3rd International Symposium on Two-Phase Flow Modelling and Experimentation
Pisa, 22-24 September 2004

NUMERICAL SIMULATION OF TURBULENT BUBBLY FLOWS

Dmitri Kuzmin and Stefan Turek

Institute of Applied Mathematics (LS III), University of Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany

A mathematical model for turbulent gas-liquid flows with mass transfer and chemical reactions is presented and a robust solution strategy based on nested iterations is proposed for the numerical treatment of the intricately coupled PDEs. In particular, the incompressible Navier-Stokes equations are solved by a discrete projection scheme from the family of Pressure Schur Complement methods. Novel high-resolution finite element schemes of FCT and TVD type are employed for the discretization of unstable convective terms. The implementation of a modified $k-\varepsilon$ turbulence model is described in detail. The performance of the developed simulation tools is illustrated by a number of three-dimensional numerical examples.

1 INTRODUCTION

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. In the present paper, we adopt a simplified two-fluid model using an analog of the Boussinesq approximation for natural convection problems. Additional transport equations with nonlinear source terms are introduced to describe the evolution of scalar quantities. To solve the resulting PDE system, we propose a numerical algorithm based on the finite element method and dwell on its practical implementation.

2 DRIFT-FLUX MODEL

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.* [21], [22] 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 ϵ 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 system of Navier-Stokes equations

$$\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L = -\nabla p_* + \nabla \cdot (\nu_T \mathcal{D}(\mathbf{u}_L)) - \epsilon \mathbf{g},$$
$$\nabla \cdot \mathbf{u}_L = 0, \qquad p_* = \frac{p - p_{\text{atm}}}{\rho_L} + \mathbf{g} \cdot \mathbf{x} - gh, \qquad (1)$$

where the $\mathcal{D}(\mathbf{u}) = \nabla \mathbf{u} + \nabla \mathbf{u}^T$ and the effective viscosity $\nu_T = C_\mu \frac{k^2}{\varepsilon}$ is a function the turbulent kinetic energy k and its dissipation rate ε. The evolution of these quantities is described by two scalar transport equations

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u}_L - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k + S_k - \varepsilon, \qquad (2)$$
$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon \mathbf{u}_L - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k} (C_1 P_k + C_\varepsilon S_k - C_2 \varepsilon), \qquad (3)$$

where the production terms $P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2$ and $S_k = -C_k \epsilon \nabla p \cdot \mathbf{u}_{\text{slip}}$ are due to the shear and bubble-induced turbulence, respectively. The involved constants $C_\mu = 0.09$, $C_1 = 1.44$, $C_2 = 1.92$, $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$ for the standard $k-\varepsilon$ model are known with high precision, whereas the BIT parameters $C_k \in [0.01, 1]$ and $C_\varepsilon \in [1, 1.92]$ are highly problem-dependent [22]. The slip velocity $\mathbf{u}_{\text{slip}}$ is proportional to the pressure gradient

$$\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_W}, \qquad C_W \approx 5 \cdot 10^4 \ \frac{kg}{m^3 s}.$$

Using the ideal gas law, we can express the gas holdup ϵ and the interfacial area a_S per unit volume as follows [10]

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

where the effective gas density $\tilde{\rho}_G$ and the number density n of bubbles satisfy the following continuity equations

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G \mathbf{u}_G) = -m_{\text{int}}, \qquad (4)$$
$$\frac{\partial n}{\partial t} + \nabla \cdot (n \mathbf{u}_G) = 0. \qquad (5)$$

Coalescence and breakup of bubbles are neglected. The sink term m_{int} is due to reaction-enhanced mass transfer. It depends on the interfacial area a_S and can be modeled in accordance with the standard two-film theory [10]. The effective concentrations of all species in the liquid phase are described by extra convection-reaction-diffusion equations. The interested reader is referred to [9],[10] for details.

In the framework of Sokolichin's *drift-flux* model, the gas velocity $\mathbf{u}_G$ is computed from the algebraic slip relation

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

where the last term is responsible for the bubble path dispersion due to turbulence in the gas phase.

The above system of coupled time-dependent PDEs is to be supplemented by appropriate initial and boundary conditions which depend on the particular application.

3 NUMERICAL ALGORITHM

The discretization in space is performed by an unstructured grid finite element method in order to provide an accurate treatment of non-Cartesian geometries with internal obstacles. A manually generated coarse mesh is successively refined to produce hierarchical data structures for the multigrid solver. The incompressible Navier-Stokes equations are solved by a discrete projection scheme from the family of Multilevel Pressure Schur Complement (MPSC) techniques which were developed by Turek [23] and implemented in the open-source software package FEATFLOW (see *http://www.featflow.de*).

The coupled subproblems are treated sequentially making use of solution values from the previous outer iteration to evaluate the coefficients and source/sink terms which depend on variables other than the one being solved for. The proposed block-iterative algorithm consists of nested loops for the constituents of system (1)–(5). 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 (2)–(3) 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 a fixed-point defect correction scheme is employed for the nonlinear convection term, 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. The updated concentration fields serve as initial data for a nodal ODE system which describes the accumulation and consumption of species. The solution to system (1)–(5) supplemented by the convection-reaction-diffusion equations for effective concentrations is integrated in time from t_n to $t_{n+1} = t_n + \Delta t_n$ as follows:

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

An implicit time discretization of Crank-Nicolson or backward Euler type is employed for all equations. The value of the implicitness parameter θ 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 Δt_n which is chosen adaptively so as to control the changes of the gas holdup distribution (see below). In what follows, we elucidate some constituents of the numerical algorithm in detail.

3.1 Treatment of convection

Convection is notoriously difficult to treat numerically. The standard Galerkin method is a 'centered' scheme which gives rise to an unstable discretization of convective terms. A common remedy is to add streamline diffusion which provides the necessary stabilization without reducing the order of approximation. However, even stabilized high-order methods tend to produce nonphysical undershoots and overshoots in the vicinity of steep gradients. As a result, negative gas holdups or concentrations may arise, which is clearly unacceptable. The positivity of k and ε is even more important, since a negative eddy viscosity would trigger numerical instabilities and eventually lead to a breakdown of the simulation. It is possible to get rid of oscillations by adding adaptive artificial diffusion depending on the local solution behavior. Roughly speaking, a high-order scheme can be used in smooth regions but near discontinuities it should be replaced by a low-order scheme like 'upwind' which is diffusive enough to prevent the formation of wiggles.

The first discretization procedure to utilize the idea of adaptive switching between high- and low-order methods was the *flux-corrected-transport* (FCT) algorithm introduced in the early 1970s by Boris and Book [2]. The state-of-the-art generalization proposed by Zalesak [25] has made it possible to incorporate FCT into unstructured grid methods. The foundations of flux correction for finite elements were laid by Löhner *et al.* [16]. In a series of recent publications [11],[12],[13] we refined the FEM-FCT methodology and extended it to implicit time stepping. Another class of high-resolution methods which has enjoyed an increasing popularity in CFD was developed by Harten [7]. Below we present an generalization of his *total variation diminishing* (TVD) schemes which is based on a fully multidimensional flux limiter and applicable to arbitrary discretizations in space and time [14].

Discrete upwinding

As a model problem, consider the multidimensional transport equation $\frac{\partial u}{\partial t} = -\nabla \cdot (\mathbf{v}u)$ discretized in space by the Galerkin FEM or any other linear high-order scheme. For discretely divergence-free velocity fields, we have

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Ku \quad \Leftrightarrow \quad m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} k_{ij}(u_j - u_i), \qquad (6)$$

where $M_L = \text{diag}\{m_i\}$ is the lumped mass matrix and $K = \{k_{ij}\}$ is the discrete transport operator. If the coefficients k_{ij} were nonnegative $\forall j \neq i$, then the semi-discrete scheme would be *local extremum diminishing* (LED) [8].

The LED property can be enforced by adding an artificial diffusion operator $D = \{d_{ij}\}$ defined by [11]

$$d_{ij} = d_{ji} = \max\{0, -k_{ij}, -k_{ji}\}, \qquad d_{ii} = -\sum_{k\neq i} d_{ik}. \quad (7)$$

Indeed, the resulting low-order operator $L = K + D$ has no negative off-diagonal coefficients. Moreover, this modification is conservative, since the diffusive terms can be represented as a sum of skew-symmetric internodal fluxes: $(Du)_i = \sum_{j\neq i} f^d_{ij}$, where $f^d_{ij} = d_{ij}(u_j - u_i)$, $f^d_{ji} = -f^d_{ij}$. These fluxes are associated with edges of the sparsity graph. Any pair of neighboring nodes i and j whose basis functions have overlapping supports ($k_{ij} \neq 0$ or $k_{ji} \neq 0$) gives rise to such an edge $\vec{ij}$ which is oriented so that $l_{ji} \geq l_{ij}$. This orientation convention implies that node i is located **upwind** and $l_{ij} = \max\{0, k_{ij}\}$.

Flux limiter of TVD type

The 'discrete upwinding' technique presented above proves very handy in multidimensions and yields the least diffusive linear LED scheme. However, **linear** monotonicity-preserving methods are at most first-order accurate. Therefore, excessive artificial diffusion must be removed by adding a **nonlinear** antidiffusion operator F depending on the local smoothness: $K^* = L + F = K + D + F$. In practice, the antidiffusive term Fu is assembled edge-by-edge from skew-symmetric internodal fluxes

$$f^a_{ij} = \min\{\Phi(r_i)d_{ij}, l_{ji}\}(u_i - u_j), \quad f^a_{ji} := -f^a_{ij} \qquad (8)$$

such that $(Fu)_i = \sum_{j\neq i} f^a_{ij}$ at node i. Here Φ is a standard TVD flux limiter (e.g. *minmod* or *superbee*) and r_i is a smoothness indicator evaluated at the upwind node. It can be defined as the slope ratio for a local one-dimensional stencil or as the ratio of auxiliary quantities $Q^\pm_i$ and $P^\pm_i$ which represent the upstream and downstream edge contributions to node i, respectively:

$$P^\pm_i = \sum_{j\neq i} \min\{0, k_{ij}\} \begin{matrix}\min\\ \max\end{matrix} \{0, u_j - u_i\}, \qquad (9)$$

$$Q^\pm_i = \sum_{j\neq i} \max\{0, k_{ij}\} \begin{matrix}\max\\ \min\end{matrix} \{0, u_j - u_i\}. \qquad (10)$$

In other words, all positive antidiffusive fluxes from a downwind node j into an upwind node i are limited using the nodal correction factor $R^+_i = \Phi(Q^+_i/P^+_i)$, while all negative ones are curtailed by $R^-_i = \Phi(Q^-_i/P^-_i)$, so that $f^a_{ij} = \min\{R^\pm_i d_{ij}, l_{ji}\}(u_i - u_j)$. Note that all the necessary information is extracted from the original matrix K. Although the final transport operator K^* does contain negative off-diagonal coefficients, the flux limiter guarantees that the discretization remains local extremum diminishing. It can be proved that for a given u there exists a matrix L^* such that $K^*u = L^*u$ and $l^*_{ij} \geq 0$, $\forall i \neq j$.

After an implicit time discretization by the standard $\theta-$ scheme, one obtains a nonlinear algebraic system

$$M_L \frac{u^{n+1} - u^n}{\Delta t} = \theta K^*(u^{n+1})u^{n+1} + (1-\theta)K^*(u^n)u^n \quad (11)$$

which can be solved iteratively e.g. by the fixed-point defect correction scheme

$$u^{(m+1)} = u^{(m)} + [A(u^{(m)})]^{-1} r^{(m)}, \qquad u^{(0)} = u^n. \quad (12)$$

The low-order operator $A(u^{(m)}) = M_L - \theta\Delta t L(u^{(m)})$ constitutes an excellent 'preconditioner' for it was designed to be an *M-matrix* which is easy to 'invert'. The limited antidiffusive fluxes f^a_{ij} are evaluated edge-by-edge and inserted into the global vectors $b^n = M_L u^n + (1-\theta)\Delta t K^*(u^n)u^n$ and $r^{(m)} = [b^n - A(u^{(m)})u^{(m)}] + \theta\Delta t F u^{(m)}$ [14].

Flux limiter of FCT type

The generalized FEM-FCT formulation [11],[12] provides an alternative way to design the antidiffusive fluxes f^a_{ij} which restore the high accuracy in smooth regions. The total amount of raw antidiffusion is given by

$$f_{ij} = a^{(m)}_{ij}(u^{(l)}_i - u^{(m)}_j) + a^n_{ij}(u^n_i - u^n_j), \quad f_{ji} = -f_{ij}, \quad (13)$$

$$a^{(m)}_{ij} = m_{ij} + \theta\Delta t d^{(m)}_{ij}, \quad a^n_{ij} = -m_{ij} + (1-\theta)\Delta t d^n_{ij}. \quad (14)$$

Solution-dependent correction factors $\alpha_{ij} \in [0, 1]$ are applied to f_{ij} to preclude the comeback of spurious wiggles. The limited antidiffusive fluxes $f^a_{ij} = \alpha_{ij} f_{ij}$ are inserted into the defect vector $r^{(m)}$ to reduce the errors induced by mass lumping and discrete upwinding.

At the beginning of each time step, the subproblem

$$M_L \tilde{u} = [M_L + (1-\theta)\Delta t L(u^n)]u^n \qquad (15)$$

is solved for $\tilde{u}$ to determine the local extrema $\tilde{u}^{\max}_i$ and $\tilde{u}^{\min}_i$. The auxiliary solution $\tilde{u}$ is associated with the time instant $t_{n+1-\theta}$ and reduces to u^n for the backward Euler method. In the iterative version of the FEM-FCT algorithm [13] it also incorporates accepted antidiffusion from the previous defect correction steps, so that only the rejected portion of the net flux needs to be processed. Following Zalesak [25] and Löhner [16], we define

$$P^\pm_i = \sum_{j\neq i} \begin{matrix}\max\\ \min\end{matrix}\{0, f_{ij}\}, \qquad Q^\pm_i = \tilde{u}_i^{\begin{matrix}\max\\ \min\end{matrix}} - \tilde{u}_i. \qquad (16)$$

For the scheme to be positivity-preserving, the flux into node i should be multiplied by

$$R^\pm_i = \begin{cases} \min\{1, m_i Q^\pm_i / P^\pm_i\} & \text{if } P^\pm_i \neq 0, \\ 0, & \text{if } P^\pm_i = 0. \end{cases} \qquad (17)$$

Due to the fact that $f_{ji} = -f_{ij}$ the 'optimal' correction factors are given by

$$\alpha_{ij} = \begin{cases} \min\{R^+_i, R^-_j\} & \text{if } f_{ij} \geq 0, \\ \min\{R^+_j, R^-_i\} & \text{if } f_{ij} < 0. \end{cases} \qquad (18)$$

The ins and outs of the FEM-FCT methodology are elucidated in references [11],[12],[13].

3.2 Implementation of the $k - \varepsilon$ model

The transport equations (2)–(3) are strongly coupled and nonlinear (recall that the turbulent eddy viscosity ν_T depends on both k and ε) so that their numerical solution is anything but trivial. 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. In light of the above, we deem it appropriate to discuss the implementation of the $k - \varepsilon$ model in some detail and present a practical algorithm that proves very robust.

Positivity-preserving linearization

As already mentioned, the coefficients of system (2)–(3) are 'frozen' during each outer iteration and updated as new values of k and ε become available. The quasi-linear transport equations can be solved by a scalar FEM-FCT or FEM-TVD algorithm but the linearization procedure must be tailored to the need to preserve the positivity of k and ε 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 [18],[19] and it is essential to guarantee that the numerical scheme will also possess this property.

Let us cast equations (2)–(3) in the following form

$$\frac{\partial k}{\partial t} + \nabla \cdot (k\mathbf{u} - d_k \nabla k) + \gamma k = P_k + S_k, \tag{19}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon\mathbf{u} - d_\varepsilon \nabla \varepsilon) + C_2 \gamma \varepsilon = \gamma(C_1 P_k + C_\varepsilon S_k), \tag{20}$$

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 ν_T and γ must also be nonnegative for the solution of the convection-reaction-diffusion equations to be well-behaved [3]. 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.* [15] 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 our implicit FEM-FCT/TVD schemes to equations (19) and (20), we arrive at a sequence of nonlinear algebraic systems which can be written as

$$A(u^{(l+1)})u^{(l+1)} = B(u^{(l)})u^{(l)} + q^{(k)}. \tag{21}$$

Here k is the index of the outermost loop in which the velocity $\mathbf{u}$ and the source terms P_k, S_k are updated. The index l refers to the outer iteration for the $k-\varepsilon$ model, while the index m is reserved for inner defect correction loops due to iterative flux correction.

The structure of the involved matrices is as follows:

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

where $K^*(u)$ is the LED transport operator incorporating nonlinear antidiffusion 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 $u^{(l)}$ carries over to the new iterate $u^{(l+1)} = A^{-1}(Bu^{(l)} + q^{(k)})$ provided that $\theta = 1$ (backward Euler method) or the time step is sufficiently small (satisfies a CFL-like condition for $\theta < 1$). In practice, the 'inversion' of A is performed by solving a sequence of linear subproblems within a defect correction loop preconditioned by the monotone low-order operator.

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 ν can be neglected. Hence, we set $\nu_{\text{eff}} = \nu_T$, where the eddy viscosity ν_T is bounded from below by ν and from above by the maximum admissible mixing length $l_{\max}$ (the size of the largest eddies, e.g. the width of the domain). Specifically, we define the limited mixing length l_* as

$$l_* = \begin{cases} \frac{\alpha}{\varepsilon} & \text{if} \quad \varepsilon > \frac{\alpha}{l_{\max}} \\ l_{\max} & \text{otherwise} \end{cases}, \qquad \text{where} \quad \alpha = C_\mu k^{3/2} \tag{22}$$

and use it to update the turbulent eddy viscosity ν_T in the outermost loop:

$$\nu_T = \max\{\nu, l_*\sqrt{k} + C_g\epsilon|\mathbf{u}_{\text{slip}}|r\} \tag{23}$$

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

$$\gamma = C_\mu \frac{k}{\nu_*}, \qquad \text{where} \quad \nu_* = \max\{\nu, l_*\sqrt{k}\}. \tag{24}$$

In the case of a FEM-TVD method, the positivity proof is only valid for the converged solution to (21) while intermediate approximations may exhibit negative values. Since it is impractical to perform many defect correction steps in each outer iteration, we use $k_* = \max\{0, k\}$ rather than k in formulae (22)–(24) in order 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 ν_T and γ 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 ε.

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. After the startup of a gas-liquid reactor, the two-phase flow remains laminar until enough bubbles have entered the flow field for the turbulent effects to become pronounced. Therefore, we activate the $k-\varepsilon$ model at a certain time $t_* > 0$ after the onset of aeration. During the 'laminar' initial phase ($t \le t_*$), a constant effective viscosity ν_0 is prescribed. The values to be assigned to k and ε at $t = t_*$ are uniquely defined by the choice of ν_0 and of the default mixing length $l_0 \in [l_{\min}, l_{\max}]$ where the threshold parameter $l_{\min}$ (e.g. the bubble diameter) corresponds to the size of the smallest admissible eddies. We have

$$k_0 = \left(\frac{\nu_0}{l_0}\right)^2, \qquad \varepsilon_0 = C_\mu \frac{k_0^{3/2}}{l_0} \qquad \text{at} \quad t \le t_*. \tag{25}$$

This strategy was adopted as the effective viscosity ν_0 and the mixing length l_0 are easier to estimate (at least for a CFD practitioner) than k_0 and ε_0. In any case, long-term simulation results are typically not very sensitive to the choice of initial data.

Boundary conditions

If the liquid is supplied along with the gas phase (in a cocurrent or countercurrent mode), then boundary conditions are required at the inlet and outlet. At the inflow boundary Γ_{in} we prescribe all velocity components and the values of k and ε:

$$\mathbf{u} = \mathbf{g}, \quad k = c_{bc}|\mathbf{u}|^2, \quad \varepsilon = C_\mu \frac{k^{3/2}}{l_0} \quad \text{on } \Gamma_{\text{in}}, \tag{26}$$

where $c_{bc} \in [0.003, 0.01]$ is an empirical constant [3] and $|\mathbf{u}| = \sqrt{\mathbf{u}\cdot\mathbf{u}}$ is the Euclidean norm of the velocity. At the outlet Γ_{out}, the normal gradients of all variables are required to vanish, which corresponds to the Neumann boundary conditions

$$\mathbf{n}\cdot\mathcal{D}(\mathbf{u}) = \mathbf{0}, \quad \mathbf{n}\cdot\nabla k = 0, \quad \mathbf{n}\cdot\nabla\varepsilon = 0 \quad \text{on } \Gamma_{\text{out}}. \tag{27}$$

The numerical treatment of inflow and outflow boundary conditions does not present any difficulty. In the finite element framework, relations (27) imply that the surface integrals resulting from integration by parts vanish and do not need to be assembled.

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

$$\mathbf{n}\cdot\mathbf{u} = 0 \qquad \text{on} \quad \Gamma_w \tag{28}$$

is nontrivial if the boundary is not aligned with the axes of the Cartesian coordinate system. In this case, condition (28) 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 [23] 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 (12). This enables us to compute the boundary values of the vector $\mathbf{u}$ explicitly before solving a sequence of linear systems for the velocity components:

$$a_{ij}^{(m)} := 0, \quad \forall j \neq i, \qquad \mathbf{u}_i^* := \mathbf{u}_i^{(m)} + \mathbf{r}_i^{(m)}/a_{ii}^{(m)} \tag{29}$$

for $\mathbf{x}_i \in \Gamma_w$. In the next step, 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

$$\mathbf{u}_i^{(m)} := \mathbf{u}_i^* - (\mathbf{n}_i\cdot\mathbf{u}_i^*)\mathbf{n}_i, \qquad \mathbf{r}_i^{(m)} := 0. \tag{30}$$

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:

$$a_{ij}^{(m)} := 0, \quad \forall j \neq i, \quad \mathbf{r}_i^{(m)} := \mathbf{r}_i^{(m)} - (\mathbf{n}_i\cdot\mathbf{r}_i^{(m)})\mathbf{n}_i. \tag{31}$$

For Cartesian geometries, the modifications to be performed affect just one velocity component (in the normal direction) as in the case of standard Dirichlet boundary conditions. Note that virtually no extra programming effort is required, which is a significant advantage as compared to another feasible approach based on local coordinate transformations during the matrix assembly [4].

Wall functions

To complete the problem statement we still need to prescribe the tangential stress as well as the boundary values of k and ε on Γ_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 Γ_δ located at a distance δ from the solid wall Γ_w [17],[18],[19].

In essence, a boundary layer of width δ is removed from the actual computational domain Ω and the equations are solved in the reduced domain Ω_δ subject to the following empirical boundary conditions:

$$\mathbf{n}\cdot\mathcal{D}(\mathbf{u})\cdot\mathbf{t} = -\frac{u_\tau^2}{\nu_T}\frac{\mathbf{u}}{|\mathbf{u}|}, \quad k = \frac{u_\tau^2}{\sqrt{C_\mu}}, \quad \varepsilon = \frac{u_\tau^3}{\kappa\delta} \quad \text{on } \Gamma_\delta. \tag{32}$$

Here the unit vector $\mathbf{t}$ refers to the tangential direction, $\kappa = 0.41$ is the von Kármán constant and u_τ is the *friction velocity* which is assumed to satisfy the nonlinear equation

$$g(u_\tau) = |\mathbf{u}| - u_\tau\left(\frac{1}{\kappa}\log y^+ + 5.5\right) = 0 \tag{33}$$

in the *logarithmic layer*, where the local Reynolds number $y^+ = \frac{u_\tau\delta}{\nu}$ is in the range $20 \le y^+ \le 100$, and be a linear function of y^+ in the *viscous sublayer*, where $y^+ < 20$. Note that $\mathbf{u}$ is the tangential velocity as long as condition (28) is imposed on Γ_δ.

Equation (33) can be solved iteratively by Newton's method [17]:

$$u_\tau^{l+1} = u_\tau^l - \frac{g(u_\tau^l)}{g'(u_\tau^l)} = u_\tau^l + \frac{|\mathbf{u}| - u_\tau f(u_\tau^l)}{1/\kappa + f(u_\tau^l)}, \tag{34}$$

where the auxiliary function f is given by

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

The friction velocity is initialized by $u_\tau^0 = \sqrt{\frac{\nu|\mathbf{u}|}{\delta}}$ 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 $y^+ \to 0$.

The friction velocity u_τ is plugged into (32) to compute the tangential stress, which yields a natural boundary condition for the velocity. Integration by parts in the weak form of the incompressible Navier-Stokes equations (1) gives rise to a surface integral over the internal boundary Γ_δ which contains the prescribed traction:

$$\int_{\Gamma_\delta} \nu_T(\mathbf{n}\cdot\mathcal{D}(\mathbf{u})\cdot\mathbf{t})\cdot\mathbf{v}\,ds = -\int_{\Gamma_\delta} u_\tau^2\frac{\mathbf{u}}{|\mathbf{u}|}\cdot\mathbf{v}\,ds. \tag{35}$$

The Dirichlet boundary conditions for k and ε are imposed in the strong sense. For further details regarding the implementation of wall laws in the finite element framework the reader is referred to [17],[18],[19].

3.3 Underrelaxation for outer iterations

Recall that the iterative solution process is based on the following hierarchy of loops

- the main *n-loop* for the global time-stepping
- the outermost *k-loop* for system (1)–(5)
- the outer *l-loop* fot the $k-\varepsilon$ model (2)–(3)
- the outer *l-loop* for the PSC equation, see [23]
- embedded *m-loops* for flux/defect correction (12)

For each time step (one $n-$loop iteration), the equations of the drift-flux model are solved repeatedly within the k-loop. The latter contains the two l-loops as well as subordinate m-loops for each scalar transport problem.

Due to the intricate coupling of the PDEs at hand, 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 solution increments before applying them to the last iterate:

$$u^{(m+1)} := u^{(m)} + \omega^{(m)}(u^{(m+1)} - u^{(m)}). \qquad (36)$$

The damping factor $\omega^{(m)}$ may be chosen adaptively so as to accelerate convergence and to minimize the error in a certain norm. 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 ϵ or ν_T.

Furthermore, the m-loops lend themselves to the use of an *impicit underrelaxation* strategy which increases the diagonal dominance of the preconditioner [5],[20]:

$$a_{ii}^{(m)} := \frac{a_{ii}^{(m)}}{\alpha^{(m)}}, \qquad \text{where} \quad 0 \le \alpha^{(m)} \le 1. \qquad (37)$$

The scaling of the diagonal entries does not affect the converged solution and proves more robust than *explicit underrelaxation* (36). In fact, no underrelaxation whatsoever is needed for moderate time steps which are typically used in dynamic simulations.

3.4 Linear solvers and time step

Last but not least, let us briefly discuss the choice of the linear solver and of the time discretization. In many cases, explicit schemes are rather inefficient due to severe stability limitations which require taking impractically small time steps. For this reason, we restrict ourselves to the implicit Crank-Nicolson and backward Euler methods which are unconditionally stable and permit large time steps at the cost of solving nonsymmetric linear systems. In our experience, BiCGSTAB and geometric multigrid constitute excellent solvers as long as the parameters are properly tuned and the underlying smoothers/preconditioners are consistent with the size of the time step. If Δt is rather small, standard components like Jacobi, Gauß-Seidel and SOR schemes will suffice. For large time steps, the condition number of the matrix deteriorates and convergence may fail. This can be rectified by resorting to an ILU factorization and an appropriate renumbering technique.

In order to capture the dynamics of the two-phase flow in a computationally efficient way, we employ adaptive time-stepping based on the PID controller [24]. Choosing the local gas holdup ϵ to be the indicator variable, one obtains the following algorithm:

1. Monitor the relative changes of the gas holdup
$$e_n = \frac{||\epsilon^{n+1} - \epsilon^n||}{||\epsilon^{n+1}||}.$$
2. If $e_n > \delta$ reject the solution 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 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:
$$\Delta t_{\min} \le \Delta t_{n+1} \le \Delta t_{\max}, \quad m \le \frac{\Delta t_{n+1}}{\Delta t_n} \le M.$$

4 NUMERICAL EXAMPLES

The developed finite element software builds on the FEATFLOW package and on the underlying FEAT libraries. The multidimensional TVD limiter was incorporated into the matrix assembly routine for the momentum equation and has proved its worth for the nonconforming $\tilde{Q}_1$ finite elements. The subroutines for the solution of scalar transport equations were equipped with both FCT and TVD limiters. The performance of these algebraic high-resolution schemes was found to be comparable.

The first example deals with the locally aerated bubble column which was investigated by Becker *et al.* [1]. The snapshots of the meandering bubble swarm displayed in Fig. 1 agree well with experimental data.

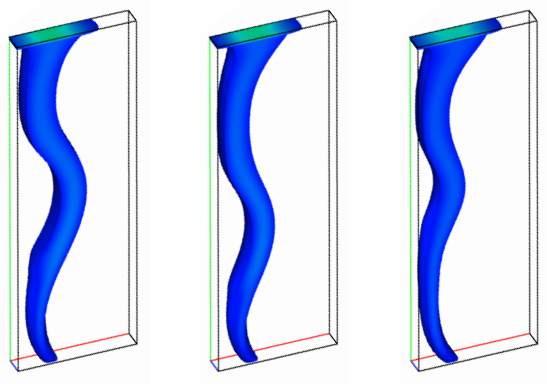

Fig. 1. Locally aerated bubble column.

The next figure shows the developing gas holdup distribution (top) and instantaneous velocity fields (bottom) in the middle cross section of a prototypical airlift loop reactor (the ragged boundary of the internal part is due to the Cartesian mesh used for visualization in MATLAB). The aeration takes place at the bottom of the riser section

where both phases flow cocurrently in the upward direction. At the upper surface, the bubbles escape while the liquid is diverted into the gas-free downcomer forming a closed loop. The two-phase flow reaches a steady state within a few seconds after the startup.

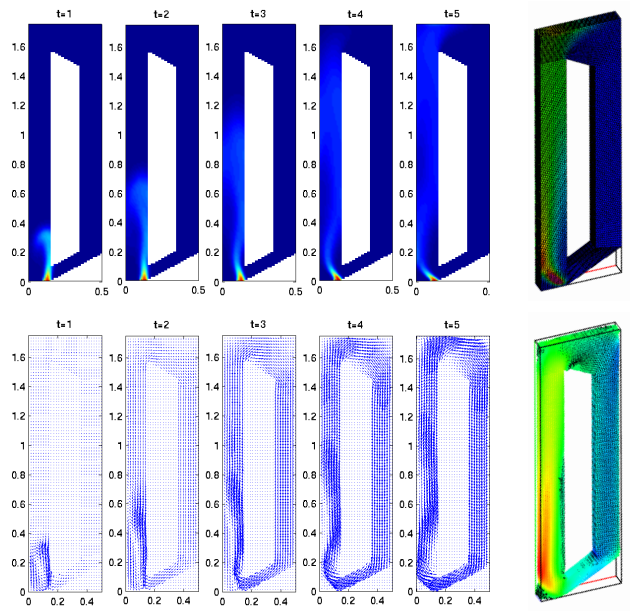

Fig. 2. Flow structure in an airlift loop reactor.

To give an insight into the complex interplay of physical and chemical phenomena, let us consider the absorption of CO_2 in a locally aerated bubble column filled with water. The construction parameters are those defined by Becker *et al.* [1] and adopted in our first 2D simulations for this test case [9]. In the absence of mass transfer, the rising bubbles would expand due to the fall of hydrostatic pressure. However, carbon dioxide is highly soluble in water so that the growth is diminished and the bubbles may even shrink. These effects are nicely reproduced by our model.

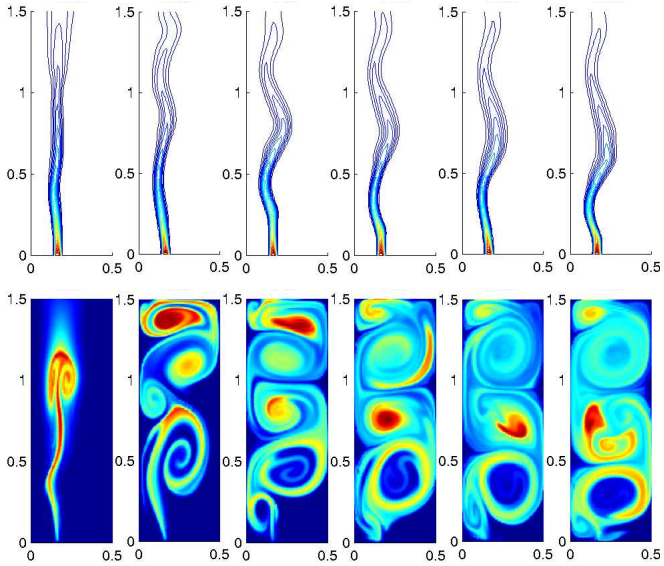

Fig. 3. Physical absorption of CO_2 (2D simulation).

In the case of physical absorption, the gas dissolves rather slowly and neither the gas holdup distribution (Fig. 3, top) nor the velocity fields are significantly affected by the absorption process [9]. The flow structure consisting of three large vortices is reflected by the concentration fields for the dissolved carbon dioxide (Fig. 3, bottom).

The flow pattern is totally different if the mass transfer is accelerated by chemical reactions in the liquid phase. Let us demonstrate this by taking an aqueous solution of sodium hydroxide instead of water. The reaction $CO_2 + 2NaOH \rightarrow Na_2CO_3 + H_2O$ is so fast that the bubbles are completely dissolved within 20 *cm* from the inlet (Fig. 4, top). There is no liquid circulation in the upper part of the reactor since the gas does not reach it. A large vortex develops in the vicinity of the gas sparger twisting the concentration of the produced sodium carbonate into a spiral shape (Fig. 4, bottom).

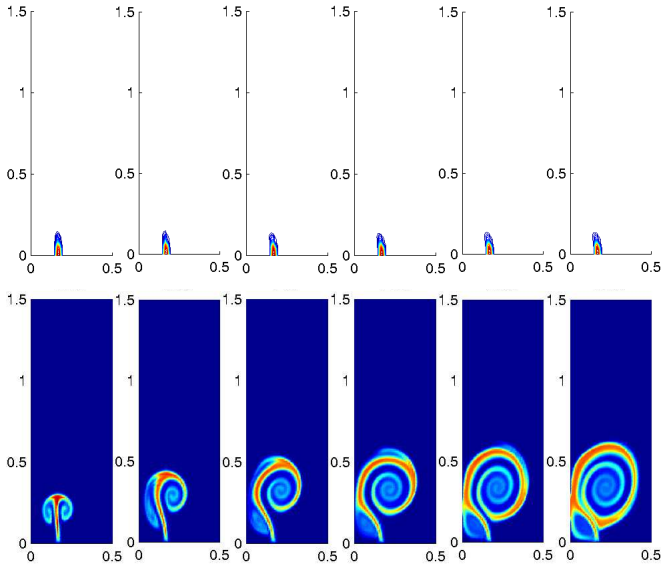

Fig. 4. Reaction-enhanced absorption of CO_2 (2D).

A 3D simulation predicts essentially the same flow behavior depicted in Fig. 5. The results are in good agreement with experimental data reported in [6].

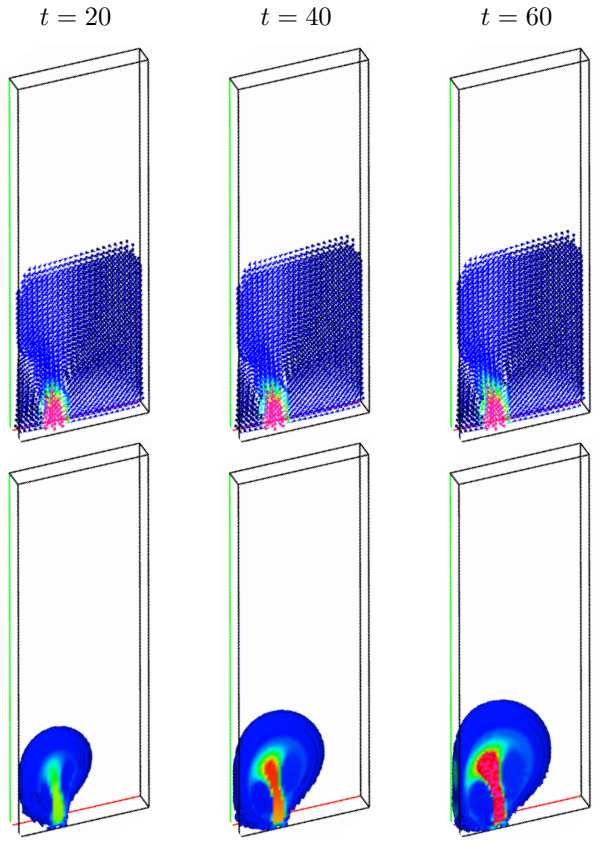

Fig. 5. Reaction-enhanced absorption of CO_2 (3D).

5 CONCLUSIONS

The drift-flux model for buoyancy-driven bubbly flows was coupled with scalar transport equations describing the absorption of gas followed (and enhanced) by chemical reactions in the liquid phase. Turbulence effects were taken into account by means of a modified $k-\epsilon$ turbulence model which was found to produce reasonable results. Further research is needed to improve the modeling of the bubble-induced turbulence. Recent advances in the field of Large Eddy Simulation (LES) indicate that it might be a good candidate for this purpose.

An unstructured grid finite element method was proposed for the numerical solution. The discretization of the troublesome convective terms was performed by a nonlinear positivity-preserving scheme equipped with a multidimensional flux limiter of FCT or TVD type. Nested iterations were used to provide the coupling of model equations, to get rid of nonlinearities and to solve the linear systems. Some subtleties of the algorithm were discussed in detail. The interplay of the hydrodynamics, mass transfer and chemical reaction was illustrated by numerical examples.

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 and D. L. Book, Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works. *J. Comput. Phys.* **11** (1973) 38–69.

[3] 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.

[4] 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.

[5] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.

[6] C. Fleischer, S. Becker and G. Eigenberger, Detailed modeling of the chemisorption of CO_2 into NaOH in a bubble column. *Chem. Eng. Sci.* **51** (1996), no. 10, 1715–1724.

[7] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* **49** (1983) 357–393.

[8] A. Jameson, Positive schemes and shock modelling for compressible flows. *Int. J. Numer. Meth. Fluids* **20** (1995) 743–776.

[9] 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.

[10] 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.

[11] D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525-558.

[12] D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. Technical report 215: University of Dortmund, 2002. Submitted to *Int. J. Numer Meth. Fluids*.

[13] D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. Submitted to *Comp. Meth. Appl. Mech. Engrg.*

[14] D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. Submitted to *J. Comput. Phys.*

[15] 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.

[16] R. Löhner, K. Morgan, J. Peraire and M. Vahdati, Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.

[17] G. Medić and B. Mohammadi, NSIKE - an incompressible Navier-Stokes solver for unstructured meshes. *INRIA Research Report* **3644** (1999).

[18] B. Mohammadi, A stable algorithm for the k-epsilon model for compressible flows. *INRIA Research Report* **1355** (1990).

[19] B. Mohammadi and O. Pironneau, *Analysis of the k-epsilon turbulence model*. Wiley, 1994.

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

[21] A. Sokolichin, G. Eigenberger and A. Lapin, Simulation of buoyancy driven bubbly flow: established simplifications and open questions. To appear in *AIChE Journal*.

[22] A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen*. Habilitationsschrift, Universität Stuttgart, 2002.

[23] S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE 6, Springer, 1999.

[24] 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.

[25] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.