Simulation of a bubble column

Final project on OpenFOAM

Team 2: Gerasimos Chourdakis, Vasileios Magioglou

Computational Thermo-Fluid Dynamics with Open Source Tools, 20/07/2016

Outline

Introduction

Case setup
- Mesh setup
- Initial & boundary conditions
- Solver configuration

Results & discussion
- Main case
- twoPhaseEulerFoam
- Larger bubbles
- Intraction with obstacles

Conclusions (& future work...)

Outline

Problem

Simulation of air bubbles being injected into water.

Questions:

- bubbles size and shape evolution?
- critical size of bubbles?
- influence on the water flow?

Applications:

- gas absorption
- gas-liquid (catalytic) reactions (liquid-liquid also)
- biomass production / wastewater treatment

bubbleFoam

Two-phase solver, based on the Euler-Euler two-fluid methodology.

Assumptions:

- incompressible phases
- constant dispersed phase diameter (and only one)
- isothermal flow
- the phases interact only through momentum exchange

bubbleFoam

Two-phase solver, based on the Euler-Euler two-fluid methodology.

Assumptions:

- incompressible phases
- constant dispersed phase diameter (and only one)
- isothermal flow
- the phases interact only through momentum exchange

Limitations:

- only one dispersed and one continuous phase
- no model for aggregation, breakage and coalescence
- simplistic drag coefficient model
- no heat/mass transfer, no chemical reactions

Similar solvers: twoPhaseEulerFoam, multiphaseEulerFoam, interFoam, multiphaseInterFoam

bubbleFoam parameters

Two phases are defined: **a** (dispersed) and **b** (continuous).
Velocities: `Ua`, `Ub` and `Ur` (relative, computed).

transportProperties:

- `rhoa`, `rhob`: densities [1, 1000]
- `nua`, `nub`: viscosities [1.6e-05, 1e-06]
- `da`, `db`: phase diameters [0.003, 0.0001]
- `Cvm`: virtual mass coefficient [0.5]
- `Cl`: lift force coefficient [0]
- `Ct`: turbulence response time coefficient [1]

bubbleFoam tutorial

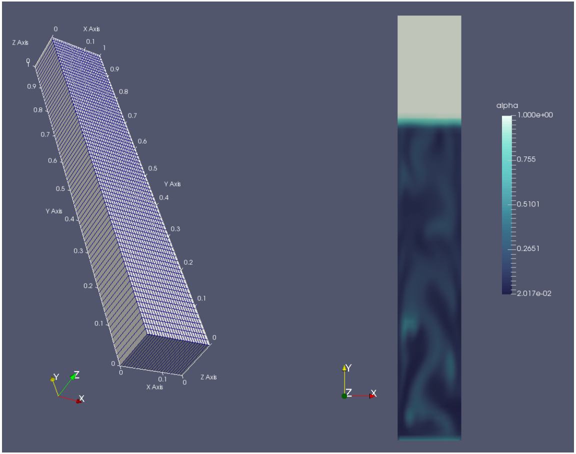


Figure: `OpenFOAM-2.0.1/tutorials/multiphase/bubbleFoam/bubbleColumn`

Outline

Mesh setup

3D cylinder produced with blockMesh:

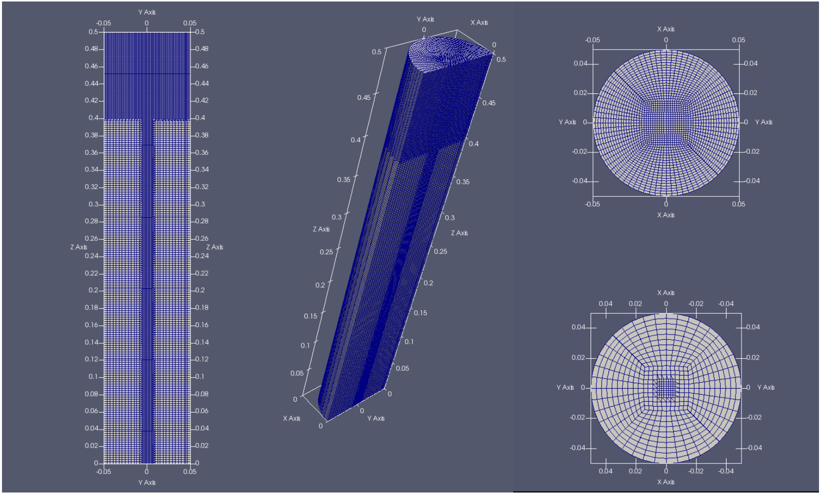
1. 8 points/level: inner and outer square
2. edges of type `arc`
3. 5 blocks: 1 central and 4 peripheral
4. partially refined using setSet and refineMesh

A template can be produced by a 3rd-party generator in Python.

As an extra case, an obstacle was introduced with setSet (`sphereToCell`) + subsetMesh.

order of execution when setting custom patches:
`blockMesh > refineMesh > createPatch > subsetMesh > setFields`.

Mesh example



Initial & boundary conditions

Square inlet created with createPatch (`boxToFace`).

Gas velocity set with `timeVaryingUniformFixedValue` (OpenFOAM 2.0.1). Five time points were set, to model quick pulses. [0.0, 0.020, 0.021, 0.080, 0.081]

`alpha` set using groovyBC with condition `mag(Ua) > 0.0`.

BC for `alpha` at atmosphere, walls: `zeroGradient`.

The column is set to partially filled using setFields.

Initial & boundary conditions - square inlet

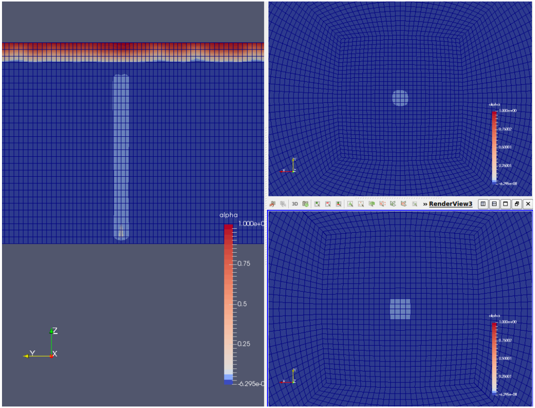


Figure: Early results. Although the bubble starts with a square cross-section, it quickly turns to round.

Solver configuration

An adjusted timestep was used, set taking the Courant number in mind (`adjustTimeStep`, `maxCo`, `maxAlphaCo`, `maxDeltaT`).

The rest of the solver parameters were set as in the bubbleColumn tutorial.

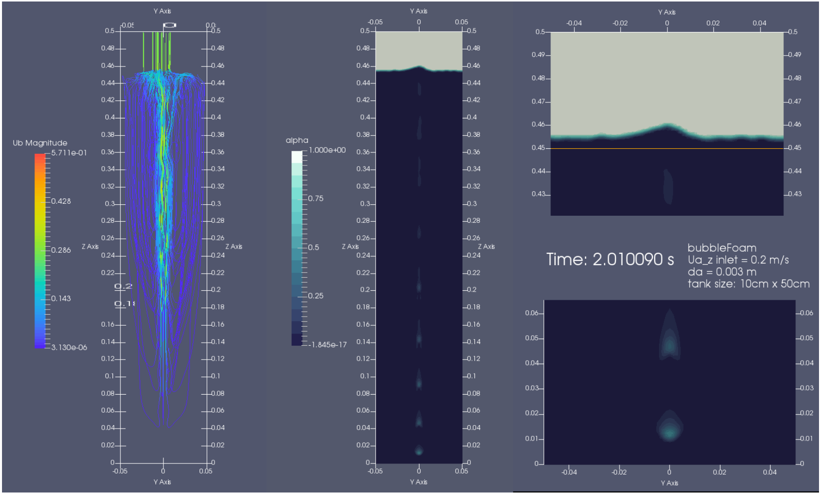
Various cases simulated, in respect to column length and number of cells.
Important lesson: eat in small bites!

Simulated time 2.5 s for the full-size problem required approximately:

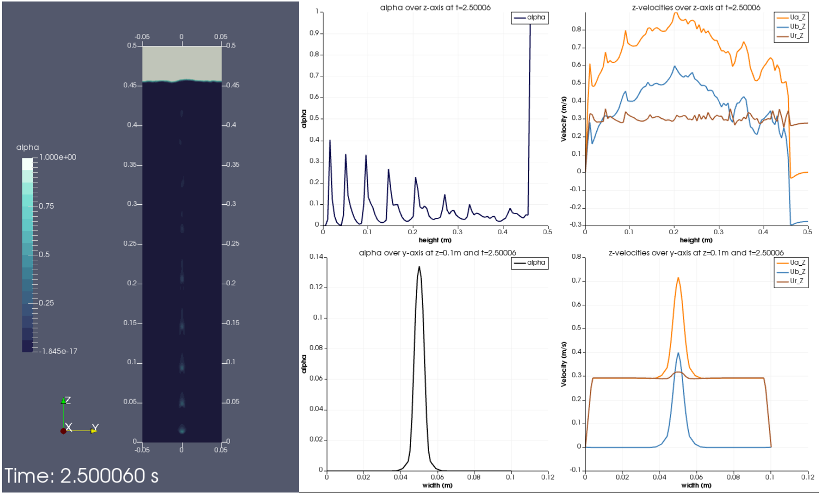
- 13 h with 2 processes on a 2-cores laptop
- 5 h with 4 processes on `klausmoesl` (slightly smaller problem)

Outline

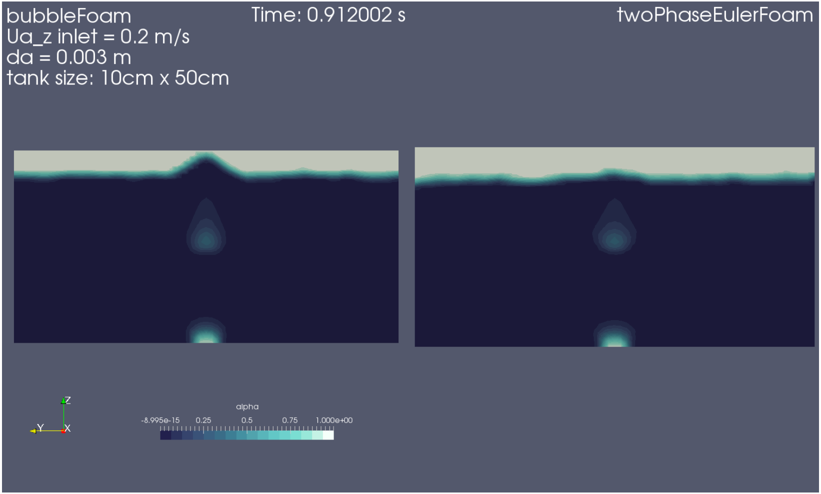
Results: main case



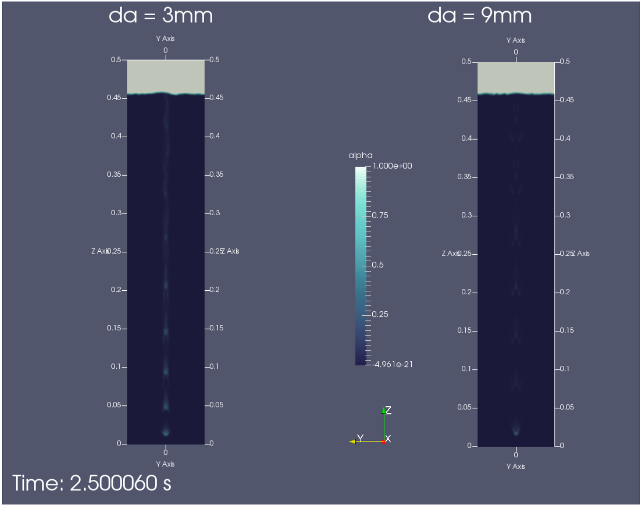
Results: main case - line plots



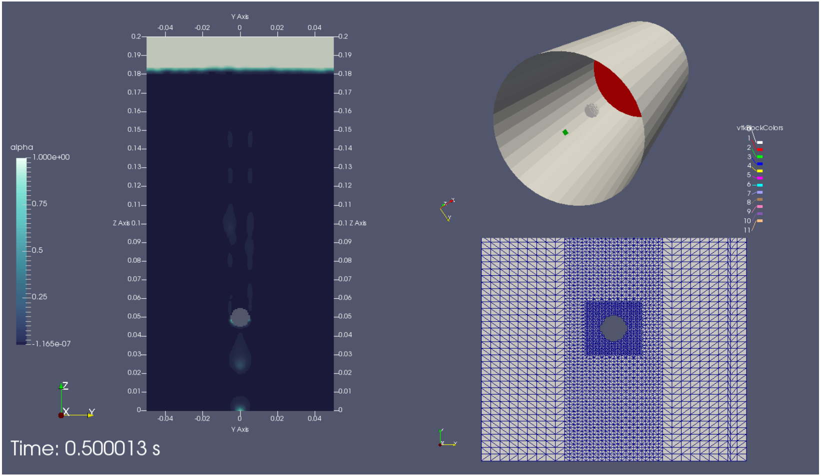
Results: main case - twoPhaseEulerFoam



Results: larger bubbles



Results: interaction with obstacles



Outline

Main observations

1. The gas fraction decreases as the height increases.
2. The bubbles (dispersed phase areas) are transformed as they travel. The square cross-section is transformed to round.
3. The gas velocity increases up to a maximum and decreases again near the surface.
4. The surface level increases when bubbles are injected and reacts to the arrival of bubbles.
5. The gas flow causes also water flow. Near the surface, the flow is reflected.

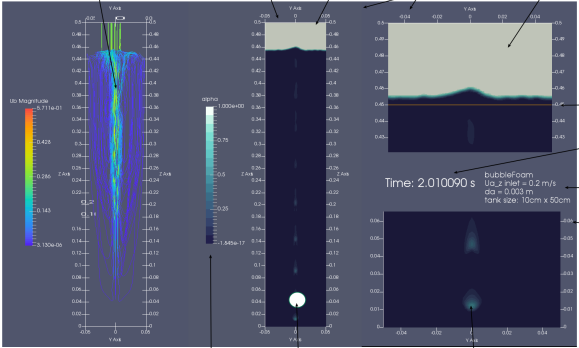
Main difficulties

1. Creating a cylindrical mesh. Used a script to generate a blockMeshDict.
2. When `alpha` is low, numerical diffusion gets important. Increased the inlet velocity and refined the mesh.
3. Setting reasonable parameters. Tried several parameters in smaller grids.
4. Converging with a fine mesh. Used an adjusted timestep, used a coarser mesh that didn't change the solution.
5. Execution time and results storage demands. Used parallel execution, stopped and restarted the simulations to transfer the results in external drive.

Ideas for future work

1. Add more bubble injection points.
2. Use a more advanced solver and study the models for coalescence.
3. Simulate heat and mass transfer, as well as chemical reactions.
4. Simulate the flow of bubbles in complex geometries. Try to maximize the air-water contact time.
5. Introduce a range of bubble diameters.
6. Simulate a three-phase system (water, oil, air).
7. Study different ways of producing bubbles.

Questions?


save animation
stream tracer
blockMesh (arc)
refineMesh
slice / clip
split view
setFields
"Refresh" bu
Ub Magnitude
alpha
line
Annotate tir
Time: 2.010090 s
bubbleFoam
Ua_z inlet = 0.2 m/s
da = 0.003 m
tank size: 10cm x 50cm
Text
Axis Grid
plot over line
bubbleFoam
adjustTimestep
twoPhaseEulerFoam
decomposePar
MPI (local, cluster)
Edit Color Map
setSet (sphereToCell)
subsetMesh
Case Type:
Decomposed Case
timeVaryingUniformFixedValue
groovyBC
createPatch
setSet boxToFace