FEM-Level Set Techniques for Multiphase Flow

Some recent results

ENUMATH09, Uppsala

Stefan Turek, Otto Mierka, Dmitri Kuzmin, Shuren Hysing

Institut für Angewandte Mathematik, TU Dortmund
http://www.mathematik.tu-dortmund.de/LS3
http://www.featflow.de

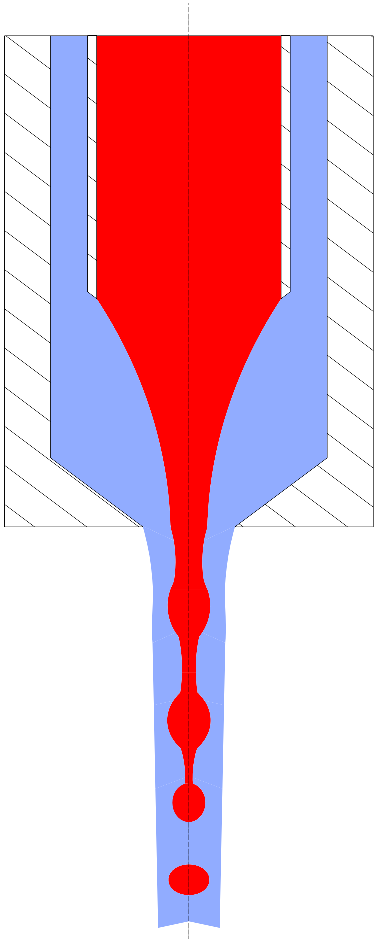
- ***Accurate, robust, flexible and efficient simulation of multiphase problems, particularly in 3D, is still a challenge***

- **Specific Application:** Monodisperse laminar droplets

  Problem: Results are due to many parameters, i.e., density (ratio), viscosity (ratio), rheological behaviour, surface tension, flow conditions,…

→ Efficient CFD techniques (**implicit FEM-Multigrid-Level Set** solver in FEATFLOW, dynamically adapted meshes, parallel/HPC techniques)

→ Benchmarking/**Validation** via experiments



Governing Equations

The incompressible Navier Stokes equation

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) - \nabla\cdot\left(\mu\left[\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}}\right]\right) + \nabla p = \mathbf{f}_{\mathrm{ST}} + \rho\mathbf{g}$$

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

unknown interphase location

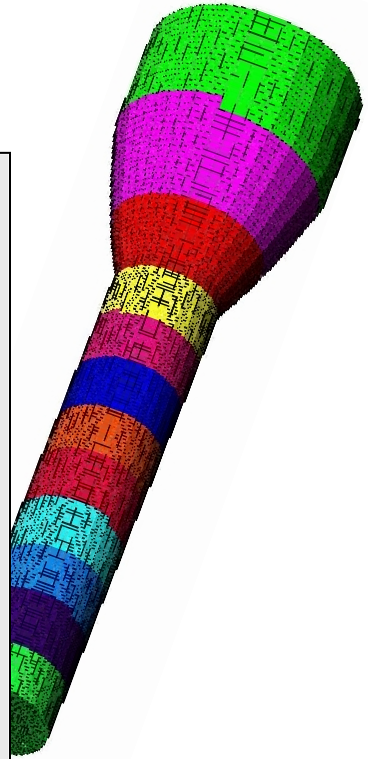
Interphase tension force

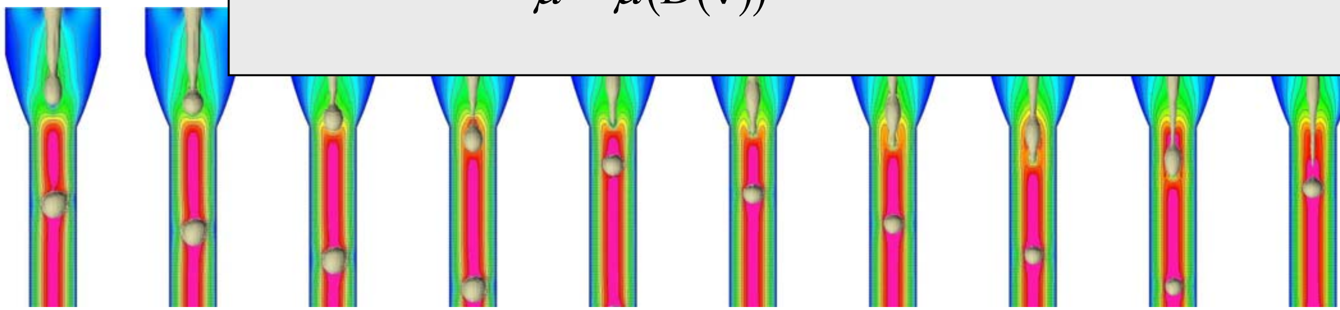
$$-\nabla\cdot\mathbf{n} \quad \text{on} \quad \Gamma$$

$$\mu = \mu(D(\mathbf{v}),\Gamma), \quad \rho = \rho(\Gamma)$$

Dependency of physical quantities

$$\mu = \mu(\mathrm{D}(\mathbf{v}))$$





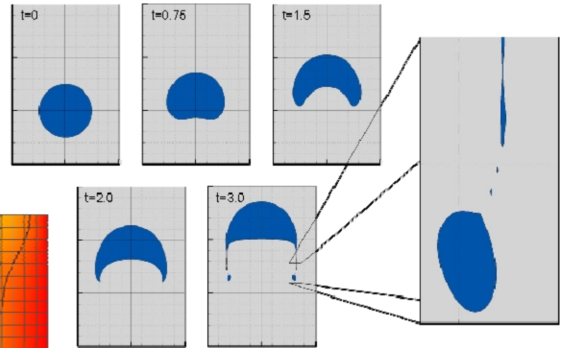
Discretization:

- a) Navier-Stokes: FEM Q_2/P_1
- b) Levelset: DG-FEM P_1
- Crank-Nicholson scheme in time

in space

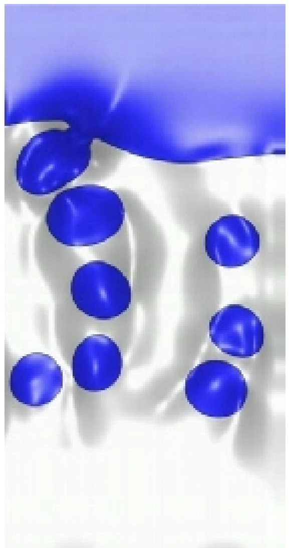
+ stabilization (TVD,EO-FEM)

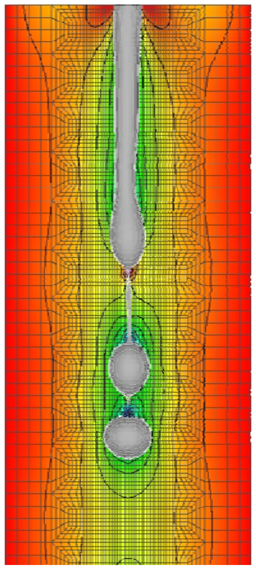
no stabilization !

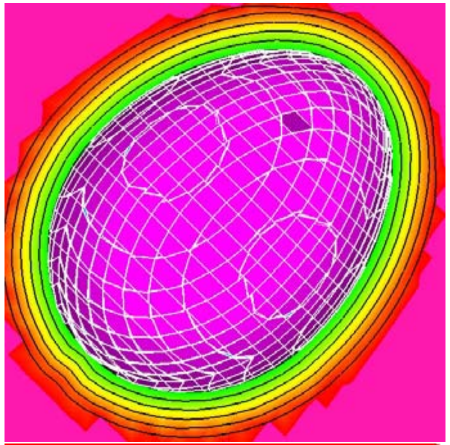


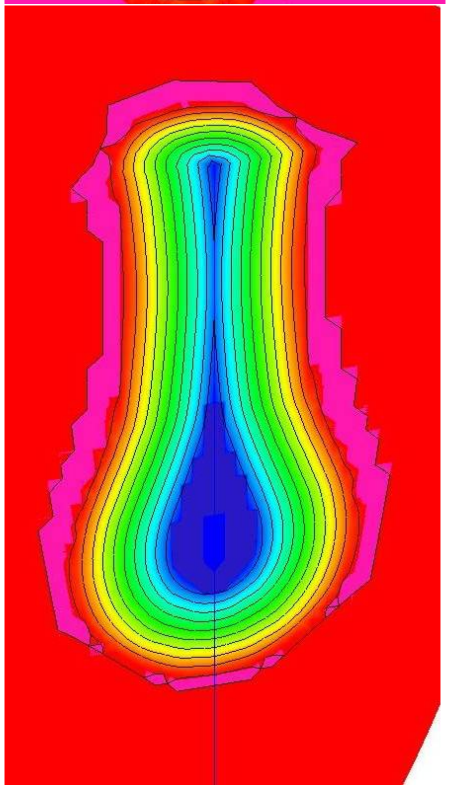
Main features of the FeatFlow approach:

- Parallelization based on domain decomposition
- High order discretization schemes
- Use of unstructured meshes
- Multigrid linear solver
- FCT & EO stabilization techniques
- Adaptive grid deformation
- UCHPC









Levelset method (→“smooth“ distance function)

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0$$

<u>Benefits:</u>

- Provides an accurate representation of the interphase
- Provides other auxiliary quantities (normal, curvature)
- Allows topology changes
- Treatment of viscosity, density and surface tension without explicit representation of the interphase
- Adaptive grid advantageous, but not necessary

<u>Problems:</u>

- It is not conservative → mass loss
- Needs to be reinitialized to maintain its distance property
- Higher order discretization?

Problems and Challenges

- Steep gradients of the velocity field and of physical properties near the interphase (oscillations!)

- Reinitialization (smoothed sign function, artificial movement of the interphase (→ mass loss), how often to perform?)

- Mass conservation (during levelset advection and reinitialization)

- Representation of interphacial tension: CSF, Line Integral, Laplace-Beltrami, Phasefield, *etc.*, explicit or implicit treatment?

Steep changes of physical quantities:

1) Elementwise averaging of the physical properties (prevents oscillations):

   $\rho_e = x\rho_1 + (1-x)\rho_2, \quad \mu_e = x\mu_1 + (1-x)\mu_2$ $\quad$ x is the volume fraction

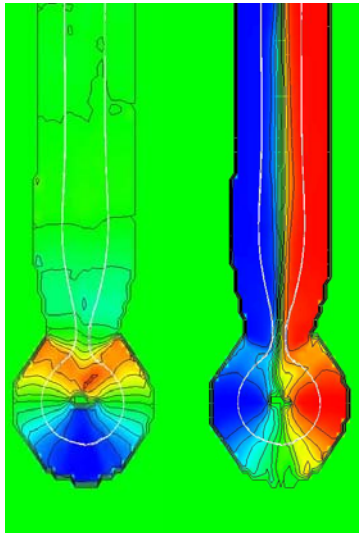
2) Flow part: Extension of nonlinear stabilization schemes (AFC, TVD) for the momentum equation for LBB stable element pairs.

3) Interphase tracking part with DG-FEM: Flux limiters satisfying LED requirements.

Reinitialisation

<u>Alternatives</u>

- Brute force (introducing new points at the zero level surfaces)
- Fast sweeping (applying „advancing front“ upwind type formulas)
- Fast marching
- Algebraic Newton method
- **Hyperbolic PDE approach**
- many more…..



Globally defined normal vectors

Maintaining the signed distance function by PDE reinitialization

$$\frac{\partial \phi}{\partial \tau} + \mathbf{u} \cdot \nabla \phi = S(\phi) \qquad \mathbf{u} = S(\phi)\frac{\nabla \phi}{|\nabla \phi|} \quad \Leftrightarrow \quad |\nabla \phi| = 1$$

Problems:

- What to do with the sign function at the interphase? (smoothing?)
- How often to perform? (expensive→steady state)

Our reinitialization is performed in combination of 2 ingredients:

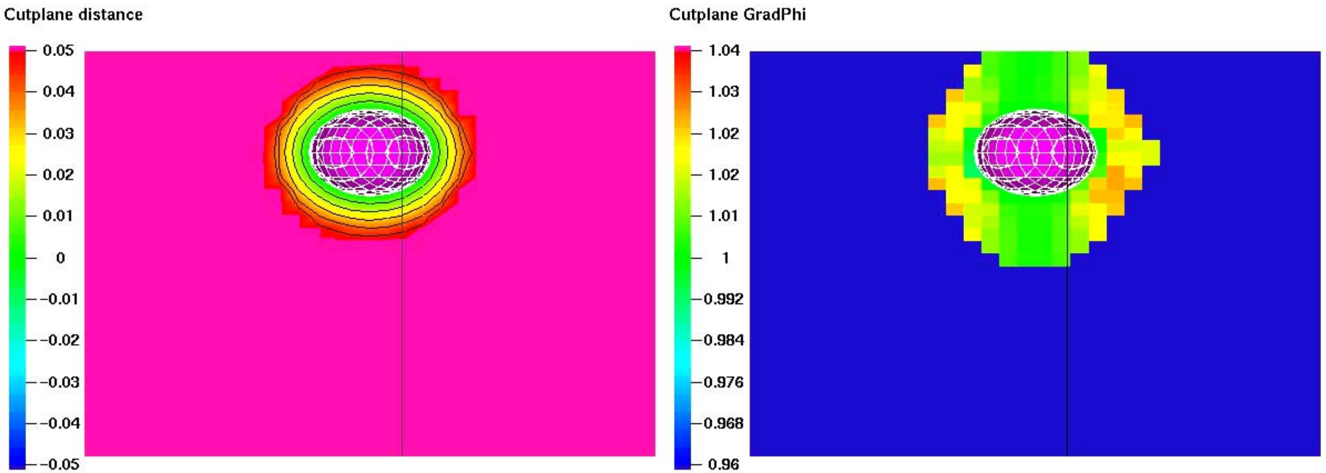
1) Elements intersected by the interphase are modified w.r.t. the slope of the distance distribution („Parolini trick“) such that

$$\left|\nabla \phi\right| = 1$$

2) Far field reinitialization: realization is based on the PDE approach („FBM“), but it does not require smoothening of the distance function!

In addition: continuous projection of the interphase (smoothening of the discontinuous P_1 based distance function)

$$\phi_{P_1} \xrightarrow{L_2 \text{ projection}} \phi_{Q_1} \xrightarrow{L_2 \text{ projection}} \phi_{P_1}$$



Mass conservation

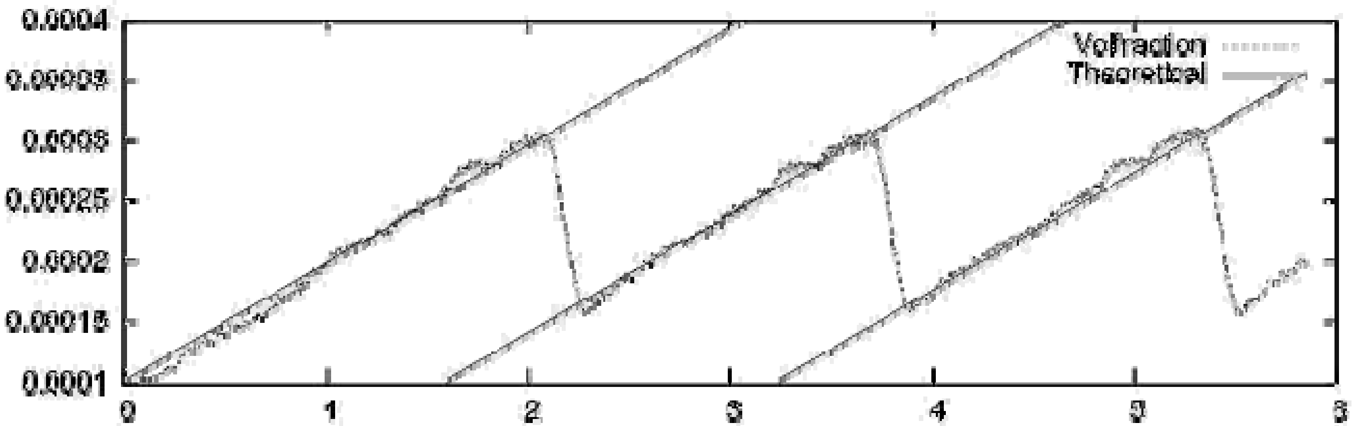
Must be satisfied on the continuous and discrete level as well!

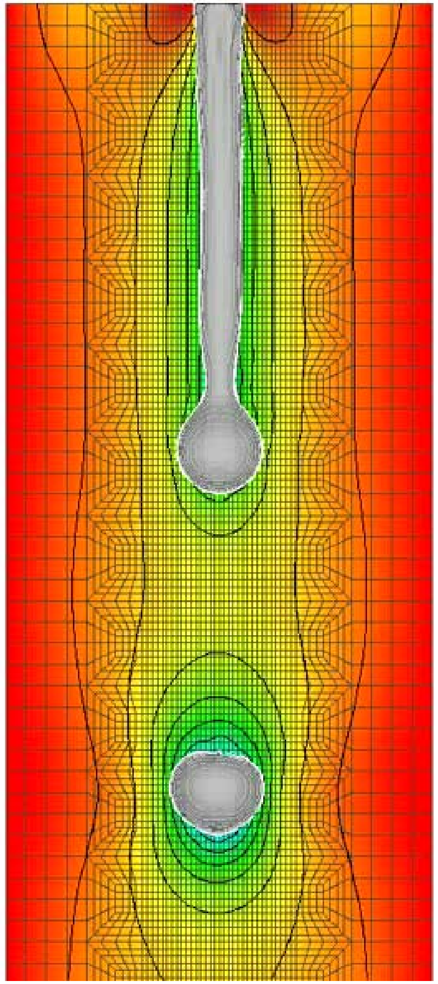
(In work) Replacement of:

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0, \quad \text{by} \quad \frac{\partial \rho(\phi)}{\partial t} + \nabla \cdot \left(\rho(\phi) \mathbf{v} \right) = 0$$

Nonlinear PDE!

Remark: No stabilization! (to avoid numerical diffusion)
→ distance function should be smooth anyway



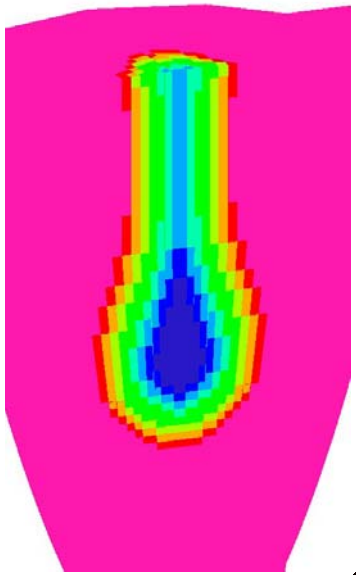


Good agreement with experimental measurements
(in terms of droplet size, frequency)

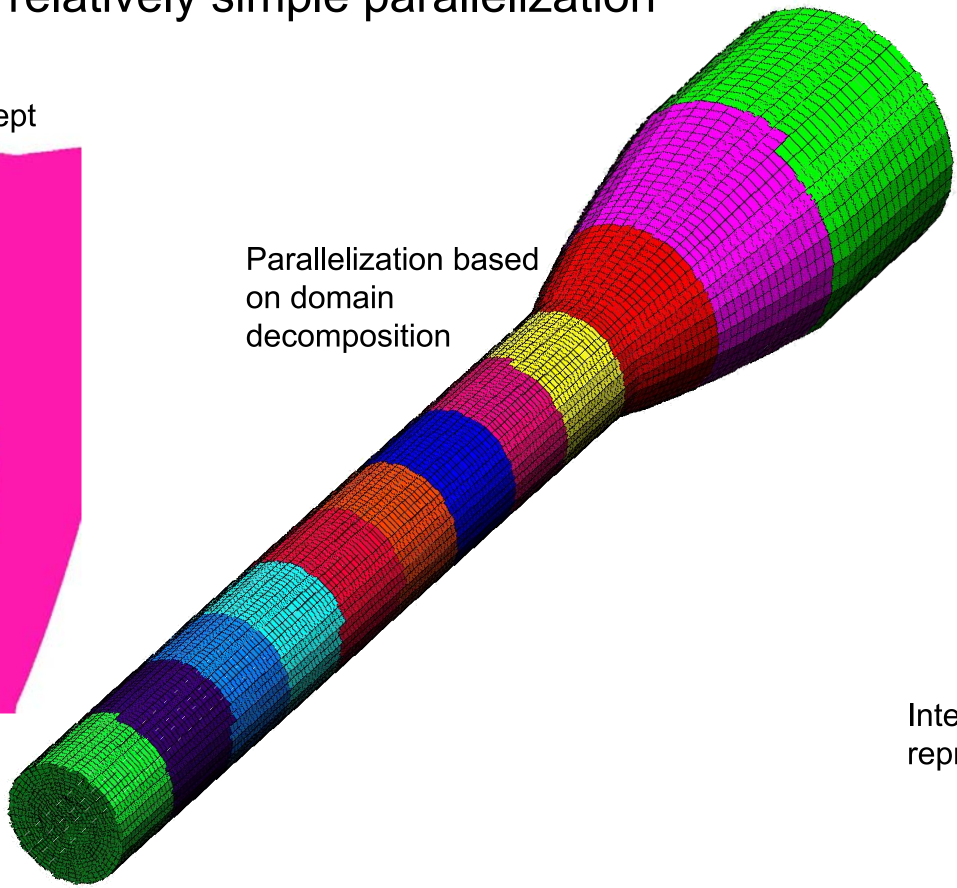
Benefits of DG-FEM for Levelset

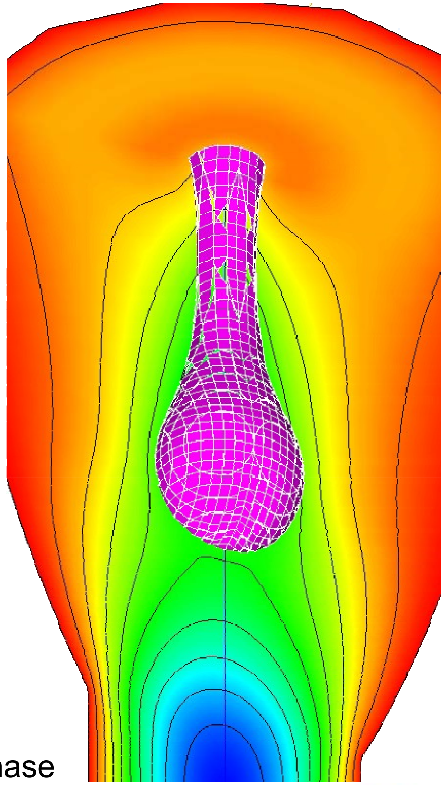
- possibility of reduction of the computational domain (only for levelset)
- mass preserving (interphase preserving!) reinitialization
- exact localization of the interphase (polygons)
- exact evaluation of volume fractions
- relatively simple parallelization

Layering concept



Parallelization based on domain decomposition





Interphase representation by polygons

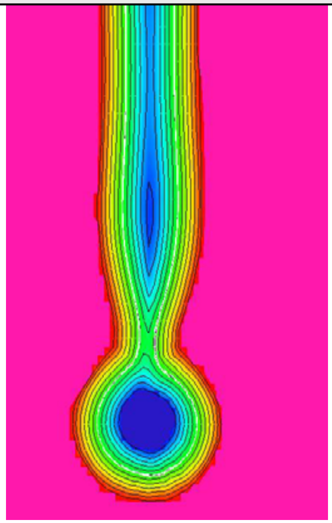
Continuum Surface Force (CSF):

- Transformation of the surface integrals to volume integrals with the help of a <u>regularized</u> Dirac delta function δ
- Requires globally defined normals and curvature
- Reduction of spurious oscillations

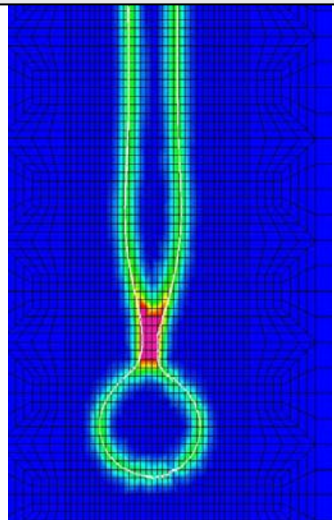
$$\mathbf{n}_{P_1} \xrightarrow{\text{L}_2\text{ projection}} \mathbf{n}_{Q_1} \qquad \text{continuous normal field}$$

$$\mathbf{f}_{\text{ST}} = \sigma\kappa\mathbf{n}\delta(x,\varepsilon)$$

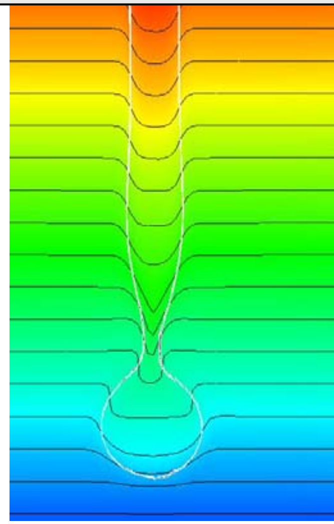
$$\kappa_{Q_1} = \frac{\int_\Omega \nabla\cdot\mathbf{n}_{Q_1}\,d\mathbf{x}}{\int_\Omega d\mathbf{x}} \qquad \text{continuous curvature field}$$



Levelset distribution



Distribution of the smoothed interphacial tension force $(\sigma\kappa\delta)_{Q_1}$



Resulting pressure distribution

Phase Field (PF) approach

$$\mathbf{f}_{\mathrm{ST}} = \sigma \nabla \cdot \left(\nabla \phi_{\mathrm{PF}} \times \nabla \phi_{\mathrm{PF}} \right)$$

- No reconstruction of normals and curvature needed
- Fully implicit treatment is possible
- Also possible for LS (?)

Semi implicit CSF formulation based on Laplace-Beltrami

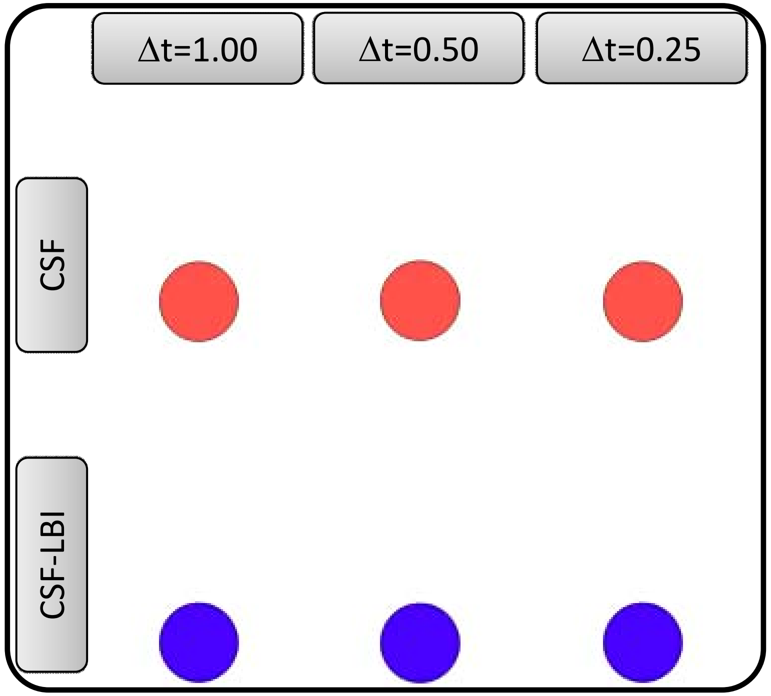
$$\mathbf{f}_{\mathrm{ST}} = \int_\Omega \sigma\kappa\hat{\mathbf{n}}\cdot\mathbf{v}\,\delta(\Gamma,\mathbf{x})\,d\mathbf{x} \qquad = \int_\Omega \sigma\left(\underline{\Delta}\mathbf{x}\big|_\Gamma\right)\cdot\left(\mathbf{v}\,\delta(\Gamma,\mathbf{x})\right)d\mathbf{x}$$

$$= -\int_\Omega \sigma\underline{\nabla}\mathbf{x}\big|_\Gamma\cdot\underline{\nabla}\left(\mathbf{v}\,\delta(\Gamma,\mathbf{x})\right)d\mathbf{x} = -\int_\Omega \sigma\underline{\nabla}\mathbf{x}\big|_\Gamma\cdot\underline{\nabla}\mathbf{v}\,\delta(\Gamma,\mathbf{x})\,d\mathbf{x}$$

Application of the semi-implicit time integration yields

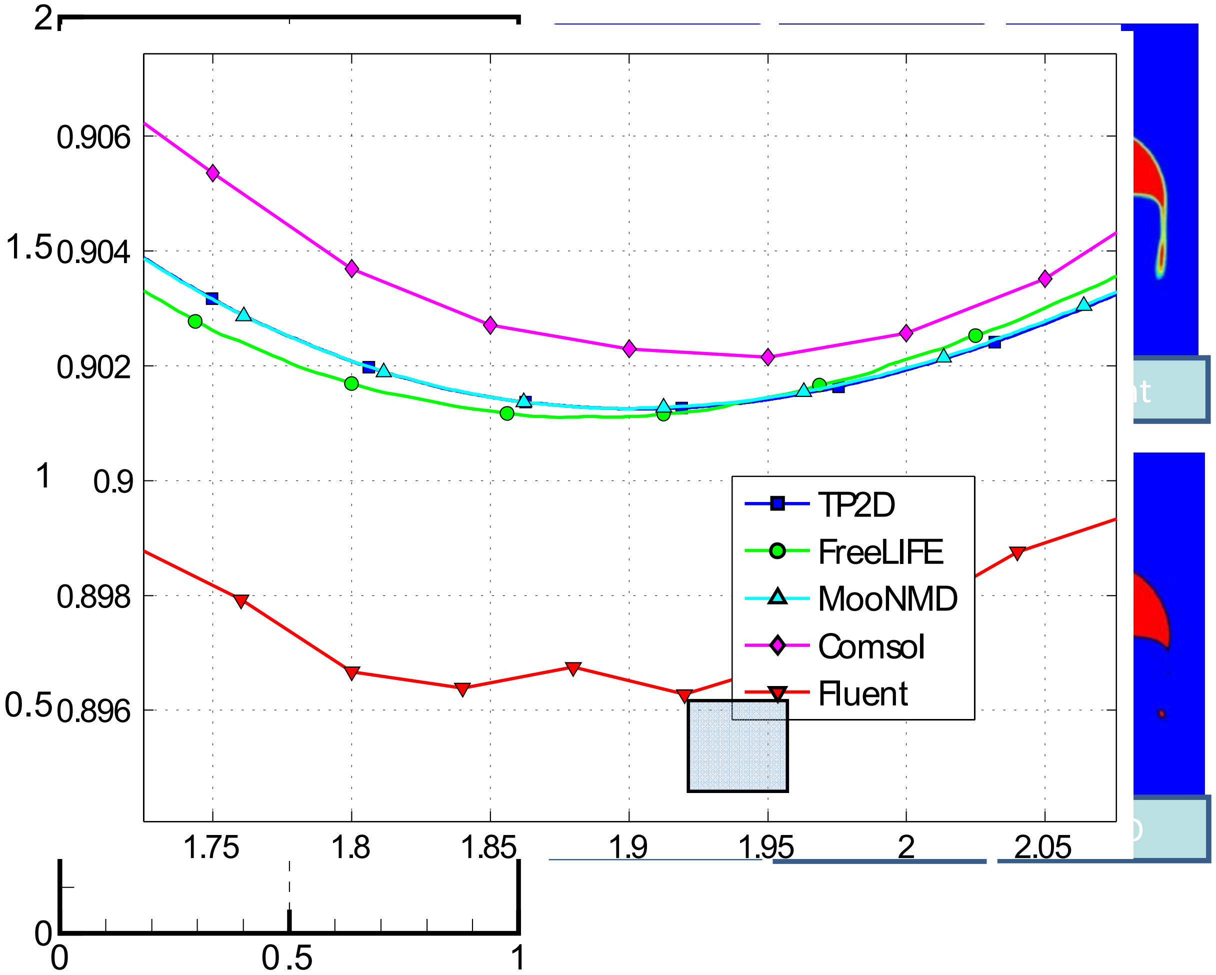
$$\mathbf{x}\big|_{\Gamma^{n+1}} = \mathbf{x}\big|_{\Gamma^n} + \Delta t\,\mathbf{u}^{n+1} \Longrightarrow \mathbf{f}_{\mathrm{ST}} = -\int_\Omega \sigma\,\delta_\varepsilon\left(dist(\Gamma^n,\mathbf{x})\right)\underline{\nabla}\tilde{\mathbf{x}}\big|_\Gamma^n\cdot\underline{\nabla}\mathbf{v}\,d\mathbf{x}$$

$$-\,\Delta t^{n+1}\int_\Omega \sigma\,\delta_\varepsilon\left(dist(\Gamma^n,\mathbf{x})\right)\underline{\nabla}\mathbf{u}^{n+1}\cdot\underline{\nabla}\mathbf{v}\,d\mathbf{x}$$



Advantages
- Relaxes capillary time step restriction
- „Optimal“ for FEM-Levelset approach

http://www.featflow.de/beta/en/benchmarks/



Benchmark quantities

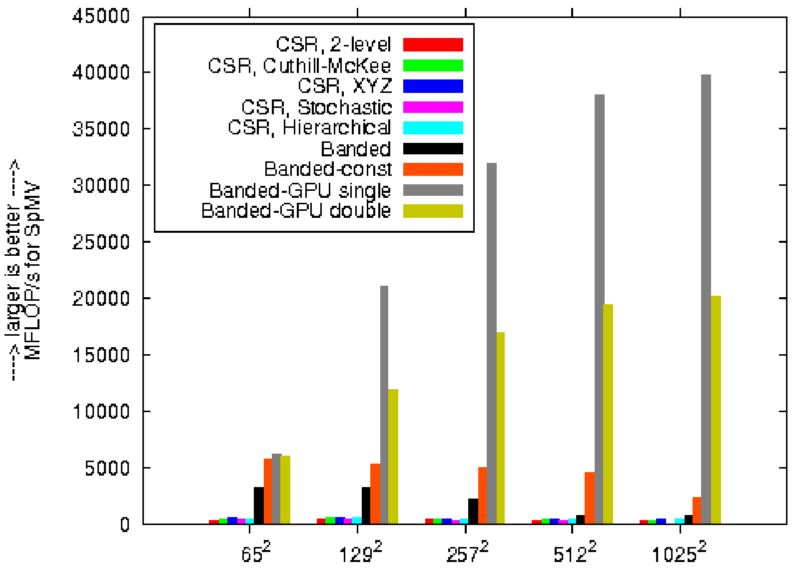
Center of mass	$\mathbf{x}_c = \dfrac{\int_{\Omega_2} \mathbf{x}\, dx}{\int_{\Omega_2} 1\, dx}$
Mean rise velocity	$\mathbf{U}_c = \dfrac{\int_{\Omega_2} \mathbf{u}\, dx}{\int_{\Omega_2} 1\, dx}$
Circularity	$\not{c} = \dfrac{P_a}{P_b} = \dfrac{\pi d_a}{P_b}$

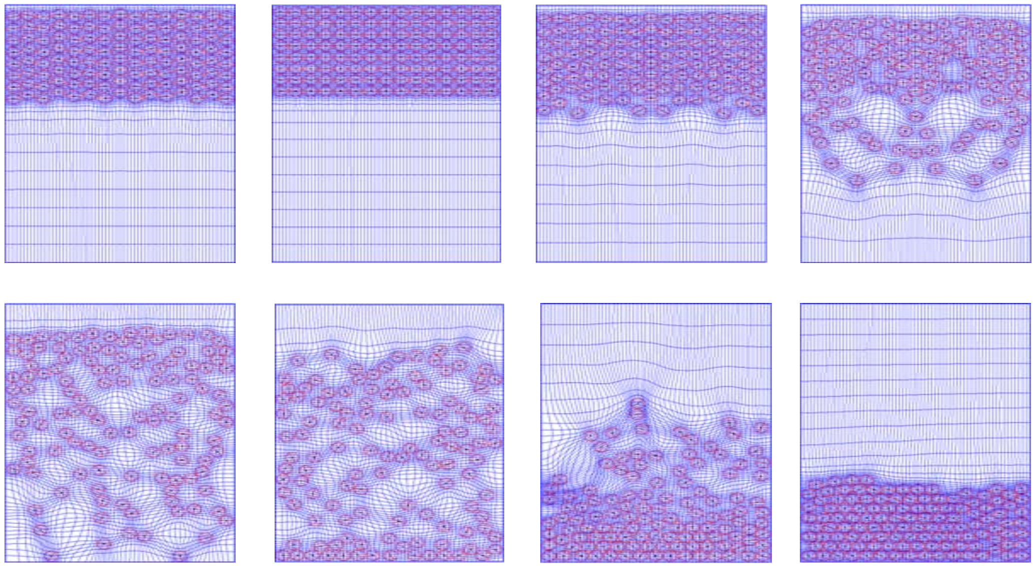
Hysing, S.; Turek, S.; Kuzmin, D.; Parolini, N.; Burman, E.; Ganesan, S.; Tobiska, L.: Quantitative benchmark computations of two-dimensional bubble dynamics, International Journal for Numerical Methods in Fluids, in press, DOI: 10.1002/fld.1934, 2009

Adaptive HPC Techniques

CELL processor
(PS3), 218 GFLOP/s,
Memory @ 3.2 GHz

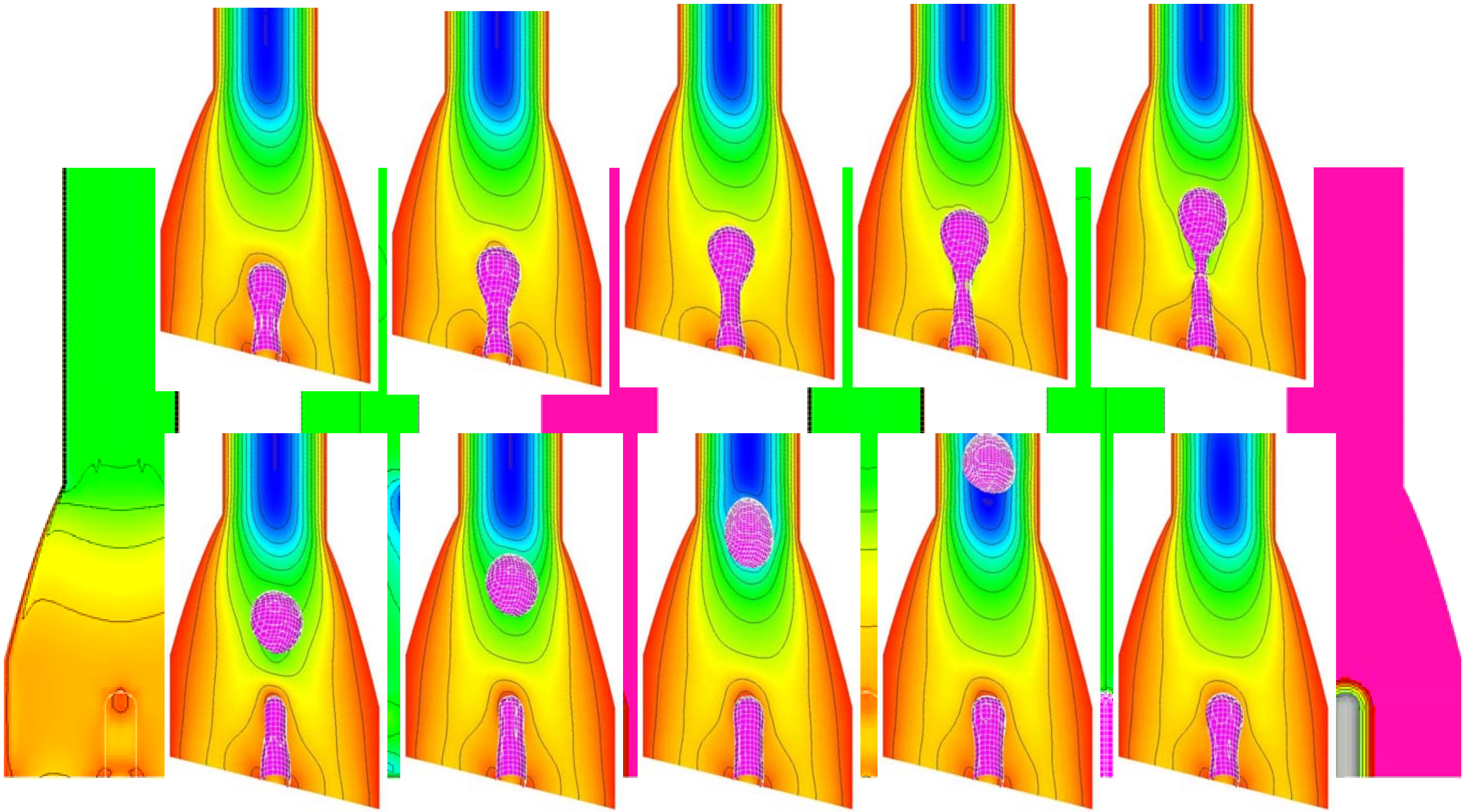
GPU (NVIDIA GTX 285):
240 cores @ 1.476 GHz,
1.242 GHz memory bus
(160 GB/s) ≈ 1.06 TFLOP/s





40 GFLOP/s, 140 GB/s on GeForce GTX 280

0.7 (1.4) GFLOP/s on 1 core of Xeon E5450



Next step: Extension to liquid-gas systems

Outlook