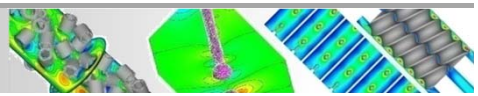
On FEM techniques for multiphase flows

Stefan Turek, Otto Mierka, Raphael Münster
Institut für Angewandte Mathematik , LS III
Technische Universität Dortmund
ture@featflow.de

http://www.mathematik.tu-dortmund.de/LS3
http://www.featflow.de



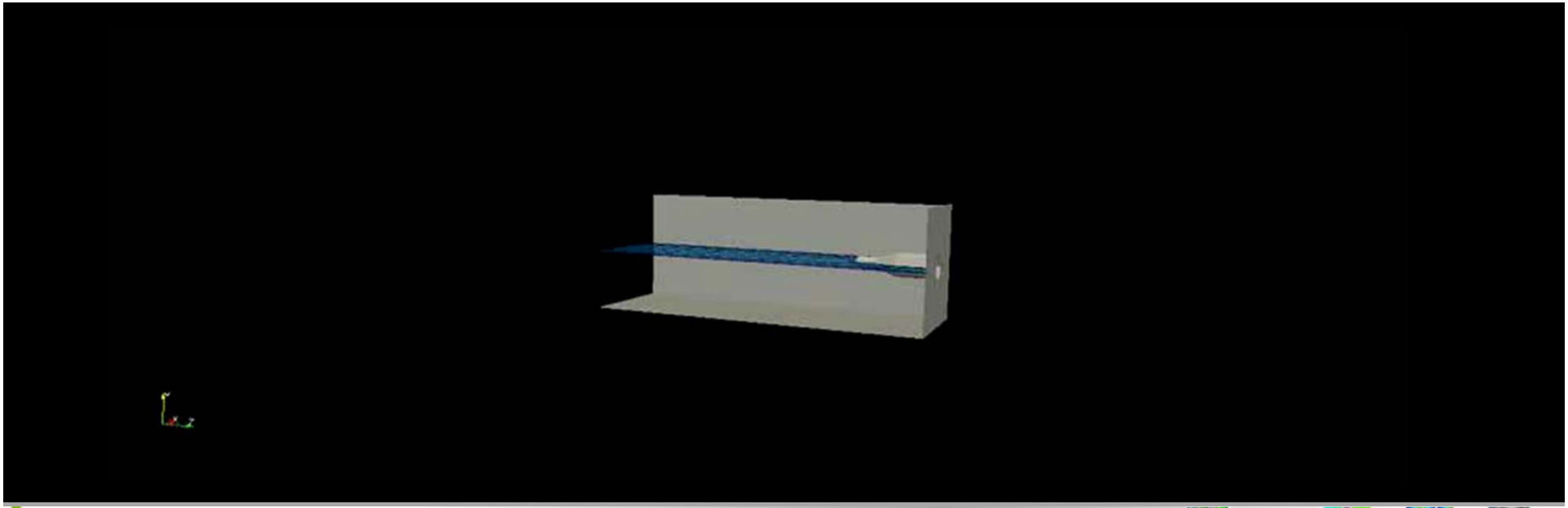
Motivation: Numerical & Algorithmic Challenges

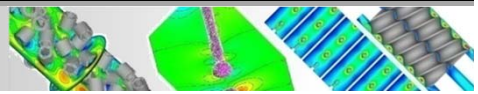
Accurate, robust, flexible and efficient simulation of multiphase problems with dynamic interfaces and complex geometries, particularly in 3D, is still a challenge!

- Mathematical Modelling of *liquid-liquid (l-l)* and *solid-liquid (s-l)* Interfaces
- Numerics / CFD Techniques
- Validation / Benchmarking
- HPC Techniques / Software

Vision: *Highly efficient, flexible and accurate „real life“ simulation tools based on modern Numerics and algorithms while exploiting modern hardware!*

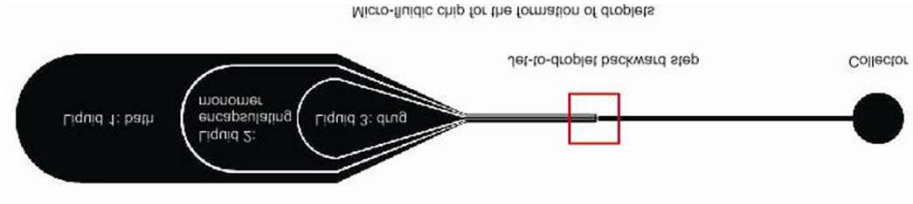
Realization: FEATFLOW

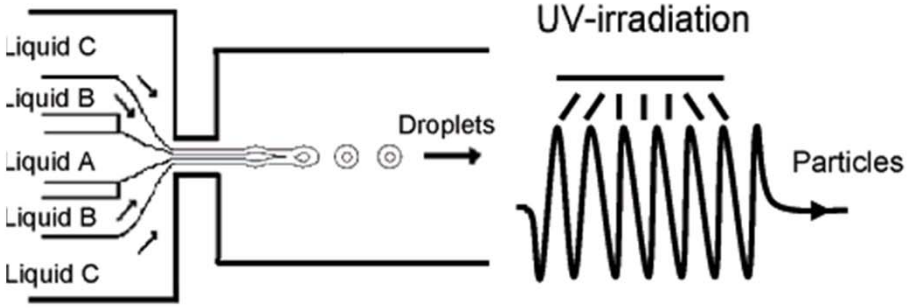


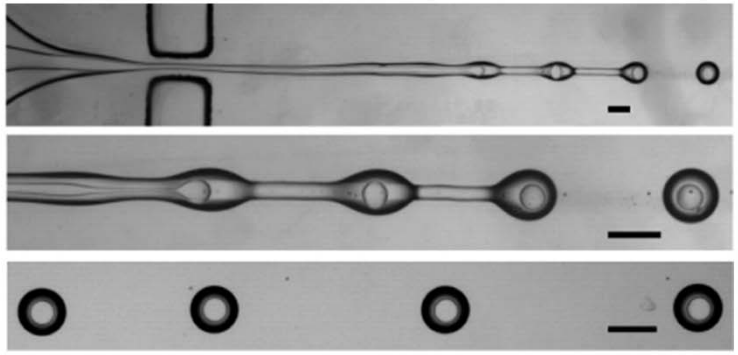


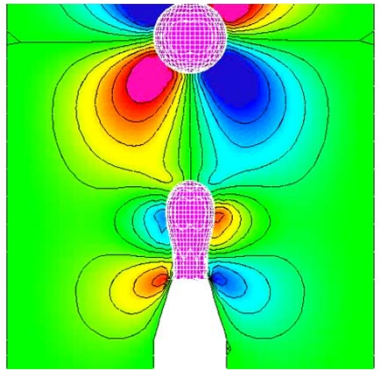
Motivation: Target Application I

- Numerical simulation of *micro-fluidic drug encapsulation ("monodisperse compound droplets")* for application in lab-on-chip and bio-medical devices
- Polymeric "bio-degradable" outer fluid with viscoelastic effects
- Optimization of chip design w.r.t. boundary conditions, flow rates, droplet size, geometry



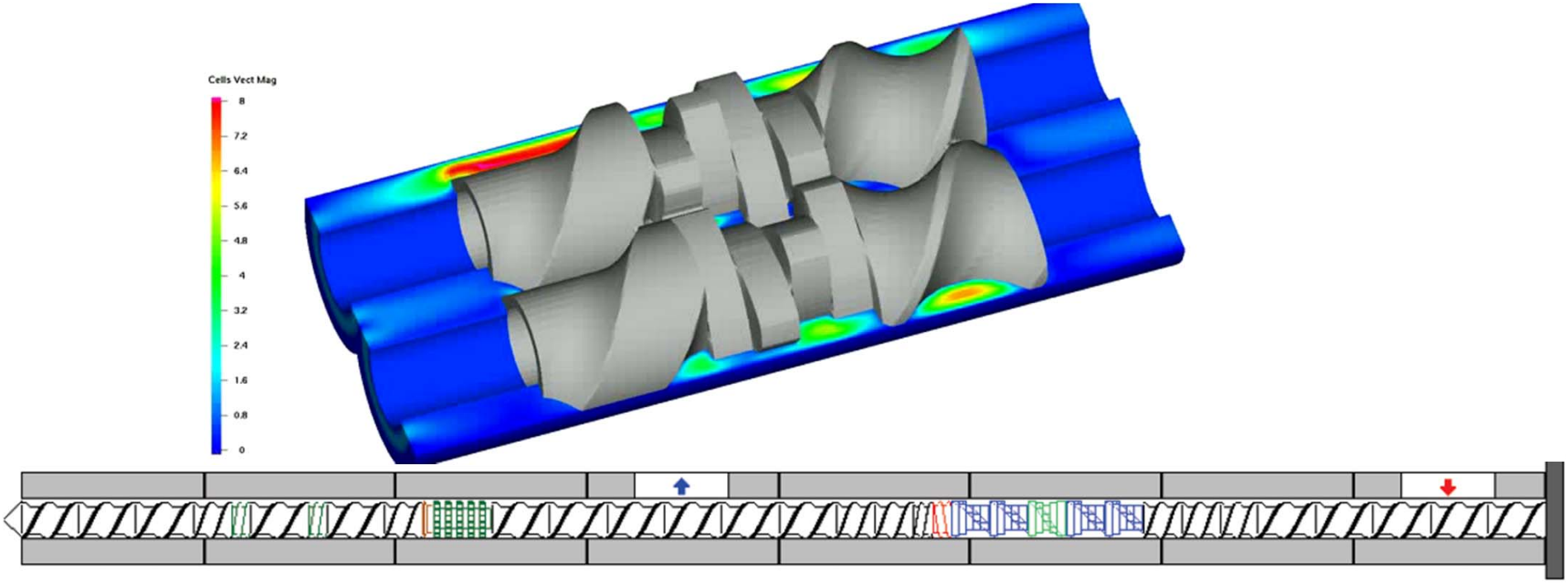






Motivation: Target Application II

- Numerical simulation of *twinscrew extruders*
- *Non-Newtonian rheological* models (shear & temperature dependent) with *non-isothermal* flow conditions (cooling from outside, heat production)
- Evaluation of torque acting on the screws, energy consumption
- Influence of local characteristics on global product quality, prediction of hotspots and maximum shear rates



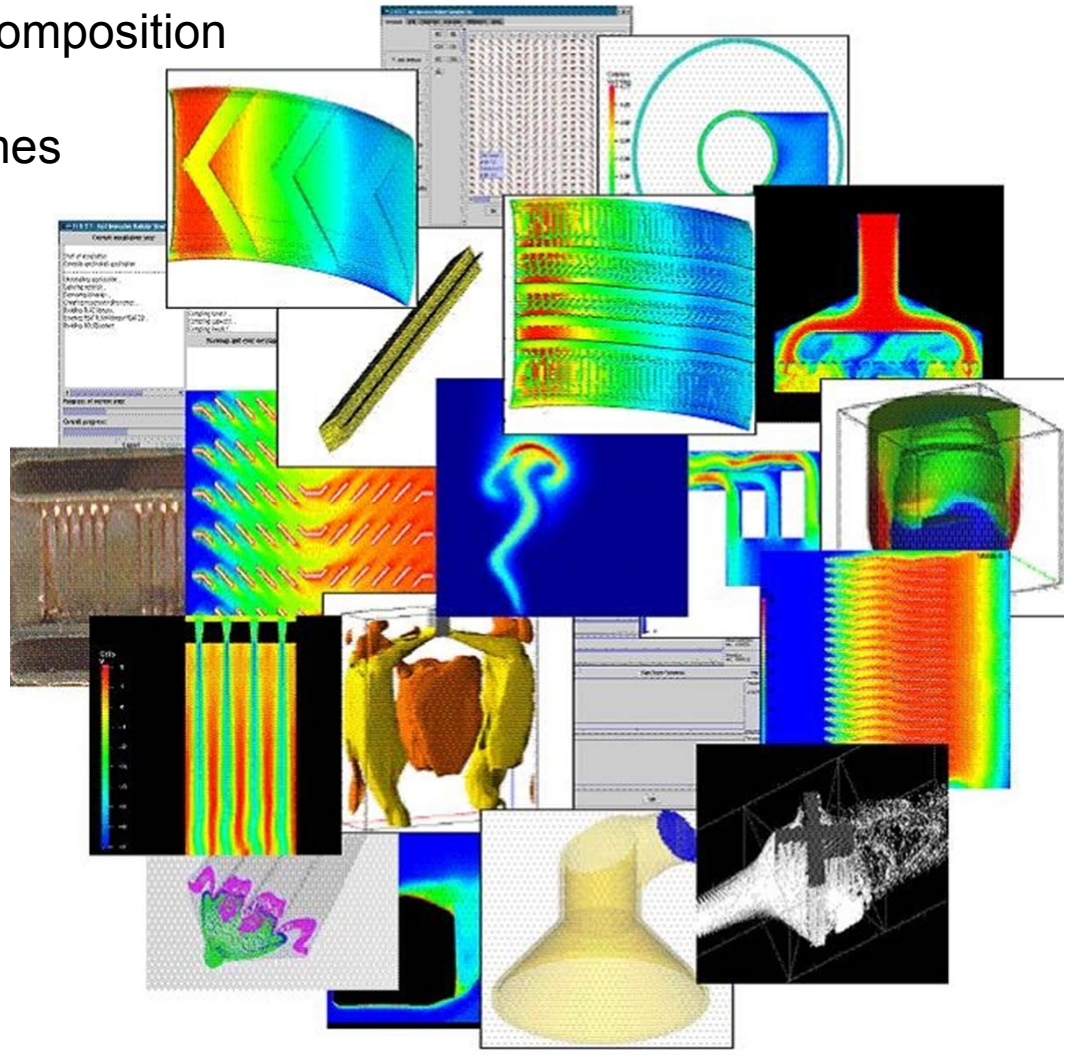
Basic Flow Solver: FeatFlow

Main features of the FEATFLOW approach:

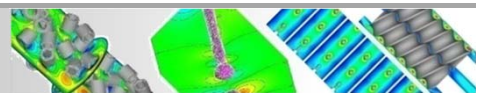
- Parallelization based on domain decomposition
- FCT & EO stabilization techniques
- High order FEM discretization schemes
- Use of unstructured meshes
- Adaptive grid deformation
- Newton-Multigrid solvers

HPC features

- Massive parallel
- GPU computing
- Open source



Hardware-oriented Numerics



Two phase flow (I-I) with resolved interphases

The incompressible Navier Stokes equations

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

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

Interphase tension force

$$\mathbf{f}_{ST}=\sigma\kappa\mathbf{n},\quad \kappa=-\nabla\cdot\mathbf{n}\quad \text{on}\quad \Gamma$$

unknown interphase location

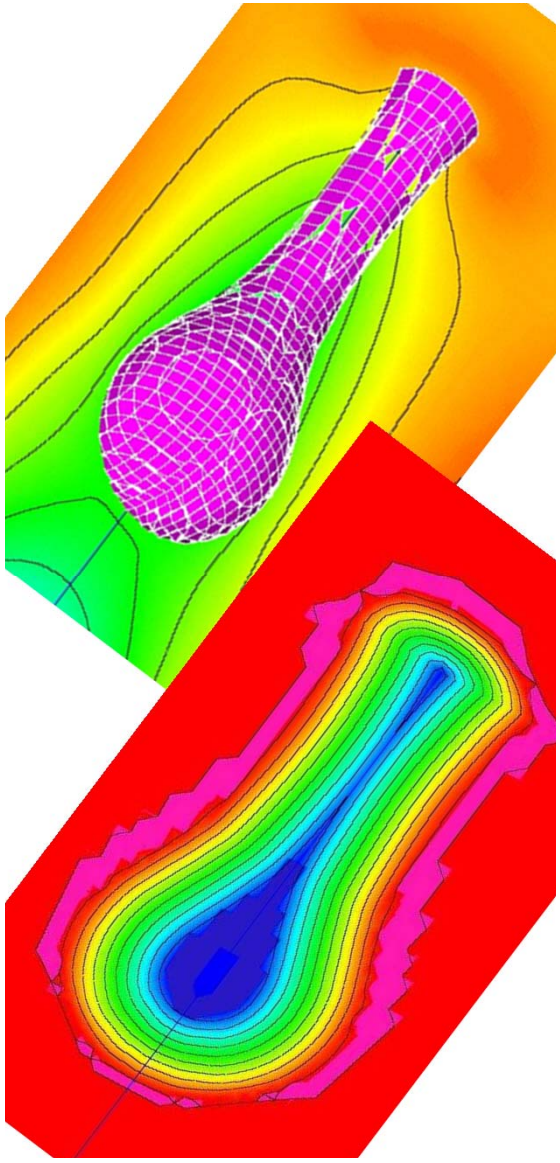
Dependency of physical quantities

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

Interphase capturing realized by **Level Set method**

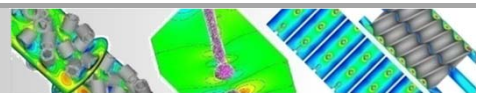
$$\frac{\partial \phi}{\partial t}+\mathbf{v}\cdot\nabla\phi=0$$

- Exact representation of the interphase
- Natural treatment of topological changes
- Provides derived geometrical quantities (**n,** κ)
- Requires reinitializion w.r.t. distance field
- Can lead to mass loss → dG(1) discretization!



Two phase flow (l-l) with resolved interphases

Problems and Challenges
• **Steep gradients** of the velocity field and of other physical quantities near the interphase (oscillations!) • **Reinitialization** (smoothed sign function, artificial movement of the interphase (→ mass loss), how often to perform?) • **Mass conservation** (during advection and reinitialization of the Level Set function) • Representation of **interphacial tension**: CSF, Line Integral, Laplace-Beltrami, Phasefield, *etc.* • **Explicit** or **implicit** treatment (→ *Capillary Time Step* restriction?)



Two phase flow (l-l) with resolved interphases

Steep changes of physical quantities:

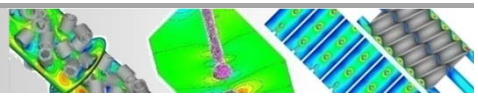
1) Elementwise averaging of the physical properties (prevents oscillations):

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

x is the volume fraction

2) Flow part: Extension of nonlinear stabilization schemes (FCT, TVD, EO-FEM) for the momentum equation for LBB stable element pairs with discontinuous pressure (Q2/P1)

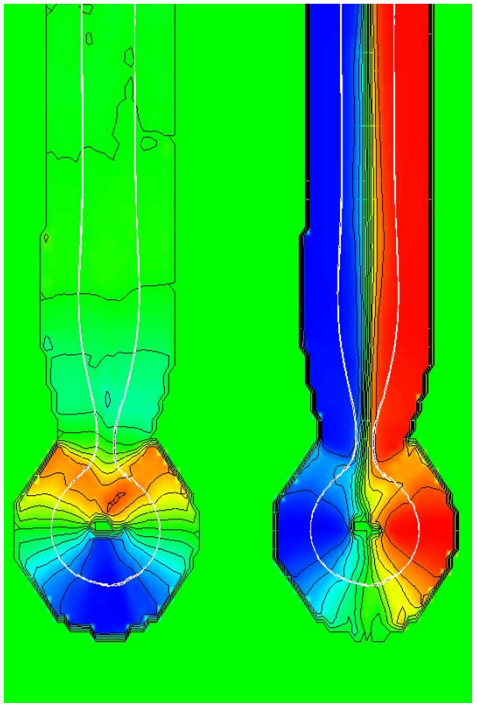
3) Interphase tracking part with DG(1)-FEM: Flux limiters satisfying LED requirements



Two phase flow (l-l) with resolved interphases

Reinitialization
- Mainly required in the vicinity of the interphase
- How often to perform?
- Which realization to implement?
- Efficient parallelization (3D!)

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

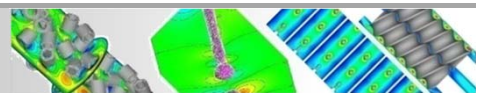


Globally defined normal vectors

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 to handle the underlying non-linearity?
- How often to perform? (expensive → steady state)



Two phase flow (I-I) with resolved interphases

Fine-tuned reinitialization

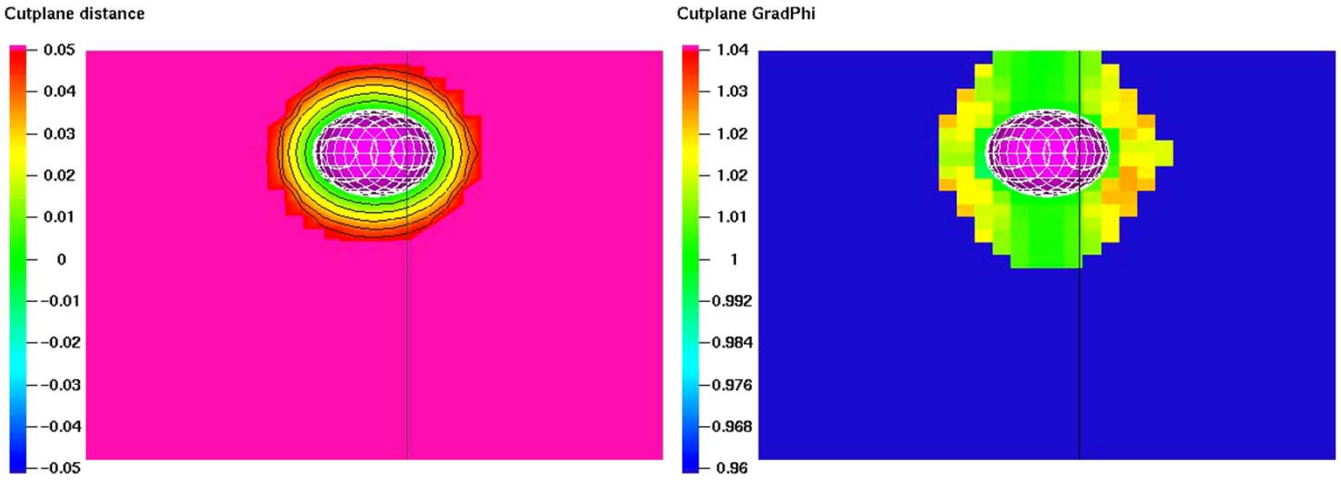
Our reinitialization is performed in combination of 2 ingredients:

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)

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



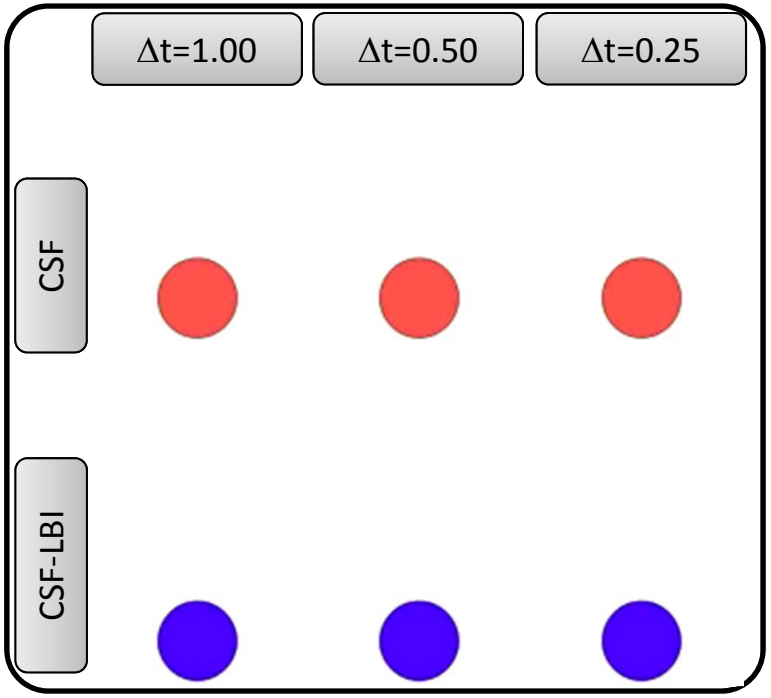
Two phase flow (I-I) with resolved interphases

Surface Tension: Semi-implicit CSF formulation based on Laplace-Beltrami

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

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

Application of the semi-implicit time integration yields $\mathbf{x}|_{\Gamma^{n+1}} = \mathbf{x}|_{\Gamma^{n}} + \Delta t\,\mathbf{u}^{n+1}$



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

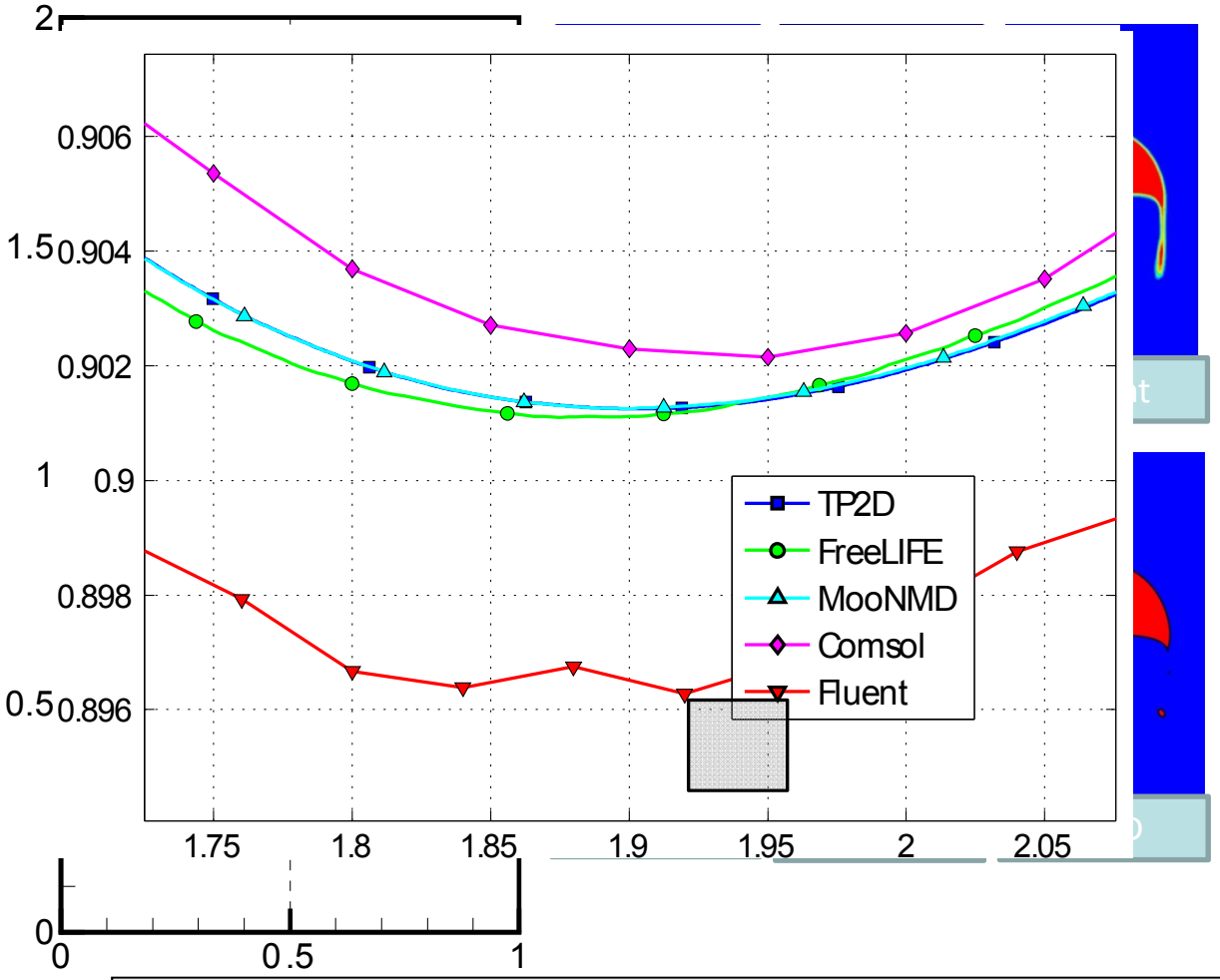
$$-\Delta t^{n+1}\int_{\Omega} \sigma\,\delta_{\varepsilon}\left(dist(\Gamma^{n},\mathbf{x})\right)\underline{\nabla}\mathbf{u}^{n+1}\cdot\underline{\nabla}\mathbf{v}\,d\mathbf{x}$$

Advantages
- Relaxes *Capillary Time Step* restriction
- „Optimal“ for FEM-Level Set approach

Benchmarking

2D Bubble Benchmarks

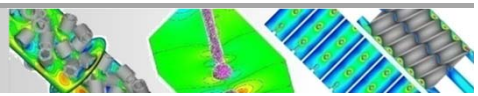
http://www.featflow.de/beta/en/benchmarks/



Benchmark quantities

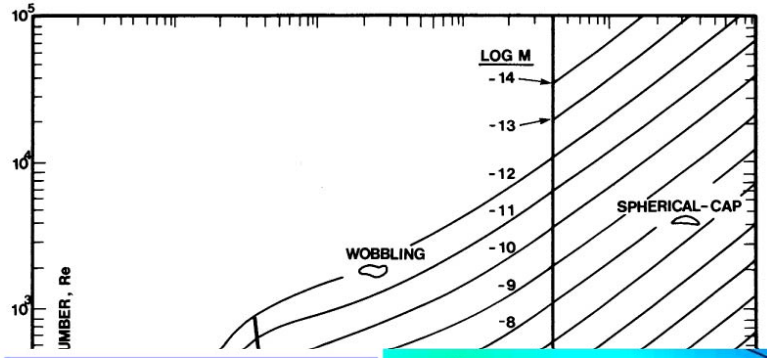
Center of mass	$\mathbf{x}_c = \frac{\int_{\Omega_2} \mathbf{x}\, dx}{\int_{\Omega_2} 1\, dx}$
Mean rise velocity	$\mathbf{U}_c = \frac{\int_{\Omega_2} \mathbf{u}\, dx}{\int_{\Omega_2} 1\, dx}$
Circularity	$\phi = \frac{P_a}{P_b} = \frac{\pi d_a}{P_b}$

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, DOI: 10.1002/fld.1934, 2009

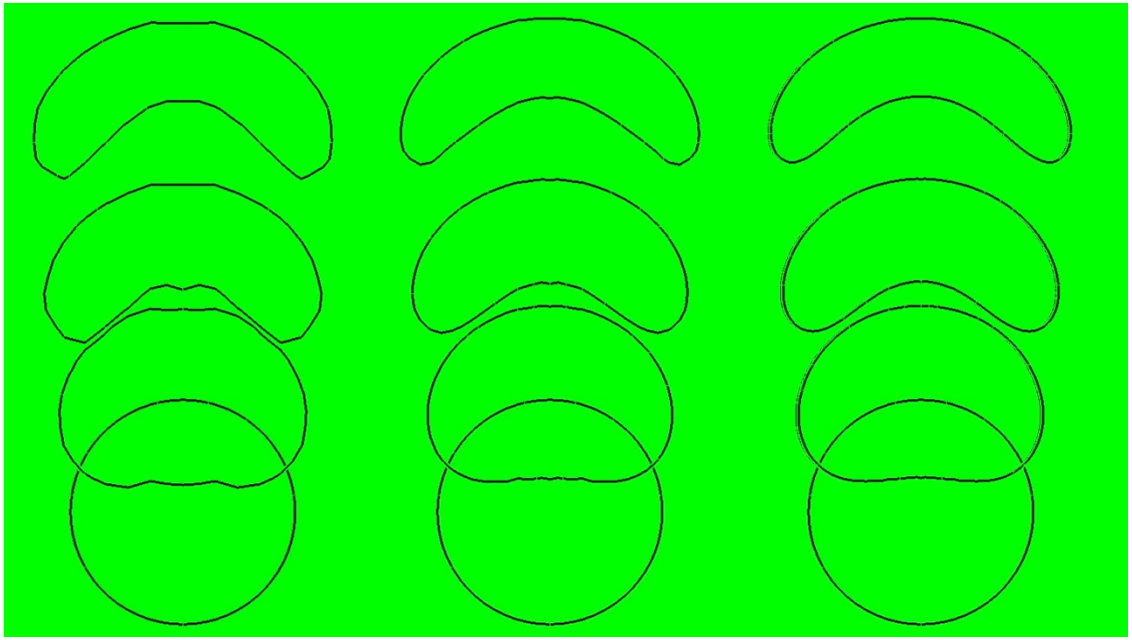


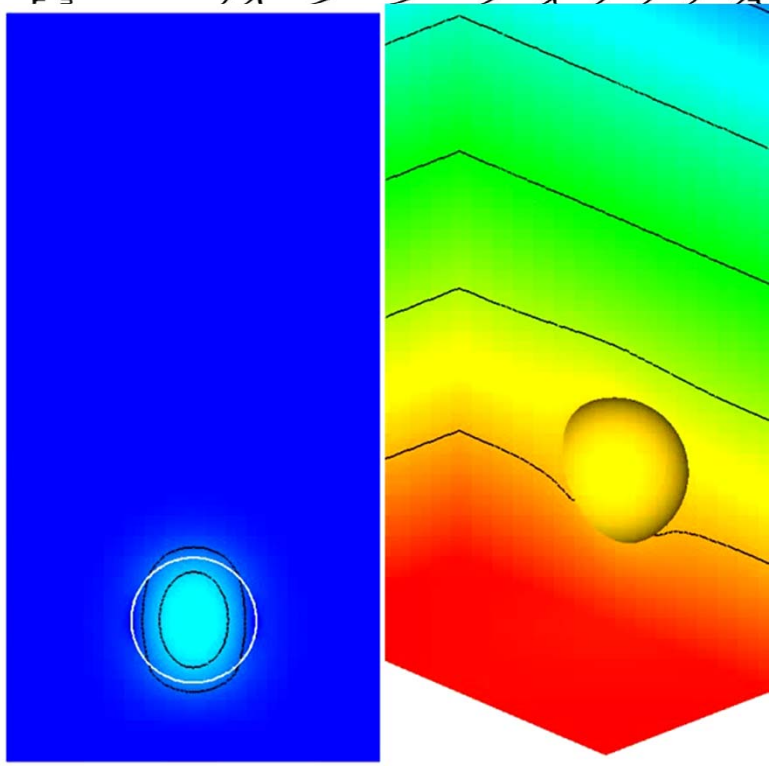
3D convergence analysis for large density jumps

Rising bubble problem for Eo = 60, Re = 34
Density jump 1:100



Level 2 Level 3 Level 4





Benchmarking with experimental results

Continuous phase:

Glucose-Water mixture

$\mu_D = 500\,mPa\,s$

$\rho_D = 972\ kg\,m^{-3}$

$\dot{V}_D = 3{,}64\,ml\,min^{-1}$

$\sigma_{CD} = 0{,}034\,N\,m^{-1}$

Silicon oil

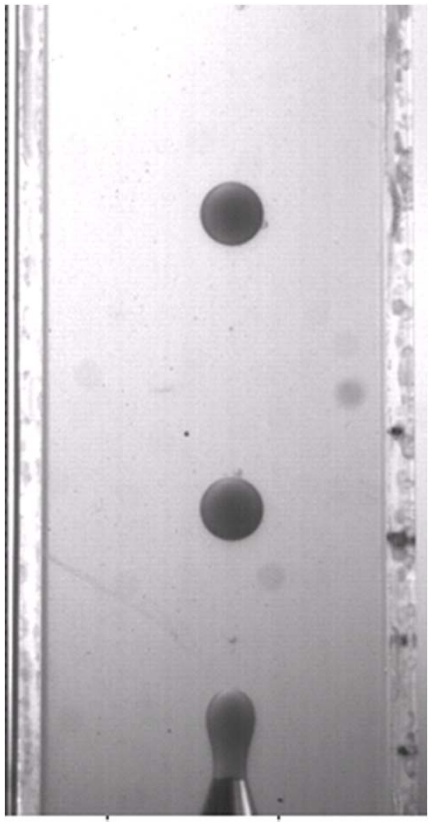
$\mu_C = 500\,mPa\,s$

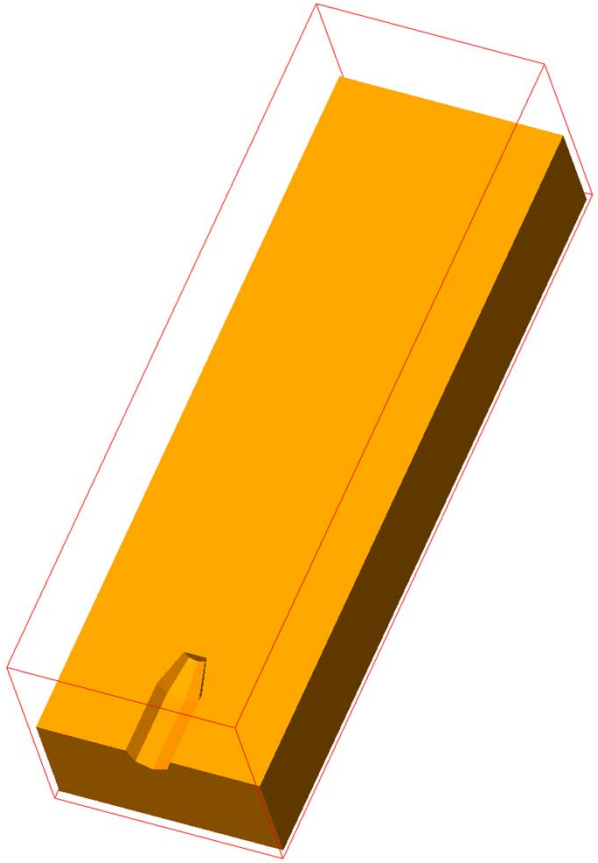
$\rho_C = 1340\ kg\,m^{-3}$

$\dot{V}_C = 99{,}04\,mlmin^{-1}$

Dispersed phase:

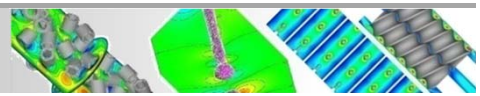
Experimental Set-up with **AG Walzel (BCI/Dortmund)**



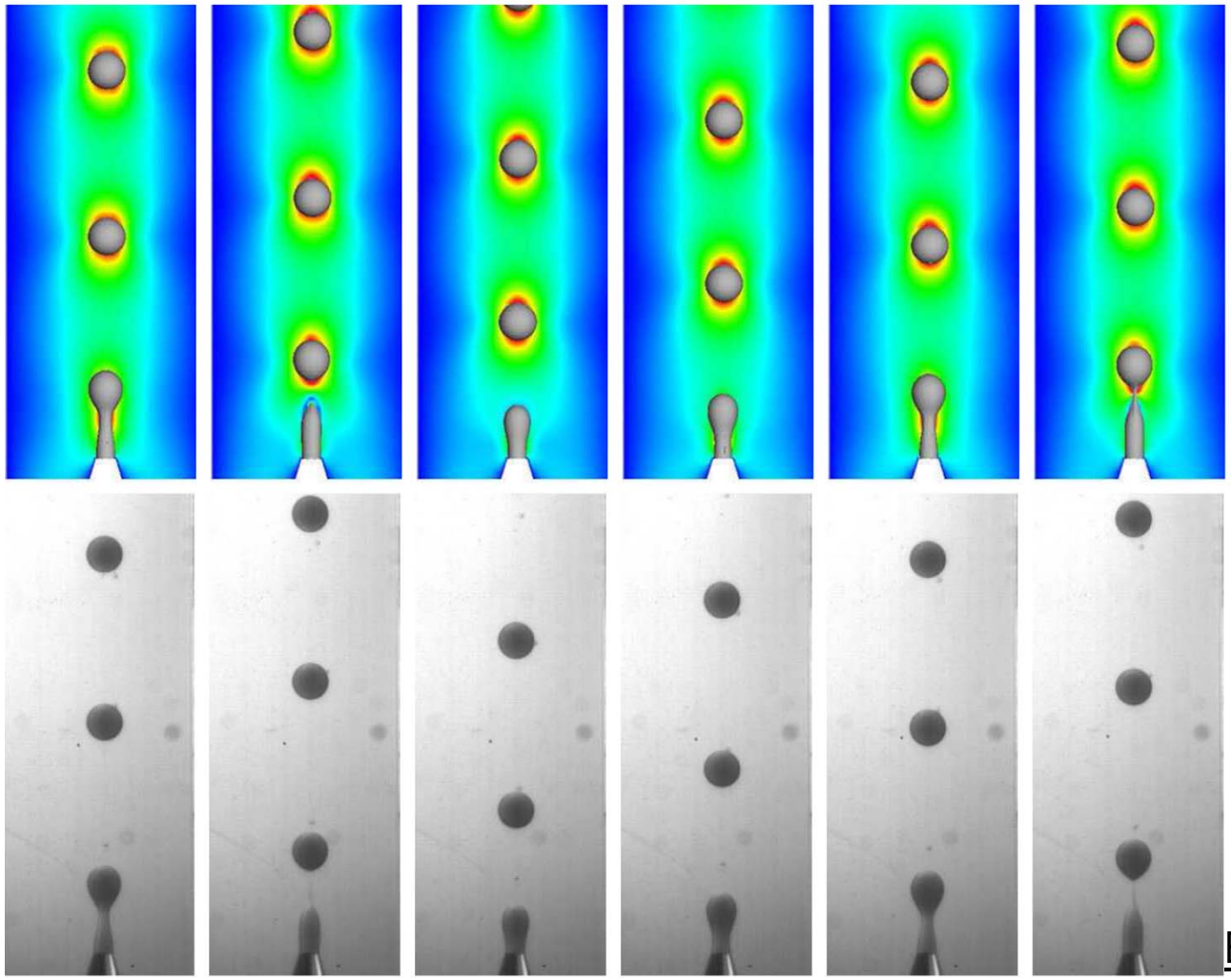


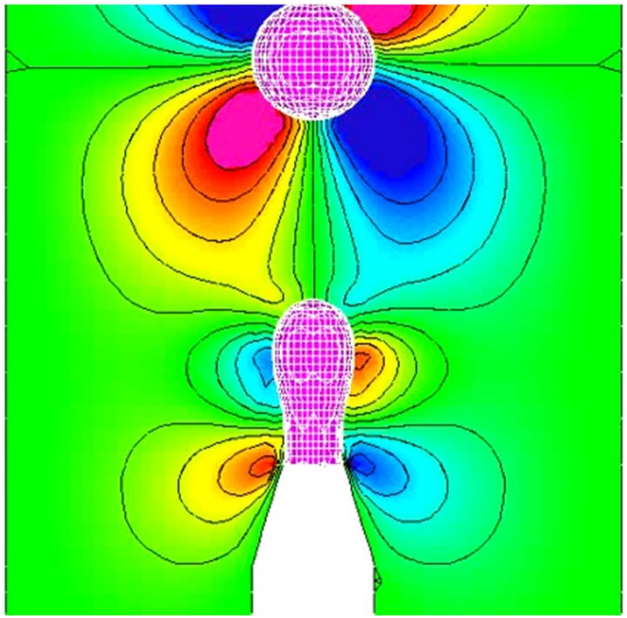
Validation parameters:

- frequency of droplet generation
- droplet size
- stream length



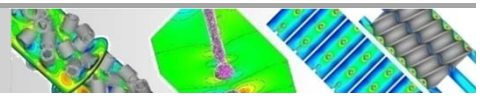
Benchmarking with experimental results





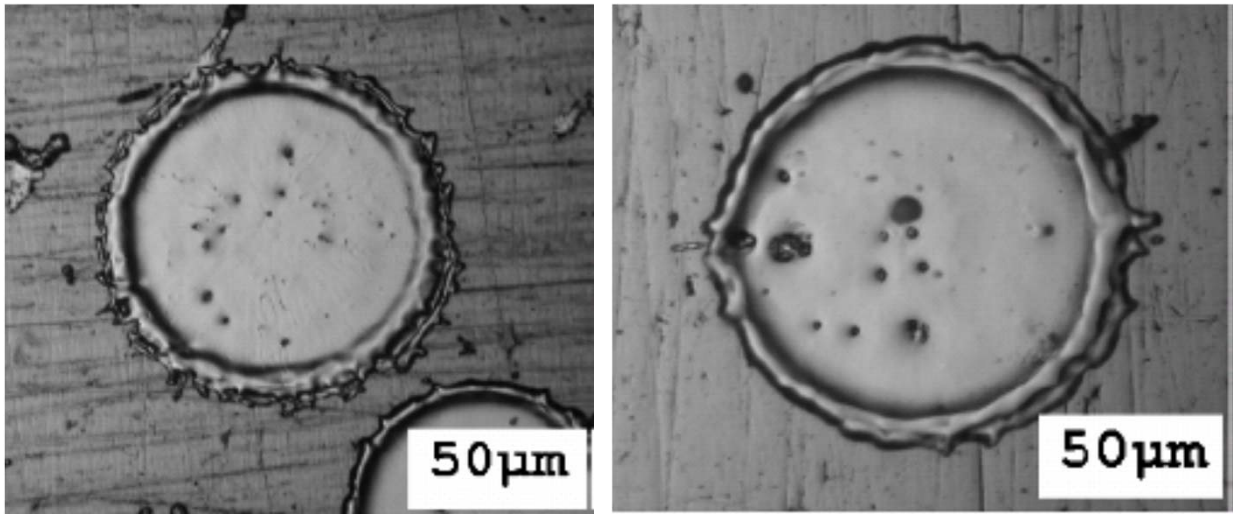
	Separation frequency [Hz]	Droplet size [dm]	Stream Length [dm]
Exp	0,58	0,062	0,122
Sim	0,6	0,058	0,102

Exp. results → Group of Prof. Walzel BCI/Dortmund

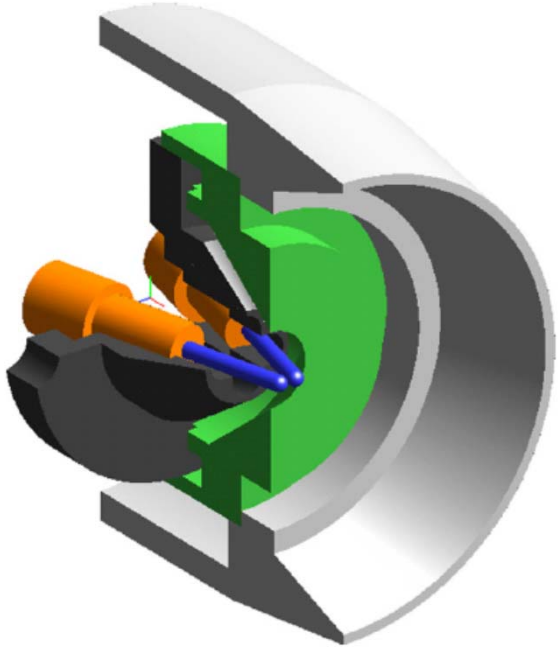


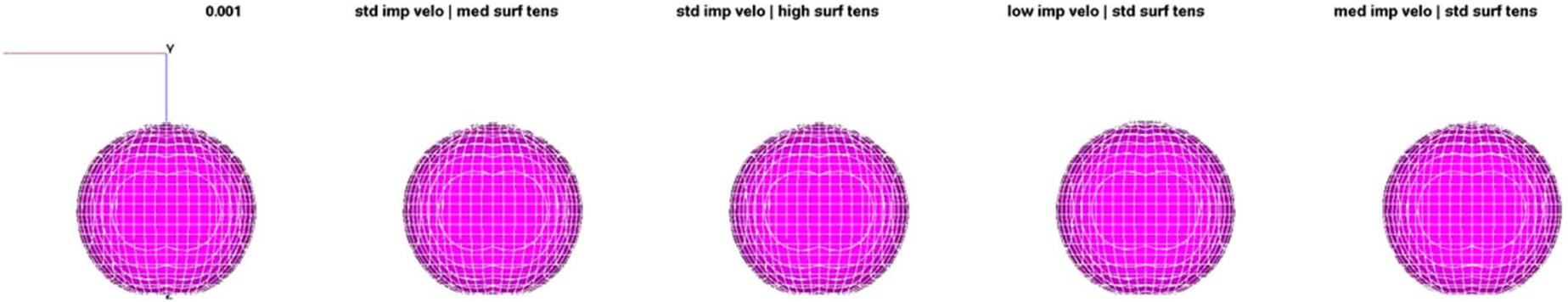
Interaction of droplets with surfaces

Pourmousa 2007

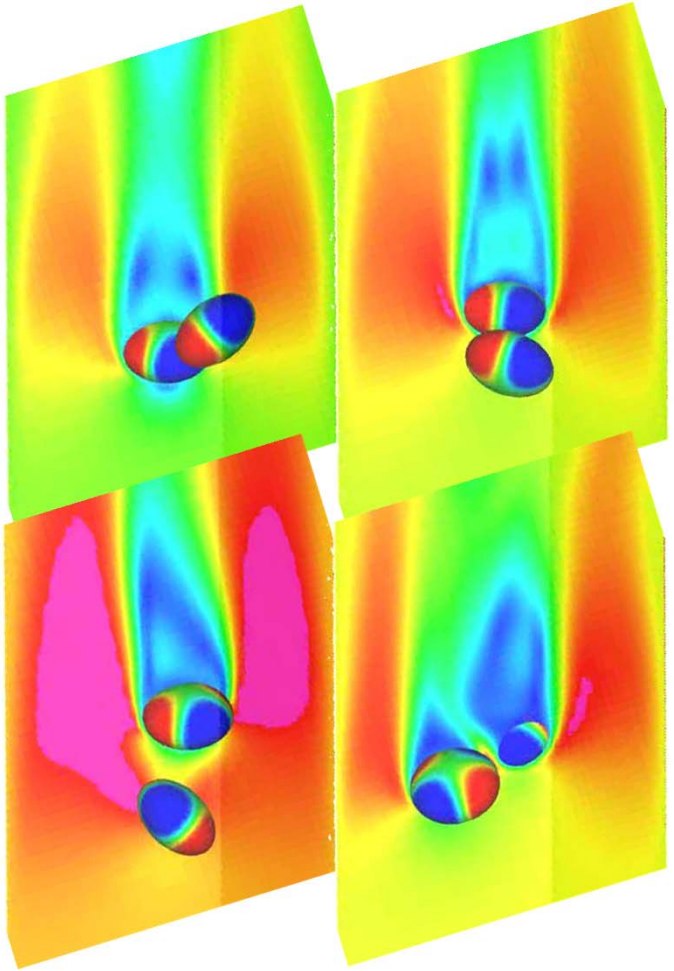


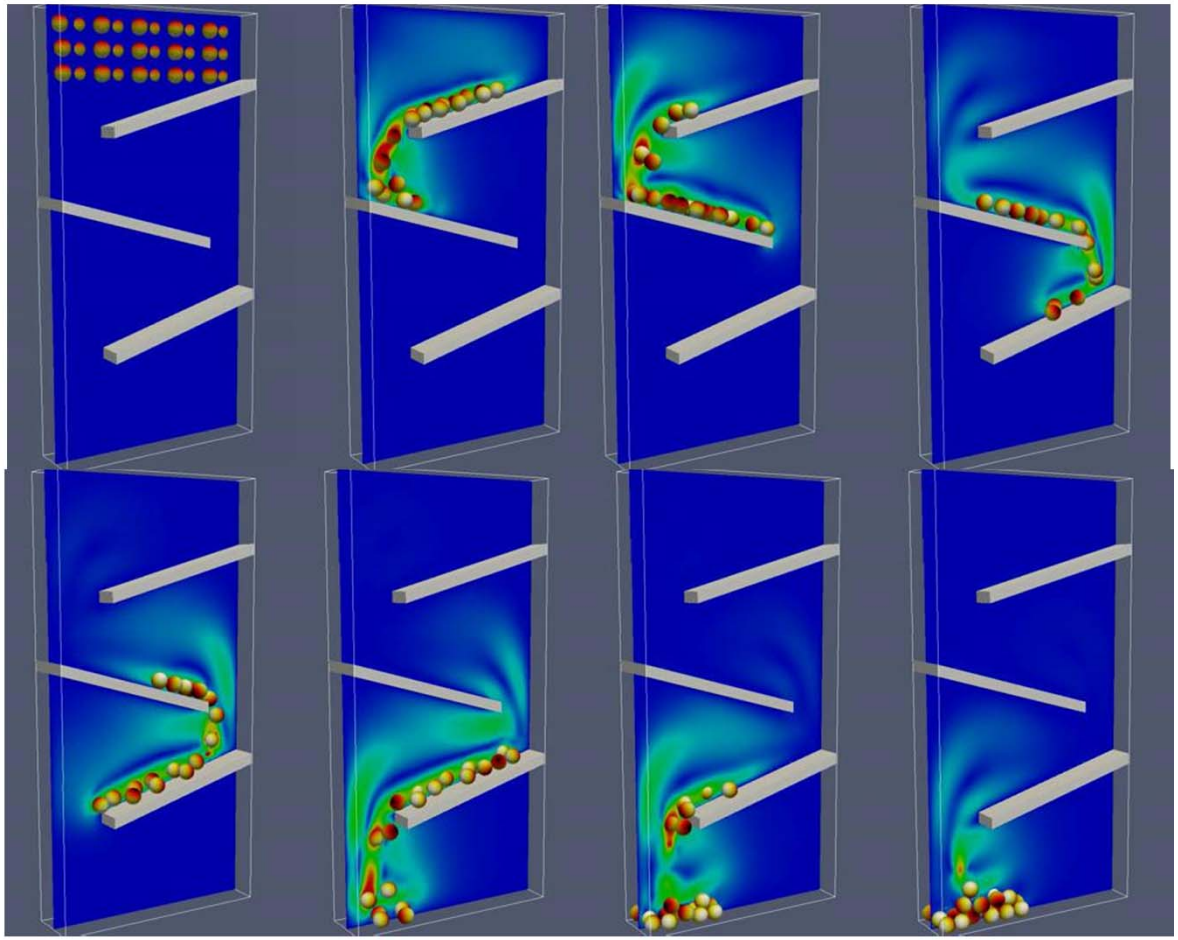
Bolot et al. 2008

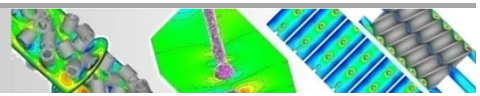




Next: Two phase flow (s-l) with resolved interphases







Two phase flow (s-l) with resolved interphases

- **Fluid** motion is governed by the **Navier-Stokes** equations
- **Particle** motion is described by **Newton-Euler** equations

$$M_p \frac{dU_p}{dt} = F_p + F_{ex,col} + (\Delta M_p) g, \qquad I_p \frac{d\omega_p}{dt} = T_p - \omega_p \times (I_p \omega_p)$$

Hydrodynamic force

Torque

$$F_p = -\int_{\Gamma_p} \sigma \cdot n_p \, d\Gamma_p \quad \longleftrightarrow \quad T_p = -\int_{\Gamma_p} (X - X_p) \times (\sigma \cdot n_p) \, d\Gamma_p$$

Postprocessing the actual flow field

Fictitious Boundary Method

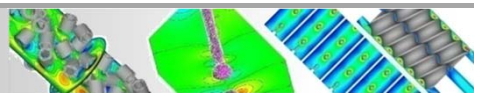
- Surface integral is replaced by volume integral
- Use of monitor function (liquid/solid)

$$\alpha_p(X) = \begin{cases} 1 & \text{for} \quad X \in \Omega_p \\ 0 & \text{for} \quad X \in \Omega_f \end{cases}$$

- Normal to particle surface vector is non-zero only at the surface of particles $n_p = \nabla \alpha_p$

$$F_p = -\int_{\Gamma_p} \sigma \cdot n_p \, d\Gamma_p = -\int_{\Omega_T} \sigma \cdot \nabla \alpha_p \, d\Omega_T$$

$$T_p = -\int_{\Gamma_p} (X - X_p) \times (\sigma \cdot n_p) \, d\Gamma_p = -\int_{\Omega_T} (X - X_p) \times (\sigma \cdot \nabla \alpha_P) \, d\Omega_T$$



Two phase flow (s-l) with resolved interphases

Fictitious Boundary Method

For computed $U_p^{n+1}, \omega_p^{n+1}$

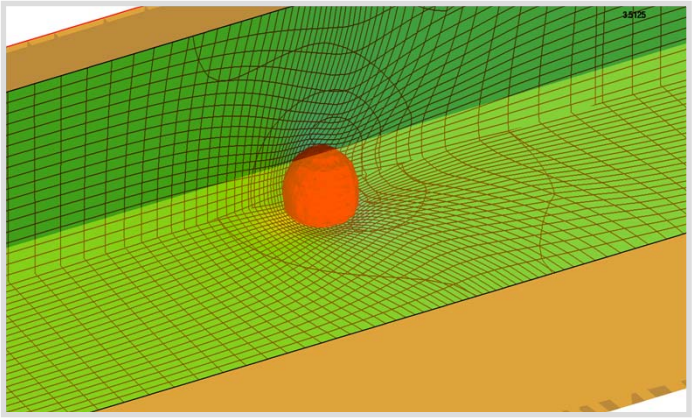
→ Position update: $\frac{dX_p}{dt} = U_{p,}$ Angle update: $\frac{d\theta_p}{dt} = \omega_p$

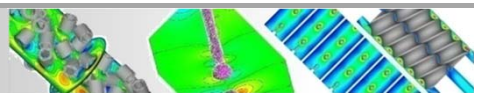
→ $X_p^{n+1}, \theta_p^{n+1}$

Velocity "boundary condition" imposed for particles:

$$u(X) = U_p + \omega_p \times (X - X_p)$$

- supports HPC concepts (no computational overhead, constant data structures, optimal load balancing)
- reduces dramatically requirements put on the computational mesh
- relatively low resolution
  - Brute force → Finer mesh resolution
  - High resolution interpolation functions
  - **Grid deformation** (+ monitor function)





Grid Deformation Method

Idea : construct transformation $\phi,\ x = \phi(\xi, t)$ with $\det \nabla \phi = f$

$\Rightarrow$ local mesh area $\approx f$

1. Compute monitor function $f(x,t) > 0,\ f \in C^1$

and

$$\int_\Omega f^{-1}(x,t)dx = |\Omega|, \quad \forall t \in [0,1]$$

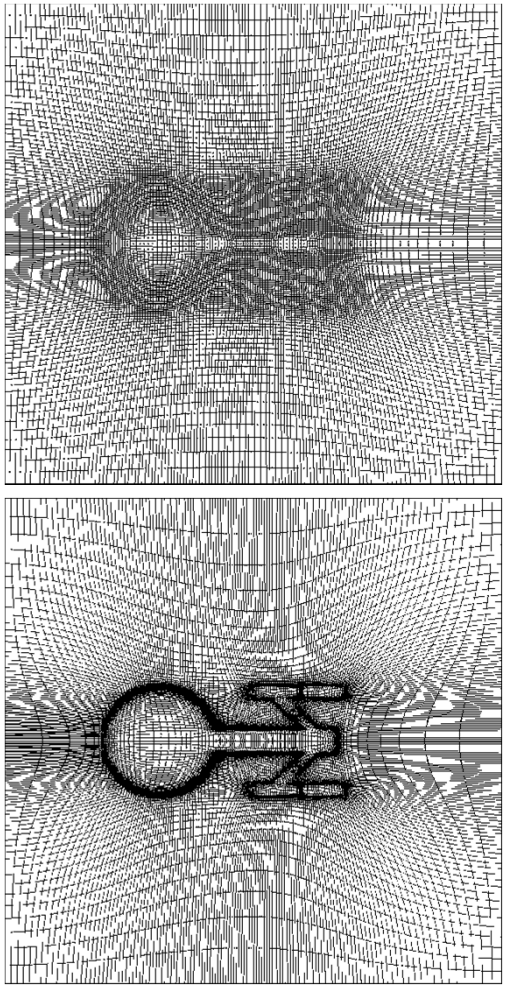
2. Solve$(t \in [0,1])$

$$\Delta v(\xi,t) = -\frac{\partial}{\partial t}\left(\frac{1}{f(\xi,t)}\right), \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0$$

3. Solve the ODE system

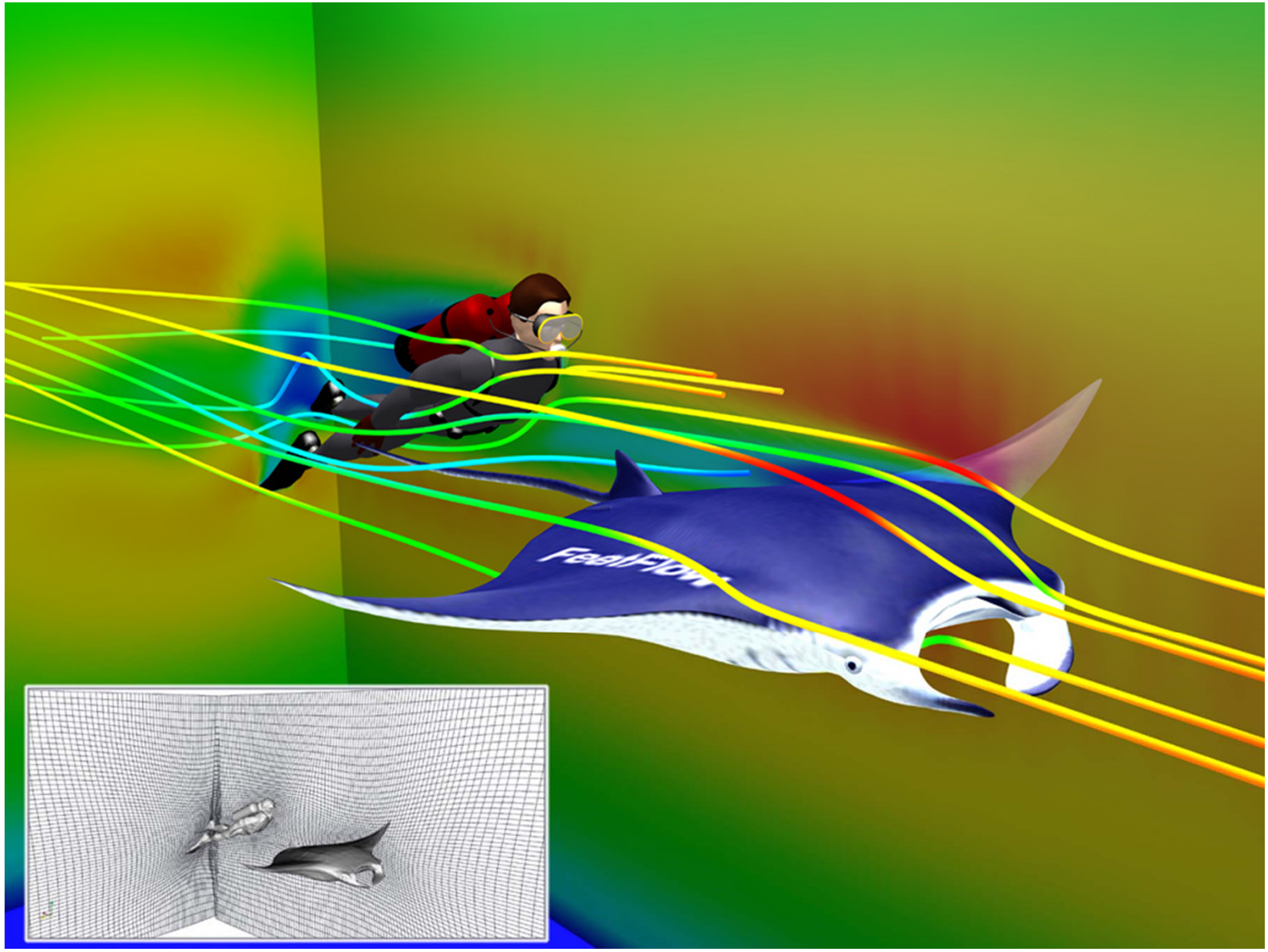
$$\frac{\partial}{\partial t}\phi(\xi,t) = f(\phi(\xi,t),t)\nabla v(\phi(\xi,t),t)$$

new grid points: $x_i = \phi(\xi_i, 1)$



Grid deformation preserves the (local) logical structure of the grid

Generalized Tensorproduct Meshes

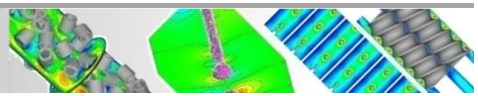


Operator-Splitting Approach

The algorithm for $t^n \rightarrow t^{n+1}$ consists of the following 4 substeps

1. Fluid velocity and pressure: $NSE\left(u_f^{n+1}, p^{n+1}\right) = BC\left(\Omega_p^n, u_p^n\right)$
2. Calculate hydrodynamic forces: F_p^{n+1}
3. Calculate velocity of particles: $u_p^{n+1} = g\left(F_p^{n+1}\right)$ (collision model)
4. Update position of particles: $\Omega_p^{n+1} = f\left(u_p^{n+1}\right)$
5. Align new mesh

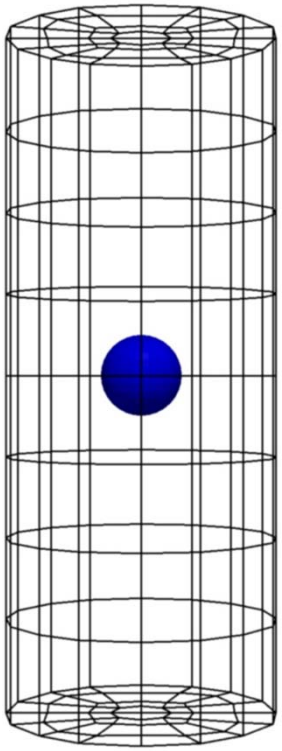
→ Required: efficient calculation of hydrodynamic forces
→ Required: efficient treatment of particle interaction (?)
→ Required: fast (nonstationary) Navier-Stokes solvers (!)

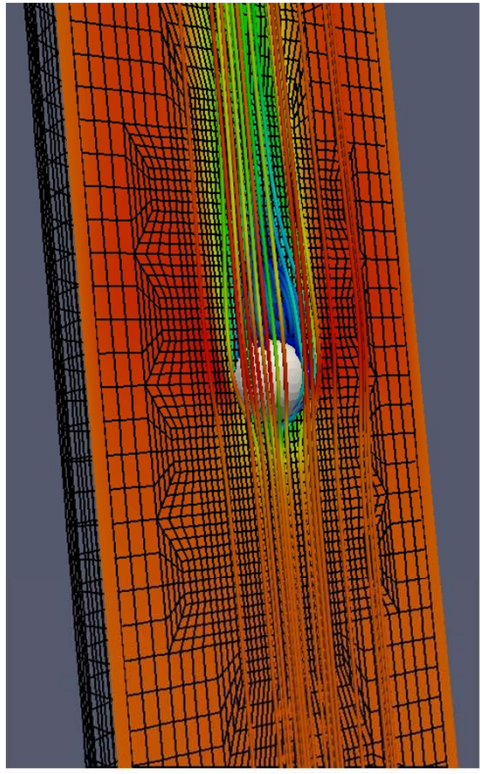


Benchmarking and Validation

Free fall of particles:

- Terminal velocity
- Different physical parameters
- Different geometrical parameters





Münster, R.; Mierka, O.; Turek, S.: Finite Element fictitious boundary methods (FEM-FBM) for 3D particulate flow, IJNMF, 2010, accepted

$d_s = 0.3, \quad \rho_s = 1.14$

ν	$U_{featflow}$	U_{exp}	Relative error (%)
0.02	5.885	6.283	6.33
0.05	4.133	3.972	4.05
0.1	2.588	2.426	6.66
0.2	1.492	1.401	6.50

$d_s = 0.2, \quad \rho_s = 1.14$

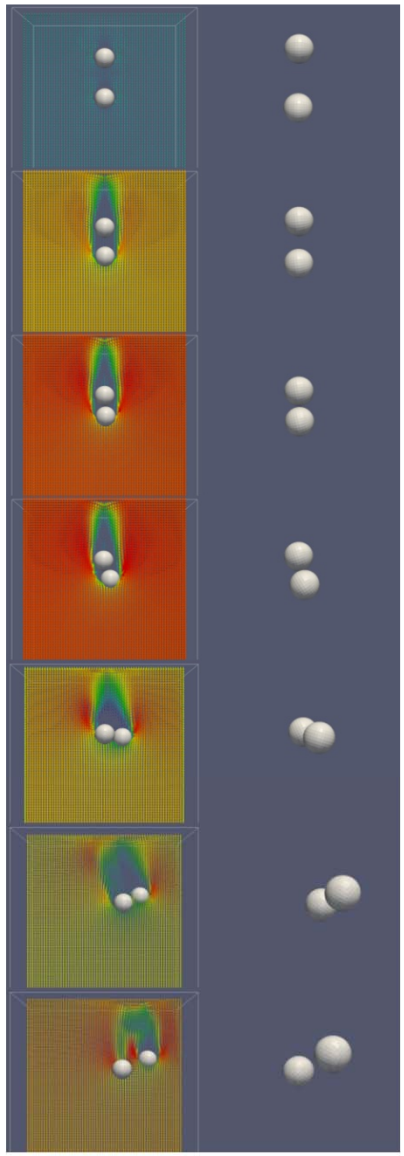
ν	$U_{featflow}$	U_{exp}	Relative error (%)
0.02	4.370	4.334	0.83
0.05	2.699	2.489	8.44
0.1	1.649	1.552	6.25
0.2	0.946	0.870	8.74

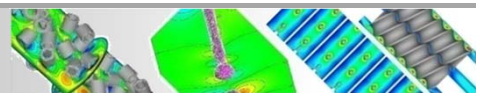
$d_s = 0.3, \quad \rho_s = 1.02$

ν	$U_{featflow}$	U_{exp}	Relative error (%)
0.01	2.167	2.107	2.84
0.02	1.495	1.436	4.11
0.05	0.809	0.749	8.01
0.1	0.402	0.404	0.44
0.2	0.218	0.216	1.02

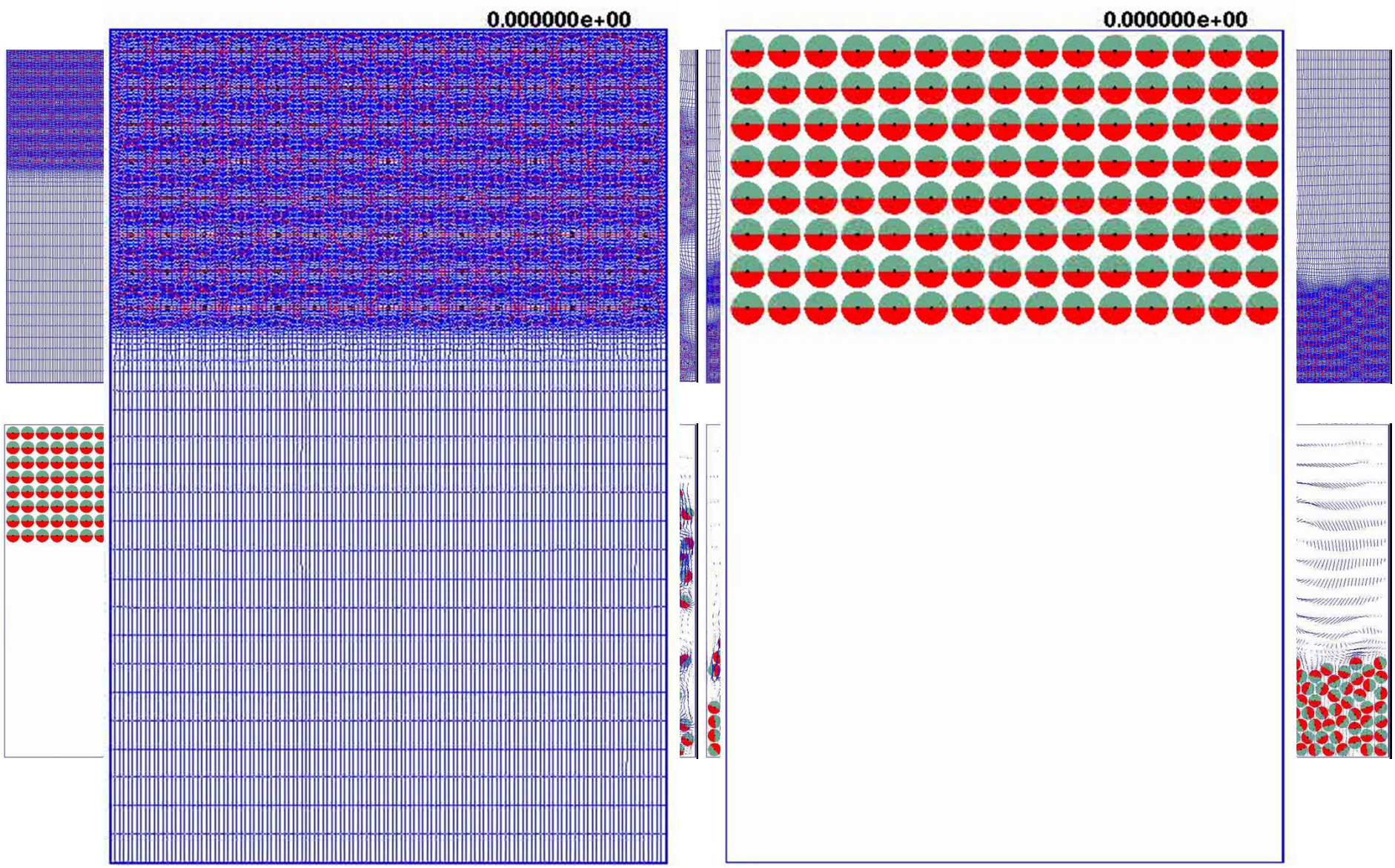
$d_s = 0.2, \quad \rho_s = 1.02$

ν	$U_{featflow}$	U_{exp}	Relative error (%)
0.01	1.4660	1.4110	3.90
0.02	0.9998	0.9129	9.52
0.05	0.4917	0.4603	6.82
0.1	0.2637	0.2571	2.57
0.2	0.1335	0.1317	1.37



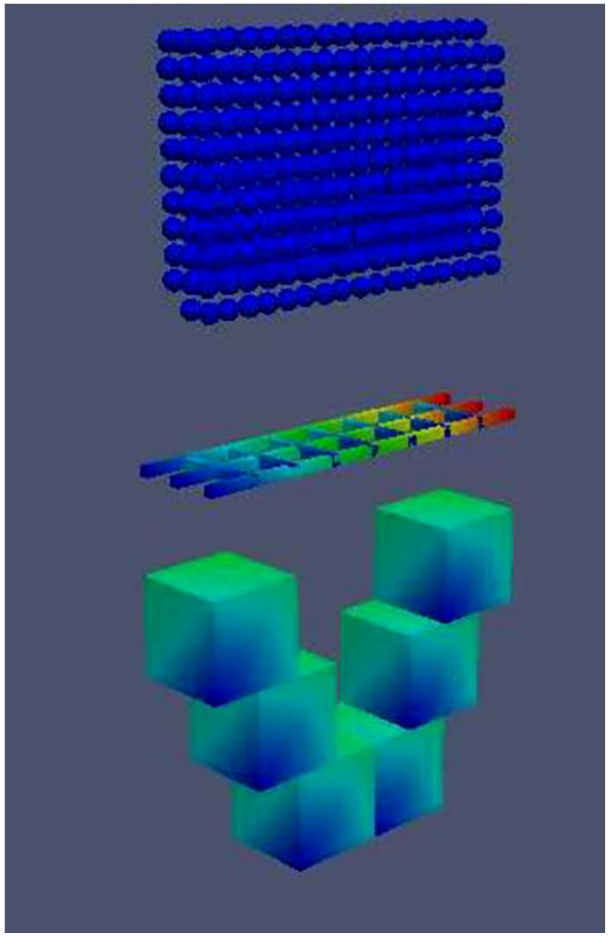


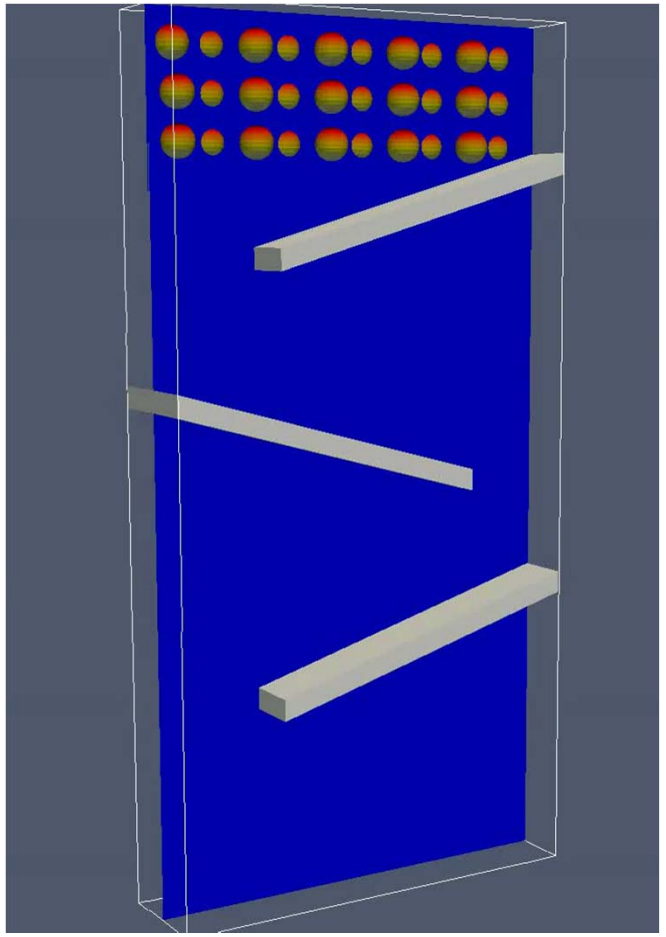
Sedimentation of many Particles



3D simulations with more complex shapes

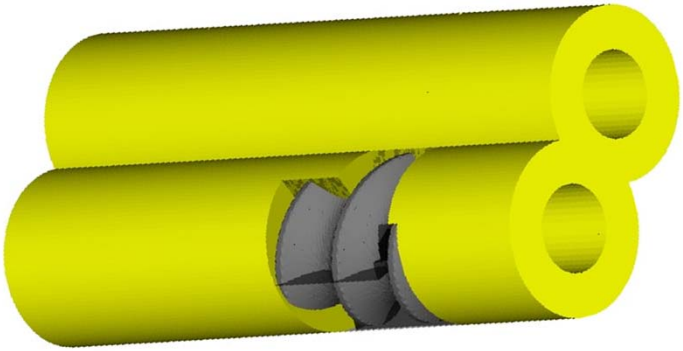
Sedimentation of particles in a complex domain

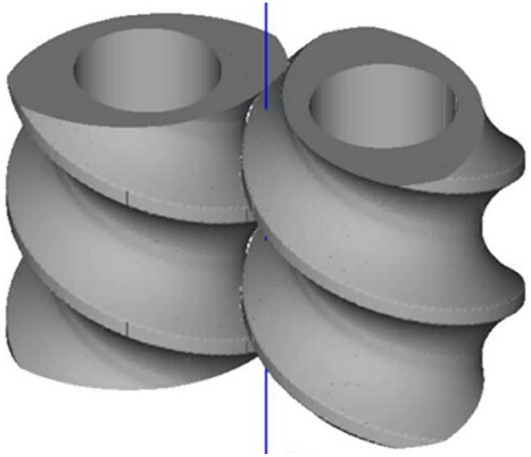




Twinscrew Flow Simulation with FeatFlow

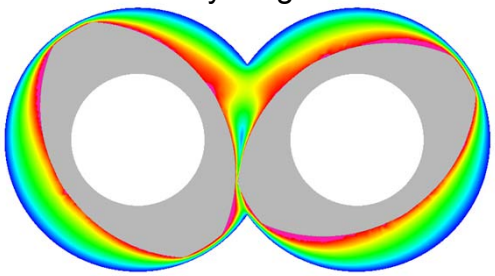
Geometrical representation of the twinscrews → Fictitious Boundary Method



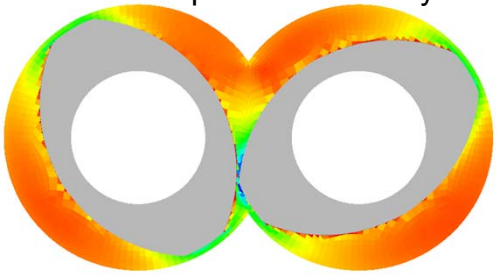


- Fast and accurate description of the rotating geometry
- Applicable for conveying and kneading elements
- Mathematical description available for
  single, double- or triplet-flighted screws
- Surface and body of the screws are known at any time
- Mathematical formulation replaces external CAD-description
- Non-Newtonian and temperature dependent physical properties
- Heat dissipation due to high shear rates

Velocity Magnitude



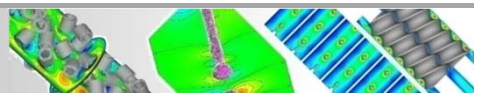
Shear dependent viscosity



In cooperation with:

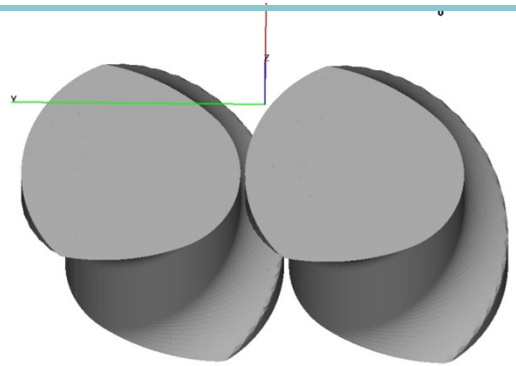
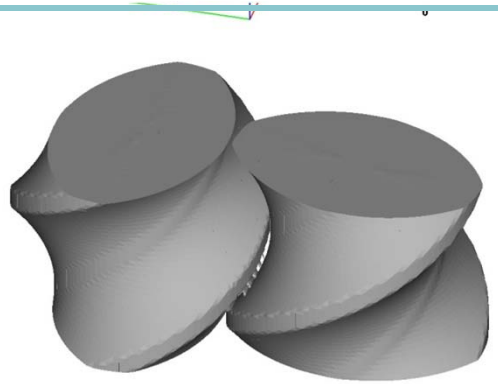
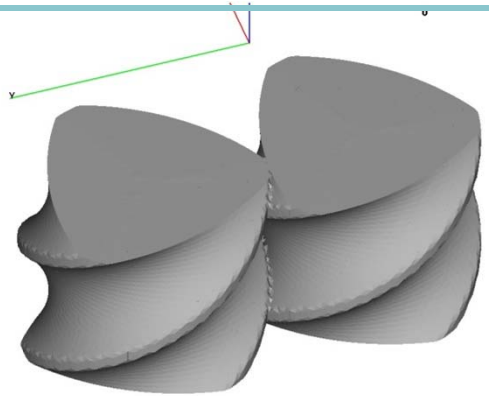
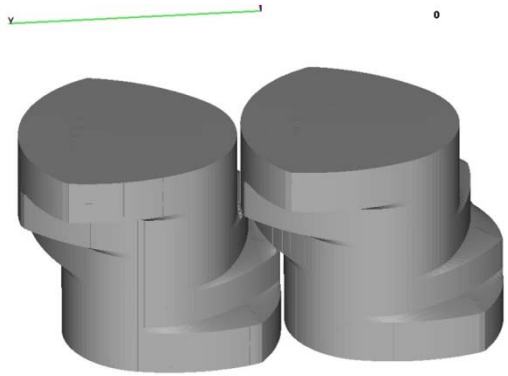
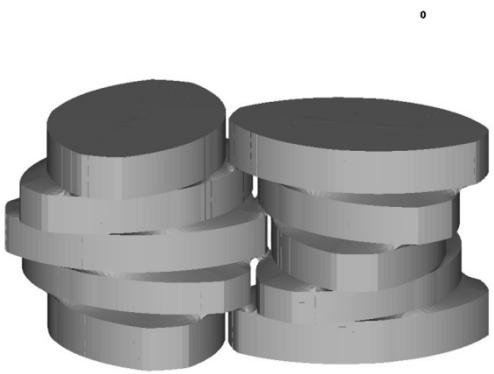
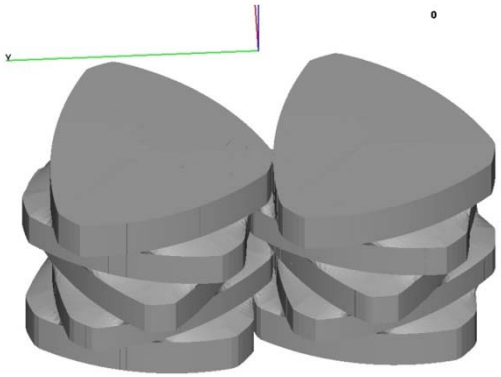
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

IANUS Simulation



Twinscrew Flow Simulation with FeatFlow

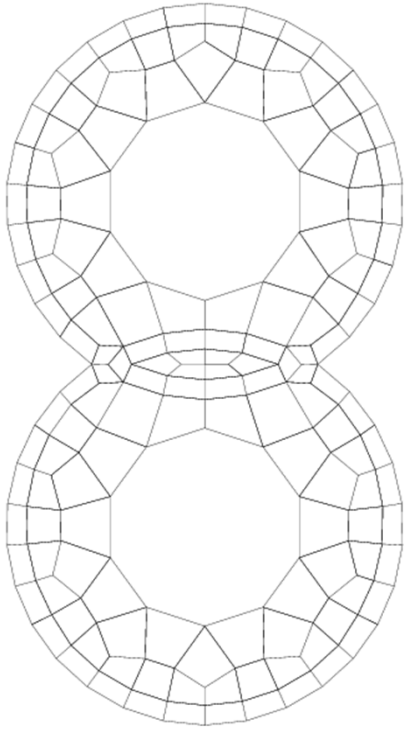
Library of conveying and mixing elements

	1 flighted	2 flighted	3 flighted
conveying			
mixing			

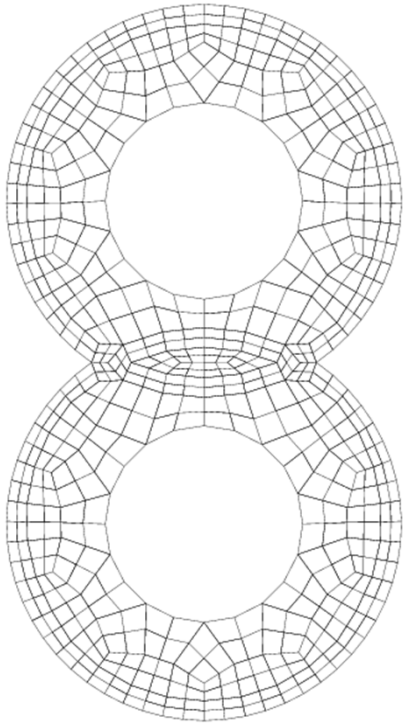
Twinscrew Flow Simulation with FeatFlow

Meshing strategy – Hierarchical mesh refinement

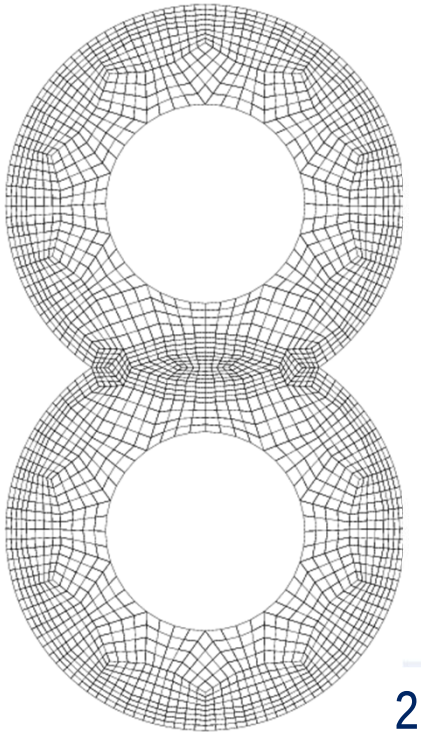
level 1

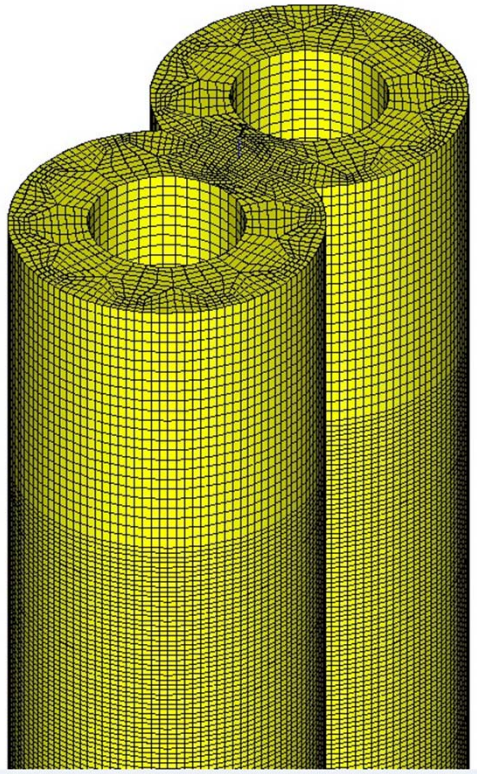


level 2



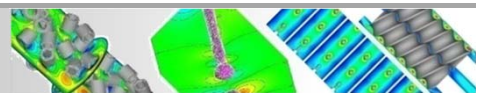
level 3



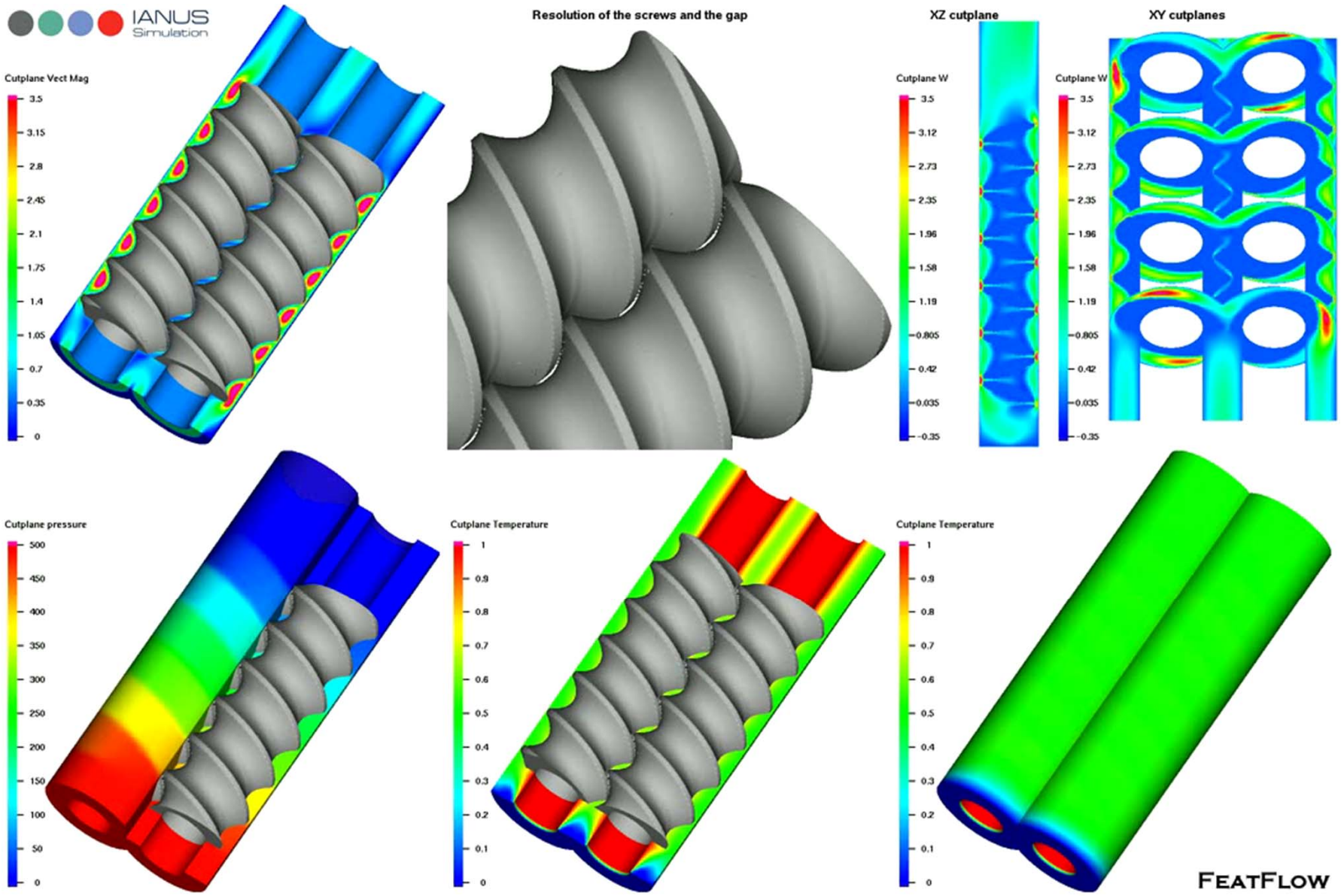


2D mesh extrusion into 3D

Pre-refined regions in the vicinity of gaps

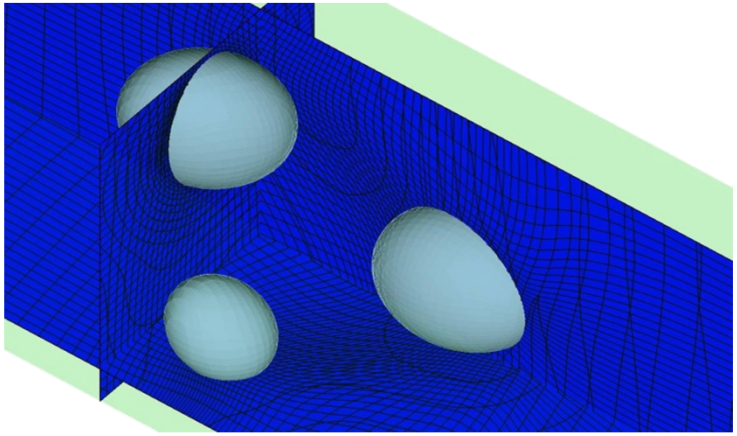


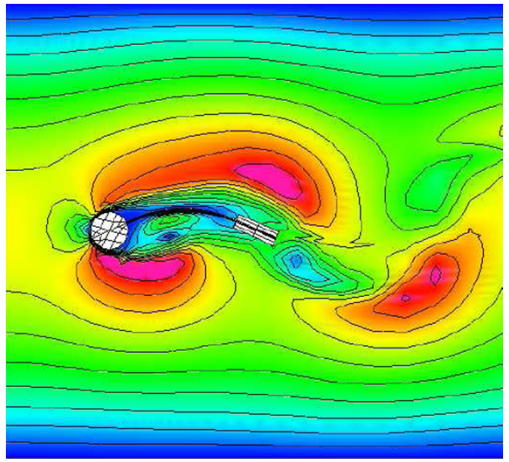
Twinscrew Flow Simulation with FeatFlow

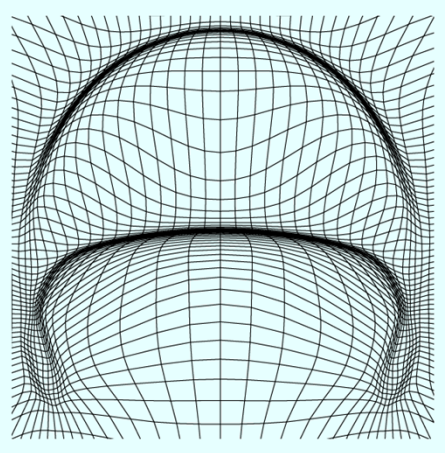


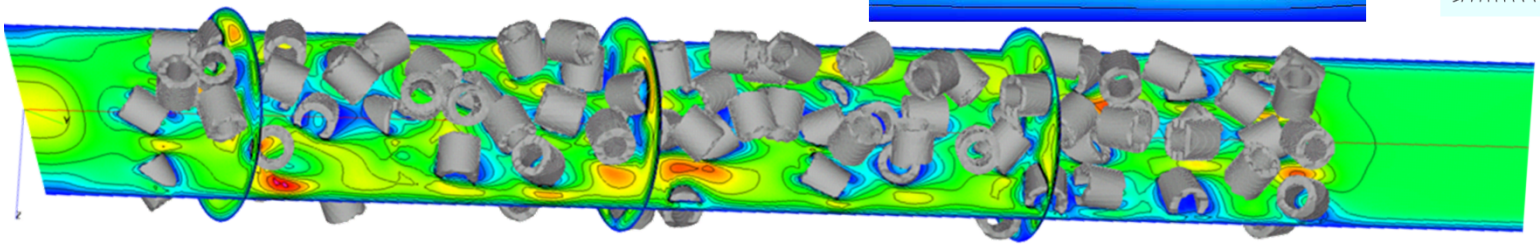
Challenges

- Adaptive time stepping + dynamical adaptive grid alignment/ALE.
- Coupling with turbulence models.
- Deformable particles/fluid-structure interaction.
- Analysis of viscoelastic effects.
- Benchmarking and experimental validation for many particles.



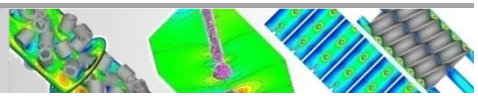






(Preliminary) State-of-the-Art

- Numerical efficiency?
  → OK
- Parallel efficiency?
  → OK (tested up to appr. 1000 CPUs on NEC / commodity clusters)
  → More than 10.000 CPUs???
- Single processor efficiency?
  → OK (for CPU)
- 'Peak' efficiency?
  → NO
  → Special ***unconventional*** FEM Co-Processors

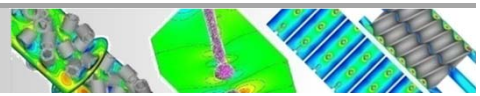


UnConventional HPC

- CELL multicore processor (PS3),
  7 synergistic processing units @ 3.2 GHz,
  Memory @ 3.2 GHz
  ≈ 218 GFLOP/s

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

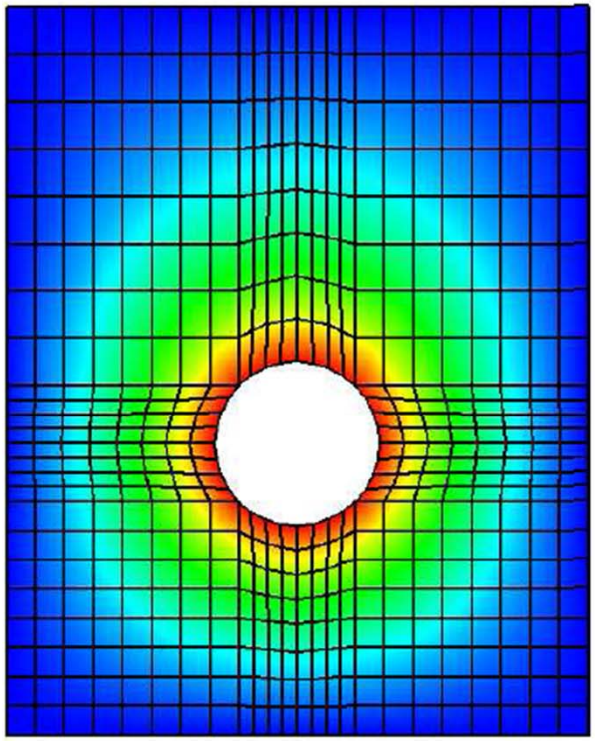
UnConventional High Performance Computing (UCHPC)

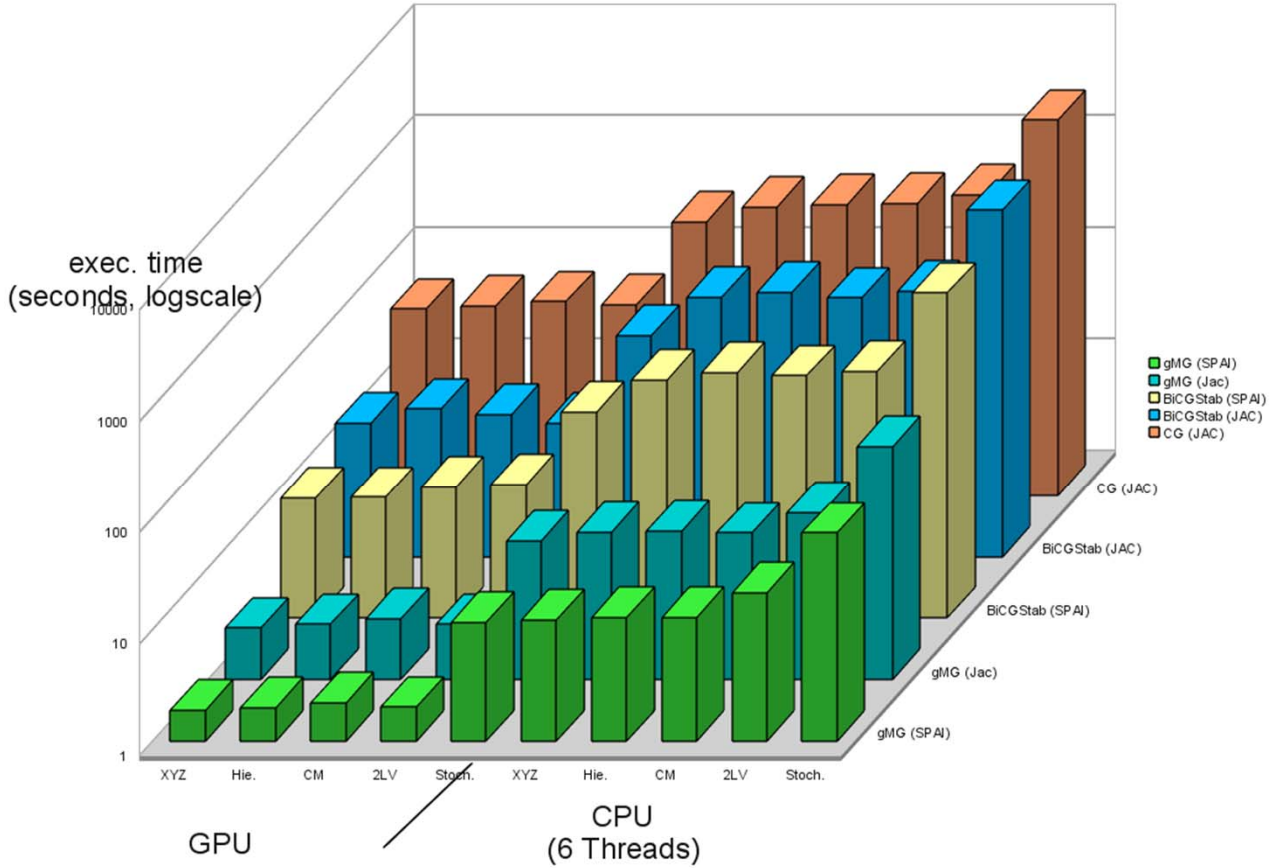


Poisson problem solver tests

$$
\begin{aligned}
-\Delta u &= 0 && \text{in } \Omega, \\
u &= 0 && \text{on } \Gamma_1 \\
u &= 1 && \text{on } \Gamma_2
\end{aligned}
$$

	Q_1		Q_2	
L	N	non-zeros	N	non-zeros
4	576	4552	2176	32192
5	2176	18208	8448	128768
6	8448	72832	33280	515072
7	33280	291328	132096	2078720
8	132096	1172480	526336	8351744
9	526336	4704256	2101248	33480704
10	2101248	18845696	-	-



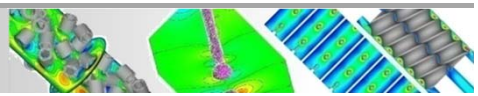


Huge Potential for the Future …

However:

- **Numerical Simulation & High Performance Computing** have to consider recent and future hardware trends, particularly for heterogeneous multicore architectures and massively parallel systems!
- The combination of **'Hardware-oriented Numerics'** and special **'Data Structures/Algorithms'** and **'Unconventional Hardware'** has to be used!

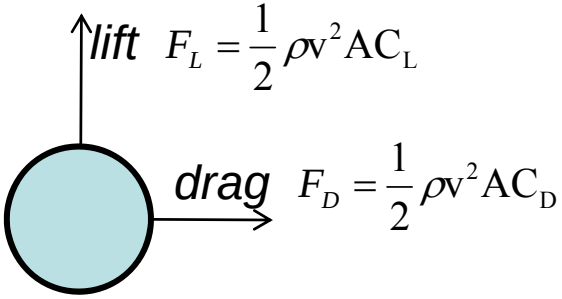
…or most of existing (academic/commercial) CFD software will be 'worthless' in a few years!

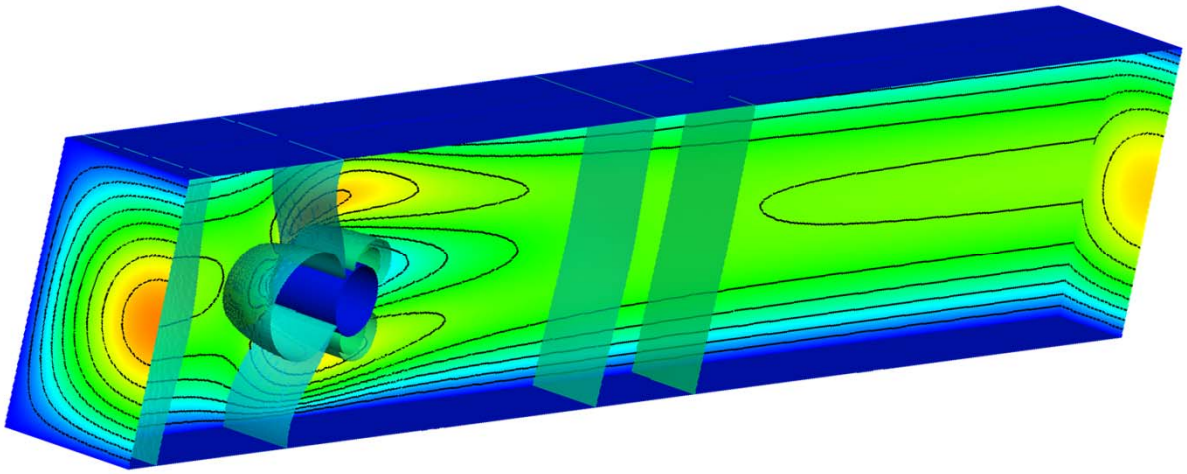


Benchmarking

Flow Simulation with CFD software available on the market

Known benchmark problem **(DFG) in the CFD community**



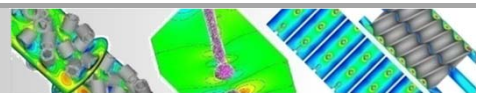


- Comparison of CFX 12.0, **OpenFoam 1.6** and **FeatFlow**
- Drag and lift coefficients behave very sensitive to mesh resolution

→ Ideal indicator for computational accuracy

- Five consequently refined meshes L1 (coarse), …, L5 (fine)
- Same meshes and physical models used in all three codes

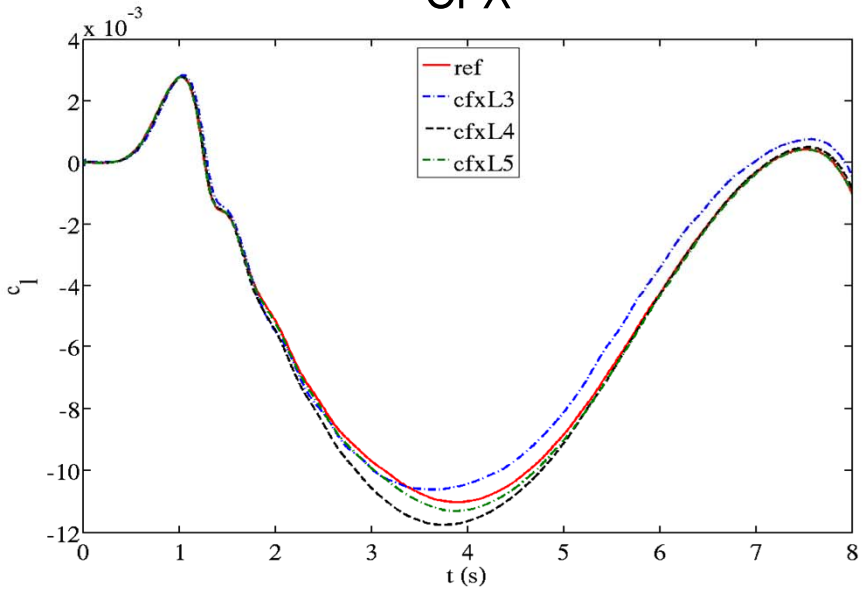
Mesh Level	# Elements
L2	6,144
L3	49,152
L4	393,216
L5	3,145,728



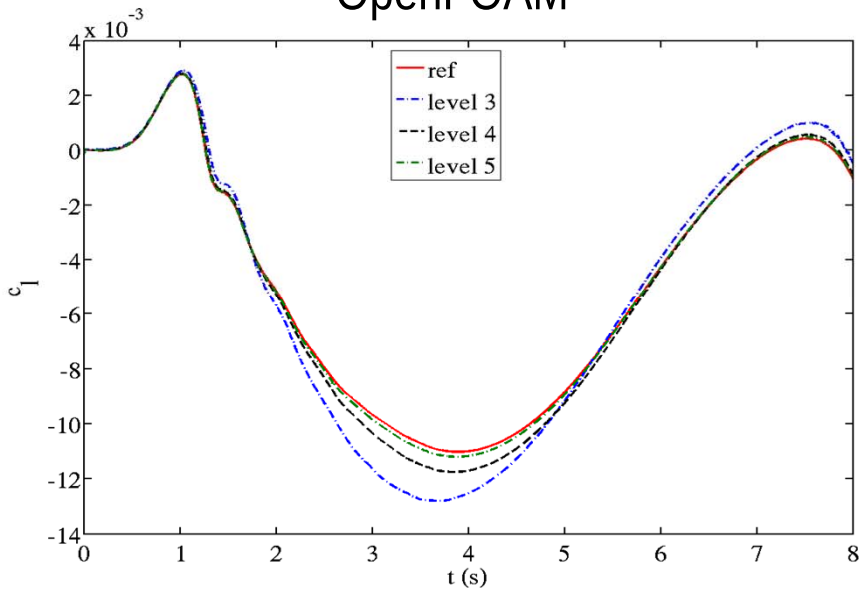
Benchmarking

Flow Simulation with CFD software available on the market

CFX

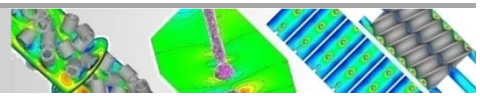


OpenFOAM



Case	L2 error		timing
	c_D	c_L	
CFX L3	0.0152	0.0781	13420
CFX L4	0.0098	0.0631	4 x 58680
CFX L5	0.0029	0.0224	8 x 205600

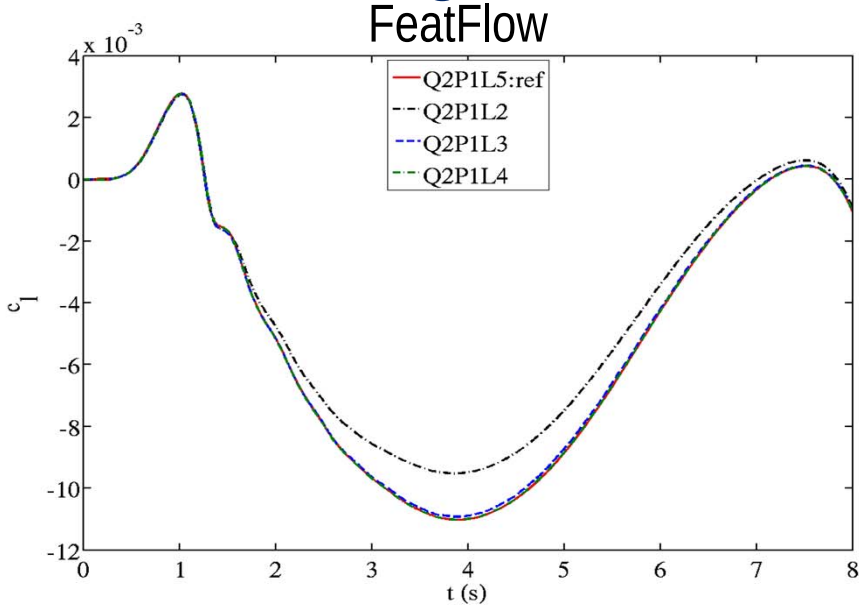
Case	L2 error		Timing
	c_D	c_L	
OF L3	0.0261	0.1449	5180
OF L4	0.0067	0.0591	4 x 19500
OF L5	0.0016	0.0147	8 x 595200



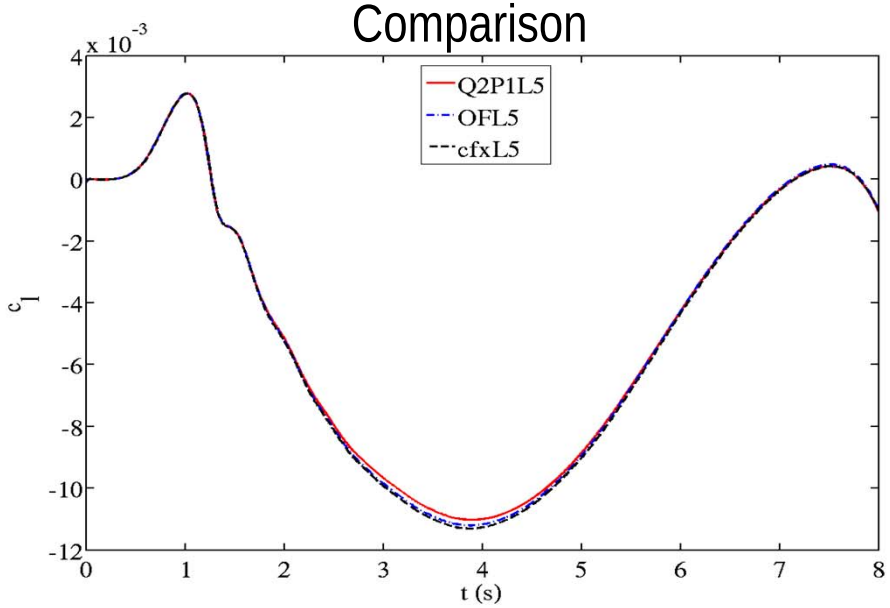
Benchmarking

Flow Simulation with FEATFLOW

FeatFlow



Comparison



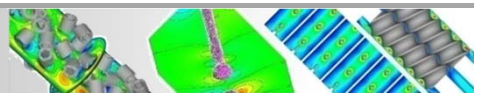
Case	L2 error		Timing
	c_D	c_L	
FF L2	0.0209	0.1378	2 x 5000
FF L3	0.0029	0.0109	3 x 25000
FF L4	0.0005	0.0015	20 x 32000
FF L5	(ref)	(ref)	23 x 242000

Case	L2 error		timing
	c_D	c_L	
FF L3	0.0029	0.0109	3 x 25000
OF L5	0.0016	0.0147	8 x 595200
CFX L5	0.0029	0.0224	8 x 205600

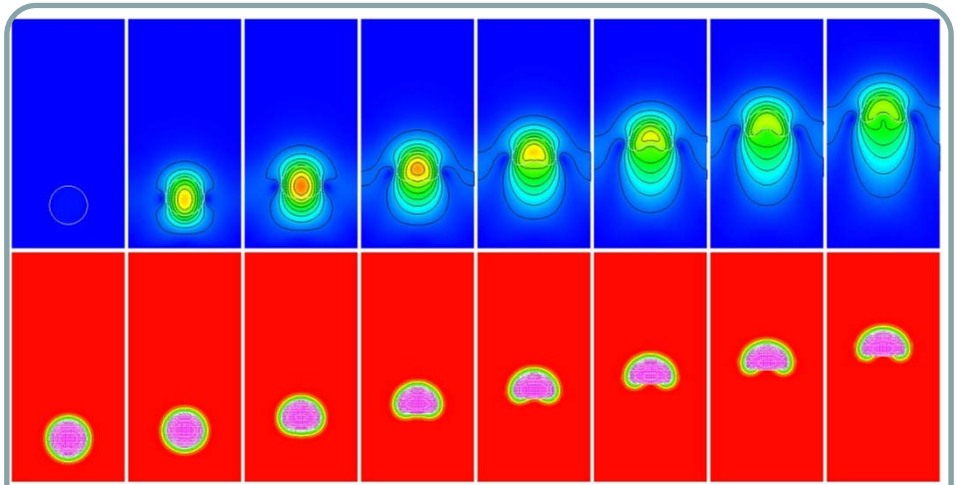
Less than 2 hours sim. time with adaptive time stepping on 3+1 processors

→ Same order of accuracy with FEATFLOW on L3 as L5 with CFX and OpenFOAM on L5!

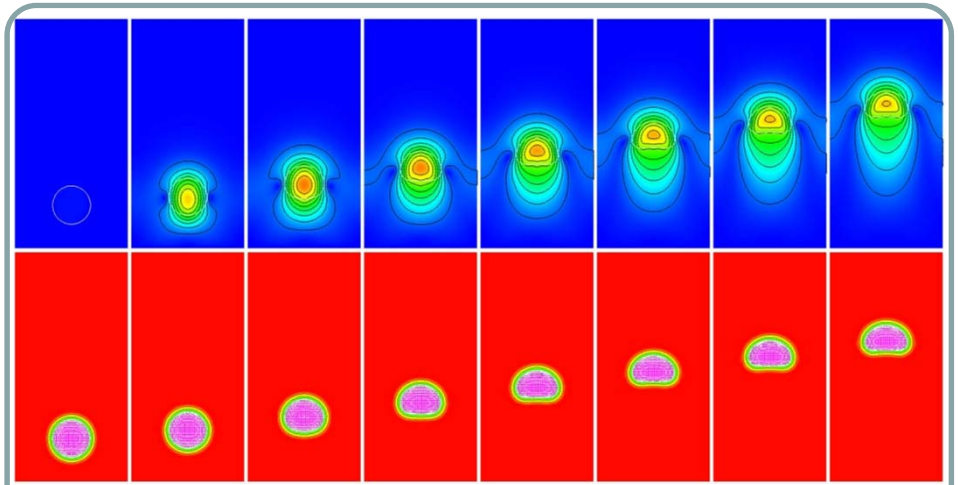
→ High order Q2/P1 FEM + (parallel) Multigrid Solver



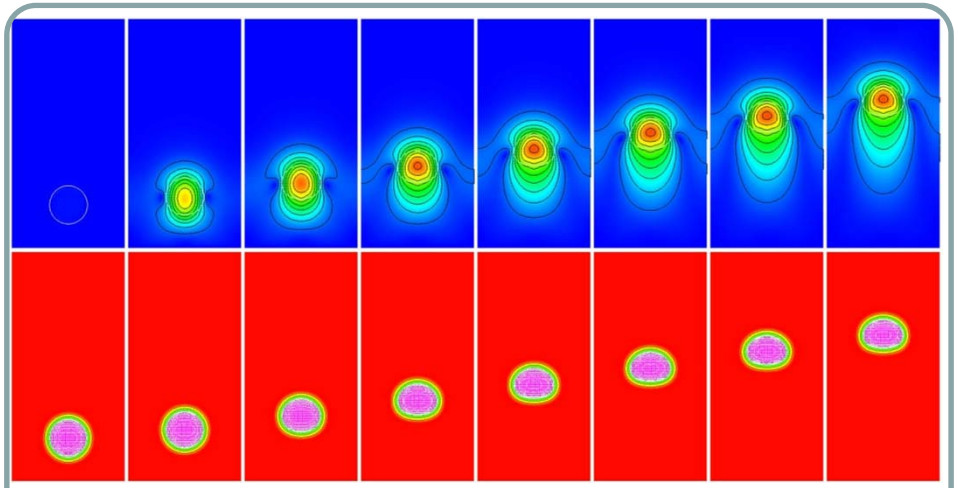
3D parameter study for large density jumps



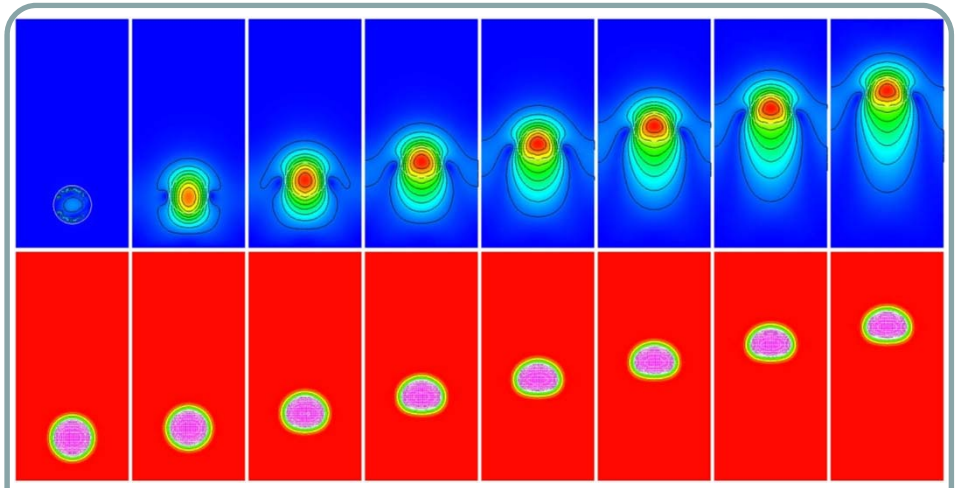
Density ratio: 1 to 10
Surface tension : ~2



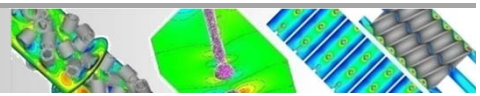
Density ratio: 1 to 10
Surface tension : ~5



Density ratio: 1 to 10
Surface tension : ~10

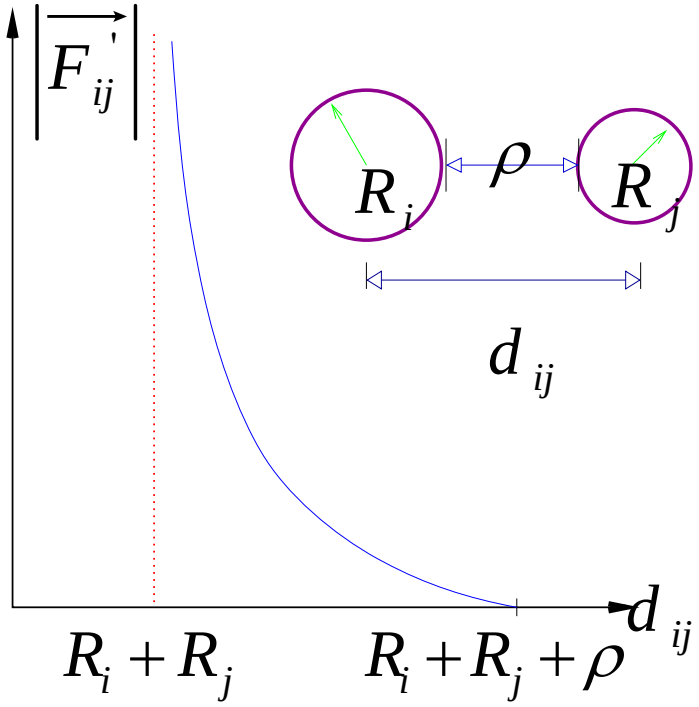


Density ratio: 1 to 100
Surface tension : ~10



Collision Models

- **Theoretically**, it is impossible that smooth particle-particle collisions take place in finite time in the **continuous system** since there are repulsive forces to prevent these collisions in the case of viscous fluids.
- **In practice**, however, particles can contact or even overlap each other in **numerical simulations** since the gap can become arbitrarily small due to unavoidable numerical errors.



$$\begin{cases} \left|\overrightarrow{F_{ij}}'\right| = 0, & d_{ij} \geq R_i + R_j + \rho, \\ \left|\overrightarrow{F_{ij}}'\right| = \dfrac{d_{ij}}{\varepsilon}, & d_{ij} = R_i + R_j. \end{cases}$$

3D Single Body Collision Model

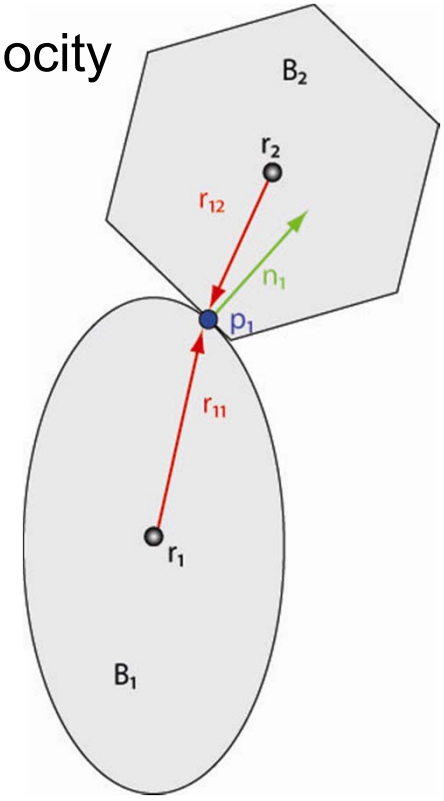
For a single pair of colliding bodies we compute the impulse f that causes the velocities of the bodies to change:

$$f = \frac{-(1+\varepsilon)(n_1(v_1 - v_2) + \omega_1(r_{11} \times n_1) - \omega_2(r_{12} \times n_1))}{m_1^{-1} + m_2^{-1} + (r_{11} \times n_1)^T I_1^{-1}(r_{11} \times n_1) + (r_{12} \times n_1)^T I_2^{-1}(r_{12} \times n_1)}$$

Using the impulse the f, the change in linear and angular velocity can be calculated:

$$v_1(t+\Delta t) = v_1(t) + \frac{fn_1}{m_1}, \omega_1(t+\Delta t) = \omega_1(t) + I_1^{-1}(r_{11} \times fn_1)$$

$$v_2(t+\Delta t) = v_2(t) + \frac{fn_2}{m_2}, \omega_2(t+\Delta t) = \omega_2(t) + I_2^{-1}(r_{12} \times fn_1)$$



n_1: contact normal r_{11}: vector from center r_1 to contact point p_1

r_{12}: contact normal r_{12}: vector from center r_2 to contact point p_1

p_1: contact point

3D Multi-Body Collision Model

In the case of multiple colliding bodies with K contact points the impulses influence each other. Hence, they are combined into a system of equations:

$$w = Ax + b, \quad w,x,b \in R^n, M \in R^{n \times n}$$

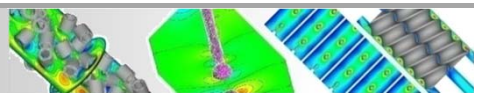
$$w,x,b \geq 0$$

$$(w_1 x_1, \ldots, w_n x_n) = (0, \ldots, 0)$$

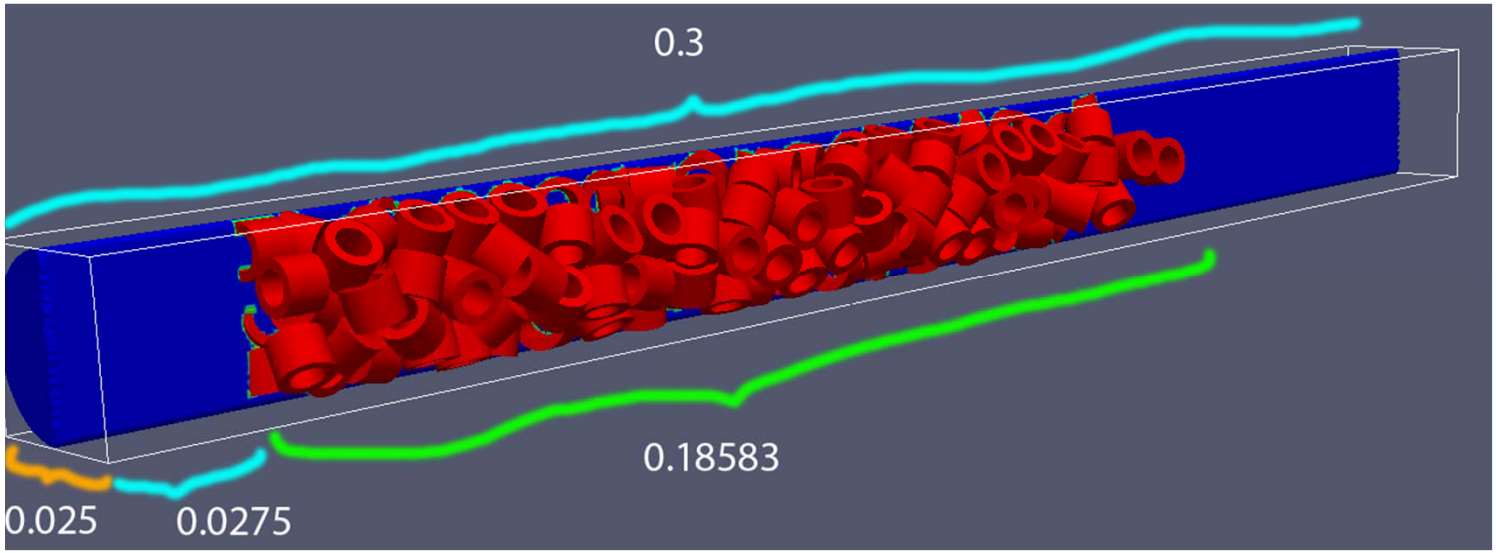
$$A = \delta_{i_1 i_k} n_l^T \left(\frac{1}{m_{i_k}} 1_{3\times 3} - r_{li_l}^{\times} I_{i_k}^{-1} r_{ki_k}^{\times} \right) n_k - \delta_{i_1 j_k} n_l^T \left(\frac{1}{m_{j_k}} 1_{3\times 3} - r_{li_l}^{\times} I_{j_k}^{-1} r_{kj_k}^{\times} \right) n_k$$

$$- \delta_{j_1 i_k} n_l^T \left(\frac{1}{m_{i_k}} 1_{3\times 3} - r_{lj_l}^{\times} I_{i_k}^{-1} r_{ki_k}^{\times} \right) n_k + \delta_{j_1 j_k} n_l^T \left(\frac{1}{m_{j_k}} 1_{3\times 3} - r_{lj_l}^{\times} I_{j_k}^{-1} r_{kj_k}^{\times} \right) n_k$$

The resulting problem is a linear complementarity problem (LCP), that can solved with efficient iterative methods like the projected Gauss-Seidel solver (PGS).



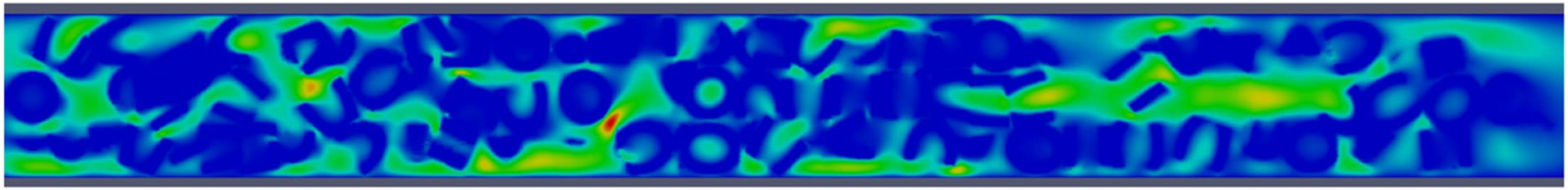
Packed bed reactor simulations (BASF)



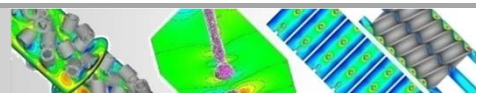
BASF
The Chemical Company

$$v_{mean} = (1 \mid 0.1 \mid 0.01)\, ms^{-1}$$
$$\rho = 1.25\ g\,cm^{-3}$$
$$\mu = 17.57\,10^{-6}\ Pa\ s$$

Velocity distribution

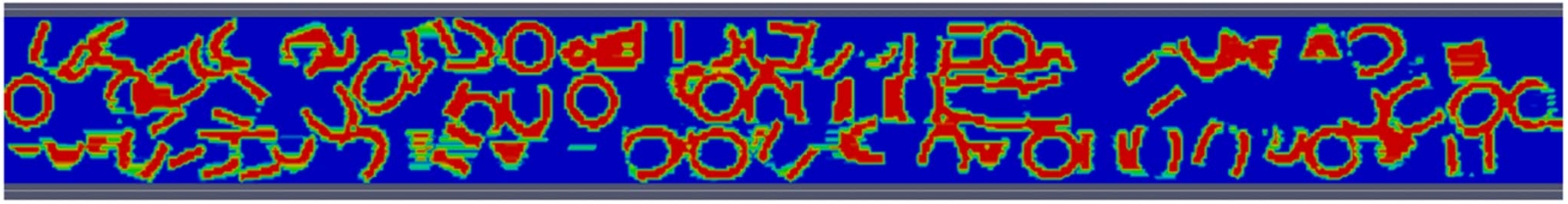


Pressure distribution



Packed bed reactor simulations (BASF)

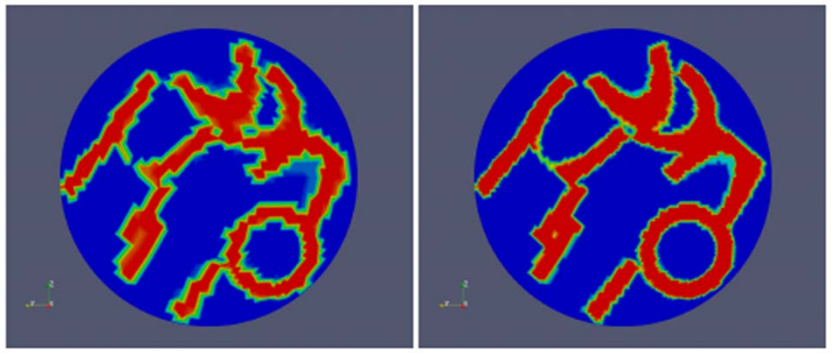
Level	Mesh points			Velocity DOFs			Pressure DOFs		
	n_x	n_{yz}	n_{xyz}	n_x	n_{yz}	n_{xyz}	n_x	n_{yz}	n_{xyz}
2	155	109	16,895	308	409	126,381	154(*4)	96(*4)	59,136
3	309	409	126,381	617	1,585	977,945	308(*4)	384(*4)	473,088
4	617	1,585	977,945	1,233	6,241	7,695,153	616(*4)	1,536(*4)	3,784,704



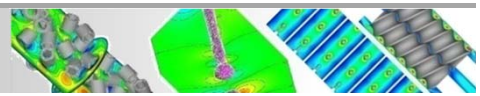
L3

L4

L3 L4



	Case: 0.01 m/s	Case: 0.10 m/s
Mesh/Resolution	Pressure Drop [Pa]	Pressure Drop [Pa]
L3-a	0.178	2.91
L3-b	0.193	3.15
L4-a	0.207	3.39
L4-b	0.218	3.54



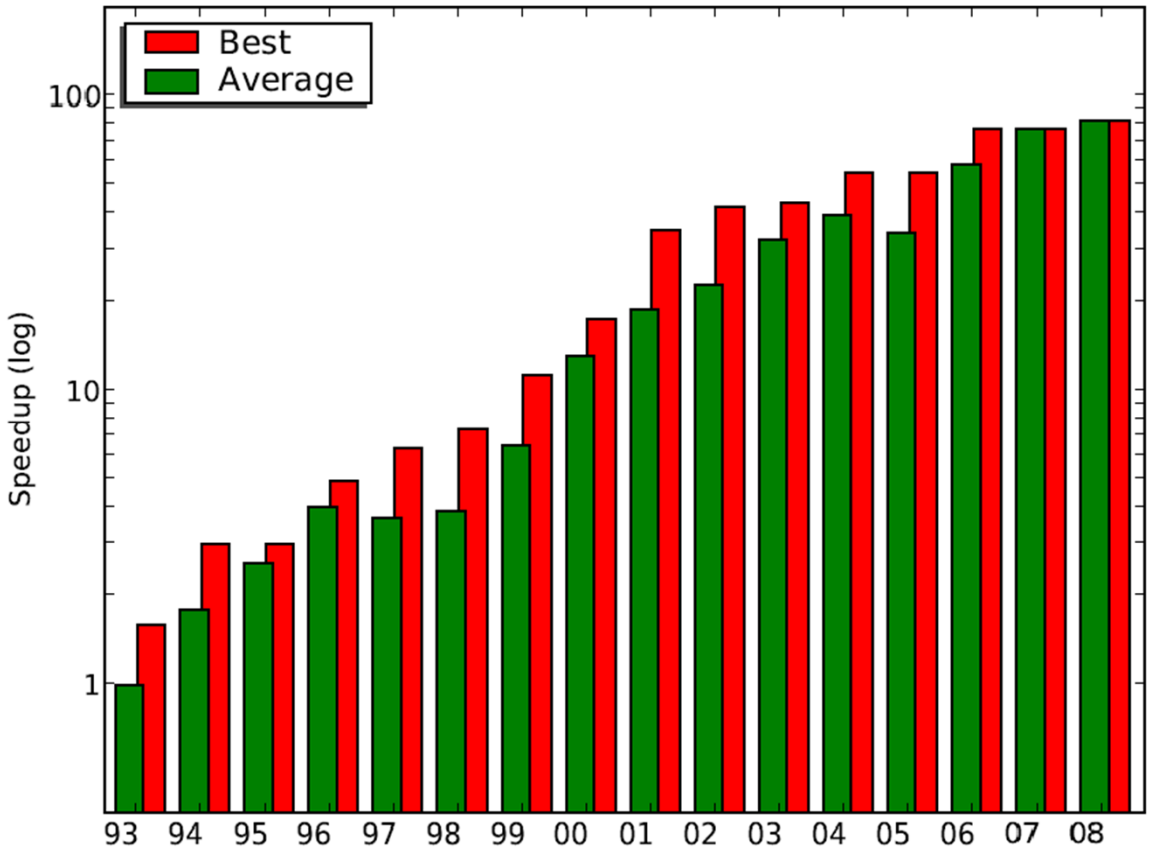
Next: Special HPC Techniques

The ‘free ride’ is over, paradigm shift in HPC:

- phys ltage)
- mem oblems)
- appli er hardware

Heterogen sors

- grap
- CELI
- HPC
- reco



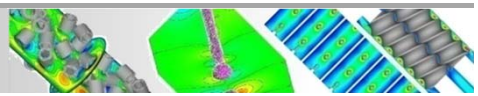
Finite Elem
solvers: mo
simulation t

Example: Fast Poisson Solvers

- 'Optimized' Multigrid methods for **scalar PDE** problems (≈Poisson problems) on **general meshes** should require ca. 1000 FLOPs per unknown (in contrast to LAPACK for dense matrices with $O(N^3)$ FLOPs)

- Problem size 106 : Much less than 1 sec on PC (???)
- Problem size 1012: Less than 1 sec on PFLOP/s computer

➔ **More realistic (and much harder) 'Criterion' for Petascale Computing in Technical Simulations**



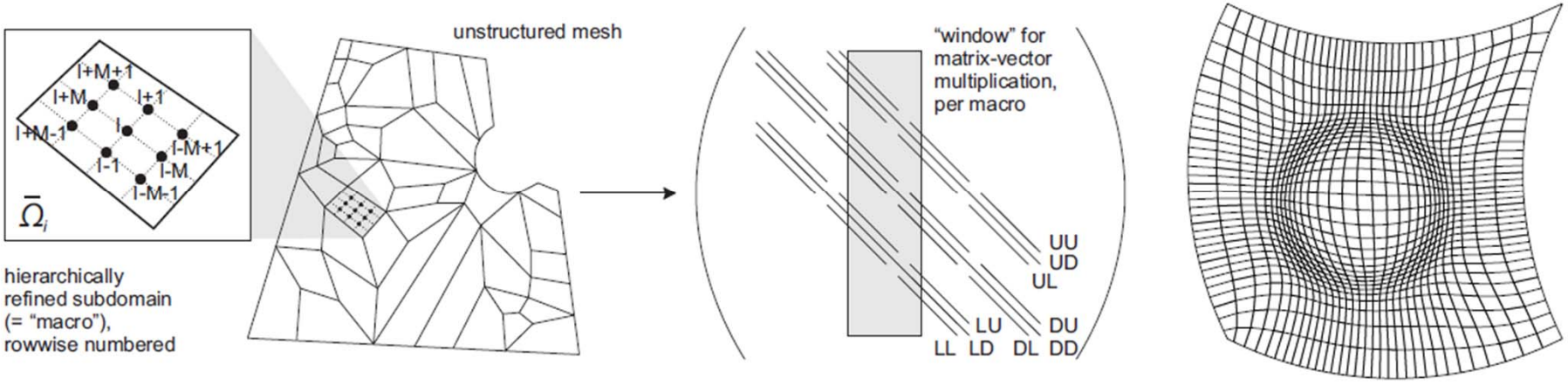
Main Component: 'Sparse' MV

- Sparse **Matrix-Vector techniques** ('indexed DAXPY') on **general unstructured grids**

```
      DO 10 IROW=1,N
         DO 10 ICOL=KLD(IROW),KLD(IROW+1)-1
10          Y(IROW)=DA(ICOL)*X(KCOL(ICOL))+Y(IROW)
```

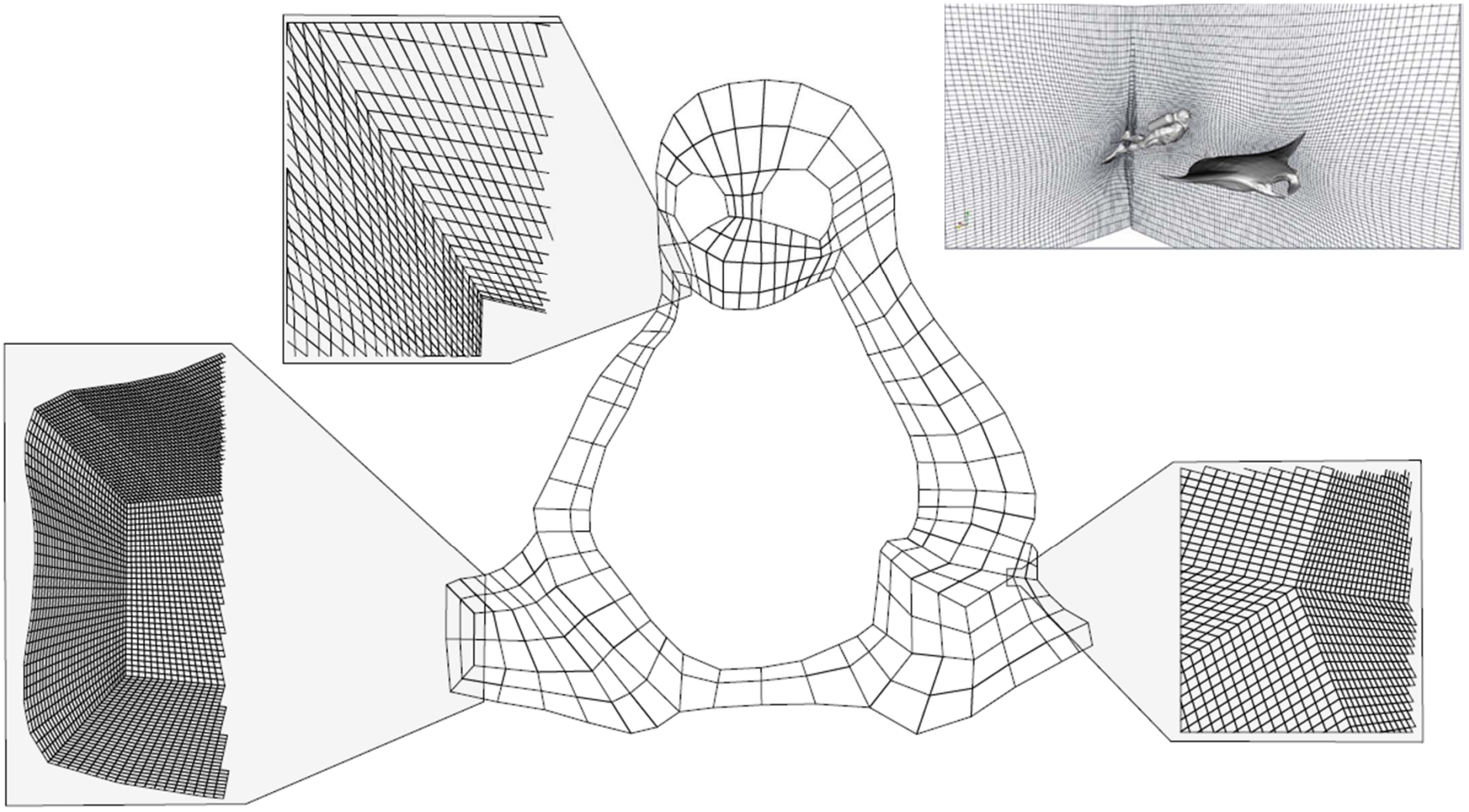
Fully adaptive grids
Maximum flexibility
'Stochastic' numbering
Unstructured sparse matrices
Indirect addressing, very slow.

- Sparse Banded **Matrix-Vector techniques** on **generalized TP grids**

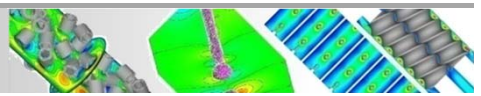


Locally structured grids
Logical tensor product
Fixed banded matrix structure
Direct addressing ($\Rightarrow$ fast)
r-adaptivity

Generalized Tensorproduct Meshes

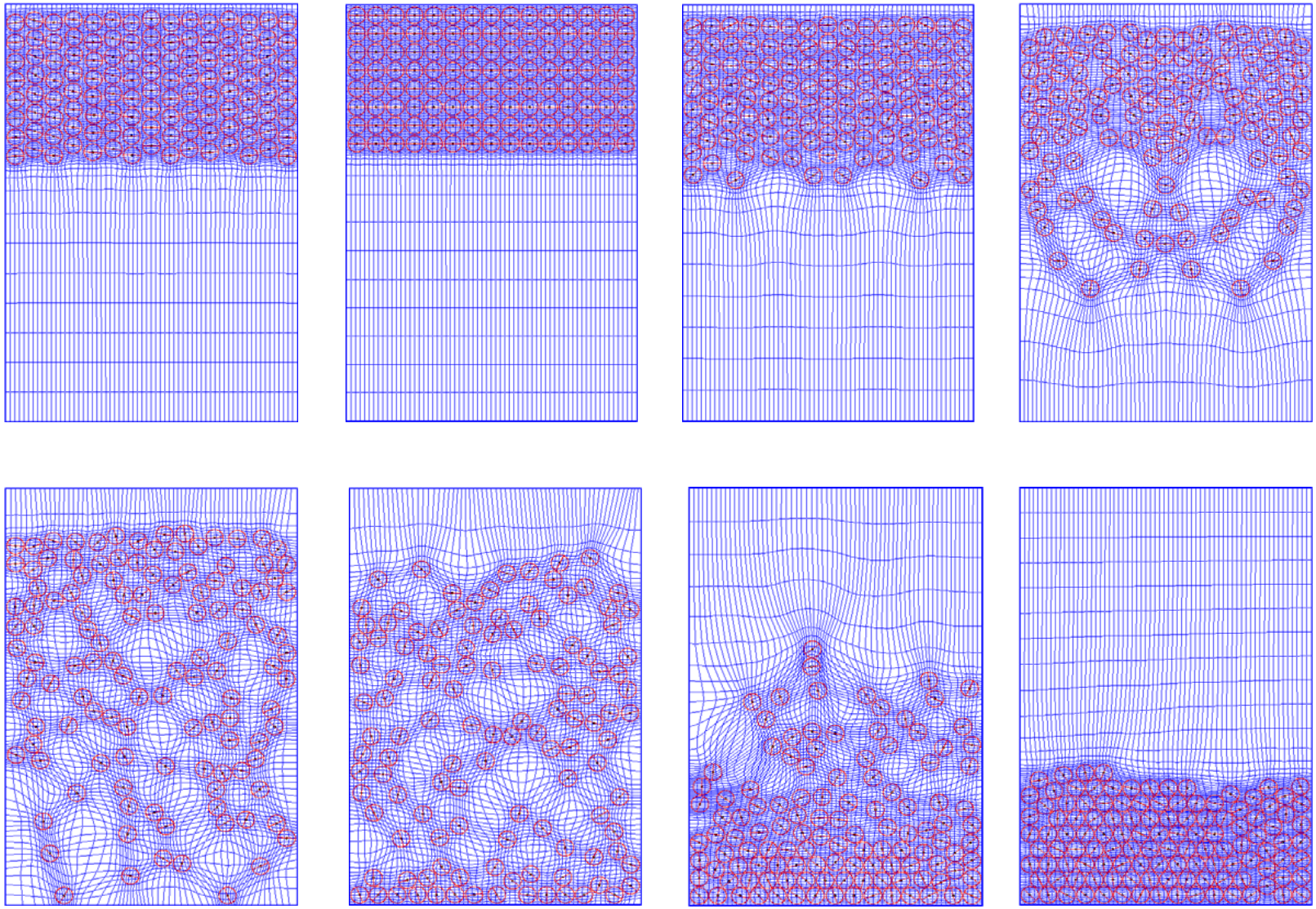


…with appropriate FBM techniques in FeatFlow…..



Generalized Tensorproduct Meshes

….dynamic CFD problems…..



Extensive Tests show.....

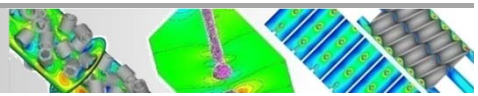
- It is (almost) impossible to come close to Single Processor Peak Performance with modern (= high numerical efficiency) simulation tools

- Parallel Peak Performance with modern Numerics even harder, already for moderate processor numbers

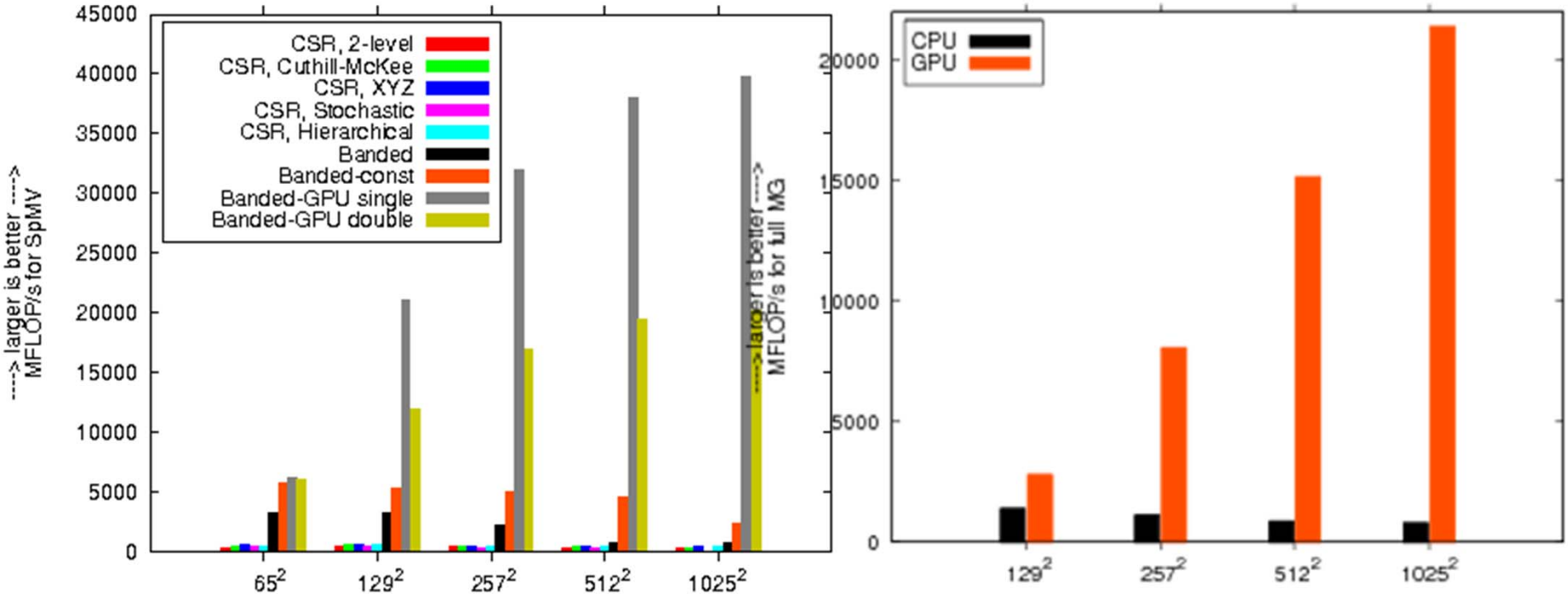
Hardware-oriented Numerics (HwoN)

Main features of the FEAST approach:

- Separation of structured and unstructured data
- Parallel multigrid solver (Scalable Recursive Clustering = ScaRC)
- Treatment of scalar and vector-valued probles
- Possibility of co-processor acceleration
- Applications (FEASTFlow, FEASTSolid, *FEASTLBM,SkaLB*)



Observation: Sparse MV on TP Grid



40 GFLOP/s on GeForce GTX 280

0.7 (1.4) GFLOP/s on Xeon E5450

- 1M unknowns in less than 0.1 seconds!
- 27x faster than CPU

Promising results,
attempt to integrate GPUs as FEM Co-Processors

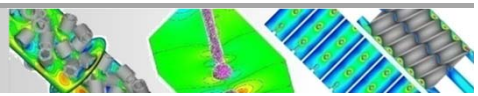
Design Goals

Include GPUs into FEAST

- without
  - changes to application codes FEASTFLOW / FEASTSOLID
  - fundamental re-design of FEAST
  - sacrificing either functionality or accuracy

- but with
  - noteworthy speedups
  - a reasonable amount of generality w.r.t. other co-processors
  - and additional benefits in terms of space/power/etc.

But: no --march=gpu/cell compiler switch



Two phase flow (I-I) with resolved interphases

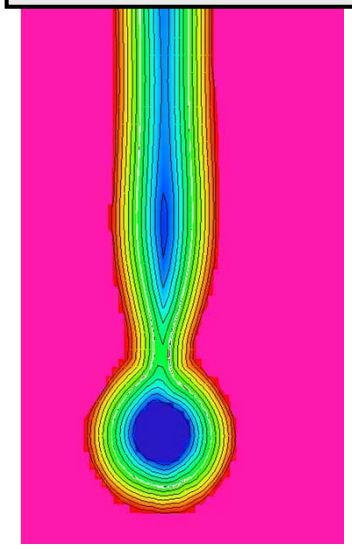
Continuum
Surface
Force

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

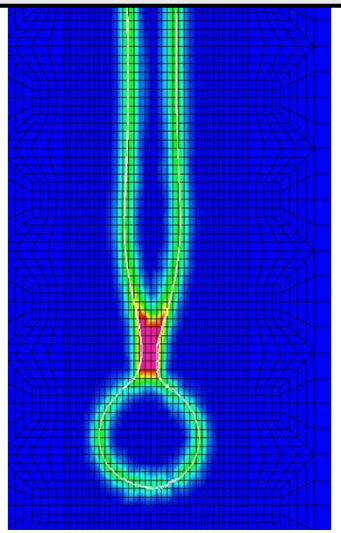
$$\mathbf{n}_{P_1} \xrightarrow{L_2 \text{ projection}} \mathbf{n}_{Q_1} \qquad \text{continuous normal field}$$

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

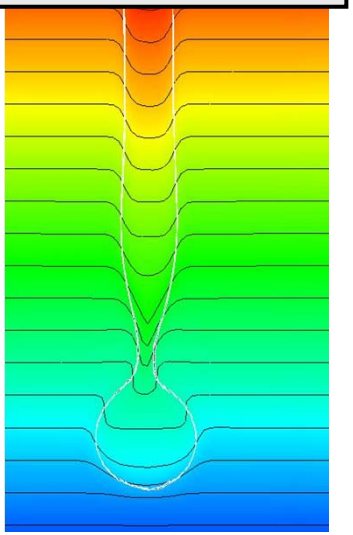
$$\kappa_{Q_1} = \frac{\int_\Omega \nabla \cdot \mathbf{n}_{Q_1}\, d\mathbf{x}}{\int_\Omega d\mathbf{x}} \qquad \text{continuous curvature field}$$



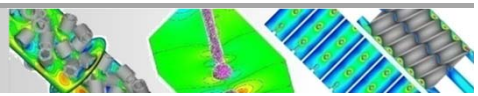
Level Set distribution



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



Resulting pressure distribution



Validation based on experimental results

Jetting mode

Experimental setup/results **Group of Prof. Walzel (BCI/Dortmund)**

Continuous phase:

Glucose-Water mixture

$\mu_D = 500\,mPa\,s$

$\rho_D = 972\,kg\,m^{-3}$

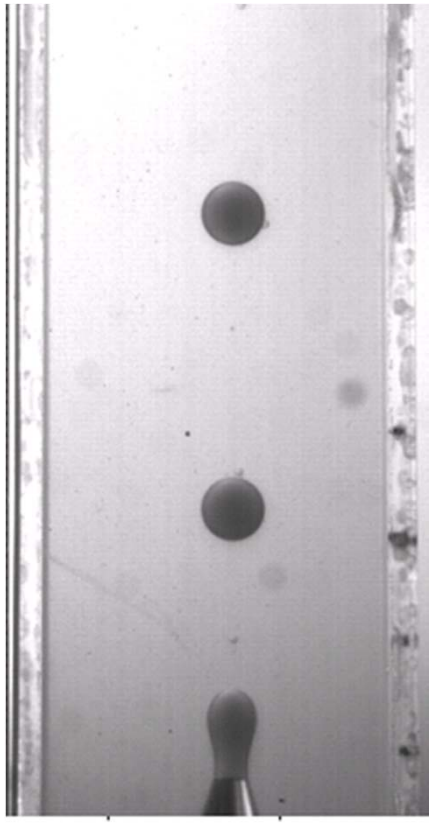
$\sigma_{CD} = 0{,}034\,N\,m^{-1}$

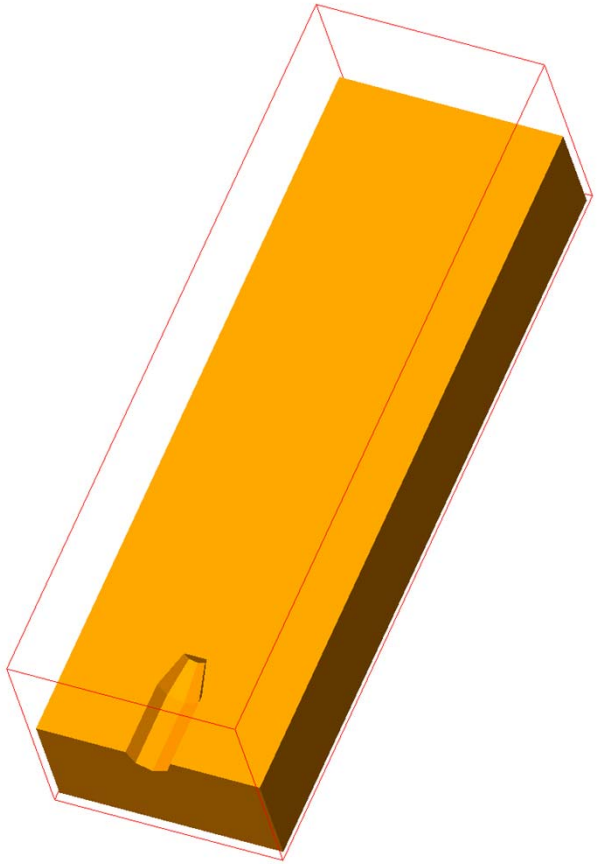
Silicon oil

$\mu_C = 500\,mPa\,s$

$\rho_C = 1340\,kg\,m^{-3}$

Dispersed phase:



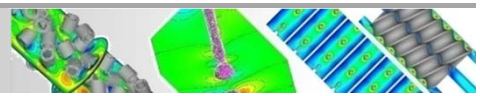


Operating conditions

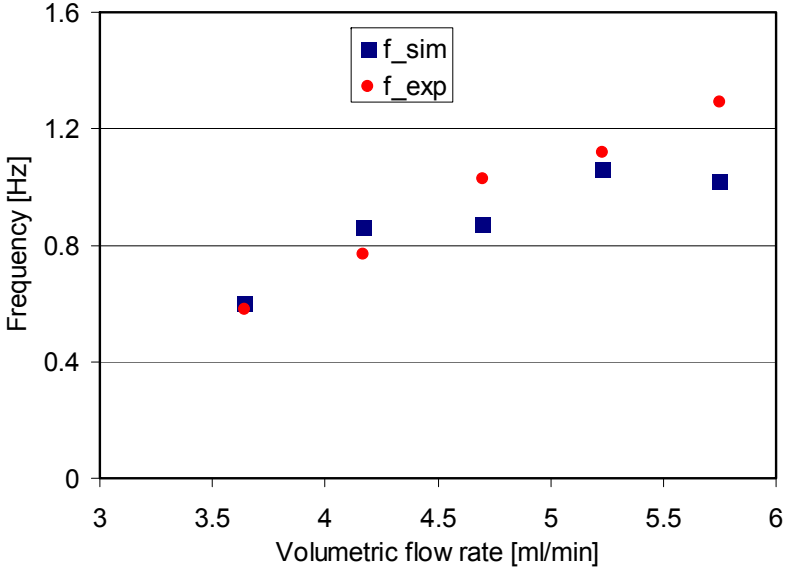
V_D [ml/min]	3,64	4,17	4,70	5,23	5,75
V_C [ml/min]	99,04	113,34	128,34	143,34	156,95

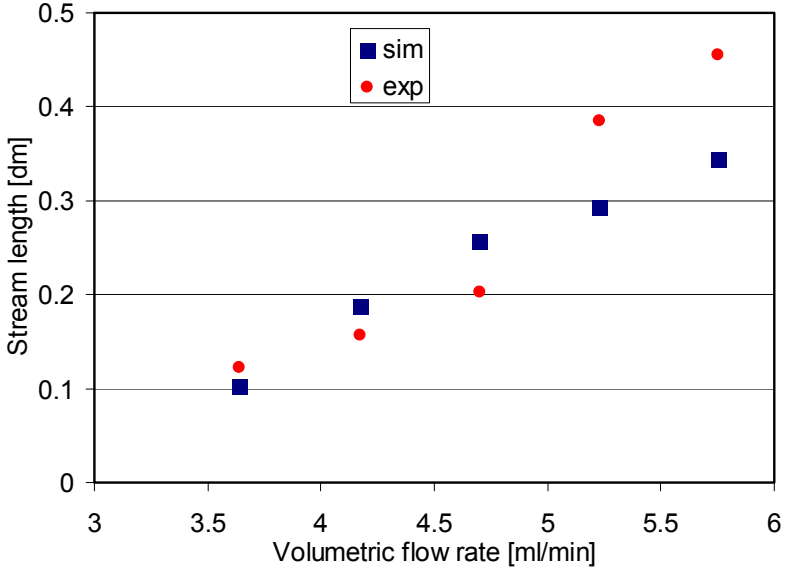
Validation parameters:

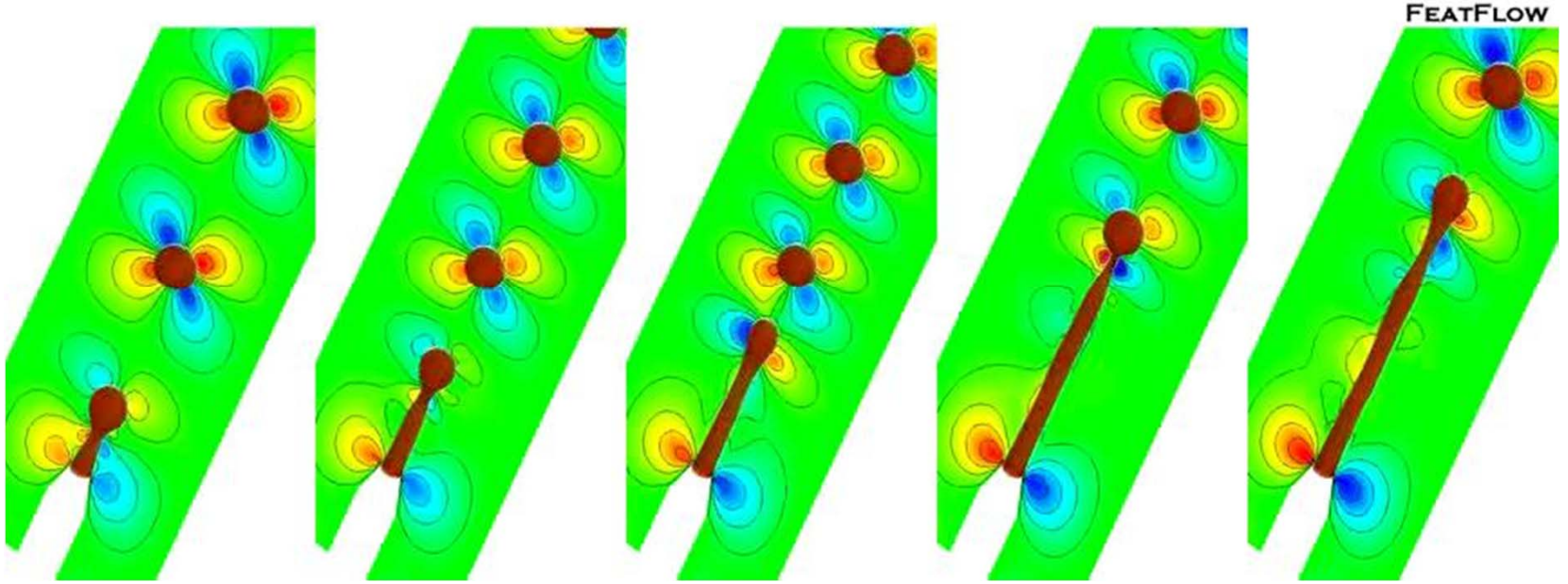
- frequency of droplet generation
- droplet size
- stream length



Validation based on experimental results







3,64 ml/min 4,17 ml/min 4,70 ml/min 5,23 ml/min 5,75 ml/min

Monodisperse droplet generation in nozzles

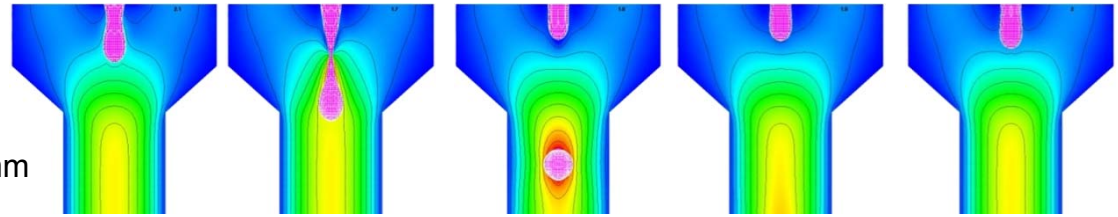
In case of monodisperse droplet generation:

$$\dot{V}_D = f V_{\text{droplet}}$$

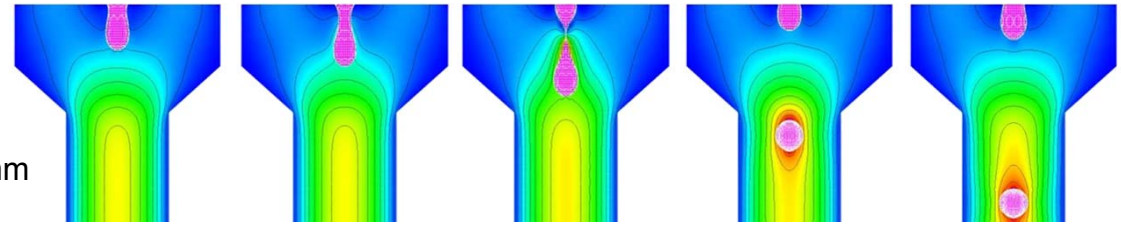
Influencable variables
On the level of the process:
- Flowrates
- Modulation frequency
- Modulation amplitude

Geometrical changes:
- Capillary size
- Contraction angle
- Contraction ratio

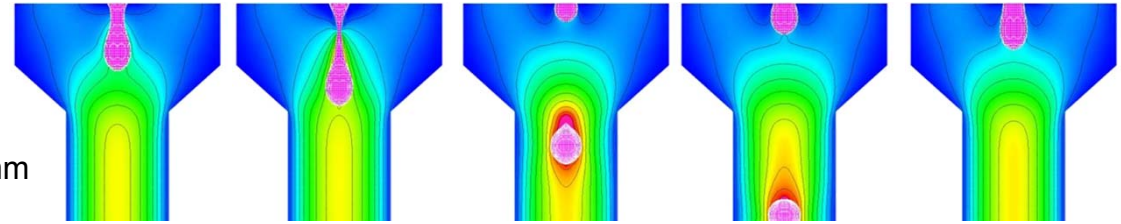
No Regulation
Flowrate: 100%
Capillary: STD
Droplet size: 5.2mm



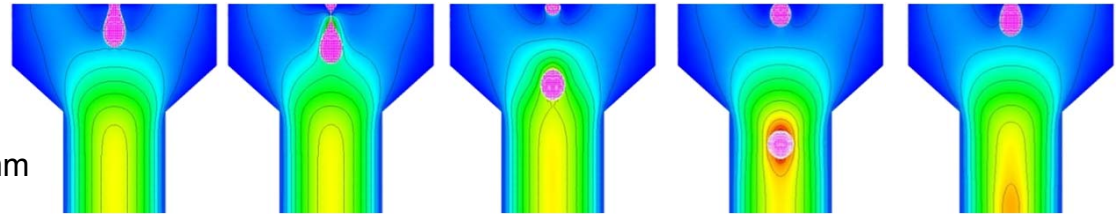
Regulated
Flowrate: 100%
Capillary: STD
Droplet size: 5.0mm



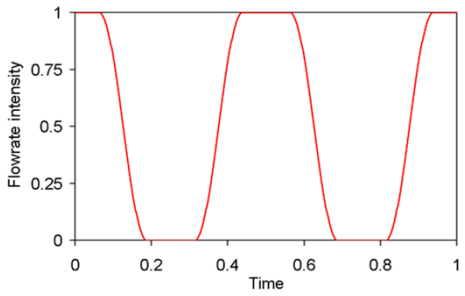
Regulated
Flowrate: 150%
Capillary: STD
Droplet size: 5.7mm

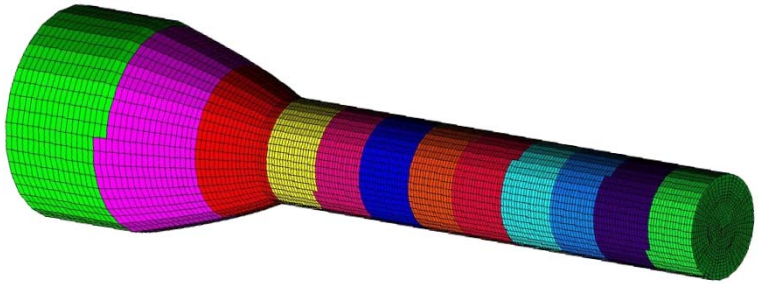


Regulated
Flowrate: 75%
Capillary: STD
Droplet size: 4.5mm

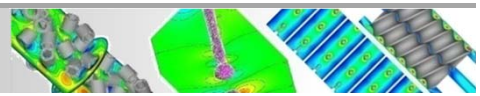


Modulation function

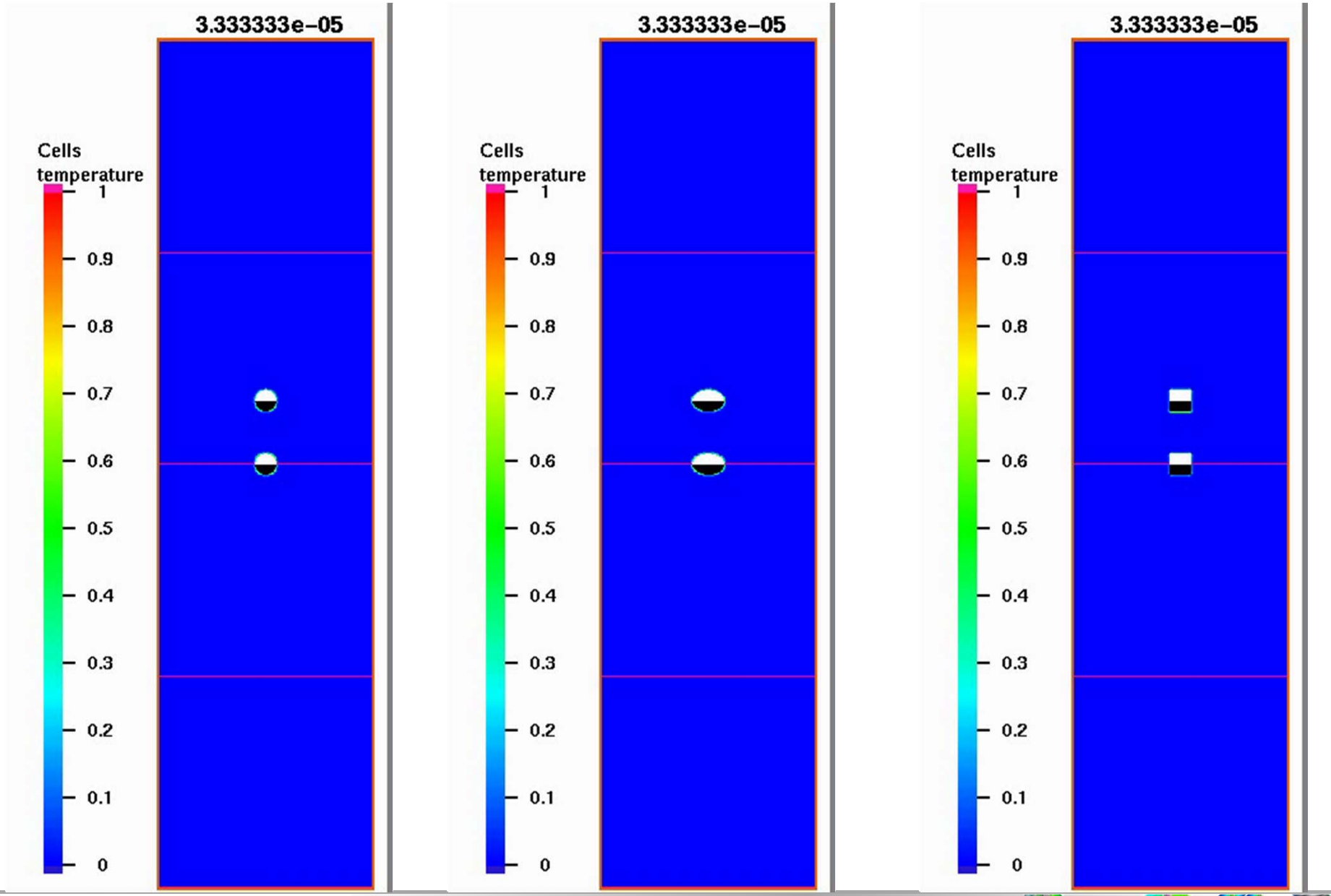




Resulting operation envelope:
- Size: 4.5 mm – 5.7 mm
- Volume: 0.38 cm^3 – 0.77 cm^3

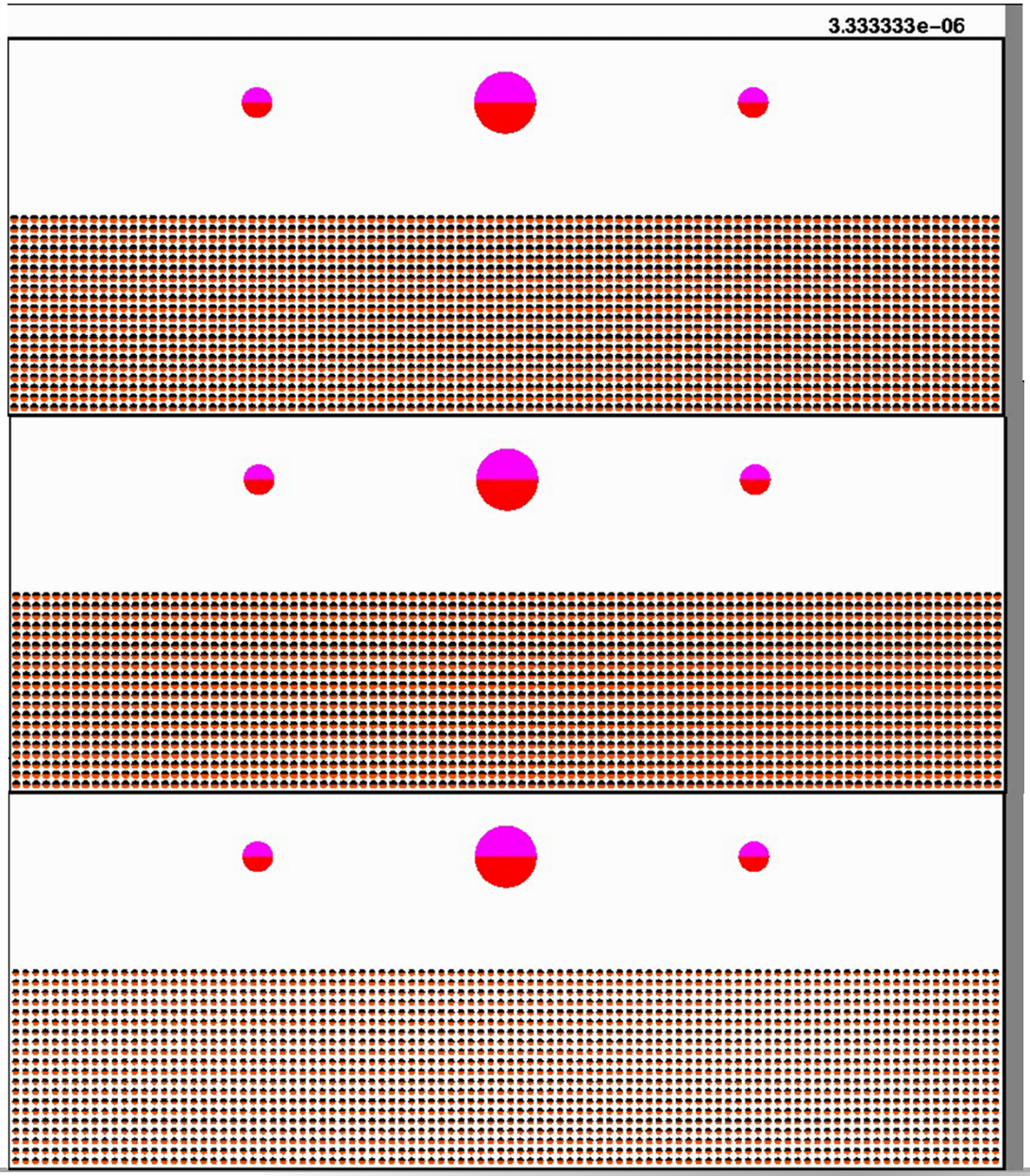


‘Kissing, Drafting, Thumbling’ of 2 Particles



Numerical Examples

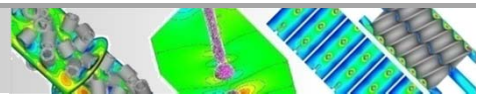
'Impact of heavy balls on 2000 small particles'



$$\rho_f = 1$$
$$\rho_{bd} = 2$$
$$\rho_{sp} = 20$$

$$\rho_f = 1$$
$$\rho_{bd} = 2$$
$$\rho_{sp} = 2$$

$$\rho_f = 1$$
$$\rho_{bd} = 2$$
$$\rho_{sp} = 1.1$$



Repulsive Force Collision Model

- Handling of small gaps and contact between particles
- Dealing with overlapping in numerical simulations

For the particle-particle collisions (analogous for the particle-wall collisions), the repulsive forces between particles read:

$$F_{ij}^{P} = \begin{cases} 0 & \text{for} \quad d_{i,j} > R_i + R_j + \rho \\ \frac{1}{\varepsilon_P}\left(X_i - X_j\right)\left(R_i + R_j + \rho - d_{i,j}\right)^2 & \text{for} \quad R_i + R_j \le d_{i,j} \le R_i + R_j + \rho \\ \frac{1}{\varepsilon_P^{'}}\left(X_i - X_j\right)\left(R_i + R_j - d_{i,j}\right) & \text{for} \quad d_{i,j} < R_i + R_j \end{cases}$$

The total repulsive forces exerted on the i-th particle by the other particles and the walls can be expressed as follows:

$$F_i^{'} = \sum_{j=1,\, j\neq i}^{N} F_{i,j}^{P} + F_i^{W}$$