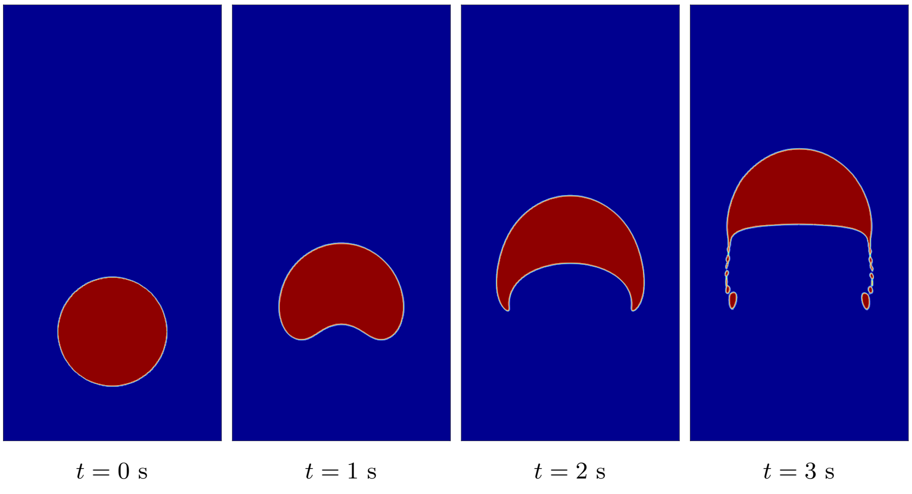
Basic Usage of OpenFOAM

A course at Chalmers University of Technology

Taught by Håkan Nilsson

Examination, part 2:
Rising bubble



Bubble shape evolution through time (320 × 640 mesh and `maxCo`= 0.01)

April, 2025

1 Introduction

In this project, you will study the rising bubble phenomenon using OpenFOAM. In 2009, Hysing et al. [1, 2, 3] published a pure numerical benchmark with two test cases for a two-dimensional rising bubble. Our goal is to compare OpenFOAM Volume of Fluid (VOF) results with other numerical results in the benchmark. At the same time you will practice to find and use solvers, utilities, and functionObjects.

2 Description of the geometry and flow field

There are two different test cases in the numerical benchmark. The initial configuration, shown in Fig. 1, is identical for both test cases and consists of a circular bubble of radius $r_0 = 0.25$ centered at $(x, y) = (0.5, 0.5)$ in a 1 by 2 rectangular domain. The density of the bubble is smaller than that of the surrounding fluid ($\rho_2 < \rho_1$) and the gravity force is downwards. Hence, the bubble will rise in the domain. The no-slip boundary condition ($\mathbf{u} = \mathbf{0}$) is used at the top and bottom boundaries, whereas the free slip condition ($\mathbf{u} \cdot \hat{\mathbf{n}} = 0$) is imposed on the vertical walls.

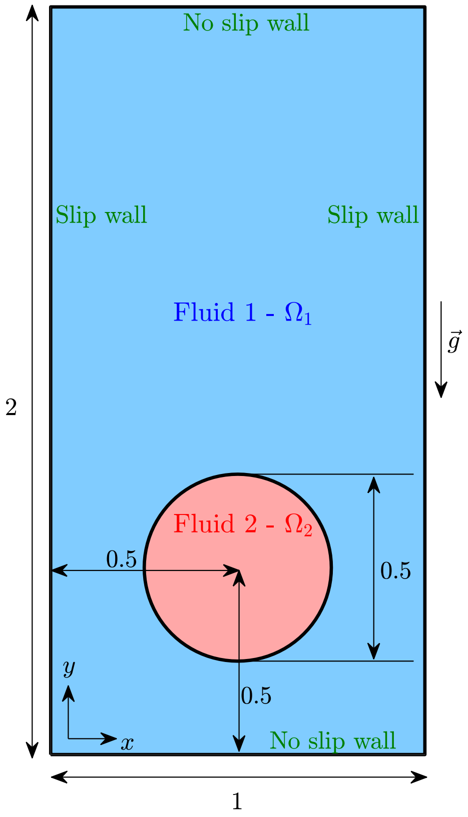


Figure 1: Initial configuration and boundary conditions for the test case (dimensions are in meters)

In this task, we want to simulate Case 2, which is the more challenging case. The fluid and physical parameters of this case are presented in Table 1. This case models a rising bubble with Reynolds number Re = 35, Eötvös number Eo = 125, and with large density and viscosity ratios (1000 and 100). The evolution of the bubbles should be tracked for 3 seconds during which the defined benchmark quantities should be measured.

Table 1: Physical parameters and dimensionless numbers defining the test cases

Test case	ρ_1	ρ_2	μ_1	μ_2	g	σ	Re	Eo	ρ_1/ρ_2	μ_1/μ_2
2	1000	1	10	0.1	0.98	1.96	35	125	1000	100

3 Numerical benchmark data

You should compare your results with the data provided by Hysing et al. [3]. All the numerical data are freely available on this website. Different sets of data are provided in the benchmark. You can use the results of the FreeLIFE code with the highest mesh density for comparison.

The provided numerical benchmark data consists of two different folders, namely, `data_bubble_shapes` and `data_bench_quantities`. The first folder contains the coordinates of the bubble interface at $t = 3$ s, while any other required data are stored in `data_bench_quantities`. All the files in this folder have the same format. Each file contains five columns, corresponding to time, bubble area, circularity, center of mass, and rise velocity, respectively.

In each folder, there are multiple data ASCII files. The naming of the provided data files in `data_bench_quantities` folder indicates the case number, employed solver, and mesh density. For instance, a file may look like `c2g3l4`. The first letter of the file name, `c`, represents the case number. Here, you only work with Case 2. The number after the second letter, `g`, denotes the employed CFD code. Three different codes were utilized in the benchmark, namely, TP2D, FreeLIFE, and MooNMD, corresponding to `g1`, `g2`, `g3`, respectively. The last letter, `l`, indicates the level of mesh density. A higher number after `l` indicates a finer mesh. The data files for the bubble shapes (`data_bubble_shapes`) have the same structure but end with an `s` character, for example, `c2g3l4s`.

4 Computational details

Use the `blockMesh` utility to create your mesh and then create the initial bubble using the `setFields` utility. The initial shape of the bubble is a circle. Therefore, you may use the `cylinderToCell` to mark the circular region. The `interFoam` solver should be used for the computations, which employs the PIMPLE algorithm for pressure correction. Perform the simulation for three seconds and use `adjustTimeStep` and maximum Courant number (`maxCo`) for time-marching.

The second-order upwind differencing scheme (`linearUpwind`) should be used for discretization of the convective term in the momentum transport equation while convection of volume fraction term should be discretized using the `vanLeer` scheme. The gradient and diffusion terms must be discretized using the central differencing scheme. In addition, you should employ a pure Crank-Nicolson scheme for temporal discretization.

The simulations should be carried out for:

- 80 × 160 mesh, `maxCo` = 0.1
- 80 × 160 mesh, `maxCo` = 0.01
- 160 × 320 mesh, `maxCo` = 0.01

5 Post-processing

In order to post-process your numerical simulations and compare your results with the provided benchmark numerical data, you need to develop a Python script for automatic post-processing. As a PhD student/researcher, you are expected to create journal publication-quality figures. Please remember that in each figure, you need to compare the results of all three cases mentioned in Section 4 with the benchmark data.

5.1 Bubble shapes

Compare the bubble shape at $t = 3$ s with the benchmark data (like Fig. 2). It can be assumed that the interface of the bubble is at `alpha.water` = 0.5. Hint: You can create the `isoSurface` of `alpha.water` = 0.5 and save it in `raw` format.

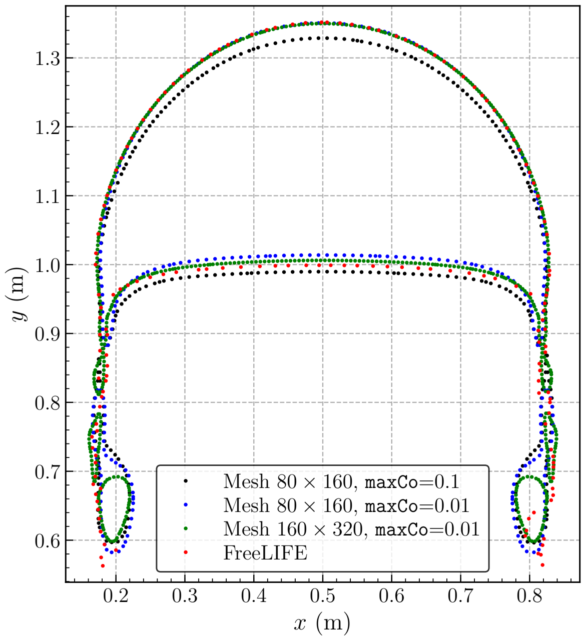


Figure 2: Bubble shape at $t = 3$ s

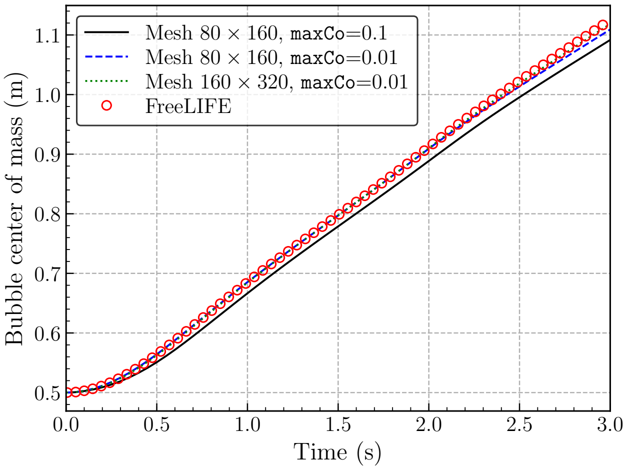


Figure 3: Bubble center of mass

5.2 Center of Mass

The bubble center of mass is defined by

$$\mathbf{x}_c = (x_c, y_c) = \frac{\int_{\Omega_2} \mathbf{x}\,\mathrm{d}x}{\int_{\Omega_2} \mathbf{1}\,\mathrm{d}x}, \tag{1}$$

where Ω_2 indicates the bubble region (Fig. 1). Calculate the bubble center of mass through time and compare it with numerical benchmark data (like Fig. 3). One way to do this, is to write the cell centered data on the mid-plane and then load the data in your post-processing script and compute the center of mass. In order to write only the cell-centered data you can set `triangulate` to `false` in your plane definition dictionary.

5.3 Rise velocity

The mean velocity of the rising bubble can be computed as

$$\mathbf{U} = \frac{\int_{\Omega_2} \mathbf{u}\,\mathrm{d}x}{\int_{\Omega_2} \mathbf{1}\,\mathrm{d}x}. \tag{2}$$

Calculate this velocity through time and compare it with numerical benchmark data (like Fig. 4).

5.4 Circularity

The bubble circularity can be defined as

$$\not{c} = \frac{P_a}{P_b} = \frac{\text{Perimeter of area-equivalent circle}}{\text{Perimeter of bubble}} \tag{3}$$

Calculate the bubble circularity through time and compare it with numerical benchmark data (like Fig. 5). To calculate this parameter, you need to

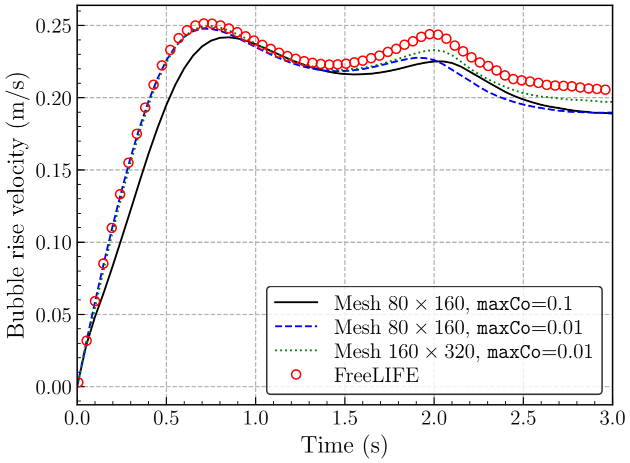


Figure 4: Bubble rise velocity

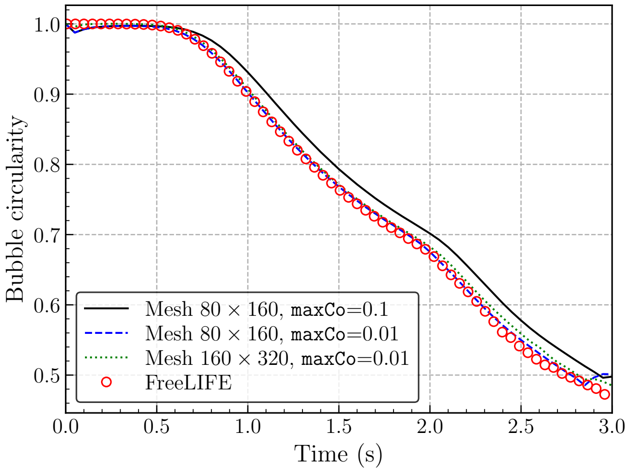


Figure 5: Bubble circularity

have the perimeter of the bubble through time (in this case, the area of the bubble surface). You may use the `surfaceFieldValue` `functionObject` with `areaIntegrate` `operation` to integrate over a surface. The `isoSurfaceTopo` of `alpha.water=0.5` can be set as the surface where integration should be computed. The `isoSurfaceTopo` is an improved version of `isoSurface` which clearly works better for this case. Since you need to define a field for area integration you can just set your field to `alpha.water`, because it is always constant at 0.5 on this surface.

6 What to submit

To pass this task, you should submit the following files and folders to the submission computer.

- All the described cases in the assignment description containing the requested results and figures (preferably in separate folders that are easy to find)
- One Allrun script which automatically runs all the cases, post-processes the results, and creates the required figures. See OpenFOAM tutorials for examples of Allrun scripts. Do not forget to source the OpenFOAM version that you want to use in your scripts.
- Post-processing scripts that read OpenFOAM results and the reference data and create all the requested figures. Your post-processing program should be called by your Allrun script.
- One Allclean script which automatically cleans all the cases, removes time folders, post-processing results, mesh, etc. See OpenFOAM tutorials for examples of Allclean scripts.

Please make sure that your scripts execute successfully on the submission computer, meaning they perfectly run all the cases in the assignment, post-process the results, and create the required figures. In other words, do not just run the cases on your own computer and then copy the results into the submission computer.

We will first examine your cases, results, and figures. Subsequently, your Allclean script will be executed, and it should clean all the unnecessary data and leave a clean case. Then, your Allrun script will be run. Your assignment will be assessed *only* by the successful execution of your Allclean and Allrun scripts on the submission computer and checking your results, i.e., requested figures. Do not forget to include any reference data that your post-processing script needs to load.

If your script fails to run or creates extremely poor results, we won't be going into your script and debugging it, and you will fail the task.

References

[1] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, "Quantitative benchmark computations of two-dimensional bubble dynamics," *International Journal for Numerical Methods in Fluids*, vol. 60, no. 11, pp. 1259–1288, 2009.

[2] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, "Proposal for quantitative benchmark computations of bubble dynamics," *International Journal for Numerical Methods in Fluids*, 2007.

[3] "Bubble benchmark." http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble.html. Accessed: 2020-07-10.