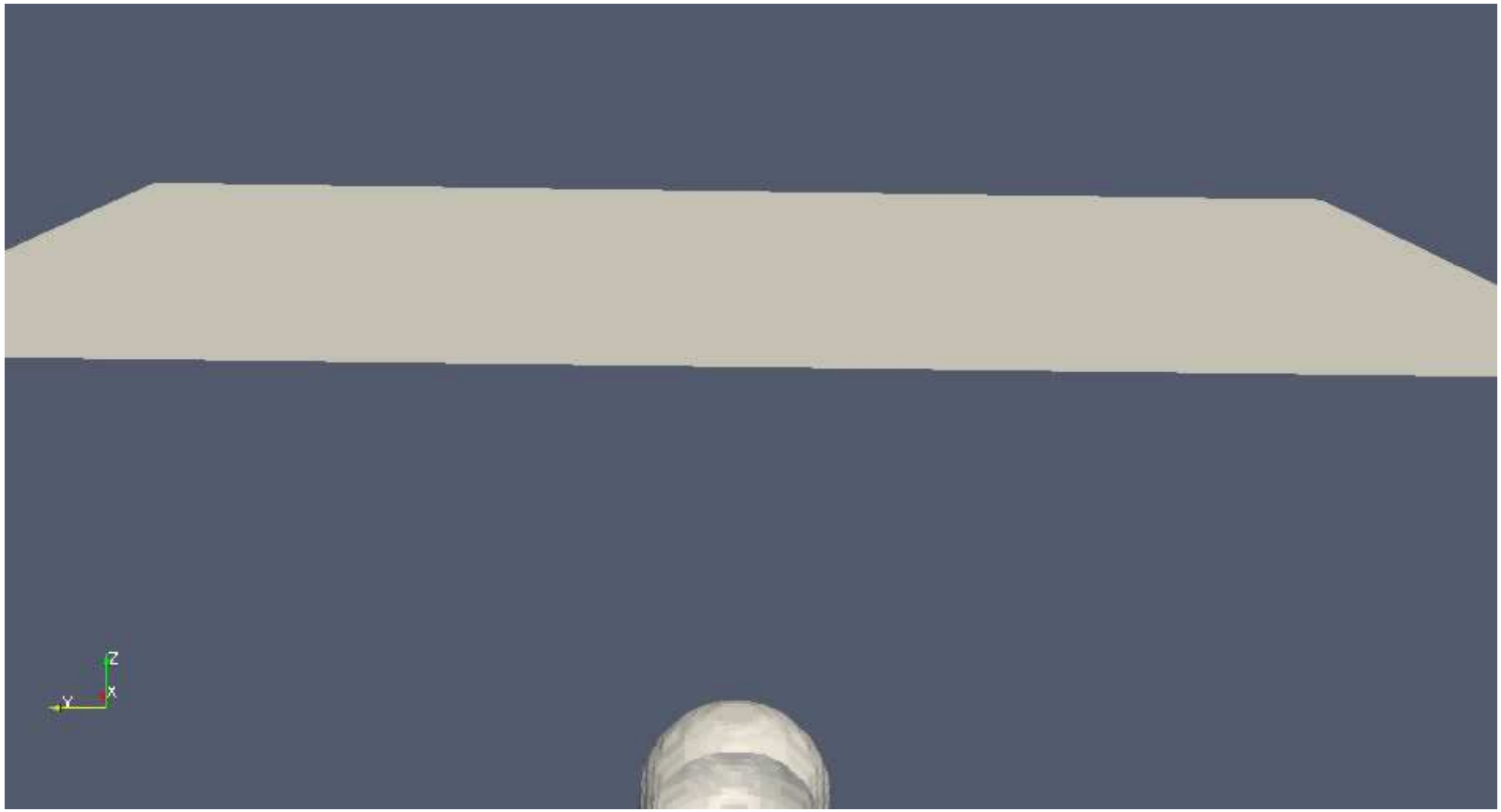
Palabos Training

Free-Surface and Multi-Phase flow in Palabos

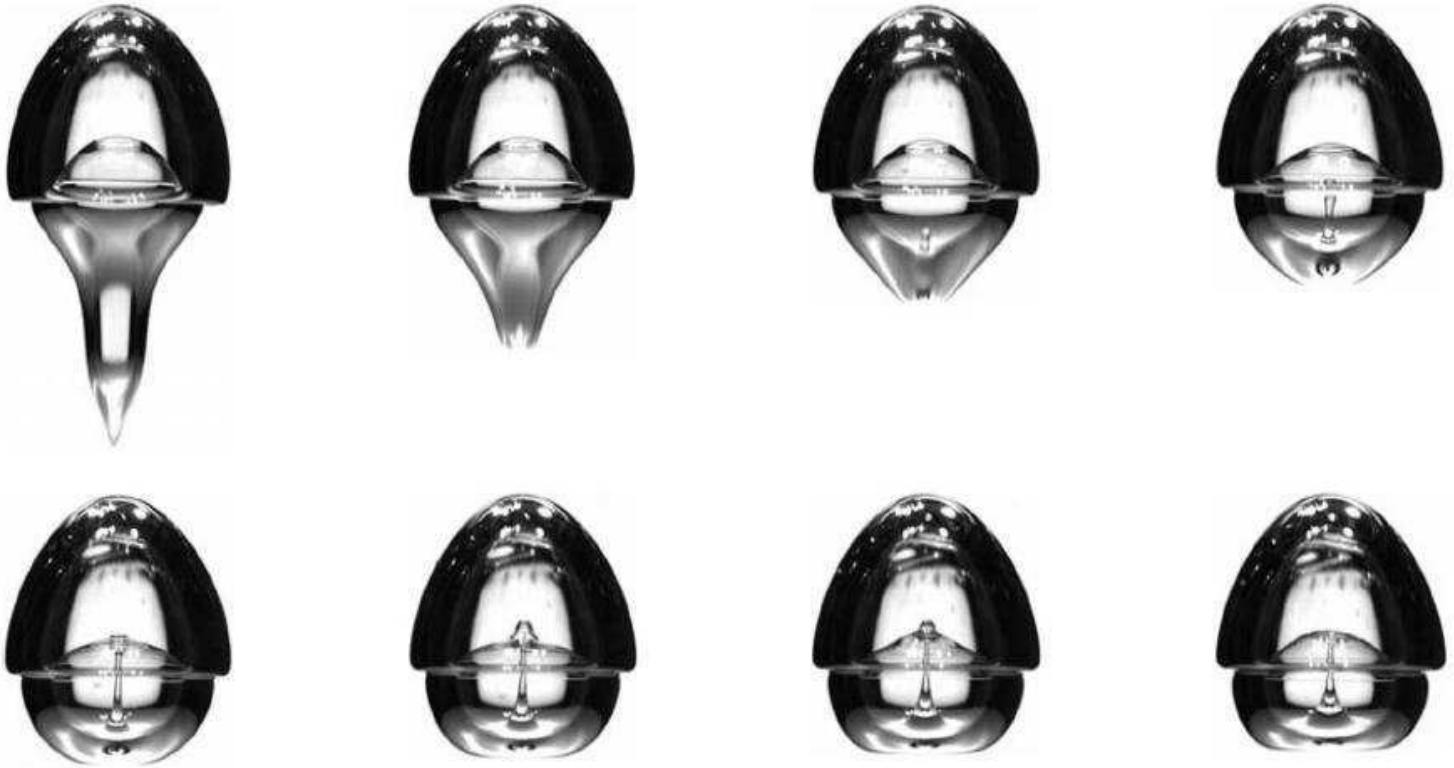
Introduction: study of a high pressure bubble burst through liquid surface

Inside the bubble …



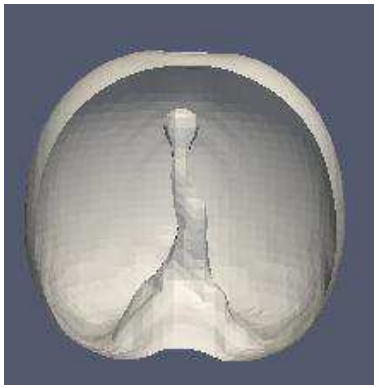
The jet propelled into the bubble is due to the flow pattern around the fast-moving bubble. It is not observed in a buoyant rise of a bubble.

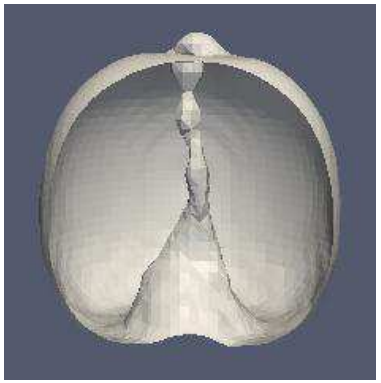
A similar experiment in lab: the bubble gun



Acknowledgment: T. Séon and A. Antkowiak, Institut d'Alembert, Paris

Comparison: Simulation-Experiment





Free-Surface model: Rough idea

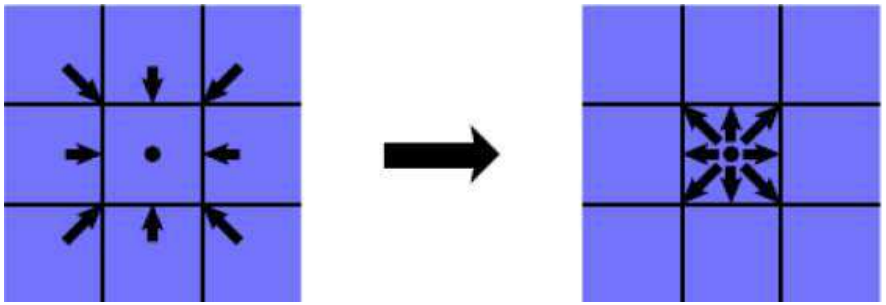
Fluid model:

- We represent a liquid-gas system but simulate only the liquid.
- The gas is represented through a constant pressure term, and its dynamics is neglected.

Volume-of-fluid approach:

- Each liquid cell does the usual lattice Boltzmann algorithm.
- An additional variable is introduced: the volume fraction.

Modified representation:
We draw the LB variables in the cell center (and not on grid vertices as usual).

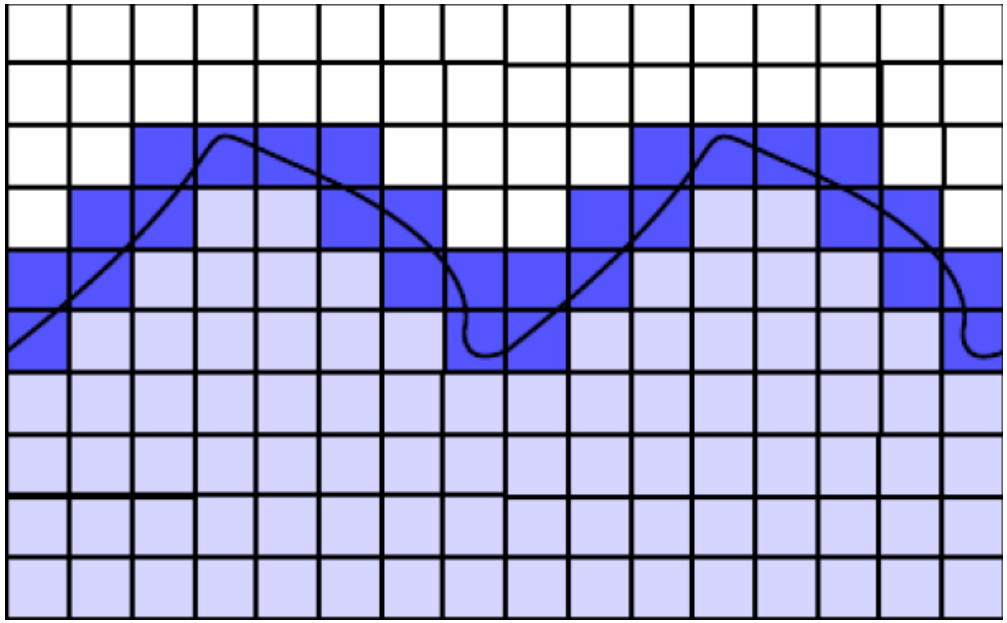


Free-surface and cell types

Each cell has a volume-fraction contained between 0 and 1.

Cell-types:
- Gas (white)
- Liquid (light-blue)
- Interface (dark-blue)

Water-tight system:
gas cells are always isolated from liquid cells through an interface layer.



Mass, Volume, and Density

Among mass, volume, and density, two variables are always computed through the hydrodynamics, while the third is tied through the following relation:

$$m = \rho \cdot V$$

V: Volume fraction.

ρ: Density.

m: Mass.

Interface Node

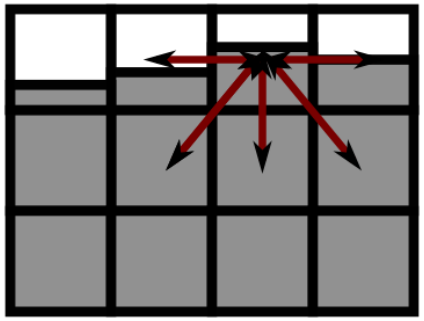
- ρ is constant (computed from "air pressure").
- m is computed from mass exchange.
- $V = m/\rho$

Bulk Node

- $V = 1$.
- ρ is computed from fluid dynamics.
- $m = \rho$

Mass exchange

- An exact mass balance is computed on interface cells.
- Like in a finite-volume approach, the mass transfer from a cell to a neighbor is exactly known.
- Algorithm: balance between population streamed to a neighbor and population received from the neighbor (in opposite direction $f_{\bar{i}}(\mathbf{x})$)



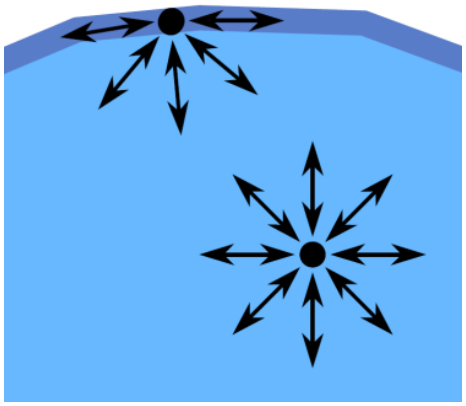
$$\Delta m_i(\mathbf{x}) = \begin{cases} 0, & \\ f_i(\mathbf{x}+\mathbf{v_i}) - f_{\bar{i}}(\mathbf{x}), & \text{From fluid neighbors.} \\ \frac{1}{2}[\varphi(\mathbf{x}) + \varphi(\mathbf{x}+\mathbf{v_i})][f_i(\mathbf{x}+\mathbf{v_i}) - f_{\bar{i}}(\mathbf{x})] & \text{From interface neighbors.} \end{cases}$$

Boundary condition on interface cells

Interface cells have unknown populations. They are computed (like boundary nodes) through a **completion scheme** as follows:

- **Velocity**: computed from the hydrodynamic equations.
- **Pressure**: constant gas pressure («air pressure») + pressure jump if surface tension.
- **Off-equilibrium** populations: Copied from opposite populations, as in standard boundary conditions.

Surface tension



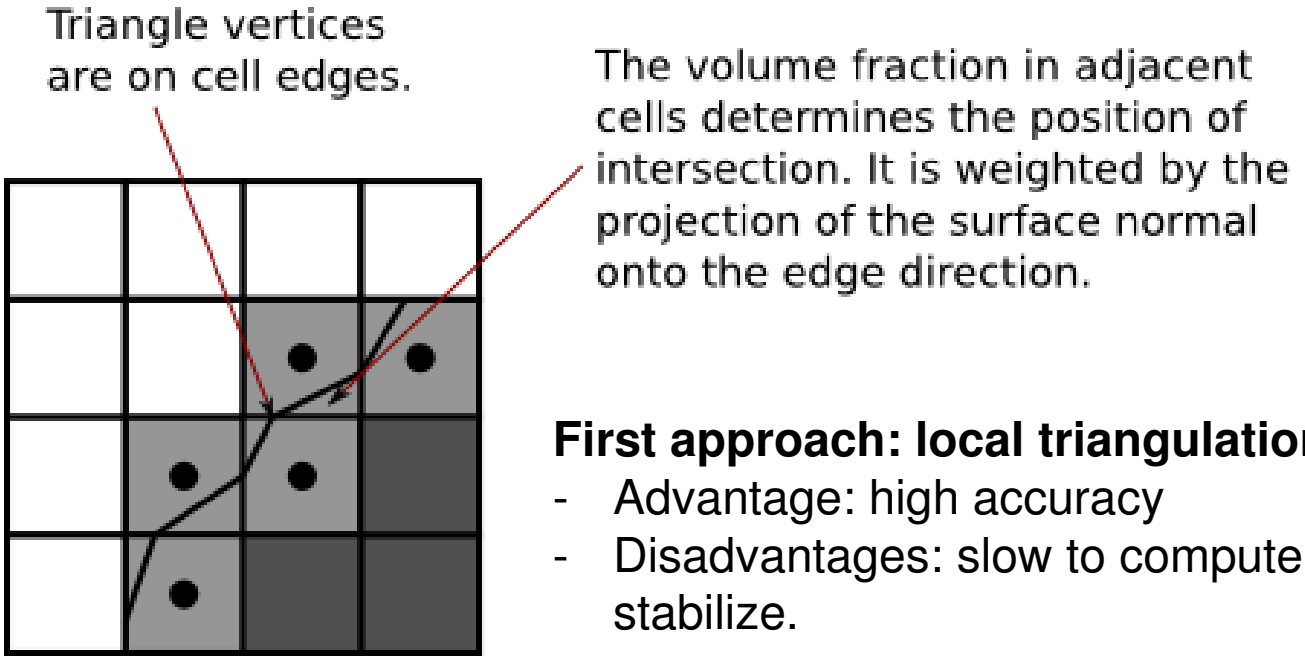
- Surface tension is induced by inter-molecular cohesion forces.
- Macroscopic effect: a pressure jump between two phases.
- Macroscopic model: the surface tension σ depends on the curvature $\kappa(\mathbf{x}, t)$.

Laplace-Young law relates curvature to the pressure jump:

$$\rho_G(\mathbf{x}) = \frac{1}{c_s^2}(p_G + 2\sigma\kappa(\mathbf{x}, t))$$

In the general case, a shearing term is added.

Computation of the curvature:



First approach: local triangulation of the surface

- Advantage: high accuracy
- Disadvantages: slow to compute and difficult to stabilize.

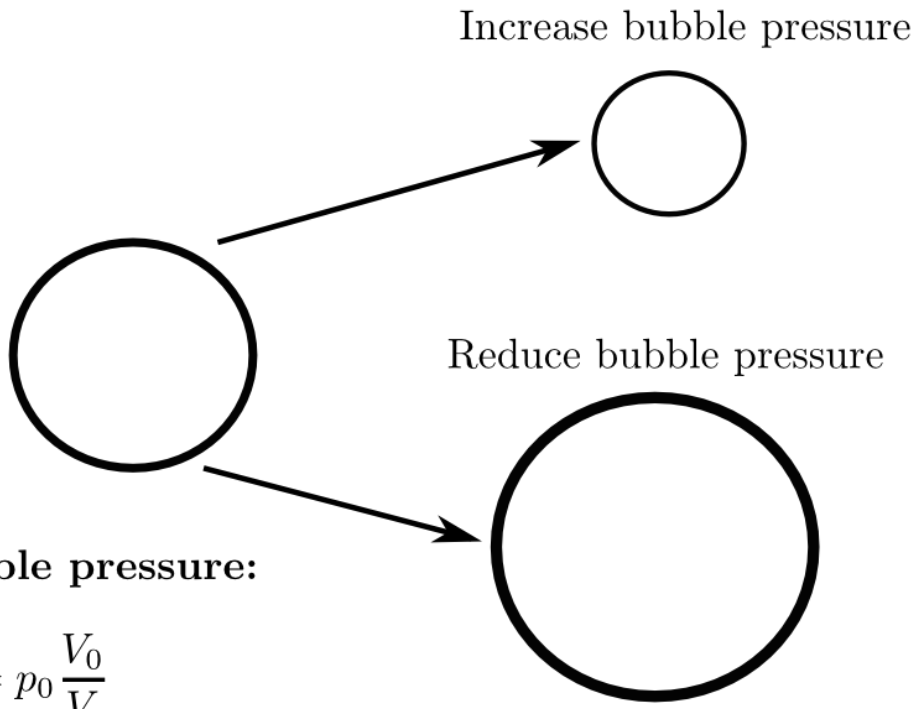
Second approach: second derivatives of volume fraction through finite differences

- Disadvantage: requires smooth volume-fraction field. Workaround: apply a filter to the volume-fraction.
- Advantages: very fast to compute and stable.

Rising bubbles: correction of the internal bubble pressure

Issue: the “air” is not simulated and can therefore not conserve volume.
Under gravity, air bubbles simply collapse.

Solution: trace the volume of air bubbles, and adjust the air pressure.

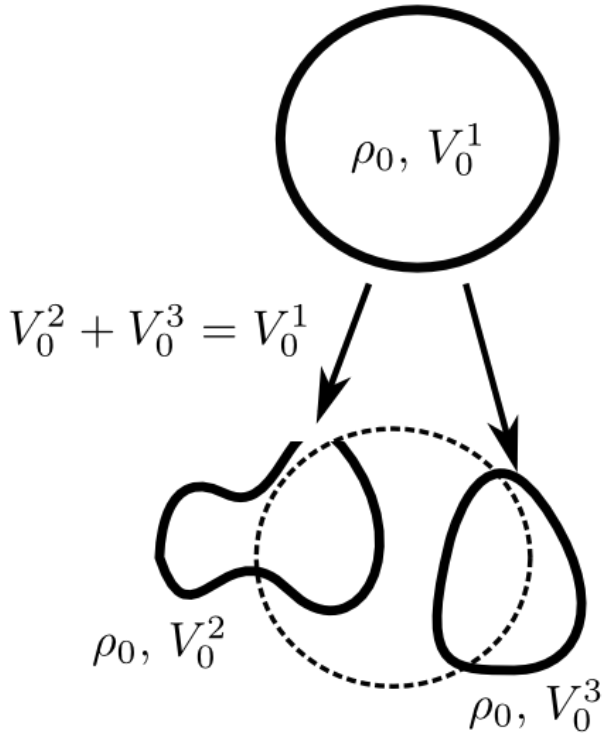


Internal bubble pressure:

$$p = p_0 \frac{V_0}{V}$$

- p_0: reference pressure.
- V_0: reference volume.

Pressure correction: algorithm



- Identify bubbles at time t.
- Identify bubbles at time $t+1$.
- Overlap with previous time and decide if splitting has occurred.
- Redistribute reference volume.

Challenges:

- At every time step, all bubbles must be identified through a bucket-fill algorithm which must be fully parallel.
- Size of bubbles is unpredictable: all-to-all MPI communication is needed.
- Pattern matching algorithm is needed to detect bubble splitting and merging.

From free-surface to two-phase

Idea:

- Each of the two fluids implements a free-surface algorithm.
- On the interface, velocity continuity is enforced by computing an average between the velocity of the two fluids.
- On the interface, pressure continuity is enforced by means of an appropriate algorithm.

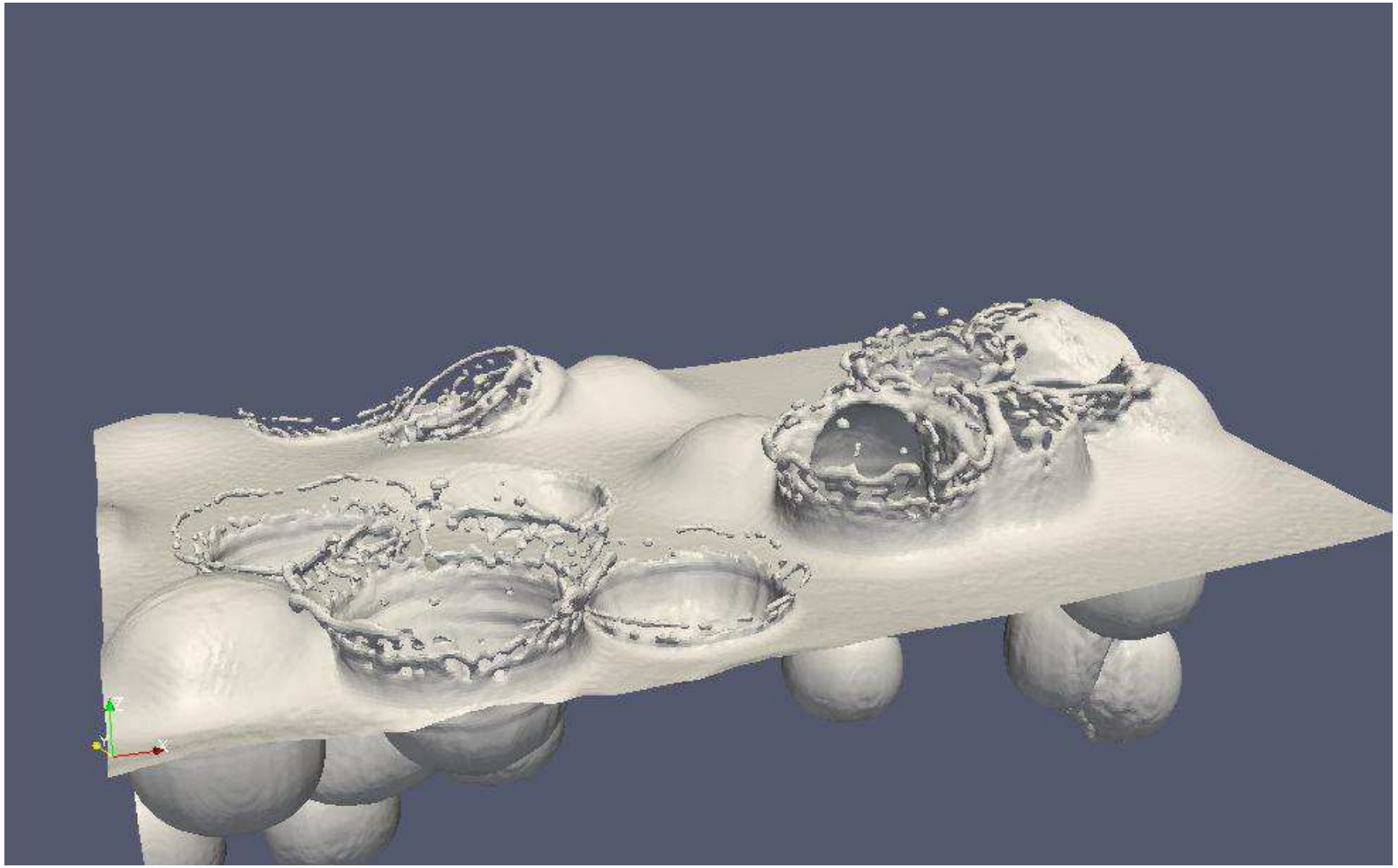
How efficient is the multi-phase fluid model?

Back to our bubble bursting through a surface

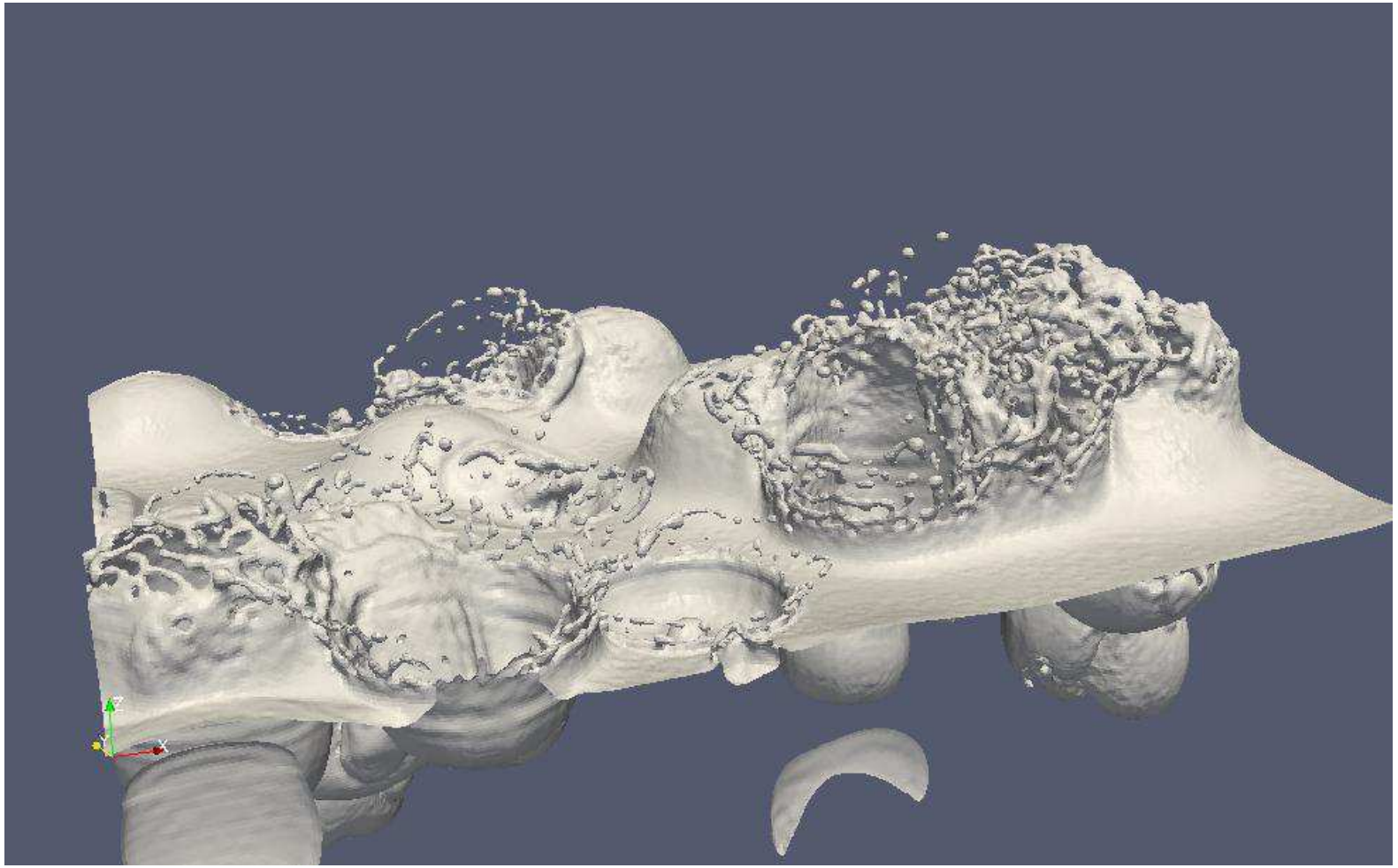
- A single bubble: 4 hours on a desktop computer.
- 30 bubbles: 4 hours on a 48-core cluster.

30 bubbles bursting through the surface

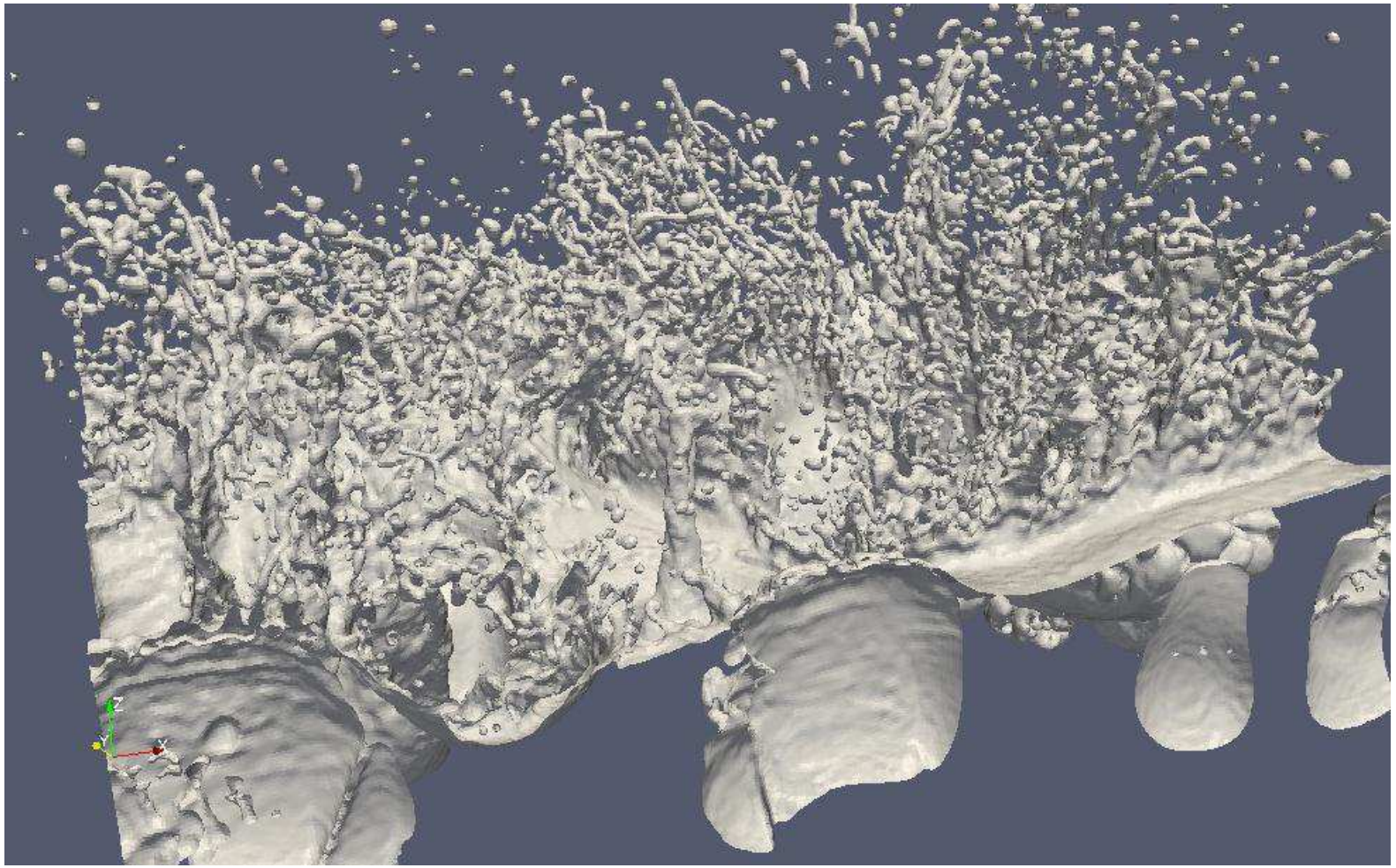
30 bubbles bursting through the surface



30 bubbles bursting through the surface

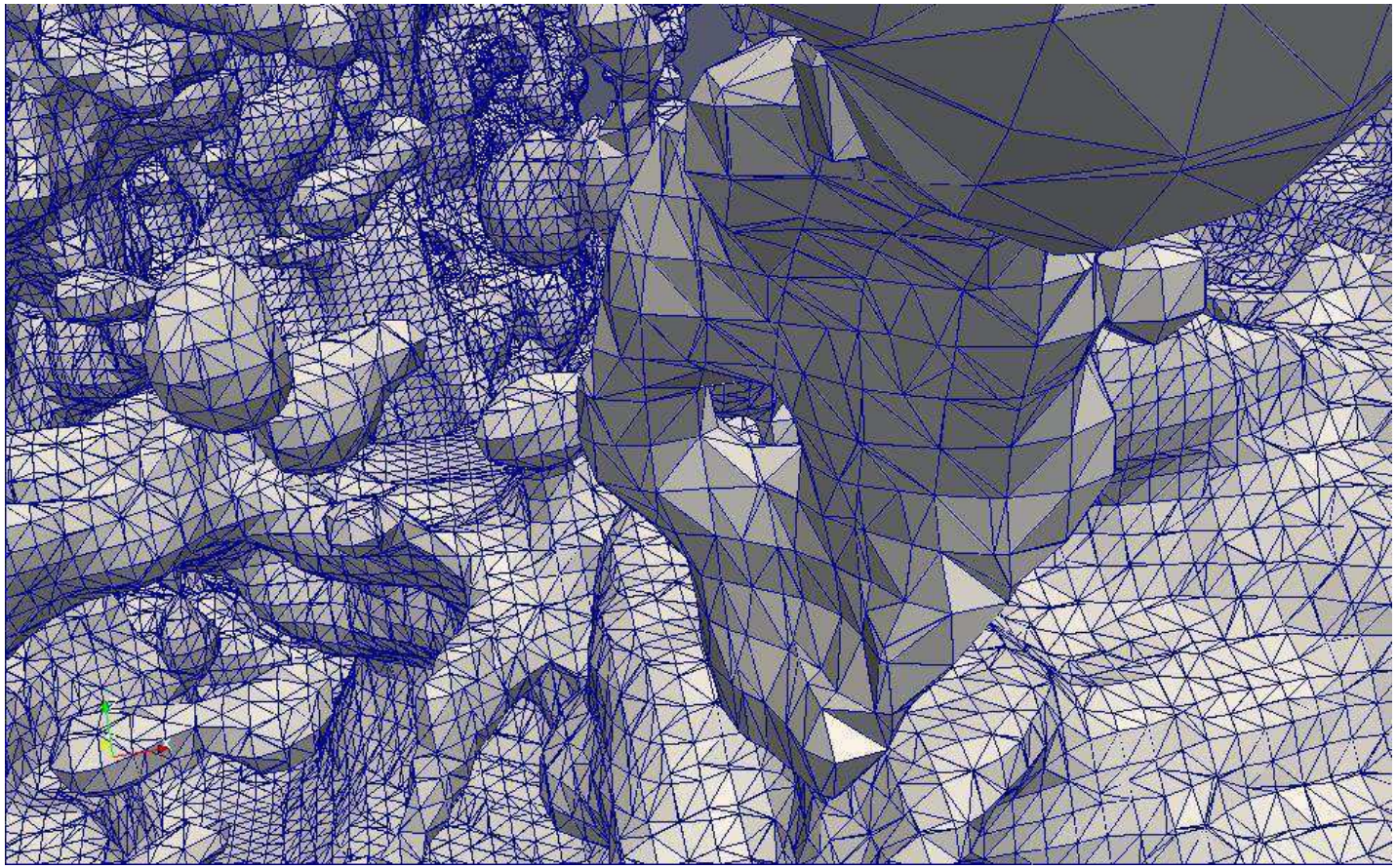


A first example: 30 bubbles bursting through the surface



30 bubbles bursting through the surface

30 bubbles bursting through the surface



30 bubbles bursting through the surface