds number is rather low and viscous effects are dominant. In order to avoid the need for resolution of strong velocity gradients, *wall functions* can be derived using the boundary layer theory and applied at an internal boundary  $\Gamma_\delta$  located at a distance  $\delta$  from the solid wall  $\Gamma_w$  [31],[32],[33].

In essence, a boundary layer of width  $\delta$  is removed from the actual computational domain  $\Omega$  and the equations are solved in the reduced domain  $\Omega_\delta$  subject to the following empirical boundary conditions:

$$\mathbf{n} \cdot \mathcal{D}(\mathbf{u}) \cdot \mathbf{t} = -u_\tau^2 \frac{\mathbf{u}}{|\mathbf{u}|}, \quad k = \frac{u_\tau^2}{\sqrt{C_\mu}}, \quad \varepsilon = \frac{u_\tau^3}{\kappa \delta} \quad \text{on } \Gamma_\delta. \quad (58)$$

Here  $\mathcal{D}(\mathbf{u}) = (\nu + \nu_T)[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$  is the viscous part of the stress tensor, the unit vector  $\mathbf{t}$  refers to the tangential direction,  $\kappa = 0.41$  is the von Kármán constant and  $u_\tau$  is the *friction velocity* which is assumed to satisfy

$$g(u_\tau) = |\mathbf{u}| - u_\tau \left( \frac{1}{\kappa} \log y^+ + 5.5 \right) = 0 \quad (59)$$

in the *logarithmic layer*, where the local Reynolds number  $y^+ = \frac{u_\tau \delta}{\nu}$  is in the range  $20 \leq y^+ \leq 100$ , and be a linear function of  $y^+$  in the *viscous sublayer*, where  $y^+ < 20$ . Note that  $\mathbf{u}$  represents the tangential velocity as long as the no-penetration condition (54) is imposed on  $\Gamma_\delta$ .

Equation (59) can be solved iteratively, e.g., by Newton's method [31]:

$$u_\tau^{l+1} = u_\tau^l - \frac{g(u_\tau^l)}{g'(u_\tau^l)} = u_\tau^l + \frac{|\mathbf{u}| - u_\tau^l f(u_\tau^l)}{1/\kappa + f(u_\tau^l)}, \quad l = 0, 1, 2, \dots \quad (60)$$

where the auxiliary function  $f$  is given by

$$f(u_\tau) = \frac{1}{\kappa} \log y_*^+ + 5.5, \quad y_*^+ = \max \left\{ 20, \frac{u_\tau \delta}{\nu} \right\}.$$

The friction velocity is initialized by the approximation

$$u_\tau^0 = \sqrt{\frac{\nu |\mathbf{u}|}{\delta}}$$

and no iterations are performed if it turns out that  $y^+ = \frac{u_\tau^0 \delta}{\nu} < 20$ . In other words,  $u_\tau = u_\tau^0$  in the viscous sublayer. Moreover, we use  $y_*^+ = \max\{20, y^+\}$  in the Newton iteration to guarantee that the approximate solution belongs to the logarithmic layer and remains bounded for  $y^+ \rightarrow 0$ .

The friction velocity  $u_\tau$  is plugged into (58) to compute the tangential stress, which yields a natural boundary condition for the velocity. Integration by parts in the weak form of the Navier-Stokes equations gives rise to a surface integral over the internal boundary  $\Gamma_\delta$  which contains the prescribed traction:

$$\int_{\Gamma_\delta} [\mathbf{n} \cdot \mathcal{D}(\mathbf{u}) \cdot \mathbf{t}] \cdot \mathbf{v} \, ds = - \int_{\Gamma_\delta} u_\tau^2 \frac{\mathbf{u}}{|\mathbf{u}|} \cdot \mathbf{v} \, ds. \quad (61)$$

The free slip condition (54) overrides the normal stress, and Dirichlet boundary conditions for  $k$  and  $\varepsilon$  are imposed in the strong sense. For further details regarding the implementation of wall laws we refer to [31],[32],[33].

### 8.6 Underrelaxation for outer iterations

Due to the intricate coupling of the governing equations, it is sometimes worthwhile to use a suitable underrelaxation technique in order to prevent the growth of numerical instabilities and secure the convergence of outer iterations. This task can be accomplished by limiting the computed solution increments before applying them to the last iterate:

$$u^{(m+1)} := u^{(m)} + \omega^{(m)}(u^{(m+1)} - u^{(m)}), \quad \text{where } 0 \leq \omega^{(m)} \leq 1. \quad (62)$$

The damping factor  $\omega^{(m)}$  may be chosen adaptively so as to accelerate convergence and minimize the error in a certain norm [47]. However, fixed values (for example,  $\omega = 0.8$ ) usually suffice for practical purposes. The sort of underrelaxation can be used in all loops (indexed by  $k$ ,  $l$  and  $m$ ) and applied to selected dependent variables like  $k$ ,  $\varepsilon$  or  $\nu_T$ .

In addition, an *implicit underrelaxation* can be performed in  $m$ -loops by increasing the diagonal dominance of the preconditioner [10],[34]

$$a_{ii}^{(m)} := a_{ii}^{(m)} / \alpha^{(m)}, \quad \text{where } 0 \leq \alpha^{(m)} \leq 1. \quad (63)$$

Of course, the scaling of the diagonal entries does not affect the converged solution. This strategy proves more robust than an *explicit underrelaxation* of the form (62). On the other hand, no underrelaxation whatsoever is needed for moderate time steps which are typically used in dynamic simulations.

## 9 Adaptive Time Step Control

A remark is in order regarding the time step selection for implicit schemes. Unlike their explicit counterparts, they are unconditionally stable so that the time step is limited only by accuracy considerations (for nonstationary problems). Thus, it should be chosen adaptively so as to obtain a sufficiently good approximation at the least possible cost. Many adaptive time stepping techniques have been proposed in the literature. Most of them were originally developed in the ODE context and are based on an estimate of the local truncation error which provides a usable indicator for the step size control.

The ‘optimal’ value of  $\Delta t$  should guarantee that the deviation of a user-defined functional  $J$  (pointwise solution values or certain integral quantities like lift and drag) from its exact value does not exceed a given tolerance

$$|J(u) - J(u_{\Delta t})| \approx TOL. \quad (64)$$

Assuming that the error at the time level  $t^n$  is equal to zero, a heuristic error indicator can be derived from asymptotic expansions for the numerical values of  $J$  computed using two different time steps. For instance, consider

$$\begin{aligned} J(u_{\Delta t}) &= J(u) + \Delta t^2 e(u) + \mathcal{O}(\Delta t)^4, \\ J(u_{m\Delta t}) &= J(u) + m^2 \Delta t^2 e(u) + \mathcal{O}(\Delta t)^4, \end{aligned}$$

where  $m > 1$  is an integer number ( $m = 2, 3$ ). The error term  $e(v)$  is supposed to be independent of the time step and can be estimated as follows

$$e(u) \approx \frac{J(u_{m\Delta t}) - J(u_{\Delta t})}{(m^2 - 1)\Delta t^2}.$$

For the relative error to approach the prescribed tolerance  $TOL$  as required by (64), the new time step  $\Delta t_*$  should be chosen so that

$$|J(u) - J(u_{\Delta t_*})| \approx \left(\frac{\Delta t_*}{\Delta t}\right)^2 \frac{|J(u_{\Delta t}) - J(u_{m\Delta t})|}{m^2 - 1} = TOL.$$

The required adjustment of the time step is given by the formula

$$\Delta t_*^2 = TOL \frac{(m^2 - 1)\Delta t^2}{|J(u_{\Delta t}) - J(u_{m\Delta t})|}.$$

Furthermore, the solution accuracy can be enhanced by resorting to Richardson’s extrapolation (see any textbook on numerical methods for ODEs).

The above considerations may lack some mathematical rigor but nevertheless lead to a very good algorithm for automatic time step control [47]

1. Make one large time step of size  $m\Delta t$  to compute  $u_{m\Delta t}$ .
2. Make  $m$  small substeps of size  $\Delta t$  to compute  $u_{\Delta t}$ .
3. Evaluate the relative changes, i. e.,  $|J(u_{\Delta t}) - J(u_{m\Delta t})|$ .
4. Calculate the ‘optimal’ value  $\Delta t_*$  for the next time step.
5. If  $\Delta t_* \ll \Delta t$ , reject the solution and go back to step 1.
6. Assign  $u := u_{\Delta t}$  or perform Richardson’s extrapolation.

Note that the computational cost per time step increases significantly since the solution  $u_{m\Delta t}$  may be as expensive to obtain as  $u_{\Delta t}$  (due to slow convergence). On the other hand, adaptive time stepping contributes to the robustness of the code and improves its overall efficiency as well as the credibility of simulation results. Further algorithmic details for this approach can be found in [47].

Another simple strategy for adaptive time step control was introduced by Valli et al. [53],[54]. Their *PID controller* is based on the relative changes of a suitable indicator variable (temperature distribution, concentration fields, kinetic energy, eddy viscosity etc.) and can be summarized as follows

1. Compute the relative changes of the chosen indicator variable  $u$

$$e_n = \frac{\|u^{n+1} - u^n\|}{\|u^{n+1}\|}.$$

2. If they are too large ( $e_n > \delta$ ), reject  $u^{n+1}$  and recompute it using

$$\Delta t_* = \frac{\delta}{e_n} \Delta t_n.$$

3. Adjust the time step **smoothly** so as to approach the prescribed tolerance  $TOL$  for the relative changes

$$\Delta t_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{k_P} \left(\frac{TOL}{e_n}\right)^{k_I} \left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{k_D} \Delta t_n.$$

4. Limit the growth and reduction of the time step so that

$$\Delta t_{\min} \leq \Delta t_{n+1} \leq \Delta t_{\max}, \quad m \leq \frac{\Delta t_{n+1}}{\Delta t_n} \leq M.$$

The default values of the PID parameters as proposed by Valli et al. [54] are  $k_P = 0.075$ ,  $k_I = 0.175$  and  $k_D = 0.01$ . Unlike in the case of adaptive time-stepping techniques based on the local truncation error, there is no need for computing an extra solution with a different time step. Therefore, the cost of the feedback mechanism is negligible. Our own numerical studies [28] confirm that this heuristic control strategy is very robust and efficient.

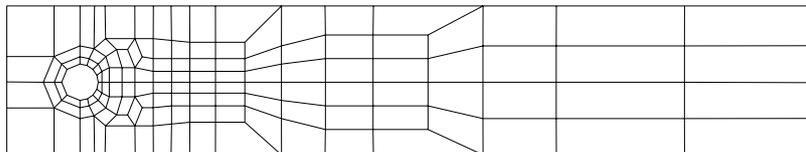
## 10 Numerical Examples

**Flow around a cylinder.** The first incompressible flow problem to be dealt with is the well-known benchmark *Flow around a cylinder* developed in 1995 for the priority research program “Flow simulation on high-performance computers” under the auspices of DFG, the German Research Association [40]. This project was intended to facilitate the evaluation of various numerical algorithms for the incompressible Navier-Stokes equations in the **laminar** flow regime. A quantitative comparison of simulation results is possible on the basis of relevant flow characteristics such as drag and lift coefficients, for which sufficiently accurate reference values are available. Moreover, the efficiency of different solution techniques can be assessed in an objective manner.

Consider the steady incompressible flow around a cylinder with circular cross-section. An in-depth description of the geometrical details and boundary conditions for the 2D/3D case can be found in references [40],[47] which contain all relevant information regarding this benchmark configuration. The flow at  $Re = 20$  is actually dominated by diffusion and could be simulated by the standard Galerkin method without any extra stabilization (as far as the discretization is concerned; the iterative solver may require using a stabilized preconditioner). Ironically, it was this ‘trivial’ steady-state problem that has led us to devise the multidimensional flux limiter of TVD type [24]. Both FCT schemes and slope limiter methods based on stencil reconstruction failed to converge, so the need for a different limiting strategy was apparent.

Furthermore, it is instructive to study the interplay of finite element discretizations for the convective and diffusive terms. As a matter of fact, discrete upwinding can be performed for the cumulative transport operator or just for the convective part. In the case of the nonconforming  $\tilde{Q}_1$ -elements, the discrete Laplacian operator originating from the Galerkin approximation of viscous terms is a positive-definite matrix but some of its off-diagonal coefficients are negative. Our numerical experiments indicate that it is worthwhile to leave it unchanged and start with a FEM-TVD discretization of the convective term. Physical diffusion can probably be taken into account **after** algebraic flux correction but the sums of upstream and downstream edge contributions  $Q^\pm$  and  $P^\pm$  for the node-oriented TVD limiter should be evaluated using the coefficients of the (antisymmetric) convective operator.

To generate hierarchical data structures for the MPSC algorithms implemented in the software package FEATFLOW [48], we introduce a sequence of successively refined quadrilateral meshes. The elements of the coarse mesh shown in Fig. 3 are subdivided into four subelements at each refinement level, and the 2D mesh is extended into the third dimension for a 3D simulation. The two-dimensional results produced by a global MPSC (discrete projection) method with a FEM-TVD discretization of the convective terms are presented in Table 1. The computational mesh for multigrid level NLEV contains NMT midpoints and NEL elements. For the employed  $\tilde{Q}_1/Q_0$  finite element pair, NMT represents the number of unknowns for each velocity component, while NEL is the number of degrees of freedom for the pressure. It can be seen that the drag and lift coefficients approach the reference values  $C_D \approx 5.5795$  and  $C_L \approx 0.01061$  as the mesh is refined. The same outcome can be obtained in the local MPSC framework without resorting to pseudo-time-stepping.



**Fig. 3.** Coarse mesh (2D) for the DFG benchmark ‘Flow around a cylinder’.

Originally, stabilization of convective terms in the FEATFLOW package was performed using streamline diffusion or Samarski’s upwind scheme, whereby the amount of artificial viscosity depends on the local Reynolds number and on the value of the user-defined parameter UPSAM as explained in [47],[48]. Table 2 illustrates that drag and lift for UPSAM=0.1 and UPSAM=1.0 differ appreciably, especially on coarse meshes. In the former case, both quantities tend to be underestimated, while the latter parameter setting results in an unstable convergence behavior. Note that  $C_L$  is negative (!) for NLEV=3.

Since the optimal value of the free parameter is highly problem-dependent, it is impossible to find from *a priori* considerations. In addition, Samarski’s hybrid method is only suitable for intermediate and low Reynolds numbers, as it becomes increasingly diffusive and degenerates into the first-order upwind scheme in the limit of inviscid flow. By contrast, the accuracy of our FEM-TVD discretization does not degrade as  $Re \rightarrow \infty$ . However, it does depend on the choice of the flux limiter (MC was employed in the above example) so the method is – arguably – not completely “parameter-free”. Moreover, the results are influenced by the type of  $\tilde{Q}_1$  basis functions (parametric or non-parametric, with midpoint- or mean-value based degrees of freedom) as well as by the approximations involved in the evaluation of  $C_D$  and  $C_L$ .

NLEV	NMT	NEL	$C_D$	$C_L$
3	4264	2080	5.5504	$0.8708 \cdot 10^{-2}$
4	16848	8320	5.5346	$0.9939 \cdot 10^{-2}$
5	66976	33280	5.5484	$0.1043 \cdot 10^{-1}$
6	267072	133120	5.5616	$0.1056 \cdot 10^{-1}$
7	1066624	532480	5.5707	$0.1054 \cdot 10^{-1}$
8	4263168	2129920	5.5793	$0.1063 \cdot 10^{-1}$

**Table 1.** Global MPSC method / TVD (MC limiter).

NLEV	UPSAM=0.1		UPSAM=1.0	
	$C_D$	$C_L$	$C_D$	$C_L$
3	5.4860	$0.5302 \cdot 10^{-2}$	5.9222	$-0.3475 \cdot 10^{-2}$
4	5.5076	$0.9548 \cdot 10^{-2}$	5.6525	$0.6584 \cdot 10^{-2}$
5	5.5386	$0.1025 \cdot 10^{-1}$	5.5736	$0.1007 \cdot 10^{-1}$
6	5.5581	$0.1044 \cdot 10^{-1}$	5.5658	$0.1048 \cdot 10^{-1}$
7	5.5692	$0.1047 \cdot 10^{-1}$	5.5718	$0.1042 \cdot 10^{-1}$

**Table 2.** Global MPSC method / Samarski’s upwind.

In Table 3, we present the drag and lift coefficients for a three-dimensional simulation of the flow around the cylinder. The hexahedral mesh for NLEV=4 consists of 49,152 elements, which corresponds to 151,808 unknowns for each velocity component. In order to evaluate the performance of the global MPSC

solver and verify grid convergence, we compare the results to those obtained on a coarser and a finer mesh. All numerical solutions were marched to the steady state by the fully implicit backward Euler method. The discretization of convective terms was performed using (i) finite volume upwinding (UPW), (ii) Samarski’s hybrid scheme (SAM), (iii) streamline diffusion stabilization (SD), and (iv) algebraic flux correction (TVD). This numerical study confirms that standard artificial viscosity methods are rather sensitive to the values of the empirical constants, whereas FEM-TVD performs remarkably well. The reference values  $C_D \approx 6.1853$  and  $C_L \approx 0.95 \cdot 10^{-2}$  for this 3D configuration were calculated in [20] by an isoparametric high-order FEM.

NLEV	UPW-1st	SAM-1.0	SD-0.25	SD-0.5	TVD/MC
3	6.08/ 1.01	5.72/ 0.28	5.78/-0.44	5.98/-0.52	5.80/ 0.36
4	6.32/ 1.20	6.07/ 0.62	6.13/ 0.26	6.26/ 0.18	6.14/ 0.46
5	6.30/ 1.20	6.14/ 0.83	6.17/ 0.70	6.23/ 0.64	6.18/ 0.80

**Table 3.** Global MPSC method: 3D simulation,  $C_D/(C_L \cdot 100)$ .

**Backward facing step.** Let us proceed to a three-dimensional test problem which deals with a turbulent flow over a backward facing step at  $Re = 44,000$ , see [31] for details. Our objective is to validate the implementation of the  $k - \varepsilon$  model as described above. As before, the incompressible Navier-Stokes equations are discretized in space using the BB-stable nonconforming  $\tilde{Q}_1/Q_0$  finite element pair, while conforming  $Q_1$  (trilinear) elements are employed for the turbulent kinetic energy and its dissipation rate. All convective terms are handled by the fully implicit FEM-TVD method. The velocity-pressure coupling is enforced in the framework of a global MPSC formulation.

Standard wall laws are applied on the boundary except for the inlet and outlet. The stationary distribution of  $k$  and  $\varepsilon$  in the middle cross-section ( $z = 0.5$ ) is displayed in Fig. 4. The variation of the friction coefficient

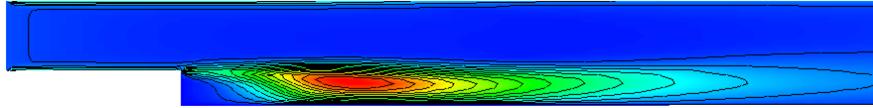
$$c_f = \frac{2\tau_w}{\rho_\infty u_\infty^2} = \frac{2u_\tau^2}{u_\infty^2} = \frac{2k}{u_\infty^2} \sqrt{C_\mu}$$

along the bottom wall is presented in Fig. 5 (left). The main recirculation length  $L \approx 6.8$  is in a good agreement with the numerical results reported in the literature [31]. Moreover, the horizontal velocity component (see Fig. 5, right) assumes positive values at the bottom of the step, which means that the weak secondary vortex is captured as well. The parameter settings for this three-dimensional simulation were as follows

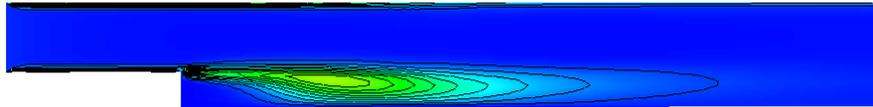
$$\delta = 0.05, \quad c_{bc} = 0.0025, \quad \nu_0 = 10^{-3}, \quad l_0 = 0.02, \quad l_{\max} = 1.0.$$

The computational mesh shown in Fig. 6 contains 57,344 hexahedral cells (pressure unknowns), which corresponds to 178,560 faces (degrees of freedom for each velocity component) and 64,073 vertices (nodes for  $k$  and  $\varepsilon$ ).

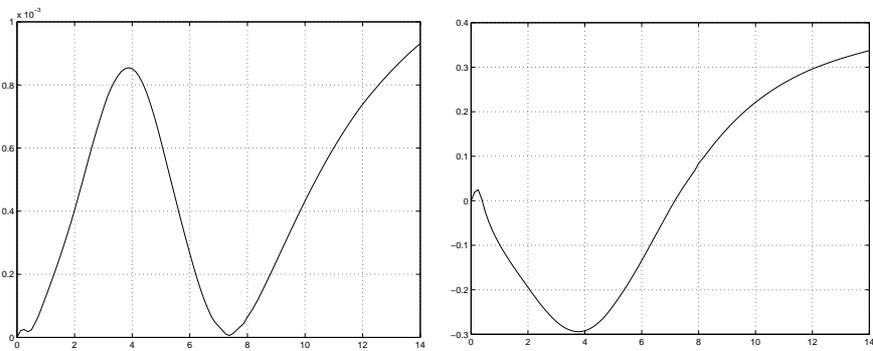
Distribution of  $k$  in the cutplane  $z = 0.5$



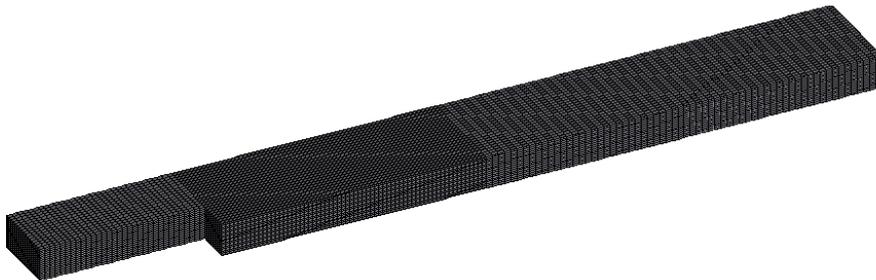
Distribution of  $\varepsilon$  in the cutplane  $z = 0.5$



**Fig. 4.** Backward facing step: stationary FEM-TVD solution,  $Re = 44,000$ .



**Fig. 5.** Distribution of  $c_f$  (left) and  $u_x$  (right) along the bottom wall.



**Fig. 6.** Hexahedral computational mesh for the 3D simulation.

## 11 Application to More Complex Flow Models

In this section, we discuss several incompressible flow models which call for the use of algebraic flux correction (FCT or TVD techniques). Specifically, let us consider generalized Navier-Stokes problems of the form

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \mathbf{f} + \mu \Delta \mathbf{u} - \nabla p \quad , \quad \nabla \cdot \mathbf{u} = 0 \quad (65)$$

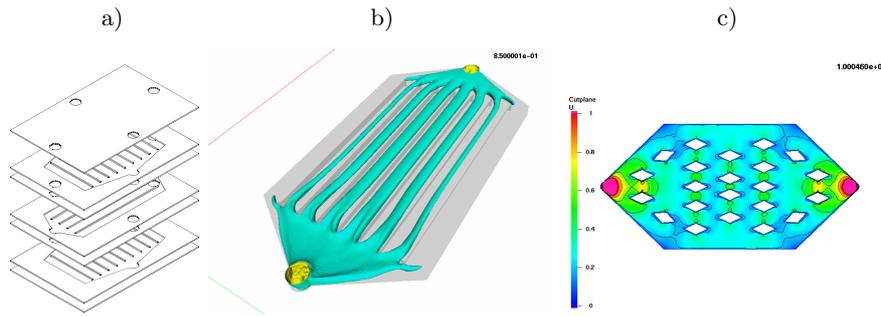
complemented by additional PDEs which describe physical processes like

1. Heat transfer in complex geometries  
→ *Ceramic plate heat exchanger*
2. Multiphase flow with chemical reaction  
→ *Gas-liquid reactors*
3. Nonlinear fluids/granular flow  
→ *Sand motion in silos*
4. Free and moving boundaries  
→ *Level set FEM methods*

Although these typical flow configurations (to be presented below) differ in their complexity and cover a wide range of Reynolds numbers, all of them require an accurate treatment of convective transport. Numerical artifacts such as small-scale oscillations/ripples may cause an abnormal termination of the simulation run due to division by zero, floating point overflow etc. However, in the worst case they can be misinterpreted as physical phenomena and eventually result in making wrong decisions for the design of industrial equipment. Therefore, nonphysical solution behavior should be avoided at any cost, and the use of FEM-FCT/TVD or similar high-resolution schemes is recommendable. Moreover, simulation results must be validated by comparison with experimental data and/or numerical solutions computed on a finer mesh.

### 11.1 Heat Transfer in ‘Plate Heat Exchangers’

The first example deals with the development of optimization tools for a constellation described by ‘coupled stacks’ with different layers (see Fig. 7). This example is quite typical for a complex flow model in the laminar regime. On the one hand, the Reynolds number is rather small due to slow fluid velocities and very small diameters. On the other hand, an unstructured grid FEM approach is required to resolve the small-scale geometrical details. Moreover, the problem at hand is coupled with additional tracer equations of convection-diffusion type. As a rule, the involved diffusion coefficients are small or equal to zero, so that the problem is transport-dominated. Hence, the key elements of an accurate and efficient solution strategy are an appropriate treatment of convective terms on unstructured meshes as well as an unconditionally stable implicit time discretization (the underlying flow field is quasi-stationary). These critical aspects will be exemplarily illustrated in what follows.



**Fig. 7.** Plate heat exchanger: a) geometric configuration; b) typical flow pattern; c) velocity field for an ‘optimal’ distribution of internal objects.

The aim of the underlying numerical study in this section is the understanding and improvement of the

- internal flow characteristics
- heat transfer characteristics

in the shown configuration which can be described by a Boussinesq model. Restricted to one stack only, the internal geometry between ‘inflow’ and ‘outflow’ holes has to be analyzed and channel-like structures or many internal ‘objects’ have to be placed to achieve a homogeneous flow field and, correspondingly, a homogeneous distribution of tracer substances in the interior. In order to determine the optimal shape and distribution of internal obstacles, the simulation software to be developed must be capable of resolving all small-scale details. Therefore, the underlying numerical algorithm must be highly accurate and, moreover, sufficiently flexible and robust. Algebraic FCT/TVD schemes belong to the few discretization techniques that do meet these requirements. Some preliminary results based on the incompressible Navier-Stokes equations for velocity and pressure only are presented in Fig. 7.

The following step beyond ‘manual optimization’ shall be a fully automatic optimization of shape, number, and distribution of the internal objects. Furthermore, the temperature equations are to be solved for each stack as well as for the whole system taking into account heat transfer both in the flow field and between the walls. In addition, chemical reaction models should be included, which gives rise to another set of coupled convection-reaction-diffusion equations. Last but not least, algebraic flux correction is to be employed at the postprocessing step, whereby the ‘residence time distribution’ inside each stack is measured by solving a pure transport equation for passive tracers. Since the flow field is almost stationary and allows large time steps due to the large viscosity parameters, the nonlinear transport equation has to be treated in an implicit way which is a quite typical requirement for the accurate and efficient treatment for such type of flow problems.

## 11.2 Bubbly Flow in Gas-Liquid Reactors

Bubble columns and airlift loop reactors are widely used in industry as contacting devices which enable gaseous and liquid species to engage in chemical reactions. The liquid is supplied continuously or in a batch mode and agitated by bubbles fed at the bottom of the reactor. As the bubbles rise, the gaseous component is gradually absorbed into the bulk liquid where it may react with other species. The geometric simplicity of bubble columns makes them rather easy to build, operate and maintain. At the same time, the prevailing flow patterns are very complex and unpredictable, which represents a major bottleneck for the design of industrial units. By insertion of internal parts, bubble columns can be transformed into airlift loop reactors which exhibit a stable circulation pattern with pronounced *riser* and *downcomer* zones (see Fig. 8). Hence, shape optimization appears to be a promising way to improve the reactor performance by adjusting the geometry of the internals.

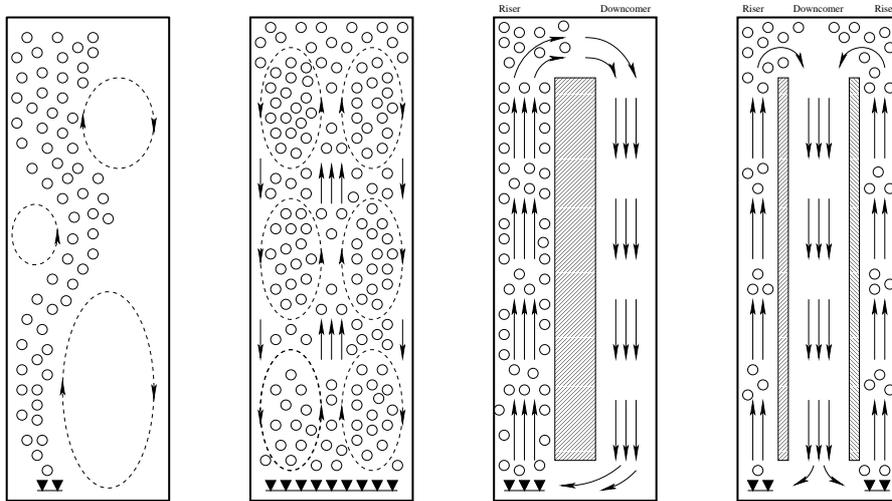


Fig. 8. Bubble columns (left) and airlift loop reactors (right).

In the present chapter, we adopt a simplified two-fluid model which is based on an analog of the Boussinesq approximation (29) for natural convection problems. At moderate gas holdups, the gas-liquid mixture behaves as a weakly compressible fluid which is driven by the bubble-induced buoyancy. Following Sokolichin *et al.* [44],[45] we assume the velocity  $\mathbf{u}_L$  of the liquid phase to be divergence-free. The dependence of the effective density  $\tilde{\rho}_L$  on the local gas holdup  $\epsilon$  is taken into account only in the gravity force, which is a common practice for single-phase flows induced by temperature gradients.

This leads to the following generalization of the Navier-Stokes equations

$$\begin{aligned} \frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L &= -\nabla p_* + \nabla \cdot (\nu_T [\nabla \mathbf{u}_L + \nabla \mathbf{u}_L^T]) - \epsilon \mathbf{g}, \\ \nabla \cdot \mathbf{u}_L &= 0, \quad p_* = \frac{p - p_{\text{atm}}}{\rho_L} + \mathbf{g} \cdot \mathbf{e}_g - gh, \end{aligned} \quad (66)$$

where the eddy viscosity  $\nu_T = C_\mu k^2 / \epsilon$  is a function of the turbulent kinetic energy  $k$  and its dissipation rate  $\epsilon$  (see above). Recall that the evolution of these quantities is described by two scalar transport equations

$$\frac{\partial k}{\partial t} + \nabla \cdot \left( k \mathbf{u}_L - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k + S_k - \epsilon, \quad (67)$$

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot \left( \epsilon \mathbf{u}_L - \frac{\nu_T}{\sigma_\epsilon} \nabla \epsilon \right) = \frac{\epsilon}{k} (C_1 P_k + C_\epsilon S_k - C_2 \epsilon), \quad (68)$$

where the extra source terms are due to the bubble-induced turbulence

$$P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2, \quad (69)$$

$$S_k = -C_k \epsilon \nabla p \cdot \mathbf{u}_{\text{slip}}. \quad (70)$$

The involved slip velocity  $\mathbf{u}_{\text{slip}}$  is proportional to the pressure gradient

$$\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_W}$$

and the ‘drag’ coefficient  $C_W \approx 5 \cdot 10^4 \frac{\text{kg}}{\text{m}^3 \text{s}}$  is determined from empirical correlations for the rise velocity of a single bubble in a stagnant liquid [44].

The gas density  $\rho_G$  is related to the common pressure  $p$  by the ideal gas law  $p = \rho_G \frac{R}{\eta} T$ , which enables us to express the local gas holdup  $\epsilon$  and the interfacial area  $a_S$  per unit volume as follows [26],[28]

$$\epsilon = \frac{\tilde{\rho}_G R T}{p \eta}, \quad a_S = (4\pi n)^{1/3} (3\epsilon)^{2/3}.$$

The *effective* density  $\tilde{\rho}_G = \epsilon \rho_G$  and the number density  $n$  (number of bubbles per unit volume) satisfy the following continuity equations

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G \mathbf{u}_G) = -m_{\text{int}}, \quad (71)$$

$$\frac{\partial n}{\partial t} + \nabla \cdot (n \mathbf{u}_G) = 0. \quad (72)$$

The interphase momentum transfer is typically dominated by the drag force and the density of gas is much smaller than that of liquid, so that the inertia and gravity terms in the momentum equation for the gas phase can be

neglected [44],[45]. Under these (quite realistic) simplifying assumptions, the gas phase velocity  $\mathbf{u}_G$  can be computed from the algebraic slip relation

$$\mathbf{u}_G = \mathbf{u}_L + \mathbf{u}_{\text{slip}} + \mathbf{u}_{\text{drift}}, \quad \mathbf{u}_{\text{drift}} = -d_G \frac{\nabla n}{n},$$

where the drift velocity  $\mathbf{u}_{\text{drift}}$  is introduced to model the bubble path dispersion by turbulent eddies. It is usually assumed that  $d_G = \nu_T/\sigma_G$ , where the Schmidt number  $\sigma_G$  equals unity. Substitution into (71)–(72) yields

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G(\mathbf{u}_L + \mathbf{u}_{\text{slip}}) - \nu_T \nabla \tilde{\rho}_G) = -m_{\text{int}}, \quad (73)$$

$$\frac{\partial n}{\partial t} + \nabla \cdot (n(\mathbf{u}_L + \mathbf{u}_{\text{slip}}) - \nu_T \nabla n) = 0. \quad (74)$$

Note that the contribution of  $\mathbf{u}_{\text{drift}}$  gives rise to diffusive terms in both equations and it is implied that  $\tilde{\rho}_G \nabla n/n = \nabla \tilde{\rho}_G$ . Strictly speaking, this relation is valid only for an (almost) constant bubble mass  $m = \tilde{\rho}_G/n$  but can also be used in the framework of ‘operator splitting’, whereby convection-diffusion and reaction-absorption processes are decoupled from one another.

The sink term  $m_{\text{int}}$  in equations (71) and (73) is due to the reaction-enhanced mass transfer. It is proportional to the interfacial area  $a_S$  and can be modeled in accordance with the standard two-film theory. The effective concentrations of all species in the liquid phase are described by extra convection-reaction-diffusion equations [23],[25],[26],[28]. If the coalescence and breakup of bubbles cannot be neglected, equation (72) should be replaced by a detailed *population balance model* for the bubble size distribution [11]. In any case, we end up with a very large system of convection-dominated PDEs which are strongly coupled and extremely sensitive to nonphysical phenomena that may result from an improper discretization of the convective terms.

In order to implement the above *drift-flux model* in a finite element code, we need to collect all the numerical tools presented so far

- algebraic flux correction schemes for the convective terms
- MPSC solvers for the incompressible Navier-Stokes equations
- block-iterative coupling mechanisms (Boussinesq approximation)
- implementation techniques for the  $k - \varepsilon$  turbulence model
- adaptive time-stepping (PID control of the local gas holdup)

The segregated algorithm proposed in [28] consists of nested loops for the intimately coupled subproblems which are solved sequentially using solution values from the previous outer iteration to evaluate the coefficients and source/sink terms. In each time step, the outermost loop is responsible for the coupling of all relevant equation blocks and contains another outer iteration loop for the equations of the  $k - \varepsilon$  turbulence model which are closely related to one another and must be solved in a coupled fashion. The buoyancy force in the Navier-Stokes equations is evaluated using the gas holdup from the

previous outer iteration and fixed-point defect correction is employed for all nonlinear convective terms, which gives rise to another sequence of outer iterations. The iterative process is repeated until the residual of the momentum equation and/or the relative changes of all variables become small enough.

Operator splitting tools are employed to separate convection-diffusion and absorption-reaction processes at each time step. First, all scalar quantities are transported without taking the sources/sinks into account. The homogeneous equations are decoupled and can be processed in parallel. An implicit time discretization of Crank-Nicolson or backward Euler type is performed for all equations. The value of the implicitness parameter  $\theta$  and of the local time step can be selected individually for each subproblem so as to maximize accuracy and/or stability. The communication between the subproblem blocks takes place at the end of the common macro time step  $\Delta t_n$  which is chosen adaptively so as to control the changes of the gas holdup distribution.

The flow chart of algorithmic steps to be performed is as follows [28]

1. Recover the pressure gradient  $\nabla p$  via  $L_2$ -projection.
2. Compute the associated slip velocity  $\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_w}$ .
3. Solve the homogeneous continuity equation for  $\tilde{\rho}_G$ .
4. Update the number density  $n$  according to (74).
5. Convert  $\tilde{\rho}_G$  and  $n$  into  $\epsilon$  and  $a_S$ ; evaluate  $m_{\text{int}}$ .
6. Solve the transport equations for concentrations.
7. Solve the ODE systems for absorption-reaction.
8. Enter the inner loop for the  $k - \epsilon$  model (67)–(68).
9. Compute the turbulent eddy viscosity  $\nu_T = C_\mu \frac{k^2}{\epsilon}$ .
10. Insert  $\nu_T$  and  $\epsilon$  into (66) and evaluate the residual.
11. If converged, then proceed to the next time step.
12. Solve the Navier-Stokes equations and go to 1.

The first example deals with the locally aerated bubble column that was investigated in detail by Becker *et al.* [1]. The snapshots of the meandering bubble swarm displayed in Fig. 9 are in a good agreement with experimental data. The evolution of the gas holdup in the middle cross section of a prototypical airlift loop reactor is shown in Fig. 10. Aeration takes place at the bottom of the riser section where both phases flow upward. At the upper surface, the bubbles escape, while the liquid is diverted into the gas-free downcomer so as to form a closed loop. The two-phase flow reaches a steady state within a few seconds after the startup (see the right diagram). Computational results for the reaction-enhanced absorption of  $\text{CO}_2$  in a locally aerated bubble column filled with an aqueous solution of NaOH are presented in [25],[28].

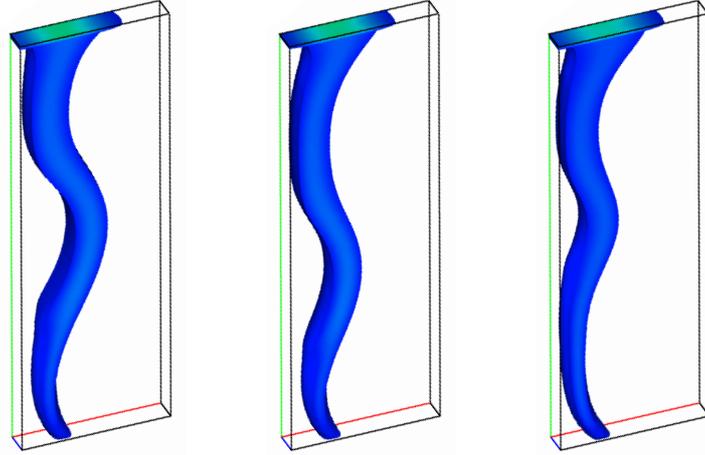


Fig. 9. Gas holdup distribution in a flat bubble column.

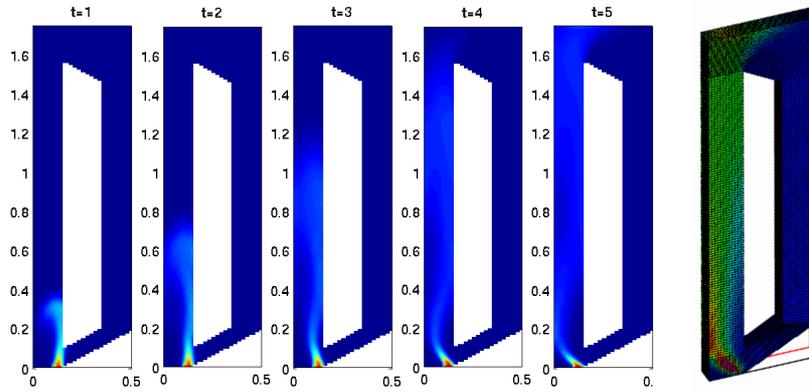


Fig. 10. Gas holdup distribution in an airlift loop reactor.

### 11.3 Nonlinear (Granular) Flow

Another interesting example for the (non-standard) use of TVD techniques for convective operators is the numerical simulation of nonlinear incompressible fluids governed by the Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \mathbf{T} + \nabla p = \mathbf{f} \quad , \quad \nabla \cdot \mathbf{u} = 0 \quad (75)$$

where  $\mathbf{T}$  is the stress tensor. Furthermore, the deformation rate tensor  $\mathbf{D}$  and the spin tensor  $\mathbf{W}$  depend on the velocity gradients as follows

$$\mathbf{D} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad , \quad \mathbf{W} = \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T) \quad (76)$$

In the case of Newtonian fluids, the stress tensor in (75) is given by

$$\mathbf{T} = 2\nu\mathbf{D} = \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T). \quad (77)$$

The most popular representatives of nonlinear fluids are described by *power law* models, whereby the viscosity  $\nu$  in the constitutive relation (77) depends on  $\mathbf{D} = \mathbf{D}(\mathbf{u})$  in a nonlinear way

$$\mathbf{T} = 2\nu(\mathbf{D})\mathbf{D} \quad , \quad \nu(\mathbf{D}) = \nu_0(\epsilon_1 + \epsilon_2|\mathbf{D}|)^\alpha \quad (78)$$

with certain parameters  $\nu_0$ ,  $\epsilon_1$ ,  $\epsilon_2$  and  $\alpha$ . Fluids with  $\alpha > 0$  correspond to *shear thickening*, in contrast to the case  $\alpha < 0$  (*shear thinning*), both of which lead to numerical challenges. Such *differential type* models — see also *Bingham* or *Reiner-Rivlin* fluids [30],[38] — do not require implicit calculations of  $\mathbf{T}$  since the tensor  $\mathbf{T}$  can be represented as a (nonlinear) function of  $\mathbf{D}(\mathbf{u})$ . Only modifications of existing Navier-Stokes solvers have to be performed, without an additional discretization of equations for  $\mathbf{T}$ .

This is in contrast to the more general class of *rate type* models which couple (nonlinear) evaluations and derivatives of  $\mathbf{T}$  with functional evaluations and derivatives of  $\mathbf{u}$  and  $\mathbf{D}$  in space and time. Examples are *Rivlin-Ericksen* and *second grade* fluids [21],[30] and particularly *Oldroyd* models [13],[38]. Defining the *objective time derivative* of the tensor  $\mathbf{T}$  as

$$\frac{\mathcal{D}_a\mathbf{T}}{\mathcal{D}t} = \frac{\partial\mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{T} + \mathbf{T}\mathbf{W} + (\mathbf{T}\mathbf{W})^T - a[\mathbf{T}\mathbf{D} + (\mathbf{T}\mathbf{D})^T], \quad (79)$$

where  $-1 \leq a \leq 1$ , a general description for *Oldroyd* models [18],[37] reads

$$\lambda_1 \frac{\mathcal{D}_a\mathbf{T}}{\mathcal{D}t} + \mathbf{T} + \gamma(\mathbf{T}, \mathbf{D}) = 2\mu \left( \lambda_2 \frac{\mathcal{D}_a\mathbf{D}}{\mathcal{D}t} + \mathbf{D} \right). \quad (80)$$

The *relaxation time*  $\lambda_1$ , *retardation time*  $\lambda_2$  and viscosity parameter  $\mu$  may additionally depend on  $\mathbf{D}$ . In variants with  $\gamma(\mathbf{T}, \mathbf{D}) \neq 0$  (*Oldroyd 8 constants* model, *Larson* model, *Phan-Thien-Tanner* model), there is another nonlinear relationship between  $\mathbf{T}$  and  $\mathbf{D}$ , while in the following we will concentrate on the case  $\gamma(\mathbf{T}, \mathbf{D}) = 0$  including the so-called *Jeffrey* models, resp., the *Maxwell* fluids. Examples of numerical simulations can be found in papers by Joseph [21], Glowinski [13] and Hron [18]. Consider the following equation for  $\mathbf{T}$

$$\mathbf{T} = 2\mu_v\mathbf{D} + \mathbf{S} \quad , \quad \lambda \frac{\mathcal{D}_a\mathbf{S}}{\mathcal{D}t} + \mathbf{S} = 2\mu_e\mathbf{D}, \quad (81)$$

where  $\lambda = \lambda_1$ ,  $\mu_v = \mu \frac{\lambda_2}{\lambda_1}$  and  $\mu_e = \mu - \mu_v$ . Then, we can rewrite the Navier-Stokes model in (75) as

$$\text{Re} \left[ \frac{\partial\mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u} \right] + \nabla p - (1 - \alpha)\Delta\mathbf{u} - \nabla \cdot \mathbf{S} = \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0 \quad (82)$$

$$\text{We} \left[ \frac{\partial\mathbf{S}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{S} + \beta_a(\mathbf{S}, \mathbf{D}) \right] + \mathbf{S} = 2\alpha\mathbf{D}. \quad (83)$$

Given  $-1 \leq a \leq 1$ , let us introduce the following definitions

$$\beta_a(\mathbf{S}, \mathbf{D}) = \mathbf{S}\mathbf{W} + (\mathbf{S}\mathbf{W})^T - a[\mathbf{S}\mathbf{D} + (\mathbf{S}\mathbf{D})^T] \quad (84)$$

$$\alpha = \frac{\mu_e}{\mu_e + \mu_v}, \quad \text{Re} = \frac{L\mathbf{V}\rho}{\mu_e + \mu_v}, \quad \text{We} = \frac{\mathbf{V}\lambda}{L}, \quad (85)$$

where  $L$  stands for a characteristic length and  $\mathbf{V}$  is the velocity.

As a result, we end up with a coupled Navier-Stokes-like system (82) for the variables  $(\mathbf{u}, p)$  with an additional term  $\nabla \cdot \mathbf{S}$ . Furthermore, there is a non-linear nonstationary tensor-valued transport-reaction equation in (83) which involves both  $\mathbf{S}$  and  $\mathbf{D}(\mathbf{u})$ . To solve this highly complex system, appropriate discretization techniques in space and time (implicit Euler, Crank-Nicolson or the fractional step method in conjunction with BB-stable FEM approximations) should be employed. In particular, the algebraic TVD techniques are to be recommended for the tensor-valued transport problems since there is an obvious need for a monotone, oscillation-free and highly accurate discretization of the tensor  $\mathbf{T}$ . Moreover, the above-mentioned MPSC approach to solving discrete saddle point problems and providing a proper coupling between the equations at hand turns out very handy for this kind of applications.

Finally, we mention the hypoplastic model proposed by Kolymbas [22] for the numerical simulation of dry cohesionless granular materials, for instance for the flow of sand in silos. This approach contains components from *rate type* as well as from *differential type* models such that the relation to the previously described *Oldroyd* model becomes evident:

$$\text{Re} \left[ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right] = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0 \quad (86)$$

$$\begin{aligned} \frac{\partial \mathbf{T}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{T} = & -[\mathbf{T}\mathbf{W} - \mathbf{W}\mathbf{T}] \\ & + C_1 \frac{1}{2}(\mathbf{T}\mathbf{D} - \mathbf{D}\mathbf{T}) + C_2 \text{tr}(\mathbf{T}\mathbf{D}) \cdot \mathbf{I} \\ & + C_3 \sqrt{\text{tr}\mathbf{D}^2} \mathbf{T} + C_4 \frac{\sqrt{\text{tr}\mathbf{D}^2}}{\text{tr}\mathbf{T}} \mathbf{T}^2 \\ & + \nu(\mathbf{D}) \left[ \frac{\partial \mathbf{D}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{D} + \mathbf{D}\mathbf{W} - \mathbf{W}\mathbf{D} \right]. \end{aligned} \quad (87)$$

Here,  $\mathbf{I}$  denotes the unit matrix,  $\nu(\mathbf{D})$  a nonlinear (tensorial) function of *power law* type, and  $C_i$  are specific material constants. Again, the robust and accurate treatment of the convective terms in the absence of any second order elliptic diffusive term is a very important aspect for numerical simulations so the use of FEM-TVD techniques appears to be especially promising.

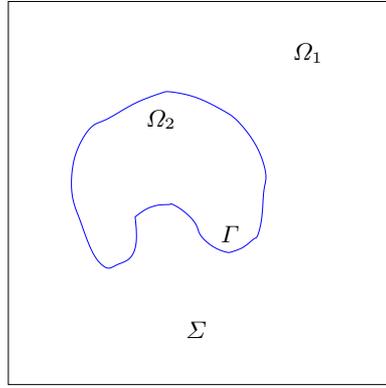
### 11.4 Free Surface/Interface Flows

Our last example is intended to illustrate the need for implicit TVD-type discretizations in the case of (laminar) incompressible flow problems which involve free surfaces, resp., free interfaces. The governing equations read

$$\rho_i \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) - \nabla \cdot (2\mu_i \mathbf{S}) + \nabla p = \rho_i \mathbf{g}, \quad (88)$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega_i, \quad i = 1, 2, \dots \quad (89)$$

where  $\mathbf{S} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$  is the deformation tensor,  $\mathbf{g}$  is the gravity and the computational domain  $\Omega = \Omega_1 \cup \Omega_2$  with boundary  $\Sigma$  is shared by two immiscible fluids separated by the interface  $\Gamma = \bar{\Omega}_1 \cap \bar{\Omega}_2$ . The density and (dynamic) viscosity of the  $i$ -th fluid are denoted by  $\rho_i$  and  $\mu_i$ , respectively.



**Fig. 11.** Free boundary problem.

This PDE system is complemented by the boundary condition

$$\mathbf{v} = \mathbf{b} \quad \text{on } \Sigma, \quad (90)$$

and the following initial values are prescribed at  $t = 0$

$$\mathbf{v}|_{t=0} = \mathbf{v}^0 \quad \text{in } \Omega; \quad \Gamma|_{t=0} = \Gamma^0. \quad (91)$$

Moreover, two extra conditions can be inferred from the conservation of mass and momentum at the free/moving interface. Specifically, we have

1. *Mass balance.* If there is no mass transfer between the two fluids, then the interface  $\Gamma$  moves with the flow in the normal direction at velocity

$$V = \mathbf{v}_1 \cdot \mathbf{n} = \mathbf{v}_2 \cdot \mathbf{n}, \quad (92)$$

where  $\mathbf{n}$  is the normal to  $\Gamma$  and  $\mathbf{v}_i$  denotes the velocity of the  $i$ -th fluid.

2. *Momentum balance.* The interfacial stresses  $\mathbf{T}_i = -p\mathbf{I} + 2\mu_i\mathbf{S}$ ,  $i = 1, 2$  are related by the following jump condition

$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \mathbf{n} = \kappa\sigma\mathbf{n}, \quad (93)$$

where  $\sigma$  is the surface tension coefficient and  $\kappa$  is twice the mean curvature of the interface.

Thus, the following relations should hold on the internal boundary  $\Gamma$

$$[\mathbf{v}]|_{\Gamma} \cdot \mathbf{n} = 0, \quad -[-p\mathbf{I} + 2\mu(\mathbf{x})\mathbf{S}]|_{\Gamma} \cdot \mathbf{n} = \kappa\sigma\mathbf{n}, \quad (94)$$

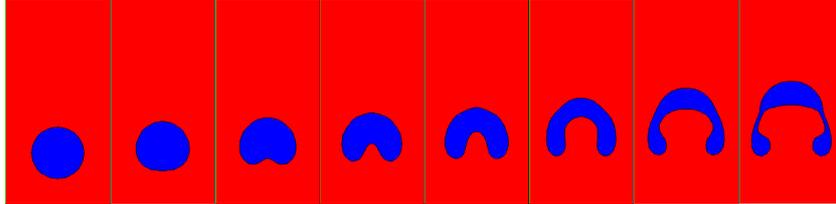
where  $[\cdot]|_{\Gamma} = \lim_{\mathbf{x} \in \Omega_2 \rightarrow \Gamma}(\cdot) - \lim_{\mathbf{x} \in \Omega_1 \rightarrow \Gamma}(\cdot)$  and it is implied that

$$\rho(\mathbf{x}) = \begin{cases} \rho_1, & \forall \mathbf{x} \in \Omega_1, \\ \rho_2, & \forall \mathbf{x} \in \Omega_2, \end{cases}, \quad \mu(\mathbf{x}) = \begin{cases} \mu_1, & \forall \mathbf{x} \in \Omega_1, \\ \mu_2, & \forall \mathbf{x} \in \Omega_2. \end{cases}$$

Unlike classical front tracking methods, the *level-set* approach is based on an (implicit) reconstruction of free interfaces via an ‘indicator function’  $\varphi$  which is equal to zero directly at the interface. In contrast to the well-known *volume of fluid* (VOF) method [17], which uses a discontinuous indicator function, the level set function  $\varphi$  is smooth and amenable to numerical treatment. In either case, the position of the free interface can be determined by solving a pure transport equation for the corresponding indicator function

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0. \quad (95)$$

In what follows, we adopt the level set formulation and opt to discretize all convective terms by a high-resolution finite element scheme of TVD type.



**Fig. 12.** Rising bubble simulation by the level-set method.

It is common practice to define  $\varphi$  as the ‘signed distance’ function which is positive in one fluid and negative in the other. By definition, we have

$$|\nabla \varphi| = 1, \quad \varphi > 0 \quad \text{in } \Omega_1, \quad \varphi < 0 \quad \text{in } \Omega_2.$$

As a rule, the initialization of  $\varphi$  does not present any difficulty but it is not trivial to preserve the ‘signed distance’ property in the course of a numerical

simulation. Therefore, it is advisable to ‘reinitialize’  $\varphi$  once in a while so as to provide an accurate representation of the free interface.

Given the numerical solution  $\varphi_{old}$  of the pure transport problem (95) after a certain number of time steps, we seek a signed distance function  $\varphi$  which has the same zero level set as  $\varphi_{old}$ . It can be recovered as the steady-state solution of another (nonlinear) transport equation which reads

$$\frac{\partial \varphi}{\partial t} = \text{sign}(\varphi_{old})(1 - |\nabla \varphi|). \quad (96)$$

This auxiliary problem for the reinitialization step can be cast in the form

$$\frac{\partial \varphi}{\partial t} + \mathbf{w} \cdot \nabla \varphi = \text{sign}(\varphi_{old}), \quad \mathbf{w} = \text{sign}(\varphi_{old}) \frac{\nabla \varphi}{|\nabla \varphi|} \quad (97)$$

and solved by a suitable pseudo-time-stepping method. In particular, this can be accomplished in the framework of an implicit TVD-like discretization which is guaranteed to produce nonoscillatory and highly accurate results. Numerical studies indicate that a proper treatment of the level set function  $\varphi$  is crucial to the quality of the interface reconstruction so that the convective terms in equations (95) and (97) should be handled with extreme care.

## 12 Conclusions

Even for laminar flow models, there is still a strong need for better mathematical approaches as far as the discretization and solution aspects are concerned. The accuracy, flexibility, robustness and, particularly, overall efficiency of many currently available simulation tools leave a lot to be desired. The next laborious step is to develop professional CFD software for grand-challenge industrial problems. However, benchmark computations and other numerical studies demonstrate that one **must** make every effort to hone and refine the ‘basic tools’ before proceeding to more complex simulations! Otherwise, there might be no chance to succeed in tackling many challenging applications and to achieve not only qualitatively, but also quantitatively accurate results. Hence, the list of topics to be addressed in the near future is as follows:

- **Higher order FEM spaces:** Preliminary investigations performed by our group and others show that polynomial approximations of higher order accuracy might be preferable. Even for the  $Q_2/P_1$  finite element pair (biquadratic velocity, piecewise linear pressure), the efficiency gains can be enormous. However, it is currently not clear how high-order finite elements would behave for nonsmooth solutions which exhibit strong gradients. In particular, an appropriate stabilization of the convective terms is necessary at high Reynolds numbers. Algebraic flux correction of FCT/TVD type seems to be feasible but its extension to the  $Q_2$  elements is not obvious. Another primary goal is the development of hierarchical multigrid solvers

which should ensure that the potential advantages of using higher-order FEM approximations can be realized in practice.

- **Nonlinear solvers:** Since TVD-like discretization techniques are *per se* nonlinear, appropriate Newton-like methods are to be applied if an implicit treatment of the convective terms is adopted, e.g., for the ‘tracer transport’ in a low Reynolds number flow or at the ‘reinitialization step’ for the level set function. However, due to the fully discrete and inherently discontinuous nature of algebraic flux correction schemes, the derivation of the (approximate) Jacobian matrices poses a formidable problem which is still largely unsolved and calls for further research.
- **Multigrid solvers:** In a similar vein, the numerical behavior of standard (geometric) multigrid solvers for the resulting linear subproblems has not yet been investigated and requires further in-depth studies.
- **Tensor-valued transport operators:** As described for nonlinear fluids, additional hyperbolic problems with tensor-valued transport operators may need to be dealt with. An extension of our algebraic high-resolution schemes, which were originally developed for scalar quantities, to such applications should be undertaken in the near future.
- **A posteriori error control:** Since the proposed flux correction techniques are applicable to finite element methods, appropriate a posteriori error control mechanisms for Galerkin-type discretizations can be incorporated. To be more specific, residual-based error estimation via dual problems for user-specified quantities in the spirit of Rannacher and his collaborators is to be examined. Its generalization to an unstructured grid FEM equipped with algebraic flux limiters would make it possible to gain ‘optimal’ control over the local mesh refinement and/or coarsening.

Apart from all these mathematical challenges, the results of recent computational studies (see [49] for a critical discussion) reveal that *memory access, not data processing is costly on modern computers*. This is the major reason why traditional numerical approaches for PDEs and their realization as CFD software have massive problems with achieving a significant percentage of the possible peak performance. On the other hand, the Numerical Linear Algebra tools for the solution of sparse linear systems with millions of unknowns can be significantly improved. This can be accomplished, for instance, by resorting to cache-based implementation techniques and exploiting the local structure of data sets for block-structured grids. Such a hardware-oriented approach may yield an overall speed-up factor of as much as 10 – 1000 even on a single processor. In addition, the use of optimal parallelization strategies may increase the performance of the code by further orders of magnitude.

To realize these ambitious goals, modern numerical methods and programming concepts are to be incorporated into CFD simulation tools. Some of the latest trends are *adaptive meshing* and *a posteriori error control* as well as generalized *multigrid/domain decomposition* solvers. Furthermore, the same

philosophy applies to fully implicit Navier-Stokes solvers of MPSC type which should also be optimized so as to shift the distribution of CPU times away from expensive memory access tasks (assembly of stiffness matrices, defects and residuals, modifications of the mesh) toward more arithmetic-intensive computational work (iterative solution of linear systems by multigrid or Krylov-space methods). It is obvious that the design and implementation of such advanced software concepts is not an easy job. However, the foundations have already been laid in the framework of the ongoing FEAST project (see [50]) aimed at the development of such strategies. Iterative solvers based on the above-mentioned data structures are already available and prove remarkably efficient as compared to conventional implementation techniques.

In fact, current technological trends indicate that the demand for high-performance CFD software will continue to increase. Although the development and optimization of such advanced software products may take a very long time and consume a lot of resources, this investment is certain to yield huge gains in the long run. Indeed, numerical simulation of many ‘real life’ problems (e.g., turbulent and/or multiphase flows in complex 3D geometries) is still prohibitively expensive. In order to exploit the available computing power to the full extent, software engineering aspects must be accompanied by corresponding efforts in the design of optimal numerical algorithms. Moreover, the peculiarities of modern hardware architecture should be taken into account. Otherwise, many ‘nice’ mathematical developments may end up being impractical as long as just a small fraction (less than one percent is not unusual) of the potential peak performance can be achieved in practice.

## References

1. S. Becker, A. Sokolichin and G. Eigenberger, Gas-liquid flow in bubble columns and loop reactors: Part II. *Chem. Eng. Sci.* **49** (1994) 5747–5762.
2. J. P. Boris, F. F. Grinstein, E. S. Oran und R. J. Kolbe, New insights into Large Eddy Simulation, *Fluid Dynamics Research* **10**, Nr. 4-6, 1992, 199-227.
3. A. J. Chorin, Numerical solution of the Navier–Stokes equations. *Math. Comp.* **22** (1968) 745–762.
4. M. A. Christon, P. M. Gresho, S. B. Sutton, Computational predictability of natural convection flows in enclosures. In: K. J. Bathe (ed) Proc. First MIT Conference on *Computational Fluid and Solid Mechanics*, Elsevier, 2001, 1465-1468, 2001.
5. R. Codina and O. Soto, Finite element implementation of two-equation and algebraic stress turbulence models for steady incompressible flows. *Int. J. Numer. Methods Fluids* **30** (1999) no. 3, 309-333.
6. M. Crouzeix and P. A. Raviart, Conforming and non-conforming finite element methods for solving the stationary Stokes equations, *R.A.I.R.O. R-3* (1973) 77–104.
7. J. Donea, S. Giuliani, H. Laval and L. Quartapelle, Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Comput. Meth. Appl. Mech. Engrg.* **30** (1982) 53–73.

8. M. S. Engelman, V. Haroutunian and I. Hasbani, Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems. *Int. J. Numer. Meth. Fluids* **17** (1993) 323–348.
9. M. S. Engelman, R. L. Sani and P. M. Gresho, The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. *Int. J. Numer. Meth. Fluids* **2** (1982) 225–238.
10. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.
11. C. Fleischer, *Detaillierte Modellierung von Gas-Flüssigkeits-Reaktoren*. Fortschr.-Ber. VDI-Reihe 3, Nr. 691, Düsseldorf: VDI Verlag, 2001.
12. V. Girault and P. A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer Verlag, Berlin–Heidelberg, 1986.
13. R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *J. Comput. Phys.* **169** (2001) 363–426.
14. P. M. Gresho, R. L. Sani and M. S. Engelman, *Incompressible Flow and the Finite Element Method: Advection-Diffusion and Isothermal Laminar Flow*. Wiley, 1998.
15. P. M. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix, Part 1: Theory, Part 2: Implementation. *Int. J. Numer. Meth. Fluids* **11** (1990) 587–659.
16. F. F. Grinstein and C. Fureby, *On Monotonically Integrated Large Eddy Simulation of Turbulent Flows Based on FCT Algorithms*. Chapter 3 in this volume.
17. C. W. Hirt, A. A. Amsden and J. L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.* **14** (1974) 227–253.
18. J. Hron, Numerical simulation of visco-elastic fluids. In: *WDS'97*, Freiburg, 1997.
19. T. J. R. Hughes, L. P. Franca and M. Balestra, A new finite element formulation for computational fluid mechanics: V. Circumventing the Babuska–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal order interpolation. *Comp. Meth. Appl. Mech. Engrg.* **59** (1986) 85–99.
20. V. John, Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **40** (2002) 775–798.
21. D. D. Joseph, R. Glowinski, J. Feng, T. W. Pan, A three-dimensional computation of the force and torque on an ellipsoid settling slowly through a viscoelastic fluid. *J. Fluid Mech.* **283** (1995) 1–16.
22. D. Kolymbas, An outline of hypoplasticity. *Archive of Applied Mechanics* **61** (1991) 143–151.
23. D. Kuzmin, Numerical simulation of mass transfer and chemical reactions in gas-liquid flows. In: *Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications*, World Scientific, 2000, 237–244.
24. D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131–158.
25. D. Kuzmin and S. Turek, Efficient numerical techniques for flow simulation in bubble column reactors. In: *Preprints of the 5th German-Japanese Symposium on Bubble Columns*, VDI/GVC, 2000, 99–104.

26. D. Kuzmin and S. Turek, Finite element discretization tools for gas-liquid flows. In: M. Sommerfeld (ed.), *Bubbly Flows: Analysis, Modelling and Calculation*, Springer, 2004, 191-201.
27. D. Kuzmin and S. Turek, Multidimensional FEM-TVD paradigm for convection-dominated flows. Technical report **253**, University of Dortmund, 2004. In: *Proceedings of the IV European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Volume II, ISBN 951-39-1869-6.
28. D. Kuzmin and S. Turek, Numerical simulation of turbulent bubbly flows. Technical report **254**, University of Dortmund, 2004. To appear in: *Proceedings of the 3rd International Symposium on Two-Phase Flow Modelling and Experimentation*, Pisa, September 22-24, 2004.
29. A. J. Lew, G. C. Buscaglia and P. M. Carrica, A note on the numerical treatment of the  $k$ -epsilon turbulence model. *Int. J. Comput. Fluid Dyn.* **14** (2001) no. 3, 201-209.
30. J. Malek, K. R. Rajagopal, M. Ruzicka, Existence and regularity of solutions and the stability of the rest state for fluids with shear dependent viscosity. *Mathematical Models and Methods in Applied Sciences* **5** (1995) 789-812.
31. G. Medić and B. Mohammadi, NSIKE - an incompressible Navier-Stokes solver for unstructured meshes. *INRIA Research Report* **3644** (1999).
32. B. Mohammadi, A stable algorithm for the  $k$ -epsilon model for compressible flows. *INRIA Research Report* **1355** (1990).
33. B. Mohammadi and O. Pironneau, *Analysis of the  $k$ -epsilon turbulence model*. Wiley, 1994.
34. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, 1980.
35. A. Prohl, *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*. Advances in Numerical Mathematics. B. G. Teubner, Stuttgart, 1997.
36. L. Quartapelle, *Numerical Solution of the Incompressible Navier-Stokes Equations*. Birkhäuser Verlag, Basel, 1993.
37. A. Quarteroni, M. Tuveri, A. Veneziani, Computational vascular fluid dynamics. *Comp. Vis. Science* **2** (2000) 163-197.
38. K. R. Rajagopal, Mechanics of non-Newtonian fluids. In G. P. Galdi and J. Necas (eds), *Recent Developments in Theoretical Fluid Mechanics*. Pitman Research Notes in Mathematics **291**, Longman, 1993, 129-162.
39. R. Rannacher and S. Turek, A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs* **8** (1992), no. 2, 97-111.
40. M. Schäfer and S. Turek (with support of F. Durst, E. Krause, R. Rannacher), Benchmark computations of laminar flow around cylinder, In E.H. Hirschel (ed.), *Flow Simulation with High-Performance Computers II*, Vol. 52 von *Notes on Numerical Fluid Mechanics*, Vieweg, 1996, 547-566.
41. R. Schmachtel, *Robuste lineare und nichtlineare Lösungsverfahren für die inkompressiblen Navier-Stokes-Gleichungen*. PhD thesis, University of Dortmund, 2003.
42. P. Schreiber, *A new finite element solver for the nonstationary incompressible Navier-Stokes equations in three dimensions*, PhD Thesis, University of Heidelberg, 1996.
43. P. Schreiber and S. Turek, An efficient finite element solver for the nonstationary incompressible Navier-Stokes equations in two and three dimensions, Proc. Workshop *Numerical Methods for the Navier-Stokes Equations*, Heidelberg, Oct. 25-28, 1993, Vieweg.

44. A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen*. Habilitation thesis, University of Stuttgart, 2004.
45. A. Sokolichin, G. Eigenberger and A. Lapin, Simulation of buoyancy driven bubbly flow: established simplifications and open questions. *AIChE Journal* **50** (2004) no. 1, 24–45.
46. S. Turek, On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach. *Comput. Methods Appl. Mech. Engrg.* **143** (1997) 271–288.
47. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE **6**, Springer, 1999.
48. S. Turek et al., *FEATFLOW: finite element software for the incompressible Navier-Stokes equations*. User manual, University of Dortmund, 2000. Available at <http://www.featflow.de>.
49. S. Turek, C. Becker and S. Kilian, Hardware-oriented numerics and concepts for PDE software. Special Journal Issue for PDE Software, Elsevier, International Conference on Computational Science ICCS'2002, Amsterdam, 2002, *FUTURE* 1095 (2003), 1-23.
50. S. Turek, C. Becker and S. Kilian, Some concepts of the software package FEAST. In: J. M. Palma, J. Dongarra, V. Hernandez (eds), *VECPAR'98 - Third International Conference for Vector and Parallel Processing*, Lecture Notes in Computer Science, Springer, Berlin, 1999.
51. S. Turek and S. Kilian, An example for parallel ScaRC and its application to the incompressible Navier-Stokes equations, Proc. *ENUMATH'97*, World Science Publ., 1998.
52. S. Turek and R. Schmachtel, Fully coupled and operator-splitting approaches for natural convection. *Int. Numer. Meth. Fluids* **40** (2002) 1109-1119.
53. A. M. P. Valli, G. F. Carey and A. L. G. A. Coutinho, Finite element simulation and control of nonlinear flow and reactive transport. In: *Proceedings of the 10th International Conference on Numerical Methods in Fluids*. Tucson, Arizona, 1998: 450-45.
54. A. M. P. Valli, G. F. Carey and A. L. G. A. Coutinho, Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Commun. Numer. Methods Eng.* **18** (2002), no.2, 131-139.
55. S. P. Vanka, Implicit multigrid solutions of Navier–Stokes equations in primitive variables. *J. Comp. Phys.* **65** (1985) 138–158.
56. J. Van Kan, A second–order accurate pressure–correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.* **7** (1986) 870–891.