A High Performance Computing Framework for Multiphase, Turbulent Flows on Structured Grids

Petr Karnakov

Computational **S**cience and **E**ngineering **Lab**
ETH Zürich

With:

Fabian Wermelinger
Michail Chatzimanolakis
Dr. Sergey Litvinov
Prof. Petros Koumoutsakos

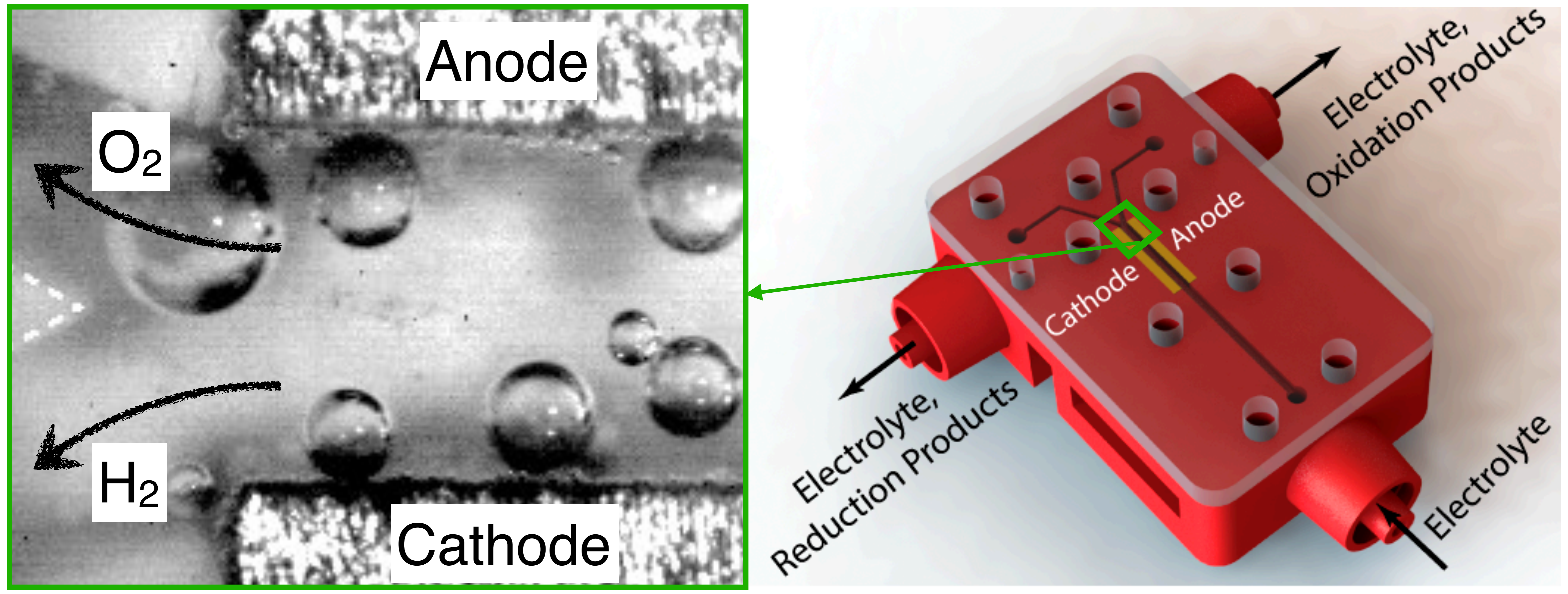
FNSNF
Swiss National Science Foundation
grant CRSII5_173860

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre
projects s754, s931

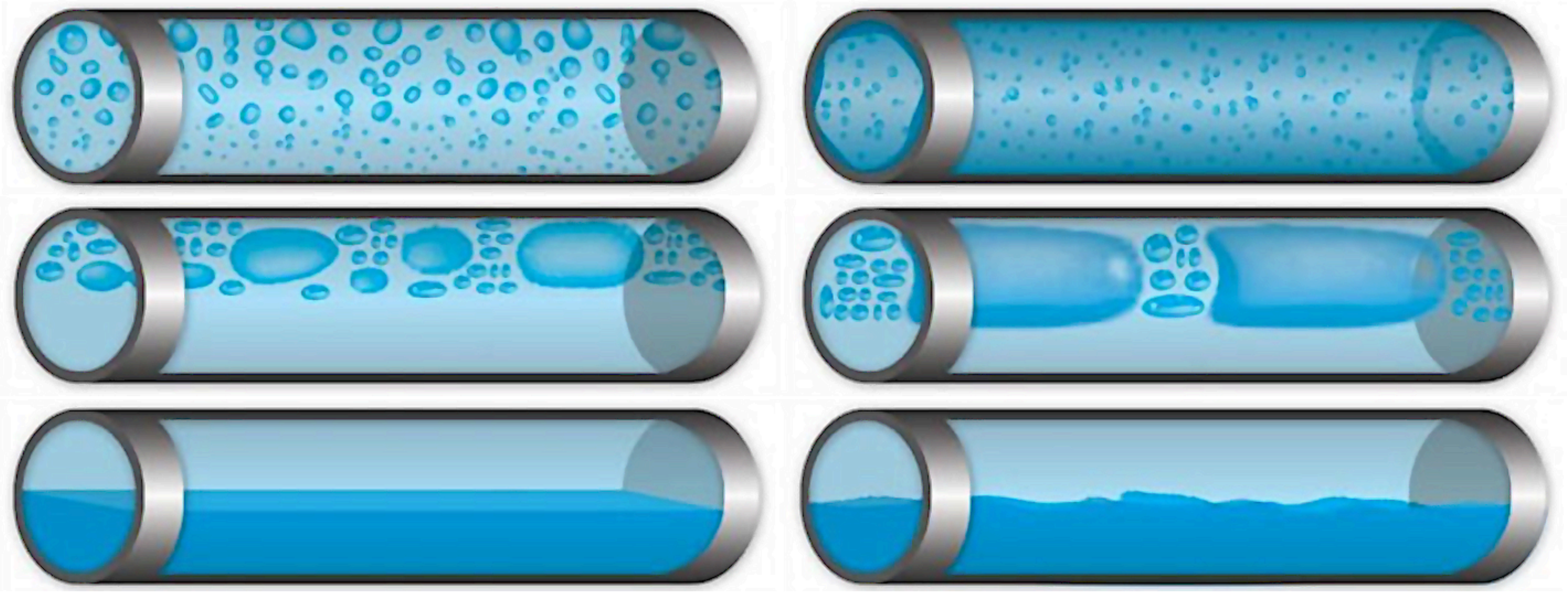
Multiphase flows

Electrochemical cells

[Hashemi 2019]

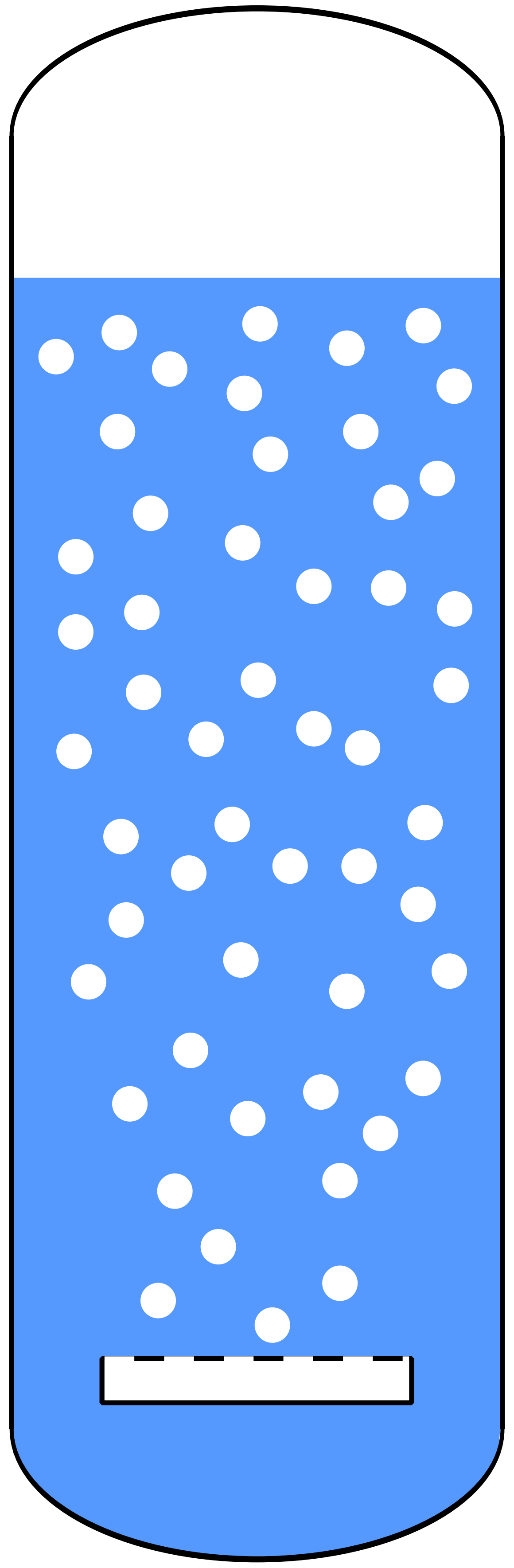


Pipes



[Bratland 2010]

Bubble column reactors



[wikipedia]

Outline

Numerical model

- new method for curvature estimation
  improving the accuracy at low resolution

Implementation

- blockwise processing with coroutines for modularity

Test cases

- curvature of a sphere
- translating droplet

Applications

- bubble coalescence
- Taylor-Green vortex with bubbles
- plunging jet with air entrainment

Numerical model

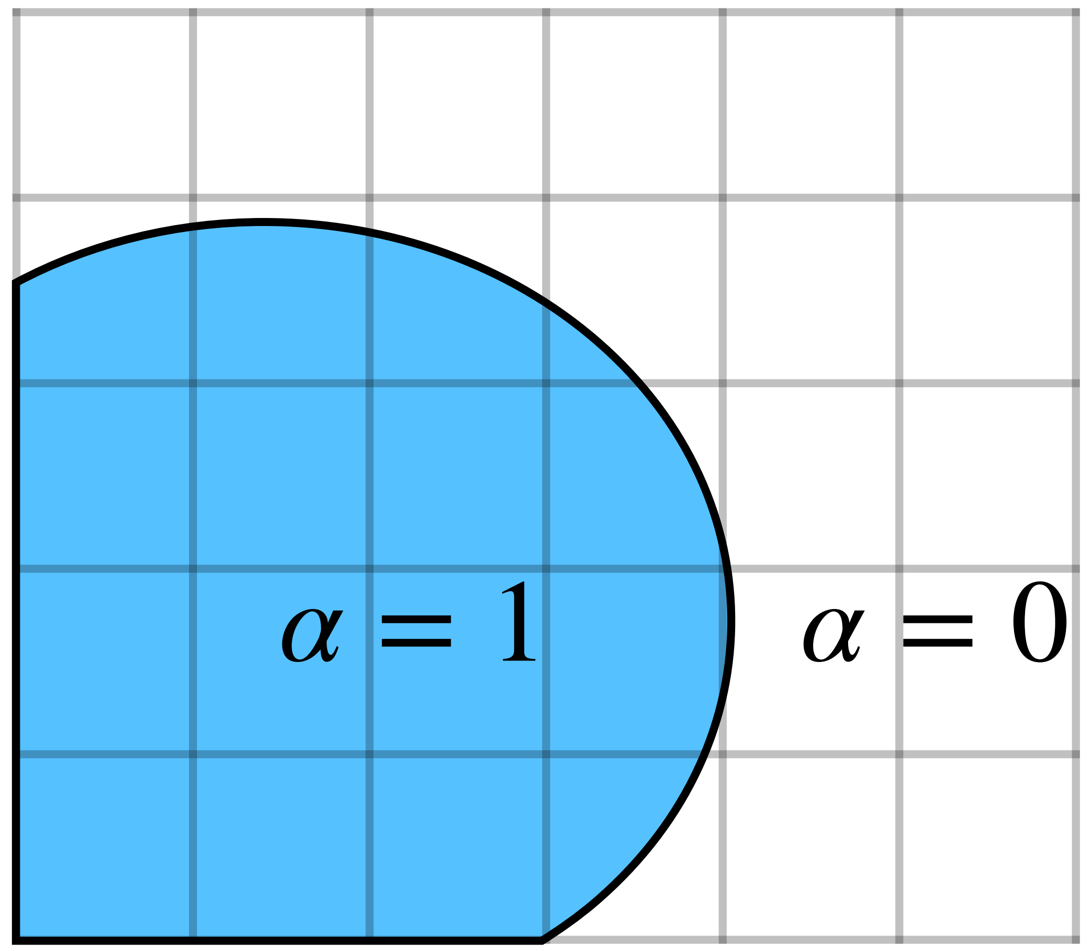
Model

Two-phase incompressible flow

- Navier-Stokes equations

$$\nabla \cdot \mathbf{u} = 0$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}\right) = -\nabla p + \nabla \cdot \left(\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)\right) + \mathbf{f}_\sigma$$



- Advection of volume fraction

$$\frac{\partial \alpha}{\partial t} + (\mathbf{u} \cdot \nabla)\alpha = 0$$

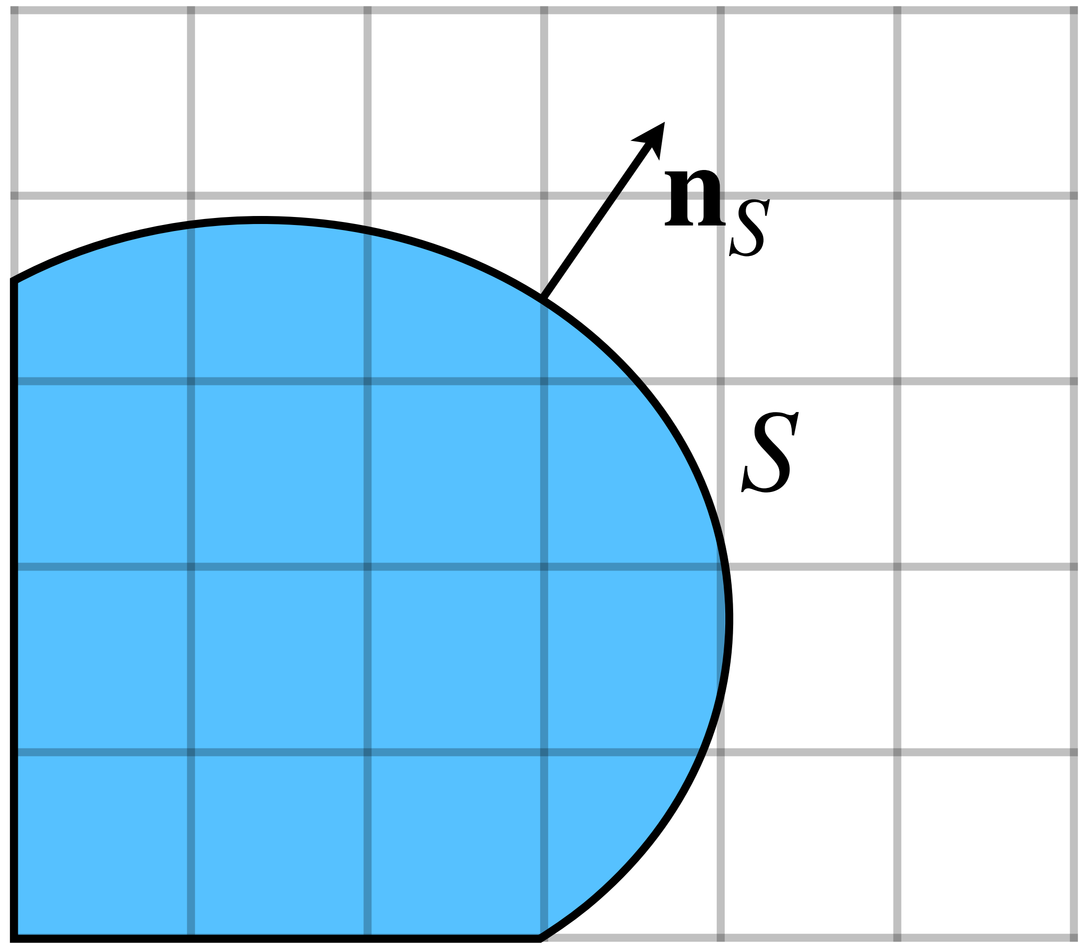
$$\rho = (1 - \alpha)\rho_1 + \alpha\rho_2$$
$$\mu = (1 - \alpha)\mu_1 + \alpha\mu_2$$

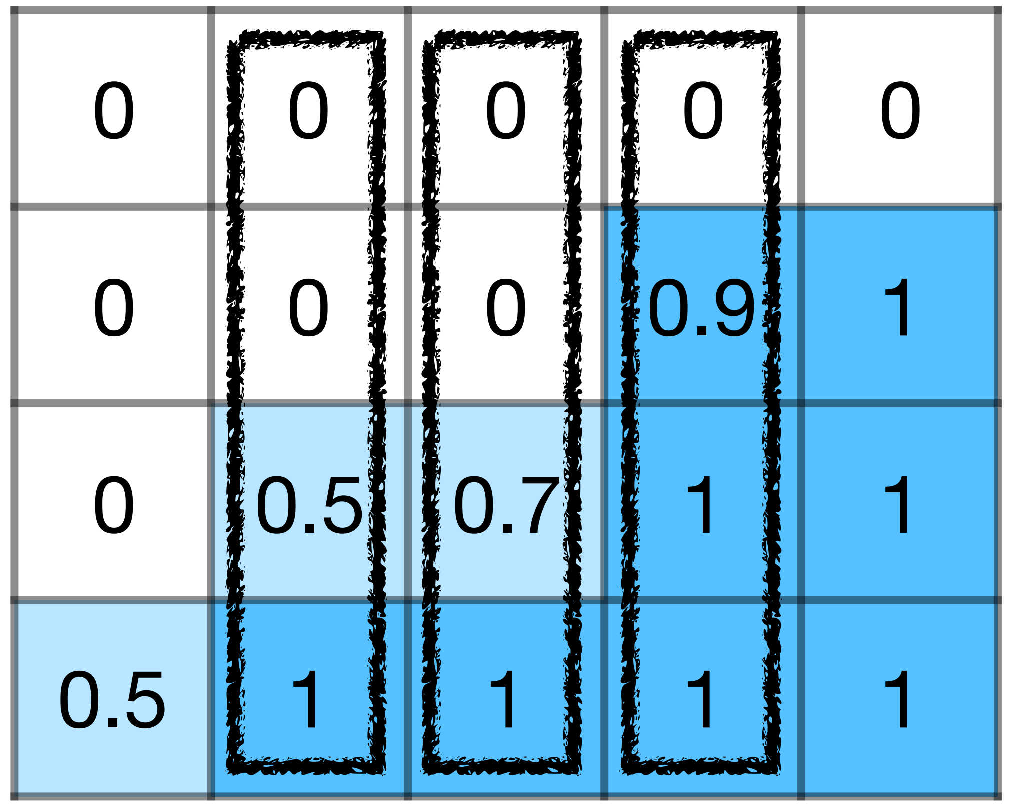
- Discretization
  - Finite Volume on Cartesian grid
  - SIMPLE for pressure coupling [Patankar 1979]
  - VOF advection with reconstruction [Aulisa 2007]
  - Linear solvers from Hypre [Falgout 2002]

Surface tension

- Calculation of surface tension $\mathbf{f}_\sigma = \sigma\,\kappa\,\mathbf{n}_S\,\delta_S$ relies on interface curvature $\kappa = \nabla_S \cdot \mathbf{n}_S$

- Existing methods show poor accuracy at low resolution
  - gradients of volume fraction [Brackbill 1992]
  - level-set [Sussman 1998]
  - height functions [Cummins 2005]
  - generalized height functions [Popinet 2009] with parabolic fit to mixed heights





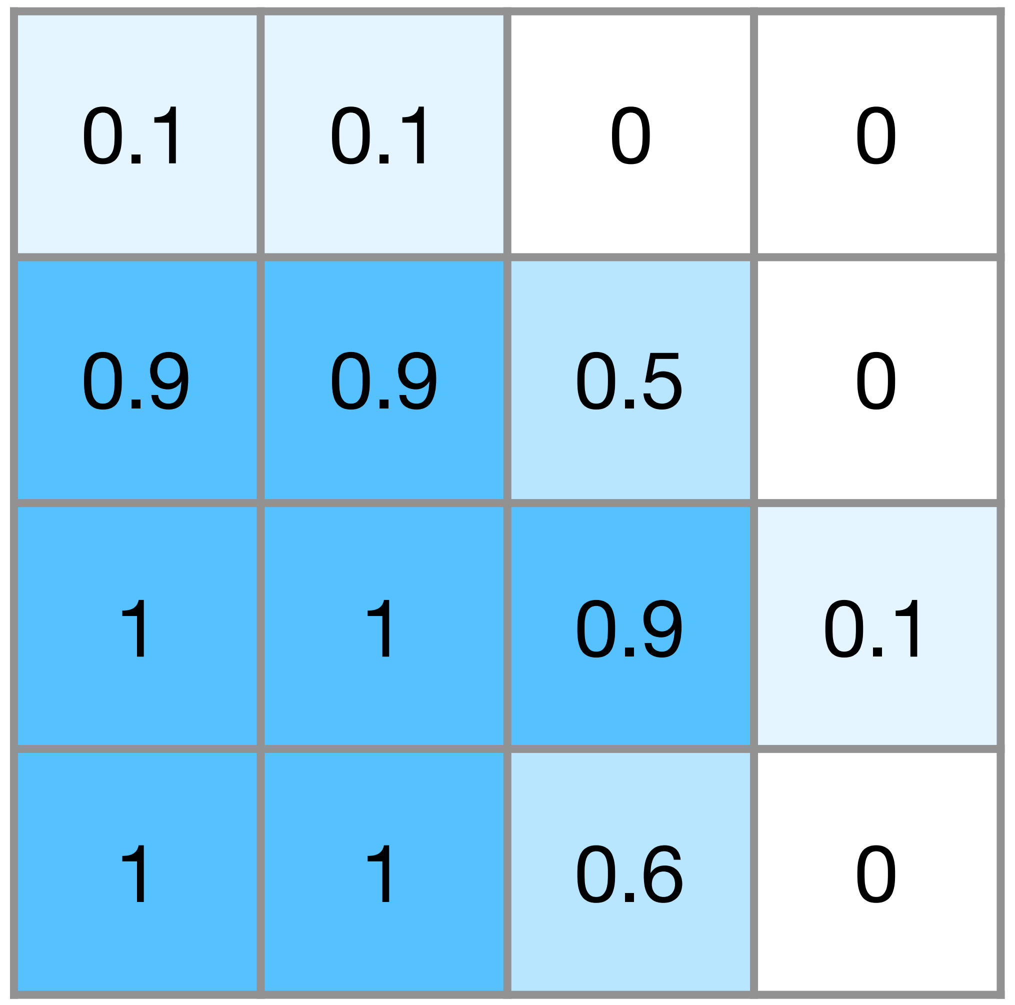
0	0	0	0	0
0	0	0	0.9	1
0	0.5	0.7	1	1
0.5	1	1	1	1

Piecewise linear interface

Line segments (2D) or polygons (3D) from volume fractions [Aulisa 2007]

- Volume fraction field

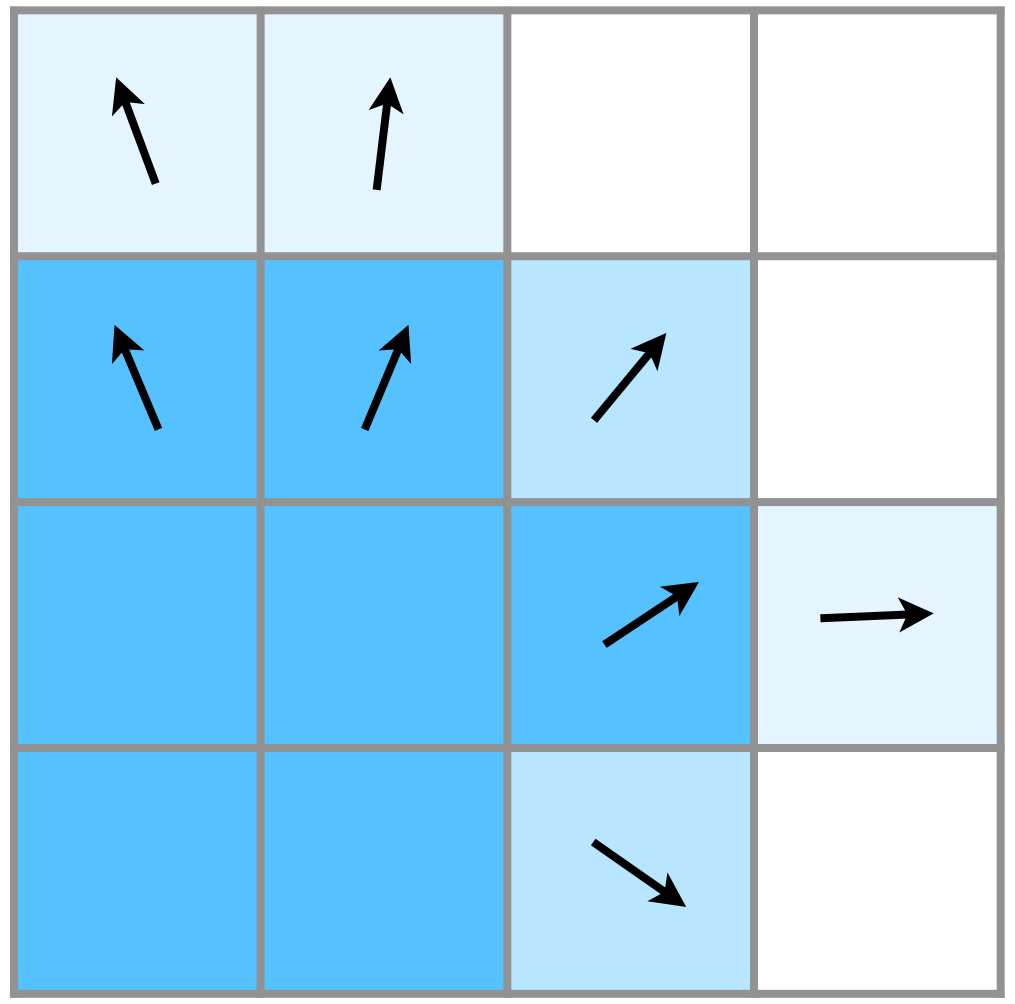
α



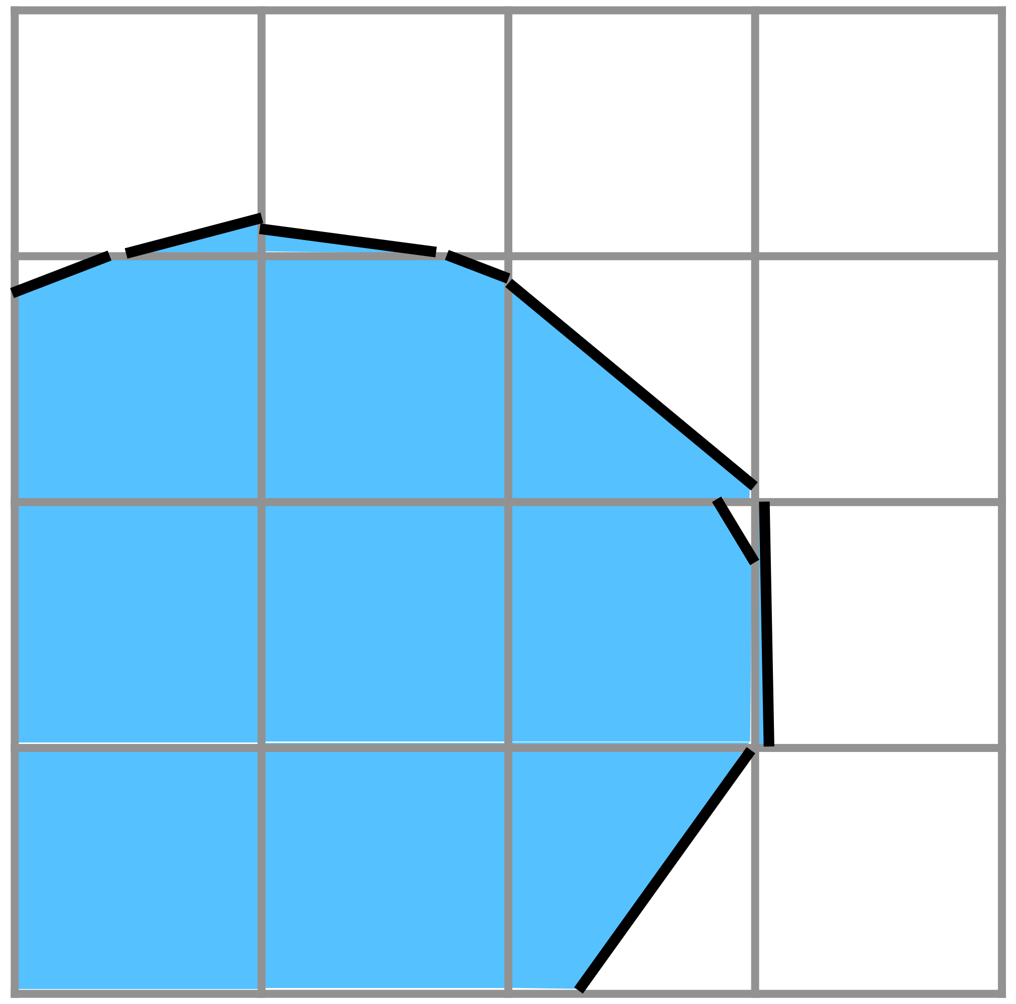
0.1	0.1	0	0
0.9	0.9	0.5	0
1	1	0.9	0.1
1	1	0.6	0

- Normals as gradients

$$\mathbf{n} = -\nabla\alpha$$

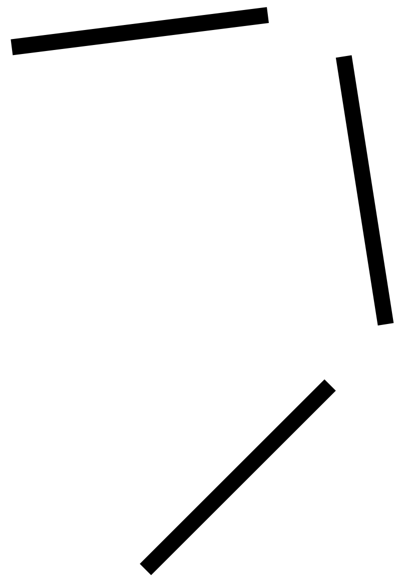


- Lines cutting the cells at given volume fraction

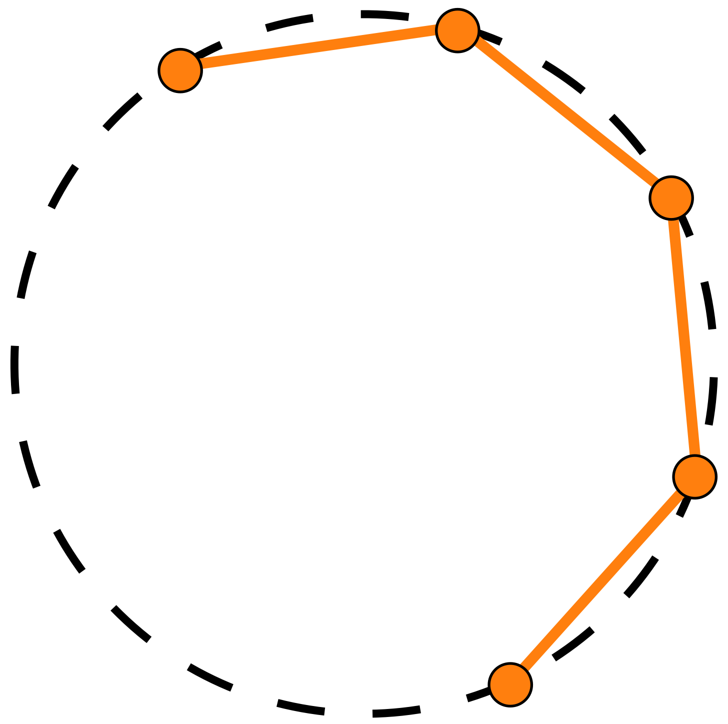


Proposed method

Estimation of curvature from line segments using particles



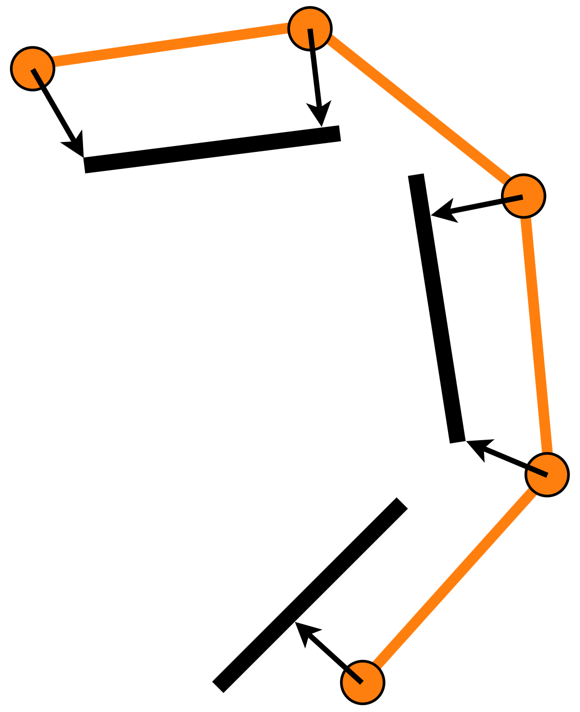
1. Line segments



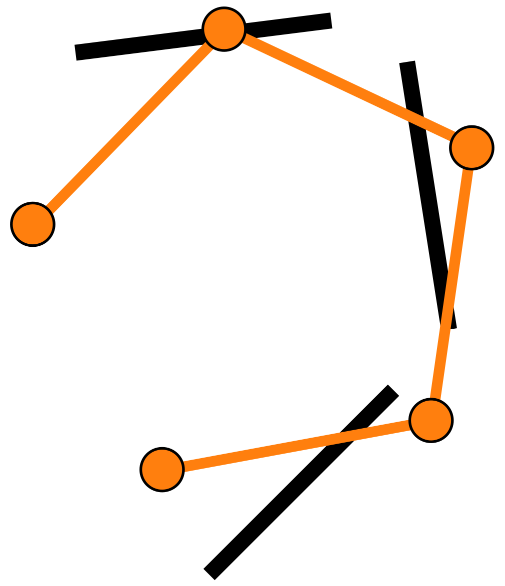
2. Constrained particles

- fixed distance
- uniform angle

⇒ particles belong to a circle



3. Attraction forces



4. Curvature from equilibrium positions

Proposed method

Evolution of constrained particles

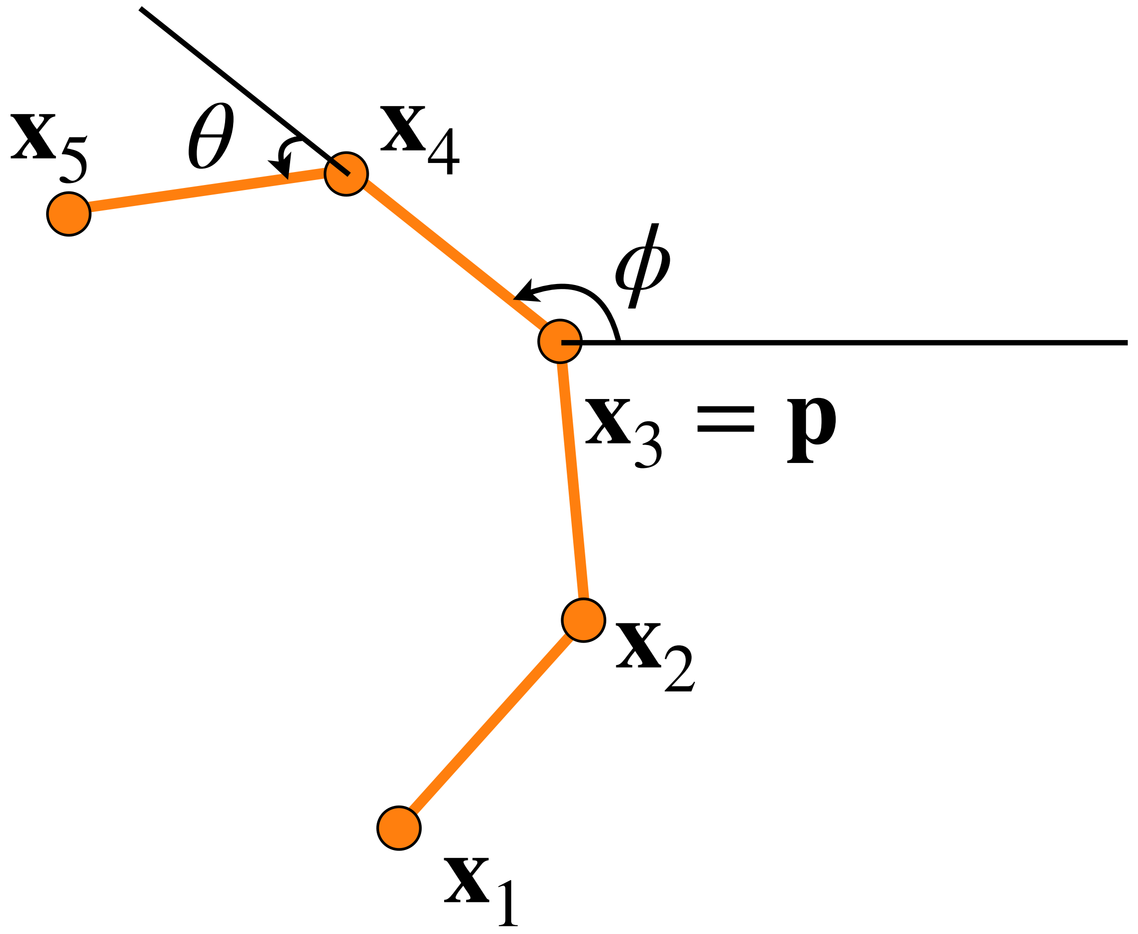
- Constraints are satisfied by parametrization

$$\mathbf{x}_i = \mathbf{x}_i(\mathbf{p}, \phi, \theta) \quad i = 1,\ldots,5$$

$\mathbf{p}$ position of central particle

ϕ orientation angle

θ bending angle



- Force $\mathbf{f}_i$ attracts $\mathbf{x}_i$ to nearest point on the interface

- Each step applies corrections from projected forces

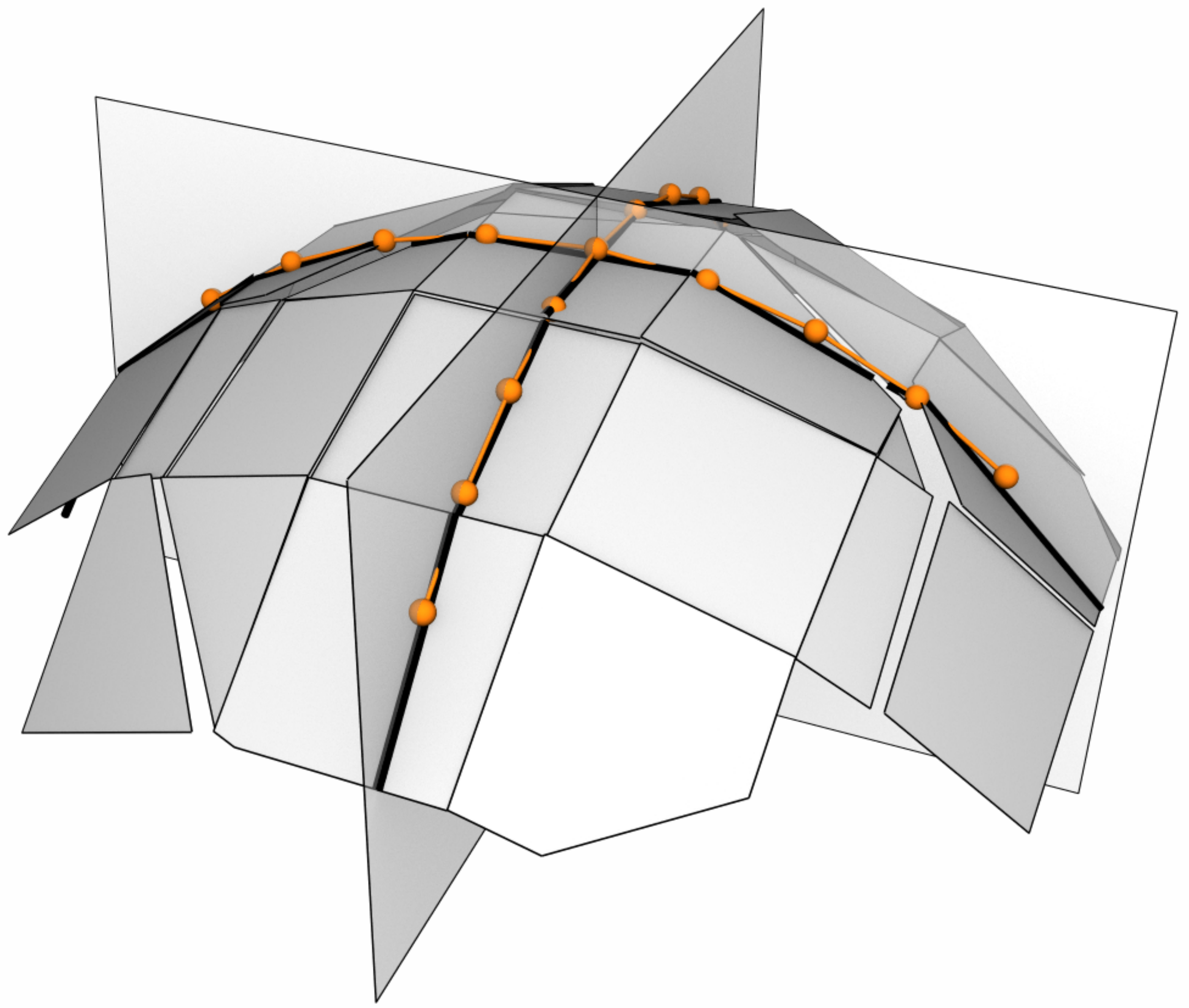
$$\Delta\phi = \sum_i \mathbf{f}_i \cdot \frac{\partial \mathbf{x}_i}{\partial \phi} \Big/ \sum_i \frac{\partial \mathbf{x}_i}{\partial \phi} \cdot \frac{\partial \mathbf{x}_i}{\partial \phi}$$

(similar for $\mathbf{p}$ and θ)

Proposed method

Mean curvature in 3D

- Interface consists of polygons
- Mean curvature is
  the average over cross-sections
  normal to the interface
- Cross-section consists of line segments
  ⇒ problem reduced to 2D

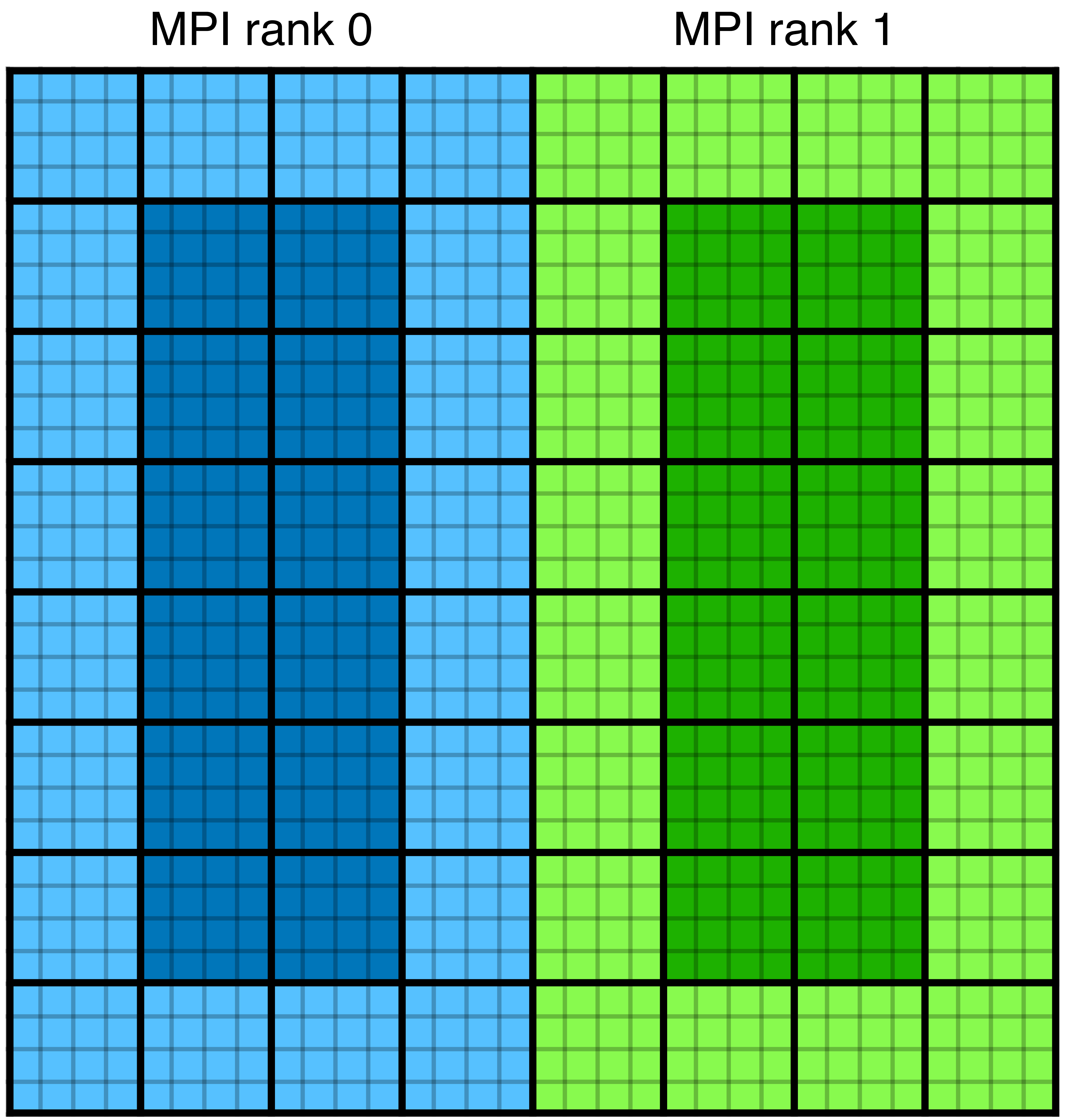


Implementation

Blockwise processing

- Each rank splits its subdomain to cubic blocks
  - cache utilization
  - compute-transfer overlap:
communication while computing inner blocks
- Used in Cubism-MPCF [Rossinelli 2013, Wermelinger 2018]

  Multiphase compressible flows
  Gordon Bell prize 2013 for high throughput computation
- Drawback: lack of modularity



Example

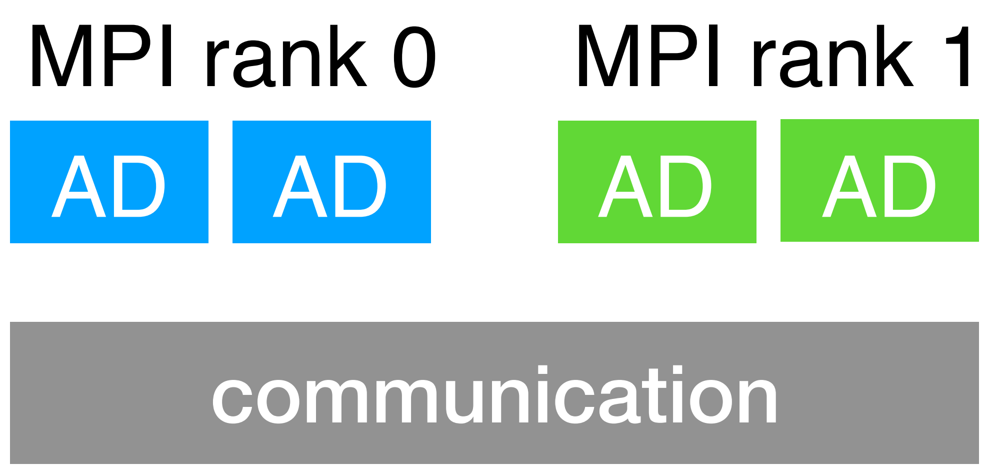
- Block processor executes kernels on blocks and calls MPI to exchange ghost cells
- Adding communication requires changing the block processor

- Algorithm 1: one stage

$$\mathbf{u}^{n+1} = \mathrm{AD}(\mathbf{u}^n)$$

block processor

```
for (Block b : bb) {
  AD(b);
}
Comm();
```



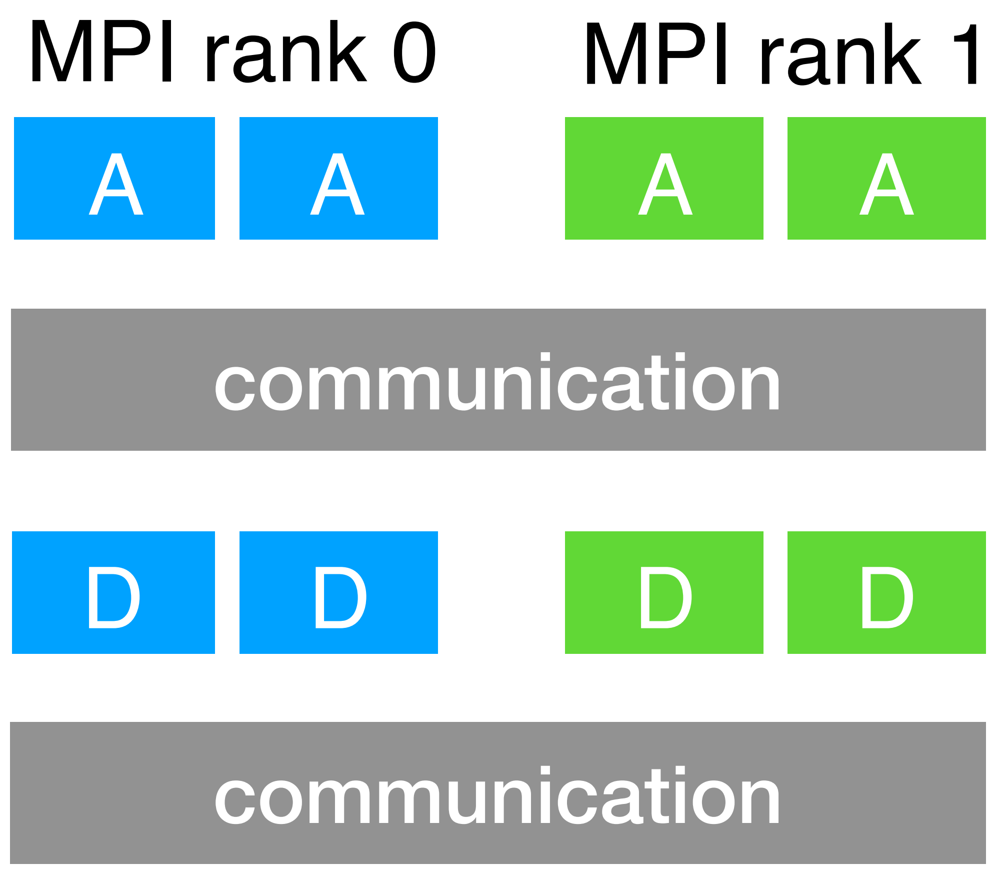
- Algorithm 2: two stages

$$\mathbf{u}^{n+1/2} = \mathrm{A}(\mathbf{u}^n)$$

$$\mathbf{u}^{n+1} = \mathrm{D}(\mathbf{u}^{n+1/2})$$

block processor

```
for (Block b : bb) {
  A(b);
}
Comm();
for (Block b : bb) {
  D(b);
}
Comm();
```



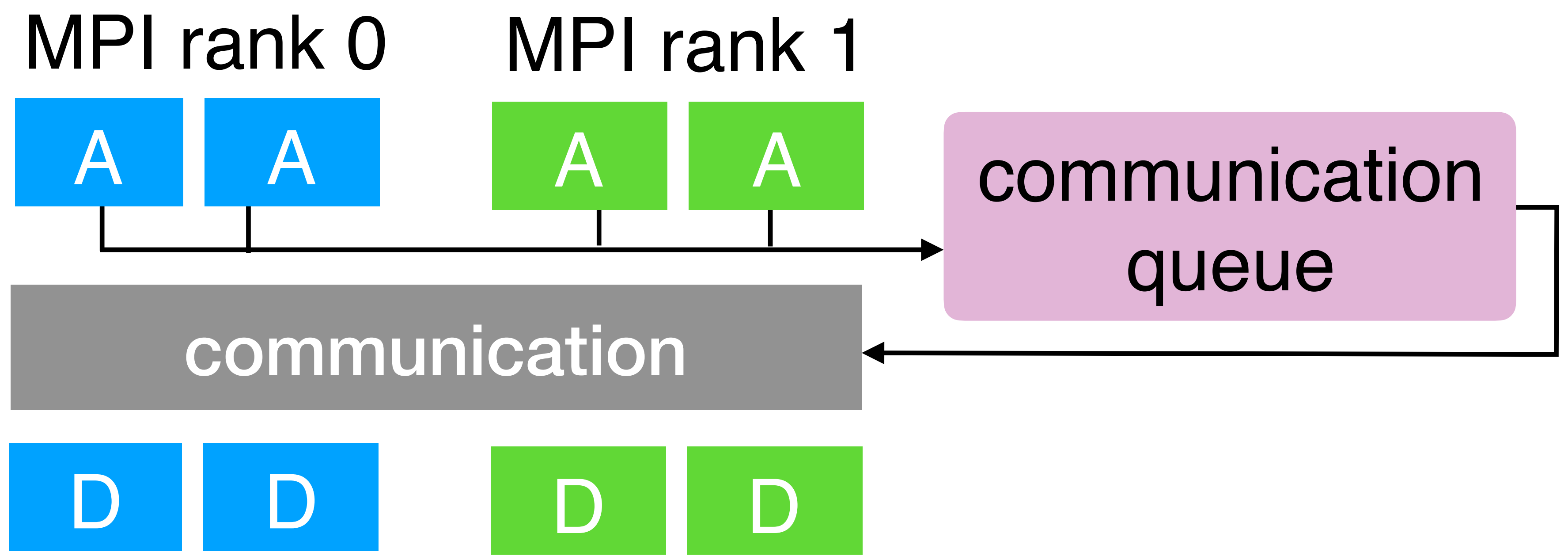
Kernels:

A	advection
D	diffusion
AD	advection+diffusion

Coroutines

kernel

```
void AD(Block b, Queue q) {
  A(b);
  q.RequestComm(b);
  yield;
  D(b);
}
```



block processor

```
Queue q;
while (!q.Done()) {
  for (Block b : bb) {
    AD(b, q);
  }
  Comm(q);
}
```

emulation of coroutines in C++

```
void AD(Block b, Queue q) {
  Stages s(b);
  if (s()) {
    A(b);
    q.RequestComm(b);
  }
  if (s()) {
    D(b);
  }
}
```

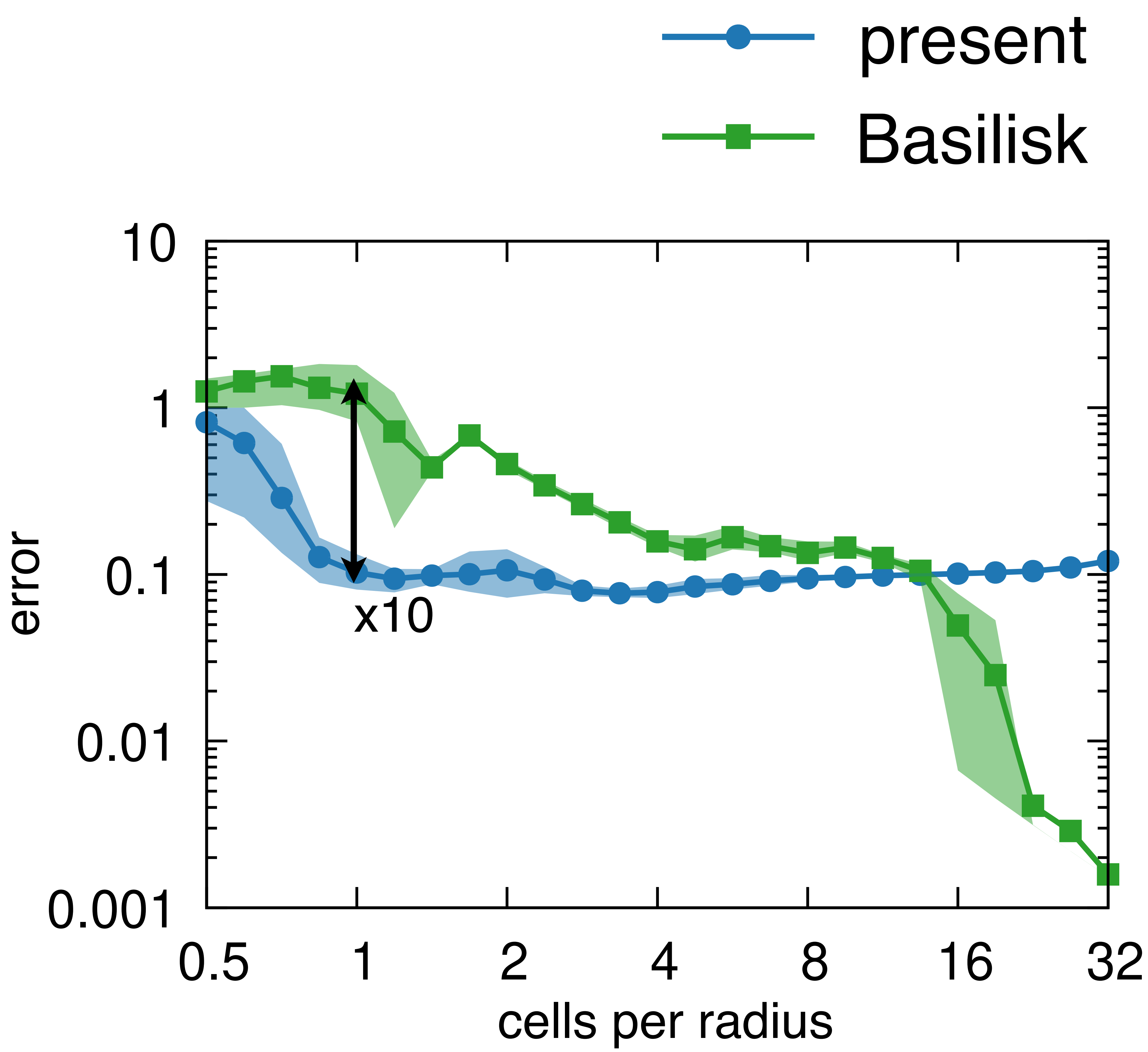
- Kernels can request communication and suspend
- Universal block processor executes the requests
- Enables modularity
  - kernels control communication
  - allows nested calls

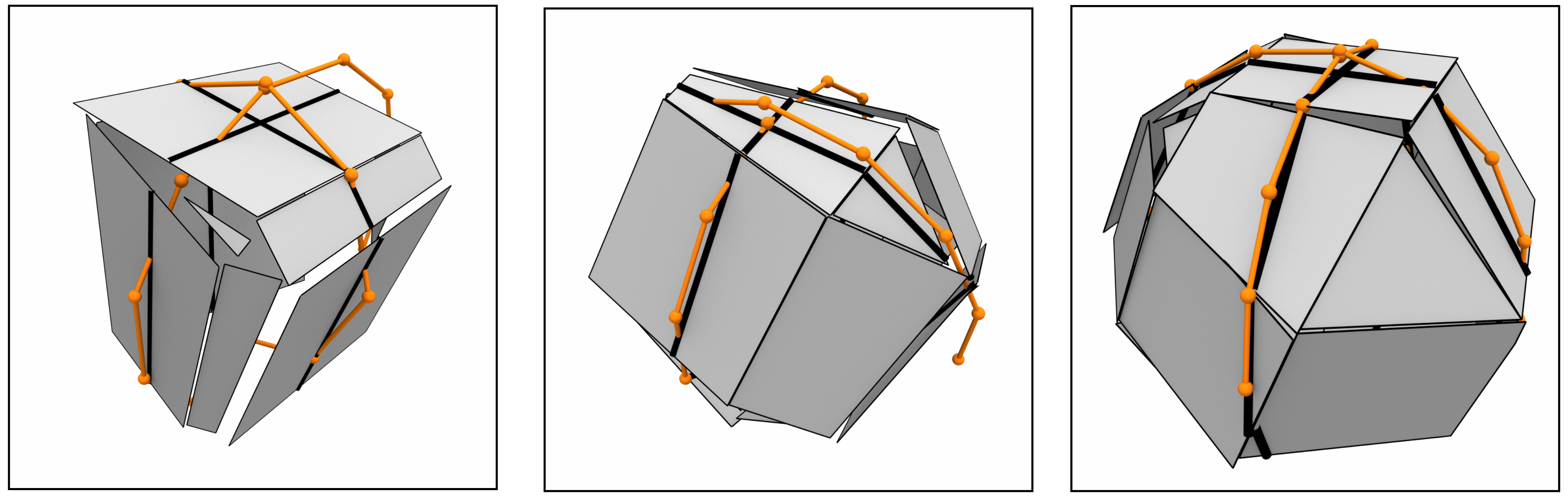
Test cases

Curvature of a sphere

- Error in curvature relative to exact value
- Comparison to Basilisk

  generalized height-function method [Popinet 2009] [basilisk.fr]
- Present method
  - more accurate at resolutions below 12 cells
  - error below 10% even with one cell per radius

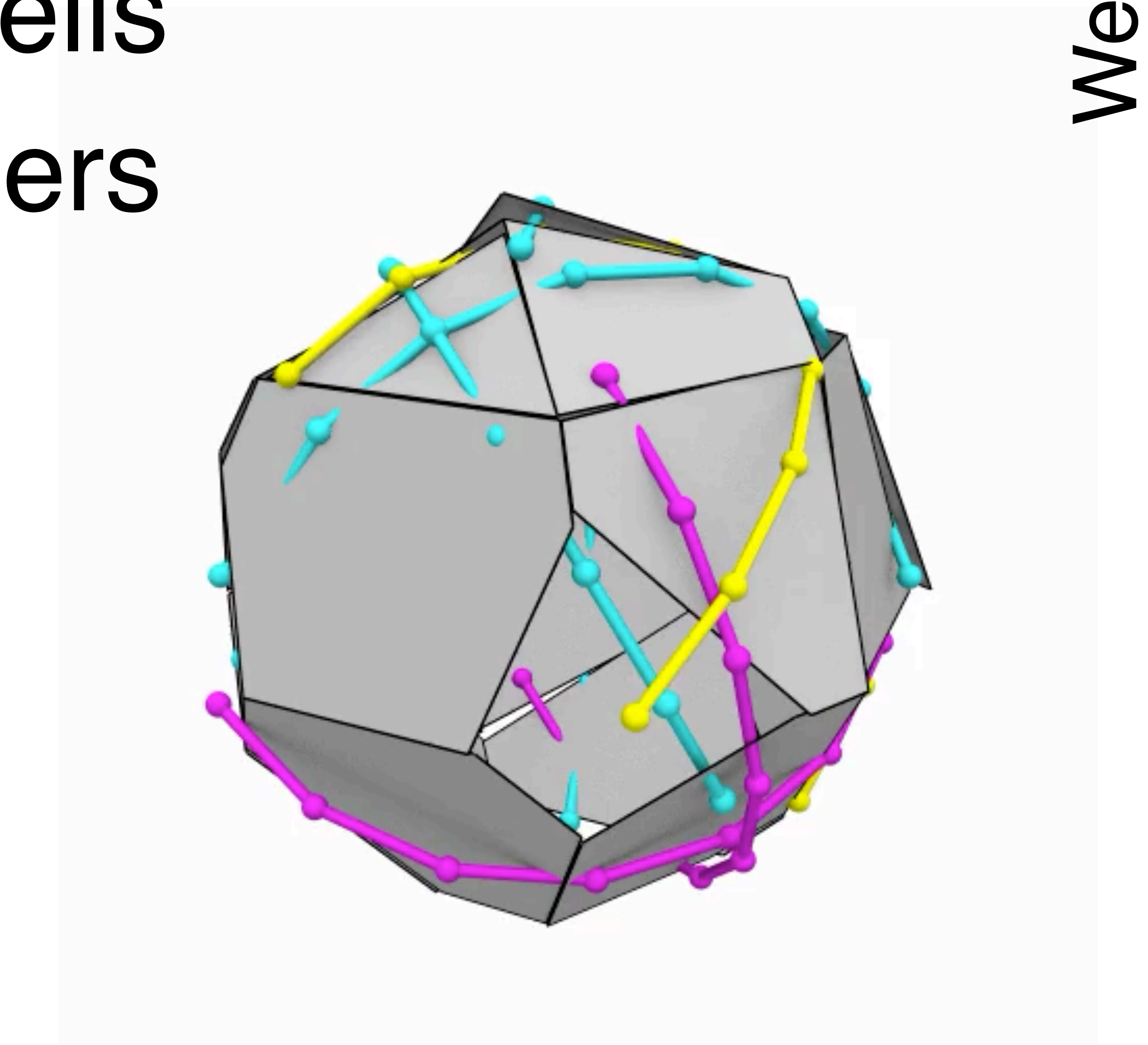




cells per radius:	0.59	0.84	1.19
error:	0.42	0.08	0.09

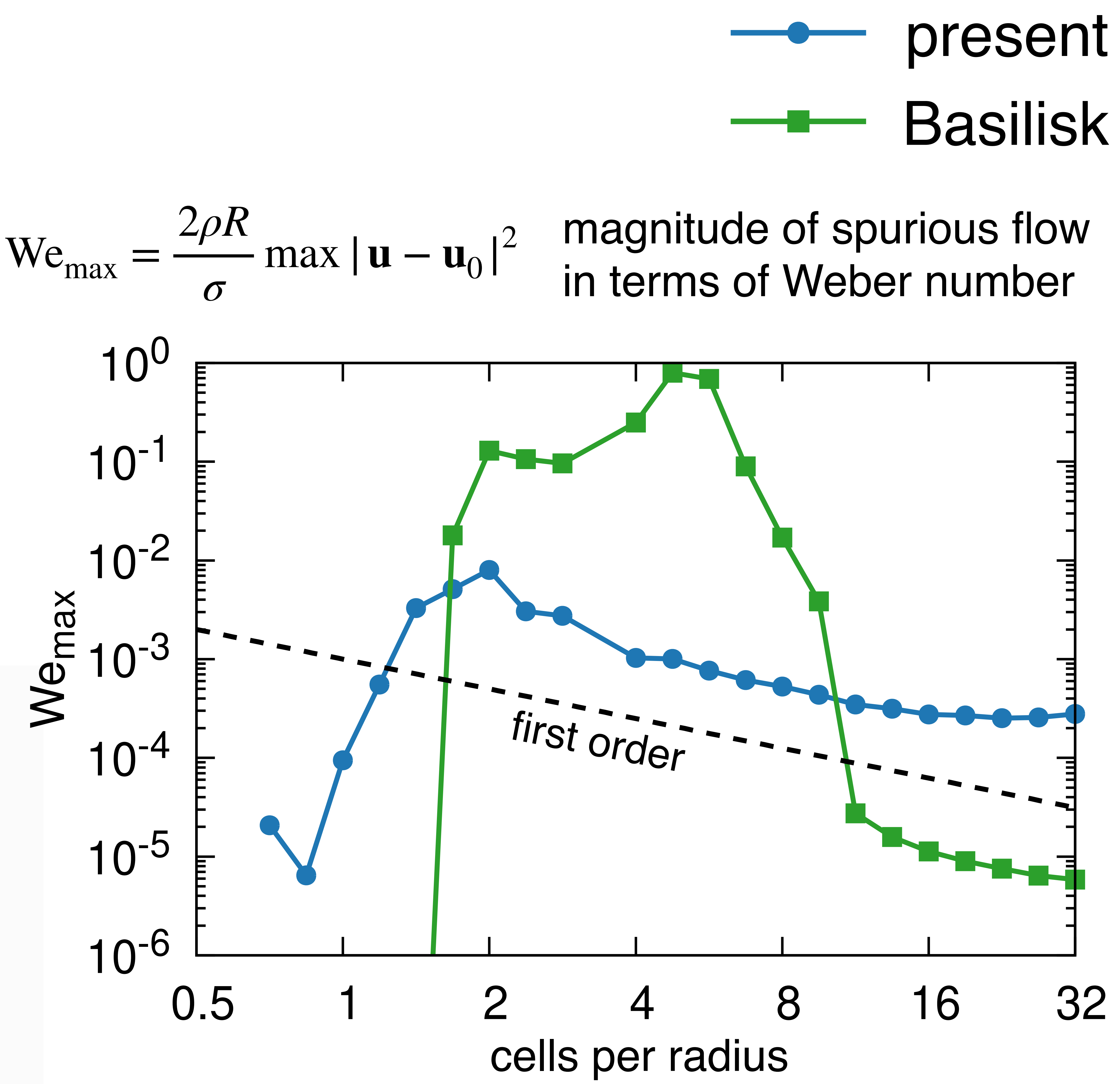
Translating droplet

- Uniform initial velocity $\mathbf{u}_0$
- Spurious flow created by inaccuracies in curvature
- Present method
  - lower magnitude of spurious flow at resolutions below 10 cells
  - small droplets act as tracers



$$\mathrm{We}_{\max} = \frac{2\rho R}{\sigma} \max|\mathbf{u}-\mathbf{u}_0|^2$$

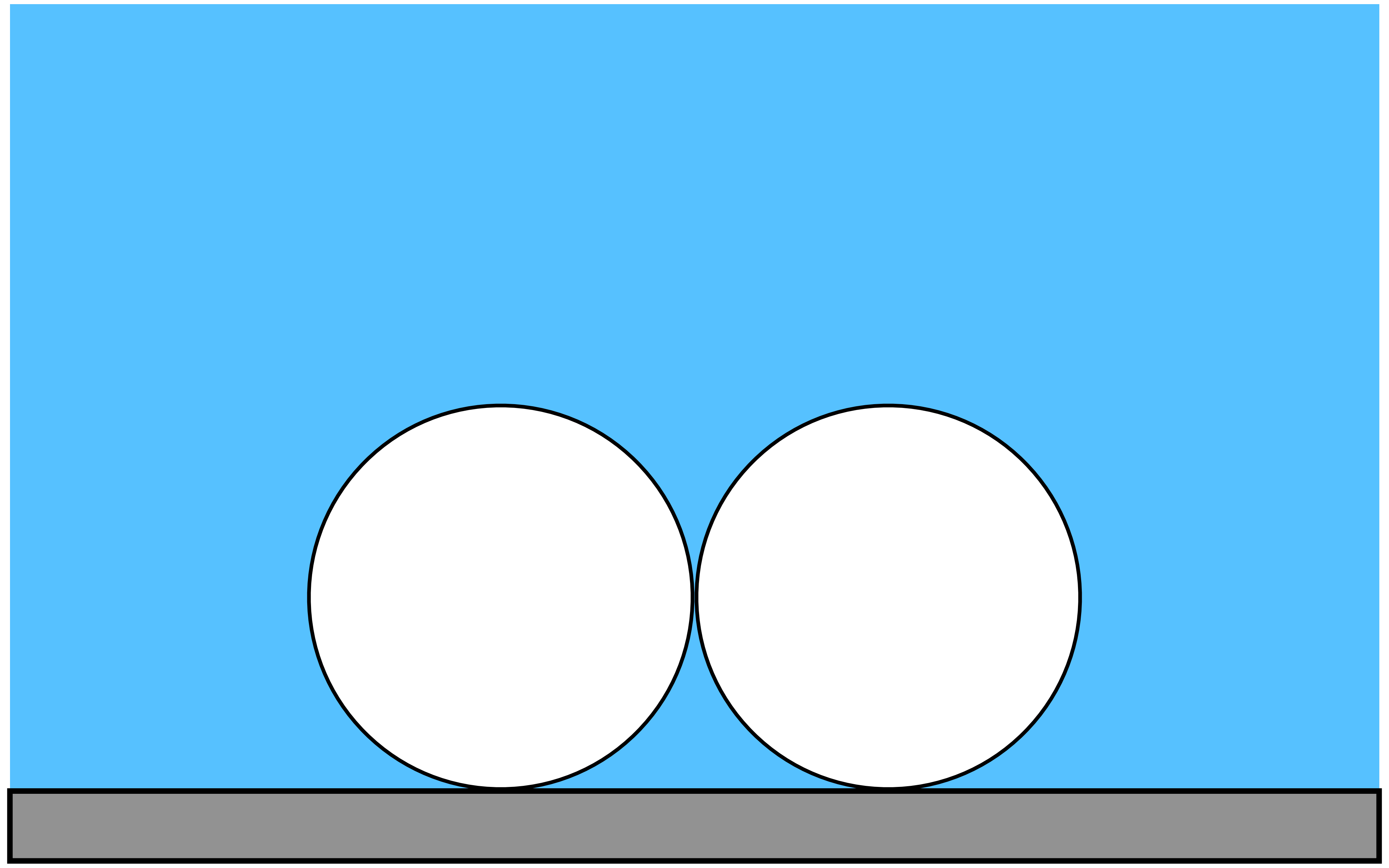
magnitude of spurious flow in terms of Weber number



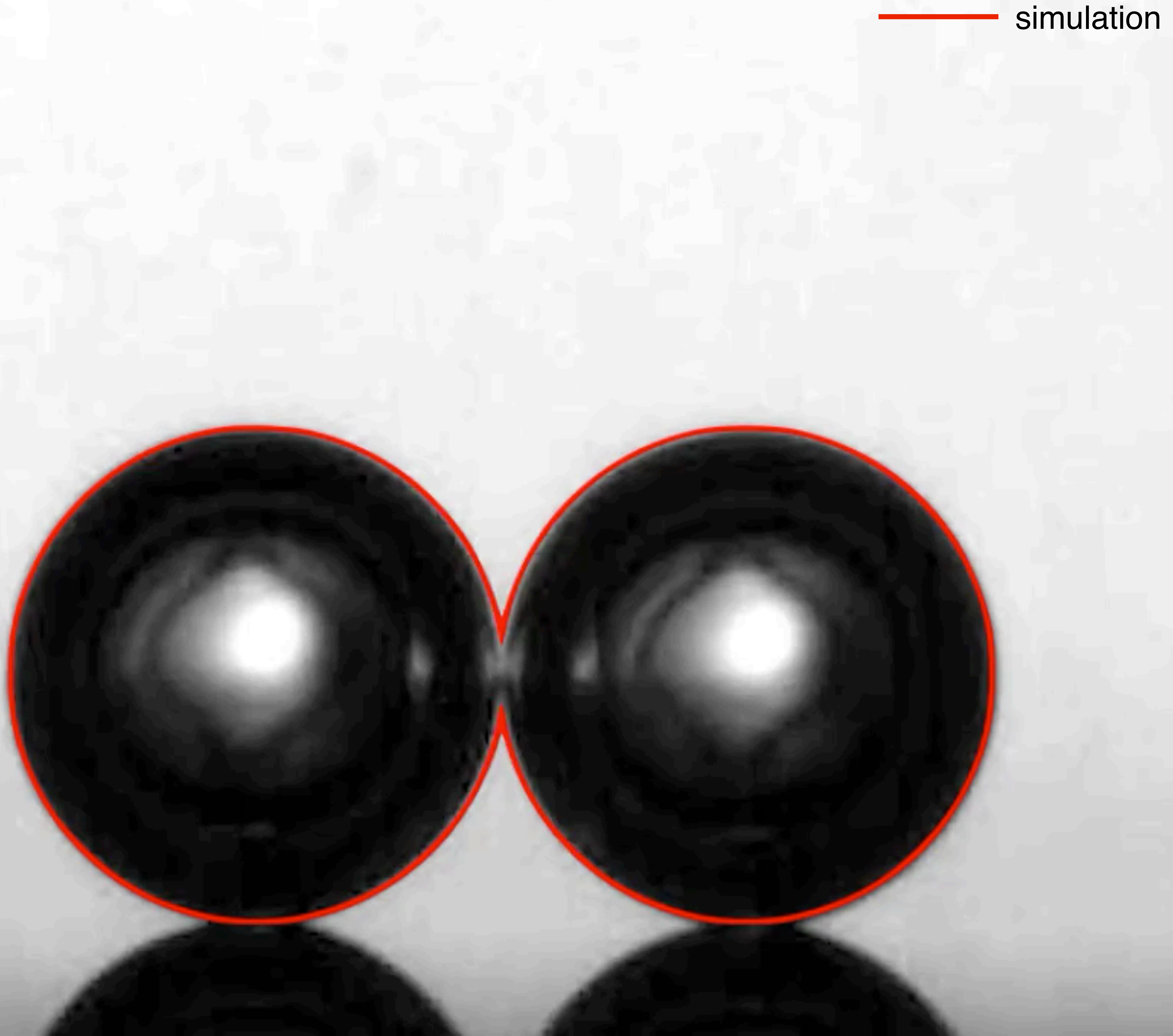
Applications

Coalescence of bubbles

- Experiment on near-wall coalescence
- Bubbles grow due to diffusion of dissolved gas
- Simulations reproduce the experiment

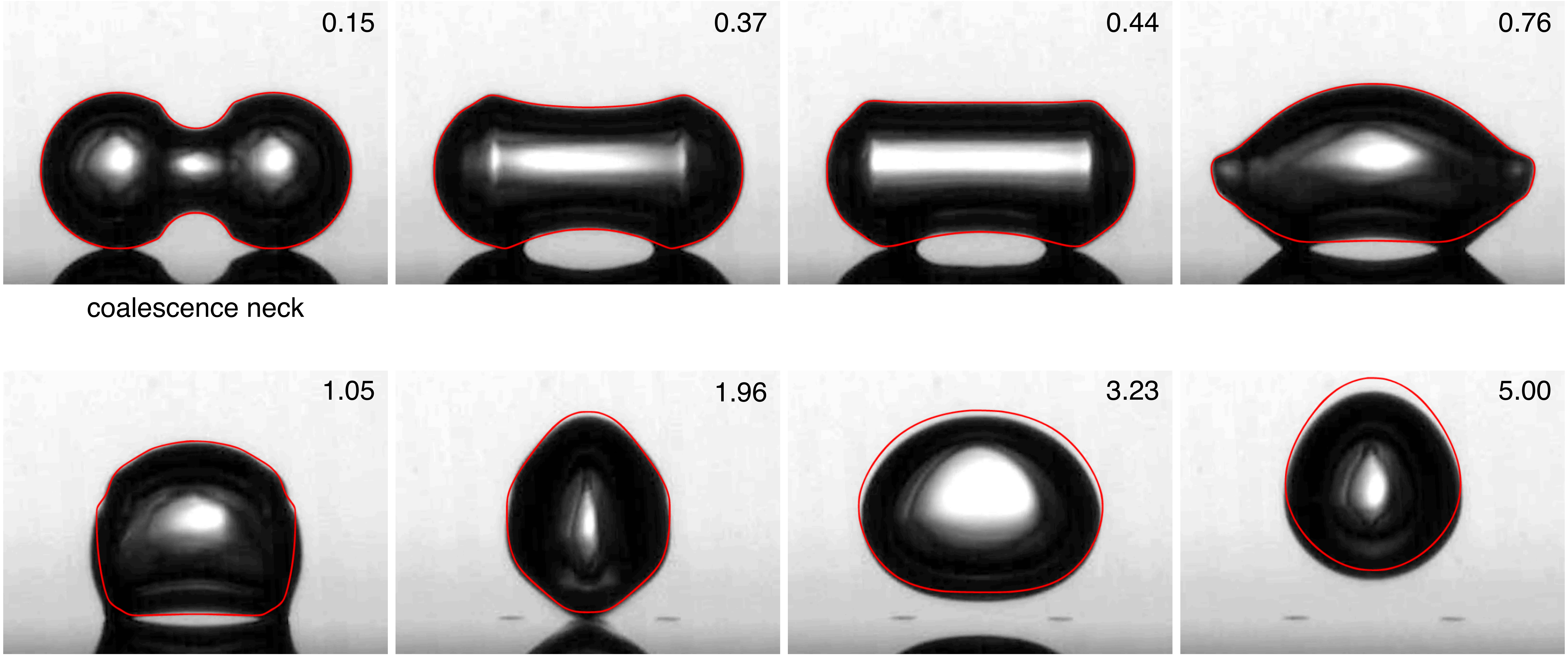


Soto ÁM, Maddalena T, Fraters A, Van Der Meer D, Lohse D.
Coalescence of diffusively growing gas bubbles.
Journal of fluid mechanics. 2018


simulation

Coalescence of bubbles

simulation

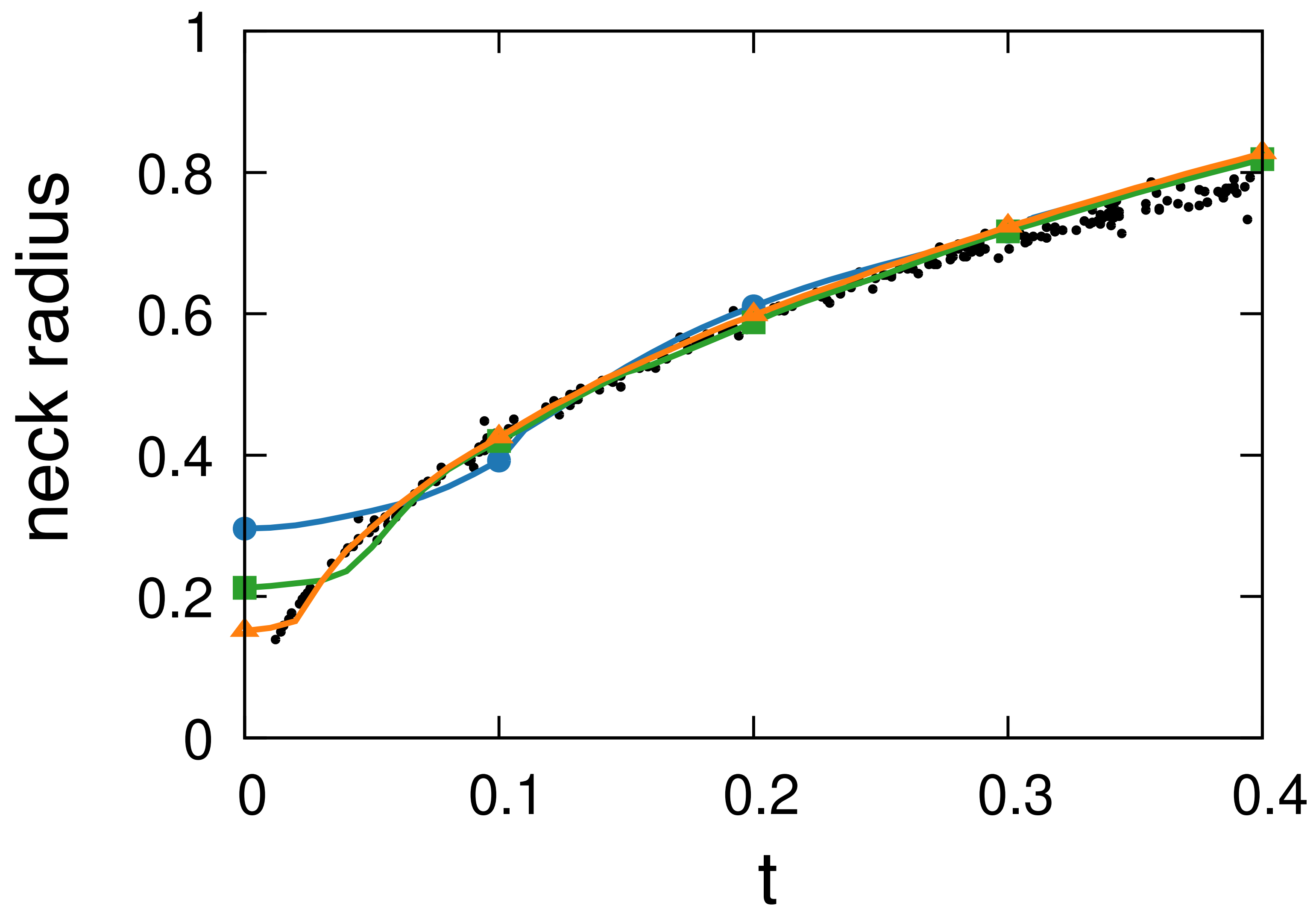


coalescence neck

detachment

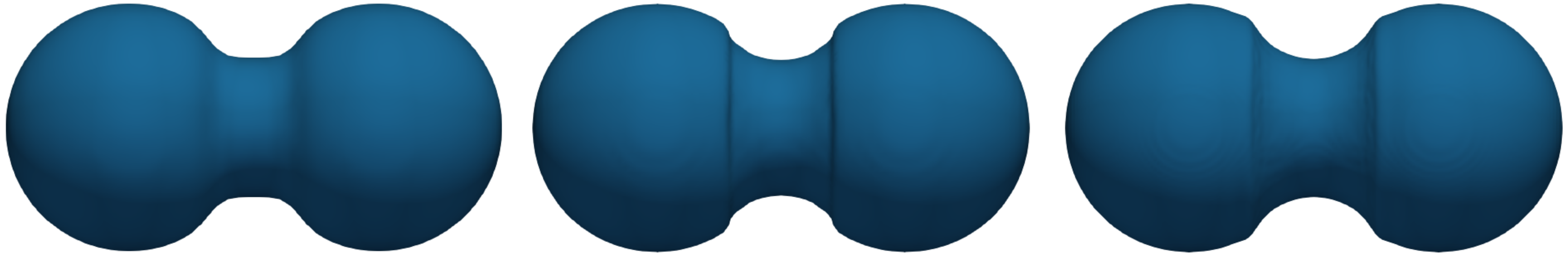
oscillations

Coalescence of bubbles



present

t = 0.17



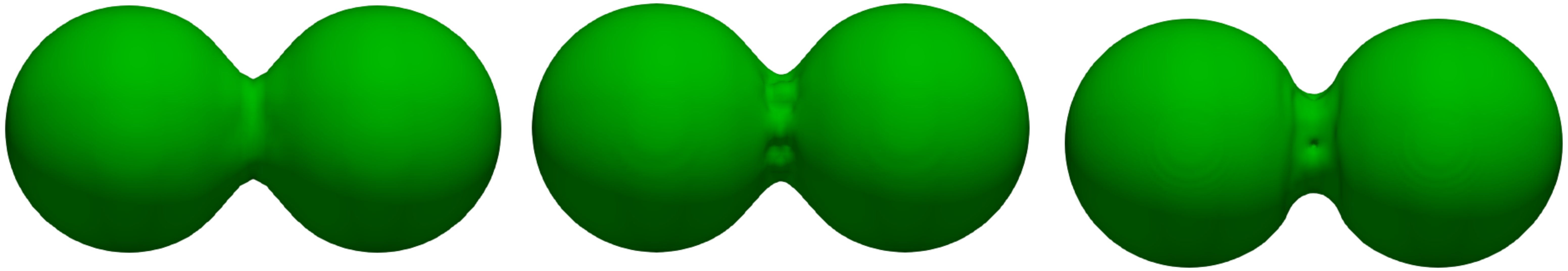
9.6 ●

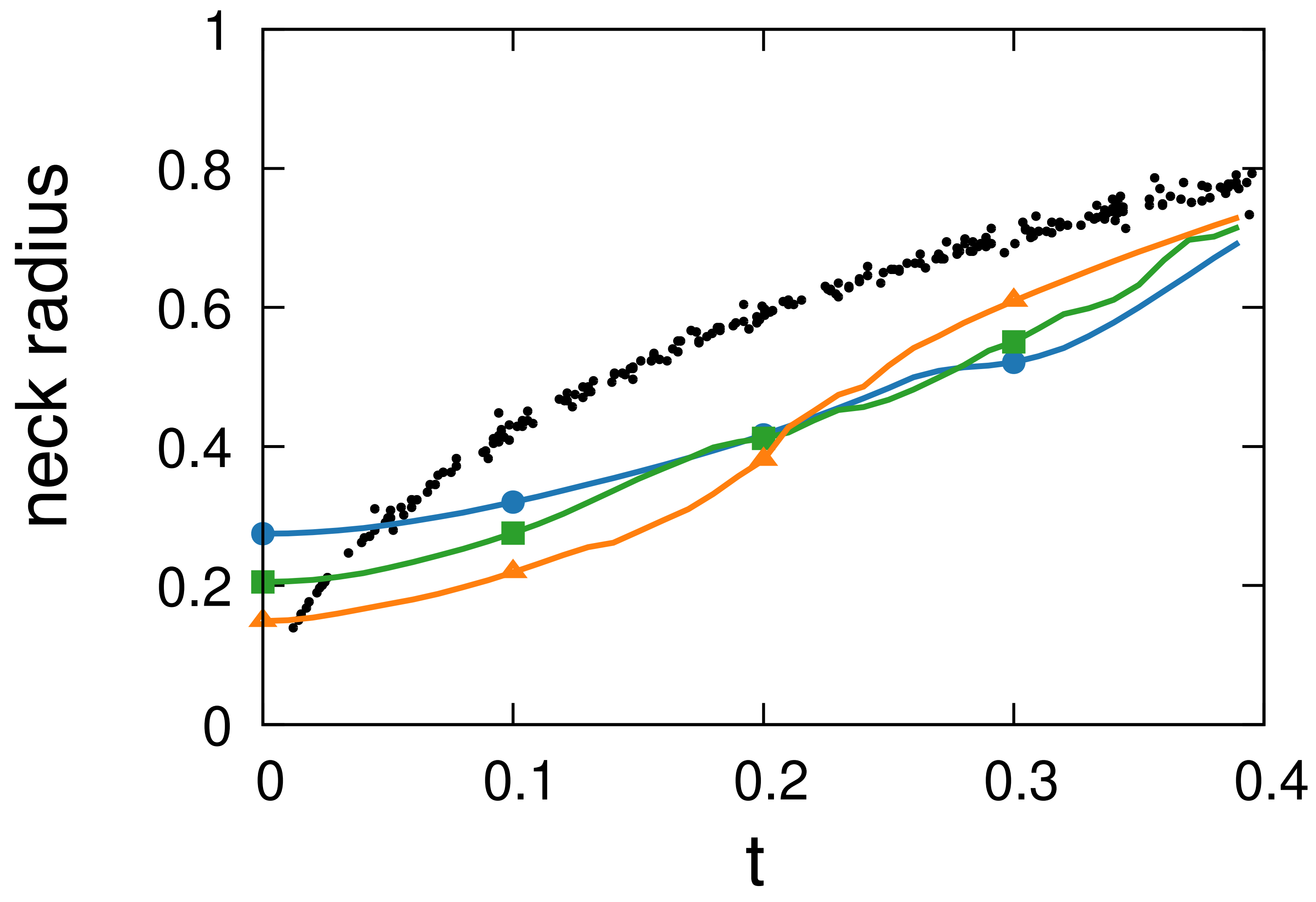
19.2 ■

38.4 ▲

cells per radius →

Basilisk





Taylor-Green vortex with bubbles

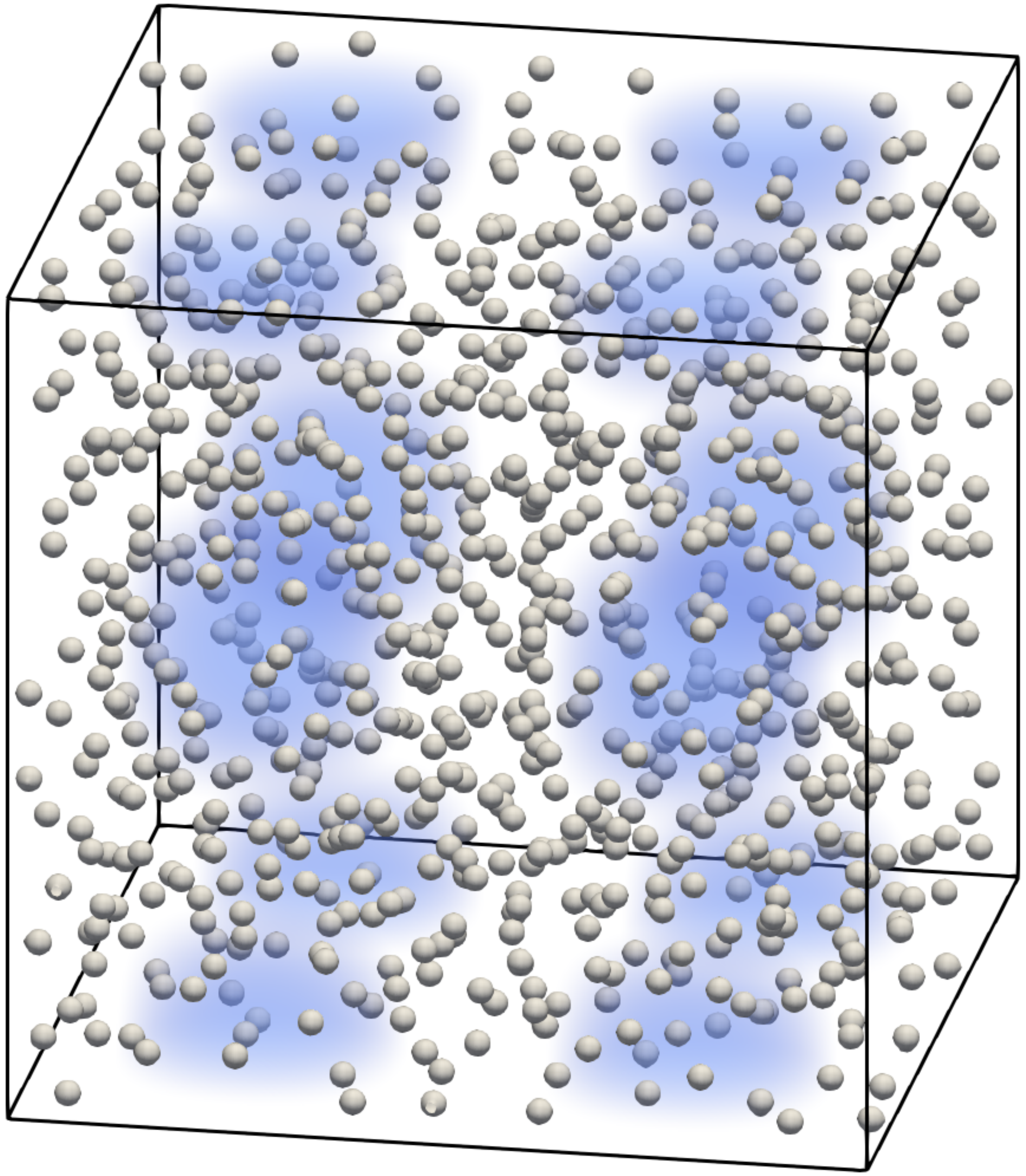
- Periodic domain $[0, 2\pi]^3$
- Initial velocity

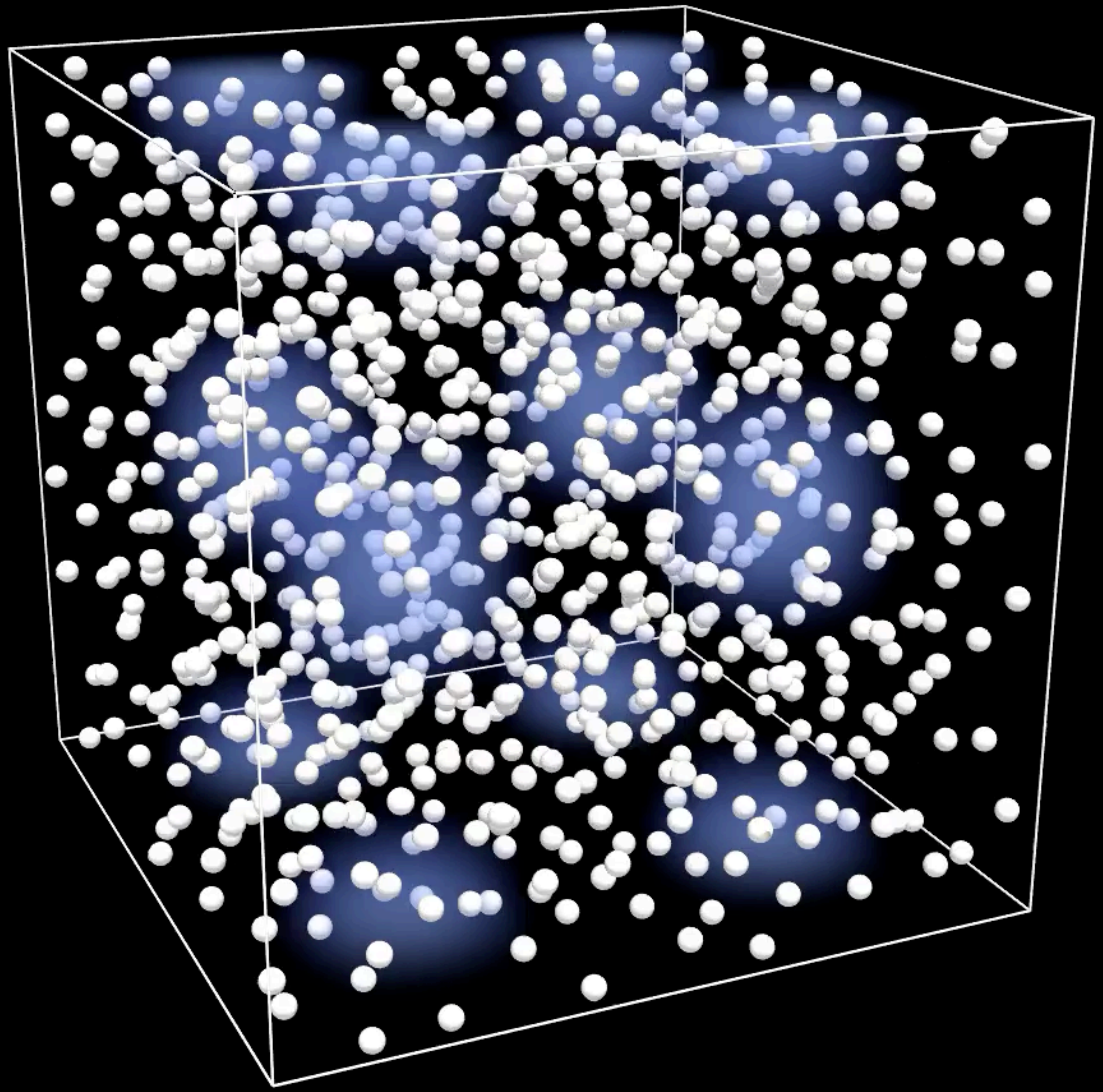
$$u_x = \sin x \ \cos y \ \cos z$$

$$u_y = -\cos x \ \sin y \ \cos z$$

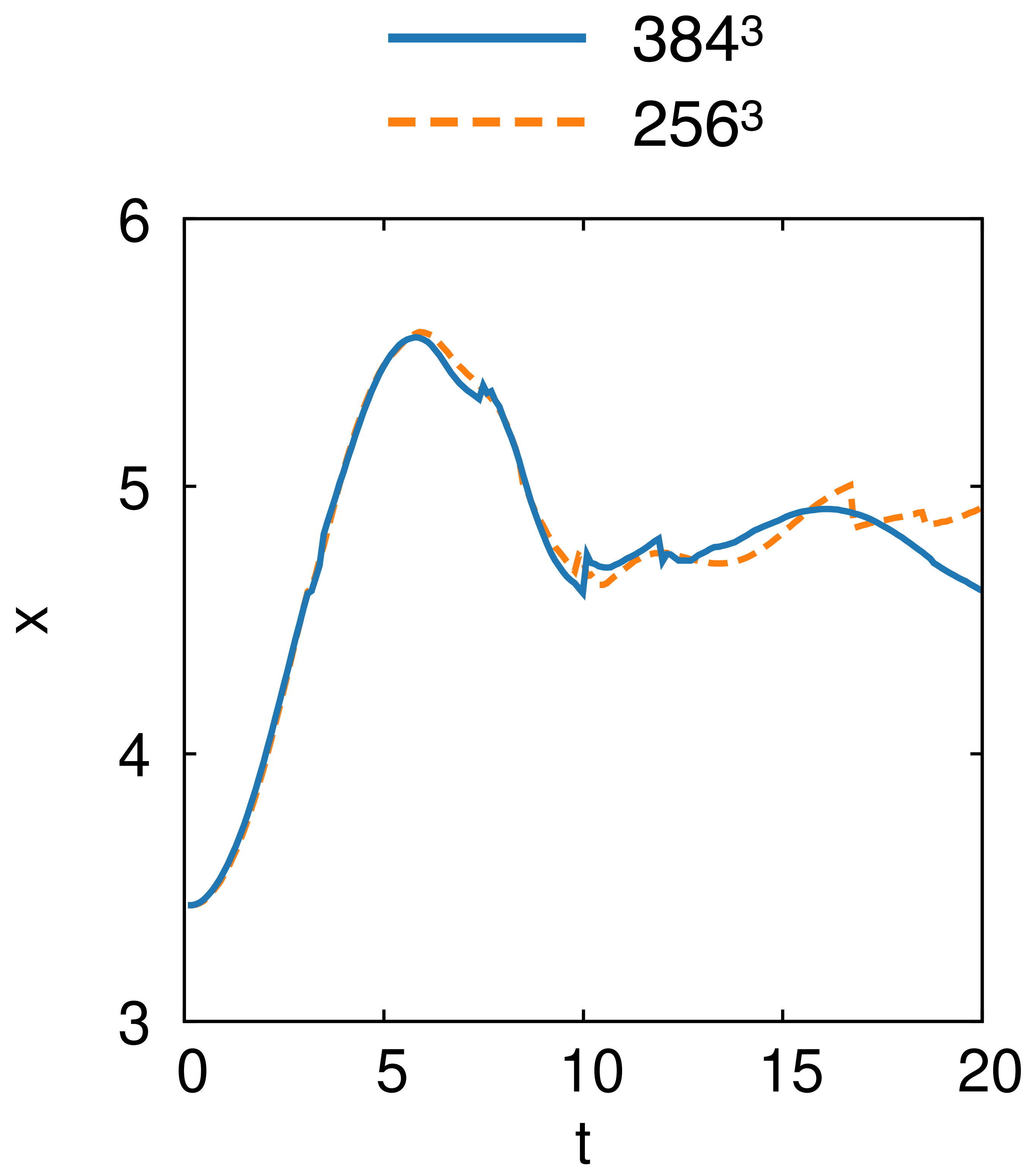
$$u_z = 0$$

- 890 bubbles, volume fraction 1.4%
- $\mathrm{Re} = \dfrac{\rho}{\mu} = 1600 \qquad \mathrm{We} = \dfrac{2\rho R}{\sigma} = 2$
- Mesh 256^3 or 384^3

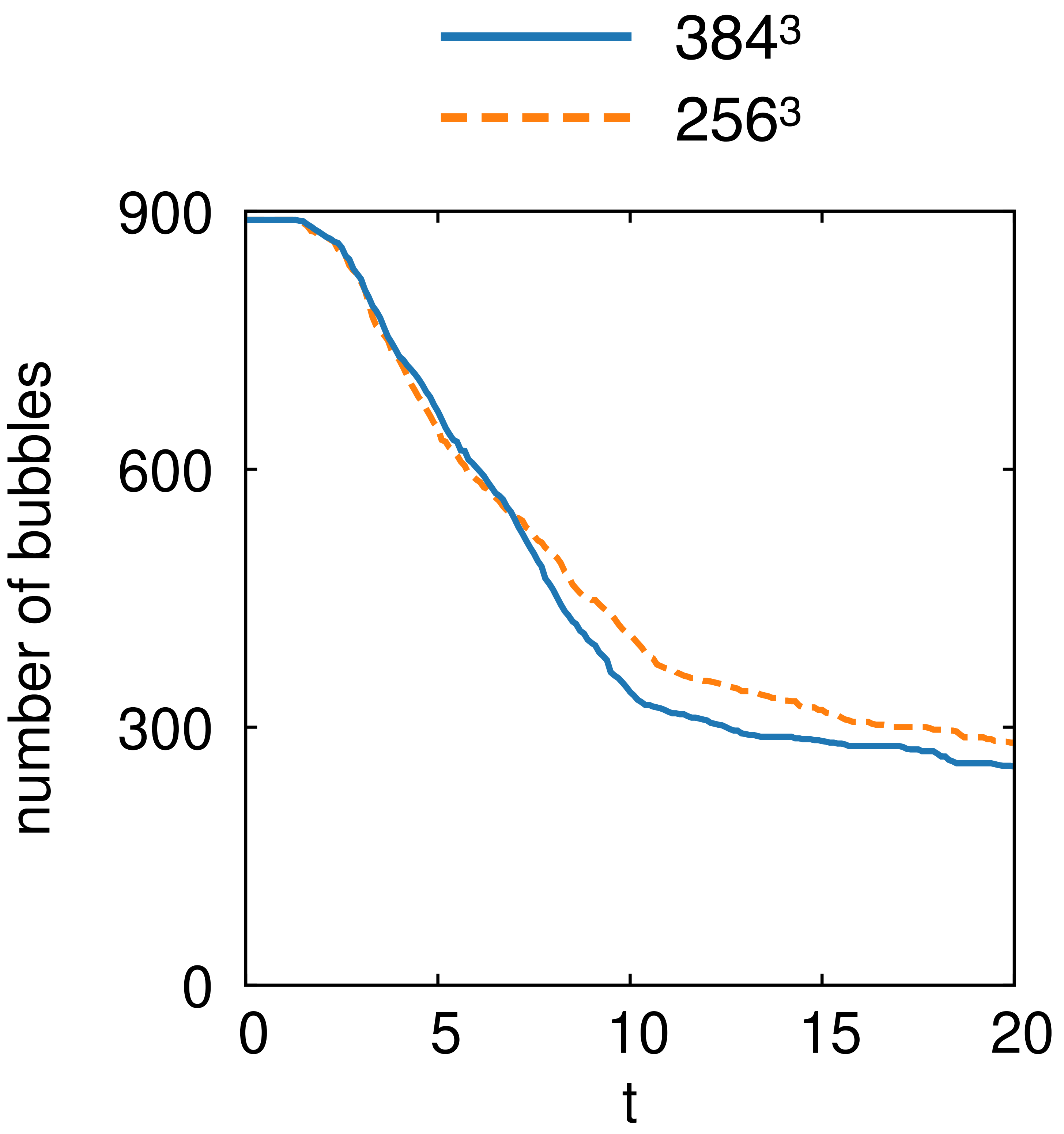




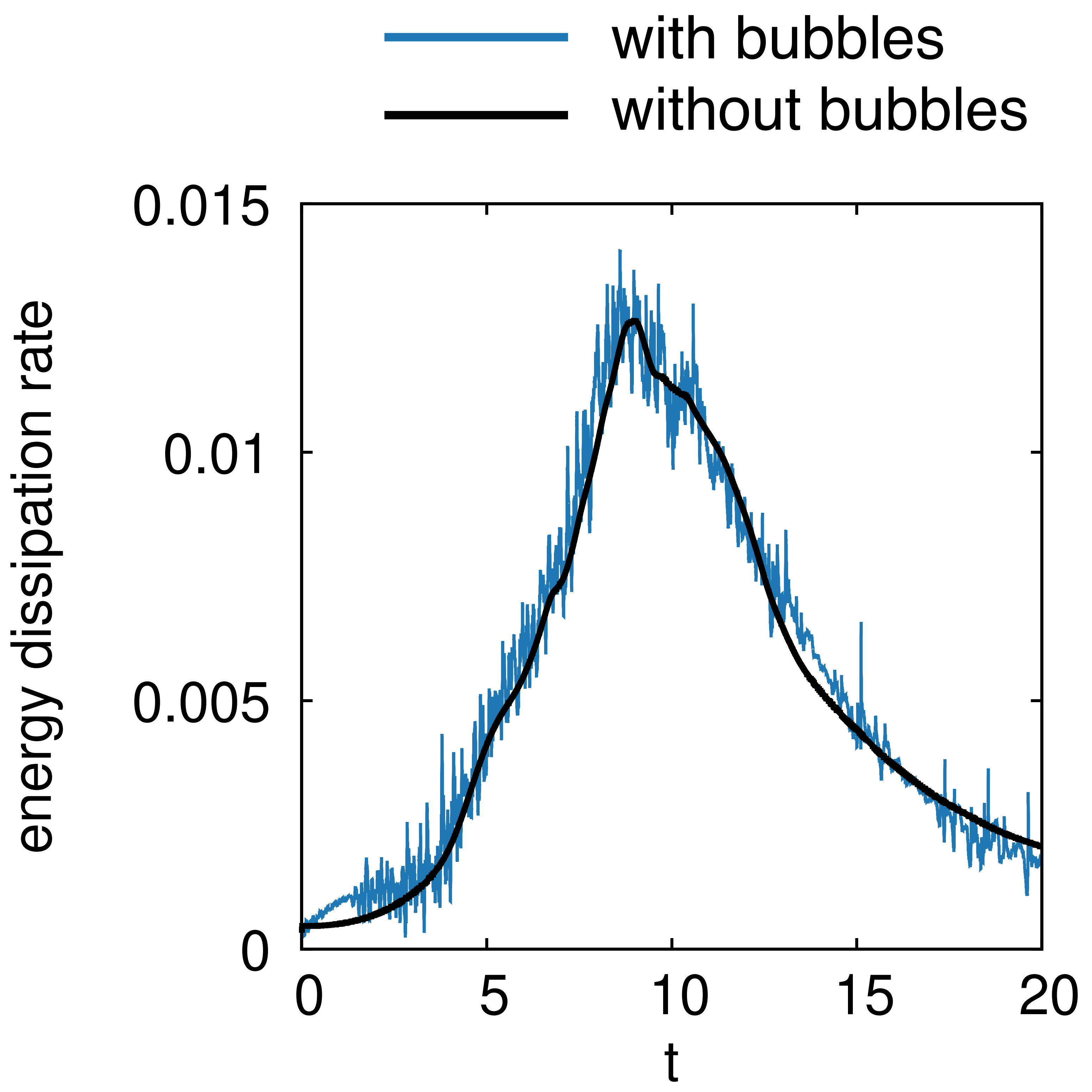
Taylor-Green vortex with bubbles



trajectory of one bubble,
no change on finer mesh



number of bubbles reduces
with time due to coalescence

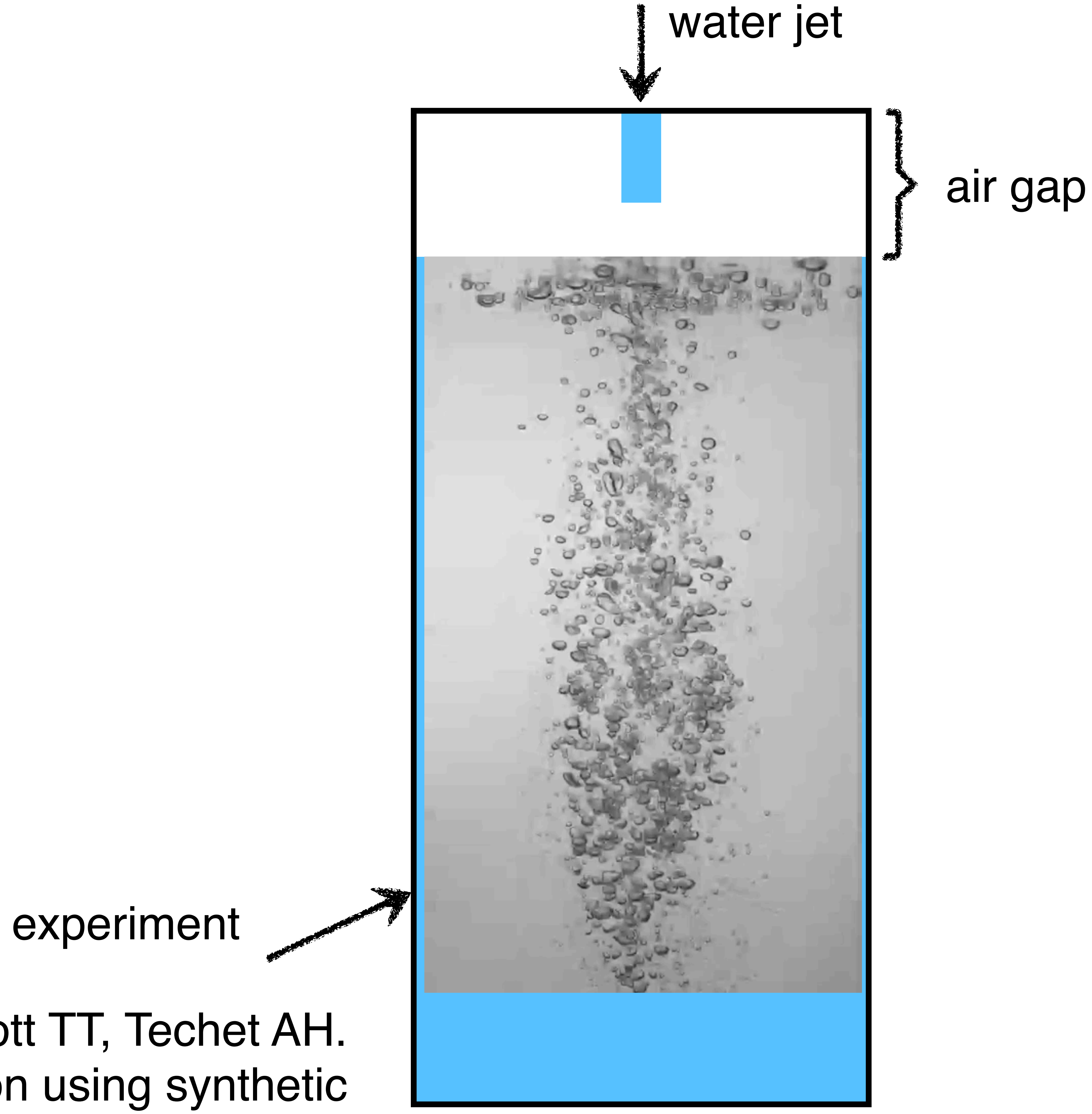


coalescence causes
fluctuations of dissipation rate

Plunging jet with air entrainment

Water jet impacts the free surface

- Box width 10 cm
- Free-slip walls
- Jet diameter 6 mm, velocity 4 m/s
- Mesh 256 x 1024 x 256

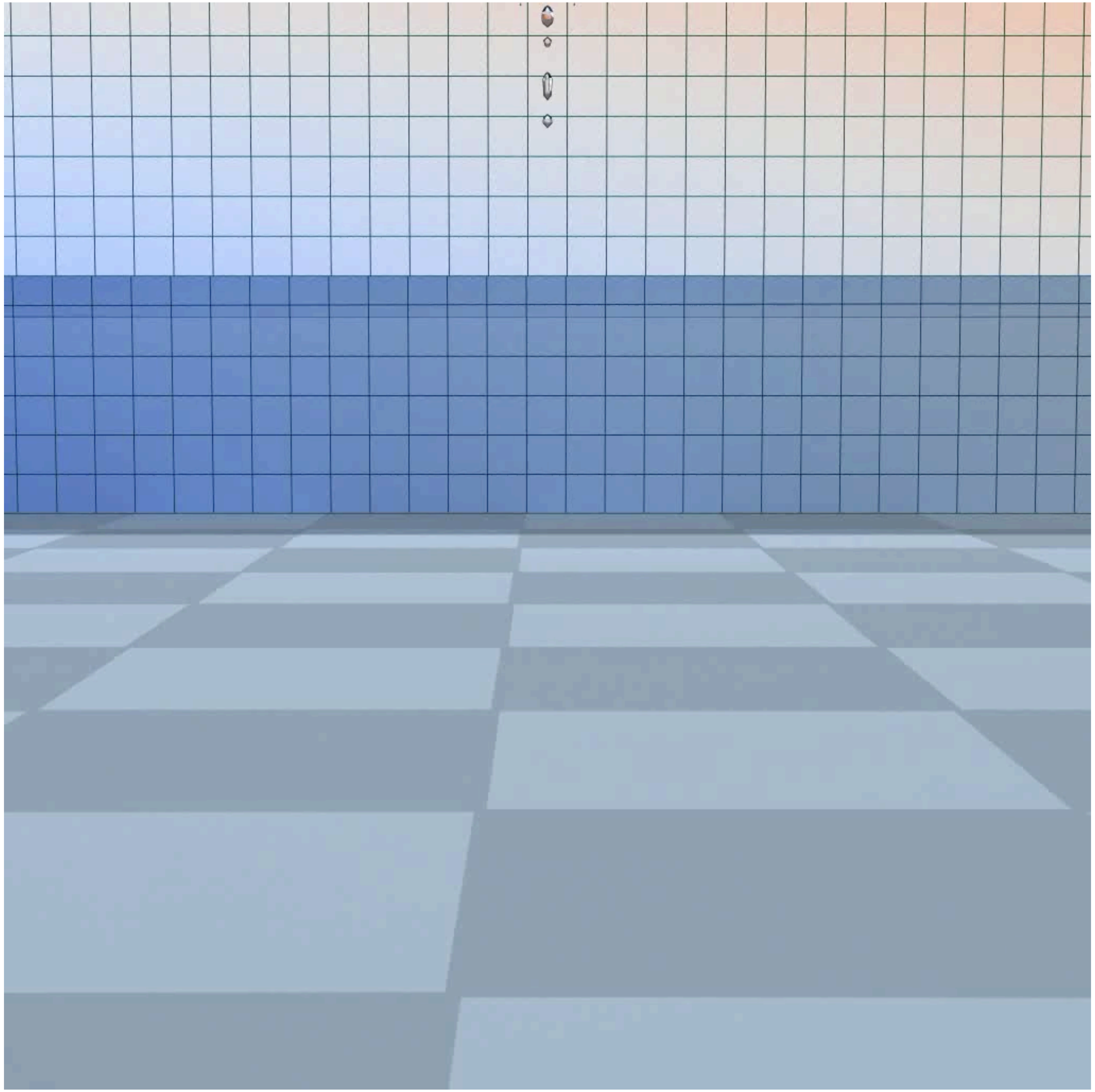


Belden J, Ravela S, Truscott TT, Techet AH.
Three-dimensional bubble field resolution using synthetic
aperture imaging: application to a plunging jet.
Experiments in fluids. 2012

Visualization by Jean M. Favre (CSCS)
with Paraview and NVIDIA VisRTX

1. Liquid jet impact

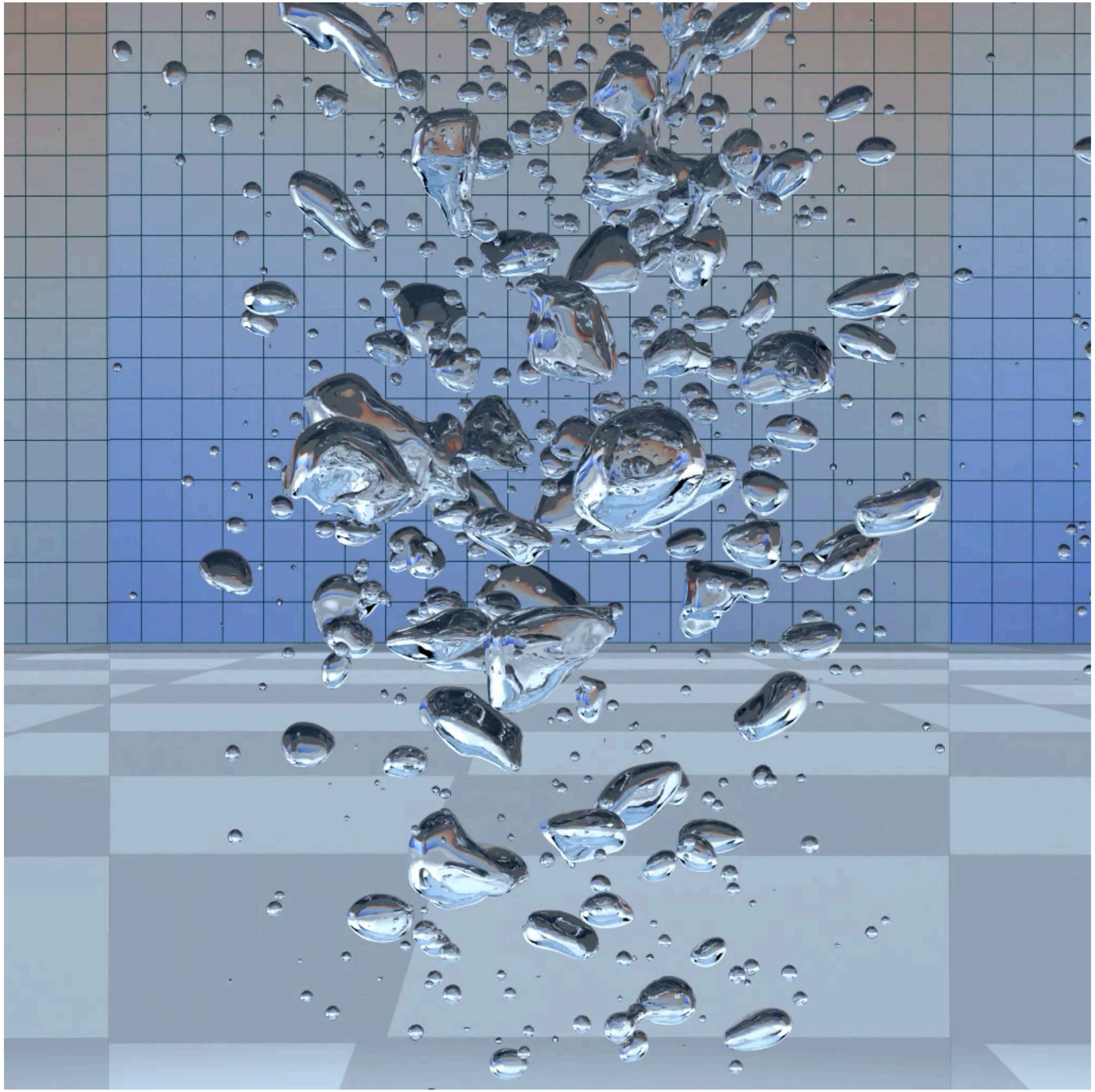
- large air cavity
- then small bubbles



Visualization by Jean M. Favre (CSCS)
with Paraview and NVIDIA VisRTX

2. Stagnation zone

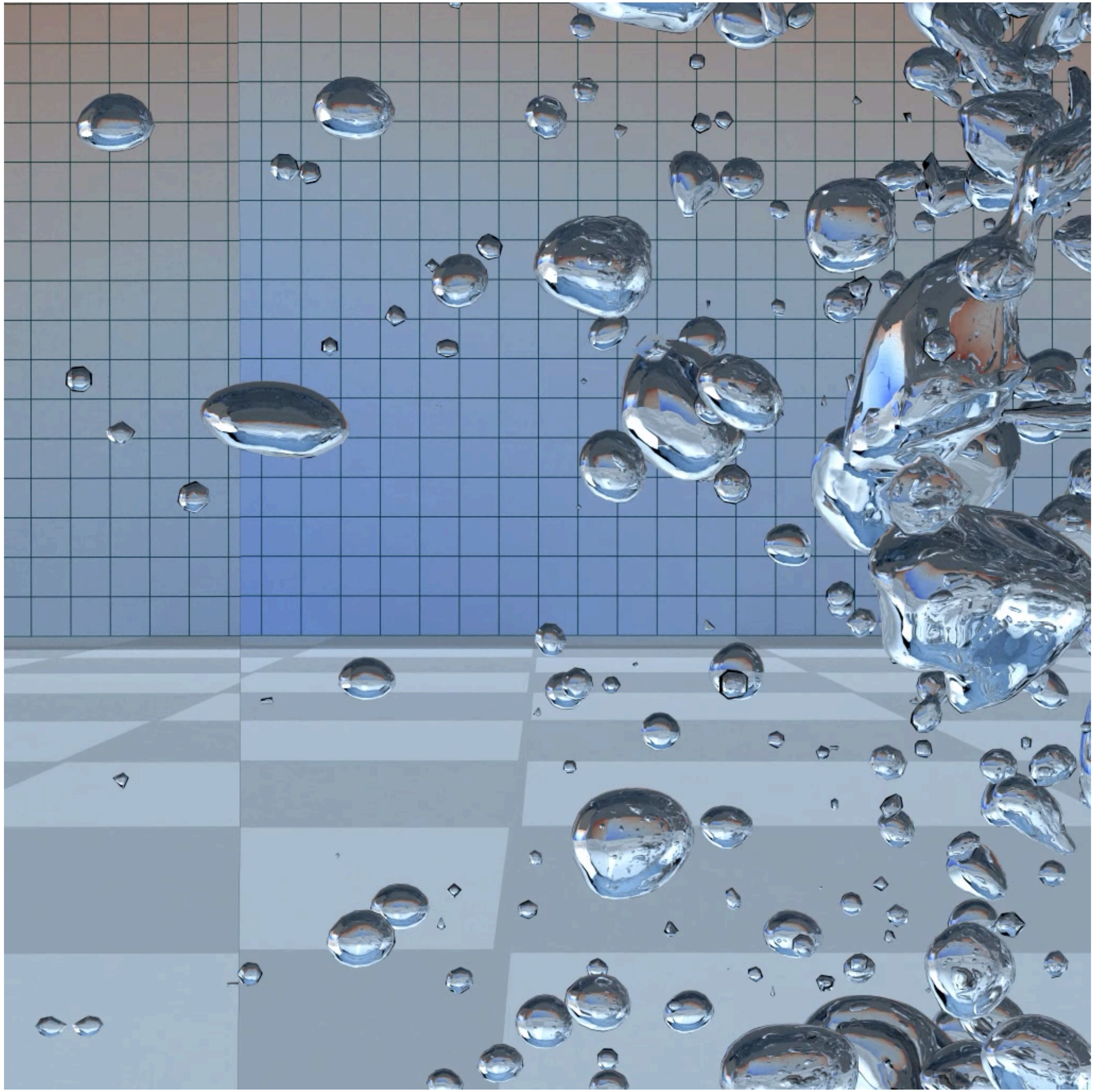
- larger bubbles
  form due to coalescence



Visualization by Jean M. Favre (CSCS)
with Paraview and NVIDIA VisRTX

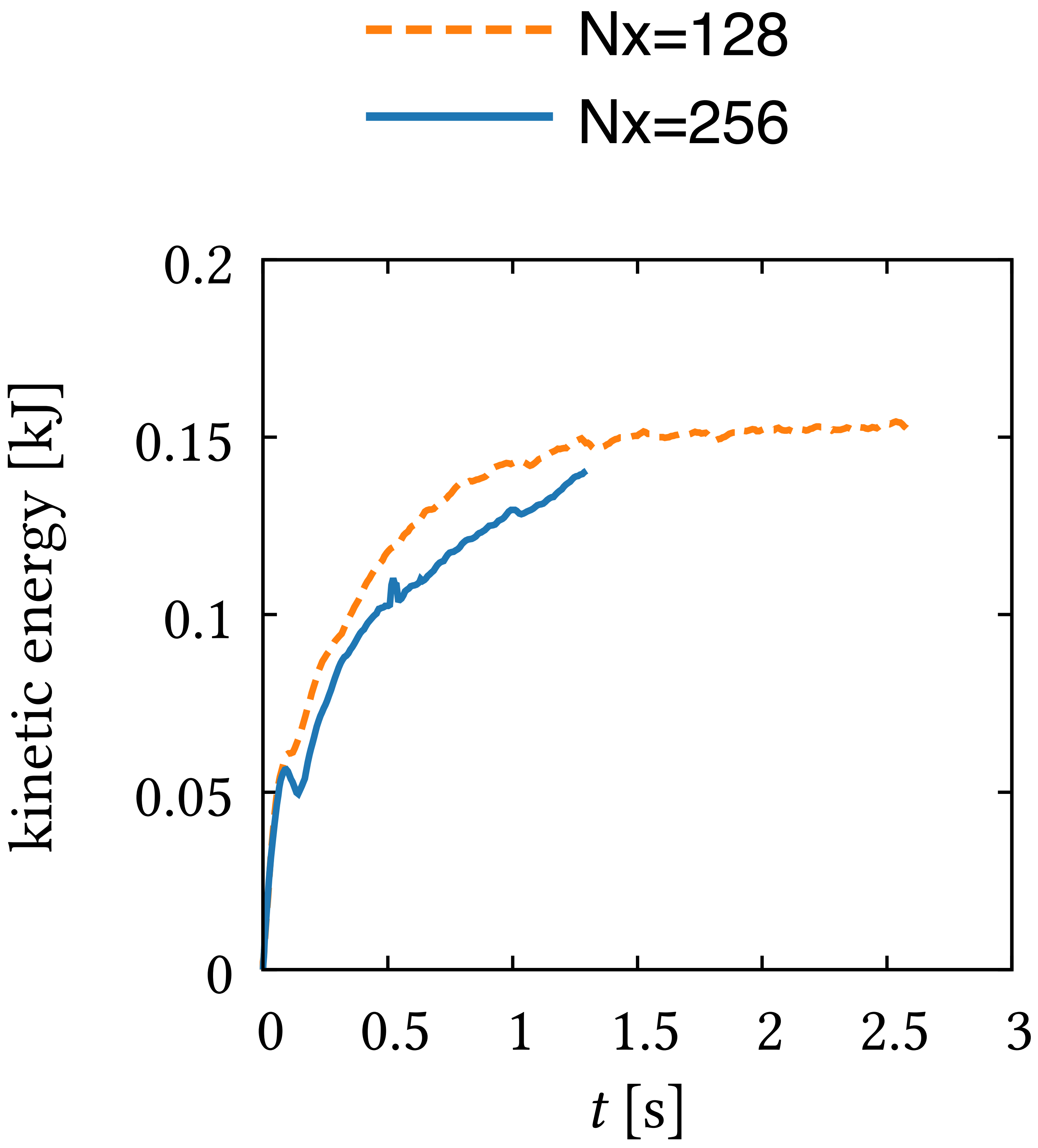
3. Rising bubbles

- larger bubbles are elliptical and follow zigzag trajectory

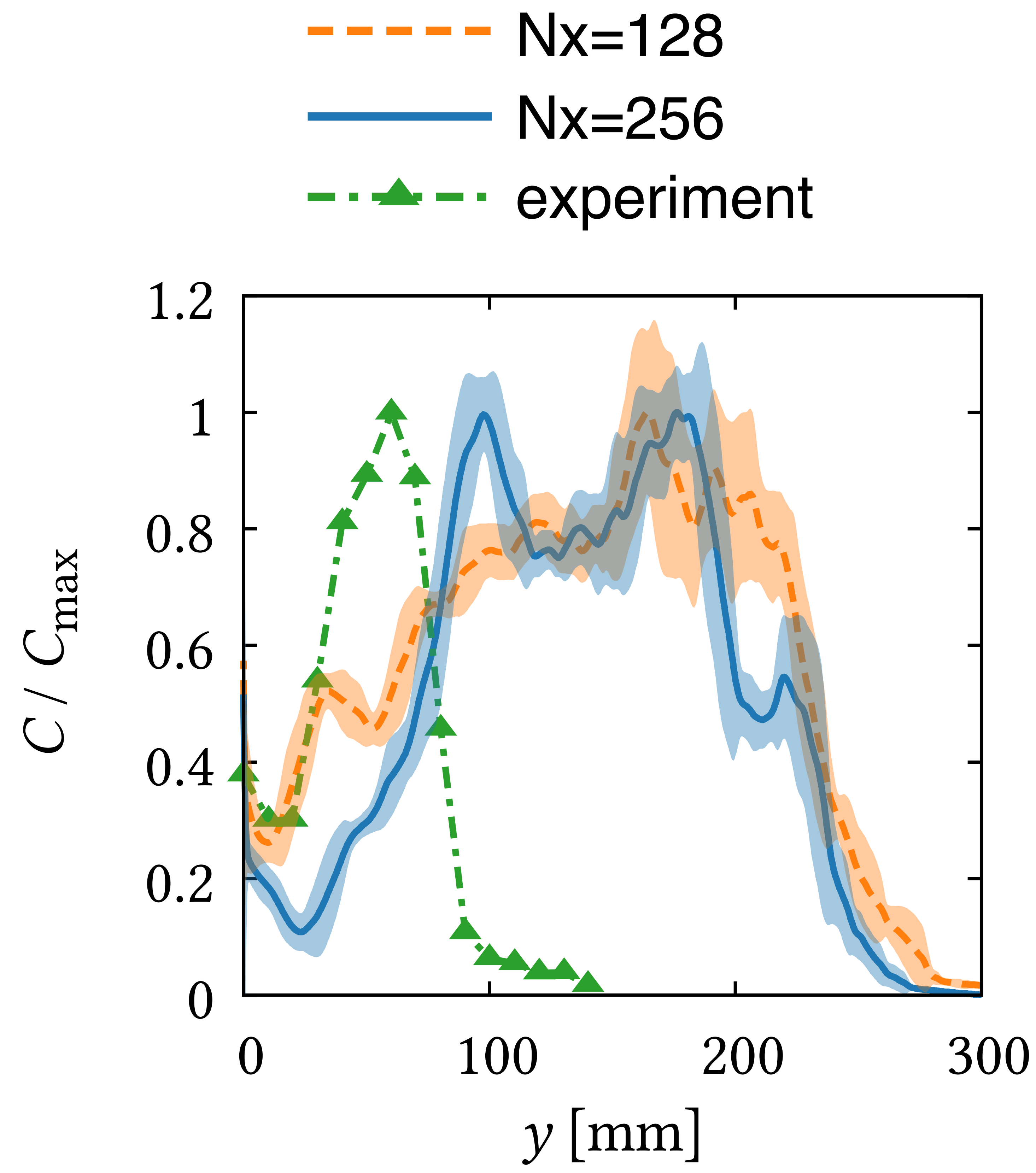


Visualization by Jean M. Favre (CSCS)
with Paraview and NVIDIA VisRTX

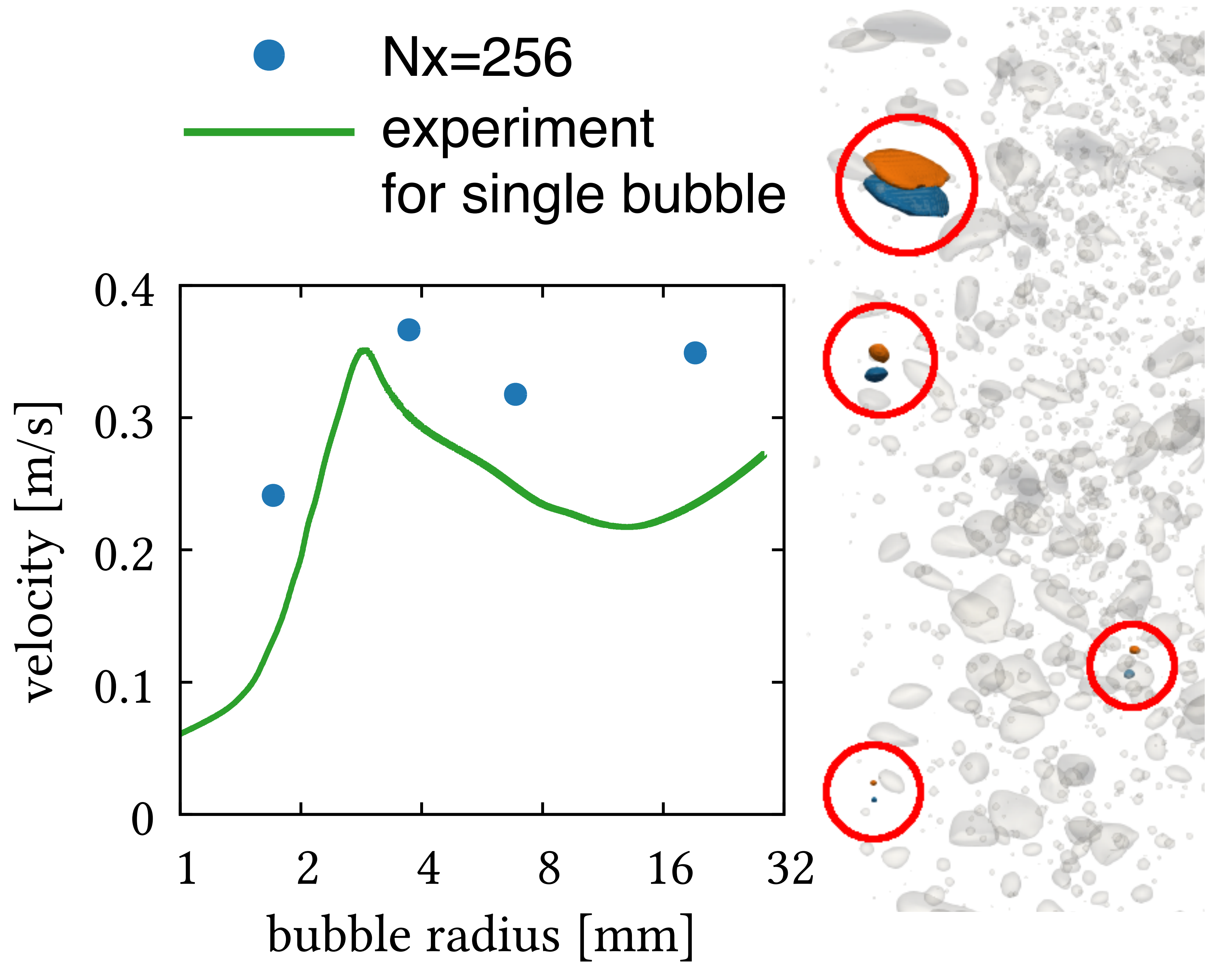
Plunging jet with air entrainment



equilibration of
mixture kinetic energy



axial concentration of gas,
penetration depth overpredicted



velocity of selected bubbles
compared to rise velocity
of single bubble [Maxworthy 1996]

Summary

- Particles for curvature estimation improve the accuracy at low resolution [arXiv:1906.00314]
- Bubbles of various scales are resolved on a uniform mesh
- Coroutines enable modularity with blockwise processing

Outlook:

- performance
  kernels on GPU, compute-transfer overlap
- more applications
  turbulent multiphase flows
- open-source release

`tinyurl.com/demogrid`