# Bubble simulations with an interface tracking technique based on a partitioned fluid-structure interaction algorithm

J. Degroote[1]*, P. Bruggeman[2]†, R. Haelterman[3] and J. Vierendeels[1]

[1] Dept. of Flow, Heat and Combustion Mechanics, Ghent University
St.-Pietersnieuwstraat 41, B-9000 Ghent, Belgium

[2] Dept. of Applied Physics, Ghent University
Jozef Plateaustraat 22, B-9000 Ghent, Belgium

[3] Dept. of Mathematics (MWMW), Royal Military Academy
Renaissancelaan 30, B-1000 Brussels, Belgium

e-mails: {*Joris.Degroote, Peter.Bruggeman, Jan.Vierendeels*}*@UGent.be*, *Robby.Haelterman@rma.ac.be*

**Abstract**

Numerical techniques frequently used for the simulation of one bubble can be classified as interface tracking techniques and interface capturing techniques. Most of these techniques calculate both the flow around the bubble and the shape of the interface between the gas and the liquid with one code. In this paper, a rising axisymmetric bubble is simulated with an interface tracking technique that uses separate codes to determine the position of the gas-liquid interface and to calculate the flow around the bubble. The grid converged results correspond well with experimental data.

The gas-liquid interface is conceived as a zero-mass, zero-thickness structure whose position is determined by the liquid forces, a uniform gas pressure and surface tension. Iterations between the two codes are necessary to obtain the coupled solution of both problems and these iterations are stabilized with a fluid-structure interaction (FSI) algorithm. The flow around the bubble is calculated on a moving mesh in a reference frame that rises at the same speed as the bubble. The flow solver first updates the mesh throughout the liquid domain given a position of the gas-liquid interface and then calculates the flow around the bubble. It is considered as a black box with the position of the gas-liquid interface as input and the liquid forces on the interface as output. During the iterations, a reduced-order model of the flow solver is generated from the inputs and outputs of the solver. The solver that calculates the interface position uses this model to adapt the liquid forces on the gas-liquid interface during the calculation of the interface position.

## 1 Introduction

Bubbles have since long been of interest to both experimental and numerical researchers. Bubbly flow frequently occurs in reactors where it improves the mixing and reaction of the constituents [1]. By contrast, cavitation erodes metal surfaces due to the high pressure during bubble collapse [2]. An excellent overview of the various bubble shapes is provided by Clift, Grace and Weber, who establish a relation between the bubble shape and the Reynolds and Eötvös number [3]. Consequently, there is a lot of experimental data available to validate numerical techniques [3, 4, 5]. With the extended knowledge of bubble behaviour, the benefits of bubbly flow can be further enhanced and damage encountered in cavitation problems can be limited.

Numerical techniques frequently used for simulating a limited number of bubbles can be subdivided based on how they treat the interface between the liquid and the gas. Interface tracking techniques

*J. Degroote gratefully acknowledges a Ph.D. fellowship and a grant for a long stay abroad at the Massachusetts Institute of Technology of the Research Foundation - Flanders (FWO).

†P. Bruggeman gratefully acknowledges a Ph.D. fellowship of the Research Foundation - Flanders (FWO)

position grid nodes on the interface and the grid is thus deformed by the bubble motion. Conversely, interface capturing techniques employ a static grid and therefore do not place grid points on the interface, but they reconstruct the interface from a marker in the flow field. Although the former techniques provide a sharper representation of the interface than the latter for the same grid spacing, the complexity of the interface motion is more limited. Interface tracking has been developed by Ryskin et al. [6], Glimm et al. [7] and Tryggvason et al. [8]. Examples of interface capturing techniques are the Volume Of Fluid (VOF) and Level Set (LS) method, reviewed respectively by Scardovelli et al. [9] and Osher et al. [10].

More recently, advances in computer power have enabled the numerical solution of fluid-structure interaction and other coupled problems. Fluid-structure interaction is of paramount importance in aeroelastic analysis of wings [11] and buildings [12], but also in biomedical applications [13, 14, 15]. Fluid-structure interaction simulations can be performed with a single code for fluid and structure, the so-called monolithic approach, as opposed to the partitioned approach with separate codes for fluid and structure. The latter approach enables the reuse of well-validated codes but requires coupling iterations between the flow and the structure to satisfy the kinematic and dynamic conditions on the common boundary of the fluid and the structure. A coupling algorithm is necessary to stabilize these iterations and the last decade has witnessed a continuous progress in the development of these coupling algorithms [14, 16, 17, 18, 19, 20].

Most of the above-mentioned bubble simulation techniques compute the gas, the liquid and the interface in between with a single code. Conversely, the interface between the gas and the liquid can be conceived as a zero-thickness structure whose position is determined by the equilibrium between fluid forces and surface tension. From this point of view, a bubble consists of a structure with a fluid on either side. The bubble simulation is thus divided in a flow simulation and a structure simulation.

The partitioned approach to bubble simulation has already been followed by Ryskin et al. [6], who calculated the steady shape of a bubble, starting from a given initial position of the bubble, with the following iterative procedure. The Cartesian coordinates $(x, r)$ are first mapped to a coordinate system $(\xi, \eta)$ where the $\xi$ axis corresponds with the interface. The flow around the bubble is subsequently calculated by performing a limited number of iterations with a Navier-Stokes solver. The surface tension is then calculated from the curvature of the liquid-gas interface and compared with the liquid's force. The difference between these forces is considered as a normal force on the liquid-gas interface and is added to the mapping functions with an underrelaxation factor which is determined by trial and error. Finally the new position of the interface is calculated by applying these modified mapping functions. These steps are repeated until all equations and boundary conditions are satisfied. It is important to mention that the new position of the interface is not determined directly from the force unbalance but indirectly by modifying the coordinate transformation to avoid instability and that both the flow problem and the structural problem are not solved completely because it is observed that this also leads to instability.

The instability of the interaction between the flow problem and the structural problem in a partitioned simulation of a bubble that was observed by Ryskin et al. can be cured in a more robust way, that is without determining an underrelaxation factor by trial and error, by using a fluid-structure interaction coupling algorithm. These coupling algorithms have been developed to deal with the instability of the interaction between the flow and the structure in the partitioned approach to fluid-structure interaction and can thus be applied to partitioned simulations of bubbles and other problems with an interface as well.

This paper focuses on the application of the fluid-structure coupling algorithm first described by Vierendeels et al. [20] to partitioned bubble simulation. It is shown that one can obtain accurate, steady-state results of an axisymmetric bubble, rising in a stagnant mineral oil due to buoyancy with this technique. The shape of the bubble is calculated by coupling a flow solver for the fluid zone and a structure solver which calculates the interface position. The fluid-structure interaction coupling algorithm from Vierendeels et al. [20] for unsteady simulations is used in this paper to determine the interface position that balances the forces on fluid and structure in a steady simulation. This coupling algorithm was developed for a black-box flow solver and structure solver but in this paper it is modified to take advantage of the accessibility of the structure solver. The numerical results are thoroughly verified with experimental

data from Hnat et al. [5] and can be used as a benchmark for other bubble simulation codes.

The remainder of this paper is organized as follows. The flow solver and structure solver are defined in Sections 2 and 3, prior to an overview of the fluid-structure interaction coupling algorithm in Section 4. In Section 5, the results of the simulations are presented and compared with experiments from literature, followed by the conclusions in Section 6.

## 2 Flow solver

To fit in the fluid-structure interaction framework, the flow solver has to calculate the sum of the pressure and normal viscous force – further called the fluid load – of the liquid on the interface for a given shape of the bubble. Therefore, it has to calculate the pressure and velocity in the liquid domain around the bubble where pressure variation stems from gravity, inertia and viscosity. As the density and viscosity of a gas are typically orders of magnitude lower than those of a liquid, the pressure gradient inside the bubble is neglected with respect to the pressure variations on the liquid side of the interface. The gas flow inside the bubble is not calculated in this paper, but for cases where this assumption is no longer valid, the flow solver can easily calculate the gas flow as well.

The axisymmetric geometry of the mesh to calculate the flow in the liquid is similar to the experimental setup from Hnat et al. [5]. It consists of a cylindrical tank of 1.0 m high with a radius of 0.75 m whose axis of symmetry is aligned with the gravitational field. The bubble is positioned 0.5 m above the bottom of the reservoir.

The motion of a bubble that rises through a liquid is unsteady in reality as the bubble volume changes due to the hydrostatic pressure gradient. However, in the cases studied in this paper the shape and rise velocity of the bubble do not change significantly after an initial settling time and, therefore, the bubble shape is calculated with a steady simulation in a reference frame that moves at the same speed and in the same direction as the bubble. In this reference frame, the interface is modelled as a static free-slip wall and the liquid surrounding the bubble is given a downward velocity equal to the rise velocity of the bubble. In the absolute reference frame, the cylindrical wall of the tank is stationary and the top of the reservoir is a velocity inlet with zero velocity. The pressure at the bottom of the reservoir is set to zero with a pressure outlet. The correct absolute pressure level and the pressure due to gravity are subsequently added in the structure solver.

The finite volume flow solver (Fluent 6.3, Fluent Inc.) receives the position of the $N$ nodes on the interface and automatically updates the position of the other grid nodes in the liquid domain with a spring model. It then calculates the pressure and velocity throughout the liquid domain, returning the fluid load in all the nodes of the interface. The action of the flow solver can be summarized as

$$P = F(X) \tag{1}$$

with $X$ an array containing the axial and radial coordinate of the interface nodes and $P$ an array with the fluid load in those nodes.

$$X = \begin{bmatrix} x_1 & r_1 & x_2 & r_2 & \dots & x_N & r_N \end{bmatrix}^{\mathrm{T}} \tag{2}$$

$$P = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^{\mathrm{T}} \tag{3}$$

The pressure-based flow solver utilizes a coupled solution of the pressure and velocity field and a second-order upwind scheme for the discretization of the momentum equation. The method presented here is independent of the flow solver and any other solver should give similar results.

## 3 Structure solver

As mentioned above, the structure solver calculates the interface position for a given fluid load on the liquid side of the interface. Each node on the interface has 2 degrees of freedom, namely its axial and radial coordinate, so $2N$ degrees of freedom have to be determined.

In every interface node $i$ ($i = 1 \ldots N$), the difference between the fluid load $p_i$ and the gas pressure $p_G$ has to compensate for the surface tension.

$$(p_i + p_{abs} - \rho_L \cdot \mathcal{G} \cdot x_i) - p_G + \sigma \cdot \kappa_i = 0 \tag{4}$$

with $p_{abs}$ the absolute pressure level at the bottom of the tank, $\rho_L$ the fluid density, $\mathcal{G}$ the gravitational acceleration, $x_i$ the height of node $i$ above the bottom of the tank, $\sigma$ the surface tension coefficient and $\kappa_i$ the local surface curvature. The gas pressure inside the bubble $p_G$ is calculated with the ideal gas law from the mass of air and the temperature, both given constants. In Eq. (4), the fluid load $p_i$ consists of the sum of the pressure and normal viscous force. The tangential forces on the interface and the variation of the surface tension coefficient have been neglected. It will be shown in section 5 that simulations with these simplifications yield good results for the parameter values of interest in this paper. The left-hand side of Eq. (4) is further called $g_i$. The surface curvature $\kappa_i$ in node $i$ is calculated by constructing local $4^{\text{th}}$ order polynomial interpolants of the axial and radial coordinate from which the principal radii of curvature are calculated.

Tangential motion of the nodes along the interface has no effect on the bubble shape or on the flow field because the interface is modelled as a free-slip wall. Equidistance of the nodes is imposed to establish their position along the interface. This results in $N - 2$ equations

$$\left[(x_i - x_{i-1})^2 + (r_i - r_{i-1})^2\right] - \left[(x_i - x_{i+1})^2 + (r_i - r_{i+1})^2\right] = 0 \tag{5}$$

for $i$ from 2 to $N - 1$. The left-hand side of Eq. (5) is further called $h_i$. The radial degree of freedom of the nodes on the axis of symmetry is constrained and thus $h_1$ and $h_N$ are set to zero.

The $2N$ residual components $g_i$ and $h_i$ are stored in an array $G(X, P)$.

$$G(X, P) = \begin{bmatrix} g_1 & h_1 & g_2 & h_2 & \ldots & g_N & h_N \end{bmatrix}^{\mathrm{T}} \tag{6}$$

The *residual* further denotes the L1-norm of this array. The structure solver calculates the interface position $X$ that results in a residual below the convergence criterion of the structure solver for a given fluid load $P$. Its action can thus be summarized as

$$X = S(P) \tag{7}$$

The structure solver is an object oriented C++ code which has been developed by the authors. A modified version of the structure solver has been used for the simulation of bubble growth and detachment [20].

# 4 Coupling algorithm

Now the coupling iterations between the flow solver $F$ and the structure solver $S$ to find the steady-state bubble shape are analyzed. The flow solver is a black box code, which means that it cannot be modified and that the sensitivity of the output with respect to the input is unknown. Conversely, the structure solver is an accessible code, which is exploited by the coupling algorithm.

The flow solver first calculates the fluid load for two different bubble shapes chosen by the user, which are rough estimates of the final solution.

$$P_1 = F(X_1) \tag{8}$$
$$P_2 = F(X_2) \tag{9}$$

These 2 inputs $X_1, X_2$ and outputs $P_1, P_2$ of the flow solver are stored in a database which will contain all the inputs and the corresponding outputs of the flow solver during the following coupling iterations.

After these initial calculations, every following coupling iteration (indicated with subscript $k$, $k > 2$) consists of the following steps.

1. A reduced-order model of the flow solver is constructed based on the database of its inputs and outputs. At the beginning of coupling iteration $k$, the database contains $k-1$ inputs and the corresponding outputs of the flow solver.

$$X_1 \quad X_2 \quad \dots \quad X_{k-1} \tag{10}$$
$$P_1 \quad P_2 \quad \dots \quad P_{k-1} \tag{11}$$

This is a database of interface positions and the corresponding fluid loads on the interface. The state of the entire fluid domain in every coupling iteration is thus not stored. Consequently, the size of this database is small compared to the storage for the flow domain. The first $k-2$ inputs and outputs are converted into differences relative to the last input and output by subtracting $X_{k-1}$, respectively $P_{k-1}$.

$$\Delta X_1 \quad \Delta X_2 \quad \dots \quad \Delta X_{k-2} \tag{12}$$
$$\Delta P_1 \quad \Delta P_2 \quad \dots \quad \Delta P_{k-2} \tag{13}$$

with

$$\Delta X_i = X_i - X_{k-1} \tag{14}$$
$$\Delta P_i = P_i - P_{k-1}. \tag{15}$$

This is only possible if there are at least 2 inputs and outputs in the database and thus the preceding calculations in Eqs. (8-9) are indispensable.

The reduced-order model has to approximate the fluid load $P$ for an arbitrary position of the interface $X$. Therefore, $X$ is also converted into a difference $\Delta X$ relative to $X_{k-1}$.

$$\Delta X = X - X_{k-1} \tag{16}$$

$\Delta X$ is then decomposed as a linear combination of the known $\Delta X_i$ with $i$ ranging from 1 to $k-2$.

$$\Delta X \approx U \cdot \alpha \tag{17}$$

with

$$U = \begin{bmatrix} \Delta X_1 & \Delta X_2 & \dots & \Delta X_{k-2} \end{bmatrix} \tag{18}$$
$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{k-2} \end{bmatrix}^{\mathrm{T}} \tag{19}$$

This decomposition is approximate as the columns of $U$ are only part of the basis for the space with all possible $\Delta X$. The coefficients $\alpha$ are calculated with the least squares approach to minimize the L2-norm of the error between $\Delta X$ and $U \cdot \alpha$.

$$\alpha = \left(U^{\mathrm{T}} \cdot U\right)^{-1} \cdot U^{\mathrm{T}} \cdot \Delta X \tag{20}$$

The matrix inversion in the previous equation is cheap as the dimension of the matrix is $k-2$. The inversion fails if the columns of $U$ are linearly dependent.

The change in fluid load $\Delta P_i$ that corresponds with every component $\Delta X_i$ (i from 1 to $k-2$) in Eq. (17) is known. If the response of the flow solver were linear, then the change in fluid load corresponding with $U \cdot \alpha$ would be $V \cdot \alpha$, with

$$V = \begin{bmatrix} \Delta P_1 & \Delta P_2 & \dots & \Delta P_{k-2} \end{bmatrix}. \tag{21}$$

The reduced-order model approximates the change in fluid load $\Delta P$ that corresponds with $\Delta X$ by $V \cdot \alpha$. According to the reduced-order model, the fluid load $P$ corresponding with interface position $X$ is

$$\begin{aligned} P &= P_{k-1} + V \cdot \alpha && (22)\\ &= P_{k-1} + V \cdot \left(U^{\mathrm{T}} \cdot U\right)^{-1} \cdot U^{\mathrm{T}} \cdot \Delta X && (23)\\ &= P_{k-1} + A \cdot (X - X_{k-1}) && (24)\\ &= \hat{P}(X) && (25) \end{aligned}$$

with $A$ the approximate Jacobian of the flow solver, which is built up with inputs and outputs from the database.

2. The structure solver calculates the position of the interface with Newton-Raphson iterations, indicated with superscript $s$. The reduced-order model of the flow solver updates the fluid load on the interface after every Newton-Raphson iteration. It would be very time consuming to use the real flow solver for this purpose. Moreover, the Jacobian $A$ of the reduced-order model can be inserted into the Newton-Raphson iterations

$$\begin{aligned} X_k^{s+1} &= X_k^s - \left[ \left.\frac{\mathrm{d}G}{\mathrm{d}X}\right|^s + \left.\frac{\mathrm{d}G}{\mathrm{d}P}\right|^s \cdot A \right]^{-1} \cdot G(X_k^s, P_k^s) && (26)\\ P_k^{s+1} &= \hat{P}(X_k^{s+1}) && (27) \end{aligned}$$

where $\ldots|^s$ signifies that the Jacobian is evaluated in $X_k^s, P_k^s$. The dimension of the matrix is $2N$, which is small in comparison with the number of cells in the liquid domain, and thus a direct solver is used. After every Newton-Raphson iteration, the residual is calculated and compared with the convergence criterion. At convergence, the nonlinear equations are satisfied (up to the convergence criterion), even though the reduced-order model is only linear. The reduced-order model merely helps to obtain convergence of the coupled problem but it does not influence the final results.

Analysis of the partitioned simulation of coupled problems has demonstrated that the displacement modes of the interface with a low wave number are most unstable and once these displacement modes are treated implicitly, as is the case when they are included in the approximate Jacobian, the coupling iterations will converge quickly [21].

3. Once the structure solver has converged, the flow solver calculates the real fluid load $P_k$.

$$P_k = F(X_k) \quad (28)$$

4. The new input $X_k$ and output $P_k$ of the flow solver are added to the database.

5. The fluid load from the flow solver is inserted into the structure solver and the residual is calculated. The calculation finishes once the residual is below the convergence criterion.

# 5 Results and discussion

The partitioned approach to the rising bubble simulation has been applied to Bubble A in Table 1 from Hnat et al. [5]. This is a spherical cap bubble without skirt, which means that the top of the bubble is spherical but that the bottom is nearly flat. The surface tension is strong enough to prevent that a skirt – a trail of small bubbles behind the main bubble – develops. The parameters for this simulation have been listed in Table 1 and they correspond with a Reynolds number of $Re = 19$, a Weber number of $We = 15$ and an Eötvös number of $Eo = 39$ if the bubble diameter is used as reference length. The terminal rise velocity of this bubble is 0.215 m/s.

Table 1: Parameters for the simulation of Bubble A in Table 1 from Hnat et al. [5].

| | | |
|---|---|---|
| $\sigma$ | 0.0322 | N/m |
| $\mu_L$ | 0.118 | Pa s |
| $\rho_L$ | 875.5 | kg/m$^3$ |
| g | 9.81 | m/s$^2$ |

Figure 1 shows the bubble shape for 80, 120, 160 and 240 nodes on the interface. The origin of this figure is located at 0.5 m above the bottom of the tank. As the number of nodes on the interface increases, the difference between the results decreases, indicating grid convergence of the calculations. This figure also depicts the initial bubble shape ($X_1$), which is a hand-made drawing of the bubble shape. The second imposed bubble shape ($X_2$) is calculated from the first one as

$$x'_i = x_i \cdot \left(1 + \delta \cdot \sin\left(\pi \cdot \frac{i-1}{N-1}\right)\right) \tag{29}$$

$$r'_i = r_i \cdot \left(1 + \delta \cdot \sin\left(\pi \cdot \frac{i-1}{N-1}\right)\right) \tag{30}$$

for $i$ from 1 to $N$. $x_i, r_i$ are the coordinates in $X_1$ and $x'_i, r'_i$ those in $X_2$. The perturbation size $\delta$ has been chosen 0.001, but $\delta = 0.01$ or 0.0001 produces the same results for 120 nodes on the interface.

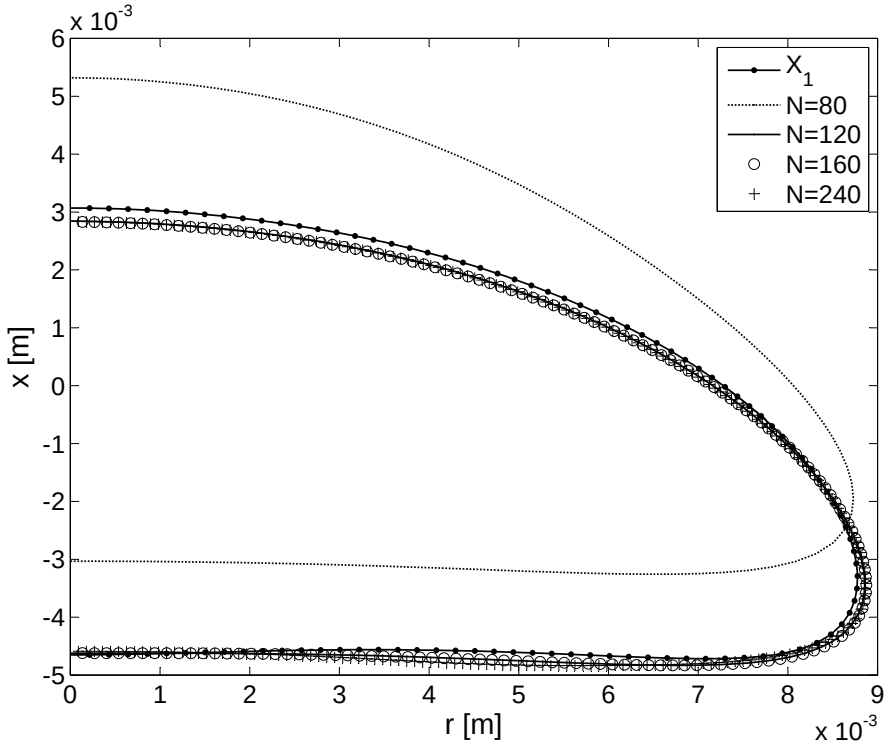


Figure 1: Bubble shape for 80, 120, 160 and 240 nodes on the interface, together with the initial bubble shape ($X_1$). The origin of this figure is located at 0.5 m above the bottom of the tank.

The convergence history of the coupling iterations is shown in Figure 2 for $N = 80, 160$ and 240. The convergence history for $N = 120$ is similar and has been omitted for clarity. The convergence criterion used is $0.1$ Pa because at that point, the maximal node displacement divided by the bubble radius decreases below $1 \cdot 10^{-3}$. The simulations with $N = 80$, 120, 160 and 240 required respectively

28, 23, 25 and 28 coupling iterations to reduce the residual below the convergence criterion. The number of coupling iterations for $N = 80$ is a little higher due to the large displacement of the bubble in that simulation.

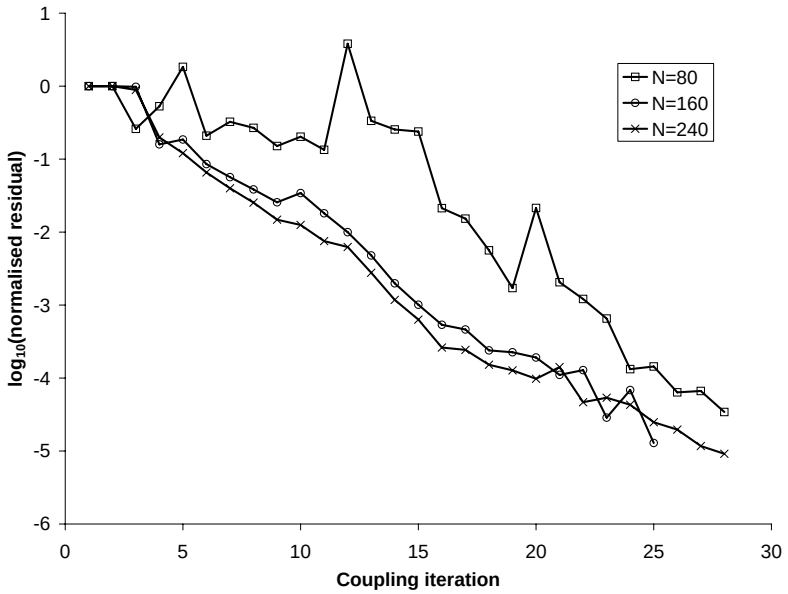


Figure 2: The convergence history of the normalized residual during the coupling iterations for 80, 160 and 240 nodes on the interface. The residual has been normalized with its initial value.

Table 2 summarizes the position of the bubble apex relative to the origin of Figure 1 and the ratio $B/A$ as a function of the number of nodes on the interface, where $B$ is the length measured from the bubble apex to the bottom of the wake and $A$ is the maximum girth dimension of the wake as indicated on Figure 3. A Richardson extrapolation based on this data has been performed and the error, which is the relative deviation from the extrapolation, is also tabulated. The decreasing error for increasing $N$ proves the grid convergence of the results.

Table 2: The position of the bubble apex relative to the origin of Figure 1 and the ratio $B/A$, where $A$ and $B$ are indicated on Figure 3, as a function of the number of nodes on the interface. Based on these data, a Richardson extrapolation has been performed and the error, being the relative deviation from the extrapolation, has been calculated.

| $N$ | Position $[10^{-3}$ m] | Error [%] | B/A [-] | Error [%] |
|---|---|---|---|---|
| 80 | 5.320 | 87.52 | 1.29 | 15.18 |
| 120 | 2.843 | 0.21 | 1.23 | 9.82 |
| 160 | 2.839 | 0.07 | 1.16 | 3.57 |
| 240 | 2.837 | 0.00 | 1.13 | 0.89 |
| Extrapolation | 2.837 | | 1.12 | |

Finally, Figure 3 compares the calculated bubble shape for $N = 240$ and the experimental bubble shape from Figure 1 in Hnat et al. [5]. The experimental part of this Figure is a shadowgraph that visualizes the bubble without skirt and the flow lines under the bubble. The agreement of experiment and calculation is excellent with regard to both bubble and wake shape.

The simulations presented above indicate that the steady-state shape of an axisymmetric bubble can

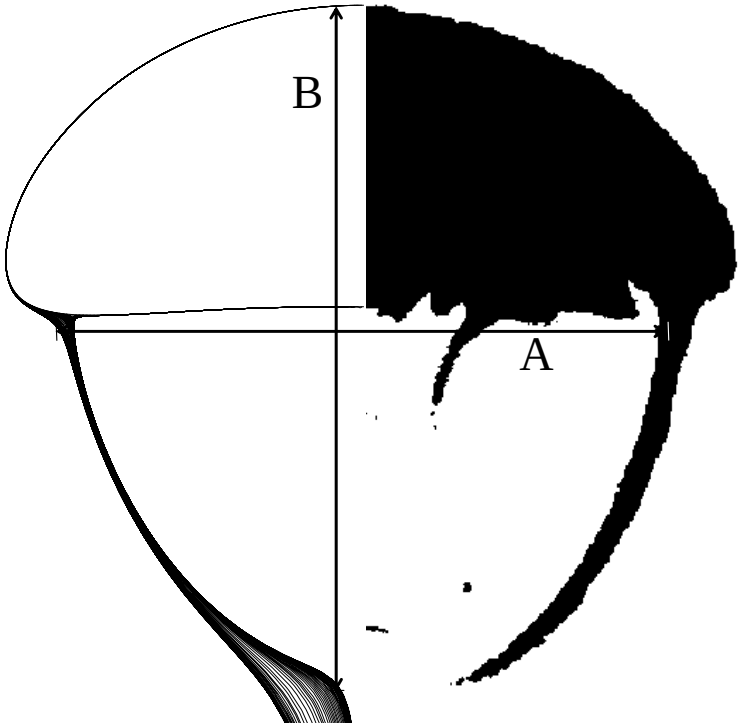


Figure 3: The bubble and wake shape from the simulation for $N = 240$ (left) and the experimental results (right) from Figure 1 in Hnat et al. [5]. The experimental part of this Figure is a shadowgraph and thus shows both the bubble without skirt and the flow lines underneath.

be calculated with a partitioned fluid-structure interaction approach as long as the rise velocity and an approximation for the initial bubble shape are known. The same accuracy should be obtained with the monolithic methods mentioned in Section 1 which means that the results in this work, obtained by coupling well-validated codes, can be used as a benchmark for new monolithic bubble simulation codes. It should be emphasized that no parameters have been tuned to obtain these results.

The coupling algorithm described in [20] has not been applied to steady-state calculations before. The inertia of the incompressible fluid does not hinder the convergence of the coupling iterations here. Not surprisingly, the number of coupling iterations is higher in these steady-state simulations as the displacement between the initial shape and the final shape is much larger than the displacement in a time step. This method is therefore limited to situations where a reasonable estimation of the bubble shape is known and where the interface is quite regular.

It has been demonstrated in [20] that the fluid-structure interaction coupling algorithm is capable of accelerating unsteady simulations. Future work will be to extend this steady partitioned bubble simulation technique to unsteady three-dimensional simulations and to ensure that it scales well such that more complex problems like mixing, reaction and cavitation can be tackled.

# 6 Conclusions

A fluid-structure interaction method has been adopted to calculate the steady-state shape of an axisymmetric bubble. This method successfully couples an existing and thoroughly validated black box flow solver with a newly developed, accessible structure solver. The former calculates the flow around the bubble and the latter the position of the interface which is conceived as a zero-thickness structure. Bubble A in Table 1 from Hnat et al. [5] has been simulated and the grid converged results presented in this paper bear strong resemblance to experiments.

## References


[1] A. Koynov, J.G. Khinast, and G. Tryggvason. Mass transfer and chemical reactions in bubble swarms with dynamic interface. *AIChE Journal*, 51(10):2786–2800, 2005.

[2] C.E. Brennen. *Cavitation and bubble dynamics*. Oxford University Press, New York, 1995.

[3] R. Clift, J.R. Grace, and M.E. Weber. *Bubbles, drops and particles*. Academic Press, Londen, 1978.

[4] M.S. Longuet-Higgins, B.R. Kerman, and K. Lunde. The release of air bubbles from an underwater nozzle. *Journal of Fluid Mechanics*, 230:365–390, 1991.

[5] J.G. Hnat and J.D. Buckmaster. Spherical cap bubbles and skirt formation. *Physics of Fluids*, 19(2):182–194, 1976.

[6] G. Ryskin and L.G. Leal. Numerical solution of free-boundary problems in fluid mechanics. part 1. the finite-difference technique. *Journal of Fluid Mechanics*, 148:1–17, 1984.

[7] J. Glimm, J.W. Grove, X.L. Li, W. Oh, and D.H. Sharp. A critical analysis of rayleigh–taylor growth rates. *Journal of Computational Physics*, 169:652–677, 2001.

[8] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, and S.and Jan Y.-J. Nas. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.

[9] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.

[10] S. Osher and R.P. Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001.

[11] C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001.

[12] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193:2087–2104, 2004.

[13] J.F. Gerbeau, M. Vidrascu, and P. Frey. Fluid-structure interaction in blood flows on geometries based on medical imaging. *Computers and Structures*, 83(2-3):155–165, 2005.

[14] M.A. Fernandez and M. Moubachir. A newton method using exact jacobians for solving fluid-structure coupling. *Computers and Structures*, 83:127–142, 2005.

[15] J. Vierendeels, K. Riemslagh, E. Dick, and P. Verdonck. Computer simulation of intraventricular flow and pressure gradients during diastole. *Journal of Biomechanical Engineering*, 122:667–674, 2000.

[16] J.-F. Gerbeau and M. Vidrascu. A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(4):631–648, 2003.

[17] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1–23, 2004.

[18] H.G. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*, 195:2028–2049, 2006.

[19] D.P. Mok, W.A. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. In K.-J. Bathe, editor, *First MIT Conference on Computational Fluid and Solid Mechanics*, pages 1325–1328, Cambridge, MA, USA, 12-14 June 2001. Elsevier.

[20] J. Vierendeels, L. Lanoye, J. Degroote, and P. Verdonck. Implicit coupling of partitioned fluid–structure interaction problems with reduced order models. *Computers and Structures*, 85:970–976, 2007.

[21] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in FSI applications. *Computers and Structures*, 86(23–24):2224–2234, December 2008.