FEM techniques for interfacial flows

How to avoid the explicit reconstruction of interfaces

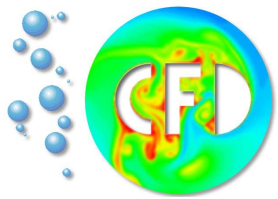
Stefan Turek, Shu-Ren Hysing

(ture@featflow.de)

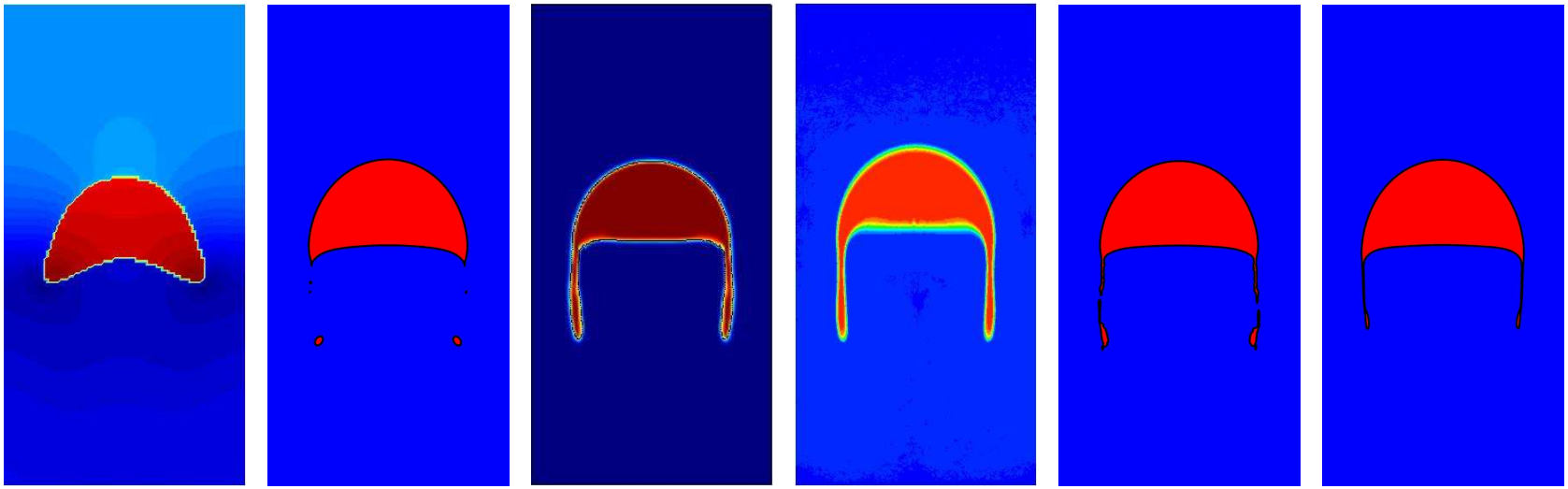
Institute for Applied Mathematics

University of Dortmund

Int. Conf. of Theoretical and Numerical Fluid Mechanics III
Vancouver – August 2007



Let's start with Benchmarking...



FELSOS	*TP2D*	*Femlab*	*CFX*	*FreeLIFE*	*MooNMD*

S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska;
Proposal for quantitative benchmark computations of bubble dynamics,
Submitted to Int. J. Numer. Meth. Fluids.

Benchmark: 2D Rising Bubble

Aim: Development of **quantitative** two-phase flow benchmarks for rigorous evaluation of new and existing methods

Bubble benchmark quantities

- Center of mass
- Circularity
- Rise velocity

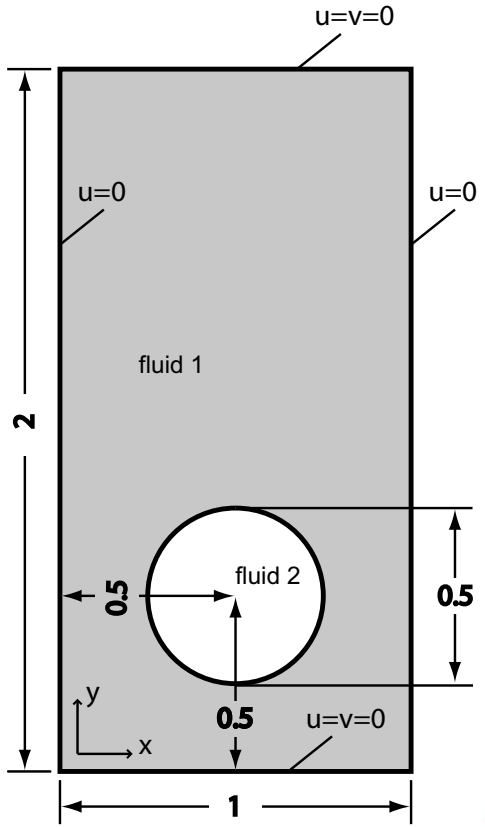
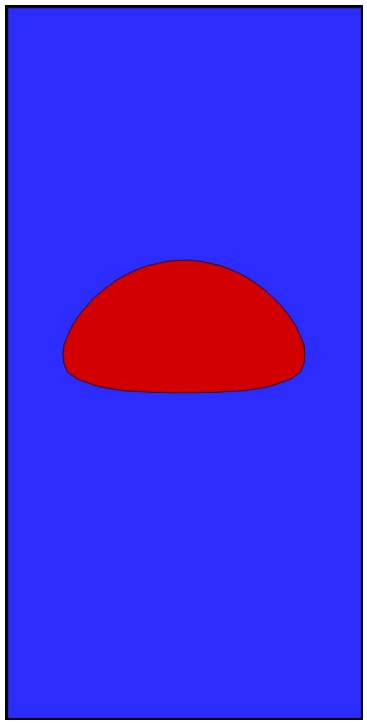


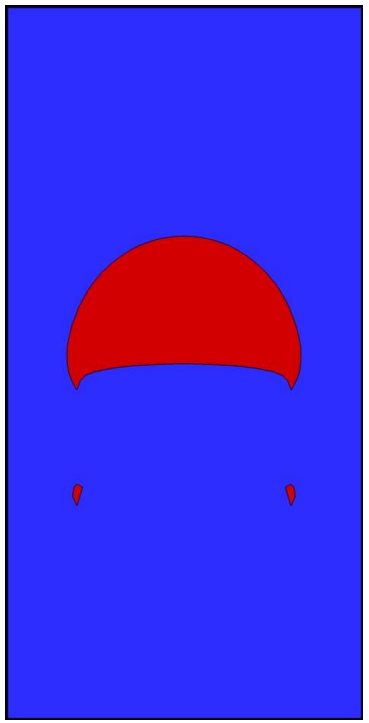
Figure: Initial configuration and boundary conditions for the test cases

Benchmark Test Cases

Test Case	1	2
ρ_1 (liquid)	1000	1000
ρ_2 (gas)	1	100
μ_1 (liquid)	10	10
μ_2 (gas)	0.1	1
g_y	-0.98	-0.98
σ	1.96	24.5
Re	35	35
Eo	125	10
ρ_1/ρ_2	1000	10
μ_1/μ_2	100	10



Test Case 1

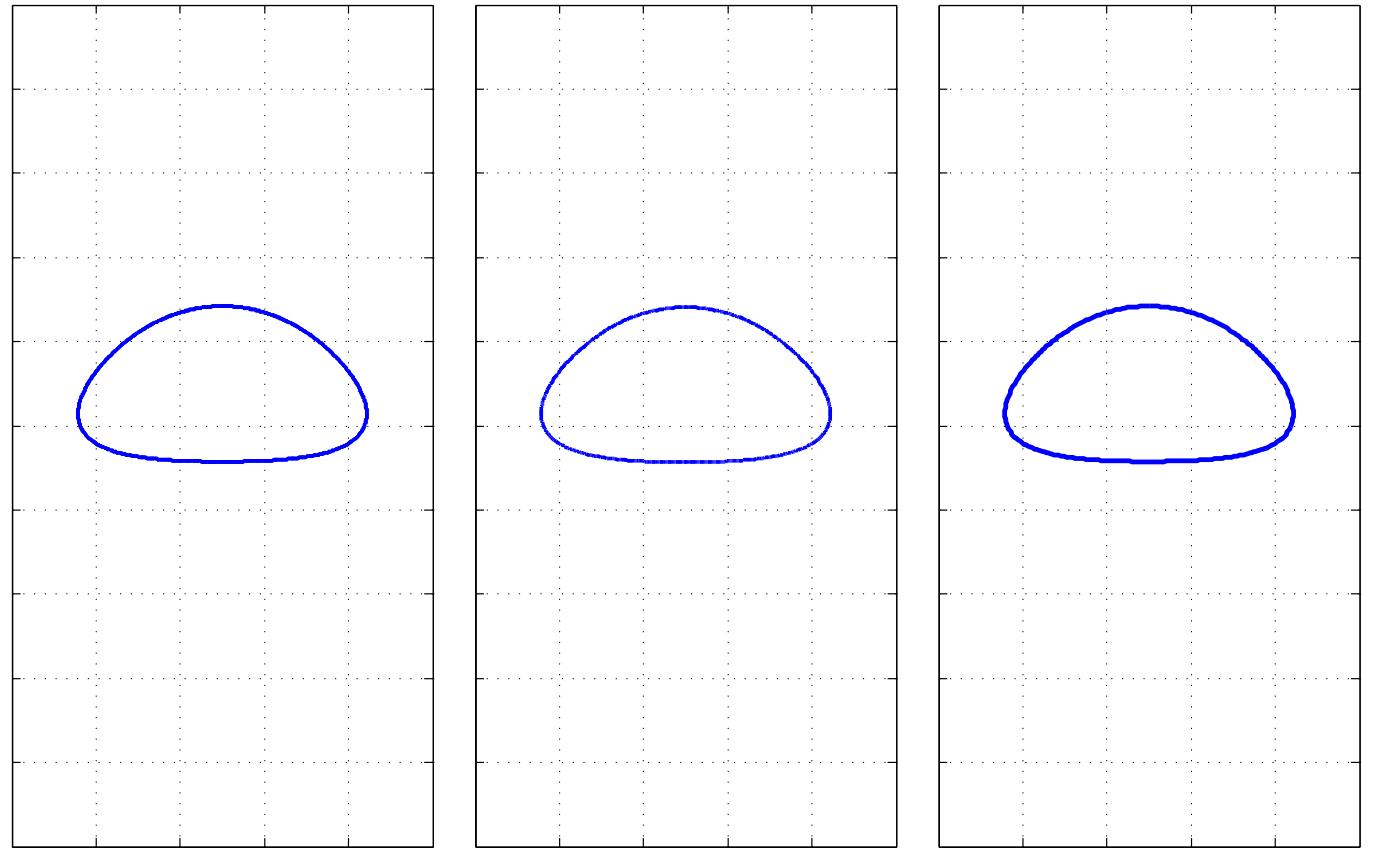


Test Case 2

Participating Groups

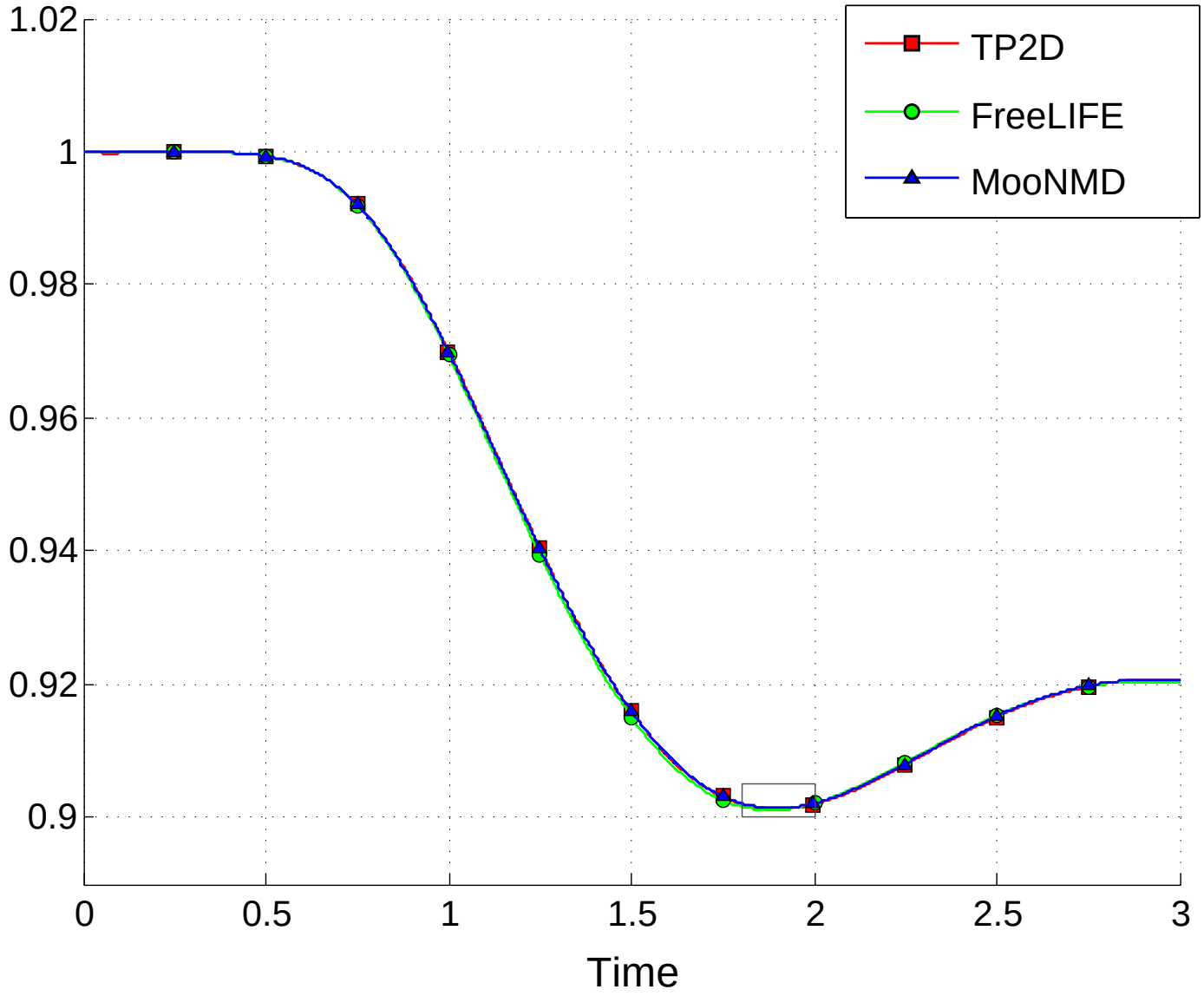
	Group and Affiliation	Code/Method
1	Uni. Dortmund, Inst. of Applied Math. *S. Turek, D. Kuzmin, S. Hysing*	**TP2D** *FEM-Level Set*
2	EPFL Lausanne, Inst. of Analysis and Sci. Comp. *E. Burman, N. Parolini*	**FreeLIFE** *FEM-Level Set*
3	Uni. Magdeburg, Inst. of Analysis and Num. Math. *L. Tobiska, S. Ganesan*	**MooNMD** *FEM-ALE*

Visual comparison: Test Case 1

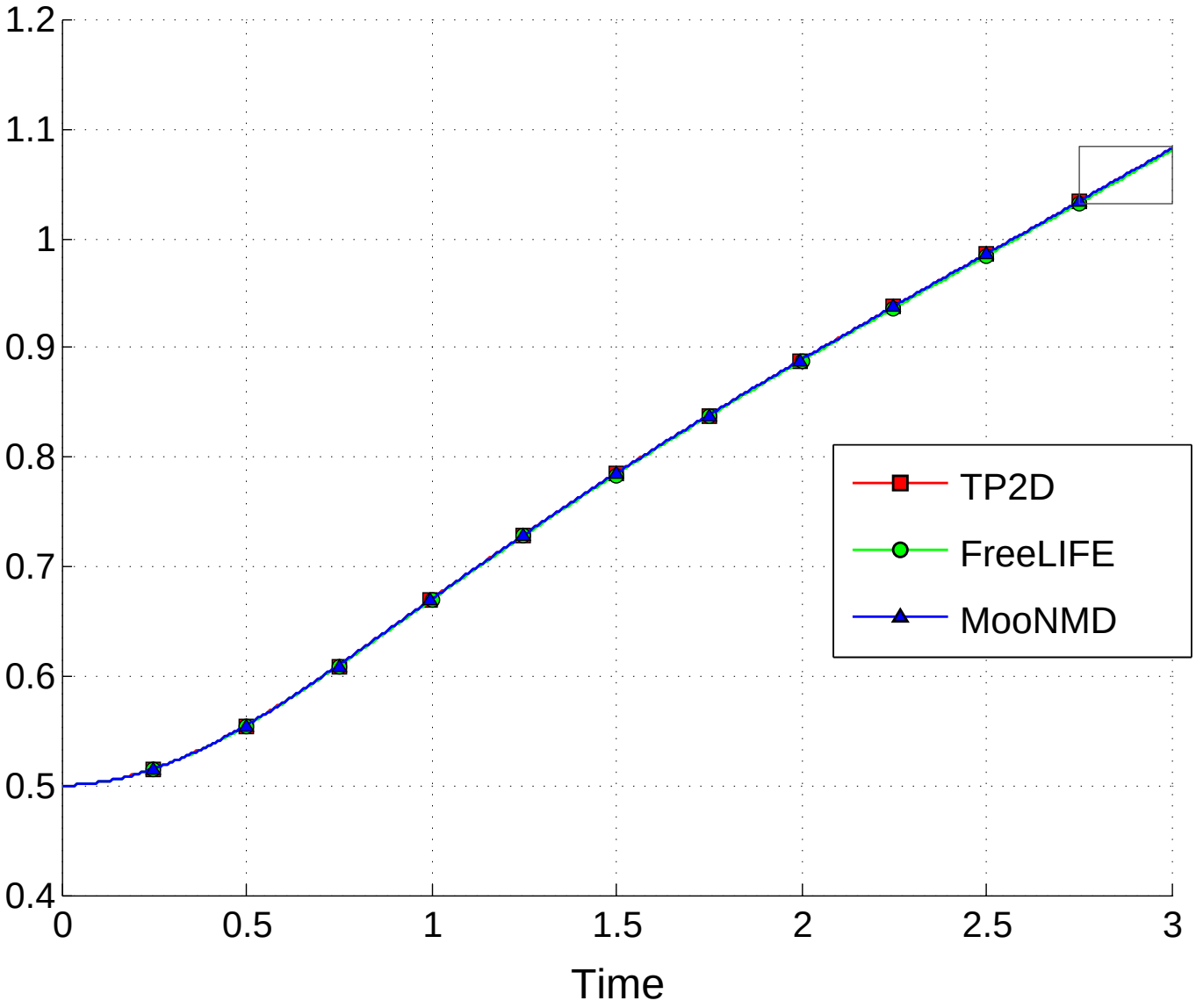


TP2D | *FreeLIFE* | *MooNMD*

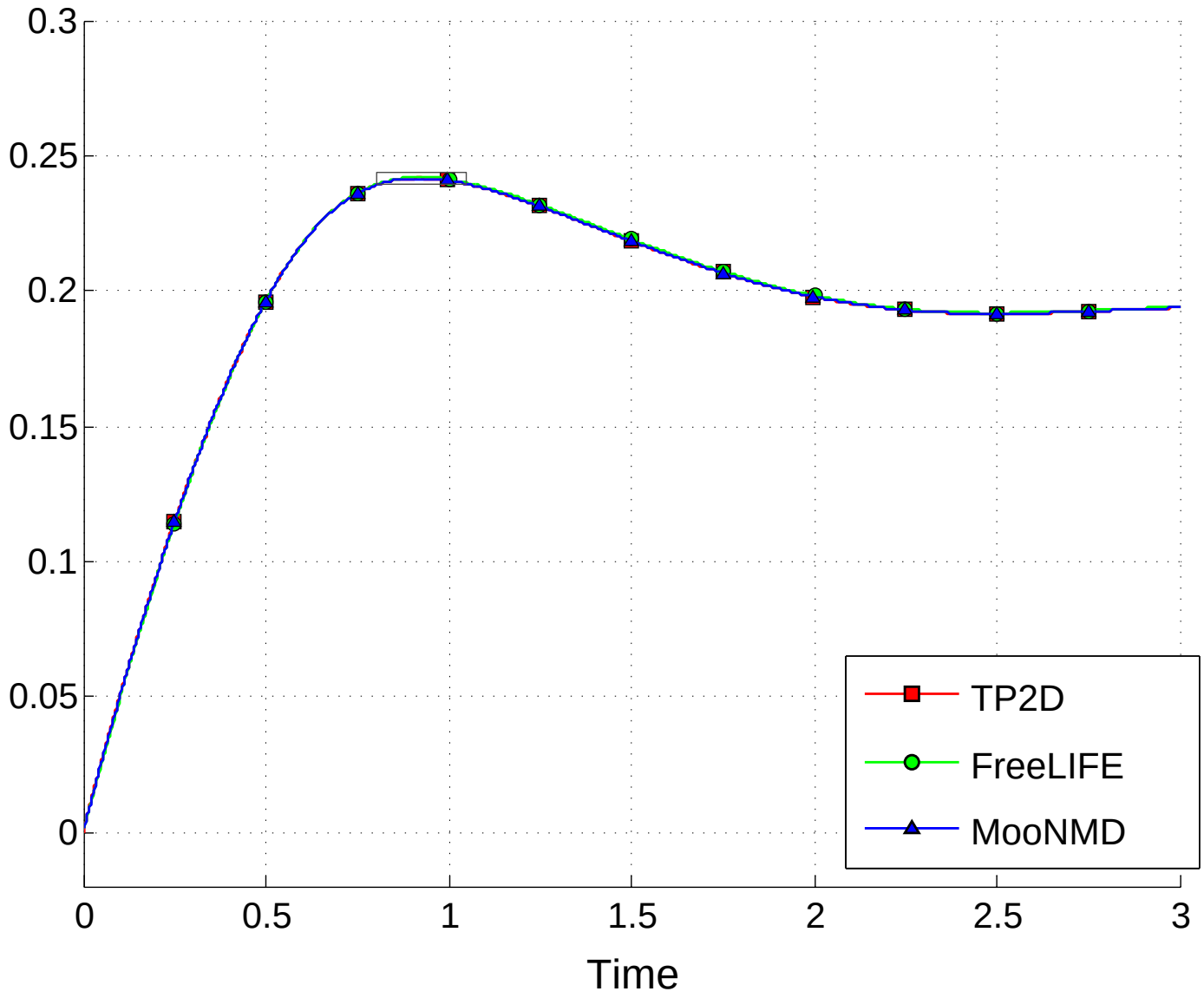
Benchmark quantities - circularity



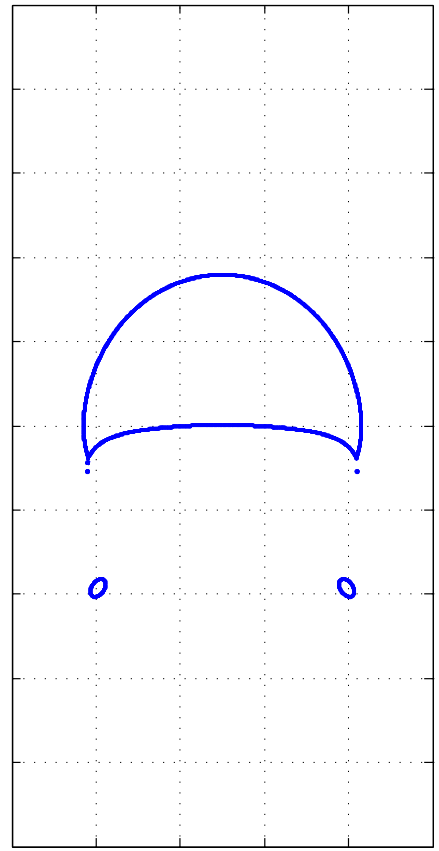
Benchmark quantities - center of mass



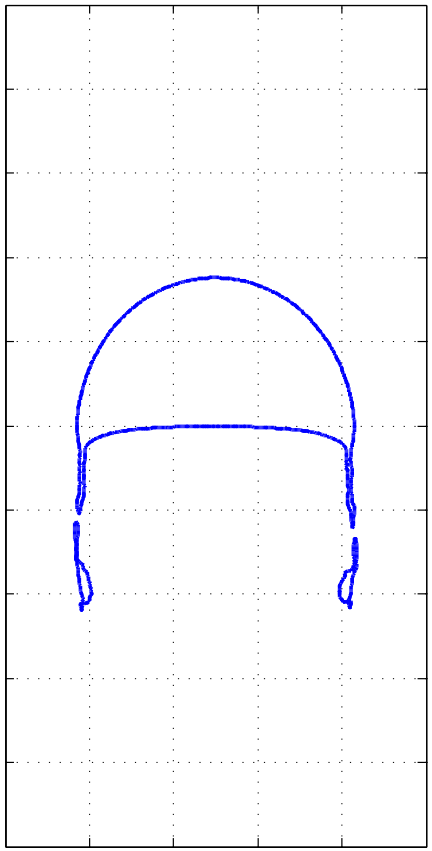
Benchmark quantities - rise velocity



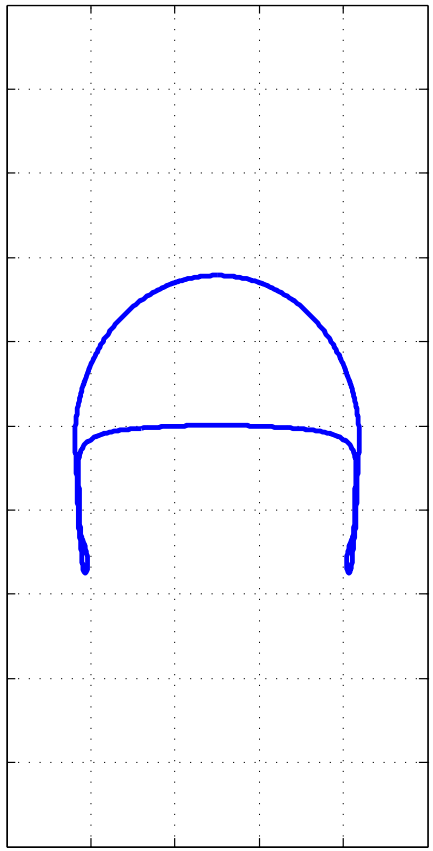
Visual comparison: Test Case 2



TP2D

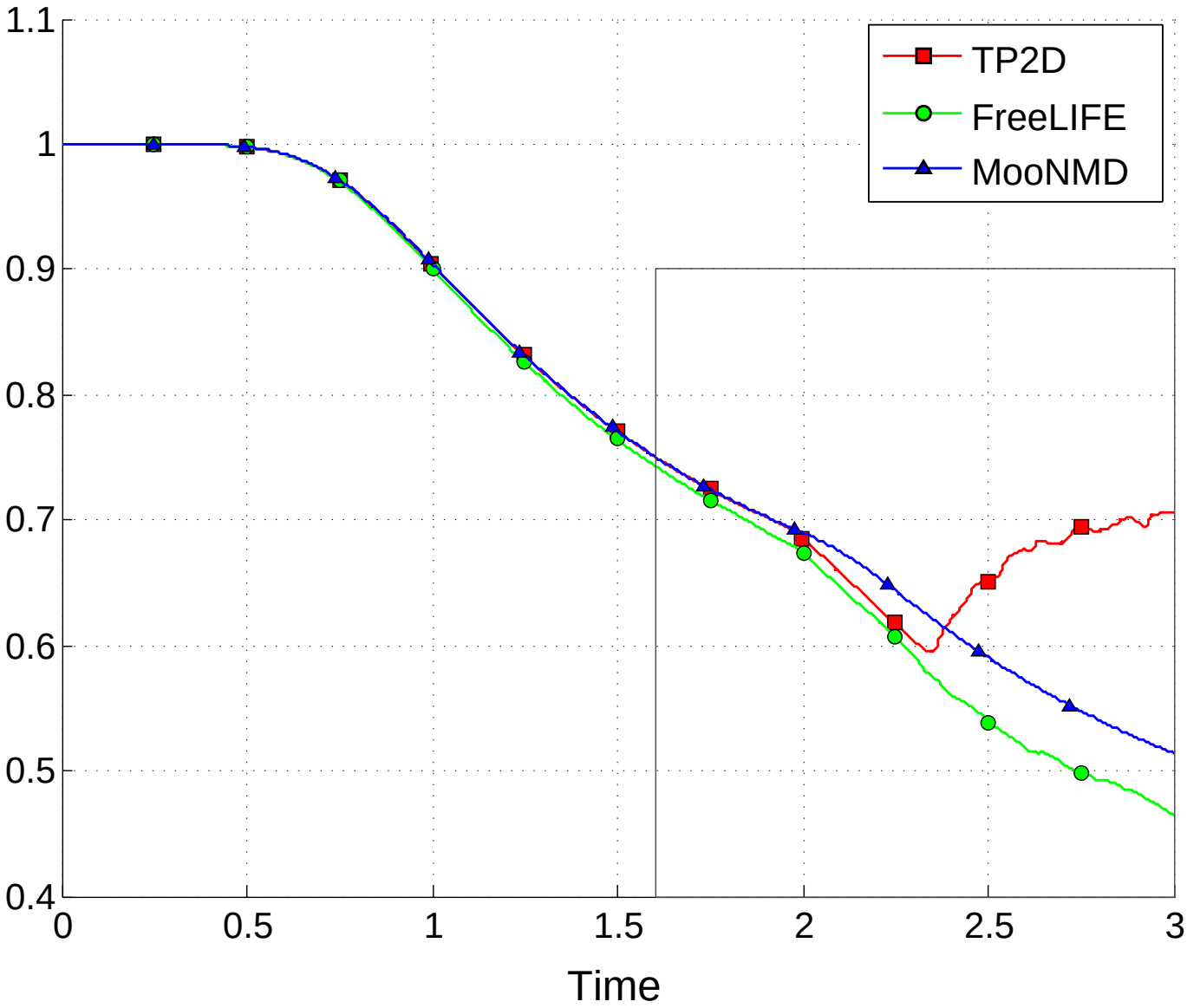


FreeLIFE

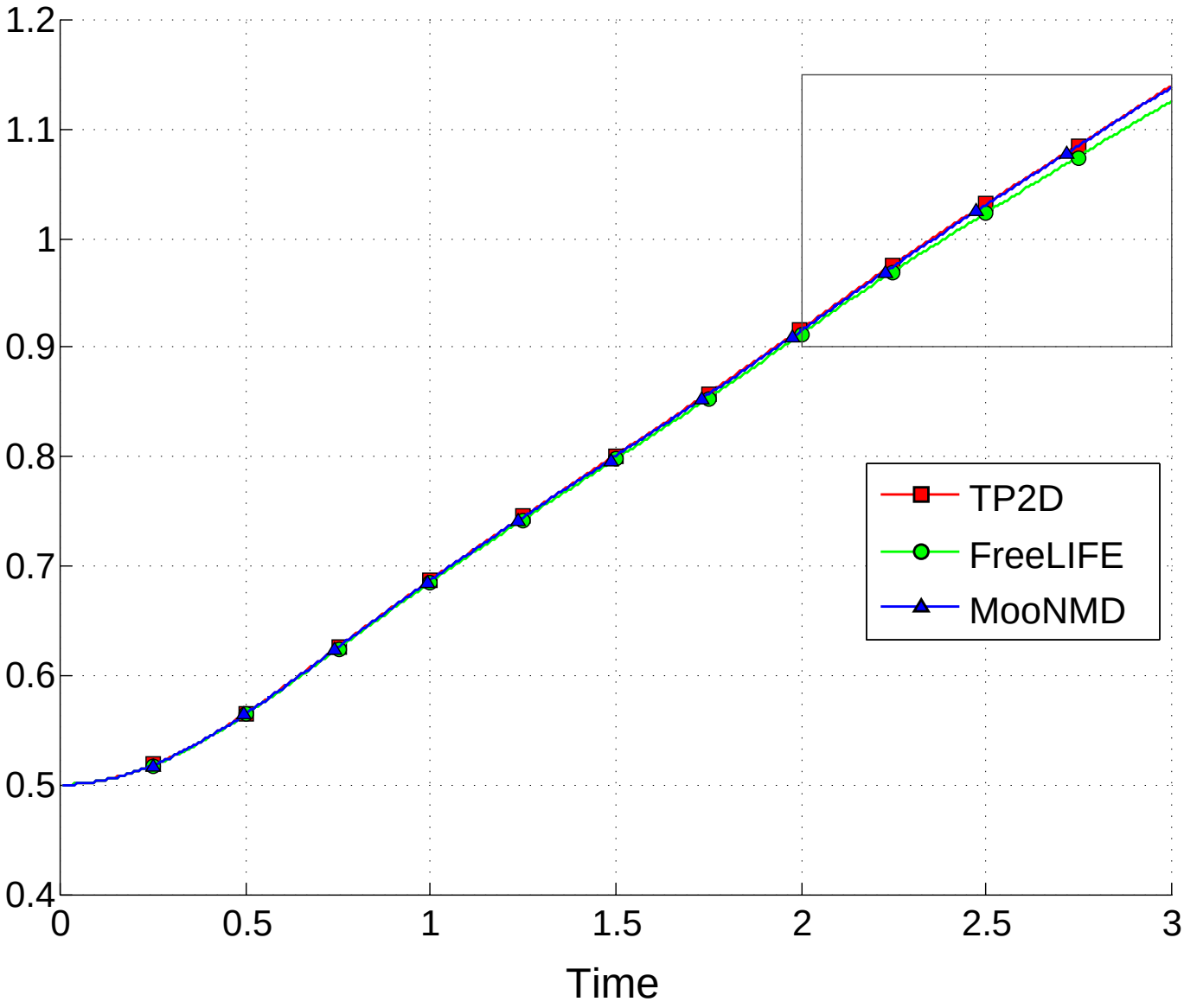


MooNMD

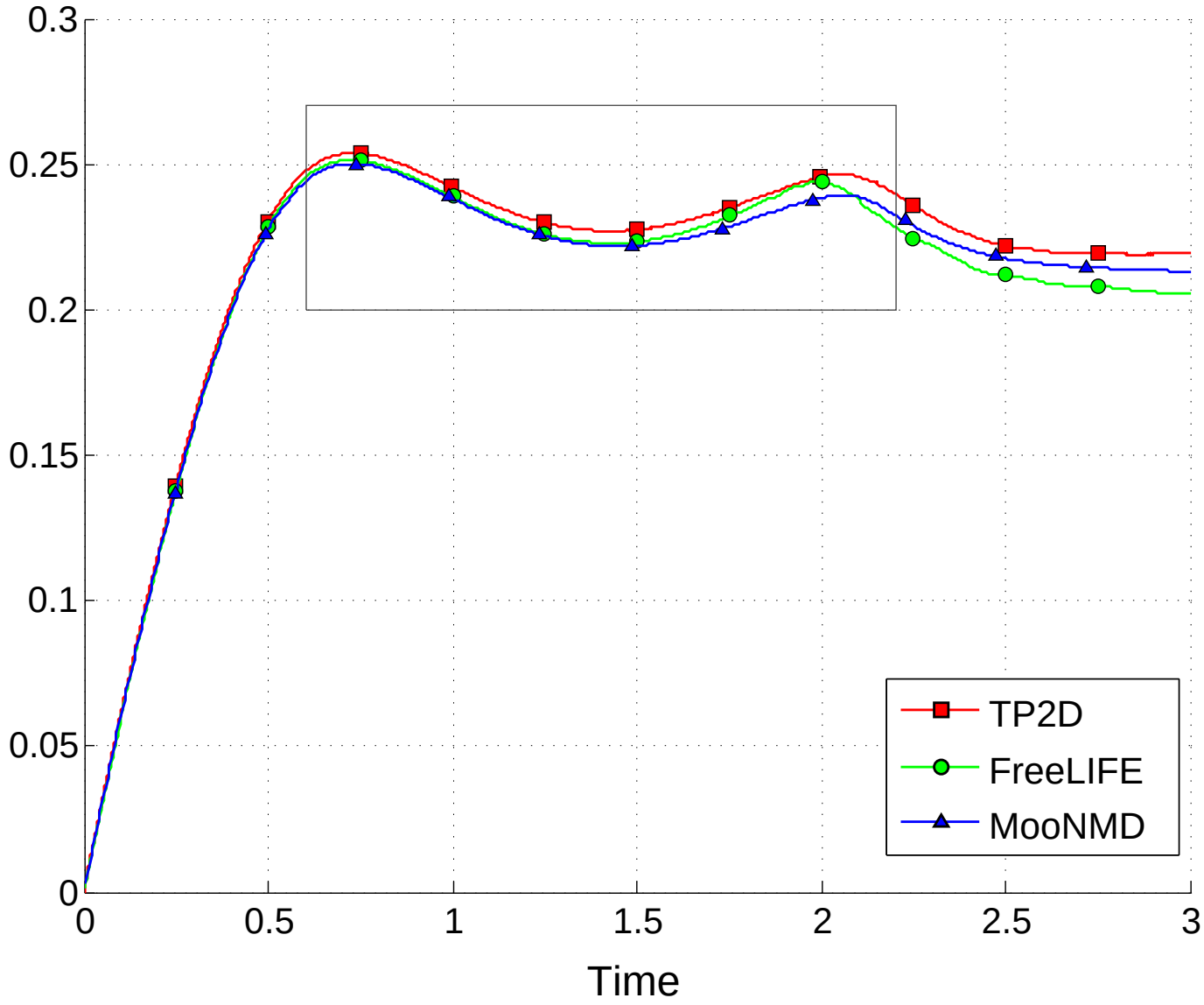
Benchmark quantities - circularity



Benchmark quantities - center of mass



Benchmark quantities - rise velocity



Conclusions

- Proposed two benchmarks for rigorous evaluation
- Established target reference values for the first benchmark
- Hinted at difficulties during break up in the second benchmark
- Look at `www.featflow.de` for participation

Interfacial flows: Challenging even in 2D !

Interfacial flows

Modeling of Interfacial Flows

Numerical simulation of interfacial or two-phase flows with immiscible fluids poses some challenging problems

Issues

- Accurate interface tracking
- Mass conservation
- Resolution of discontinuous fluid properties
- Treatment of interfacial boundary conditions

Modeling of Interfacial Flows

Numerical simulation of interfacial or two-phase flows with immiscible fluids poses some challenging problems

Issues

How to devise an efficient solution algorithm which addresses these issues?

Modeling of Interfacial Flows

- The Navier-Stokes equations govern incompressible fluid flow

$$\begin{aligned} \rho(\mathbf{x})\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) &= -\nabla p + \nabla\cdot\left(\mu(\mathbf{x})(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\right) + \rho(\mathbf{x})\mathbf{g} \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned}$$

with *varying* density $\rho(\mathbf{x})$ and viscosity $\mu(\mathbf{x})$ fields

Interfacial Boundary Conditions

- Direct interface conditions

$$[\mathbf{u}]|_\Gamma = 0, \quad -[-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)]\big|_\Gamma \cdot \hat{\mathbf{n}} = \sigma\kappa\hat{\mathbf{n}}$$

- Implicit conditions by weighted volume forces

$$\mathbf{f}_{st}|_\Gamma = \sigma\kappa\hat{\mathbf{n}}$$

Modeling of Interfacial Flows

- The Navier-Stokes equations govern incompressible fluid flow

$$\begin{aligned} \rho(\mathbf{x})\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) &= -\nabla p + \nabla\cdot\left(\mu(\mathbf{x})(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\right) + \rho(\mathbf{x})\mathbf{g} \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned}$$

  with *varying* density $\rho(\mathbf{x})$ and viscosity $\mu(\mathbf{x})$ fields

Interfacial Boundary Conditions

- Direct interface conditions

$$[\mathbf{u}]|_\Gamma = 0, \quad -[-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)]\big|_\Gamma \cdot \hat{\mathbf{n}} = \sigma\kappa\hat{\mathbf{n}}$$

- Implicit conditions by weighted volume forces

$$\mathbf{f}_{st}|_\Gamma = \sigma\kappa\hat{\mathbf{n}}$$

Saddle-point systems after Discretization

Given $\mathbf{u}^n$ and time step $k = \Delta t$, then solve for $\mathbf{u} = \mathbf{u}^{n+1}$ and $p = p^{n+1}$

$$\begin{cases} S\mathbf{u} + kBp &= \mathbf{f}^{n+1} \\ B^T\mathbf{u} &= 0 \end{cases}$$

with

$$S\mathbf{u} = [M_\rho + \theta k N(\mathbf{u})]\mathbf{u}$$

$$N(\mathbf{v}) = -\nabla \cdot \left(\mu(\mathbf{x})(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\right) + \rho(\mathbf{x})(\mathbf{u}\cdot\nabla)\mathbf{v}$$

B and B^T are discrete counterparts to the *grad* and *div* operators

Solver Structure

We prefer to use an operator splitting approach:

Given $\mathbf{u}^n$ and p^n, then do for time level $n+1$:

1. Solve the Navier-Stokes equations, with the interface given at Γ^n, to obtain $\mathbf{u}^{n+1}$ and p^{n+1}
2. (Optionally perform grid adaptation)
3. Move the interface according to the given physics
4. Perform post-processing; e.g. computation of normals and curvature
5. (Optionally perform grid adaptation)
6. (Re-iterate time loop if deemed necessary)

Interface Tracking

Interface Tracking

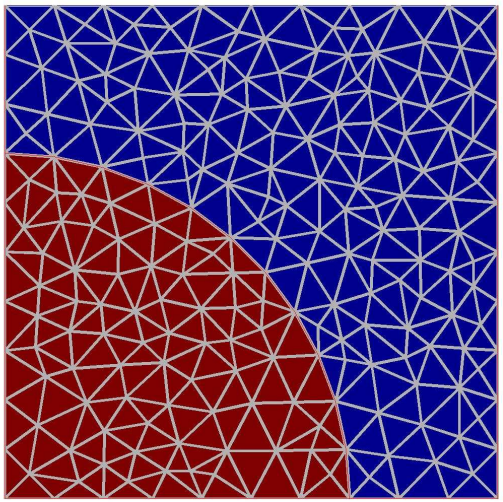
The tasks of an interface tracking algorithm are:

- Correctly propagate the interface given a velocity field
- Minimize additional mass loss due to the algorithm itself
- Enable easy and quick identification of both the interface and the different phases
- Enable easy reconstruction of interface normal vectors and curvature
- Be relatively easy to implement and efficient to solve
- Ideally treat coalescence and break-up automatically

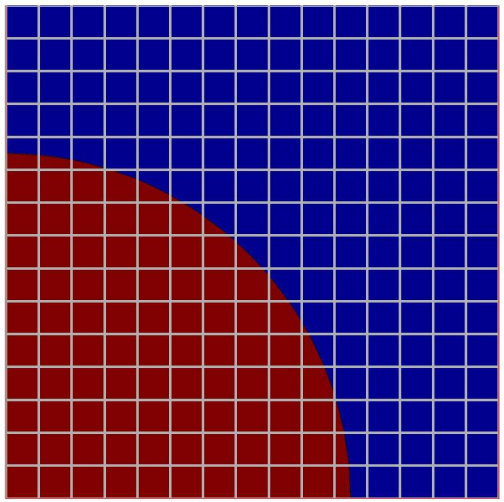
Interface Tracking

Two main schools of thought...

Lagrangian



Eulerian



Interface Tracking

Which method should we use then?

- Pure Lagrangian Approach
- Front Tracking Method
- Segment Projection Method
- Marker and Cell (MAC)
- Volume of Fluid (VOF)
- Phase Field Method
- Level Set Method (LS)
- Particle Level Set Method (PLS)
- Combination (LS-VOF)

Interface Tracking

Our selection criteria

- Eulerian, simplifies implementation
- Allow for discretization and solution with fast and efficient PDE solver techniques
- Handle coalescence and break-up automatically
- Allow for **global** reconstruction of normals and curvature

Interface Tracking

Our selection criteria

Level Set Method

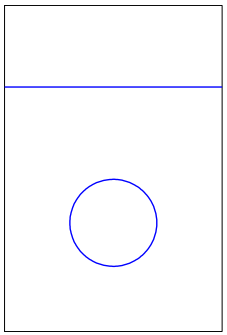
Interface Tracking

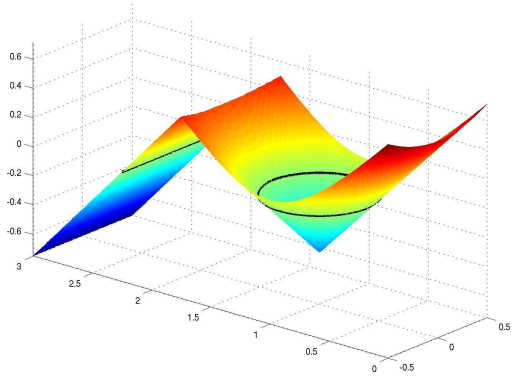
The Level Set Method

The general idea is to embed an interface in a higher dimensional function ϕ. The interface movement is then governed by a standard transport equation

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

with initial condition $\phi(\mathbf{x} = \Gamma, t = 0) = 0$





Properties

The Level Set Method - Properties

- A smoothness constraint on the level set field relaxes the requirements on the solver
- The natural choice is to restrict the level set field to be a signed distance function

$$|\nabla \phi| = 1$$

- At any given point the magnitude of the level set function will thus represent the shortest distance to the interface

Properties

The Level Set Method - Advantages

- The smooth LS function allows for higher order discretization techniques to be employed
- The governing transport equation can be solved with efficient standard solvers
- The interface is implicitly but also exactly defined
- Break up and coalescence is treated automatically
- Geometrical quantities such as normals and curvature can be reconstructed globally

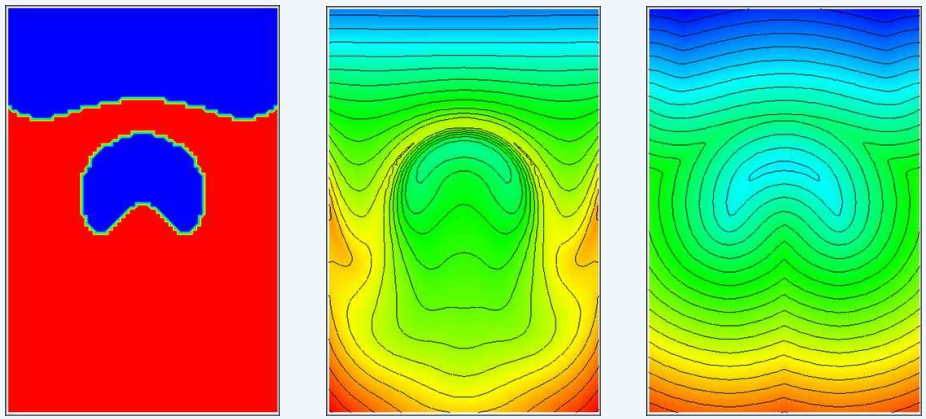
Problems

The Level Set Method - Problems

- The velocity field **v** convecting the level set function can not in general be expected to maintain ϕ as a distance function

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

- The distance function property is only preserved if $\nabla \mathbf{v} \cdot \nabla \phi = 0$
- Stretching and folding can cause eventual solver failure



Problems

The Level Set Method - Problems

Periodic Reinitialization

Reinitialization Methods

Several methods exist for constructing a distance function from given interface data. The simplest ones use interface approximations for the reconstruction

- Redistancing via "Brute Force"
- Algebraic Newton approach

The more complex algorithms deal with solving the Eikonal equation $|\nabla\phi| = F$ on a fixed mesh

- The Fast Marching Method
- The Fast Sweeping Method
- PDE Based redistancing
- Branch and Bound approach to redistancing

Reinitialization Methods

The different methods naturally have their respective strengths and weaknesses

- Brute force method
  - \+ Very robust
  - \- Scales quite badly, $\mathcal{O}(N * M)$
- Algebraic Newton Approach
  - \+ Potentially very fast, $\mathcal{O}(N)$
  - \- Convergence dependent upon approximate distance function
- The Fast Marching Method
  - \+ Easy to implement in an unstructured context
  - \- Needs a heap structure to sort marching order, $+\mathcal{O}(log N)$
- The Fast Sweeping Method
  - \+ Potentially very fast, $\mathcal{O}(N)$
  - \- Difficult to generalize to fully unstructured grids

PDE Based Redistancing

- The stationary limit of the following PDE can also be used to apply reinitialization to a given approximate distance field ϕ_0

$$\frac{\partial \phi^*}{\partial t} + \mathbf{q} \cdot \nabla \phi^* = S(\phi_0), \qquad \mathbf{q} = S(\phi_0) \frac{\nabla \phi^*}{|\nabla \phi^*|}$$

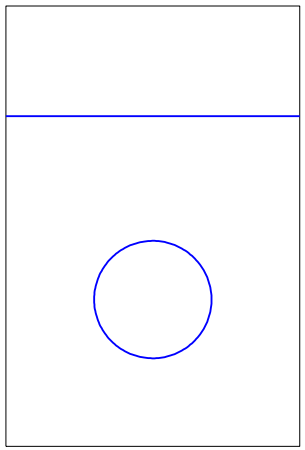
- where $S(\phi_0)$ is an appropriately chosen sign function
- **no local reconstruction necessary, but very expensive !**

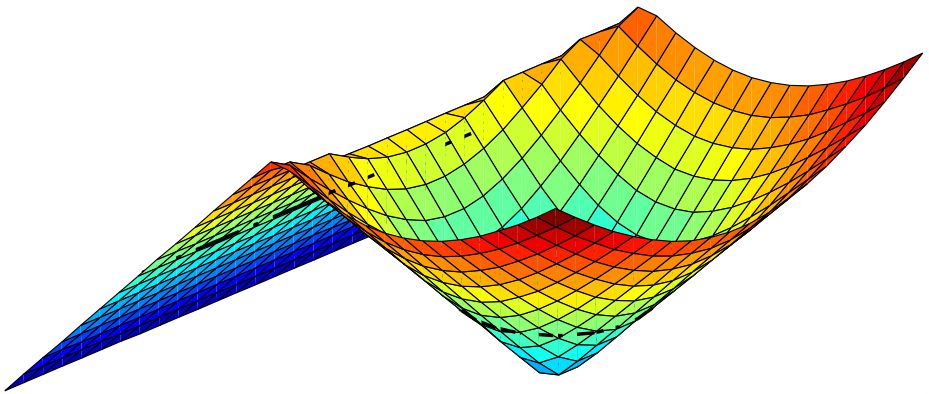
Test Case

A non-trivial test case was chosen to represent a typical level set computation with the following properties

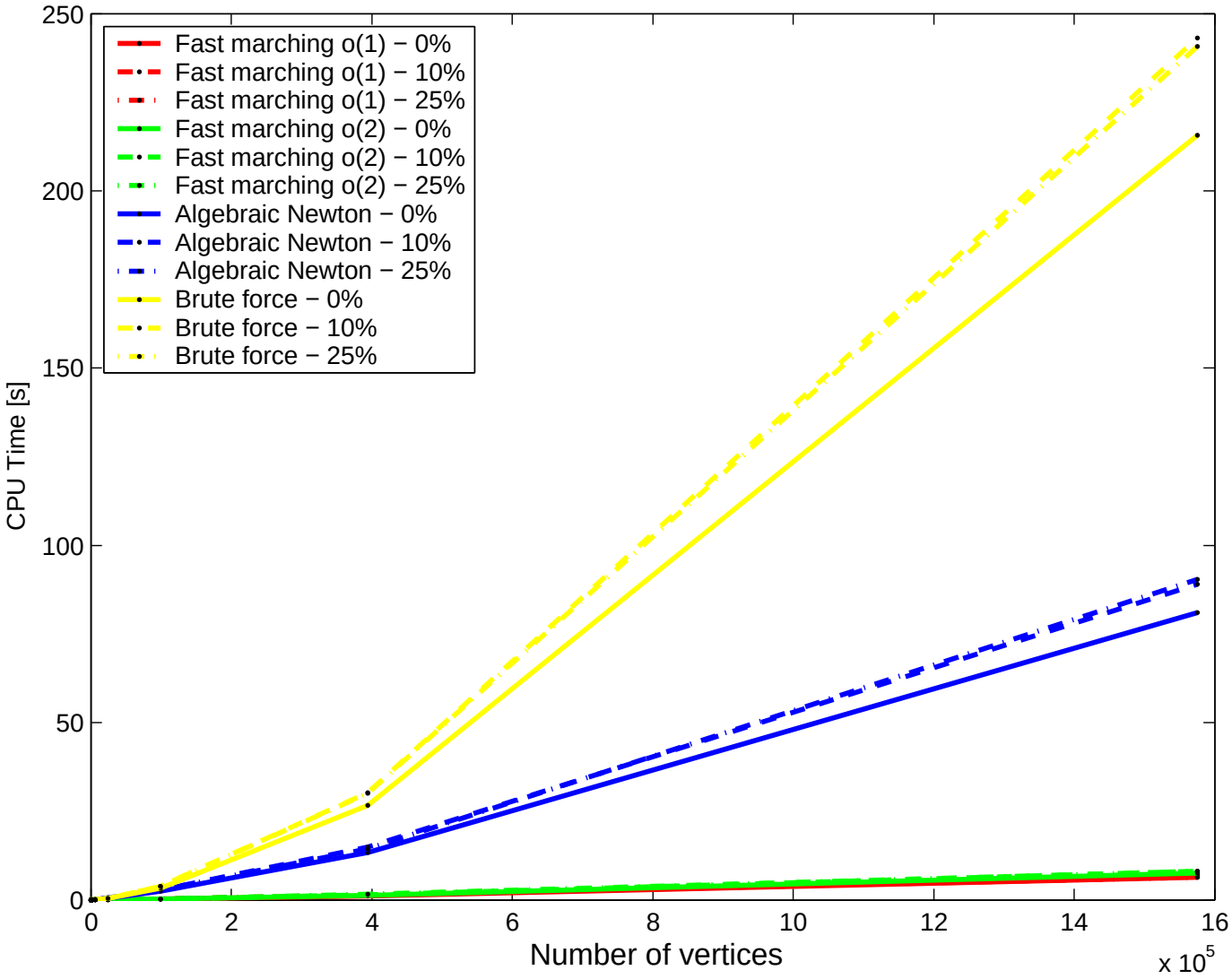
- Include both smooth regions and shocks
- Exact solution available:

$$\phi(\mathbf{x}) = min(2.25 - y, \sqrt{x^2 + (y-1)^2} - 0.4)$$

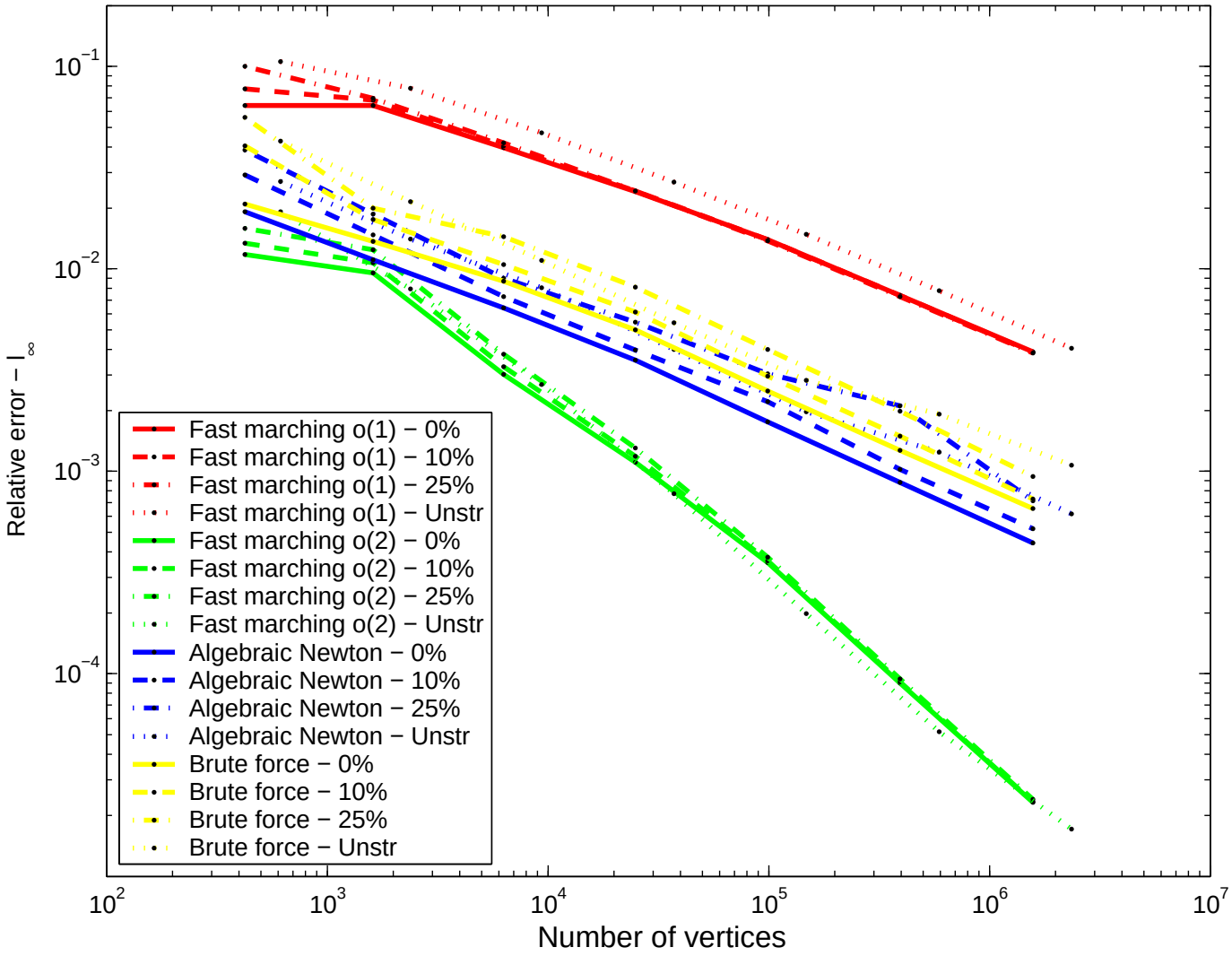




Results - CPU Time



Results - Accuracy



Reinitialization

- The fast marching method scales very well with increasing grid density
- Fast marching is very fast even for very dense grids ($O(10^6)$ grid points)
- The second order update only costs marginally more than the first order version
- The fast marching method is therefore our preferred algorithm! (**but: explicit local reconstruction**)

Treatment of Surface Tension

Modeling of Interfacial Flows

- Interfacial or two-phase flow where capillary surface tension forces are dominant poses some challenging problems

- Surface tension effects are generally modeled both explicitly in time and space leading to the capillary time step restriction

$$\Delta t_{num}^{(ca)} < \sqrt{\frac{\langle \rho \rangle \, h^3}{2\pi\sigma}}$$

Goal

Remove the capillary time step constraint
while retaining a fully Eulerian interface description

Modeling of Interfacial Flows

- Interfacial or two-phase flow where capillary surface tension forces are dominant poses some challenging problems

- Surface tension effects are generally modeled both explicitly in time and space leading to the capillary time step restriction

$$\Delta t_{num}^{(ca)} < \sqrt{\frac{\langle \rho \rangle \, h^3}{2\pi\sigma}}$$

Goal

Remove the capillary time step constraint
while retaining a fully Eulerian interface description

Computation of Surface Tension

Surface tension effects can essentially be included in the Navier-Stokes equations in two different ways

- Explicit interface reconstruction and direct evaluation

$$\mathbf{f}_{st} = \int_{\Gamma} \sigma \kappa \hat{\mathbf{n}} \, d\Gamma$$

- Implicit incorporation via the continuum surface force (CSF) model

$$\mathbf{f}_{st} = \int_{\Omega} \sigma \kappa \hat{\mathbf{n}} \delta(\Gamma) \, d\Omega$$

Definitions

Definition (Tangential gradient)

The tangential gradient *of a function f, which is differentiable in an open neighborhood of Γ, is defined by*

$$\underline{\nabla} f(x) = \nabla f(x) - (\hat{\mathbf{n}}(x) \cdot \nabla f(x))\hat{\mathbf{n}}(x), \quad x \in \Gamma$$

where ∇ denotes the usual gradient in $\mathbb{R}^d$

Definition (Laplace-Beltrami operator)

If f is two times differentiable in a neighborhood of Γ, then we define the Laplace-Beltrami operator *of f as*

$$\underline{\Delta} f(x) = \underline{\nabla} \cdot (\underline{\nabla} f(x)), \quad x \in \Gamma$$

Definitions and Derivation

Theorem

A theorem of differential geometry states that

$$\underline{\Delta}\mathrm{id}_\Gamma = \kappa\hat{\mathbf{n}}$$

where κ is the mean curvature and id_Γ is the identity mapping on Γ

Derivation

First take surface tension force source term, multiply it with the test function space $\mathbf{v}$, and apply partial integration

$$\begin{aligned}\mathbf{f}_{st} &= \int_\Gamma \sigma\kappa\hat{\mathbf{n}}\cdot\mathbf{v}\; d\Gamma = \int_\Gamma \sigma(\underline{\Delta}\mathrm{id}_\Gamma)\cdot\mathbf{v}\; d\Gamma = \\ &= -\int_\Gamma \sigma\underline{\nabla}\mathrm{id}_\Gamma\cdot\underline{\nabla}\mathbf{v}\; d\Gamma + \int_\gamma \sigma\partial_\gamma\mathrm{id}_\Gamma\cdot\mathbf{v}\; d\gamma\end{aligned}$$

Fully Implicit Evaluation in Space

- Boundary integrals can be transformed to volume integrals with the help of a Dirac delta function $\delta(\Gamma, \mathbf{x})$

$$\begin{aligned}\mathbf{f}_{st} &= \int_\Omega \sigma\kappa\hat{\mathbf{n}} \cdot \mathbf{v}\ \delta(\Gamma, \mathbf{x})\ d\mathbf{x} = \int_\Omega \sigma(\underline{\Delta}\mathrm{id}_\Gamma) \cdot (\mathbf{v}\delta(\Gamma, \mathbf{x}))\ d\mathbf{x} \\ &= -\int_\Omega \sigma\underline{\nabla}\mathrm{id}_\Gamma \cdot \underline{\nabla}(\mathbf{v}\delta(\Gamma, \mathbf{x}))\ d\mathbf{x} = -\int_\Omega \sigma\underline{\nabla}\mathrm{id}_\Gamma \cdot \underline{\nabla}\mathbf{v}\ \delta(\Gamma, \mathbf{x})\ d\mathbf{x}\end{aligned}$$

- Application of the semi-implicit time integration

$$\Gamma^{n+1} = \Gamma^n + \Delta t\ \mathbf{u}^{n+1}$$

yields

$$\begin{aligned}\mathbf{f}_{st} &= -\int_\Omega \sigma\underline{\nabla}(\mathrm{id}_\Gamma)^n \cdot \underline{\nabla}\mathbf{v}\ \delta(\Gamma^n, \mathbf{x})\ d\mathbf{x} \\ &\quad - \Delta t^{n+1} \int_\Omega \sigma\underline{\nabla}\mathbf{u}^{n+1} \cdot \underline{\nabla}\mathbf{v}\ \delta(\Gamma^n, \mathbf{x})\ d\mathbf{x}\end{aligned}$$

Regularization

- Regularization of $\delta(\Gamma, \mathbf{x})$ can easily be accomplished with the help of a **distance function**

$$\delta_{\boldsymbol{\epsilon}}(\Gamma, \mathbf{x}) = \delta_{\epsilon}(\mathit{dist}(\Gamma, \mathbf{x})),$$

  where $\mathit{dist}(\Gamma, \mathbf{x})$ gives the minimum distance from $\mathbf{x}$ to Γ

- The regularized continuous delta function δ_ϵ is defined as

$$\delta_\epsilon(x) = \begin{cases} \frac{1}{\epsilon}\varphi(x/\epsilon) & |x| \leq \epsilon \quad = \quad mh, \\ 0 & |x| > \epsilon \quad = \quad mh, \end{cases}$$

  where h is the mesh spacing which together with the constant m defines the support ϵ of the regularized delta function,
  φ is a characteristic function determining the kernel shape

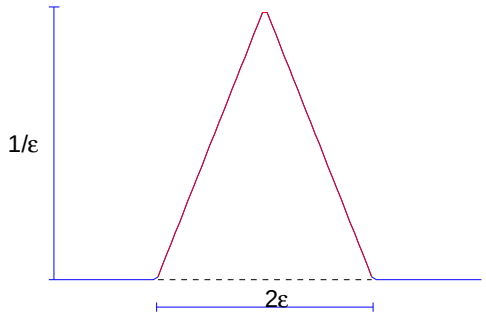
- A second order method can alternatively be constructed according to [Engquist, Tornberg, and Tsai: J. Comput. Phys. 207:28-51, 2005]

Regularization

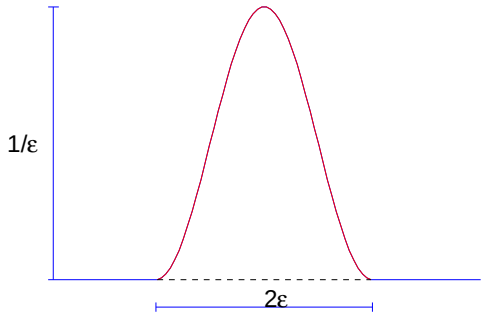
There are several possible choices for $\varphi(\xi)$, common choices are:

- The linear hat function $\varphi^1(\xi) = 1 - |\xi|$
- The commonly used cosine approximation $\varphi^2(\xi) = \frac{1}{2}(1 + cos(\frac{\pi\xi}{2}))$
- Higher order polynomials, for example;

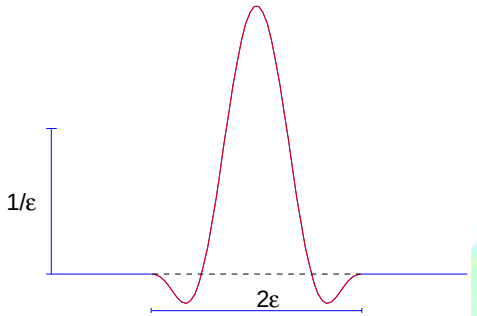
$$\varphi^3(\xi) = \frac{312}{512}(3 - 20\xi^2 + 42\xi^4 - 36\xi^6 + 11\xi^8)$$



(a) $\delta_\epsilon(\varphi^1)$



(b) $\delta_\epsilon(\varphi^2)$



(c) $\delta_\epsilon(\varphi^3)$

Implicit Surface Tension Force Expression

The surface tension forces are finally given by ...

Implicit Surface Tension Force Expression

$$\begin{aligned}\mathbf{f}_{st} &= -\int_{\Omega} \sigma\ \delta_{\epsilon}(dist(\Gamma^n, \mathbf{x}))\ \underline{\nabla}(\widetilde{\mathrm{id}}_{\Gamma})^n \cdot \underline{\nabla}\mathbf{v}\ d\mathbf{x} \\ &\quad -\Delta t^{n+1} \int_{\Omega} \sigma\ \delta_{\epsilon}(dist(\Gamma^n, \mathbf{x}))\ \underline{\nabla}\mathbf{u}^{n+1} \cdot \underline{\nabla}\mathbf{v}\ d\mathbf{x}\end{aligned}$$

Implicit Surface Tension Force Expression

The surface tension forces are finally given by ...

Implicit Surface Tension Force Expression

$$\begin{aligned}\mathbf{f}_{st} &= -\int_{\Omega} \sigma\ \delta_{\epsilon}(dist(\Gamma^n, \mathbf{x}))\ \underline{\nabla}(\widetilde{\mathrm{id}}_{\Gamma})^n \cdot \underline{\nabla}\mathbf{v}\ d\mathbf{x} \\ &\quad -\Delta t^{n+1} \int_{\Omega} \sigma\ \delta_{\epsilon}(dist(\Gamma^n, \mathbf{x}))\ \underline{\nabla}\mathbf{u}^{n+1} \cdot \underline{\nabla}\mathbf{v}\ d\mathbf{x}\end{aligned}$$

Level Set Method

A number of key points makes the *level set method* an ideal candidate for interface tracking algorithm when implementing the proposed surface tension force expressions

- Distance functions are in general readily available allowing for simple construction of the regularized Dirac delta functions
- Geometrical quantities such as normal and tangent vectors can be reconstructed globally, eliminating the need to extend these quantities from the interface separately
- The level set method can be coupled with the finite element method giving access to the variational form of the equations

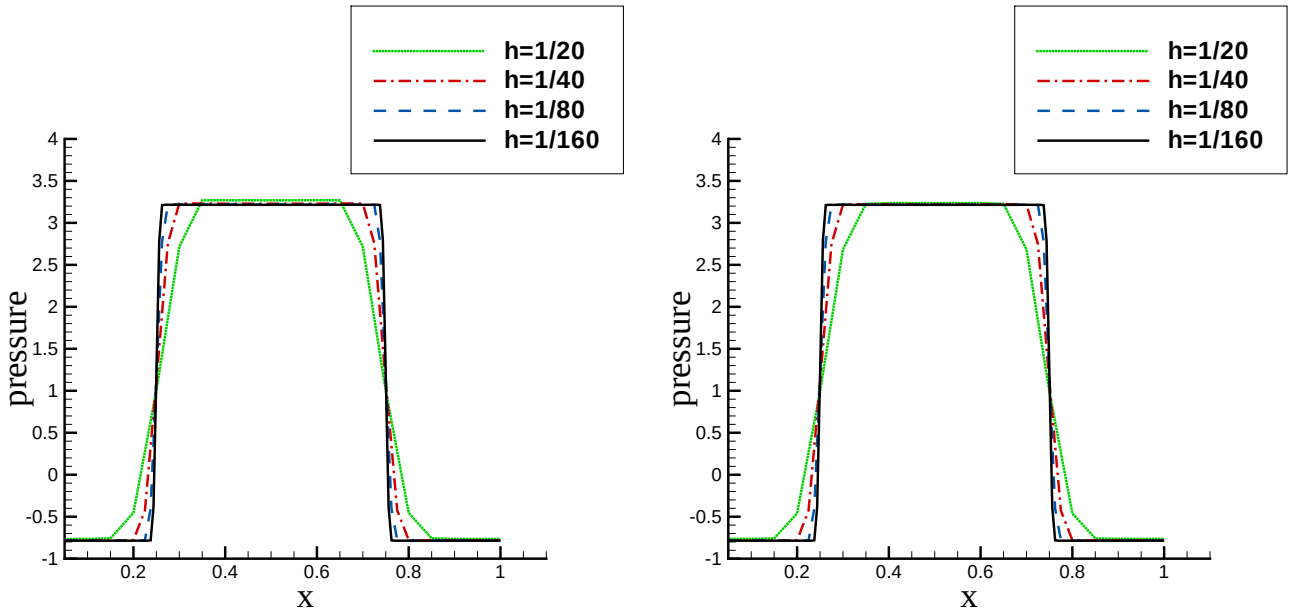
Method Validation

Numerical Examples

Validation, Laplace-Young Law

A perfect circular static bubble should follow the Laplace-Young law

$$p_{inside} = p_{outside} + \frac{\sigma}{r}$$



(a) CSF (b) CSF-LBI

Figure: Pressure cut-line for four different mesh sizes.

Example, Oscillating Bubble (CSF)

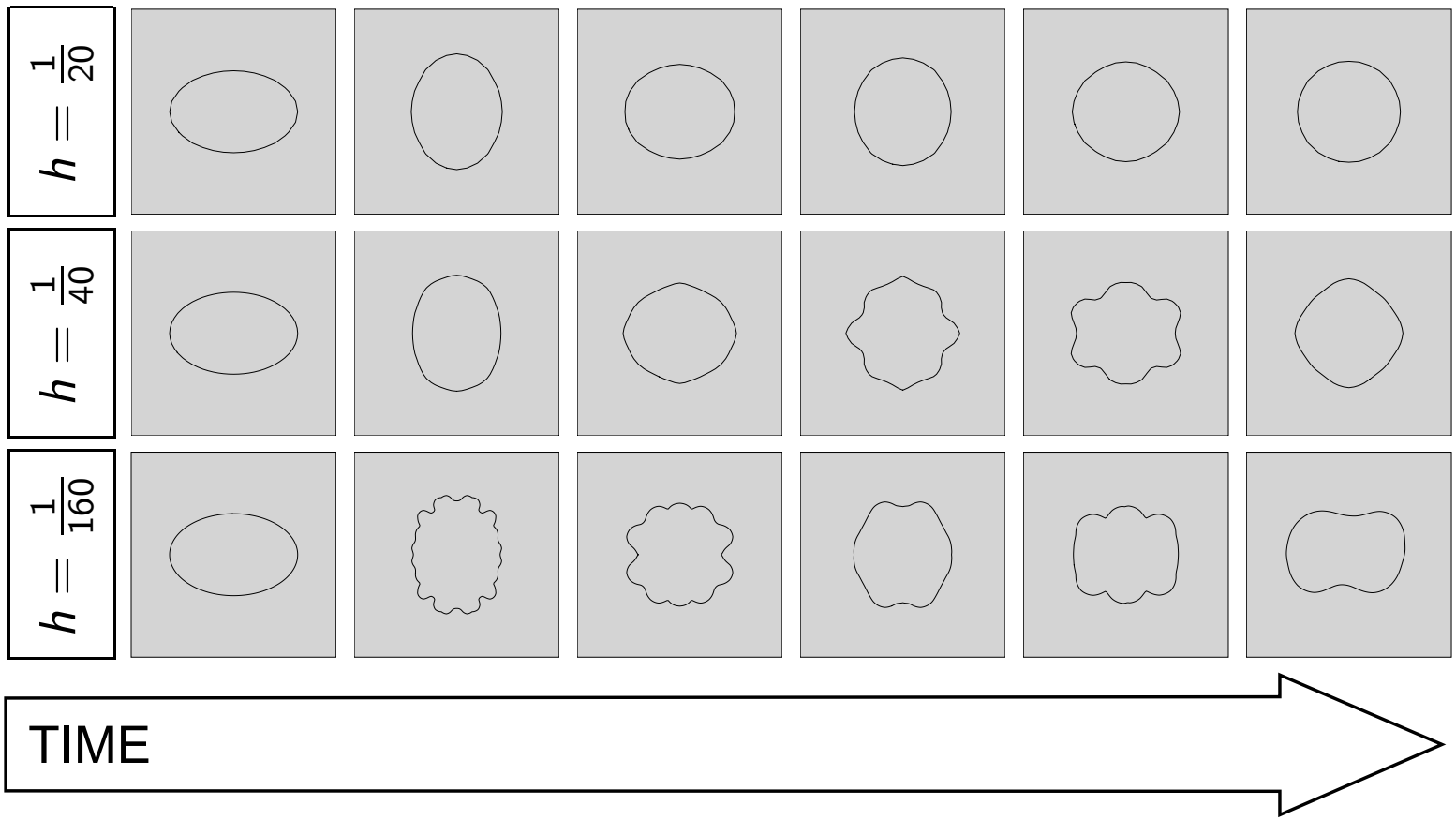


Figure: Evolution of an oscillating bubble; standard explicit CSF method.

Example, Oscillating Bubble (CSF-LBI)

Figure: Evolution of an oscillating bubble; semi-implicit CSF-LBI method.

Example, Rising Bubble (CSF)

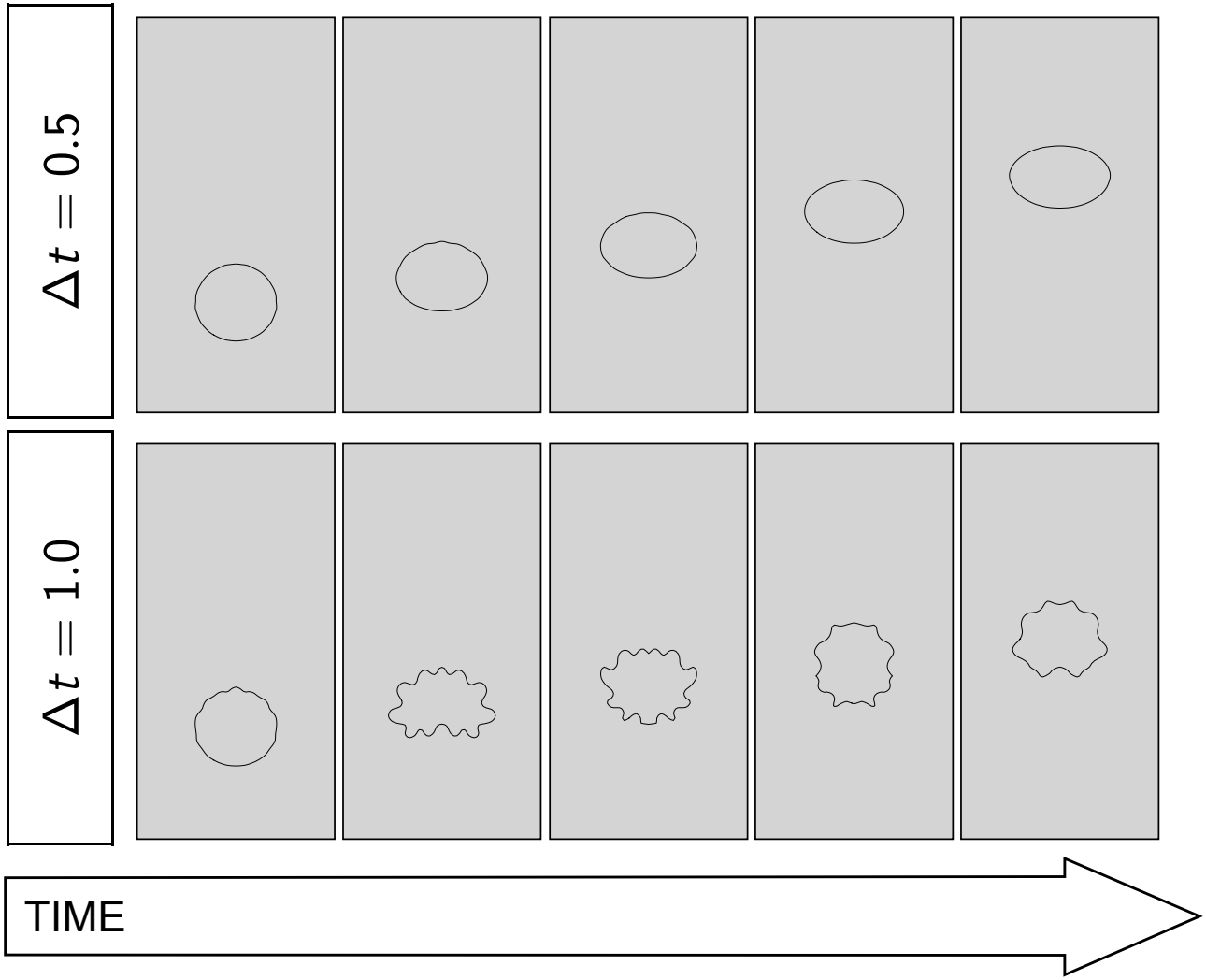


Figure: Evolution of a rising bubble with the standard explicit CSF method.

Example, Rising Bubble (CSF-LBI)

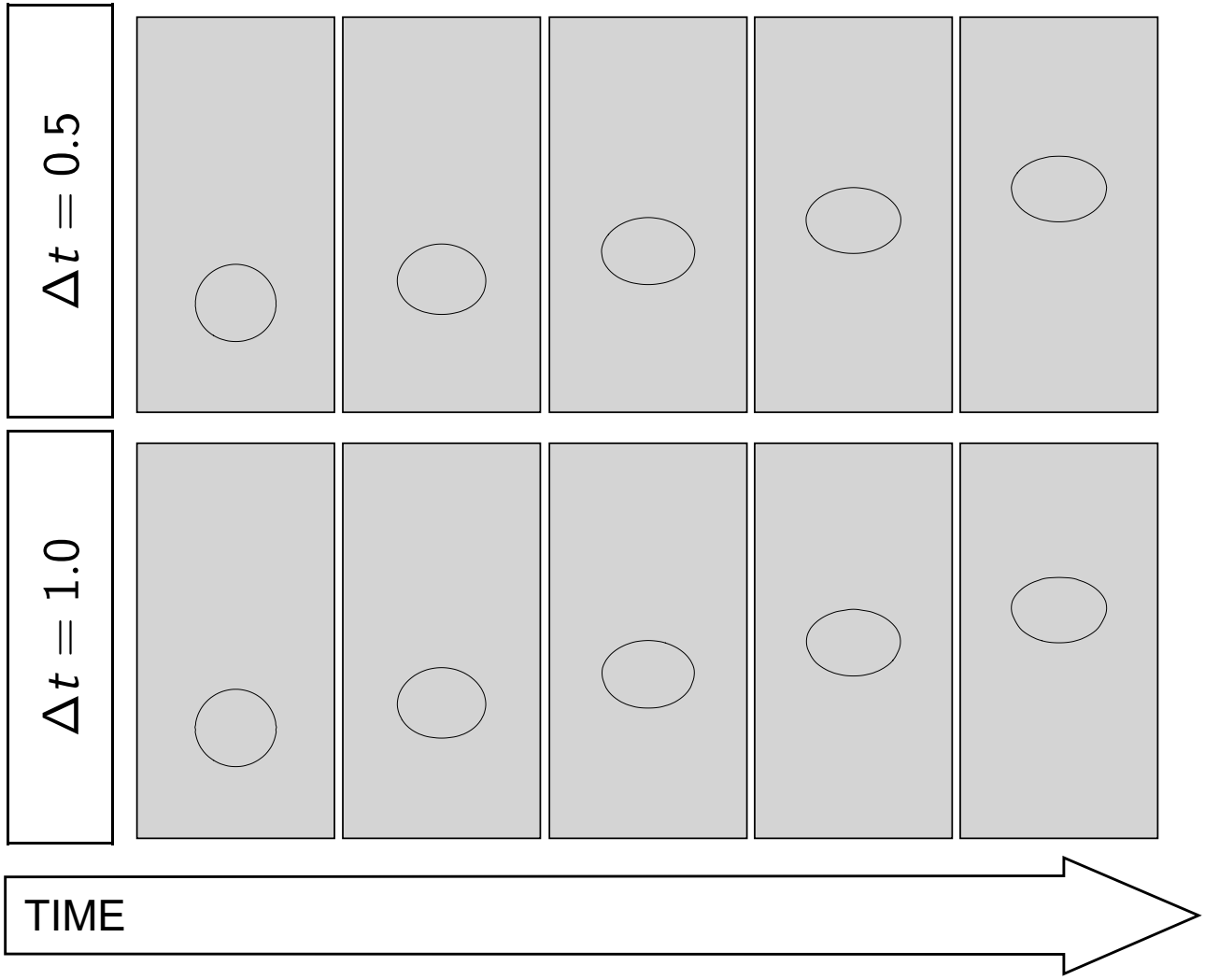


Figure: Evolution of a rising bubble with the semi-implicit CSF-LBI method.

A new implicit surface tension variant has been proposed which relaxes the capillary time step restriction imposed on explicit implementations

Additional advantages

- Fully implicit in space
- Is easily implemented when using the level set method together with finite elements
- Explicit computation of curvature not necessary
- Conceptually identical algorithm in 3D

Grid Deformation Techniques

Grid Deformation

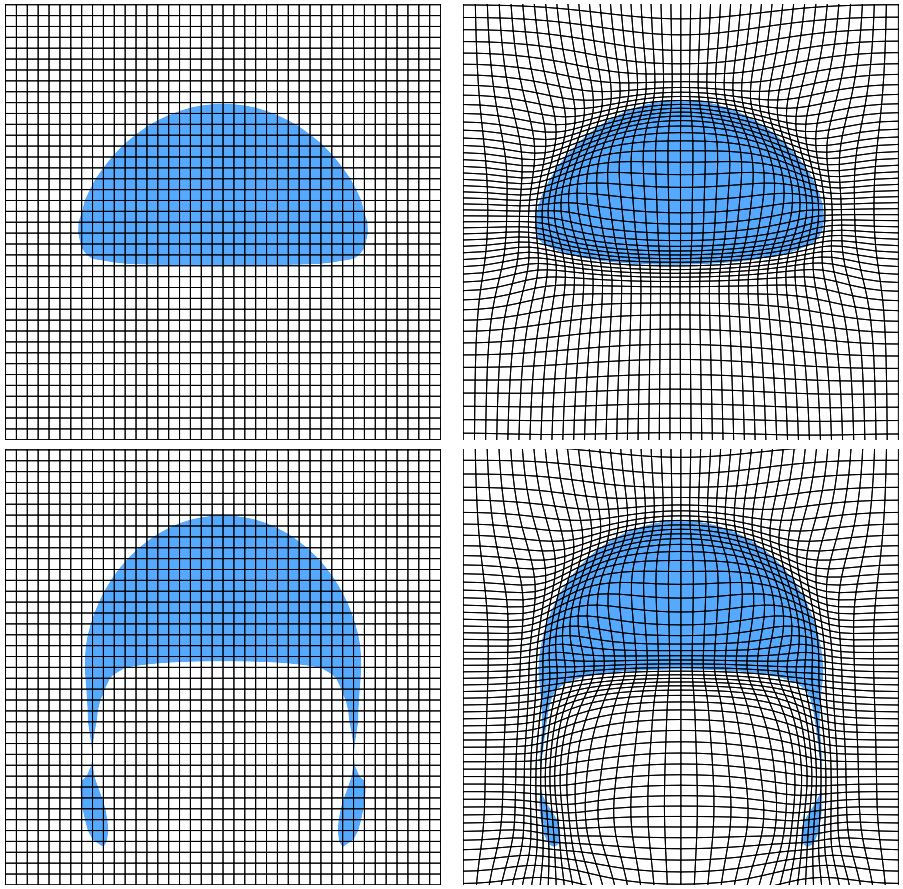
Monitor function

- In the case of interfacial flow, then the **monitor function** f can be constructed from a distance function, giving the shortest distance to the interface, and possibly also weighting the interface curvature κ

$$f = f\left(|\phi(x)|, \kappa(x)\right)$$

- This makes the *level set method* ideally suited to use as interface tracking algorithm, since both the **distance function and curvature** are defined **globally**
- Special high-performance-computing techniques and hardware (GPU computing, Cell processor, Sony PSP3) can be exploited

Grid Deformation



Static Bubble Test

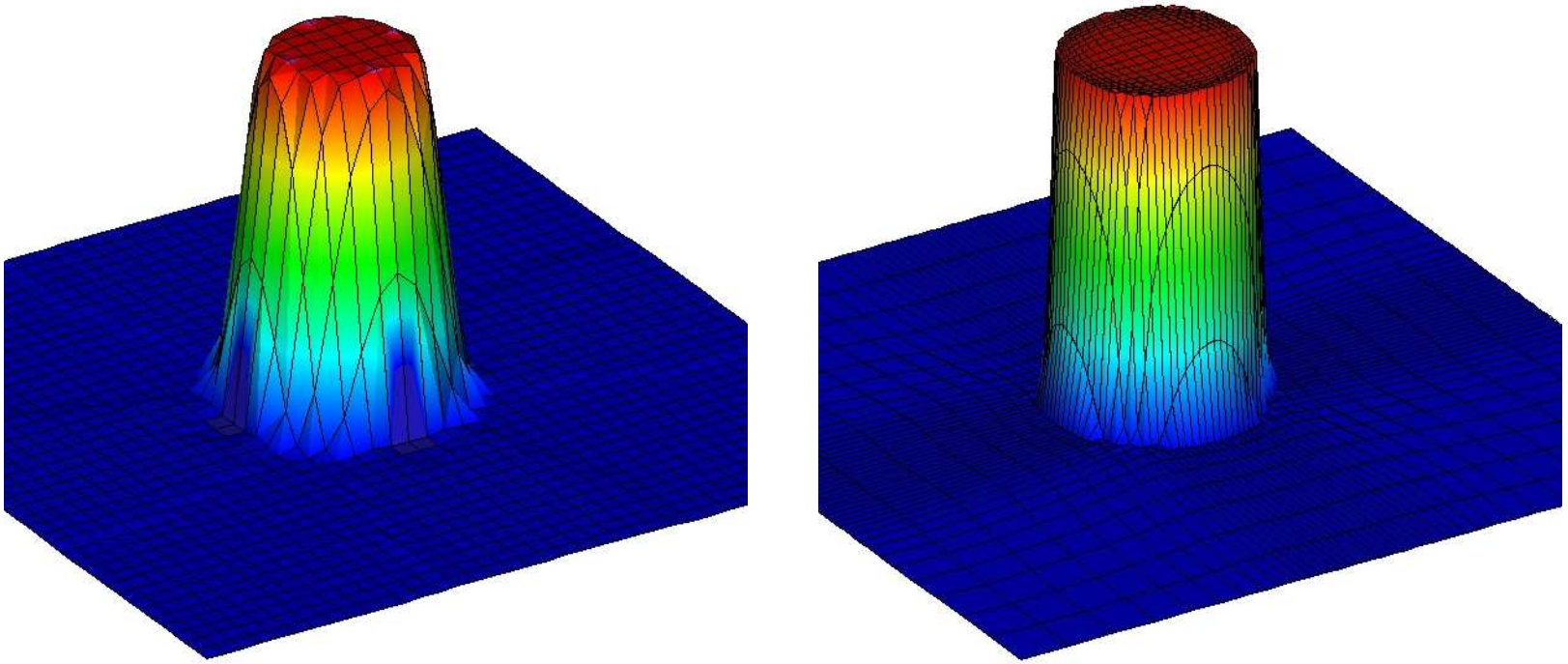


Figure: Pressure field for a static bubble with and without grid deformation

Static Bubble Test

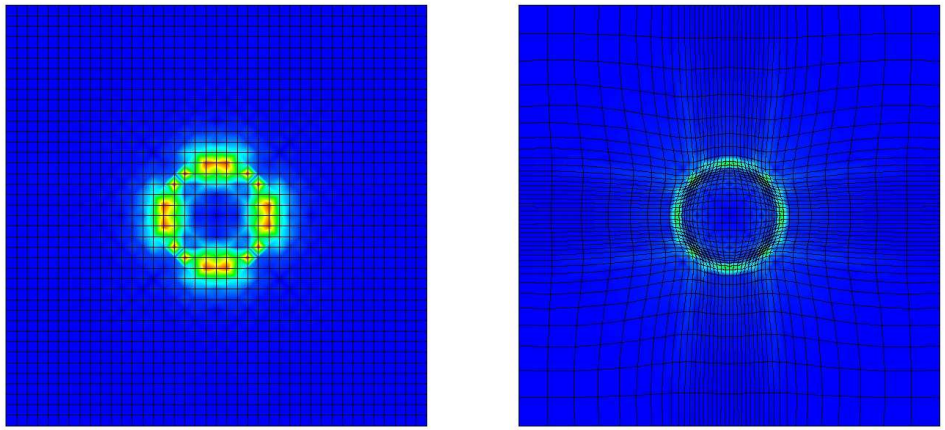


Figure: Velocity error for a static bubble with and without grid deformation

GridLevel	3	4	5	6
Tensor product grid				
U error	$3.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$5.6 \cdot 10^{-3}$
P error	5.583%	1.433%	0.212%	0.037%
Adapted grid				
U error	$2.0 \cdot 10^{-2}$	$7.3 \cdot 10^{-3}$	$3.5 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$
P error	2.969%	0.261%	0.042%	0.013%

Future Research Directions

In development...

- ALE techniques coupled with time dependent grid deformation
- Inclusion of *pressure separation* techniques to improve the velocity and pressure
- Linear high order *edge stabilization* for convection of the level set field
- $\mathbb{Q}_2\mathbb{P}_1$ finite element approximation for the NS-equations and $\mathbb{Q}_n$ for the level set equation
- Contact angle, heat transfer, and solidification effects
- 3D + benchmarking