

QUANTITATIVE BENCHMARK OF A SINGLE RISING BUBBLE USING VOF METHODS

LIONEL GAMET¹, JOHAN ROENBY², MARCO SCALA¹, HAMZA ZEHARA¹

¹IFPEN, Process Experimentation Division,

Lionel.Gamet@ifpen.fr, Marco.Scala@ifpen.fr, Hamza.Zehara@ifpen.fr

²STROMNING, *johan@stromning.com*

Keywords: *Multiphase flows, Bubbly flows, Incompressible, MULES, isoAdvector, VOF*

1 Introduction

Gas-liquid flow systems are often encountered in the chemical process industry, with various scales ranging from large bubble columns, plate columns, agitated vessels, surface aerators, jets, static mixers or micro-reactors. Their applications are generally reactive flow systems, mixing, stripping or saturation systems. Complex phenomena of turbulent hydrodynamics involving breakup and coalescence, coupled with gas-liquid mass and heat transfers, appear in such systems. Thanks to the increase of computing resources, highly resolved simulations gain more and more interest to analyze in detail the physics of such multiphase flows. Many numerical methods have emerged to attempt to simulate gas-liquid flows. Among these, implicit interface capturing approaches like volume of fluid (VoF) [1] and level set (LS) [2] have proven to be efficient in simulating multiphase flows.

Before running into complex geometries, elementary quantitative benchmark configurations are essential for validation and comparison of interfacial flow solvers. Hysing et al. [3] have proposed a 2D benchmark consisting in a single rising bubble in a quiescent liquid. Two different cases are described in [3], corresponding to different density, viscosity and surface tension ratios. More recently, Adelsberger et al. [4] have published a 3D equivalent of the same benchmark. For both these benchmarks, result data was made available online by the authors at the URLs mentioned in the bibliography.

Roenby et al. [5] have recently developed a new geometric VoF method, called isoAdvector, for advecting the interface between two incompressible fluids. This method was implemented in OpenFOAM-v1706 in the new solver `interIsoFoam`. IsoAdvector gives a sharper interface, but validation data for this new method is still sparse, especially for surface tension dominated flows. Following the single rising bubble benchmarks [3, 4], the objective of this paper is to perform quantitative comparisons of the isoAdvector method with the benchmark data. Furthermore, the isoAdvector results are compared with results from OpenFOAM's original interfacial flow solver, `interFoam`.

2 Numerical methods

We consider here an unsteady, laminar and incompressible two-phase flow. The governing equations are the continuity equation for the fluid-wise constant mass density field, the Navier-Stokes equations for the momentum evolution, and the incompressibility condition for the velocity field.

In the VoF method, the continuity equation is converted into an evolution equation for the volume fraction field α , representing for each cell the fraction of its volume, which is occupied by one of the two fluids. Special care must be taken in the numerics to prevent smearing of the α -field and at the same time keeping it bounded ($0 \leq \alpha \leq 1$). In the `interFoam` solver, sharpness is obtained by introducing an artificial interface compression term in the α -equation [6], and boundedness is ensured by employing the MULES limiter (Multidimensional Universal Limiter with Explicit Solution). More details can be found in Deshpande [7].

The solver `interFoam` has been widely used and validated [8, 9, 10, 11], but under some conditions the described method may fail in keeping the interface sufficiently sharp. Furthermore, the heuristic nature of the added compression term can lead to inaccurate interface advection and undesirable features such as unphysical ripples on the interface [12, 13]. This motivated the development of the isoAdvector geometric VoF method, which was first presented by Roenby et al. [5]. Here it was tested with a variety of pure advection cases yielding very good results in terms of volume conservation, interface sharpness, boundedness and shape preservation. With recent improvements, the method has been made consistently second order for all mesh types (See Scheufler and Roenby [14]). The isoAdvector method implements new ideas in both the interface reconstruction step and the interface advection step. The reconstruction step uses efficient isosurface calculations to compute the distribution of fluids in a grid cell. The interface advection step uses a novel division of a physical time step into sub-time steps on which the volume fraction flux through a cell face can be calculated analytically under the assumption that the interface is moving steadily across the face during the interval. In

the development of this procedure, no assumptions are made on the shape of a cell face, which makes the advection step applicable on arbitrary meshes.

Except for the interface advection step, the `interIsoFoam` (`isoAdvector`) solver is identical to the `interFoam` (`MULES`) solver. They both solve the governing system of equations in a segregated manner using the PIMPLE algorithm (a combination of the SIMPLE and PISO algorithms) for pressure-velocity coupling. Strictly speaking, `isoAdvector` and `MULES` also differ in the way `rhoPhi` (used in the momentum convection term) is calculated, which is described in [13].

3 Definition of test case

The test case number 2 as described by Hysing et al. [3] has been used here. We have used only this second test case as it is more representative of final industrial applications. The 2D case setup is schematized in Figure 1. First phase (liquid) properties are $\rho_1 = 1000 \text{ kg/m}^3$, $\mu_1 = 10 \text{ kg/(ms)}$, while second phase (gas) takes $\rho_2 = 1 \text{ kg/m}^3$, $\mu_2 = 0.1 \text{ kg/(ms)}$ as physical parameters. The surface tension is $\sigma = 1.96 \text{ kg/s}^2$. Gravity is taken as $g = 0.98 \text{ m/s}^2$. Extension of the case to 3D is straightforward.

Square/cubic cells were used for all simulations in 2D/3D. Six grid resolutions have been used ranging from 20 to 640 cells along the x direction, with the cell size halved in each refinement. The bubble was initialized as a cylinder in 2D and as a sphere in 3D using the `setFields` utility. The Crank-Nicolson second order time scheme with blending coefficient 0.9 has been used for all computations. The convective term was treated with `Gauss limited linearV 1`, and in the `MULES` simulations `Gauss vanLeer` was used for the α convection. The `Gauss linear` scheme was used for gradient terms. The GAMG implicit solver was used for pressure terms, while the smooth solver was used for the velocity. Constant time steps have been used, starting at $\Delta t = 0.002 \text{ s}$ for the coarsest level and reduced for finer grids to keep the maximum CFL number below 0.05. The PIMPLE algorithm was run in PISO mode (`nOuterCorretors` set to 1) with 3 PISO correctors. Setting `momentumPredictor` to `true` was important for accuracy with `isoAdvector`. Computations were run up to time $t = 3 \text{ s}$ in 2D and $t = 3.5 \text{ s}$ in 3D.

4 Results

Post-processing quantities of interest are described in details in [3, 4]. These are the vertical position of the bubble centroid, the bubble rise velocity, the bubble circularity/sphericity, bubble area/volume and surface perimeter/area in 2D/3D. The circularity/sphericity is the inverse ratio of the bubble surface perimeter/area to the area/volume-equivalent circle/sphere in 2D/3D. It takes the value 1 at the beginning of the computation and decreases as the bubble deforms.

The Figure 2 shows the bubble shape obtained with `MULES` and `isoAdvector` for the finest 2D simulation at time $t = 3 \text{ s}$ together with results from the reference data from [3]. Slight differences appear in the prediction of the front and back main interface positions (see zooms 1 and 2), with the most noticeable difference being `isoAdvector` falling slightly behind the other methods. All the codes predict rather different trains of detached bubbles (see zoom 3 frame). The time evolution of Figure 3 shows a comparison between `MULES` and `isoAdvector` at grid resolution 160, and for the 2D calculations. Again results are very similar, especially for the main bubble shape, and we see a small retardation of the `isoAdvector` bubble compared with the `MULES` bubble towards the end of the simulation.

Figure 4 shows 3D results for the rise velocity and sphericity. `MULES` and `isoAdvector` results are almost identical. When compared with the available reference data, the bubble rise velocity is, however, overpredicted by both solvers. This discrepancy will require further investigation, as the results obtained in reference [4] using `interFoam` version 2.2.2 differ from the currently used version taken from `OpenFOAM-v1712`. The right hand side of Figure 4 shows that sphericity results start to differ between `MULES` and `isoAdvector` at times larger than roughly 2.5 s. This is mainly explained by the presence of small trains of detached bubbles in the `isoAdvector` run. The available reference data shows that the scatter between all the different solvers is important at large times. This is again explained by the prediction of different structures in the bubble queue, as observed in 2D (see also Figure 2).

3D computations have been run on the Occigen supercomputer at CINES, on Intel Xeon E5-2690V3. The largest computation with $320 \times 640 \times 320 = 65536000$ cells was run on 504 cores for 29.25 hours for `MULES` versus 26.78 hours for `isoAdvector` (so about 10% faster).

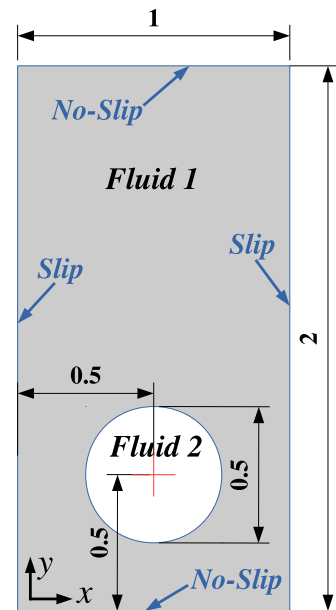


Figure 1: Configuration and boundary conditions for 2D bubble benchmark.

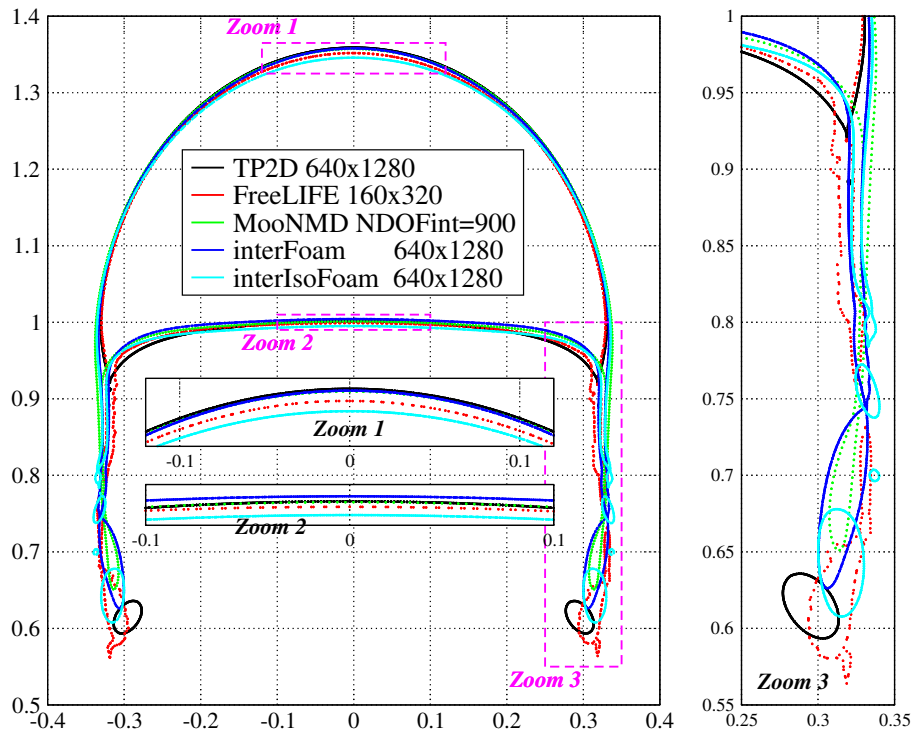


Figure 2: Comparison of bubble shapes at final time $t = 3$ s for 2D test case.

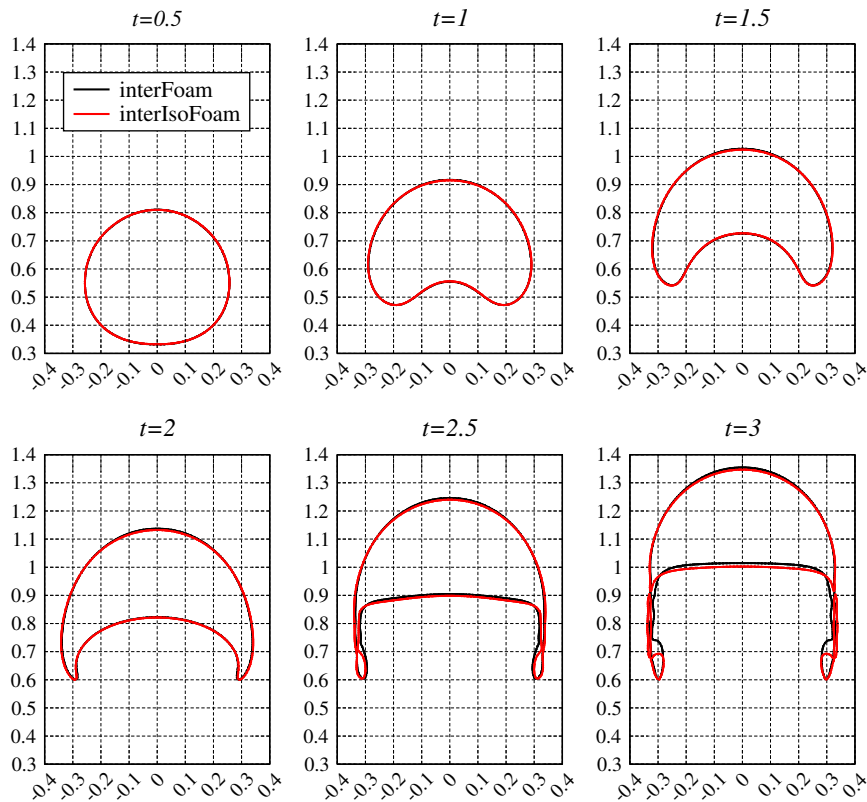


Figure 3: Time evolution of bubble shapes for 2D test case at resolution 160×320 . Comparison of MULES with isoAdvector.

5 Summary

This paper has shown a quantitative validation of the isoAdvector method against MULES and other codes. The test case used is the Hysing benchmark, both in 2D and 3D. isoAdvector has been verified to work for rising bubble simulation with similar accuracy as MULES and with a sharper interface and slightly smaller calculation times. This reference case will be contributed to the OpenFOAM[®] tutorial wiki.

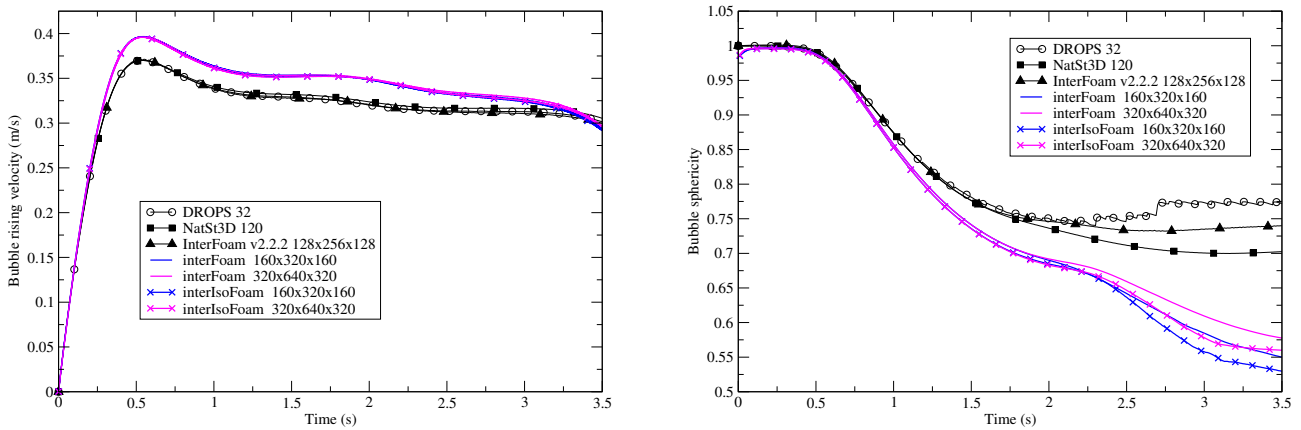


Figure 4: Time evolution of rise velocity (left) and sphericity (right) for 3D test case.

Acknowledgments

This work was granted access to the HPC resources of CINES under the allocation 2018-AP012B10362 made by GENCI.

References

- [1] C. Hirt and B. Nichols, “Volume of fluid (VOF) method for the dynamics of free boundaries,” *J. Comp. Phys.*, vol. 39, no. 1, pp. 201 – 225, 1981.
- [2] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *J. Comp. Phys.*, vol. 79, no. 1, pp. 12 – 49, 1988.
- [3] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, “Quantitative benchmark computations of two-dimensional bubble dynamics,” *I.J.N.M.F.*, vol. 60, no. 11, pp. 1259–1288, 2009. [Online]. Available: <http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble.html>
- [4] J. Adelsberger, P. Esser, M. Griebel, S. Groß, M. Klitz, and A. Rüttgers, “3D incompressible two-phase flow benchmark computations for rising droplets,” 2014, proceedings of the 11th World Congress on Computational Mechanics (WCCM XI), Barcelona, Spain, also available as INS Preprint No. 1401 and as IGPM Preprint No. 393. [Online]. Available: <http://wissrech.ins.uni-bonn.de/research/projects/risingbubblebenchmark/>
- [5] J. Roenby, H. Bredmose, and H. Jasak, “A computational method for sharp interface advection,” *Royal Society Open Science*, vol. 3, no. 11, 2016.
- [6] H. G. Weller, “A new approach to vof-based interface capturing methods for incompressible and compressible flow,” OpenCFD Ltd., Report TR/HGW/04, 2008.
- [7] S. S. Deshpande, L. Anumolu, and M. F. Trujillo, “Evaluating the performance of the two-phase flow solver interFoam,” *Comp. Sc. & Discovery*, vol. 5, no. 1, p. 014016, 2012. [Online]. Available: <http://stacks.iop.org/1749-4699/5/i=1/a=014016>
- [8] H. Marschall, K. Hinterberger, C. Schler, F. Habla, and O. Hinrichsen, “Numerical simulation of species transfer across fluid interfaces in free-surface flows using openfoam,” *Ch. Eng. Sc.*, vol. 78, pp. 111 – 127, 2012.
- [9] A. Q. Raeini, M. J. Blunt, and B. Bijeljic, “Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method,” *J. Comp. Phys.*, vol. 231, no. 17, pp. 5653 – 5668, 2012.
- [10] D. A. Hoang, V. van Steijn, L. M. Portela, M. T. Kreutzer, and C. R. Kleijn, “Benchmark numerical simulations of segmented two-phase flows in microchannels using the volume of fluid method,” *Computers & Fluids*, vol. 86, pp. 28 – 36, 2013.
- [11] C. Bilger, M. Aboukhedr, K. Vogiatzaki, and R. Cant, “Evaluation of two-phase flow solvers using level set and volume of fluid methods,” *J. Comp. Phys.*, vol. 345, pp. 665 – 686, 2017.
- [12] J. Roenby, B. E. Larsen, H. Bredmose, and H. Jasak, “A new Volume-of-Fluid method in OpenFOAM,” in *VII International Conference on Computational Methods in Marine Engineering, MARINE 2017*, 2017.
- [13] J. Roenby, H. Bredmose, and H. Jasak, “IsoAdvector: Geometric VOF on general meshes,” in *OpenFOAM - Selected papers of the 11th Workshop*. Springer, 2018.
- [14] H. Scheufler and J. Roenby, “Accurate and efficient surface reconstruction from volume fraction data on general meshes,” *arXiv*, Jan. 2018. [Online]. Available: <http://arxiv.org/abs/1801.05382>