INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **55**:537–564
Published online 28 February 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1468

Critical evaluation of CFD codes for interfacial simulation of bubble-train flow in a narrow channel

Furkan Özkan[1,2,3,*,†], Martin Wörner[2], Achim Wenka[3] and Hakan S. Soyhan[4]

[1] *University of Sakarya, Institute of Science and Technology, Sakarya, Turkey*
[2] *Forschungszentrum Karlsruhe, Institut für Reaktorsicherheit, Karlsruhe, Germany*
[3] *Forschungszentrum Karlsruhe, Institut für Mikroverfahrenstechnik, Karlsruhe, Germany*
[4] *Engineering Faculty, Mechanical Engineering Department, University of Sakarya, Sakarya, Turkey*

SUMMARY

Computational fluid dynamics (CFD) codes that are able to describe in detail the dynamic evolution of the deformable interface in gas–liquid or liquid–liquid flows may be a valuable tool to explore the potential of multi-fluid flow in narrow channels for process intensification. In the present paper, a computational exercise for co-current bubble-train flow in a square vertical mini-channel is performed to investigate the performance of well-known CFD codes for this type of flows. The computations are based on the volume-of-fluid method (VOF) where the transport equation for the liquid volumetric fraction is solved either by the methods involving a geometrical reconstruction of the interface or by the methods that use higher-order difference schemes instead. The codes contributing to the present code-to-code comparison are an in-house code and the commercial CFD packages CFX, FLUENT and STAR-CD. Results are presented for two basic cases. In the first one, the flow is driven by buoyancy only, while in the second case the flow is additionally forced by an external pressure gradient. The results of the code-to-code comparison show that only the VOF method with interface reconstruction leads to physically sound and consistent results, whereas the use of difference schemes for the volume fraction equation shows some deficiencies. Copyright © 2007 John Wiley & Sons, Ltd.

Received 5 October 2006; Revised 24 January 2007; Accepted 27 January 2007

KEY WORDS: micro-process engineering; code-to-code comparison; volume-of-fluid method; bubble-train flow; Taylor flow; square channel

[*] Correspondence to: Furkan Özkan, Forschungszentrum Karlsruhe, Institute for Micro Process Engineering, P.O. Box 3640, Karlsruhe 76021, Germany.

[†] E-mail: oezkan@imvt.fzk.de

Contract/grant sponsor: EU Erasmus program
Contract/grant sponsor: Institute of Science and Technology at Sakarya University

1. INTRODUCTION

A current trend in engineering and industrial applications such as automotive emissions and fuel processing, fuel cells in internal combustion engines, desulphurization of vapour gas oil fractions in the petrochemical industry, etc. is termed *process intensification*. According to Tsouris and Porcelli [1], process intensification refers to technologies that replace large, expensive, energy-intensive equipment or processes with the ones that are smaller, less costly and more efficient, or that combine multiple operations into fewer devices (or a single apparatus). One way towards process intensification is therefore by miniaturization [2–5]. It is expected that the potential for process intensification is in particular large for systems that involve multiple phases, such as the flow of a gas and a liquid or that of two immiscible liquids. The reason for this is that multi-fluid flows in small dimensions are associated with large specific interfacial areas, thin liquid films and short diffusion length which, all together, results in high mass transfer rates. Examples for devices where the potential of two-fluid flow in narrow channels has already been shown are micro-bubble columns [6, 7], micro-falling film reactors [6], micro-fluidic channel networks [8] and monolith reactors [9, 10]. The monolith reactor was developed for the cleaning of exhaust gases from combustion processes, both in cars and large power plants. There is, however, now an increasing interest to employ monolithic structures for new reactor applications in the chemical process and refining industries [11, 12].

The idea of process intensification is to provide optimal physical–chemical conditions for the respective system. This requires knowledge of the local hydrodynamic, thermal and mass transfer phenomena. However, it is very difficult to provide the required local and time-resolved information within both phases by experiments because of the small dimensions. An alternative way is therefore to use computational fluid dynamics (CFD) to get insight into the relevant transport phenomena. While, for single-phase flow, the methods available in commercial CFD codes such as CFX, FLUENT and STAR-CD have now reached a certain level of maturity, this is not valid to the same extent for two-phase flows where the information about the shape of the deformable interface is part of the solution. The reason for this lies to one part in the difficulties of numerical methods to deal with the discontinuous density and viscosity at the interface. Another difficulty lies in the numerical description of the surface tension force, which is singular at the interface. It is known for some time that the continuous surface force model of Brackbill *et al.* [13], that is now the standard surface tension force model in finite volume codes, may lead to the so-called artificial or spurious currents. This unphysical flow phenomenon caused by inadequate representation of the surface tension force may become in particular troublesome for miniaturized devices because the magnitude of the surface tension force increases when the length scale decreases. In the past, different methods have been developed for computation of two-fluid flows with deformable interfaces. Most widely used ones are the volume-of-fluid (VOF) method [14], the level-set method [15] and the front-tracking method [16]. While the level-set method and the front-tracking method are widely used in the academic field, in the codes CFX, FLUENT and STAR-CD only the VOF method is available. To foster the reliable application of commercial CFD codes for interfacial two-fluid flow applications in small channels, it is therefore useful to assess the capabilities of the VOF method in these CFD codes by test or benchmark problems. We note that it was the objective of the ECORA project to evaluate the capabilities of CFD software packages for reactor safety analysis [17]. However, naturally this project was devoted to flow in large dimensions and also the capabilities of the VOF method were not investigated.

DOI: 10.1002/fld

In the present paper, we perform test case computations for the co-current vertical bubble-train flow (BTF) in a square channel of 2 mm × 2 mm cross-section using different variants of the VOF method. We compare the results obtained for BTF by CFX, STAR-CD and FLUENT with results of direct numerical simulations obtained by an in-house computer code called TURBIT-VOF. In Section 2 of this paper, we give a description of the test case and of the different computer codes used in the present code-to-code comparison. In Section 3, we present the simulation results and provide a critical discussion. The paper is completed by conclusions to be presented in Section 4.

2. DESCRIPTION OF NUMERICAL TEST CASE

In this section, we first describe the BTF test case and then give some details about the physical and numerical model. Next, we give a short description of the different computer codes used for this code-to-code comparison and also give details of the numerical methods and parameters used as well as an overview on all the simulations performed within the present study. Since most commercial codes are still evolving, the results of the present code-to-code comparison are temporary in nature. Also, there is not a reference or 'truth' solution and the code-to-code comparison cannot be relied on for any sort of validation or verification. Nevertheless, code-to-code comparisons are useful in terms of identifying shortcomings in code capabilities.

2.1. *Bubble-train flow*

Bubble-train flow is a common flow pattern for gas–liquid flow in small channels. It consists of a regular sequence of bubbles of identical size and shape. The bubbles, which are separated by liquid slugs, fill almost the entire channel cross-section (Taylor bubbles) and travel with the same axial velocity. Therefore, BTF or Taylor flow is fully described by a unit cell, which consists of one bubble and one liquid slug. Thulasidas *et al.* [18] performed experiments of bubble-train in circular capillaries with 2 mm inner diameter and in a square capillary with a cross-section of 2 mm × 2 mm using air as gaseous and silicon oil as liquid phase. The relevant non-dimensional group for two-fluid flows in narrow channels is the capillary number $Ca \equiv \mu_\mathrm{L} U_\mathrm{B}/\sigma$, where U_B is the bubble velocity, μ_L is the viscosity of the continuous liquid phase and σ is the coefficient of surface tension. The relevance of the capillary number is due to the fact that it represents the ratio of the two dominant forces for two-fluid flow in small dimensions, namely viscous friction and surface tension. To cover a wide range of capillary numbers, Thulasidas *et al.* [18] performed their experiments using silicon oil of different viscosities.

2.2. *Numerical model*

Figure 1 shows a sketch of the basic computational set-up. The coordinate system is defined by taking y as axial direction and x and z as the two wall-normal directions. The gravity vector points in negative y-direction. At the four sidewalls of the channel, no slip conditions are applied. To set up a model for BTF, we follow the approach of Ghidersa *et al.* [19]; we consider one unit cell only and use periodic (or cyclic) boundary conditions in the streamwise direction. Thus, within the computational domain, there is only one bubble and the influence of the trailing and leading bubbles is taken into account by the periodic boundary conditions. In correspondence to the experiment of Thulasidas *et al.* [18], we consider a square mini-channel with a cross-section of $L_x \times L_z = 2\ \mathrm{mm} \times 2\ \mathrm{mm}$. For the length of the unit cell, we choose $L_y = 2$ mm, so that the

DOI: 10.1002/fld

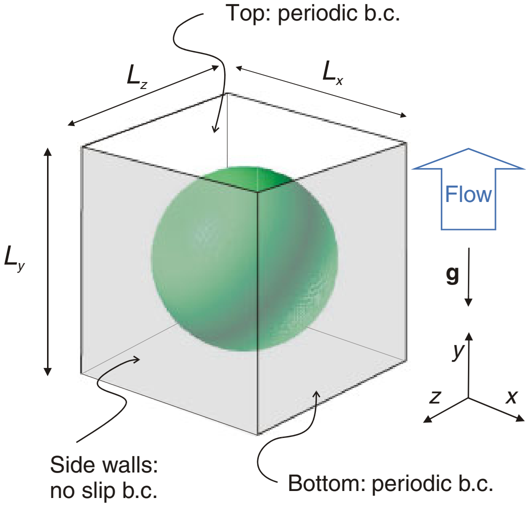

Figure 1. Sketch of coordinate system, computational domain, boundary conditions (b.c.) and initial bubble shape used in the simulations with TURBIT-VOF, STAR-CD and CFX. The numerical values of the dimensions are $L_x = L_y = L_z = 2$ mm.

computational domain is a cube. This choice results in rather short bubbles, in contrast to the experiments of Thulasidas *et al.* [18] where the bubble length is several times the channel width. The influence of the length of the unit cell has been investigated by Wörner *et al.* [20], where it covers eight distinct values in the range $2\,\text{mm} \leqslant L_y \leqslant 4\,\text{mm}$. In that paper, a good agreement of the numerical results with the experimental ones of Thulasidas *et al.* [18] has been obtained for the bubble velocity, phase relative velocity and bubble diameter only for cases where $L_y \geqslant 2.5$ mm. While longer unit cells are more realistic, the present study is nevertheless restricted to the case $L_y = 2$ mm for two reasons. The first one is to limit the computational costs, while the second is related to the length of the liquid slug. For the same overall gas volume fraction within the unit cell, small values of L_y lead to shorter liquid slugs and thus a smaller distance between consecutive bubbles. For any CFD code this situation is more severe, since any inaccuracy will immediately lead to bubble coalescence which then results not in BTF but in annular flow. The computational set-up just described is used for computations with the codes TURBIT-VOF, STAR-CD and CFX. It was, however, necessary to modify the set-up for the FLUENT code. The reason for this and the modifications made are explained in Section 2.3.4.

The computations in this paper are based on the VOF method which was originally developed by Hirt and Nichols [14]. The basic concept of the VOF method is the definition of a non-dimensional scalar quantity f (or α_L), which represents the fraction of the mesh cell volume occupied by the continuous phase, which is here the liquid phase. Thus, for $f = 1$, the mesh cell is entirely filled with liquid while for $f = 0$ it is entirely filled with gas. In a mesh cell which instantaneously contains a part of the interface, both phases coexist and it is $0 < f < 1$. Based on f, it is possible to define a mixture density, ρ_m, mixture viscosity, μ_m, and centre-of-mass velocity, $\mathbf{v}_\text{m}$,

$$\rho_\text{m} \equiv f\rho_\text{L} + (1 - f)\rho_\text{G} \tag{1}$$

$$\mu_\mathrm{m} \equiv f\mu_\mathrm{L} + (1-f)\mu_\mathrm{G} \tag{2}$$

$$\mathbf{v}_\mathrm{m} \equiv \frac{f\rho_\mathrm{L}\mathbf{v}_\mathrm{L} + (1-f)\rho_\mathrm{G}\mathbf{v}_\mathrm{G}}{\rho_\mathrm{m}} \tag{3}$$

Using the above definitions, the equations governing the motion of the liquid and the gas phase as well as the dynamic boundary condition at the interface can be combined into one single set of continuity and momentum equations which are valid in the entire two-fluid domain. For Newtonian fluids with constant physical properties, these read

$$\frac{\partial f}{\partial t} + \nabla\cdot f\mathbf{v}_\mathrm{m} = 0 \tag{4}$$

$$\nabla\cdot\mathbf{v}_\mathrm{m} = 0 \tag{5}$$

$$\frac{\partial}{\partial t}(\rho_\mathrm{m}\mathbf{v}_\mathrm{m}) + \nabla\cdot\rho_\mathrm{m}\mathbf{v}_\mathrm{m}\mathbf{v}_\mathrm{m} = -\nabla p + \nabla\cdot\mu_\mathrm{m}(\nabla\mathbf{v}_\mathrm{m} + (\nabla\mathbf{v}_\mathrm{m})^\mathrm{T}) + \rho_\mathrm{m}\mathbf{g} + \mathbf{f}_\sigma \tag{6}$$

Here, $\mathbf{g} = (0, -g, 0)^\mathrm{T}$ is the gravity vector, $g = 9.81\ \mathrm{m/s^2}$ is the gravitational acceleration and $\mathbf{f}_\sigma$ represents the surface tension force per unit volume. Note that in this form of the momentum equation it is already assumed that in interfacial mesh cells (i.e. for $0<f<1$) both phases obey the same pressure and share the same velocity field (homogeneous model).

Because the pressure p in Equation (6) is not periodic, we have to recast this equation in a form which is appropriate for a domain with periodic boundary conditions. For this purpose, we split the pressure as

$$\begin{aligned} p &\equiv P - \rho_\mathrm{L} g\hat{\mathbf{e}}_y\cdot\mathbf{x} + \left(\frac{\bar{p}|_{y=L_y} - \bar{p}|_{y=0} + \rho_\mathrm{L} g L_y}{L_y}\right)\hat{\mathbf{e}}_y\cdot\mathbf{x} = P + \rho_\mathrm{L}\mathbf{g}\cdot\mathbf{x} + \frac{\Delta\tilde{p}}{L_y}\hat{\mathbf{e}}_y\cdot\mathbf{x} \\ &= P + \rho_\mathrm{L}\mathbf{g}\cdot\mathbf{x} + \mathbf{f}_\mathrm{pd}\cdot\mathbf{x} \end{aligned} \tag{7}$$

Here, $\hat{\mathbf{e}}_y = (0, 1, 0)^\mathrm{T}$ is the unit vector in axial direction and $\Delta\tilde{p}$ represents the non-hydrostatic axial pressure drop across the distance L_y, while $\bar{p}|_{y=0}$ and $\bar{p}|_{y=L_y}$ represent the cross-section averaged pressure at axial position $y=0$ and $y=L_y$, respectively. Introducing definition (7) in Equation (6), we obtain the Navier–Stokes equation in the form

$$\frac{\partial}{\partial t}(\rho_\mathrm{m}\mathbf{v}_\mathrm{m}) + \nabla\cdot\rho_\mathrm{m}\mathbf{v}_\mathrm{m}\mathbf{v}_\mathrm{m} = -\nabla P + \nabla\cdot\mu_\mathrm{m}(\nabla\mathbf{v}_\mathrm{m} + (\nabla\mathbf{v}_\mathrm{m})^\mathrm{T}) + (\rho_\mathrm{m} - \rho_\mathrm{L})\mathbf{g} - \mathbf{f}_\mathrm{pd} + \mathbf{f}_\sigma \tag{8}$$

This equation can be used for a periodic domain since it is $\bar{P}|_{y=L_y} = \bar{P}|_{y=0}$ and P is periodic. In the present numerical study, two cases will be considered. In the first one, it is $\Delta\tilde{p} = 0\,\mathrm{Pa}$ and the only driving force is buoyancy by means of the term $(\rho_\mathrm{m} - \rho_\mathrm{L})\mathbf{g}$. In the second case, the bubble rises both due to buoyancy and due to an external pressure gradient of $\Delta\tilde{p} = -18\,\mathrm{Pa}$ which results in a body force of $-\Delta\tilde{p}/L_y = 18\,\mathrm{Pa}/0.002\,\mathrm{m} = 9000\,\mathrm{N/m^3}$.

As constant physical properties of the liquid phase (silicon oil), we use a density $\rho_\mathrm{L} = 957\mathrm{kg/m^3}$ and viscosity $\mu_\mathrm{L} = 0.048\,\mathrm{Pa\,s}$. The value of the coefficient of surface tension is $\sigma = 0.02218\,\mathrm{N/m}$. While these values correspond to the experiment of Thulasidas *et al.* [18], we increase for our numerical tests the gas density and viscosity of that experiment by a factor of 10, so that

DOI: 10.1002/fld

$\rho_G = 11.7\,\mathrm{kg/m^3}$ and $\mu_G = 1.85 \times 10^{-5}$ Pa s. This artificial increase of the gas density and viscosity is justified by results of a numerical study performed with the in-house code TURBIT-VOF where we compared the results to those obtained with $\rho_G = 1.17\,\mathrm{kg/m^3}$ and $\mu_G = 1.85 \times 10^{-6}$ Pa s. We found that the differences in bubble velocity, mean liquid velocity and in bubble shape are very small [20], while for the increased gas properties much larger time steps could be used in the explicit time integration scheme. This results in a considerable saving of CPU time.

The present fluid properties result in a magnitude of the buoyancy force of $|\rho_G - \rho_L|g = 9273.4\,\mathrm{N/m^3}$. This is the same order of magnitude as the body force due to the external pressure gradient. Both body forces act in the same direction, namely vertically upwards. However, the body force due to buoyancy acts only in the gas phase while the body force due to the external pressure gradient acts in the entire domain, i.e. both in the gas and in the liquid phase.

2.3. *Computer codes*

2.3.1. Computer code TURBIT-VOF. The in-house computer code TURBIT-VOF is based on a finite volume method and uses a staggered grid. The surface tension term is given by [21]

$$\mathbf{f}_\sigma = a_\mathrm{i}\sigma\kappa\hat{\mathbf{n}}_\mathrm{i} \quad \text{with } \kappa = -\nabla\cdot\hat{\mathbf{n}}_\mathrm{i} \tag{9}$$

Here, a_i is the volumetric interfacial area concentration, κ is twice the mean interface curvature and $\hat{\mathbf{n}}_\mathrm{i}$ is the unit normal vector to the interface pointing into the liquid. In TURBIT-VOF, the governing equations (4), (5) and (8) are solved in non-dimensional form, see [19]. A projection method is used to solve the momentum equation (8) while enforcing a divergence-free velocity field according to Equation (5). For time integration of Equation (8), an explicit third-order Runge–Kutta method is used. All spatial derivatives are approximated by second-order central differences.

To account for the interface evolution, in the VOF method with interface reconstruction, Equation (4) is not solved by a difference scheme but in a rather geometrical manner that involves two steps. In the first step, the interface orientation and location inside each mesh cell is reconstructed using the piecewise linear interface calculation (PLIC) method EPIRA (exact plane interface reconstruction algorithm) that locally approximates the interface in a mesh cell by a plane, see Sabisch *et al.* [22] for details. In a second step, the fluxes of liquid across the faces of the mesh cell are computed. This ensures that mass is conserved with high accuracy. TURBIT-VOF has been verified for single bubbles of different shapes in conventional channels [22] and has also been used for investigations of bubble swarms in a flat bubble column [23].

The present simulations are performed on one processor of a Siemens Fujitsu VPP5000 parallel vector computer with 8 GB RAM and a theoretical peak performance of 9.6 GFlops per processor. The typical CPU time per mesh cell and time step is about 70 μs.

2.3.2. Computer code STAR-CD. Together with CFX and FLUENT, the computer code STAR-CD of CD-adapco is one of the leading commercial CFD codes. All three software packages claim to be able to cover almost all the areas which are related to fluid dynamics phenomena and to solve problems coming both, from industry and academia. Like CFX and FLUENT, STAR-CD is based on a finite volume method and uses a co-located grid. When the present computational study was started, the actual version of STAR-CD was 3.26. In the VOF method implemented in that version, the *Compressive Interface Capturing Scheme for Arbitrary Meshes* (CICSAM) [24] is used for solution of the volume fraction equation (4) in order to maintain the interface sharp. The CICSAM scheme as implemented in version 3.26 of STAR-CD involves a restriction that the

DOI: 10.1002/fld

Courant number based on the local velocity and mesh cell width may not exceed a value of 0.3. While a certain number of time steps could be successfully computed by STAR-CD version 3.26 for the BTF set-up described above, eventually the Courant number limit was exceeded which resulted in a stop of the simulation. The time step at which this stop occurred was not always reproducible. Although a number of different numerical schemes and convergence parameters have been tried, eventually no converging simulation of BTF could be obtained with STAR-CD version 3.26, where the bubble has reached a constant terminal velocity.

During the course of the present research, version 4.0 of STAR-CD became available. The capabilities of the code to deal with free surface and interfacial applications have been substantially improved by the implementation of numerical methods from the COMET code, which is now also distributed by CD-adapco. With STAR-CD version 4.0, the convergence problems of version 3.26 have not been faced with and all simulations to be presented in this paper have been obtained by the VOF method of version 4.0. In the VOF method of STAR-CD, no interface reconstruction is available and Equation (4) can only be solved by difference schemes. Here, for the convective terms in the volume fraction equation (4), the *High-Resolution Interface Capturing* (HRIC) scheme [25] is used while the convective and diffusive terms in the momentum equation are discretized by the MARS scheme (blending factor 0.5). The velocity–pressure coupling is treated by the SIMPLE algorithm. For our BTF simulations, it was essential to increase the maximum number of iterations within the SIMPLE algorithm from the standard value 5 to a value of 50. For temporal discretization the first-order Euler implicit method is used which is the only available option in STAR-CD in combination with the VOF method. Since the implementation of the cyclic boundary conditions in STAR-CD allows for a direct specification of the pressure drop across the periodic domain, the simulation for the case corresponding to a pressure drop of $\Delta\tilde{p} = -18$ Pa have been realized by this option. Surface tension effects are accounted for *via* the continuum surface model. The magnitude of the surface tension force is given by [26]

$$|\mathbf{f}_\sigma| = -\sigma\nabla\cdot\left(\frac{\nabla\alpha_\mathrm{L}}{|\nabla\alpha_\mathrm{L}|}\right) \tag{10}$$

All STAR-CD simulations were performed in double precision in serial mode on a Linux Cluster where each node consists of two 2.2 GHz AMD Opteron 248 processors with 2 GB RAM and a theoretical peak performance of 4.4 GFlops per processor. The typical CPU time per mesh cell and time step is about 50–100 μs.

2.3.3. Computer code CFX. For the present code-to-code comparison, version 10.0 of ANSYS CFX is used. The CFX code is based on a finite volume method and uses a co-located grid. Formally, CFX does not offer the VOF method. For interfacial flows, where the distinct resolvable interface between both phases remains well defined, the homogeneous model can be used instead. A CFX simulation where the continuity equation of the homogeneous model is solved by a high-order difference scheme and where the surface tension force is encountered for in the momentum equation by the continuum surface force model can thus be considered equivalent to the difference scheme VOF method in STAR-CD. The surface tension force in CFX is, adopting the present nomenclature, given by [27]

$$\mathbf{f}_\sigma = \sigma\kappa\hat{\mathbf{n}}_\mathrm{i}|\nabla\alpha_\mathrm{L}| \quad \text{with } \kappa = -\nabla\cdot\hat{\mathbf{n}}_\mathrm{i} \tag{11}$$

For computation of the surface tension force in CFX, one of the three available options for the smoothing of the volume fraction field needs to be specified. These are ‘no smoothing’, ‘Laplacian

DOI: 10.1002/fld

smoothing' and 'volume-weighted smoothing'. In the user manual, the Laplacian smoothing is recommended so this option is used here. Another parameter related to the modelling of surface tension in CFX is the 'curvature under-relaxation factor'. The respective real value is in the range from 0 to 1. As for flows that are strongly driven by surface tension the use of a smaller value is recommended in the manual, a value of 0.25 is adopted here. In the CFX-free surface control panel furthermore a value of the 'interface compression level' has to be chosen. This integer parameter may take values between 0 and 2. In the CFX user manual, the use of level 2 is recommended in order to keep the interface sharp; therefore, this value is adopted here.

For the discretization of the momentum equation and the continuity equation, the high-resolution advection scheme is used. Time integration of the momentum equation is performed by the first-order backward-Euler scheme for almost all simulation in order to agree with STAR-CD conditions. To investigate the influence of the time integration scheme, in one simulation the second-order backward-Euler scheme is used instead. In all simulations, periodic boundary conditions are used. In order to account for the external pressure gradient, in the y-component of the momentum equation, a source term of $9000\,\mathrm{N/m^3}$ is specified.

The CFX simulations are performed on a Linux cluster consisting of Dual AMD Opteron 275 processors with 16 GB RAM. The code was run in parallel mode with up to six processors. Some simulations were run in double precision and some in single precision. For one case, the results of a single precision run have been compared to those of a double precision run and no notable differences have been found. For a CFX run, the typical CPU time per mesh cell and time step on one processor is about 8 ms. Thus, for the present problem CFX is by far the slowest code. For this reason, CFX simulations with a fine grid and a small time step width could not be advanced in time as much as it was done for the other codes.

2.3.4. Computer code FLUENT. From the three commercial CFD codes used in the present study, FLUENT is the only one which offers the VOF method with geometrical reconstruction of the interface. However, the code also provides the possibility to use difference schemes for solution of the volume fraction equation. FLUENT does not solve the volume fraction equation for the primary phase (i.e. α_L) but for the secondary phase, i.e. for α_G. Unfortunately, version 6.2.16 of FLUENT does not allow the combination of the VOF method with periodic boundary conditions. For this reason, some changes of the basic computational set-up had to be made.

Unlike the computations by the other codes, the FLUENT simulations are not performed in a fixed frame of reference, but in a frame of reference moving with the bubble. Such simulations with FLUENT have been recently performed by Taha and Cui [28]. Here, this frame of reference is realized in the code by downward moving walls, where the downward velocity of the walls, U_{wall}, is set according to the terminal bubble velocity, U_B, computed by TURBIT-VOF. At the top of the computational domain, inlet boundary conditions are specified for the liquid phase. The downward liquid inlet velocity is uniform over the inlet cross-section and is given by $U_B - U_{\mathrm{slug}}$. The value of the liquid slug velocity is again taken from the respective TURBIT-VOF simulation. Since the mean axial velocity in the liquid slug agrees with the total superficial velocity, it is computed as $U_{\mathrm{slug}} = \varepsilon U_B + (1-\varepsilon)U_L$. While the wall velocity U_{wall} is kept constant in all FLUENT computations, for the slug velocity U_{slug}, the respective TURBIT-VOF value is only taken as an initial guess and is then slightly adjusted so that in the FLUENT computations the bubbles stay almost at rest and steady-flow conditions are obtained.

A sketch of the computational set-up used in FLUENT is displayed in Figure 2. The size of the computational domain corresponds to five unit cells. However, only three equal-sized spherical

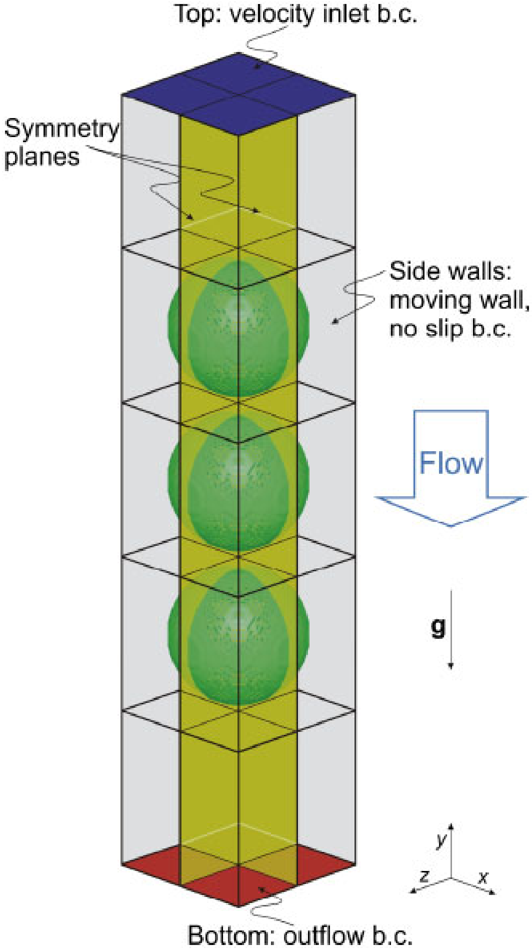

Figure 2. Sketch of computational domain, boundary conditions (b.c.) and initial bubble distribution used in the simulations with FLUENT. While in the figure the entire channel cross-section is displayed, only a quarter of it is actually solved for by using symmetry planes. The size of the computational domain is therefore 1 mm × 10 mm × 1 mm.

bubbles are initialized within this domain, so that the fictitious unit cells no. 1 and 5 at the top and bottom of the computational domain contain only liquid. The volume of each of the three bubbles is so that the gas volume fraction in the fictitious unit cells no. 2, 3 and 4 is $\varepsilon = 33.06\%$. This three-bubble configuration is chosen so that the influence of the top and bottom bubble on the bubble in the middle should resemble the influence of the leading and trailing bubbles in BTF. For laminar flow, the entrance length L_{e} can be estimated by formula $L_{\mathrm{e}} = 0.06 Re\, D_{\mathrm{h}}$. In the present simulations, the Reynolds number is always less than 4 which yields $L_{\mathrm{e}} \leqslant 0.24 D_{\mathrm{h}}$. Thus, the liquid-filled fictitious unit cell 1 ensures that the uniform inlet velocity can develop downstream in a parabolic profile when it enters the fictitious unit cell 2. The fictitious unit cell 5 ensures that the outflow boundary specified for the bottom of the domain is sufficiently far away from the lowest bubble.

DOI: 10.1002/fld

An open question for which no final answer can be given here is, in how far the above three bubble configuration used in the FLUENT computations adequately represents an infinitely long train of bubbles. Unfortunately, a numerical study with substantially more than three bubbles could not be performed in 3-D because of the limited computer resources. However, as a preliminary investigation of this topic, 2-D axisymmetric simulations for the flow in a circular pipe have been performed with FLUENT for three, five, seven and 11 bubbles within the domain. The fluid properties and hydrodynamic conditions in the 2-D case have been chosen similar to the 3-D case. The comparison of the results of the various 2-D simulation shows that the variations of the central bubble from case to case are in general small so that—as a first approximation—the three-bubble configuration may be reasonable.

In all FLUENT computations, the initial velocity within the domain is zero, which means that in a fixed frame of reference both the bubbles and liquid move upwards with the same velocity, namely U_{wall}. The present choice of inflow and outflow conditions implies that the volumetric flow rate is specified while the pressure drop across the computational domain adjusts accordingly and thus is a result of the simulation. This situation is therefore *vice versa* as in the computational set-up for TURBIT-VOF, STAR-CD and CFX.

For the numerical solution of the governing equations, the segregated solver of FLUENT is used. For the pressure–velocity coupling, the PISO algorithm is adopted with the non-iterative time-advancement scheme which means that there are no outer iterations. For the momentum equation, the QUICK scheme is employed and time integration is performed by the first-order implicit scheme.

The FLUENT code offers within its VOF method four different numerical schemes for solution of the volume fraction equation. Two of these are performing an interface reconstruction (the ‘geometric interface reconstruction method’ and the ‘donor–acceptor method’), while the other two employ difference schemes (the ‘Euler-explicit method’ and the ‘implicit method’). The geometric interface reconstruction which is based on a generalization of the method of Youngs [29] can be considered as the most accurate scheme, since in this method the interface is, similar to TURBIT-VOF, represented by piecewise linear elements whereas in the donor–acceptor scheme of Hirt and Nichols [14] the interface representation is piecewise constant. For comparison, however, computations are performed by all four different VOF schemes.

In FLUENT, the following formulation of the CSF model for the surface tension force is used [30]:

$$\mathbf{f}_\sigma = \frac{\rho_{\mathrm{m}} \sigma \kappa \nabla \alpha_{\mathrm{L}}}{\frac{1}{2}(\rho_{\mathrm{G}} + \rho_{\mathrm{L}})} \quad \text{with } \kappa = -\nabla \cdot \frac{\nabla \alpha_{\mathrm{L}}}{|\nabla \alpha_{\mathrm{L}}|} \tag{12}$$

The FLUENT simulations have all been run in double precision on a work station with Dual XEON processors of 3.4 GHz and 8 GB RAM. The typical CPU time per mesh cell and time step is about 50 μs.

2.4. Overview on simulation runs

In order to allow for a reasonable comparison of the numerical results for BTF, similar numerical methods and discretization parameters have been used in the different CFD codes, as far as possible. Table I gives an overview about all the simulations performed within the present numerical study. In all runs, a uniform, rectilinear and isotropic hexahedral grid is used. To allow for grid refinement studies, three different values of the mesh size are considered, namely $h^* = h/2\,\mathrm{mm} = 1/48$, $1/64$

Table I. Overview on simulations performed with different computer codes.

Code	Case*	$\Delta\tilde{p}$ (Pa)	ε_0 (%)	Domain	Grid	Δt (μs)	N_t (dimensionless)	U_B (cm/s)	U_L (cm/s)	D_B (mm)	Remark
TURBIT-VOF	TBL48F	0.0	30.38	Full	48 × 48 × 48	1.892	20 000	2.917	1.011	1.670	
	TBH48F	0.0	33.07	Full	48 × 48 × 48	1.892	22 000	3.012	1.066	1.628	
	TBH64F	0.0	33.07	Full	64 × 64 × 64	0.757	94 000	3.023	1.066	1.711	
	TPH48F	−18.0	33.07	Full	48 × 48 × 48	1.892	40 000	9.435	3.179	1.628	
	TPH64F	−18.0	33.07	Full	64 × 64× 64	0.757	100 000	9.409	3.171	1.629	
STAR-CD	SBL48F	0.0	30.37	Full	48 × 48 × 48	0.750	80 000	2.699	0.956	1.713	
	SBL64F	0.0	30.37	Full	64 × 64 × 64	0.750	119 600	2.691	0.954	1.712	
	SBL128Q	0.0	30.37	Quarter	64 × 128 × 64	0.750	60 000	2.574	0.931	1.718	
	SBH48F	0.0	33.07	Full	48 × 48 × 48	0.750	60 000	2.714	0.989	1.750	
	SBH64F	0.0	33.05	Full	64 × 64 × 64	0.750	88 400	2.700	0.992	1.756	
	SPH64Q	−18.0	33.05	Quarter	32 × 64 × 32	0.750	82 000	8.098	3.009	1.693	
	SPH64Q-1	−18.0	33.05	Quarter	32 × 64 × 32	0.750	60 000	(‘Annular flow’)			Upwind scheme for continuity equation
	SPH64Q-2	−18.0	33.05	Quarter	32 × 64 × 32	0.750	60 000	8.089	3.009	1.699	Upwind scheme for momentum equation
	SPH64Q-T	−18.0	33.05	Quarter	32 × 64 × 32	7.500	12 000	8.203	3.021	1.693	Tenfold Δt
CFX	CBL48Q-T	0.0	30.97	Quarter	24 × 48 × 24	7.500	4000	3.117	0.999	1.658	Tenfold Δt
	CBH48Q	0.0	34.15	Quarter	24 × 48 × 24	0.750	40 000	3.143	0.951	1.704	
	CBH48Q-T	0.0	34.15	Quarter	24 × 48 × 24	7.500	6000	2.796	0.931	1.700	Tenfold Δt
	CBH48Q-T-1	0.0	34.15	Quarter	24 × 48 × 24	7.500	4000	3.000	0.948	1.708	2nd order time integration
	CBH48Q-T-2	0.0	34.15	Quarter	24 × 48 × 24	3.785	7927	2.988	0.941	1.708	Fivefold Δt
	CBH64Q	0.0	33.78	Quarter	32 × 64 × 32	0.750	10 000	—	—	—	(Only simulation of initial transient)
	CBH64Q-T	0.0	33.78	Quarter	32 × 64 × 32	7.500	6000	3.385	1.134	1.701	Tenfold Δt
	CPH48Q-T	−18.0	34.14	Quarter	24 × 48 × 24	7.500	4000	(‘Annular flow’)			Tenfold Δt
	CPH64Q	−18.0	33.01	Quarter	32 × 64 × 32	0.750	24 160	10.120	3.136	1.641	

DOI: 10.1002/fld

Table I. *Continued.*

Code	Case*	$\Delta\tilde{p}$ (Pa)	ε_0 (%)	Domain	Grid	Δt (μs)	N_t (dimensionless)	U_B (cm/s)	U_L (cm/s)	D_B (mm)	Remark
FLUENT	FBH48Q-G	−0.13	33.05	Quarter	24 × 240 × 24	0.750	30 000	3.009	1.015	1.699	Geo-reconstruct
	FBH48Q-T-G	−0.02	33.05	Quarter	24 × 240 × 24	7.570	3000	3.010	1.014	1.698	Geo-reconstruct, tenfold Δt
	FBH64Q-T-G	−0.7	33.06	Quarter	32 × 320 × 32	7.570	3000	3.013	1.038	1.709	Geo-reconstruct, tenfold Δt
	FBH64Q-T-D	1.4	33.06	Quarter	32 × 320 × 32	7.570	3000	2.943	1.073	1.727	Donor–acceptor, tenfold Δt
	FBH64Q-T-E	0.8	33.06	Quarter	32 × 320 × 32	7.570	3000	2.969	1.060	1.689	Euler-explicit, tenfold Δt
	FBH64Q-T-I	−1.0	33.06	Quarter	32 × 320 × 32	7.570	3000	2.974	1.057	1.689	Implicit, tenfold Δt
	FPH64Q-T-G	−21.5	33.06	Quarter	32 × 320 × 32	7.570	4500	9.584	3.322	1.634	Geo-reconstruct, tenfold Δt
	FPH64Q-T-D	−17.9	33.06	Quarter	32 × 320 × 32	7.570	3000	9.228	3.108	1.684	Donor–acceptor, tenfold Δt
	FPH64Q-T-E	—	33.06	Quarter	32 × 320 × 32	7.570	2050	(Execution stop by code)			Euler-explicit, tenfold Δt
	FPH64Q-T-I	—	33.06	Quarter	32 × 320 × 32	7.570	300	(No converging solution)			Implicit, tenfold Δt

Note: The values of U_B, U_L and D_B are the terminal ones with exception of D_B which is, for CFX and STAR-CD only, for $t = 0.03$ s.

*Different cases are abbreviated as follows: the first letter indicates the computer code (C = CFX, F = FLUENT, S = STAR-CD, T = TURBIT-VOF); the second letter indicates the driving forces (B = buoyancy only, P = pressure gradient and buoyancy); the third letter indicates the gas content (H = high, L = low); the digits indicate the number of mesh cells per unit cell in the y-direction, while the next letter indicates the computational domain (F = full, Q = quarter). Any further digit or character indicates the variation of numerical scheme or time step width.

DOI: 10.1002/fld

and $1/128$ (only for STAR-CD). Grid refinement studies are performed for all four CFD codes. While in some of the simulations, the computational domain consists of the entire cross-section, in most of the simulations a quarter of the channel cross-section is considered only together with symmetry conditions in x- and z-direction at the two vertical mid-planes of the channel. With CFX and FLUENT only simulations for a quarter of the channel cross-section are performed.

All simulations are performed transient and in most of the simulations a time step width $\Delta t = 0.757\ \mu\text{s}$ is used. This time step width and the grid size $h/2\ \text{mm} = 1/64$ correspond to previous simulations performed with TURBIT-VOF for BTF [19, 20]. Since the commercial CFD codes CFX, STAR-CD and FLUENT use implicit time integration schemes, much larger time steps are possible than can be used with the explicit time integration scheme in TURBIT-VOF. Therefore, in some simulations with the commercial CFD codes, a fivefold or tenfold larger time step width is used. While the largest time step width $\Delta t = 7.57\ \mu\text{s}$ did not lead to any convergence problems, it was found that a further increase of the time step width in FLUENT leads to a breakdown of the computation.

All simulations are started from fluid at rest. The initial phase distribution corresponds to a spherical bubble placed in the centre of the computational domain. Essentially, two different initial gas volume fractions within the unit cell are considered, namely $\varepsilon \approx 33\%$ and 30%. These values correspond to an initial spherical bubble with a diameter of about 1.714 and 1.661 mm, respectively. The initialization of a bubble which corresponds to a prescribed gas volume fraction within the domain is, however, not always a trivial task in commercial CFD codes and requires in general some trial and error. This is because usually the initialization of the volume fraction distribution is associated with some smearing of the interface. It was, therefore, not possible to obtain for all the simulations exactly the same initial gas hold up. Because the fluids are immiscible and there is no phase change or mass transfer, the gas volume fraction should be constant throughout the entire simulation and should not deviate from the initial value ε_0.

3. RESULTS AND DISCUSSION

In this section, we will present the results of our code-to-code comparison, first for the cases where the flow is driven by buoyancy only ($\Delta\tilde{p} = 0\ \text{Pa}$) and then where in addition to buoyancy a constant body force corresponding to a pressure gradient $-\Delta\tilde{p}/L_y = 9000\ \text{N/m}^3$ is specified. For a comparison of the different computer codes, we first present results for the temporal evolution of the mean vertical gas and liquid velocity within the computational domain. For any instant of time, these mean velocities are evaluated by relations

$$U_{\text{B}} = \frac{\sum_{i=1}^{N_{\text{i}}} \sum_{j=1}^{N_j} \sum_{k=1}^{N_k} (1 - f_{i,j,k}) v_{y;i,j,k}}{\sum_{i=1}^{N_{\text{i}}} \sum_{j=1}^{N_j} \sum_{k=1}^{N_k} (1 - f_{i,j,k})} \tag{13}$$

$$U_{\text{L}} = \frac{\sum_{i=1}^{N_{\text{i}}} \sum_{j=1}^{N_j} \sum_{k=1}^{N_k} f_{i,j,k} v_{y;i,j,k}}{\sum_{i=1}^{N_{\text{i}}} \sum_{j=1}^{N_j} \sum_{k=1}^{N_k} f_{i,j,k}} \tag{14}$$

Here, v_y is the y-component of $\mathbf{v}_{\text{m}}$. The subscripts i, j and k denote the mesh cell index in x-, y-, and z-direction, respectively, and N_i, N_j and N_k denote the number of mesh cells in these

DOI: 10.1002/fld

directions within the unit cell. For FLUENT, the velocities U_{B} and U_{L} are obtained by adding to the values computed by Equations (13) and (14) for all three bubbles the constant wall velocity U_{wall}. If all the three bubbles would stay exactly at rest within the domain, then U_{B} according to Equation (13) would be zero for FLUENT. However, in practice, there was always a slight intermittent upward/downward variation of the positions of the three bubbles. As a consequence, U_{B} may differ from U_{wall}. While the three bubbles were slightly moving, they nevertheless stayed always within their individual initial fictitious unit cell. So during the course of the FLUENT computations, there was no temporal variation of the gas volume fraction in the fictitious unit cells. However, one may speculate that for much longer problem times than are considered here, this may happen and coalescence between bubbles may occur.

A good performance for industrial devices, such as monoliths, is obtained when the BTF reaches a steady zone [18]. Therefore, keeping BTF in steady conditions is one of the most important issues to get high mass and heat transfer rates *via* very thin diffusion paths and large interfacial area. Our interest here is therefore mainly the steady terminal bubble and liquid velocity, and not on the transient phase when the bubble accelerates and deforms from its initial spherical shape. Nevertheless, in the following, we show figures which display the entire temporal evolution of U_{B} and U_{L}.

In this section, we will also show figures of the steady bubble shape. In all simulations, the bubble is axisymmetric and, therefore, any axial cross-section through the bubble results in a circle. To quantify the bubble shape, we give numerical values for the bubble diameter D_{B}, which is the largest diameter of this circle for all axial cross-sections of the bubble. The steady-state numerical values of U_{B}, U_{L} and D_{B} obtained by the different runs are given in Table I.

The validation of the in-house code TURBIT-VOF against experiments from Thulasidas *et al.* [18] has already been done in [20] resulting in good agreement for long enough unit cells. The TURBIT-VOF results can, therefore, be considered here as reference values. Since the unit cell length used in the present study is rather short, and is much shorter than in the experiment of Thulasidas *et al.* [18], a direct comparison of the present numerical results with these experimental data is not meaningful.

3.1. Bubble-train flow driven by buoyancy only

In this section, we will discuss results for BTF where the only driving force is buoyancy and where, therefore, $|\mathbf{f}_{\mathrm{pd}}| = 0\,\mathrm{N/m^3}$. Figure 3 shows the temporal evolution of the bubble velocity U_{B} and the mean liquid velocity U_{L} for the basic case BH[‡] ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 33\%$) computed by the different computer codes. As the simulations with TURBIT-VOF, CFX and STAR-CD are started from fluid at rest in a fixed frame of reference, both U_{B} and U_{L} strongly increase initially but reach an almost constant value after about 0.015 s. For the FLUENT computations, the geometric interface reconstruction scheme has been used. Since this simulation is started from fluid at rest in a frame of reference moving with the terminal bubble velocity computed by TURBIT-VOF, the FLUENT results show initially a sharp decrease of U_{B} and U_{L} and then an increase till almost constant values are reached for both velocities, too. These terminal velocities are, however, attained more early at about 0.007 s. The terminal bubble velocity in TURBIT-VOF and FLUENT corresponds to

[‡] For an explanation of the nomenclature used for the description of the different cases, see the footnote in Table I.

DOI: 10.1002/fld

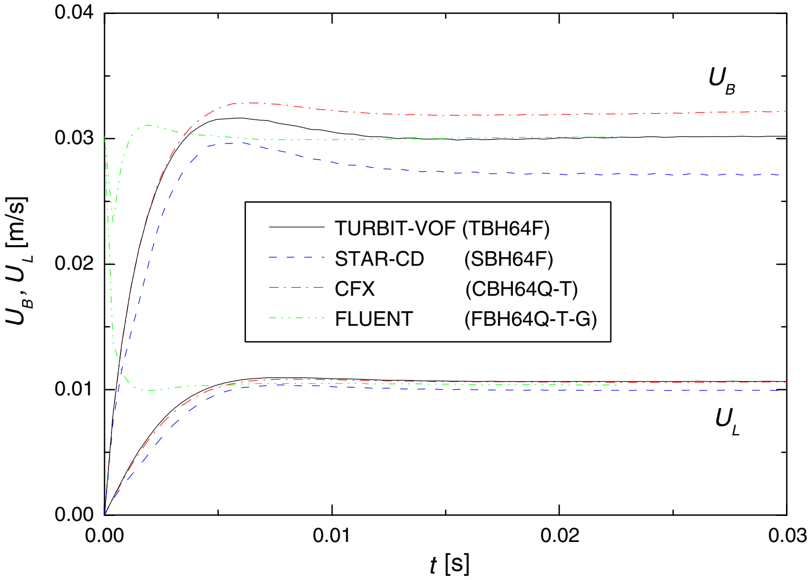

Figure 3. Temporal evolution of bubble velocity and mean liquid velocity for test case BH ($\Delta\tilde{p}=0$ Pa, $\varepsilon\approx 33\%$) and mesh width $h^*=1/64$. The time step width used for TURBIT-VOF and STAR-CD is 0.757 µs while for CFX and FLUENT it is about 7.5 µs. For an explanation of the nomenclature of the test cases given in brackets, see footnote of Table I.

a capillary number $Ca=0.065$ and a bubble Reynolds number $Re_{\mathrm{B}}\equiv\rho_{\mathrm{L}}D_{\mathrm{h}}U_{\mathrm{B}}/\mu_{\mathrm{L}}=1.205$, where $D_{\mathrm{h}}=2$ mm is the hydraulic diameter.

The comparison of the different curves in Figure 3 shows that the terminal value of U_{L} obtained by STAR-CD is slightly smaller than the respective values obtained by the other codes. As the bubble velocity is concerned, the TURBIT-VOF result is strongly underestimated by STAR-CD and substantially overestimated by CFX. The terminal value of U_{B} for FLUENT and TURBIT-VOF is almost the same as they should. Figure 3 shows that the terminal mean liquid velocity U_{L} computed by FLUENT also corresponds well with the TURBIT-VOF value as it is only about 2.6% smaller. This indicates that the TURBIT-VOF and FLUENT result do well agree, though performed in different frames of reference. A further indication for this is the pressure difference $\Delta\tilde{p}$ across the unit cell. While in TURBIT-VOF the value $\Delta\tilde{p}=0$ Pa was an input, in FLUENT it is a result of the simulation and a value of -0.7 Pa is obtained. We thus can conclude that the VOF methods with piecewise linear geometrical interface reconstruction as implemented in the TURBIT-VOF and FLUENT code give very similar results which differ from those obtained by the STAR-CD and CFX code, which do not perform an interface reconstruction but use a higher-order difference method to solve the volume fraction equation.

In Figure 4, we show a lateral view of the terminal bubble shape computed by the four different codes for case BH ($\Delta\tilde{p}=0$Pa, $\varepsilon\approx 33\%$). For TURBIT-VOF, the bubble shape is obtained as follows. For each mesh cell that contains both phases ($0<f<1$), the centroid of the plane representing the interface is computed. For neighbouring mesh cells, the points computed in this way are then connected to form triangles or quadrangles so that a closed surface is obtained. For the three

DOI: 10.1002/fld

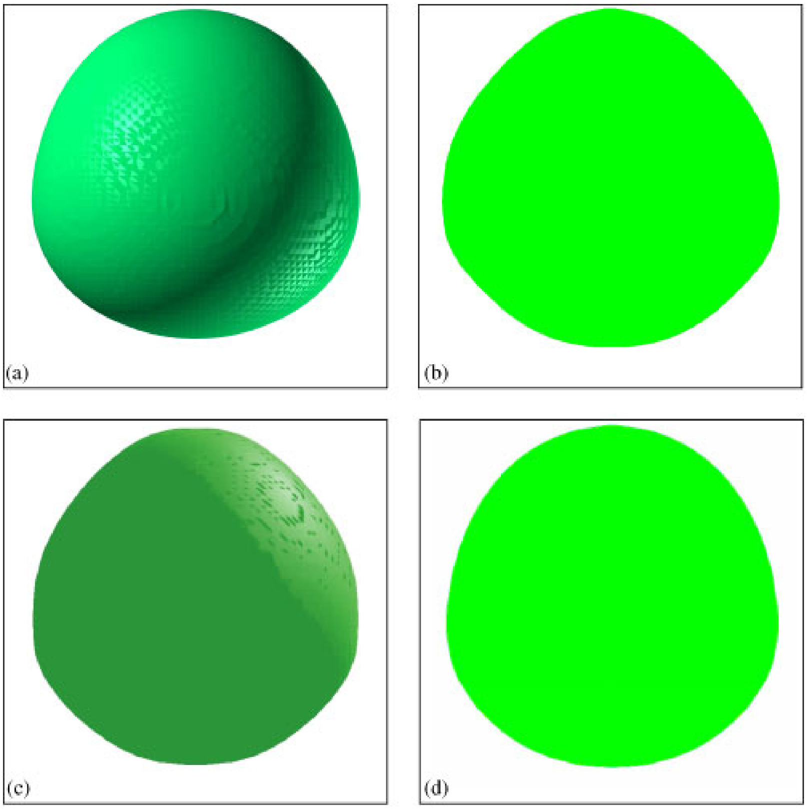

Figure 4. Lateral view of steady bubble shape computed for case BH ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 33\%$) by: (a) TURBIT-VOF (case TBH64F); (b) STAR-CD (case SBH64F); (c) FLUENT (case FBH64Q-T-G); and (d) CFX (case CBH64Q-T). For FLUENT only the bubble in the middle is shown.

commercial CFD codes, the bubble shape is obtained in a simpler but quite common manner, namely as the iso-surface corresponding to the value $f = 0.5$. While at a first look the bubble shapes in Figure 4 appear to be very similar, there are some differences to note. These concern mainly the bubble tip, which is rounded for TURBIT-VOF and CFX, is pointed for STAR-CD and is flat for FLUENT. The bubble diameter D_{B} is almost the same for TURBIT-VOF, FLUENT and CFX, while the value for STAR-CD is slightly higher, see Table I. Differences in bubble shape can be observed for the lower-half of the bubble, too. While the rear part of the bubble is rather smoothly curved for TURBIT-VOF and CFX, there is for FLUENT and STAR-CD a part where the interface is almost flat in this lateral view.

From the visual impression of the bubble shape and the values for the bubble diameter, the underestimation of the bubble terminal velocity by STAR-CD of about 10% and the overestimation by CFX of about 8% (for $t \approx 0.03$ s) as compared to the TURBIT-VOF and FLUENT results is surprising. To investigate possible reasons for these discrepancies additional simulations have been performed by CFX, STAR-CD and TURBIT-VOF, where the initial overall void fraction ε is reduced from about 33% to about 30% and a coarser grid with $h^* = 1/48$ is used. The results

DOI: 10.1002/fld

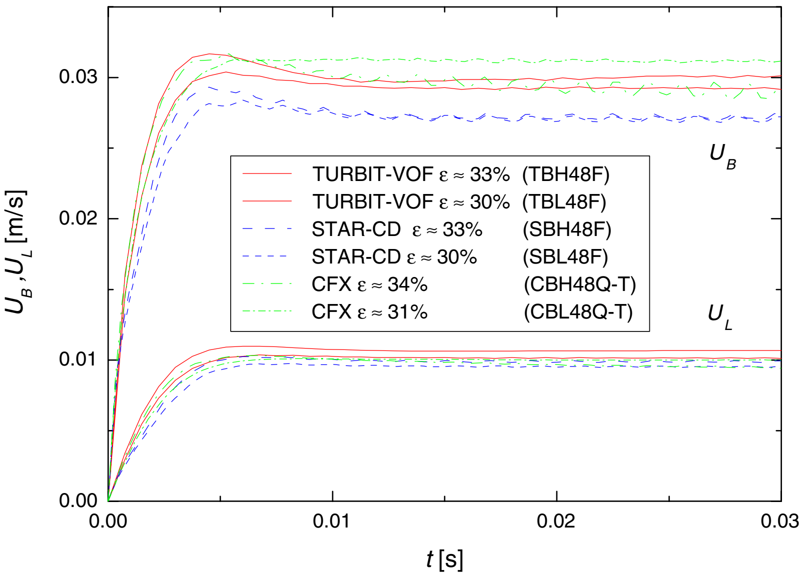

Figure 5. Temporal evolution of bubble velocity and mean liquid velocity for test case BH ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 33\%$) and BL ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 30\%$) and mesh width $h^* = 1/48$. The time step width used for TURBIT-VOF is about 1.9 μs while it is 0.757 μs for STAR-CD and 7.57 μs for CFX.

obtained by CFX, STAR-CD and TURBIT-VOF for both void fractions are displayed in Figure 5. Since the flow is driven by buoyancy only, it can be expected that the decrease of ε will result in a decrease of both, U_B and U_L. Figure 5 shows that this is valid for TURBIT-VOF. For STAR-CD, however, there is only a decrease of U_L while the bubble velocity U_B remains unaffected. For CFX, the situation is even more confusing for two reasons. First, for the high-void-fraction case and the coarser grid now the bubble velocity is no more substantially overestimated but slightly underestimated as compared to TURBIT-VOF. Second, the bubble velocity for the low-void-fraction case is much higher than that for the high-void-fraction case. This result appears to be rather unphysical.

A possible reason for the discrepancies in the CFX and STAR-CD simulations may be an insufficient grid resolution. To investigate this topic, simulations with STAR-CD for three different grids and otherwise unchanged conditions have been performed. The results are displayed in Figure 6. For the terminal mean liquid velocity, no notable difference can be identified on the three different grids. However, the bubble velocity is somewhat smaller on the finer grid. Also, we observe a slight oscillation of the mean bubble velocity on the coarsest grid (SBL48F) which is not present on the two finer grids. Nevertheless, we may conclude that the underestimation of the bubble velocity by STAR-CD as compared to TURBIT-VOF and FLUENT is not a result of insufficient grid resolution. Instead, it may be related to the overestimated bubble diameter and thus to the bubble shape, which has a strong influence on the bubble drag. Differences in the bubble drag are directly linked to differences in bubble velocity.

For TURBIT-VOF, a grid refinement study is performed, too. The comparison of the data for U_B and U_L in Table I for cases TBH48F and TBH64F shows that the differences are very small, while

DOI: 10.1002/fld

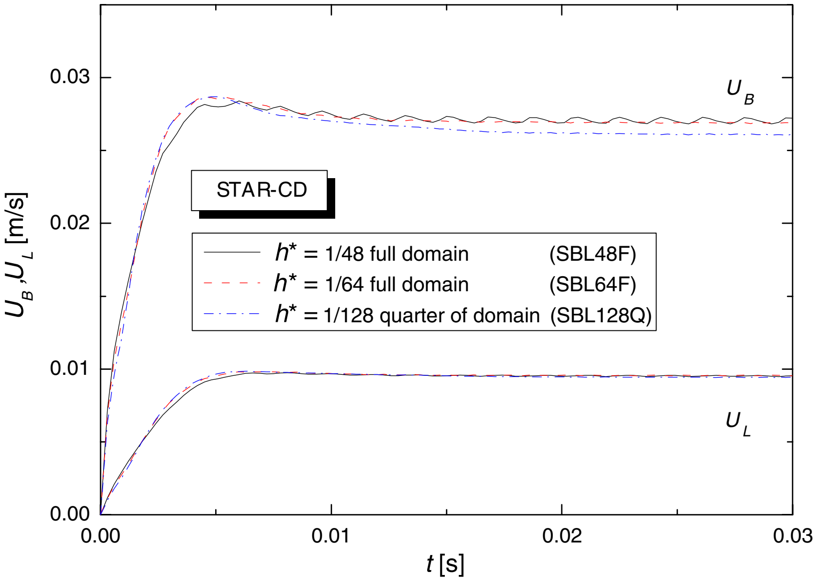

Figure 6. Temporal evolution of bubble velocity and mean liquid velocity computed by STAR-CD for test case BL ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 30\%$) on three different grids ($\Delta t = 0.75$ μs).

the difference for the bubble diameter is somewhat larger. For the case with pressure gradient, however, the results of cases TPH48F and TPH64F are almost identical.

For CFX, the influence of mesh size and time step width is illustrated in Figure 7. Here, results for case BH ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 33\%$) are compared for two different grids and three different values of the time step width. Both, for the fine grid ($h^* = 1/64$) and for the coarse grid ($h^* = 1/48$), the bubble velocity U_B depends strongly on the time step width while U_L does not. Namely, on both grids the increase of the time step width by a factor of 10 results in a decrease of U_B of about 5–10%. A comparison of results for U_B obtained for the same time step width on the fine and the coarse grid shows that for the small Δt there is no influence of the mesh size while there is an influence of the mesh size for the tenfold increased time step width. We also note that for all CFX simulations with $h^* = 1/48$, the bubble velocity oscillates in time. This effect was also observed in the STAR-CD simulation on this grid (see Figure 6). Finally, from comparing cases CBH48Q-T and CBH48Q-T-1 in Figure 7, we can conclude that the influence of the order of the time integration scheme is very small in CFX.

In Figure 8, results of the four different VOF methods of FLUENT (‘geometric interface reconstruction’, ‘donor–acceptor’, ‘Euler-explicit’ and ‘implicit’) are compared. While the differences between the methods are somewhat larger for the initial transient phase, which is associated with a deformation of the bubble from the initial spherical shape, the differences in the terminal velocities of gas and liquid are notable but are not dramatic. The best agreement with the TURBIT-VOF results is obtained for the geometric interface reconstruction scheme. The bubble shapes obtained with the different VOF schemes are displayed in Figure 9. While some small differences can be noted, in general, the bubble shapes are very similar.

DOI: 10.1002/fld

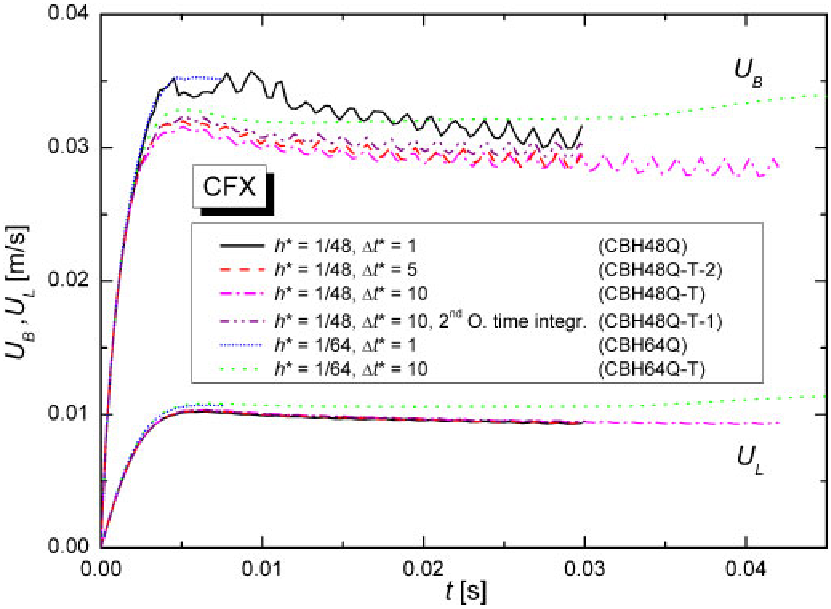

Figure 7. Temporal evolution of bubble velocity and mean liquid velocity computed by CFX for test case BH ($\Delta\tilde{p} = 0$ Pa, $\varepsilon \approx 33\%$) using two different grids and different values of the time step width.

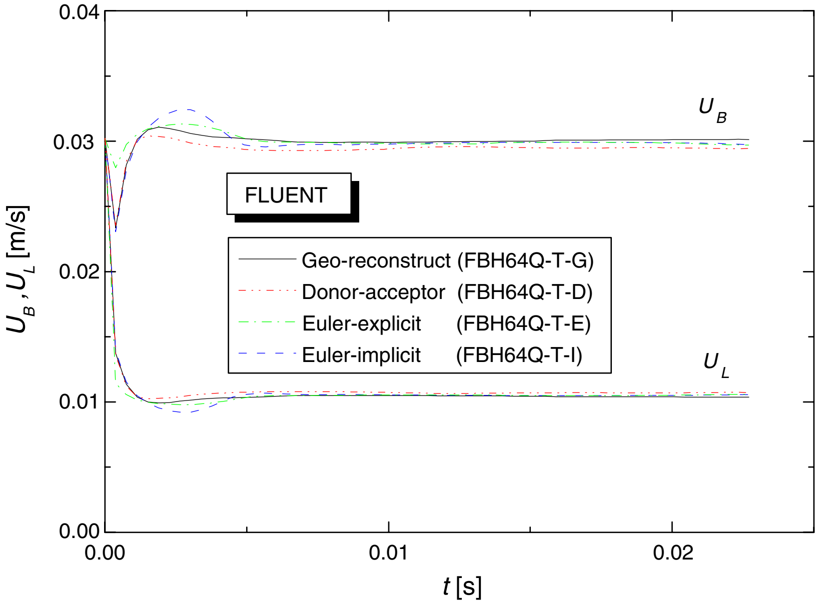

Figure 8. Temporal evolution of bubble velocity and mean liquid velocity for the different VOF schemes of FLUENT for buoyancy-driven flow.

DOI: 10.1002/fld

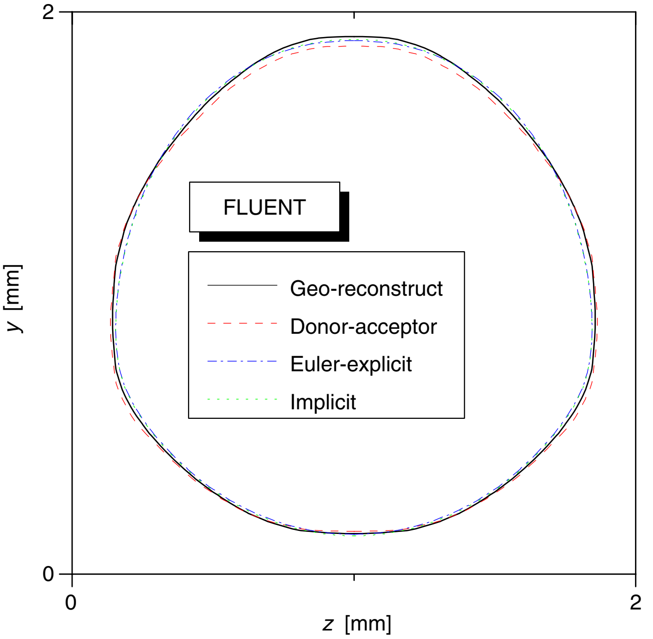

Figure 9. Comparison of FLUENT results for the bubble shape obtained by the different VOF schemes for buoyancy-driven flow. The symbols for the Euler-explicit scheme (case FBH64Q-T-E) are hardly visible because they are mostly below the ones for the implicit scheme (case FBH64Q-T-I). (Only the shape of the bubble in the middle is displayed.)

For the geometric interface reconstruction scheme of FLUENT, two additional computations have been performed to investigate the influence of grid size and time step width. A comparison of cases FBH48Q-G and FBH48Q-T-G (not shown graphically) yields almost identical time histories of U_B and U_L though the time step width in both cases differs by a factor of 10. Thus, for this scheme, the influence of the time step width is marginal. However, there is a slight influence of the mesh size. This affects in particular the mean liquid velocity U_L, which is about 2% lower in case FBH48Q-T-G than in case FBH64Q-T-G (see Table I).

3.2. Bubble-train flow driven by buoyancy and external pressure gradient

In this section, we now present the results for the BTF driven by buoyancy and by an external axial pressure gradient which corresponds to a body force of $9000\,\mathrm{N/m^3}$. The results for the time-dependent bubble velocity and mean liquid velocity obtained by the different computer codes are displayed in Figure 10. For FLUENT, the geometric interface reconstruction scheme has been used. As expected, in the case with pressure gradient both U_B and U_L are higher than in the case with buoyancy only. As a consequence, the transient phase till the bubble reaches its terminal velocity takes more time. The terminal bubble velocity in TURBIT-VOF and FLUENT corresponds to a capillary number $Ca = 0.204$ and a bubble Reynolds number $Re_B = 3.75$. Similar to the case with buoyancy only, displayed in Figure 3, the STAR-CD results are again substantially lower than their TURBIT-VOF and FLUENT counterparts while for the CFX run with $h^* = 1/64$ the bubble

DOI: 10.1002/fld

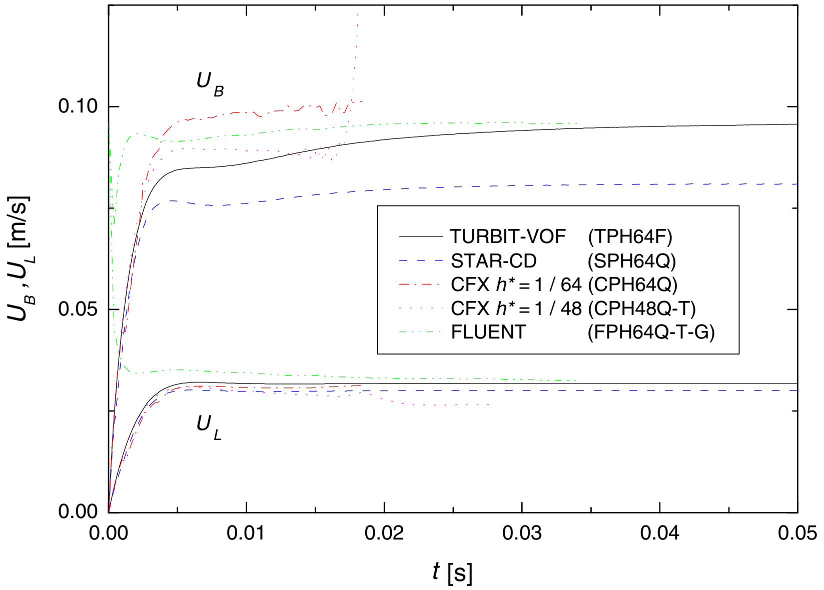

Figure 10. Temporal evolution of bubble velocity and mean liquid velocity for test case PH ($\Delta\tilde{p} = -18$ Pa, $\varepsilon \approx 33\%$) and $h^* = 1/64$. For CFX results are also shown for a grid with $h^* = 1/48$.

velocity is again overestimated. In Figure 10, CFX results are also displayed for a coarse grid ($h^* = 1/48$) and a tenfold increased time step width. For this case, a runaway of U_B is observed at $t \approx 0.017$ s. This is because the bubble, which is initially spherical, continues to elongate until finally the liquid slug disappears and an annular flow is formed. Thus, with CFX on this grid, no stable BTF is computed.

For FLUENT and the geo-reconstruction scheme, the mean liquid velocity is about 5% higher than the TURBIT-VOF value. The pressure difference $\Delta\tilde{p}$ across the unit cell evaluated from the FLUENT computations is -21.5 Pa which is in acceptable agreement with the TURBIT-VOF input value $\Delta\tilde{p} = -18$ Pa. Here, for the donor–acceptor method, a much better agreement is obtained as the respective value is -17.9 Pa. When comparing the results of TURBIT-VOF and FLUENT, we have to recall that for the first code due to the periodic boundary conditions a BTF consisting of an ‘infinite’ number of identical bubbles is considered, while in FLUENT instead only a three bubble configuration is simulated. Having this in mind, one may argue that the agreement between both codes is reasonable.

In Figure 11, we show a lateral view of the terminal bubble shape computed by the four different codes for case PH ($\Delta\tilde{p} = -18$ Pa, $\varepsilon \approx 33\%$). Overall, the bubble shapes are similar. From Table I, one can note that the bubble diameter for TURBIT-VOF and FLUENT is almost the same. However, for STAR-CD again the value of D_B is somewhat larger as was also observed for the case with buoyancy only. It also appears that for FLUENT the length of the liquid slug is slightly shorter than for the other codes.

For STAR-CD, additional simulation runs have been performed to investigate the influence of the difference scheme and the time step width. While in the basic case SPH64Q and in case SPH64Q-T, the high-resolution advection scheme is used both, for the volume fraction equation

DOI: 10.1002/fld

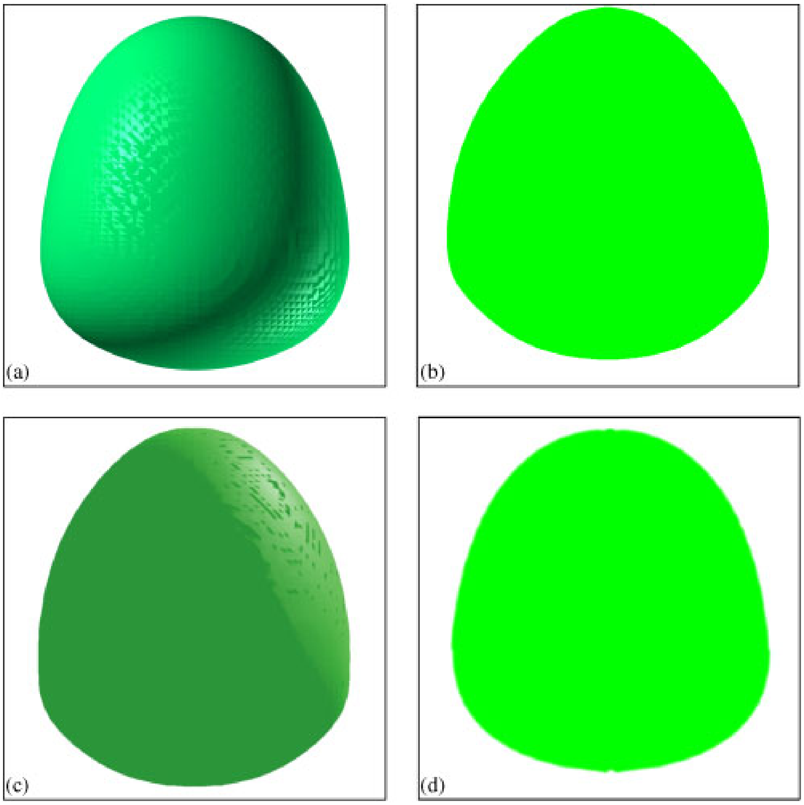

Figure 11. Lateral view of steady bubble shape computed for case PH ($\Delta\tilde{p} = -18$ Pa, $\varepsilon \approx 33\%$) by: (a) TURBIT-VOF (case TPH64F); (b) STAR-CD (case SPH64Q); (c) FLUENT (case FPH64Q-T-G); and (d) CFX (case CPH64Q). For FLUENT only the bubble in the middle is shown.

and for the momentum equation, in case SPH64Q-1 instead the upwind scheme is used for the volume fraction equation, while in case SPH64Q-2 the upwind scheme is used for the momentum equation. Cases SPH64Q and SPH64Q-T differ only by the time step width, which is increased in the latter case by a factor of 10. The results are shown in Figure 12. The comparison of cases SPH64Q and SPH64Q-2 shows that the choice of the high-resolution scheme or upwind scheme for the momentum equation makes no difference. However, the comparison of cases SPH64Q and SPH64Q-1 reveals that the replacement of the high-resolution scheme in the volume fraction equation by the upwind scheme does not give reliable results. The use of the upwind scheme for the volume fraction equation is associated with a considerable smearing of the interface over several mesh cells and does not lead to BTF but to annular flow. A comparison of cases SPH64Q and SPH64Q-T shows that the influence of the time step width in STAR-CD is very small.

In Figure 13, results of the four different VOF methods in FLUENT are compared for the case with pressure gradient. For the geometric interface reconstruction scheme and the donor–acceptor method, BTF is computed. However, the use of the implicit scheme did not yield a converging

DOI: 10.1002/fld

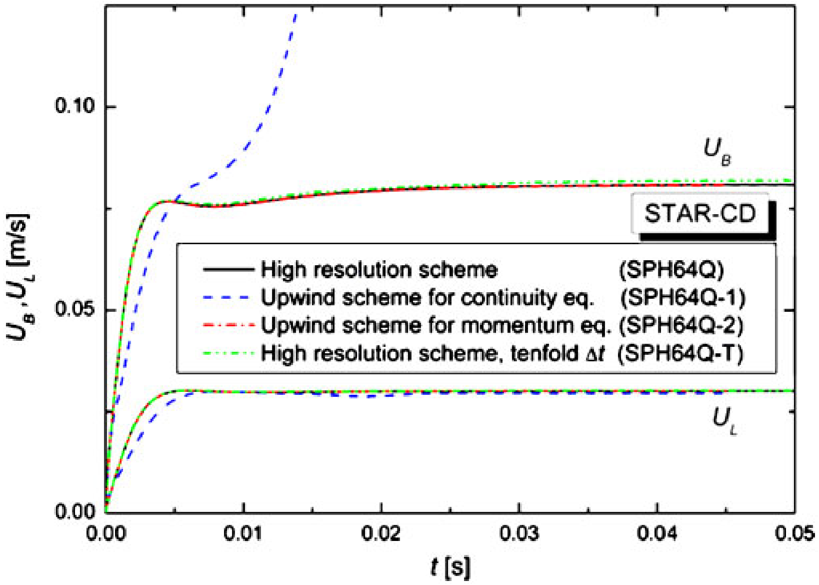

Figure 12. Temporal evolution of bubble velocity and mean liquid velocity computed by STAR-CD for test case PH ($\Delta\tilde{p} = -18$ Pa, $\varepsilon \approx 33\%$) for different schemes in the volume fraction and momentum equation and for different time step widths.

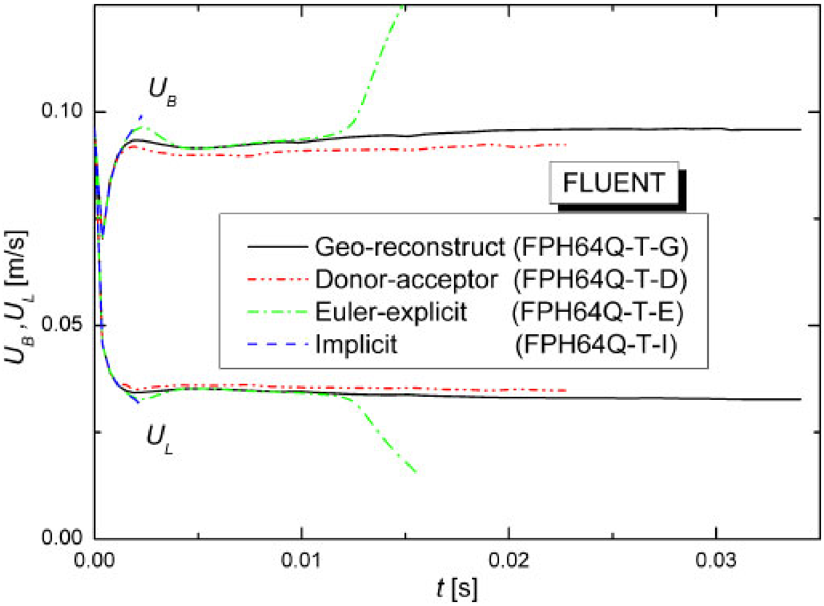

Figure 13. Temporal evolution of bubble velocity and mean liquid velocity for the different VOF schemes of FLUENT for pressure-driven flow.

DOI: 10.1002/fld

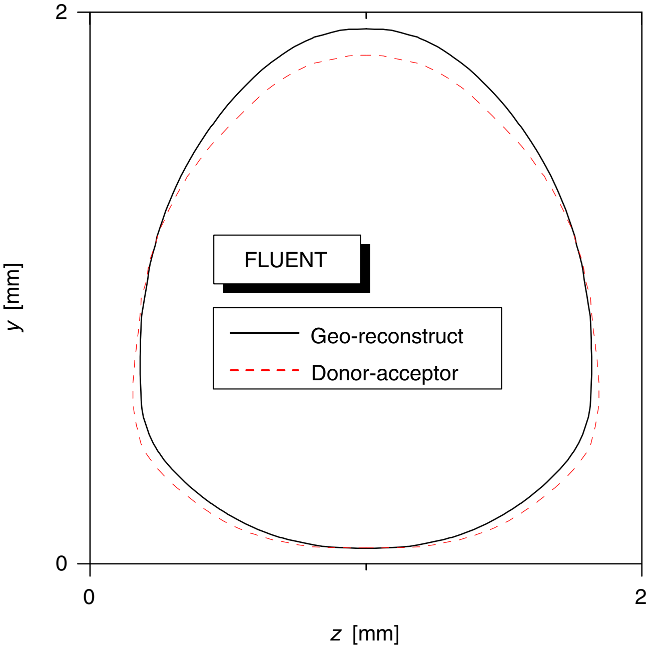

Figure 14. Comparison of FLUENT results for the bubble shape obtained by the different VOF schemes for buoyancy- and pressure-driven flow. While the results for case FPH64Q-T-D are for time step 3000 and those for case FPH64Q-T-G are for time step 4500, it was verified that the difference in shape for case FPH64Q-T-G at time steps 3000 and 4500 is very small. (Only the shape of the bubble in the middle is displayed.)

solution at all, while the use of the Euler-explicit scheme resulted in a stop of the code execution and the message ‘to many VOF sub-timesteps’. The void fraction distribution computed up to this time step and the strong increase of U_B and the strong decrease of U_L suggest that coalescence of the bubbles will occur. It may be possible to obtain converging solutions with the implicit scheme and the Euler-explicit scheme of FLUENT when smaller time steps are used. However, this topic has not been investigated yet. The bubble shapes obtained with the geometric interface reconstruction scheme and the donor–acceptor scheme are displayed in Figure 14. While for buoyancy-driven flow, the difference in bubble shape between both methods was rather small (see Figure 9) now substantial differences can be observed. In the donor–acceptor method, the bubble diameter is larger, while the length of the bubble is much smaller and, as a consequence, the length of the liquid slug is much larger as compared to the case with the geometric interface reconstruction scheme.

A critical issue for the computation of two phase flows by any CFD code is the mass conservation of the phases. In the present study, the density of both phases is constant and mass conservation is equivalent to volume conservation. In transient computations typically a certain amount of mass, respectively, volume is gained or lost per time step, e.g. because of numerical diffusion or truncation errors. A suitable measure for this amount is given by the

DOI: 10.1002/fld

quantity

$$\frac{V_{\mathrm{G}}(t=N_t\Delta t)-V_{\mathrm{G}}(t=0)}{V_{\mathrm{G}}(t=0)}\frac{1}{N_t}=\frac{\varepsilon_{N_t}-\varepsilon_0}{\varepsilon_0 N_t}=\frac{\Delta\varepsilon}{\varepsilon_0 N_t} \tag{15}$$

Here, V_{G} is the volume of the gas phase. For TURBIT-VOF, the order of magnitude of $-\log_{10}(|\Delta\varepsilon|/(\varepsilon_0 N_t))$ is in the range 9.1–9.5, for FLUENT it is in the range 7.7–8.3 and for STAR-CD it is in the range 6.5–7.7. For CFX, surprisingly V_{G} does not change in time at all and is constant up to 30 digits if this quantity is evaluated in the CFX post-processing by volume integration (i.e. *volumeInt*(<*Expression*>)@<*Location*>). However, if V_{G} is instead evaluated by function average (i.e. *ave*(<*Expression*>)@<*Location*>), then its value differs from that obtained by volume integration and also changes in time. The order of magnitude of $-\log_{10}(|\Delta\varepsilon|/(\varepsilon_0 N_t))$ is then in the range 3.1–6.1. Overall, these results indicate that the mass conservation error per time step in the piecewise linear interface reconstruction methods of TURBIT-VOF and FLUENT is about one order of magnitude better than the one that can be obtained by the higher-order difference scheme methods in FLUENT and STAR-CD.

4. CONCLUSIONS

In the present paper, a comprehensive code-to-code comparison exercise for bubble-train flow (BTF) in a square mini-channel has been performed in order to asses the capabilities of CFD codes and numerical methods to deal with interfacial flows in small dimensions where surface tension effects are predominant and where the shape of the deformable interface is part of the solution. The flow configuration investigated consists of the co-current vertical BTF of air bubbles through silicone oil in a square mini-channel of 2 mm × 2 mm cross-section.

From the results obtained, we conclude that the VOF method with piecewise linear geometrical interface reconstruction as implemented in the TURBIT-VOF and FLUENT codes gives very similar and consistent results. Also the donor–acceptor method of FLUENT which is based on a piecewise constant interface approximation yields satisfactory results. In contrast, all methods that do not perform an interface reconstruction but use high-order difference schemes to solve the volume fraction equation show some deficiencies. These concern the Euler-explicit scheme and the implicit scheme of FLUENT and the methods implemented in STAR-CD and CFX. These deficiencies do not only affect the bubble and liquid velocity, as is the case for BTF driven by buoyancy only, but also may affect the entire flow structure. This is demonstrated for the case where the BTF is driven by an additional pressure gradient. While the methods based on geometric interface reconstruction are all able to resolve and maintain the thin liquid slug, the higher rate of smearing of the interface which is associated with the difference scheme VOF methods results in artificial bubble coalescence and thus annular flow instead of BTF or even results in a breakdown of the simulation run. The deficiencies of the difference scheme VOF methods have also been demonstrated by investigating the influence of gas hold up for buoyancy-driven flow (for STAR-CD and CFX) and by investigating the influence of the time step width (CFX).

In summary, one may conclude that VOF method computations for interfacial flows using higher-order difference schemes for solution of the volume fraction equation may lead to reliable results only for certain situations but cannot be in general recommended. This is especially valid for flows

DOI: 10.1002/fld

in small dimensions where surface tension effects are predominant. In contrast, the use of VOF methods that involve a piecewise linear geometric interface reconstruction gave reliable and consistent results and their use is highly recommended. This should also hold for two-phase flows in automotive applications (e.g. the sloshing of fuel in a tank) which will be a future task for us.

NOMENCLATURE

a_i	interfacial area concentration, 1/m
Ca	capillary number, $Ca \equiv \mu_\mathrm{L} U_\mathrm{B}/\sigma$, dimensionless
D_B	bubble diameter, m
D_h	hydraulic diameter, m
$\hat{\mathbf{e}}_y$	unit vector in y-direction, dimensionless
f	liquid volumetric fraction, dimensionless
$\mathbf{f}_\mathrm{pd}$	body force due to external pressure drop, N/m^3
$\mathbf{f}_\sigma$	surface tension force, N/m^3
g	gravitational acceleration, m/s^2
$\mathbf{g}$	gravity vector, m/s^2
h	mesh cell width, m
h^*	non-dimensional mesh cell width, $h^* \equiv h/2\,\mathrm{mm}$, dimensionless
L_e	entrance length, m
L_x, L_y, L_z	length of computational domain, m
$\hat{\mathbf{n}}_\mathrm{i}$	unit normal vector to interface pointing into the liquid, dimensionless
N_i, N_j, N_k	number of mesh cells, dimensionless
N_t	number of time steps, dimensionless
p	pressure, Pa
P	periodic part of pressure, Pa
$\Delta\tilde{p}$	non-hydrostatic pressure drop across L_y, Pa
Re_B	bubble Reynolds number, $Re_\mathrm{B} \equiv \rho_\mathrm{L} D_\mathrm{h} U_\mathrm{B}/\mu_\mathrm{L}$, dimensionless
t	time, s
Δt	time step width, s
Δt^*	non-dimensional time step width, $\Delta t^* \equiv \Delta t/0.75\,\mu\mathrm{s}$, dimensionless
U_B	bubble velocity, m/s
U_L	mean liquid velocity in unit cell, m/s
U_slug	mean axial velocity in liquid slug, m/s
U_wall	downward velocity of the walls in FLUENT computations, m/s
$\mathbf{v}_\mathrm{m}$	centre-of-mass velocity, m/s
V_G	volume of gas phase in domain, m^3
x, y, z	Cartesian coordinates, m
$\mathbf{x}$	position vector $\mathbf{x} = (x, y, z)^\mathrm{T}$, m

Greek symbols

α	local phase volume fraction, dimensionless
ε	overall gas volume fraction in domain, dimensionless
κ	interface curvature, 1/m

μ	dynamic viscosity, Pa s
ρ	density, $\mathrm{kg/m^3}$
σ	coefficient of surface tension, N/m

Subscripts

B	bubble
G	gas phase
i	interface
L	liquid phase
m	mixture quantity

ACKNOWLEDGEMENTS

The supports of EU Erasmus program and of Institute of Science and Technology at Sakarya University are gratefully acknowledged. The corresponding author also acknowledges the support of his main supervisor, Dr. Hakan S. Soyhan and his co-supervisor Dr. Martin Wörner during this thesis research. He is also grateful to A. Wenka for his contribution in Fluent simulations and for the support of M. Böttcher and S. Gordeev in getting into CFD simulations.

REFERENCES

1. Tsouris C, Porcelli JV. Process intensification—has its time finally come? *Chemical Engineering Progress* 2003; **10**:50–55.
2. Wegeng RS, Drost MK, Brenchley DL. Process intensification through miniaturization of chemical and thermal system in the 21st century. *Proceedings of the 3rd International Conference on Microreaction Technology* (*IMRET-3*), Ehrfeld W (ed.). Springer: Berlin, 2000; 2–13.
3. Charpentier J-C. Process intensification by miniaturization. *Chemical Engineering and Technology* 2005; **28**(3): 255–258.
4. Schubert K, Brandner J, Fichtner M, Linder G, Schygulla U, Wenka A. Microstructure devices for applications in thermal and chemical process engineering. *Microscale Thermophysical Engineering* 2001; **5**:17–39.
5. Pfeifer P, Haas-Santo K, Görke O, Bohn L, Schubert K. Fuel to hydrogen—an overview over fuel conversion activities at the institute for micro-process engineering. *Proceedings of the AIChE Spring National Meeting/IMRET-8*, Atlanta, U.S.A., 11–14 April 2005.
6. Jähnisch K, Baerns M, Hessel V, Ehrfeld W, Haverkamp V, Löwe H, Wille C, Guber A. Direct fluorination of toluene using elemental fluorine in gas/liquid microreactors. *Journal of Fluorine Chemistry* 2000; **105**(1):117–128.
7. Haverkamp V, Hessel V, Löwe H, Menges G, Warnier MJF, Rebrov EV, de Croon MHJM, Schouten JC, Liauw M. Hydrodynamics and mixer-induced bubble formation in microbubble columns with single and multiple channels. *Chemical Engineering and Technology* 2006; **29**(9):1015–1026.
8. Günther A, Khan SA, Thalmann M, Trachsel F, Jensen KF. Transport and reaction in microscale segmented gas–liquid flow. *Lab Chip* 2004; **4**:278–286.
9. Kreutzer MT, Kapteijn F, Moulijn JA, Ebrahimi S, Kleerebezem R, van Loosdrecht MCM. Monoliths as bioreactors: smart gas–liquid contacting for process intensification. *Industrial and Engineering Chemistry Research* 2005; **44**:9646–9652.
10. Bauer T, Guettel R, Roy S, Schubert M, Al-Dahhan M, Lange R. Modelling and simulation of the monolithic reactor for gas–liquid–solid reactions. *Chemical Engineering Research and Design* 2005; **83**:811–819.
11. Williams JL. Monolith structures, materials, properties and uses. *Catalysis Today* 2001; **69**:3–9.
12. Roy S, Bauer T, Al-Dahhan M, Lehner P, Turek T. Monoliths as multiphase reactors: a review. *AIChE Journal* 2004; **50**:2918–2938.
13. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**(2):335–354.

14. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
15. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
16. Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan Y-J. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics* 2001; **169**:708–759.
17. Scheuerer M, Heitsch M, Menter F, Egorov Y, Toth I, Bestion D, Pigny S, Paillere H, Martin A, Boucker M, Krepper E, Willemsen S, Muhlbauer P, Andreani M, Smith B, Karlsson R, Henriksson M, Hemstrom B, Karppinen I, Kimber G. Evaluation of computational fluid dynamic methods for reactor safety analysis (ECORA). *Nuclear Engineering and Design* 2005; **235**:359–368.
18. Thulasidas TC, Abraham MA, Cerro RL. Bubble-train flow in capillaries of circular and square cross-section. *Chemical Engineering Science* 1995; **50**:183–199.
19. Ghidersa B, Wörner M, Cacuci DG. Exploring the flow of immiscible fluids in a square vertical mini-channel by direct numerical simulation. *Chemical Engineering Journal* 2004; **101**(1–3):285–294.
20. Wörner M, Ghidersa B, Onea A. A model for the residence time distribution of bubble-train flow in a square mini-channel based on direct numerical simulation results. *International Journal of Heat and Fluid Flow* 2007; **28**:83–94.
21. Wörner M, Sabisch W, Grötzbach G, Cacuci DG. Volume-averaged conservation equations for volume-of-fluid interface tracking. *Proceedings of the 4th International Conference on Multiphase Flow*, New Orleans, LA, U.S.A., CD-ROM, 27 May –1 June, 2001.
22. Sabisch W, Wörner M, Grötzbach G, Cacuci DG. 3D volume-of-fluid simulation of a wobbling bubble in a gas–liquid system of low Morton number. *Proceedings of the 4th International Conference on Multiphase Flow*, New Orleans, LA, U.S.A., CD-ROM, 27 May–1 June, 2001.
23. Wörner M, Ghidersa BE, Ilić M, Cacuci DG. Volume-of-fluid method based numerical simulations of gas–liquid two-phase flow in confined geometries. *Houille Blanche-Revue Internationale De L Eau* 2005; (6):91–104.
24. Ubbink O, Issa RI. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics* 1999; **153**:26–50.
25. Muzaferija S, Peric M. Computation of free surface flows using interface-tracking and interface capturing methods. In *Nonlinear Water Wave Interaction*, Mahrenholtz O, Markiewicz M (eds). WIT Press: Southampton, 1999; 59–100.
26. STAR-CD Version 4.00 Methodology, CD-adapco, 2006.
27. CFX 10 User Manual. ANSYS CFX, 2005.
28. Taha T, Cui ZF. CFD modelling of slug flow inside square capillaries. *Chemical Engineering Science* 2006; **61**:665–675.
29. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: New York, 1982; 273–285.
30. FLUENT 6.2 User's guide. Fluent Inc., 2005.

DOI: 10.1002/fld