

The 2D Hysing benchmark

Lionel GAMET

July 18, 2018

1 Introduction

This case is a reference test case for VoF simulations. It is an elementary quantitative benchmark configuration that was originally established in 2D by Hysing et al. [1]. The case consists in a single rising bubble in a quiescent liquid. The test case number 2 as described by Hysing et al. [1] is modeled here.

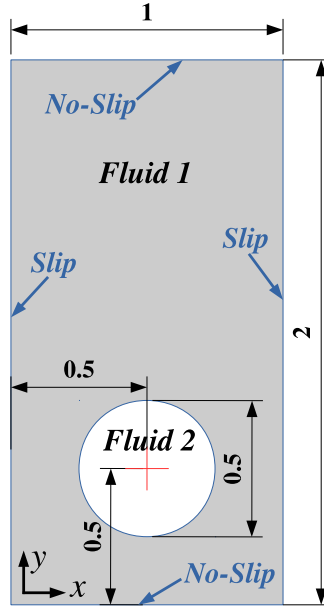


Figure 1: Configuration and boundary conditions for 2D bubble benchmark.

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 $g = 0.98 \text{ m/s}^2$. The chosen values are not representative of existing fluids. This test case has originally been designed for

comparison purposes between CFD solvers. The domain is of size 1×2 in the x and y directions.

2 Setting up the test case in OpenFOAM

This test case uses incompressible VoF solvers in OpenFOAM, namely `interFoam` or `interIsoFoam` solvers.

A uniform Cartesian grid built with `blockMesh` is used for 2D simulations. The default grid size is 20×40 . When running the grid sensitivity script (see section 3), finer grid levels will be used. The bubble is initialized as a cylinder in 2D using the `setAlphaField` utility.

NB: `setAlphaField.C` *needs to be patched so that it generates a bubble of gas instead of a drop of liquid. This can be done by adding the line `alpha1 = 1.0 - alpha1;` in the source code:*

`$FOAM_APP/utilities/preProcessing/setAlphaField/setAlphaField.C:`

```
[...]
    // Calculating alpha1 volScalarField from f = f0 isosurface
    isoCutCell icc(mesh, f);
    icc.volumeOfFluid(alpha1, f0);
+   alpha1 = 1.0 - alpha1;

    // Writing volScalarField alpha1
    ISstream::defaultPrecision(18);
[...]
```

In the `fvSchemes` file, a Crank-Nicolson second order time scheme with blending coefficient 0.9 is chosen. `Gauss limitedLinearV 1` is used to treat the convective term, and `Gauss vanLeer` is used for the α convective term for MULES simulations. The `Gauss linear` scheme is used by default for all gradient terms.

In the `fvSolution` file, the GAMG implicit solver is used for pressure terms, while the smooth solver is used for the velocity. The PIMPLE algorithm uses 3 `nOuterCorrectors` and 3 PISO correctors (`nCorrectors=3`). It was found that `momentumPredictor` needed to be set to true to get a correct solution in terms of rising velocity and bubble circularity, in particular with `isoAdvector`. Both `isoAdvector` and MULES numerical parameters are present and appear separately in `fvSolution`, so that both `interFoam` and `interIsoFoam` can be run from the same input files.

A constant time step is used, starting at $\Delta t = 0.002$ s for the 20×40 coarsest level and reduced for finer grids to keep the maximum CFL number below 0.05. Computations are run up to time $t = 3$ s in 2D.

Post-processing quantities of interest are described in details in [1]. These are the vertical position of the bubble centroid, the bubble rise velocity, the bubble circularity, bubble volume and area. All these quantities are computed through

a coded functionObject inlined in the `controlDict` file. The results appear in the log file of the solver. We use a `sampledIsoSurfaceCell` object, defined as the isosurface $\alpha = 0.5$, to compute the bubble area. We also use volume integrals of the gas fraction overall the computational domain to compute the bubble volume, centroid and velocity. A bubble equivalent diameter named D_A is computed from the bubble volume. A bubble equivalent diameter named D_B is computed from the bubble area. The circularity is defined in 2D as the ratio D_A/D_B . This number takes the value 1 at $t = 0$ as the bubble is initialized as a perfect cylinder, and then decreases with time as the bubble rises and deforms.

Finally, the bubble shape is output at `writeTime` frequency through a `surfaces` sampling functionObject, based upon an isosurface $\alpha = 0.5$. Both interpolated and constant iso bubble shapes are output.

3 Running the case

The `Allrun` script will run the default grid size at 20×40 . The grid is first constructed by running `blockMesh`. Then the `setAlphaField` utility is run to initialize the bubble as a cylinder. Finally, the application (default to `interFoam`) is run. Post-processing quantities of interest (bubble volume, area, centroid, velocity and circularity) are extracted from the solver log file through a `grep` command.

The `Allrun_sensitivity` is a script at a level above `Allrun`. It is used to run a grid sensitivity for both solvers `interFoam` and `interIsoFoam` with a number of cells 20, 40, 80, 160, 320 and 640. If line 39 is commented, the script `Allrun_sensitivity` will proceed in running all cases simultaneously. Otherwise (and this is the default), the sensitivity script will only generate running directories that you can execute manually later, or on another computer.

For comparisons, reference 2D data can be found at the bibliography link.

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,” *I.J.N.M.F.*, vol. 60, no. 11, pp. 1259–1288, 2009. [Online]. Available: <http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble.html>