Finite element simulation of turbulent bubbly flows in gas-liquid reactors

D. Kuzmin[1] and S. Turek

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

H. Haario

Laboratory of Applied Mathematics, Lappeenranta University of Technology
P.O. Box 20, FIN-53851, Lappeenranta, Finland

Abstract

A mathematical model for turbulent gas-liquid flows is presented. It is shown that bubble-induced buoyancy resembles natural convection and can be readily incorporated into an incompressible flow solver by using an analog of the Boussinesq approximation. Extra transport equations are introduced to describe the evolution of the gas holdup and compute the turbulent eddy viscosity in the framework of a generalized $k-\varepsilon$ model. A robust solution strategy based on nested iterations is proposed for the numerical treatment of the resulting PDE system. The incompressible Navier-Stokes equations are solved by a discrete projection scheme from the family of Pressure Schur Complement methods. High-resolution finite element schemes designed on the basis of algebraic flux correction are employed for the discretization of convective terms. Numerical results for prototypical gas-liquid reactor configurations illustrate the performance of the developed simulation tools.

Key Words: CFD; dispersed two-phase flow; convection-dominated transport; incompressibility; turbulence; finite elements; flux correction

1 Introduction

Bubble columns and airlift loop reactors are widely used in industry as contacting devices in which gaseous and liquid species are brought together to engage in chemical reactions. The liquid supplied continuously or in a batch mode is 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. In airlift loop reactors, internal parts are installed in order to enforce a stable circulation of liquid characterized by the presence of pronounced *riser* and *downcomer* zones. Shape optimization appears to be a promising way to improve the reactor performance by adjusting the geometry of internals.

In this paper, we touch upon the mathematical modeling of gas-liquid flows and propose a numerical algorithm based on the finite element method. A hierarchy of successively

[1]Correspondence to: kuzmin@math.uni-dortmund.de

refined unstructured meshes is used to deal with complex domains. The velocity-pressure coupling via the incompressibility constraint is imposed in the framework of the Multilevel Pressure Schur Complement formulation originally developed for single-phase flows [40]. The numerical instability of convective terms poses a hazard to the positivity of the gas holdup, concentrations, turbulent kinetic energy and other quantities which must remain nonnegative for physical reasons. Standard remedies like upwinding degrade the accuracy of the numerical solution and produce unacceptable results. The only way to obtain a sharp resolution of steep fronts without generating wiggles is to use a nonlinear combination of high- and low-order methods. To this end, we construct a high-resolution finite element scheme making use of algebraic flux correction [21] based on a fully multidimensional flux limiter. Furthermore, we discuss some other algorithmic aspects relevant for the simulation of gas-liquid flows and implementation of the $k-\varepsilon$ turbulence model. Numerical examples are presented to give a flavor of feasible applications.

2 Mathematical model

Detailed hydrodynamic models for dispersed two-phase flow can be classified into those of Euler-Euler and Euler-Lagrange type. The former approach implies that both phases are treated as space-sharing interpenetrating continua. The macroscopic conservation laws can be postulated using some heuristic arguments or derived mathematically by applying a suitable averaging procedure to the associated single-phase continuity and momentum equations [7], [8]. In the Euler-Lagrange formulation, only the liquid phase is considered to be continuous, while individual bubbles or bubble clusters are tracked in a Lagrangian way [23],[38]. The most detailed treatment of gas-liquid flow is provided by direct numerical simulation (DNS), which amounts to solving a formidable free boundary problem for the deformation of all bubbles and interactions between them. At present, this is only feasible for a very limited number of bubbles. Nevertheless, DNS contributes to the understanding of processes taking place at the microscopic level and constitutes a valuable tool for the derivation of input parameters and correlations for less sophisticated CFD models.

In this paper, we follow the Euler-Euler approach, whereby the computational cost is largely independent of the number of bubbles present in the flow field. Let the local gas holdup be denoted by $\alpha_G = \alpha$, so that the volume fraction of liquid equals $\alpha_L = 1 - \alpha$. In the absence of mass transfer, the macroscopic continuity equations read [7],[8]

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G \mathbf{u}_G) = 0, \qquad \tilde{\rho}_G = \alpha_G \rho_G, \tag{1}$$

$$\frac{\partial \tilde{\rho}_L}{\partial t} + \nabla \cdot (\tilde{\rho}_L \mathbf{u}_L) \ = 0, \qquad \tilde{\rho}_L = \alpha_L \rho_L. \tag{2}$$

Here and below the tilde denotes multiplication by the volume fraction of the respective phase. The associated momentum equations are given by [7],[8]

$$\tilde{\rho}_G \left[\frac{\partial \mathbf{u}_G}{\partial t} + \mathbf{u}_G \cdot \nabla \mathbf{u}_G \right] = -\alpha_G \nabla p + \nabla \cdot \mathcal{S}_G + \tilde{\rho}_G \mathbf{g} + \mathbf{f}_{\text{int}}, \tag{3}$$

$$\tilde{\rho}_L \left[\frac{\partial \mathbf{u}_L}{\partial t} + \ \mathbf{u}_L \cdot \nabla \mathbf{u}_L \right] = -\alpha_L \nabla p + \nabla \cdot \mathcal{S}_L + \tilde{\rho}_L \mathbf{g} - \mathbf{f}_{\text{int}}, \tag{4}$$

where $\mathcal{S}$ is the viscous stress tensor and the coupling term $\mathbf{f}_{\text{int}}$ in both right-hand sides is responsible for the momentum exchange at the moving gas-liquid interface.

The above *two-fluid model* is widely used in the CFD community. For the number of equations to match the number of unknowns, this PDE system should be complemented by appropriate constitutive relations. In accordance with the ideal gas law, the common pressure p is assumed to satisfy the following equation of state

$$p = \rho_G R T, \tag{5}$$

where T denotes the (constant) temperature and R is the specific gas constant. Furthermore, it is necessary to express the interphase force $\mathbf{f}_{\text{int}}$ and the viscous stress tensor $\mathcal{S}$ in terms of the independent variables being solved for. Due to the complex phase interactions, the derivation of such models turns out to be a highly challenging task. In what follows, we present constitutive equations suitable for buoyancy-driven bubbly flows.

2.1 Turbulence modelling

The eddy viscosity hypothesis for the Reynolds-averaged Navier-Stokes (RANS) equations leads to the following representation of the viscous stress tensor

$$\mathcal{S}(\mathbf{u}) = \rho\nu_{\text{eff}}[\nabla\mathbf{u} + (\nabla\mathbf{u})^T], \qquad \nu_{\text{eff}} = \nu + \nu_T, \tag{6}$$

where the laminar viscosity ν depends solely on the physical properties of the fluid, while the turbulent eddy viscosity ν_T is supposed to emulate the influence of microscopic velocity fluctuations $\mathbf{u}'$ that were filtered out in the course of averaging.

Turbulence modelling for multiphase flows is subject to a great deal of uncertainty, and no universally applicable models are available to date. For this reason, straightforward generalizations of single-phase RANS models are commonly used in practice. In particular, the standard $k-\varepsilon$ model appears to produce sensible results [2],[37] and can be extended to allow for the bubble-induced turbulence (BIT) [12],[31],[36]. As a rule, the gas phase is assumed to be 'laminar'. The turbulent eddy viscosity ν_T of the liquid phase is a function of the turbulent kinetic energy k and its dissipation rate ε, namely

$$\nu_T = C_\mu \frac{k^2}{\varepsilon}, \qquad \text{where} \quad k = \frac{1}{2}\langle|\mathbf{u}'|^2\rangle, \quad \varepsilon = \frac{\nu}{2}\langle|\nabla\mathbf{u}' + (\nabla\mathbf{u}')^T|^2\rangle. \tag{7}$$

A naive approach to the modelling of bubble-induced turbulence consists of adding an extra contribution $\nu_G(\alpha)$ to the eddy viscosity ν_T. Sokolichin [36] showed that this straightforward modification is not to be recommended as a stand-alone model, since it tends to underestimate the influence of the dispersed phase and neglect the convective transport of turbulent kinetic energy produced by the bubbles in aerated regions. The following generalization of the $k-\varepsilon$ model was found to perform much better [12],[31],[36]

$$\frac{\partial k}{\partial t} + \nabla\cdot\left(k\mathbf{u} - \left(\nu + \frac{\nu_T}{\sigma_k}\right)\nabla k\right) = P_k + S_k - \varepsilon, \tag{8}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla\cdot\left(\varepsilon\mathbf{u} - \left(\nu + \frac{\nu_T}{\sigma_\varepsilon}\right)\nabla \varepsilon\right) = \frac{\varepsilon}{k}(C_1 P_k + C_\varepsilon S_k - C_2\varepsilon), \tag{9}$$

where the production terms $P_k = \frac{\nu_T}{2}|\nabla\mathbf{u} + (\nabla\mathbf{u})^T|^2$ and $S_k = -\alpha C_k|\nabla p|^2$ 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 and C_ε are highly problem-dependent and their values must be determined by trial and error [36]. Therefore, the applicability of the above model is rather limited and should be verified in each particular case.

2.2 Interphase force

The structure of the interphase transfer term $\mathbf{f}_{\text{int}}$, which links the momentum balance equations (3) and (4), has been a major source of controversy in the two-phase flow modelling community. In spite of the ongoing dispute over the relevance and importance of various mechanisms for momentum transfer, it is generally believed that $\mathbf{f}_{\text{int}}$ involves contributions from the drag force, the virtual mass force, and the lift force [8]

$$\mathbf{f}_{\text{int}} = \mathbf{f}_D + \mathbf{f}_{VM} + \mathbf{f}_L. \tag{10}$$

By far the most important constituent is the drag force $\mathbf{f}_D$ experienced by a bubble as it moves steadily in the surrounding liquid. The ensuing interphase friction is given by

$$\mathbf{f}_D = -\alpha C_D \frac{3}{8}\frac{\rho_L}{r}|\mathbf{u}_G - \mathbf{u}_L|(\mathbf{u}_G - \mathbf{u}_L), \tag{11}$$

where r stands for the bubble radius and C_D is the dimensionless drag coefficient which can be assessed by measuring the terminal rise velocity of a single bubble. Numerous empirical correlations are available for C_D as a function of the Reynolds number. Note that the drag force is proportional to the relative velocity $\mathbf{u}_G - \mathbf{u}_L$ so that the bubbles slow down, whereas the liquid gains momentum.

The 'added mass' of liquid entrained in the wake of accelerating bubbles gives rise to an extra force which tends to equalize the accelerations in the long run

$$\mathbf{f}_{VM} = -\alpha C_{VM}\rho_L\left(\frac{\mathrm{d}\mathbf{u}_G}{\mathrm{d}t} - \frac{\mathrm{d}\mathbf{u}_L}{\mathrm{d}t}\right). \tag{12}$$

Depending on the properties of the gas-liquid mixture, the virtual mass coefficient C_{VM} may deviate appreciably from the default value $C_{VM} = 0.5$ valid for spherical particles.

Additional forces transverse to the direction of motion are typically represented by

$$\mathbf{f}_L = -\alpha C_L \rho_L(\mathbf{u}_G - \mathbf{u}_L) \times (\nabla \times \mathbf{u}_L), \tag{13}$$

where the empirical lift coefficient C_L equals 0.25 for dilute flows of spheres.

The implications of the three contributions to the interphase force were analyzed in detail by Sokolichin *et al.* [34],[35]. They demonstrated that the virtual mass force $\mathbf{f}_{VM}$ has hardly any influence on the simulation results and, therefore, can/should be neglected in order to reduce the computational cost. At the same time, they found the impact of the lift force to be significant but questioned its existence in the form (13) which was originally derived for solid particles. In the literature, $\mathbf{f}_L$ is frequently misused for fitting the results to experimental data. In some publications even the sign of the lift coefficient is reversed. Sokolichin *et al.* condemned this practice and argued that the lift force should be omitted as long as its origins, form, and magnitude remain unclear.

2.3 Algebraic slip relation

The above arguments indicate that the viscous drag force $\mathbf{f}_D$ is the only one which needs to be taken into account. Moreover, the density of gas is much smaller than that of liquid. Therefore, the inertia and gravity terms in the gas phase momentum balance (3) can be neglected and it reduces to the equilibrium relation [34],[35]

$$0 = -\alpha \nabla p + \mathbf{f}_D \tag{14}$$

which makes it possible to compute the relative velocity $\mathbf{u}_{\text{slip}}$. To this end, it is worthwhile to linearize the drag force as proposed by Schwarz and Turner [33]:

$$\mathbf{f}_D = -\alpha C_W \mathbf{u}_{\text{slip}}, \qquad C_W = C_D \frac{3}{8} \frac{\rho_L}{r} |\mathbf{u}_{\text{slip}}| \approx 5 \cdot 10^4 \ \frac{kg}{m^3 s}. \tag{15}$$

Unlike the standard drag coefficient C_D, the linearization parameter C_W is a dimensional quantity which is assigned a constant value independent of the Reynolds number.

The use of approximation (15) in (14) yields the (laminar) slip velocity

$$\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_W} \tag{16}$$

which is approximately equal to the terminal rise velocity of a single bubble in a stangant liquid. The above value of C_W corresponds to $\mathbf{u}_{\text{slip}} \approx 20 \ cm/s$.

In the presence of turbulence, the motion of the gas phase is subject to 'bubble path dispersion' which can be included into the hydrodynamic model as an extra contribution to the slip velocity. This 'drift velocity' $\mathbf{u}_{\text{drift}}$ is proportional to the gradient of the gas holdup α and directed opposite to it. The associated dispersion coefficient depends on the turbulent eddy viscosity ν_T and on the turbulent Schmidt number σ [34],[36]

$$\mathbf{u}_{\text{drift}} = -\frac{\nu_T}{\sigma} \frac{\nabla \alpha}{\alpha}. \tag{17}$$

In light of the above, the velocities of the gas and liquid phase are related by

$$\mathbf{u}_G = \mathbf{u}_L + \mathbf{u}_{\text{slip}} + \mathbf{u}_{\text{drift}}. \tag{18}$$

This algebraic relation can be used instead of the gas phase momentum equation.

2.4 Drift-flux model

The two-fluid model (1)–(4) is not the only way to deal with dispersed two-phase flows in a fully continuous framework. Alternatively, the gas-liquid mixture can be treated as a single fluid whose effective density and momentum are defined by

$$\rho = \tilde{\rho}_L + \tilde{\rho}_G, \qquad \rho \mathbf{u} = \tilde{\rho}_L \mathbf{u}_L + \tilde{\rho}_G \mathbf{u}_G. \tag{19}$$

The weakly compressible Navier-Stokes equations for these quantities constitute the so-called drift-flux model. If the mixture velocity $\mathbf{u}$ is available, the velocities of both phases can be recovered using the algebraic slip relation (18).

Interestingly enough, the sum of (1) and (2) yields the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{20}$$

Furthermore, if the simplified gas phase momentum balance (14) is added to equation (4) with the stress tensor given by (6), the interphase force term vanishes and we obtain

$$\tilde{\rho}_L \left[\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L \right] = -\nabla p + \nabla \cdot \mathcal{S}(\mathbf{u}_L) + \tilde{\rho}_L \mathbf{g}. \tag{21}$$

Recall that the gas density is very small as compared to that of liquid. Therefore, its contribution to the mixture density and momentum is negligible so that

$$\rho \approx \tilde{\rho}_L, \quad \mathbf{u} \approx \mathbf{u}_L. \tag{22}$$

Under these assumptions, equation (21) reduces to the momentum balance for the mixture, which means that the two-fluid and drift-flux models are largely equivalent.

2.5 Boussinesq approximation

The effective density $\tilde{\rho}_L = (1-\alpha)\rho_L$ depends on the local gas holdup, which creates buoyancy in aerated regions. Similar processes occur in single-phase flows in the presence of temperature gradients. Following Sokolichin and Eigenberger [34],[35], we use an analog of the Boussinesq approximation for natural convection problems and replace $\tilde{\rho}_L$ by ρ_L everywhere except for the gravity force in (21). This trick leads to the incompressible Navier-Stokes equations with an extra buoyancy term proportional to the gas holdup:

$$\begin{aligned} &\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L = -\nabla p_* + \nabla \cdot (\nu_{\text{eff}} [\nabla \mathbf{u}_L + (\nabla \mathbf{u}_L)^T]) - \alpha \mathbf{g}, \\ &\nabla \cdot \mathbf{u}_L = 0, \qquad p_* = \frac{p - p_0}{\rho_L}, \qquad \nu_{\text{eff}} = \nu + \nu_T, \end{aligned} \tag{23}$$

where p_0 denotes the hydrostatic pressure. In general, the isotropic part the Reynolds stress tensor increases p_* by $\frac{2}{3}k$ but this contribution can usually be neglected [27].

In order to evaluate the buoyancy force in the right-hand side of the momentum equation, one should solve the continuity equation (1) and convert the effective density $\tilde{\rho}_G$ into the local gas holdup α making use of the ideal gas law (5) which yields

$$\alpha = \frac{\tilde{\rho}_G}{\rho_G} = \frac{\tilde{\rho}_G}{p} RT. \tag{24}$$

At high gas throughputs it is advisable to relax the Boussinesq approximation and solve the momentum equation in its original form (21). As far as the continuity equation is concerned, the constant density assumption can be retained. This model simplification is particularly advantageous from the computational viewpoint, since it eliminates the need for dealing with the evolution of the free surface on top of the reactor [34],[35].

2.6 Summary of equations

Let us summarize the governing equations of the simplified two-fluid / drift-flux model which provides a fairly detailed information about the flow at a relatively low computational cost. The problem to be solved consists of the following equation blocks:

- Incompressible Navier-Stokes equations / Boussinesq approximation

$$\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L = -\nabla p_* + \nabla \cdot (\nu_{\text{eff}}[\nabla \mathbf{u}_L + (\nabla \mathbf{u}_L)^T]) - \alpha \mathbf{g},$$

$$\nabla \cdot \mathbf{u}_L = 0, \qquad p_* = \frac{p - p_0}{\rho_L}, \qquad \nu_{\text{eff}} = \nu + \nu_T. \tag{25}$$

- Coupled equations of the generalized $k - \varepsilon$ turbulence model

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u}_L - \left(\nu + \frac{\nu_T}{\sigma_k} \right) \nabla k \right) = P_k + S_k - \varepsilon, \tag{26}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon \mathbf{u}_L - \left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \nabla \varepsilon \right) = \frac{\varepsilon}{k}(C_1 P_k + C_\varepsilon S_k - C_2 \varepsilon) \tag{27}$$

  supplemented by the following set of auxiliary relations

$$\nu_T = C_\mu \frac{k^2}{\varepsilon}, \qquad P_k = \frac{\nu_T}{2} |\nabla \mathbf{u}_L + (\nabla \mathbf{u}_L)^T|^2, \qquad S_k = \alpha C_k |\nabla p|^2.$$

- Algebraic slip relation for the computation of the gas phase velocity

$$\mathbf{u}_G = \mathbf{u}_L + \mathbf{u}_{\text{slip}} + \mathbf{u}_{\text{drift}}, \tag{28}$$

  where the laminar and turbulent slip velocities are given in the form

$$\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_W}, \qquad \mathbf{u}_{\text{drift}} = -\frac{\nu_T}{\sigma} \frac{\nabla \alpha}{\alpha}. \tag{29}$$

- Continuity equation for the effective density / volume fraction of gas

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G \mathbf{u}_G) = 0, \qquad \alpha = \frac{\tilde{\rho}_G}{p} RT. \tag{30}$$

This mathematical model is to be endowed with appropriate initial and boundary conditions which depend on the particular application. Mass transfer and chemical reaction models can be readily included as explained in [13],[14],[19],[20]. In this case, additional transport equations are to be solved for the number density of bubbles as well as for the effective concentrations of all species in the liquid phase.

Admittedly, the above equations do not represent the whole range of bubbly flows which occur in real-life applications. Many additional forces and interactions may play an important role, especially at high superficial gas velocities. In particular, it is common practice to consider the virtual mass force and questionable 'radial' forces in the gas phase

momentum equation. Furthermore, some authors argue that the linear relation (15) is unrealistic and propose other empirical correlations for the drag coefficient which depend on the Reynolds number and on the local gas holdup. The influence of the bubble wake, compressibility effects, coalescence/breakup, bubble-induced turbulence and many other phenomena are known to be important but no universal models are available to date.

An alarming trend is to include 'all relevant forces' and fit the involved 'free' parameters to experimental data. This practice renders the results of a CFD simulation worthless because the values of the empirical constants are problem-dependent and not applicable to other flow configurations for which measurements are impossible or prohibitively expensive. As a result, such 'advanced' models do not improve the predictive capability of CFD software but merely increase the computational cost. Thus, we subscribe to the argument of Sokolichin *et al.* [35] who indicate that 'relevant forces' whose

- existence is not experimentally verified and/or
- influence on the simulation results is negligible

do not belong into the mathematical model. It is not unusual that simplified models yield better results. However, one should be aware of their limitations and check if the underlying assumptions are valid in each particular case.

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. The incompressible Navier-Stokes equations call for the use of an LBB-stable finite element pair. A suitable candidate is the nonconforming Rannacher-Turek element (rotated multilinear velocity, piecewise-constant pressure) [32]. Standard multilinear elements are employed for other variables. The manually generated coarse mesh is successively refined to produce hierarchical data structures for the multigrid solver [41],[42].

The coupled subproblems are solved 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 (25)–(30). 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 $k-\varepsilon$ model (26)–(27) which are closely related to each other 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 resulting linear systems are solved by a suitable iterative method (e.g. Gauß-Seidel, preconditioned BiCGSTAB or multigrid) whereby the convergence criteria for the inner iterations may be rather loose, as long as all the variables and coefficients are updated again in the next outer iteration. The iterative process terminates when the residual of the momentum equation and/or the relative changes of all variables become small enough.

The solution to (25) –(30) is integrated in time from t_n to $t_{n+1} = t_n + \Delta t_n$ as follows:

1. Solve the Navier-Stokes equations (25) for $\mathbf{u}_L$ and p.
2. Use an L_2-projection to recover the pressure gradient ∇p.
3. Calculate the gas velocity $\mathbf{u}_G$ from the slip relation (28).
4. Solve the continuity equation (30) for the gas phase.
5. Convert the density $\tilde{\rho}_G$ into the gas holdup $\alpha = \frac{\tilde{\rho}_G}{p} RT$.
6. Solve the coupled equations of the $k - \varepsilon$ model (26)–(27).
7. Recalculate the turbulent eddy viscosity $\nu_T = C_\mu \frac{k^2}{\varepsilon}$.
8. Insert $\nu_{\text{eff}} = \nu + \nu_T$ and α into (25), evaluate the residual.
9. If the convergence criteria are satisfied, then exit the outer iteration loop and proceed to the next time step.
10. Otherwise increment the iteration counter 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. This provides an efficient treatment of physical processes occurring simultaneously but on entirely different time scales. 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 incompressibility

System of equations (25) for the velocity and pressure of the weakly compressible gas-liquid mixture is solved by a simple discrete projection scheme which belongs to the family of Multilevel Pressure Schur Complement (MPSC) techniques developed by Turek [39],[40] and implemented in the software package FEATFLOW [41] (see *http://www.featflow.de*). The fully discretized Navier-Stokes equations represent a saddle point problem, in which the pressure acts as the Lagrange multiplier for the incompressibility constraint

$$\begin{bmatrix} S & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \tag{31}$$

The velocity can be formally written as $\mathbf{u} = S^{-1}(\mathbf{f} - \Delta t B p)$ and plugged into the discrete continuity equation $B^T \mathbf{u} = 0$ to derive the missing equation for the pressure

$$B^T S^{-1} B p = \Delta t^{-1} B^T S^{-1} \mathbf{f}. \tag{32}$$

This discrete problem can be solved by the preconditioned Richardson iteration [39]

$$p^{(l+1)} = p^{(l)} + [B^T M_L^{-1} B]^{-1} \Delta t^{-1} B^T S^{-1} (\mathbf{f} - \Delta t B p^{(l)}), \tag{33}$$

where l is the outer iteration counter and M_L is the lumped mass matrix which proves to be a reasonable approximation to the evolution operator S at high Reynolds numbers.

Each Pressure Schur Complement iteration can be interpreted and implemented as a projection cycle which consists of the following algorithmic steps:

1. Compute the velocity $\tilde{\mathbf{u}}$ from the Burgers equation

$$S\tilde{\mathbf{u}} = \mathbf{f} - \Delta t B p^{(l)}$$

2. Solve the discrete 'Pressure-Poisson' problem

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

3. Correct the pressure and the intermediate velocity

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

In a nutshell, the right-hand side of the momentum equation is assembled using the old pressure iterate and the resulting intermediate velocity $\tilde{\mathbf{u}}$ is projected onto the subspace of divergence-free functions so as to satisfy the condition $B^T \mathbf{u}^{(l+1)} = 0$. The Pressure Schur Complement approach constitutes a very general framework which unites coupled solution techniques and classical projection schemes, the degree of coupling being determined by the number of outer iterations. For details the interested reader is referred to [21],[40].

3.2 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 consequence, negative gas holdups 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 cause an abnormal termination of the simulation run. 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 [3]. The state-of-the-art generalization proposed by Zalesak [44] 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.* [25]. In a series of recent publications, we refined the FEM-FCT methodology and extended it to implicit time stepping [15],[16],[17]. Another class of high-resolution methods which has enjoyed an increasing popularity in CFD was developed by Harten [10]. His *total variation diminishing* (TVD) schemes prove very robust and rest on a firm mathematical basis. At the same time, it has been largely unclear how to apply them in the finite element context. In the scarce publications on that subject, TVD-like artificial viscosities were designed using an *ad hoc* reconstruction of local three-point stencils [26]. The use of P_1 elements was essential to the derivation of the underlying edge-based data structure.

Below we outline an algebraic approach to the design of flux limiters which combines the advantages of classical FCT and TVD schemes and carries over to multidimensions. In what follows, we restrict ourselves to the case of lumped-mass Galerkin discretizations, which makes the algorithm easier to explain and reduces the cost of a three-dimensional simulation. The treatment of the consistent mass matrix is addressed in [22]. For a detailed presentation of the underlying *algebraic flux correction* (AFC) paradigm the interested reader is referred to the original pubications [17],[18],[21].

3.2.1 Discrete upwinding

As a model problem, consider the multidimensional continuity equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \tag{34}$$

discretized in space by the lumped-mass Galerkin FEM or a similar high-order scheme. The resulting ODE system for the vector of time-dependent nodal values reads

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Ku \qquad \text{or} \qquad m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} k_{ij}(u_j - u_i) + r_i u_i, \quad r_i = \sum_j k_{ij}, \tag{35}$$

where M_L is the lumped mass matrix and K 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* [11] in the absence of the term $r_i u_i$ which vanishes for divergence-free velocity fields and is responsible for a physical growth of local extrema otherwise. The operator K can be transformed into its nonoscillatory counterpart L by adding an artificial diffusion operator D designed so as to get rid of negative coefficients [15]

$$L = K + D, \qquad d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}, \qquad d_{ii} = -\sum_{k \neq i} d_{ik}. \tag{36}$$

This modification is conservative, since diffusive terms can be represented as a sum of internodal fluxes: $(Du)_i = -\sum_{j \neq i} f_{ij}$, where $f_{ij} = d_{ij}(u_i - u_j),\ f_{ji} = -f_{ij}$. These skew-symmetric fluxes are associated with edges of the sparsity graph for the global matrix. 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 the one located **upwind**.

3.2.2 Algebraic flux correction

In light of the above, the high- and low-order discretizations are related by the formula

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j\neq i} l_{ij}(u_j - u_i) + r_i u_i + \sum_{j\neq i} \alpha_{ij} f_{ij}, \qquad 0 \leq \alpha_{ij} \leq 1. \tag{37}$$

Note that the original high-order method is recovered for $\alpha_{ij} \equiv 1$, whereas the low-order one corresponds to $\alpha_{ij} \equiv 0$. The flux limiter should choose the correction factors α_{ij} automatically so as to guarantee that the resulting semi-discrete scheme remains positivity-preserving, at least for sufficiently small time steps or in the steady-state limit.

For an upwind-biased algorithm of TVD type, the raw antidiffusive flux may need to be 'prelimited' so as to satisfy the positivity constraint for the downwind node [22]. Due to the above orientation convention, the corresponding *target flux* is given by

$$f_{ij} := \min\{d_{ij}, l_{ji}\}(u_i - u_j), \qquad f_{ji} := -f_{ij}. \tag{38}$$

Following Zalesak [44], let us distinguish between the positive and negative fluxes and limit them separately node-by-node. In order to satisfy the positivity condition for row i of the discrete transport operator, suitable upper/lower bounds $Q_i^{\pm}$ should be imposed on the sum of antidiffusive fluxes denoted by $P_i^{\pm}$. In general, these auxiliary quantities are supposed to admit the following representation

$$P_i^{\pm} = \sum_{j\neq i} p_{ij} \min_{\max}\{0, u_j - u_i\}, \qquad Q_i^{\pm} = \sum_{j\neq i} q_{ij} \min_{\max}\{0, u_j - u_i\}, \tag{39}$$

where $p_{ij} \leq 0$ and $q_{ij} \geq 0$, $\forall j \neq i$. Recall that the 'prelimited' flux f_{ij} from a downwind node j into an upwind node i is given by (38) and poses no hazard to the positivity of the off-diagonal coefficient l_{ji}. Thus, the contribution of edge $\vec{ij}$ to the sum $P_i^{\pm}$ corresponds to $p_{ij} := -\min\{d_{ij}, l_{ji}\}$ and $p_{ji} := 0$. In order to construct the upper/lower bounds $Q_i^{\pm}$ so that they become less restrictive for small time steps, we take $q_{ij} := m_{ij}/\Delta t + l_{ij}$, where m_{ij} and l_{ij} are the coefficients of the consistent mass matrix and of the low-order operator, respectively. Finally, the correction factors α_{ij} are defined as follows:

$$\alpha_{ij} = \begin{cases} R_i^+ & \text{if} \quad f_{ij} > 0, \\ R_i^- & \text{if} \quad f_{ij} < 0, \end{cases} \qquad \text{where} \quad R_i^{\pm} = \min\{1, Q_i^{\pm}/P_i^{\pm}\}. \tag{40}$$

The discretization in time can be performed by the standard θ-scheme. The resulting nonlinear algebraic systems are solved iteratively as explained in [17],[18],[21].

3.3 Implementation of the $k - \varepsilon$ model

Transport equations (26)–(27) 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.

3.3.1 Positivity-preserving linearization

As already mentioned above, the coefficients of system (26)–(27) 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 high-resolution scheme 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 [28],[29] 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 [24]

$$\frac{\partial k}{\partial t} + \nabla \cdot (k\mathbf{u} - d_k \nabla k) + \gamma k = P_k + S_k, \tag{41}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon\mathbf{u} - d_\varepsilon \nabla \varepsilon) + C_2 \gamma \varepsilon = \gamma(C_1 P_k + C_\varepsilon S_k), \tag{42}$$

where the parameter $\gamma = \frac{\varepsilon}{k}$ is proportional to the specific dissipation rate ($\gamma = C_\mu \omega$). The effective diffusion coefficients are given by $d_k = \nu + \frac{\nu_T}{\sigma_k}$ and $d_\varepsilon = \nu + \frac{\nu_T}{\sigma_\varepsilon}$. By definition, the source terms $P_k = \frac{\nu_T}{2}|\nabla\mathbf{u} + (\nabla\mathbf{u})^T|^2$ and $\gamma C_1 P_k = C_1 \frac{c_\mu k}{2}|\nabla\mathbf{u} + (\nabla\mathbf{u})^T|^2$ are nonnegative. Furthermore, the parameters ν_T and γ must also be nonnegative for the solution of the above equations to be well-behaved [6]. 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.* [24] 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 high-resolution finite element schemes to equations (41) and (42), we obtain two nonlinear algebraic systems of the form

$$A(u^{(l+1)})u^{(l+1)} = B(u^{(l)})u^{(l)} + q^{(k)}, \qquad l = 0, 1, 2, \ldots \tag{43}$$

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 A and B is as follows:

$$A(u) = M_L - \theta\Delta t(K^*(u) + T), \qquad B(u) = M_L + (1-\theta)\Delta t(K^*(u) + T), \tag{44}$$

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. Obviously, 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 the matrix A is performed by solving a sequence of linear subproblems within a defect correction loop preconditioned by the monotone low-order operator. For the mathematical background and algorithmic details the interested reader is referred to [15],[16],[17],[18].

3.3.2 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 associated eddy viscosity $\nu_T^{(l+1)}$ is bounded from below by a small fraction of the laminar viscosity $0 < \nu_{\min} \leq \nu$ and from above by $\nu_{\max} = l_{\max}\sqrt{k}$, where $l_{\max}$ is the maximum admissible mixing length (the size of the largest eddies, e.g., the width of the domain). Specifically, we define the limited mixing length l_* as

$$l_* = \begin{cases} C_\mu \frac{k^{3/2}}{\varepsilon} & \text{if} \quad C_\mu k^{3/2} < \varepsilon l_{\max} \\ l_{\max} & \text{otherwise} \end{cases} \tag{45}$$

and calculate the turbulent eddy viscosity ν_T from the formula

$$\nu_T = \max\{\nu_{\min}, l_*\sqrt{k}\}. \tag{46}$$

The resulting value of ν_T is used to update the linearization parameter

$$\gamma = C_\mu \frac{k}{\nu_T}. \tag{47}$$

In the case of a TVD-like method, the positivity proof is only valid for the converged solution to (43) 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 (45)–(47) 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.

3.3.3 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. 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 ($t \leq t_*$), a constant effective viscosity $\nu_0 = \mathcal{O}(\nu)$ 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

$$k_0 = \left(\frac{\nu_0}{l_0}\right)^2, \qquad \varepsilon_0 = C_\mu \frac{k_0^{3/2}}{l_0} \qquad \text{at} \quad t \leq t_*. \tag{48}$$

Alternatively, the initial values of k and ε can be estimated by means of a zero-equation (mixing length) turbulence model or computed using an extension of the inflow or wall boundary conditions (see below) into the interior of the computational domain.

3.3.4 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:

$$\mathbf{u} = \mathbf{g}, \qquad k = c_{bc}|\mathbf{u}|^2, \qquad \varepsilon = C_\mu \frac{k^{3/2}}{l_0} \qquad \text{on} \quad \Gamma_{\text{in}}, \tag{49}$$

where $c_{bc} \in [0.003, 0.01]$ is an empirical constant [6] 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 ('do-nothing') boundary condition:

$$\mathbf{n} \cdot \mathcal{D}(\mathbf{u}) = \mathbf{0}, \qquad \mathbf{n} \cdot \nabla k = 0, \qquad \mathbf{n} \cdot \nabla \varepsilon = 0 \qquad \text{on} \quad \Gamma_{\text{out}}. \tag{50}$$

In the finite element framework, the numerical treatment of inflow and outflow boundary conditions does not present any difficulty. The prescribed gas inflow rate / superficial gas velocity can also be readily implemented as a natural boundary condition after integration by parts in the weak form of the gas phase continuity equation, see [19].

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 ('free slip') boundary condition

$$\mathbf{n} \cdot \mathbf{u} = 0 \qquad \text{on} \quad \Gamma_w \tag{51}$$

is nontrivial if the boundary of the computational domain is not aligned with the axes of the Cartesian coordinate system. In this case, condition (51) 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 [40] 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 [21]. 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:

$$a_{ij}^{(m)} := 0, \quad \forall j \neq i, \qquad \mathbf{u}_i^* := \mathbf{u}_i^{(m)} + \mathbf{r}_i^{(m)}/a_{ii}^{(m)} \qquad \text{for} \quad \mathbf{x}_i \in \Gamma_w. \tag{52}$$

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

$$\mathbf{u}_i^{(m)} := \mathbf{u}_i^* - (\mathbf{n}_i \cdot \mathbf{u}_i^*)\mathbf{n}_i, \qquad \mathbf{r}_i^{(m)} := 0 \qquad \text{for} \quad \mathbf{x}_i \in \Gamma_w. \tag{53}$$

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:

$$a_{ij}^{(m)} := 0, \quad \forall j \neq i, \qquad \mathbf{r}_i^{(m)} := \mathbf{r}_i^{(m)} - (\mathbf{n}_i \cdot \mathbf{r}_i^{(m)})\mathbf{n}_i \qquad \text{for} \quad \mathbf{x}_i \in \Gamma_w. \tag{54}$$

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.

3.3.5 Wall functions

To complete the problem statement we still need to prescribe the tangential stress as well as the boundary values of k and 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 where [27],[28],[29]

$$\mathbf{n}\cdot\mathcal{D}(\mathbf{u})\cdot\mathbf{t} = -\frac{u_\tau^2}{\nu_T}\frac{\mathbf{u}}{|\mathbf{u}|}, \qquad k = \frac{u_\tau^2}{\sqrt{C_\mu}}, \qquad \varepsilon = \frac{u_\tau^3}{\kappa\delta} \qquad \text{on} \quad \Gamma_\delta. \tag{55}$$

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

$$g(u_\tau) = |\mathbf{u}| - u_\tau\left(\frac{1}{\kappa}\log y^+ + \beta\right) = 0 \tag{56}$$

in the *logarithmic layer*, where the local Reynolds number $y^+ = \frac{u_\tau\delta}{\nu}$ is in the range $11.06 \le y^+ \le 300$. The empirical constant β equals 5.2 for smooth walls. Note that $\mathbf{u}$ is the tangential velocity as long as condition (51) is imposed on the internal boundary Γ_δ.

Strictly speaking, a boundary layer of width δ should be removed from the computational domain Ω. However, it is supposed to be very thin, so that the equations can be solved in the whole domain Ω with wall functions prescribed on the boundary part Γ_w rather than on Γ_δ. Equation (56) can be solved iteratively by Newton's method [27]:

$$u_\tau^{l+1} = u_\tau^l - \frac{g(u_\tau^l)}{g'(u_\tau^l)} = u_\tau^l + \frac{|\mathbf{u}| - u_\tau f(u_\tau^l)}{1/\kappa + f(u_\tau^l)}, \qquad l = 0, 1, 2, \ldots \tag{57}$$

where the auxiliary function f is given by

$$f(u_\tau) = \frac{1}{\kappa}\log y_*^+ + \beta, \qquad y_*^+ = \max\left\{1.0, y^+\right\}.$$

The friction velocity is initialized by $u_\tau^0 = \sqrt{\frac{\nu|\mathbf{u}|}{\delta}}$ and no iterations are performed if $y^+ = \frac{u_\tau^0\delta}{\nu} < 11.06$. In other words, $u_\tau = u_\tau^0$ in the viscous sublayer. Moreover, in each Newton iteration the lower bound of 1.0 is imposed on $y^+ = \frac{u_\tau\delta}{\nu}$ to prevent its logarithm from becoming negative. Due to (55) the boundary value of the eddy viscosity is given by $\nu_T = c_\mu k^2/\varepsilon = \kappa u_\tau\delta = \kappa y_*^+\nu$, which yields a natural lower bound $\nu_{\min} = \kappa\nu$.

The friction velocity u_τ is plugged into (55) 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 (25) gives rise to a surface integral over the internal boundary Γ_δ which contains the prescribed traction:

$$\int_{\Gamma_\delta}\nu_T(\mathbf{n}\cdot\mathcal{D}(\mathbf{u})\cdot\mathbf{t})\cdot\mathbf{v}\,ds = -\int_{\Gamma_\delta}u_\tau^2\frac{\mathbf{u}}{|\mathbf{u}|}\cdot\mathbf{v}\,ds. \tag{58}$$

The free slip condition (51) overrides the normal stress, and Dirichlet boundary conditions for k and ε are imposed in the strong sense. For further details regarding the implementation of wall laws the reader is referred to [27],[28],[29].

3.4 Underrelaxation for outer iterations

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

- the main *n-loop* for the global time-stepping
- the outermost *k-loop* for the coupled system (25)–(30)
- the outer *l-loop* for the Pressure Schur Complement equation (33)
- the outer *l-loop* for equations (26)–(27) of the $k-\varepsilon$ model
- embedded *m-loops* for iterative flux/defect correction

For each time step (one $n-$loop iteration), equations (25)–(30) 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 to be solved (see Fig. 1).

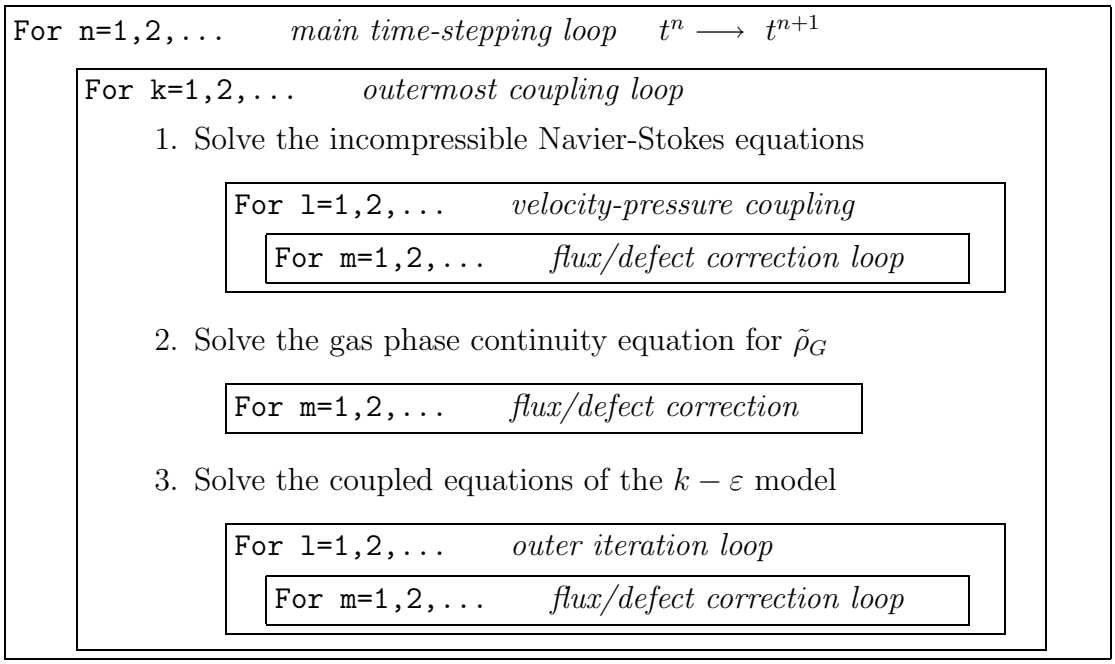

Figure 1. Nested iteration strategy.

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:

$$u^{(m+1)} := u^{(m)} + \omega^{(m)}(u^{(m+1)} - u^{(m)}) \qquad \text{where} \quad 0 \leq \omega^{(m)} \leq 1. \tag{59}$$

The damping factor $\omega^{(m)}$ may be chosen adaptively so as to accelerate convergence and minimize the error in a certain norm [40]. 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 α 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 [9],[30]:

$$a_{ii}^{(m)} := a_{ii}^{(m)}/\omega^{(m)}, \qquad \text{where} \quad 0 \leq \omega^{(m)} \leq 1. \tag{60}$$

The scaling of the diagonal entries does not affect the converged solution and proves more robust than *explicit underrelaxation* (59). In fact, no underrelaxation whatsoever is needed for moderate time steps which are typically used in dynamic simulations.

3.5 Linear solvers and time step

Finally, let us briefly discuss the choice of the linear solver and of the time discretization. In general, 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 expense 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 in conjunction with an appropriate renumbering scheme.

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 [43]. Choosing the local gas holdup α to be the indicator variable, one obtains the following algorithm:

1. Monitor the relative changes of the gas holdup distribution

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

2. If $e_n > \delta$ reject the solution and repeat the time step 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 so that

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

4 Numerical examples

The developed finite element software builds on the FEATFLOW package [41] and on the underlying FEAT libraries [42]. Algebraic flux correction was implemented as a black-box postprocessing tool for the matrix assembly routine. The fully multidimensional flux limiter introduced in Section 3.2.2. proved its worth for both multilinear and the nonconforming $\tilde{Q}_1$ finite elements. The three-dimensional simulation results presented in this section illustrate the utility of the drift-flux model and of the proposed numerical algorithm. In the present paper, no mass transfer or chemical reactions are considered. We refer to [14],[19],[20] for a numerical study of absorption/reaction phenomena.

4.1 Cylindrical bubble column

The first numerical example was computed using the laminar version of the presented drift-flux model. It deals with the startup of a locally aerated cylindrical bubble column. The gas is injected at the center into the initially quiescent liquid. The buoyancy induced by the ascending bubbles results in a circulation of liquid with upflow in the middle. The evolution of the gas holdup distribution during the laminar initial phase is shown in Fig. 2 along with fragments of the employed computational mesh. Similar mushroom-like shapes of the bubble swarm were observed previously in 2D simulations [13],[23],[38].

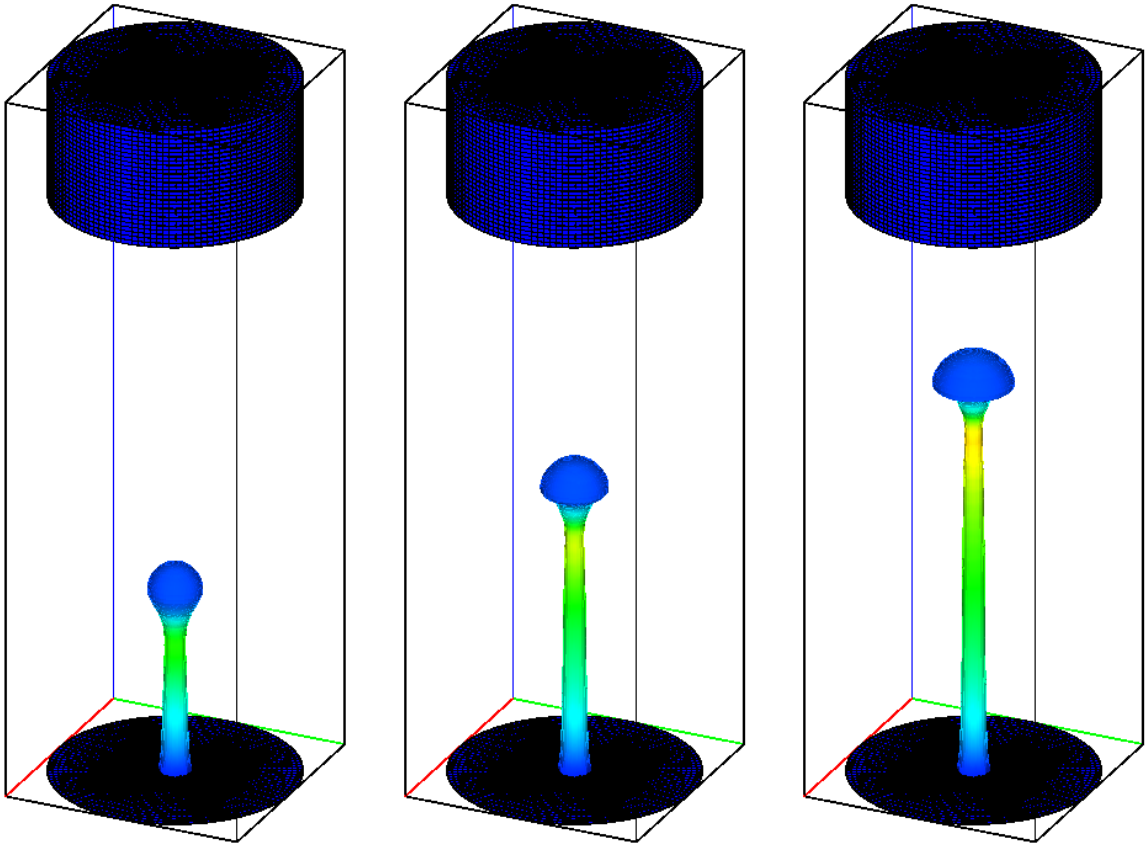

Figure 2. Startup of a locally aerated cylindrical bubble column.

The use of a nonuniform finite element mesh provides an accurate representation of the boundary and illustrates the flexibility of the developed three-dimensional simulation tools. Unfortunately, detailed measurements of the local gas holdup and velocity profiles are difficult to perform in cylindrical bubble columns, so there are hardly any experimental data to compare with. This is why, turbulence effects will be investigated for two benchmark configurations in relatively simple rectangular geometries, although arbitrarily complex domains can be handled in the framework of unstructured grid FEM.

4.2 Flat bubble column

A classical test problem for gas-liquid flow solvers is the locally aerated 'flat' bubble column which was studied both experimentally and numerically by Sokolichin *et al.* [1],[2],[34],[36]. The computational domain $\Omega = (0, 0.5) \times (0, 1.5) \times (0, 0.08)$ is discretized using a Cartesian mesh of just 6,912 hexahedral finite elements, which corresponds to 9,125 vertices and 22,848 degrees of freedom for each velocity component. The circular gas sparger has a diameter of 6 cm and is centered at the point $(0.15, 0.00, 0.04)$. The superficial gas velocity, i.e., the volume of gas fed per unit time per unit cross-section area, equals 0.66 mm/s which corresponds to the volumetric inflow rate of 1.6 l/min.

The second-order accurate Crank-Nicolson time-stepping with $\Delta t = 0.01$ was employed to capture the transient behavior of the turbulent bubbly flow. The snapshots of the gas holdup distribution displayed in Fig. 3 give an insight into the underlying flow pattern. A rather large and almost stationary vortex, which is formed to the right of the gas sparger, invariably pushes the incoming bubbles toward the left wall. As the bubbles rise, they are being spread by turbulent dispersion and convected by a cascade of moving vortices. The resulting turbulent flow structure gives rise to a meandering motion of the

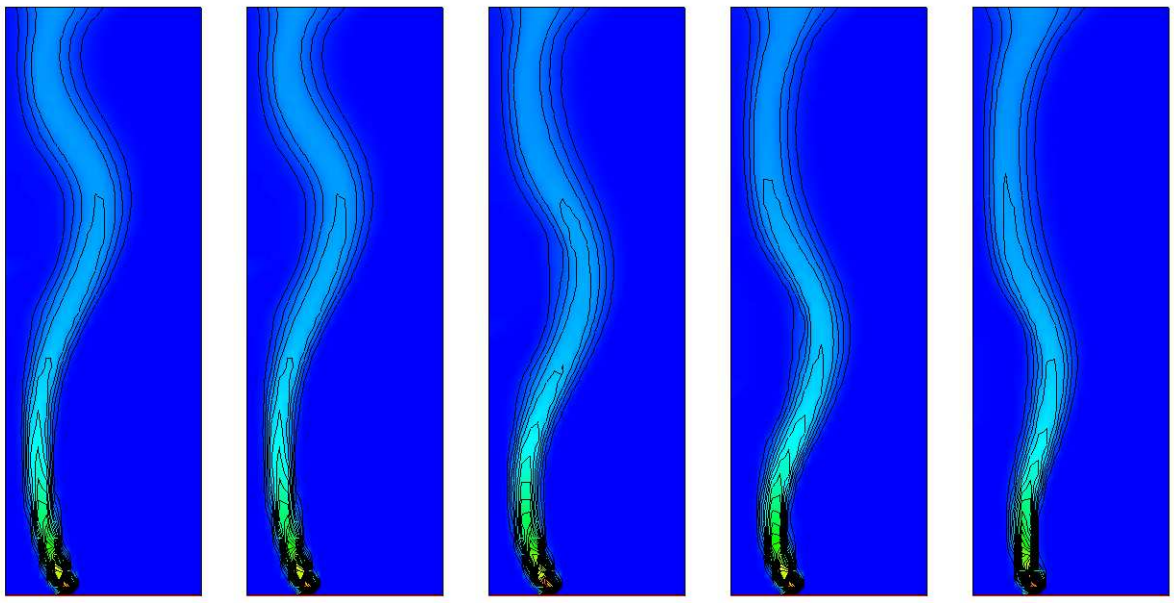

Figure 3. Gas holdup distribution in the xy-plane at 5 different times (3D simulation).

bubble swarm in the upper part of the column as observed in [1]. Even though the standard $k-\varepsilon$ model works reasonably well for this particular test problem, other applications may call for the inclusion of extra terms due to the bubble-induced turbulence [13]. As of now, turbulence modelling for bubbly flows remains a largely unexplored research area, which makes it hard to make reliable quantitative predictions in general.

Furthermore, this test problem reveals that the employed nonconforming $\tilde{Q}_1 Q_0$ finite element pair gives rise to nonnegligible approximation errors in the case of dynamic buoyancy-driven flows. In fact, the numerical solution tends to become stationary in the long run, at least on coarse meshes. This can be attributed to the fact that just two degrees of freedom per face are available for evaluation of the pressure gradient in the momentum equations. Even a constant gravity force induces spurious currents, because the hydrostatic pressure cannot be recovered. A remedy to this problem is yet to be found.

4.3 Airlift loop reactor

The last configuration to be considered in the present paper is the prototypical airlift loop reactor introduced in [36]. It was constructed from a flat bubble column with dimensions $(0, 0.5) \times (0, 1.90) \times (0, 0.08)$ by insertion of a rectangular internal obstacle, see Fig. 4.

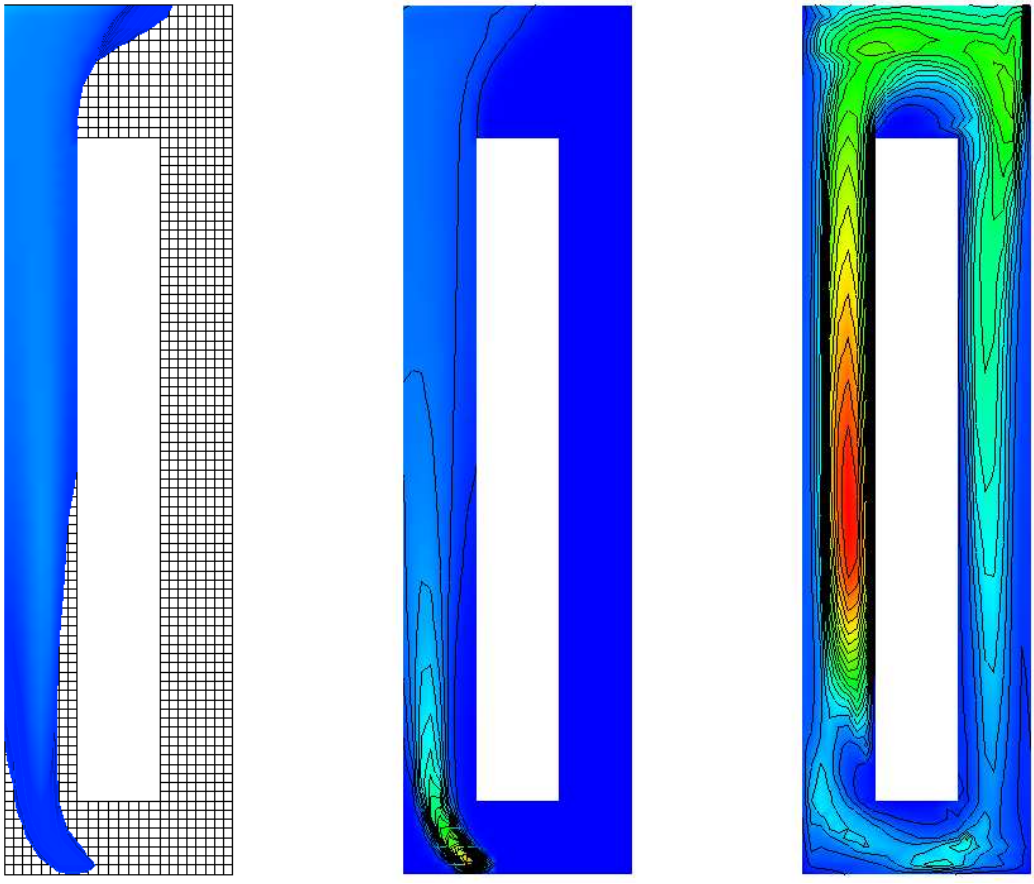

Figure 4. Steady circulation pattern in an airlift loop reactor.

The block-structured computational mesh for this three-dimensional simulation consists of 13,056 cells with 16,452 vertices and 42,368 degrees of freedom for each velocity component. The location and diameter of the circular gas sparger are the same as in the previous example, and the gas inflow rate equals 1.5 *l/min*. As the bubbles reach the upper surface, most of them escape, while the liquid is diverted into the gas-free downcomer forming a closed loop. Eventually, the two-phase flow reaches a steady-state circulation pattern. The stationary solution depicted in Fig. 4 shows the gas holdup distribution as the isovolume in which $\alpha \geq 0.2\%$ (left diagram) as well as the isolines of the gas holdup and eddy viscosity in the middle cross section (middle and right diagram, respectively). These simulation results agree well with those presented by Sokolichin [36].

5 Conclusions

A simplified drift-flux model for buoyancy-driven bubbly was derived using an analog of the Boussinesq approximation as proposed in [35]. Turbulence effects were taken into account in the framework of a (generalized) $k-\epsilon$ turbulence model with logarithmic wall functions applied in the vicinity of solid walls. The resulting system of partial differential equations was solved by an unstructured grid finite element method. The discretization of the troublesome convective terms was performed by a nonlinear positivity-preserving scheme equipped with a fully multidimensional flux limiter. Nested iterations were used to provide the coupling of model equations, get rid of nonlinearities and solve the linear systems. Some salient features of the numerical algorithm were discussed in detail.

The numerical examples presented in this paper indicate that the drift-flux model based on the Boussinesq approximation constitutes a viable approach to simulation of buoyancy-driven bubbly flows at relatively low superficial gas velocities. The available computing power and numerical methods are already sufficient for large-scale computations in three dimensions. However, the utility of the presented simulation tools for a quantitative analysis of turbulent bubbly flows is currently limited by the uncertainties connected to the two-phase flow modelling, especially due to the lack of reliable models for the bubble-induced turbulence and interphase momentum transfer. Recent developments in the field of Large Eddy Simulation (LES) indicate that it can shed some light on the nature of turbulence in gas-liquid systems [5]. Indeed, the bubble size defines a natural filter width, while high-resolution schemes based on algebraic flux correction provide an effective mechanism for implicit subgrid scale modeling in the MILES framework [4],[21]. It is hoped that further advances in CFD research will make it possible to reduce the modelling errors and enhance the credibility of simulation results.

Acknowledgements

The authors would like to thank Dr. A. Sokolichin and Dr. A. Lapin (University of Stuttgart, Germany) for many fruitful discussions and valuable advice.

References

[1] S. Becker, A. Sokolichin and G. Eigenberger, Gas-liquid flow in bubble columns and loop reactors: Part II. Comparison of detailed experiments and flow simulations. *Chem. Eng. Sci.* **49** (1994), no. 24B, 5747–5762.

[2] O. Borchers, C. Busch, A. Sokolichin, and G. Eigenberger, Applicability of the standard turbulence model to the dynamic simulation of bubble columns: Part II. Comparison of detailed experiments and flow simulations. *Chem. Eng. Sci.* **54** (1999) 5927-5935.

[3] 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.

[4] J.P. Boris, F.F. Grinstein, E.S. Oran and R.J. Kolbe, New Insights into Large Eddy Simulation, *Fluid Dynamics Research*, **10** (1992) no. 4-6, 199-227.

[5] S. Bove, T. Solberg and B.H. Hjertager, Numerical Aspects of Bubble Column Simulations. *Int. J. of Chemical Reactor Engineering*, Vol. **2**: A1, 2004.

[6] 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.

[7] D. A. Drew, Mathematical modeling of two-phase flow. *Ann. Rev. Fluid Mech.* **15** (1983) 261-291.

[8] D. A. Drew and S. L. Passman, *Theory of Multicomponent Fluids.* Springer, 1999.

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

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

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

[12] I. Kataoka and A. Serizawa, Basic equations of turbulence in gas-liquid two-phase flow. *Int. J. Multiphase Flow* **15** (1989) 843-855.

[13] D. Kuzmin, *Numerical Simulation of Reactive Bubbly Flows.* Dissertation, Jyväskylä Studies in Computing 2, University of Jyväskylä, 1999.

[14] 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.

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

[16] D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. Int. J. Numer. Meth. Fluids 42 (2003) 265-295.

[17] D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. *Computer Meth. Appl. Mech. Engrg.* **193** (2004) 4915-4946.

[18] D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131-158.

[19] D. Kuzmin and S. Turek, Finite element discretization and iterative solution techniques for multiphase flows in gas-liquid reactors. In: M. Krizek *et al.* (eds), *Conjugate Gradient Algorithms and Finite Element Methods*. Springer, 2004, 297-324.

[20] D. Kuzmin and S. Turek, Numerical simulation of turbulent bubbly flows. In: G.P. Celata et al. (eds) *Proceedings of the Third International Symposium on Two-Phase Flow Modeling and Experimentation.* (Pisa, Italy, September 22-25, 2004.)

[21] D. Kuzmin, R. Löhner and S. Turek (Eds.), *Flux-Corrected Transport*. Springer, 2005.

[22] D. Kuzmin, On the design of flux limiters for finite element discretizations with a consistent mass matrix. Technical report **283**, University of Dortmund, 2005. Submitted to *J. Comput. Phys.*

[23] A. Lapin and A. Lübbert, Numerical simulation of the dynamics of two-phase gas-liquid flows in bubble columns. *Chem. Eng. Sci.* **49** (1994), no. 21, 3661–3674.

[24] 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.

[25] 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.

[26] P. R. M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction.* PhD thesis, University of Wales, Swansea, 1994.

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

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

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

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

[31] D. Pfleger, S. Gomes, N. Gilbert and H.-G. Wagner, Hydrodynamic simulations of laboratory scale bubble columns - Fundamental studies of the Eulerian-Eulerian modelling approach. *Chem. Eng. Sci.* **54** (1999) 5091–5098.

[32] R. Rannacher and S. Turek, A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs* **8** (1992), no. 2, 97-111.

[33] M. P. Schwarz and W. J. Turner, Applicability of the standard k–ϵ turbulence model to gas-stirred baths. *Appl. Math. Modelling* **12** (1988) 273–279.

[34] A. Sokolichin and G. Eigenberger, Modellierung und effiziente numerische Simulation von Gas-Flüssigkeits-Reaktoren mit Blasenströmungen nach dem Euler-Euler-Konzept. DFG technical report. University of Stuttgart, 1995-1997. Available at *http://pcvt12.verfahrenstechnik.uni-stuttgart.de/alex/icvt2.html*.

[35] 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.

[36] A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen*. Habilitationsschrift, Institut für chemische Verfahrenstechnik, Universität Stuttgart, 2002.

[37] A. Sokolichin, and G. Eigenberger, Applicability of the standard turbulence model to the dynamic simulation of bubble columns: Part I. Detailed numerical simulations. *Chem. Eng. Sci.* **54** (1999) 2273-2284.

[38] A. Sokolichin, G. Eigenberger, A. Lapin and A. Lübbert, Dynamic numerical simulation of gas-liquid two-phase flows: Euler-Euler versus Euler-Lagrange. *Chem. Eng. Sci.* **52** (1997), no. 4, 611–626.

[39] S. Turek, On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach. *Comput. Methods Appl. Mech. Engrg.* **143** (1997) 271–288.

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

[41] 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*.

[42] S. Turek *et al.*, *FEAT2D/FEAT3D: Finite Element Analysis Tools.* User manual, University of Heidelberg, 1995. Available at *http://www.featflow.de*.

[43] 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.

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