Evaluation of commercial and academic CFD codes for a two-phase flow benchmark test case

S. Hysing*and S. Turek†

Abstract

The performance of two commercial simulation codes, Ansys Fluent and Comsol Multiphysics, is thoroughly examined for a recently established two-phase flow benchmark test case. In addition, the commercial codes are directly compared with the newly developed academic code, FeatFlow TP2D. The results from this study show that the commercial codes fail to converge and produce accurate results, and leave much to be desired with respect to direct numerical simulation of flows with free interfaces. The academic code on the other hand was shown to be computationally efficient, produced very accurate results, and outperformed the commercial codes by a magnitude or more.

1 Introduction

Commercial software tools are widely used today by industrial engineers to simulate various physical processes. Except for cost, they offer many benefits over academic tools; commercial codes are reasonably easy to use, are often documented extensively, have user support, and can produce qualitatively good results. However, what is often not known is how accurate these codes really are, on an absolute level, and what performance can be expected for a specific problem. Mirroring a recent study which examined the performance of CFD codes for single phase flows [1], accuracy and performance when computing two-phase flows was examined by simulating a corresponding numerical benchmark test case. Two different commercial codes were employed, the general and flexible multipurpose simulation package Comsol Multiphysics and the dedicated computational fluid dynamics (CFD) flow solver Ansys Fluent. The results from these codes were also compared with those computed with a recently developed academic code, FeatFlow TP2D [11]. The test problem was chosen to examine the capabilities of the different codes to accurately simulate two-phase fluid flows with immersed interfaces, and consisted of directly tracking the evolution of a bubble rising and deforming in a liquid column [12]. For this benchmark test case accurate reference solutions have been established [17] which enabled quantitative validation and comparison of the codes.

*Department of Mathematics, Shanghai Jiaotong University, 800 Dongchuan Road, Shanghai 200240, China (shuren.hysing@udo.edu)

†Institute of Applied Mathematics (LS III), TU Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany (stefan.turek@math.tu-dortmund.de).

This paper is organized as follows. Section 2 outlines the employed numerical benchmark test case which consists of directly tracking a rising and deforming bubble. The employed commercial software codes are then described and their performance on this test case is evaluated in Section 3. Section 4 analogously treats the academic code FeatFlow TP2D, after which the paper is concluded in Section 5 with a discussion comparing all the codes.

2 Benchmark test case

Rigorous validation of simulation software and modeling approaches attempting to describe and predict real physical phenomena involves two very important and distinct steps. Firstly, one must verify that the numerical approach solves the posed mathematical problem correctly. And secondly, one should verify that the mathematical model and the corresponding numerical solution agrees with the physical phenomena to be predicted, which is generally done by comparing with experimental data. Unfortunately, the first step is often less rigorously performed even though a number of numerical benchmark test cases have been published for which very accurate reference solutions are available [3, 15, 16].

With this in mind, a purely numerical test case was chosen to test the codes which involves simulating and tracking the evolution of a bubble rising in a liquid column. This test case has been throughly studied by three independent research groups who also have established accurate reference solutions for the bubble shape as well as other quantitative measures [12]. Initially a bubble with radius $r = 0.25$ is at rest in the in the lower half of a 1×2 rectangular domain. The bubble is tracked for 3 time units as the buoyancy force causes the bubble to rise and deform. Aside from computing the shape of the bubble, the center of mass, bubble rise velocity, and degree of circularity are also measured. The circularity is in two dimensions defined as

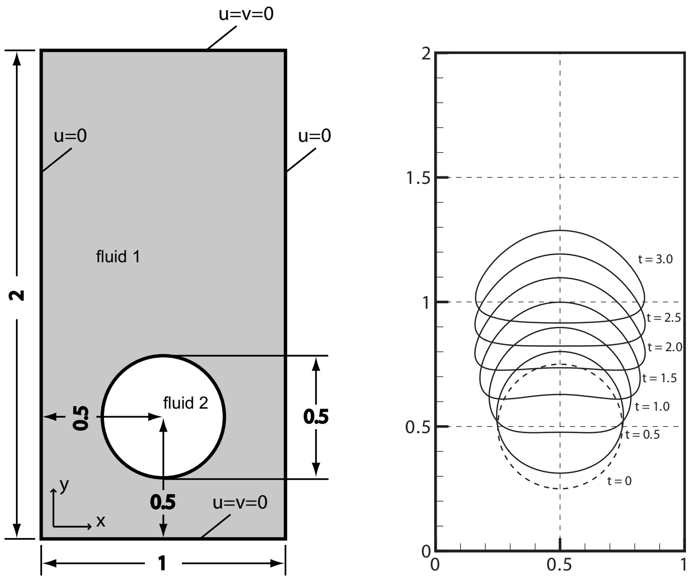
$$\not{c} = \frac{P_a}{P_b} = \frac{\text{perimeter of area-equivalent circle}}{\text{perimeter of bubble}} = \frac{\pi d_a}{P_b}$$

where P_a denotes the perimeter or circumference of a circle with diameter d_a which has an area equal to that of a bubble with perimeter P_b. For a perfectly circular bubble the circularity is equal to unity and will decrease as the bubble is deformed.

Figure 1(a) shows the initial and boundary conditions while Figure 1(b) depicts the expected evolution of the bubble interface contour. The physical parameters of the two fluids and dimensionless Reynolds, Re, and Eötvös, Eo, numbers which define the test case are given in Table 1. A subscript 1 is assigned to the heavier surrounding fluid and 2 to the lighter fluid of the bubble.

ρ_1	ρ_2	μ_1	μ_2	g	σ	Re	Eo	ρ_1/ρ_2	μ_1/μ_2
1000	100	10	1	0.98	24.5	35	10	10	10

Table 1: Physical parameters and the dimensionless Re and Eo numbers defining the test case (ρ denotes density, μ viscosity, g the gravitational constant, and σ the coefficient of surface tension).



(a) Initial configuration (b) Bubble evolution

Figure 1: Initial configuration, boundary conditions, and bubble evolution.

3 Commercial software tools

This section explores the accuracy and performance of the two commercial codes, Comsol Multiphysics and Ansys Fluent for simulating the chosen benchmark test problem.

3.1 Comsol Multiphysics

The Comsol Multiphysics software suite (previously marketed under the name Femlab) is a finite element software package for solving coupled systems of partial differential equations (PDEs) [4]. Although the software is very user friendly, has a nice graphical user interface, and very easily allows for almost arbitrary PDE based problems to be treated, the monolithic fully coupled approach and heavy dependence on direct solvers limits its practical use to rather small problem sizes. Despite this, Comsol Multiphysics was applied to the benchmark test case in order to establish what a general multipurpose commercial simulation tool, not optimized for CFD problems, can accomplish.

The following simulations were performed with version 3.3a of the Comsol Multiphysics Chemical Engineering Module coupled with the conservative level set application mode which is used to model two-phase flow phenomena [13, 14]. A pure Cartesian quadrilateral tensor product grid was employed in all calculations with continuous biquadratic and discontinuous linear finite element basis functions, the Q_2P_1 Stokes elements, discretizing the velocity and pressure. The level set function used to track the interface and the different fluid phases was correspondingly discretized with conforming 2nd order Q_2 basis functions.

All the following computations (including those of Fluent and TP2D for comparison purposes) were performed on a compute server with a 2.0 GHz Intel Core2Duo processor for which simulation statistics are given in Table 2. The first column, $1/h$, shows the reciprocal of the cell size which is equal to the number of cells resolving the width of the computational domain. The total number of computational cells or elements is denoted by NEL, the number of degrees of freedom by NDOF, and the number of time steps by NTS. The computational effort required can be seen from the peak memory consumption in Megabytes (MEM) and the required time to complete the simulations (CPU).

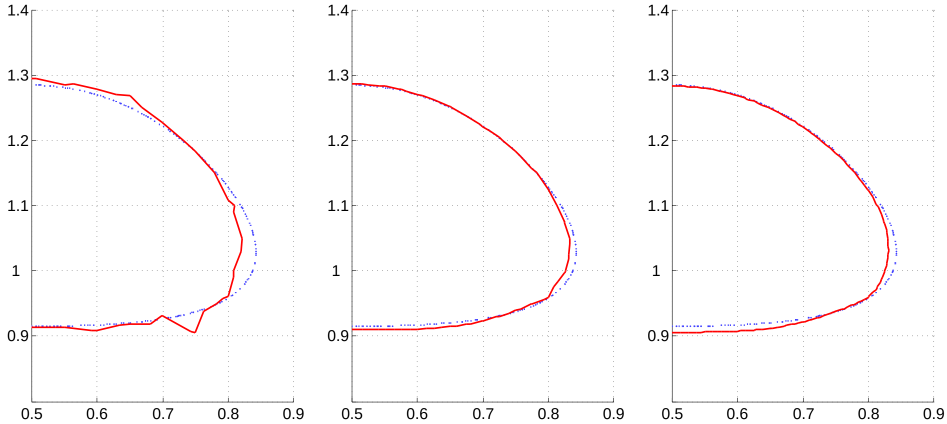
$1/h$	NEL	NDOF	NTS	MEM	CPU
20	800	13163	277	175	101
40	3200	51923	115	446	483
60	7600	116283	120	890	986
80	12800	206243	120	1568	2361

Table 2: Simulation statistics and timings for Comsol Multiphysics.

Although a comprehensive selection of iterative linear solvers is included in Comsol Multiphysics, the default and most robust choice is to use a direct solver, in this case UMFPACK [5]. The magnitude of the peak memory consumption, although scaling linearly with the number of degrees of freedom, was very high due to the fully coupled approach. It was in fact impossible to obtain a solution for anything larger than a 80×160 grid, even when switching to the iterative solvers, which either failed to converge or still allocated too much memory. The adaptive time stepping scheme on the other hand worked very well, only requiring about 120 time steps to complete each simulation for all but the coarsest grid. The underlying algorithm employed the variable order DAE solver DASPK (where up to fifth order accuracy was allowed) [2].

The bubble shapes at the final time (t=3) can be seen in Figure 2 (due to symmetry only the right half is shown even though all computations were performed over the whole domain). On the coarsest 20×40 grid the computed bubble shape is reasonably well aligned with that of the reference solution but exhibits a very jagged and unphysical contour (Figure 2(a)). Refining the grids improved the computed shapes significantly but seemed to result in slightly more rounded contours than that of the reference solution (Figure 2(b)). Even the computation on the finest grid did not manage to produce perfectly aligned solution (Figure 2(c)).

The use of the reference benchmark quantities from Section 2 and [12] makes it much easier to spot convergence trends and judge accuracy quantitatively than merely examining the bubble shapes. The circularity of the bubble is therefore also computed and compared against the established reference curve (Figure 3). The results for the two coarsest grids, 20×80 and 40×80, show very oscillatory behaviors for which the means deviate significantly from the reference curve. The curves corresponding to the two finer grids behave better but show an increasing circularity towards the end of the simulation (after $t = 2.5$) instead of converging to the stable shape indicated by the blue reference solution.



(a) 20×40 (b) 40×80 (c) 80×160

Figure 2: Bubble shapes computed with Comsol Multiphysics on different grids (solid red), and a reference solution (dashed blue).

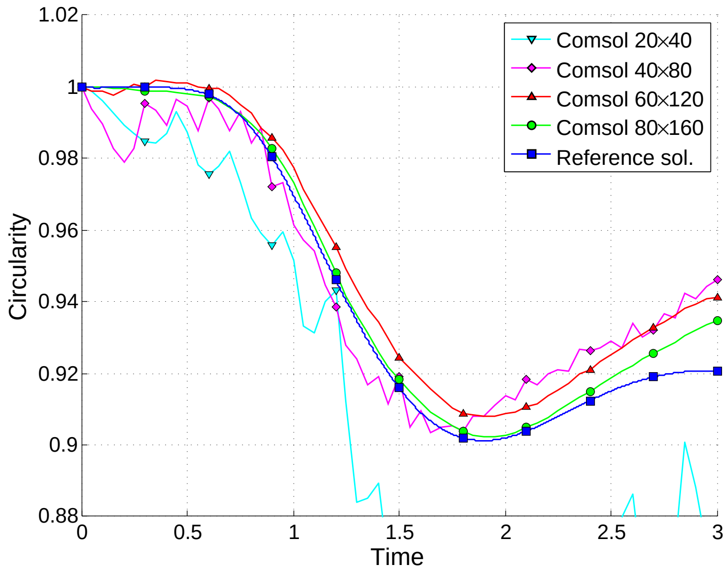


Figure 3: Computed circularity curves for Comsol Multiphysics.

Table 3 shows the inflection points and minimum circularity, $\not{c}_{min}$, with corresponding incidence times, $t|_{\not{c}=\not{c}_{min}}$, the absolute error at the minimum, $e_{\not{c}_{min}}$, and also the time averaged relative error of the circularity, $|e_{\not{c}}|$. Except for the very coarsest grid, the minimum circularity is quite close to the reference value, with an error of $1.0{\cdot}10^{-3}$ on the finest grid. The corresponding incidence times fluctuate somewhat and one would ideally like to have more data for finer grid levels to really be able to establish a convergence trend. The time averaged relative error is three times larger than that of the minimum on the finest grid due to the diverging behavior after $t = 2.5$.

$1/h$	$\not{c}_{min}$	$e_{\not{c}_{min}}$	$t\|_{\not{c}=\not{c}_{min}}$	$\|e_{\not{c}}\|$
20	0.8402	0.0610	2.25	$3.6 \cdot 10^{-2}$
40	0.9034	-0.0022	1.65	$1.0 \cdot 10^{-2}$
60	0.9081	-0.0069	1.90	$8.0 \cdot 10^{-3}$
80	0.9022	-0.0010	1.95	$2.9 \cdot 10^{-3}$
$Ref.$	0.9012		1.90	

Table 3: Minimum circularity, $\not{c}_{min}$, and error, $e_{\not{c}_{min}}$, with corresponding incidence times, $t|_{\not{c}=\not{c}_{min}}$, and time averaged error, $|e_{\not{c}}|$, for Comsol Multiphysics ($Ref.$ indicates reference values).

These tests have shown that although the Comsol Multiphysics package actually can simulate two-phase flows some real difficulties do exist. Firstly, the results do not seem to fully converge towards the reference solution for longer time periods, and secondly the monolithic coupled approach with the direct methods used to solve and invert the discretized matrices consume far too much memory to be able to run simulations with even moderately dense grids.

3.2 Ansys Fluent

The CFD package Ansys Fluent [6] has been marketed as "the world leader in Computational Fluid Dynamics" [7] and "a state-of-the-art computer program for modeling fluid flow and heat transfer in complex geometries" [8], and thus has a lot to live up to. Fluent includes a comprehensive set of models to simulate various flow related phenomena such as heat transfer, turbulence, combustion, chemical reactions, and also multiphase flows. Flows with immiscible fluids are treated with the Eulerian volume of fluid (VOF) methodology which employs the use of a scalar volume fraction function indicating the relative amounts of the fluids present in each computational cell.

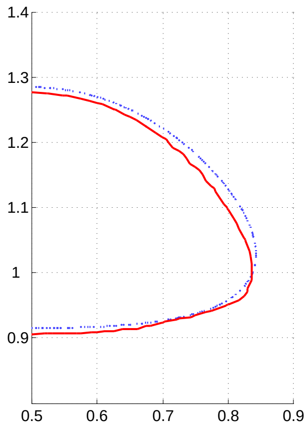
Fluent employs a finite volume discretization in space with unknowns located at the cell centers. In the time domain there are a number of discretization schemes to choose from, of which the recommended implicit Fractional step operator spitting scheme has been used in the following tests. This scheme, which is a form of projection method, effectively separates the solution of the pressure from the velocity calculations, thus saving computational effort in contrast to the direct monolithic approach of Comsol Multiphysics. To solve the arising linear equation systems Fluent employs an algebraic multigrid approach. In the following, version 6.3 of Ansys Fluent is used to perform benchmark tests identical to the ones previously done with Comsol Multiphysics.

$1/h$	NEL	NDOF	NTS	MEM	CPU
40	3200	12800	150	96	45
80	12800	51200	480	111	254
160	51200	204800	1200	210	2106
320	204800	819200	3000	439	21091

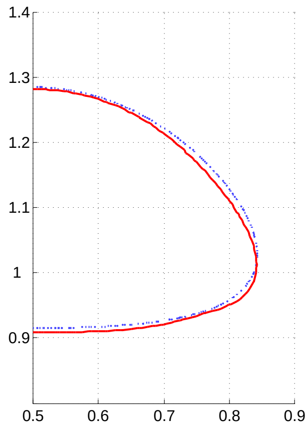
Table 4: Simulation statistics and timings for Ansys Fluent.

The simulation statistics and timings for Fluent can be seen in Table 4. Compared to Comsol Multiphysics, Fluent allocated significantly less memory and allowed the use of finer grids. However, it should be pointed out that since Fluent uses cell centered degrees of freedom the total number of unknowns is four times fewer than used by the Comsol software for a grid of the same size (Comsol Multiphysics also allows for higher accuracy with the employed Q_2 finite element discretization). The time steps were selected so that the capillary time step restriction was respected. Comparing the required CPU effort on the (40×80) and (80×160) grids with those of Comsol Multiphysics (compare Tables 2 and 4) one can see that the operator splitting approach used by Fluent is much less CPU intensive even though the simulations required more time steps to finish.

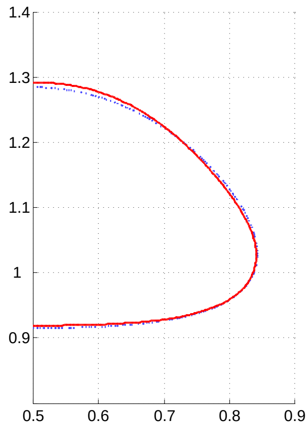
Figure 4 shows the computed bubble shapes at the final time (t=3). The computation on the coarsest grid (Figure 4(a)) produced a result which deviated significantly from the reference bubble, and one can see that there is a lot of room for improvement. Refining the grid allowed the results to converge better towards the reference shape which is evident from Figure 4(b). From examining the solution from the finest grid (320×640) it is hard to see any significant differences between the computed and the reference solution (see Figure 4(c)).



(a) 40×80



(b) 80×160



(c) 320×640

Figure 4: Bubble shapes computed with Ansys Fluent on different grids (solid red), and a reference solution (dashed blue).

Since the final shape was simulated quite accurately one might expect that the overall temporal evolution also is correct. However, if one looks at the curves for the circularity (Figure 5) one can see that this is actually not the case. Although mesh independent solutions were obtained with the two finest grids, they do not converge at all towards the reference solution. It is apparently a period around the maximum deformation, between $t = 1.2$ and $t = 2.5$ (corresponding to the point of minimum circularity), that causes the most difficulty for Fluent. The curves corresponding to the calculations on the two finest grids do seem to converge better towards the reference solution after $t = 2.75$, however, it is more likely that the solutions will diverge again from $t = 3$ and onward considering the large error around the time of minimum circularity.

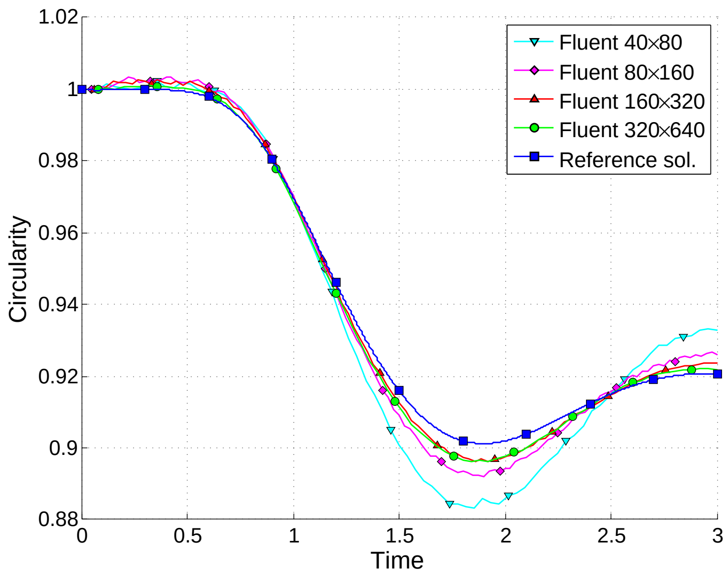


Figure 5: Computed circularity curves for Ansys Fluent.

The maximum and time averaged errors of the circularity are quite large as can be seen from Table 5. The values from the two finest grids show that a mesh independent solution has indeed been obtained. However, this solution does not converge towards the reference solution. The minimum circularity is predicted to occur slightly too late with a smaller value than expected. Comparing these errors with those of Comsol Multiphysics (Table 3) one can see that both codes achieve quite similar levels of accuracy. Fluent shows slightly better accuracy in in the averaged error norm while Comsol achieves better values for the minimum circularity.

$1/h$	$\not{c}_{min}$	$e_{\not{c}_{min}}$	$t\vert_{\not{c}=\not{c}_{min}}$	$\vert e_{\not{c}}\vert$
40	0.8834	0.0178	1.86	$8.2{\cdot}10^{-3}$
80	0.8922	0.0090	1.90	$4.3{\cdot}10^{-3}$
160	0.8962	0.0050	1.92	$2.4{\cdot}10^{-3}$
320	0.8963	0.0049	1.92	$2.3{\cdot}10^{-3}$
$Ref.$	0.9012		1.90	

Table 5: Minimum circularity, $\not{c}_{min}$, and error, $e_{\not{c}_{min}}$, with corresponding incidence times, $t|_{\not{c}=\not{c}_{min}}$, and time averaged error, $|e_{\not{c}}|$, for Ansys Fluent ($Ref.$ indicates reference values).

To conclude the study of commercial simulation tools recall the curves for the circularity (Figures 3 and 5). It unfortunately seems that if one desires a good final solution one should choose Fluent, and if one prefers a better intermediate solution one ought to use Comsol Multiphysics. Thus neither of the commercial codes produce overall satisfactory results, and this is for relatively simple test case of a single rising bubble not undergoing break up or separation.

4 Academic software tools

Academic software tools utilize novel and experimental algorithms in contrast to commercial tools which generally apply tried and tested routines. Consequently, it would be interesting to see if there exist any significant differences in the performance and accuracy between academic codes and commercial software. The benchmark test case is therefore also used to measure the performance of an academic two-phase flow code.

4.1 FeatFlow TP2D

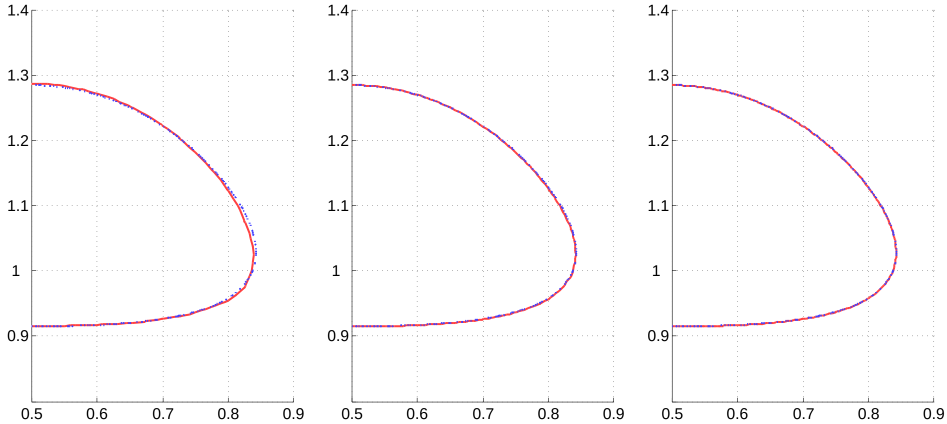
FeatFlow TP2D is a newly developed code designed to accurately be able to simulate immiscible fluid flows. This approach essentially consists of combining a non-conforming finite element flow solver with a conforming level set interface tracking method and incorporating the surface tension forces semi-implicitly [10]. In TP2D the velocity and pressure fields are discretized with non-conforming Rannacher-Turek $\tilde{Q}_1 Q_0$ finite elements which are decoupled with a discrete projection method and solved for separately. The level set function used to track the bubble interface is discretized with conforming Q_1 finite elements and solved independently with a FEM-TVD solver [9]. This technique, although somewhat unconventional, has resulted in a simulation code which has proved to be able to simulate two-phase flows with free interfaces both very accurately and efficiently [11].

The simulation statistics and timings for TP2D are shown in Table 6. Compared to the commercial codes, the required CPU time and used memory was significantly smaller with respect to both grid size and the total number of degrees of freedom (compare Table 6 with Tables 2 and 4).

$1/h$	NEL	NDOF	NTS	MEM	CPU
40	3200	19561	150	15	15
80	12800	77521	450	55	185
160	51200	308641	1000	212	1674

Table 6: Simulation statistics and timings for FeatFlow TP2D.

As can be seen from Figure 6 the resulting bubble shapes and curves for the benchmark quantities were in fact virtually indistinguishable from the reference solution on all but the very coarsest grid.



(a) $h = 1/40$ (b) $h = 1/80$ (c) $h = 1/160$

Figure 6: Bubble shapes computed with FeatFlow TP2D on different grids (solid red), and a reference solution (dashed blue).

From examining the curves for the circularity (Figure 7) it is clear that TP2D produced very good solutions during the whole time interval. The solution on the very coarsest grid exhibited a few wiggles in the beginning and towards the end of the simulation but otherwise followed the reference curve nicely. Comparing the TP2D curves with those of Comsol and Fluent (compare Figure 7 with Figures 3 and 5) it is very apparent that the commercial codes cannot produce the same overall level of accuracy as the academic approach.

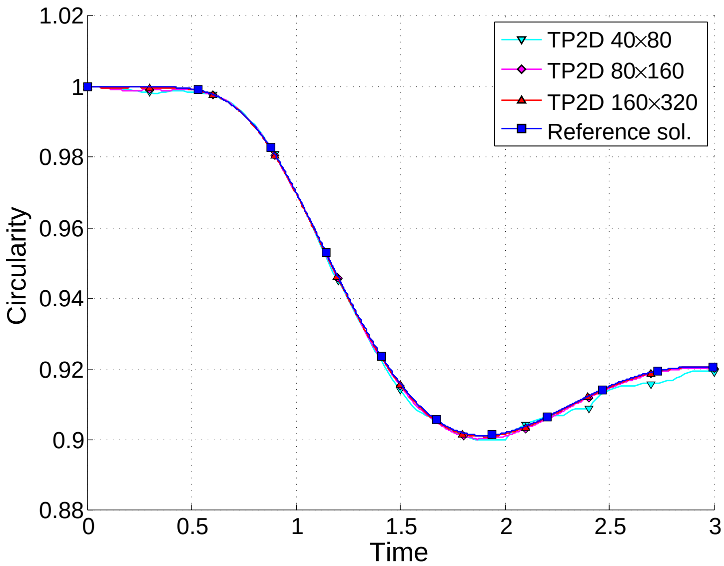


Figure 7: Computed circularity curves for FeatFlow TP2D.

The resulting error levels together with reference values are shown in Table 7. From these it is clear that TP2D is able to produce very accurate results and converges towards the reference values with respect to both the magnitude of the minimum circularity, $\not{c}_{min}$, and the corresponding incidence time, $t|_{\not{c}=\not{c}_{min}}$. The time averaged error in the circularity, $|e_{\not{c}}|$, also converged well, and most notable is that, even for the very coarsest grid, the errors were significantly smaller than anything that the commercial codes could achieve.

$1/h$	$\not{c}_{min}$	$e_{\not{c}_{min}}$	$t\vert_{\not{c}=\not{c}_{min}}$	$\vert e_{\not{c}}\vert$
40	0.9002	0.0010	1.88	$7.2{\cdot}10^{-4}$
80	0.9007	0.0005	1.88	$2.8{\cdot}10^{-4}$
160	0.9010	0.0002	1.91	$1.9{\cdot}10^{-4}$
$Ref.$	0.9012		1.90	

Table 7: Minimum circularity, $\not{c}_{min}$, and error, $e_{\not{c}_{min}}$, with corresponding incidence times, $t|_{\not{c}=\not{c}_{min}}$, and time averaged error, $|e_{\not{c}}|$, for FeatFlow TP2D ($Ref.$ indicates reference values).

5 Conclusions

In this study two modern commercial simulation tools have been directly compared with a newly developed academic code to assess their capabilities to simulate two-phase flows with immersed interfaces. The chosen test problem was a rising bubble benchmark test case for which an accurate reference solution has been established. Computations were performed on different grid levels while measuring the required CPU time and simultaneously calculating the error in the circularity.

Figure 8 shows the time averaged error against the CPU time for the different codes. It is clear that the solution produced by Comsol Multiphysics initially had quite a large error but also converged at a high rate due to the higher order Q_2P_1 finite element discretization. Unfortunately, solutions at very fine grids were practically impossible to compute due to the strong dependence on direct solvers. Ansys Fluent on the other hand started with a somewhat lower initial error but converged much slower. By the third grid level Fluent and Comsol had achieved roughly the same level of efficiency and further refinements yielded no improvements at all, Fluent completely stopped converging. The academic TP2D code converged with first order and showed a much better overall efficiency, requiring about ten times less effort to achieve a certain accuracy than the commercial codes would had they been able to compute on finer grids. Note that even the error on the very coarsest grid was already lower than anything that either of the commercial codes could produce.

Altogether, the newly developed FeatFlow TP2D simulation code has been rigorously validated together with two commercial codes by simulating a reference benchmark test case and comparing the resulting performance. The comparison highlighted real problems in the commercial codes and showed the merit of the academic approach which was able to outperform them by a magnitude or more.

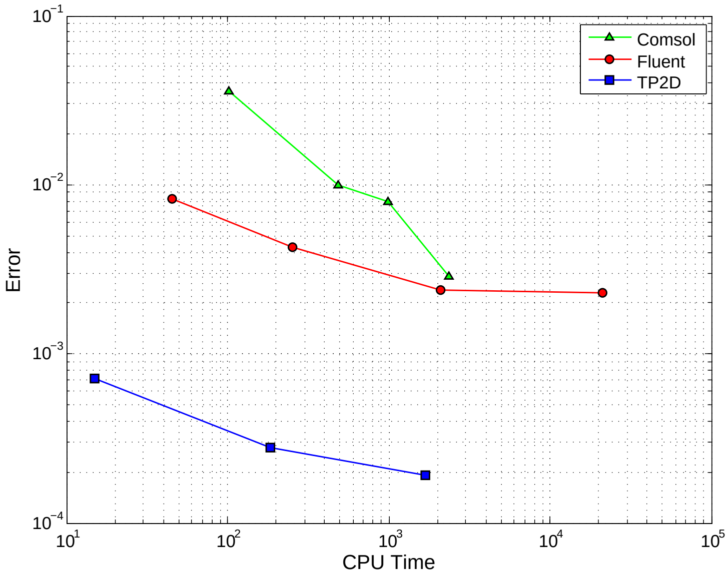


Figure 8: Averaged error in the circularity vs. CPU time.

ACKNOWLEDGEMENTS

The author would like to thank the German Research foundation (DFG) and the EU for supporting this work under grants Paketantrag PAK178 (Tu102/27-1, Ku1530/5-1), Sonderforschungsbereich SFB708 TP B7, and STF2/41+CRIS: 235-833.

References

[1] Bayraktar E, Mierka O, Turek S. Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow. *International Journal of Computational Science and Engineering*, 2012.

[2] Brown PN, Hindmarsh AC, Petzold LR. Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM Journal on Scientific Computing*, 1994; **15**:1467–1488, doi:10.1137/0915088.

[3] Christon MA, Gresho PM, Sutton SB. Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution). *International Journal for Numerical Methods in Fluids*, 2002; **40**(8):953–980, doi:10.1002/fld.395.

[4] Comsol Multiphysics product web site: http://www.comsol.com/

[5] Davis TA. Algorithm 832: UMFPACK V4.3, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 2004; **30**(2):196–199, doi:10.1145/992200.992206.

[6] Ansys Fluent product web site: http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/ANSYS+Fluent

[7] *Fluent announces first international CFD conference dedicated to the oil & gas industry*, Press release, Fluent Europe Ltd., Sheffield, UK, 7th April 2006.

[8] *FLUENT 6.3 Getting Started Guide*, Fluent Inc. 2006.

[9] Kuzmin D., Löhner RL, Turek S. Flux-Corrected Transport: Principles, Algorithms, and Applications. Scientific Computation, Springer, 2005, ISBN: 978-3-540-23730-3.

[10] Hysing S. A new implicit surface tension implementation for interfacial flows. *International Journal for Numerical Methods in Fluids* 2006; **51**(6):659–672, doi:10.1002/fld.1147.

[11] Hysing S. Mixed element FEM level set method for numerical simulation of immiscible fluids. *Journal of Computational Physics* 2012; **231**(6):2449–2465, doi:10.1016/j.jcp.2011.11.035.

[12] Hysing S, Turek S, Kuzmin D, Parolini N, Burman E, Ganesan S, Tobiska L. Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids* 2009; **60**(11):1259–1288, doi:10.1002/fld.1934.

[13] Olsson E, Kreiss G. A conservative level set method for two phase flow. *Journal of Computational Physics* 2005; **210**(1):225–246, doi:10.1016/j.jcp.2005.04.007.

[14] Olsson E, Kreiss G, Zahedi S. A conservative level set method for two phase flow II. *Journal of Computational Physics* 2007; **225**(1):785–807, doi:10.1016/j.jcp.2006.12.027.

[15] Turek S, Hron J. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. Lecture Notes in Computational Science and Engineering, **53**, 371–385, Fluid-Structure Interaction - Modelling, Simulation, Optimization, Springer, ISBN: 3-540-34595-7, 2006.

[16] Turek S, Schäfer M. Benchmark computations of laminar flow around cylinder. Flow Simulation with High–Performance Computers II (Notes on Numerical Fluid Mechanics), **52**, 547–566, Vieweg, 1996.

[17] Rising bubble benchmark reference data:
http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble.html