FEM Techniques for Multiphase Flow Simulation

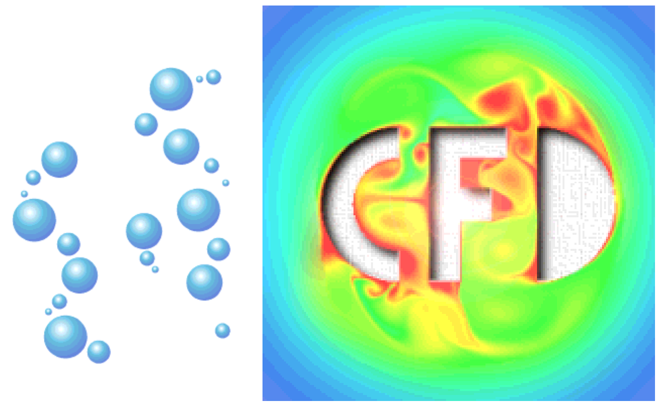
Stefan Turek

Institut für Angewandte Mathematik, Univ. Dortmund

`http://www.mathematik.uni-dortmund.de/LS3`

`http://www.featflow.de`

- FEM discretization and solution techniques
- Gas-liquid configurations
- Liquid-solid configurations



Multiphase Flow Models

Incompressible Navier-Stokes Equations

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = \mathbf{f} + \mu \Delta \mathbf{u} - \nabla p \quad , \quad \nabla \cdot \mathbf{u} = 0$$

+

‘Complex’ Extensions for Multiphase Flows

$\Rightarrow$ **Discretization:** ‘Find $\mathbf{u}_h \in \mathbf{V}_h$: $a_h(\mathbf{u}_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h) \, \forall \mathbf{v}_h \in \mathbf{V}_h$’

$\Rightarrow$ **Solver:** ‘Solve (nonlinear) System $A \cdot X = F$’

$\Rightarrow$ **Software:** ‘FEATFLOW/FEASTFLOW’

Aspects of FEM Discretization

1) BB Condition: $$\min_{q_h \in L_h} \max_{\mathbf{v}_h \in \mathbf{V}_h} \frac{(q_h, \nabla \cdot \mathbf{v}_h)}{\|\mathbf{v}_h\|_{1,h} \|q_h\|_0} \geq \alpha > 0$$

$\rightarrow$ Deformed and anisotropic meshes ?
$\rightarrow$ Quantitative (nonlinear) stabilization strategies ?

2) Korn's Inequality: $$\|D(\mathbf{v}_h)\|_0 \geq \beta \|\mathbf{v}_h\|_{1,h}$$

$\rightarrow$ For 'discontinuous' FEM ?

3) Stabilization of the convective term '$\mathbf{u}_h \cdot \nabla \mathbf{u}_h$'

$\rightarrow$ Upwind/Streamline-Diffusion for 'diffusion-dominated' case !
$\rightarrow$ FEM-Stabilization for pure 'transport-dominated' case ?

4) Higher Order FEM Spaces w.r.t. 1), 2), 3)

Discretization in Time

1) Second Order Accurate

2) (Semi) Implicit + (Strongly) A-Stable

$\longrightarrow$ Allowing huge time steps for (quasi-) stationary flow !
$\longrightarrow$ No CFL-type restrictions ! $\longleftarrow$ *local mesh adaptivity*

3) Accuracy-Based (Implicit) Error Indicator

$\longrightarrow$ Based on local truncation error for <u>all</u> physical quantities !

4) Rigorous Error Control

$\longrightarrow$ Residual-based a posteriori error control via dual problem ?
$\longrightarrow$ For user-defined time/space-averaged quantities ?

Aspects of 'Fast Solvers'

1) Newton Schemes: $U^{n+1} = U^n - \tilde{F}_\delta(U^n)^{-1} F(U^n)$

$\longrightarrow$ Frechet derivative $F_\delta(U^n)$? Simplifications $\tilde{F}_\delta(U^n)$?
$\longrightarrow$ Solvers for linear problems ? Dependence on parameters (Re,...) ?

2) Linear Multigrid/DD Schemes: $A_{lev} U_{lev} = F_{lev}$

$\longrightarrow$ Robustness ! Efficiency ! Parallelism !
$\longrightarrow$ Re ? Mesh size/deformations ? FEM types ?

3) (De)Coupling: $U^{n+1} = f(P^n), P^{n+1} = g(U^{n+1})$

$\longrightarrow$ Coupling of NSE with other components ?
$\longrightarrow$ Decoupling of pressure from velocity/temperature/etc. ?

$\Rightarrow$ *Pressure Schur Complement (PSC) solvers !*

Key Ideas of MPSC Approaches

‘Re-interpretation of Navier-Stokes solvers (Chorin, Van Kan, Uzawa, etc.) as "incomplete solvers" for discrete saddle-point problems’

LOCAL MPSC (‘MULTILEVEL PRESSURE SCHUR COMPLEMENT’):

‘*Fully coupled Newton-like solver as outer nonlinear procedure*’
‘*Solve "exactly" on "subsets/patches" and perform an outer Block–Gauß-Seidel/Jacobi iteration as smoother*’

$\Rightarrow$ **For (quasi-) stationary flow with ”large” time steps**

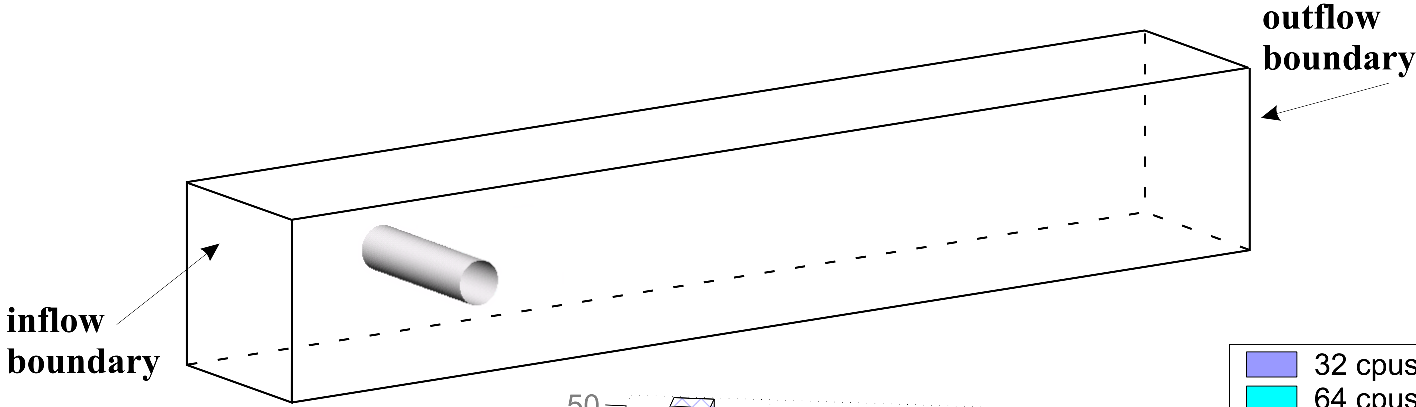
GLOBAL MPSC (‘MULTILEVEL PRESSURE SCHUR COMPLEMENT’):

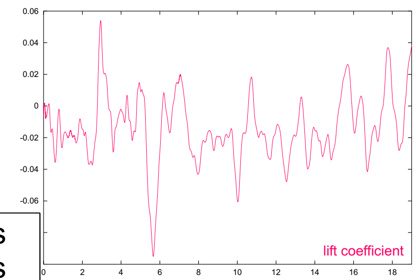
‘Outer (multigrid) coupling of velocity and pressure’
‘Newton/Multigrid solver for all scalar subproblems’

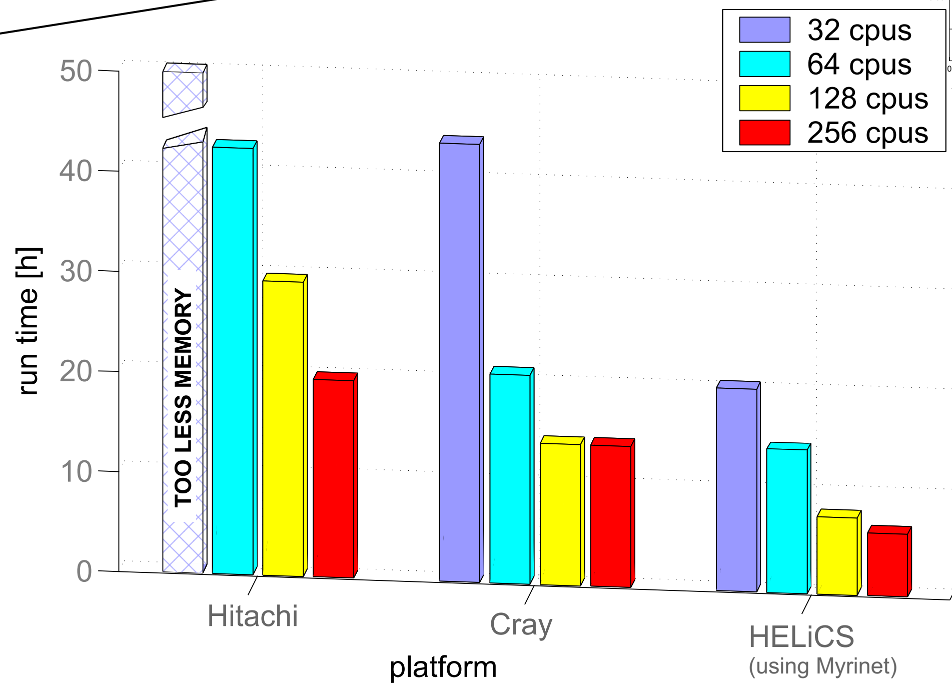
$\Rightarrow$ **For highly nonstationary flow**

Parallel Realization of GLOBAL MPSC

Dynamic simulations for ‘Channel Flow around Cylinder’ with 32 Mill. d.o.f.s for 6.500 implicit time steps (20 sec real time)







250 Mill. d.o.f.s are possible on LINUX clusters !

LOCAL MPSC: Flow around Cylinder

Nonlinear ‘Power-Law’ model				
	Conforming $Q_2 - P_1$ FEM			discrete Newton GMRES(ILU(1),200)
Lev	Drag	Lift	Δp	NNL/AVGMRES
3	$0.9475+03$	$0.3453+01$	16.02	**21/34**
4	$0.9534+03$	$0.3957+01$	15.82	**20/94**
5	$0.9569+03$	$0.4069+01$	15.86	**39/186**

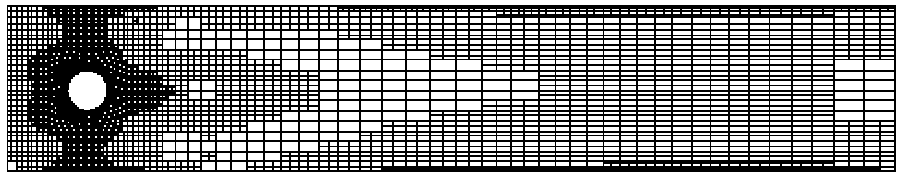
	Nonconforming $\tilde{Q}_1 - Q_0$ FEM			continuous Newton Multigrid
Lev	Drag	Lift	Δp	NNL/AVMG
4	$0.9160+03$	$0.3738+01$	15.74	**13/2**
5	$0.9351+03$	$0.3995+01$	15.82	**13/2**
6	$0.9462+03$	$0.4059+01$	15.85	**13/2**

Continuous Newton + Multigrid Solvers !!!

'Exploit hierarchical, but locally regular structures !!!'

I) *Patch-oriented adaptivity*

'Many' tensorproduct grids
'Few' unstructured grids



II) *Generalized MG-DD solver:* ScaRC

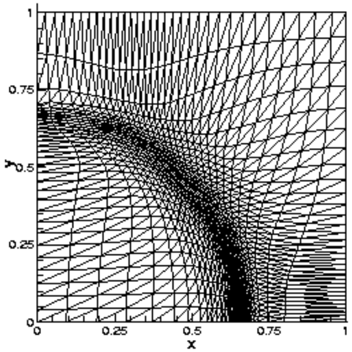
Exploit locally 'regular' structures **(efficiency)**
Recursive 'clustering' of anisotropies **(robustness)**

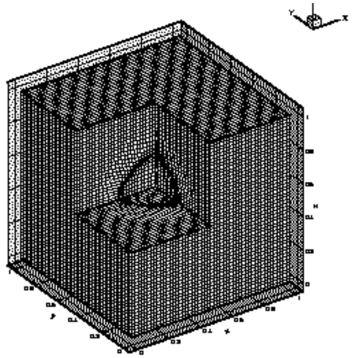
'Strong local solvers improve global convergence !'

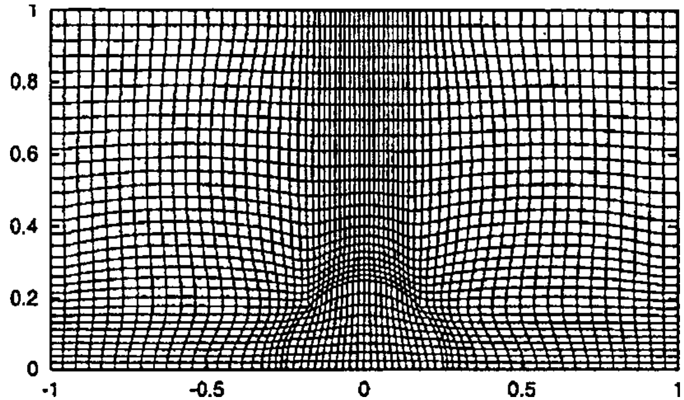
Concepts for Adaptive Meshing

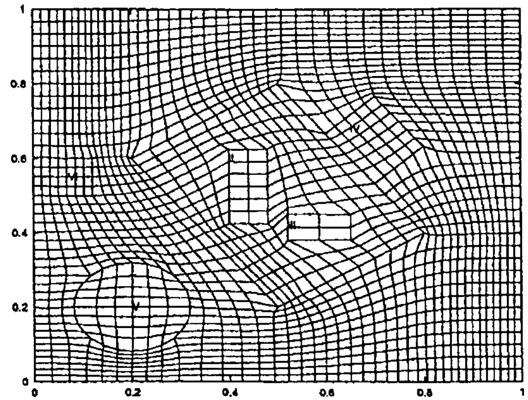
1) macro-oriented adaptivity

2) patchwise 'deformation' adaptivity

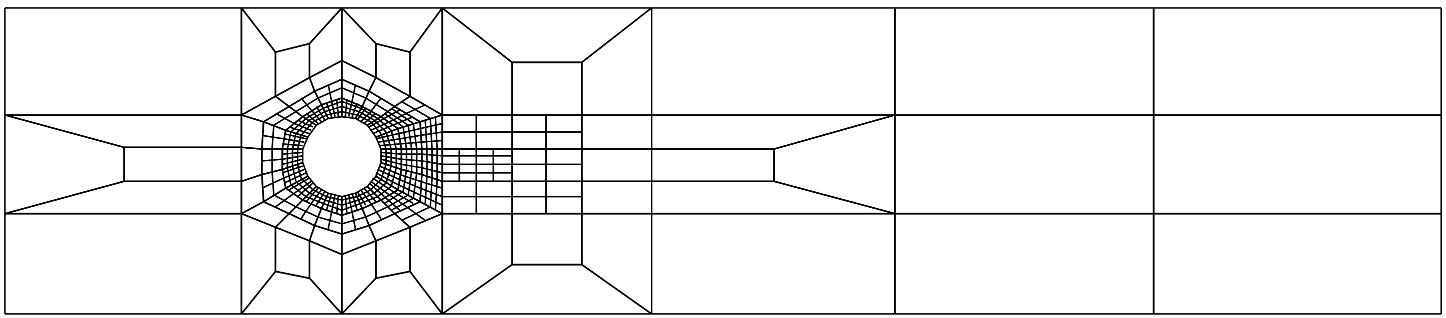








3) (patchwise) 'local' adaptivity



Example for ScaRC

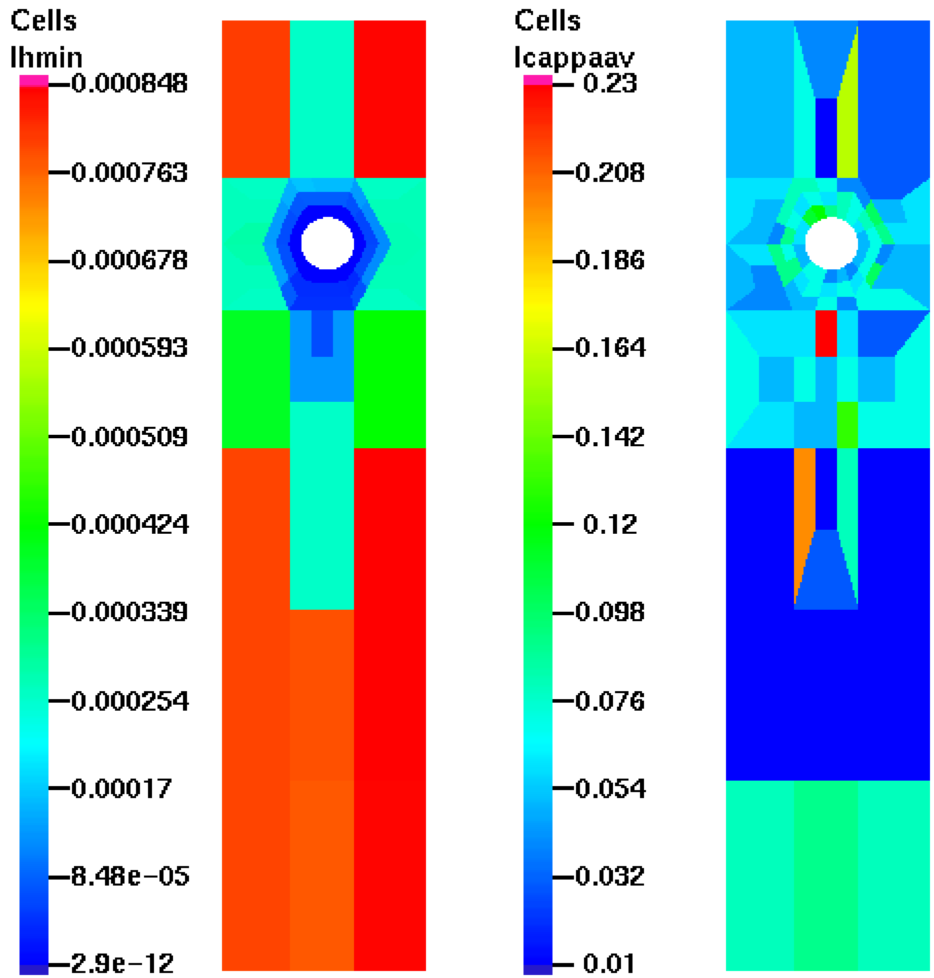
Parallel convergence rates for ScaRC-CG (2 global smoothing, V-cycle; 2 local 'MGTRI', F-cycle) with direct coarse mesh solver

#NEQ-global	AR	ρ (#IT)
1,704	3	0.03 (5)
44,544	3	0.03 (5)
431,317	10	0.09 (6)
1,722,773	10	0.08 (6)
9,344,533	10	0.09 (6)
37,366,805	10^7	0.09 (6)

Global (parallel) convergence rates

#NEQ-local	MV-V/C	MGTRI-V/C
65^2	245/1275	299/706
257^2	193/638	173/382
1025^2	168/614	146/288

Local MFLOP/s rates (AMD 'XP 1500+')



Applications

Gas-Liquid Reactors

Liquid-Solid Particulated Flow

Gas-Liquid Reactors

Drift-Flux model with mass transfer and chemical reaction

(Gas holdup ϵ, Number density n, Mass flux N, Interfacial area a_S)

$$\frac{\partial \mathbf{v}_L}{\partial t} + \mathbf{v}_L \cdot \nabla \mathbf{v}_L = -\nabla P + \nu \Delta \mathbf{v}_L - \epsilon \mathbf{g} \, , \; \nabla \cdot \mathbf{v}_L = 0 \quad , \quad \mathbf{v}_G = \mathbf{v}_L + \mathbf{v}_{\mathrm{slip}} + \mathbf{v}_{\mathrm{drift}}$$

Gas-Liquid Reactors

Drift-Flux model with mass transfer and chemical reaction

(Gas holdup ϵ, Number density n, Mass flux N, Interfacial area a_S)

$$\frac{\partial \mathbf{v}_L}{\partial t} + \mathbf{v}_L \cdot \nabla \mathbf{v}_L = -\nabla P + \nu \Delta \mathbf{v}_L - \epsilon \mathbf{g} \, , \, \nabla \cdot \mathbf{v}_L = 0 \quad , \quad \mathbf{v}_G = \mathbf{v}_L + \mathbf{v}_{\text{slip}} + \mathbf{v}_{\text{drift}}$$

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G \mathbf{v}_G) = -a_S m_\mu N \quad , \quad N = E k_L^0 (p/H - c_A)$$

$$\frac{\partial n}{\partial t} + \nabla \cdot (n \mathbf{v}_G) = 0 \quad , \quad \epsilon = \frac{\tilde{\rho}_G R T}{p m_\mu} \quad , \quad a_S = (4\pi n)^{1/3} (3\epsilon)^{2/3}$$

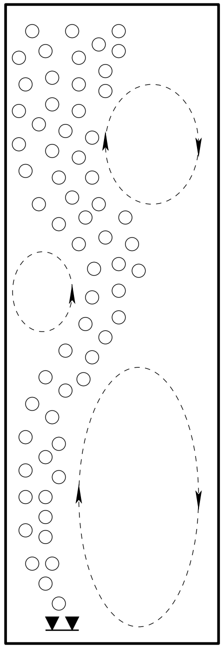
$$\frac{\partial \tilde{c}_A}{\partial t} + \nabla \cdot (\tilde{c}_A \mathbf{v}_L) = \nabla \cdot (\tilde{D}_A \nabla c_A) - \tilde{k}_2 c_A c_B + a_S N$$

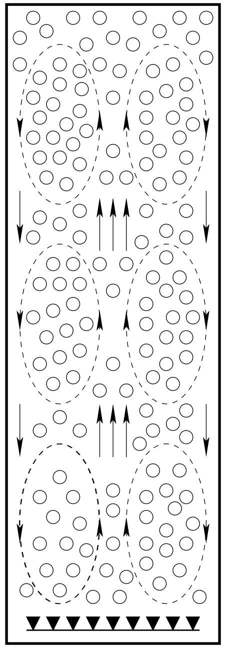
$$\frac{\partial \tilde{c}_B}{\partial t} + \nabla \cdot (\tilde{c}_B \mathbf{v}_L) = \nabla \cdot (\tilde{D}_B \nabla c_B) - \nu_B \tilde{k}_2 c_A c_B$$

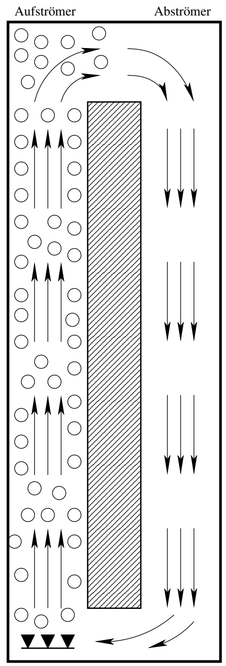
$$\frac{\partial \tilde{c}_P}{\partial t} + \nabla \cdot (\tilde{c}_P \mathbf{v}_L) = \nabla \cdot (\tilde{D}_P \nabla c_P) + \nu_P \tilde{k}_2 c_A c_B$$

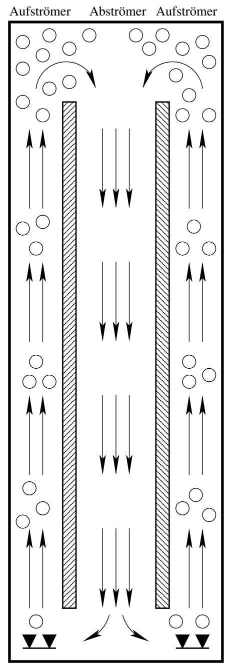
Properties of Drift-Flux Simulations

No explicit tracking of bubbles via Euler-Euler formulation









- Modelling of ‘small-scale’ effects (single bubbles) ?
- Improved Two-Fluid model ?
- Two-Phase turbulence ?   Population Balance models ?

Extension I: Two-Phase Turbulence

'Effective' viscosity ($k - \varepsilon$ model)

$$\nu_{\text{eff}} = \nu + \nu_T, \qquad \nu_T = C_\mu \frac{k^2}{\varepsilon} + \underbrace{C_g \epsilon |\mathbf{u}_{\text{slip}}| r}_{BIT}$$

with

$k = \frac{1}{2} \langle |\mathbf{u}'|^2 \rangle$ turbulent kinetic energy

$\varepsilon = \frac{\nu}{2} \langle |\nabla \mathbf{u}' + (\nabla \mathbf{u}')^T|^2 \rangle$ dissipation rate

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u} - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k + S_k - \varepsilon$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon\mathbf{u} - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k} (C_1 P_k + C_\varepsilon S_k - C_2 \varepsilon)$$

Modelling of production terms

$P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2$ shear induced turbulence

$S_k = -C_k \epsilon \nabla p \cdot \mathbf{u}_{\text{slip}}$ bubble induced turbulence

Extension II: Population Balance Models

$$\frac{\partial f}{\partial t} + \nabla \cdot (f\, \mathbf{u}_G) + \frac{\partial}{\partial m}(f\, \dot{m}) = Q - S$$

$f = f(\mathbf{x}, m, t)$	Bubble size distribution
$\mathbf{u}_G = \mathbf{u} + \mathbf{u}_{\text{slip}}(m)$	Gas velocity
$\dot{m} = E k_L^0 \left(c_A - \frac{p}{H}\right) \eta a_B$	Mass transfer rate
$Q - S$	Coalescence and breakup rates

Calculation of the local gas holdup

$$\epsilon(\mathbf{x}, t) = \frac{RT}{p\eta} \int_0^\infty m\, f(\mathbf{x}, m, t)\, dm$$

Treatment of Integro-Differential equation ???

Coupling with CFD ???

Extension III: Forces in Two-Fluid Model

$$\tilde{\rho}_G \left[\frac{\partial \mathbf{u}_G}{\partial t} + \mathbf{u}_G \cdot \nabla \mathbf{u}_G \right] = \epsilon \nabla \cdot \mathcal{S}_G + \tilde{\rho}_G \mathbf{g} + \mathbf{f}_{\mathrm{int}}$$

$$\tilde{\rho}_L \left[\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L \right] = (1-\epsilon) \nabla \cdot \mathcal{S}_L + \tilde{\rho}_L \mathbf{g} - \mathbf{f}_{\mathrm{int}}$$

with: $\mathbf{f}_{\mathrm{int}} = \mathbf{f}_D + \mathbf{f}_{VM} + \mathbf{f}_L$

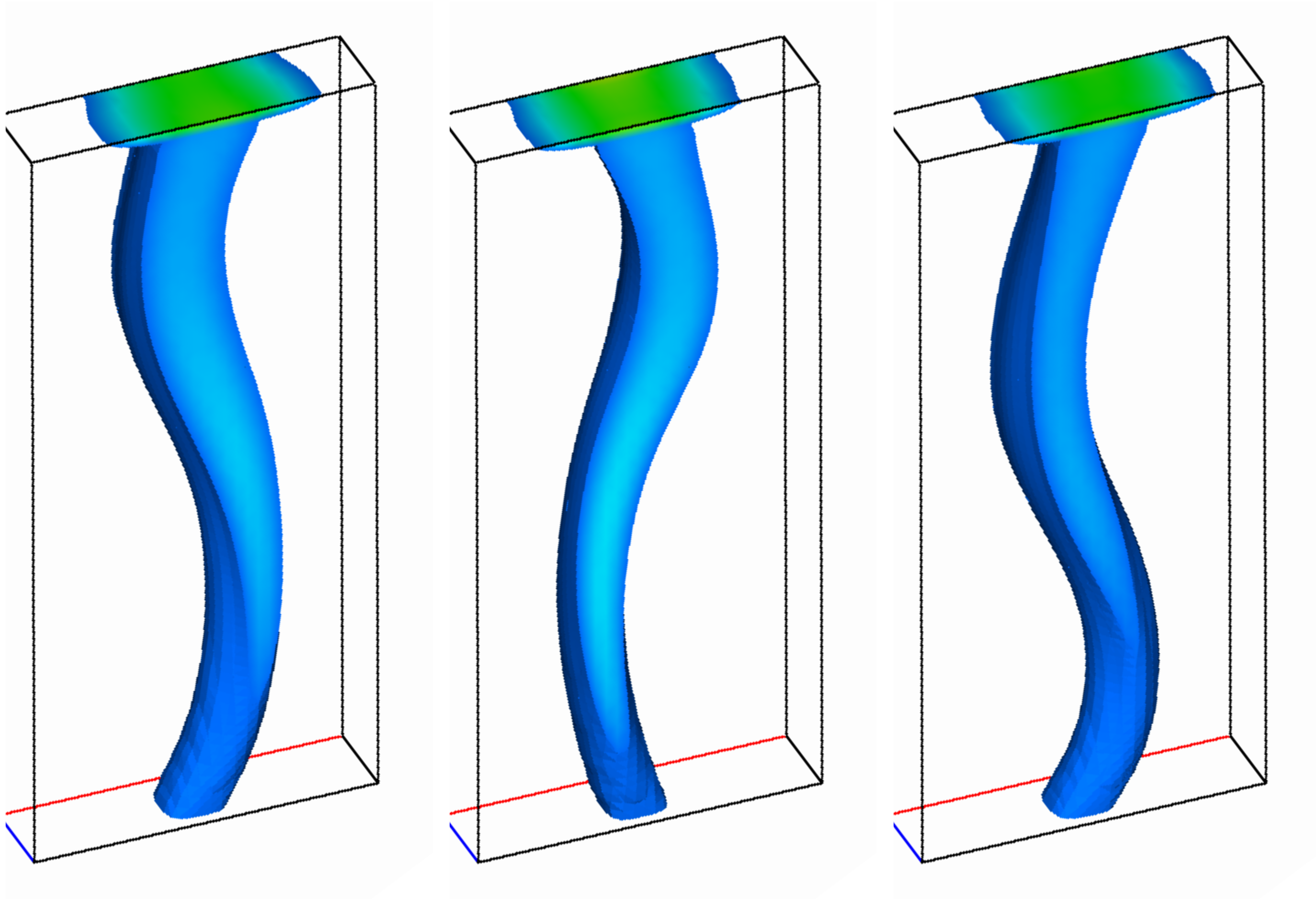
$$\mathbf{f}_D = -\epsilon C_D \frac{3}{8} \frac{\rho_L}{r} |\mathbf{u}_G - \mathbf{u}_L| (\mathbf{u}_G - \mathbf{u}_L) \quad \textit{(drag force)}$$

$$\mathbf{f}_{VM} = -\epsilon C_{VM} \rho_L \left(\frac{\mathrm{d}\mathbf{u}_G}{\mathrm{d}t} - \frac{\mathrm{d}\mathbf{u}_L}{\mathrm{d}t} \right) \quad \textit{(virtual mass force)}$$

$$\mathbf{f}_L = -\epsilon C_L \rho_L (\mathbf{u}_G - \mathbf{u}_L) \times (\nabla \times \mathbf{u}_L) \quad \textit{(lift force)}$$

Recent Application

‘ADMIRE Benchmark’ (cf. Pfleger/BASF)



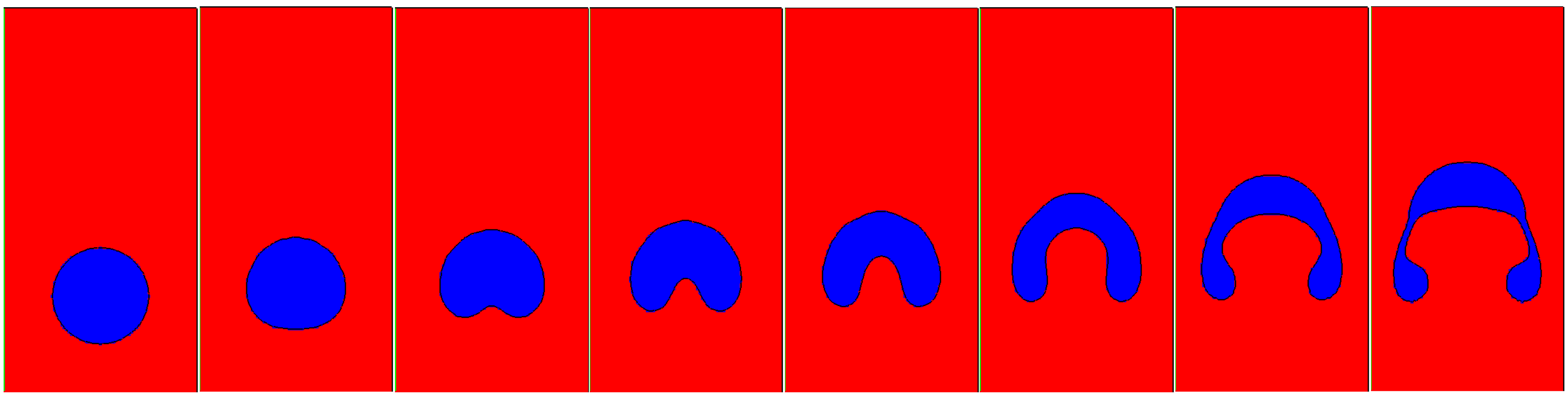
$\Rightarrow$ **Full Model !?**

Extension IV: ‘Single Bubble’

$$\rho_i \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) - \nabla \cdot (2\mu_i \mathbf{S}) + \nabla p = \rho_i \mathbf{g}$$

$$\nabla \cdot \mathbf{v} = 0 \qquad\qquad \text{in}\, \Omega_i \,,\; i = 1, 2, \ldots$$

$$[\mathbf{v}]_{|_\Gamma} \cdot \mathbf{n} = 0 \,, \qquad\qquad - [-p\mathbf{I} + 2\mu(\mathbf{x})\mathbf{S}]_{|_\Gamma} \cdot \mathbf{n} = \kappa\sigma\mathbf{n}$$



$\rightarrow$ **(Implicit) Reconstruction via ‘Indicator function’ ϕ !**

Level-Set vs. VOF Method

1. "Easy" integration of surface tension (in weak formulation)
2. High order schemes due to high smoothness

$\rightarrow$ **Reinitialisation:** Signed distance function

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \quad , \quad |\nabla \phi| = 1$$

(Nonstationary) Reformulation:

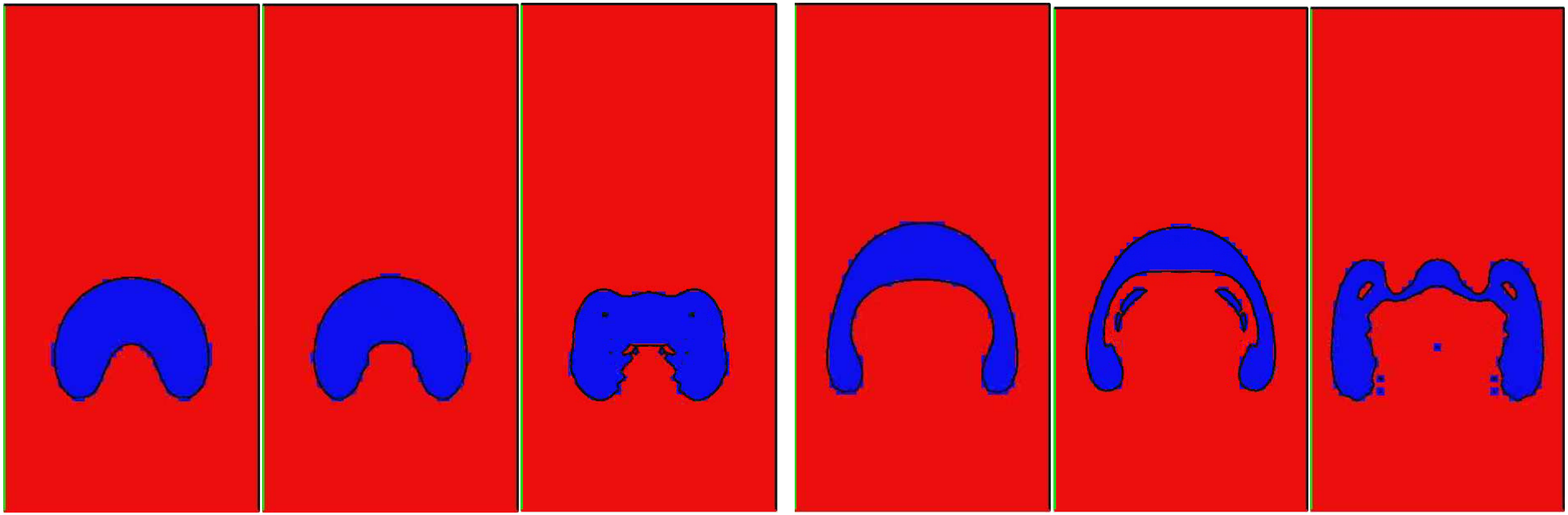
$$\frac{\partial \phi}{\partial t} = sign(\phi_{old})(1 - |\nabla \phi|)$$

$$\frac{\partial \phi}{\partial t} + sign(\phi_{old})|\nabla \phi| = sign(\phi_{old})$$

$$\frac{\partial \phi}{\partial t} + sign(\phi_{old})\frac{\nabla \phi \cdot \nabla \phi}{|\nabla \phi|} = \frac{\partial \phi}{\partial t} + \mathbf{w} \cdot \nabla \phi = sign(\phi_{old})$$

Problems for Discretization

TG for Level Set with/without reinitialisation vs. VOF



→ **Oscillation free + highly accurate discretization !!!**

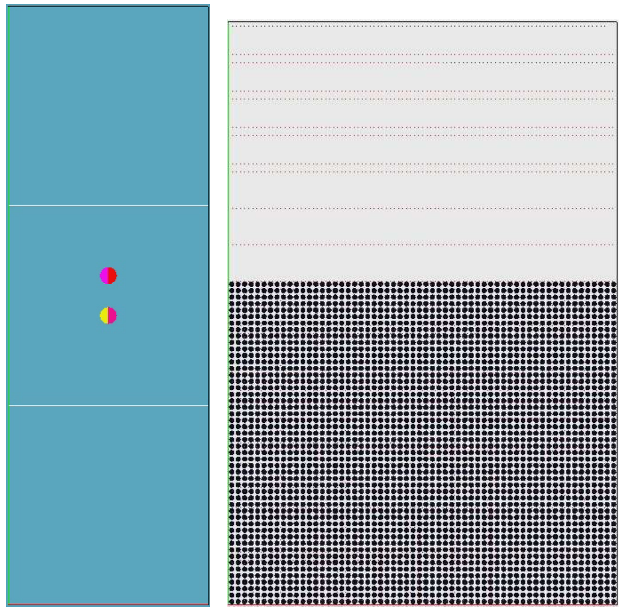
Future Challenges

- A posteriori error control + adaptive mesh refinement
- Robust + accurate FEM discretization of higher order
- Efficient solution of (nonlinear) ‘reinitialisation part’
- (Implicit) coupling with CFD part
- **Preserve the high efficiency of FEATFLOW !**

Liquid-Solid Flow

Development of simulation tools for the understanding of:

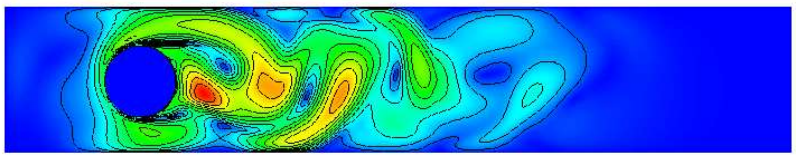
1. Interaction of solid particles with flow ($\rightarrow$ ‘many complex objects’ !)
2. Behaviour of non-newtonian fluids

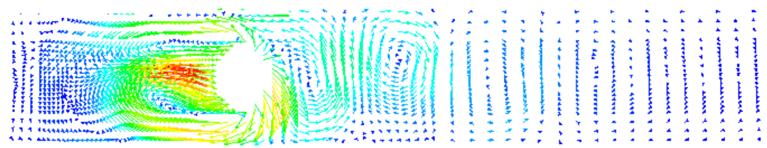


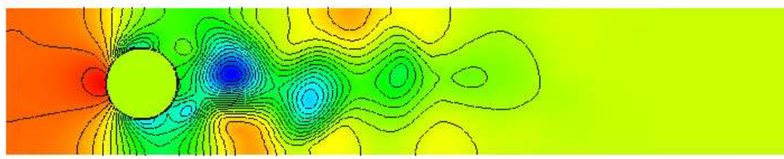
- Exploit the high efficiency and accuracy of FEATFLOW !
- ‘Fictitious Boundary Method’ on fixed meshes + Operator-Splitting

The ‘Fictitious Boundary Method’

1. Use (rough) boundary parametrization and locally refined coarse mesh for large-scale structures!
2. Describe fine-scale structures and time-dependent objects via (level-dependent) inner points!
3. Use projectors onto the "right" b.c.’s in iterative components!







Computational mesh independent of ‘internal objects’

How to Calculate (Surface) Forces?

Define a function α as

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

Remark: $\nabla \alpha_p = 0$ everywhere except at wall surface of the particles, and equal to the normal vector $\mathbf{n_p}$ defined on the global grid

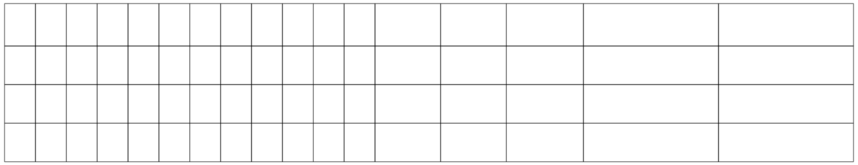
$$\mathbf{n}_p = \nabla \alpha_p \tag{0}$$

Force acting on the wall surface of the particles can be computed by

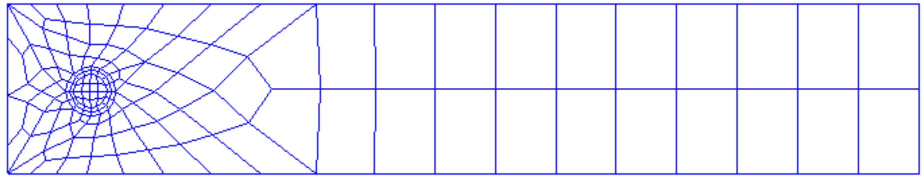
$$F_p = \int_{\Gamma_p} \sigma \cdot \mathbf{n}_p \, d\,\Gamma_p = \int_{\Omega_T} \sigma \cdot \nabla \alpha_p \, d\,\Omega_T$$

with $\bar{\Omega}_T = \bar{\Omega}_f \cup \bar{\Omega}_p$ (analogously for the torque)

‘Flow around Cylinder’ Benchmark

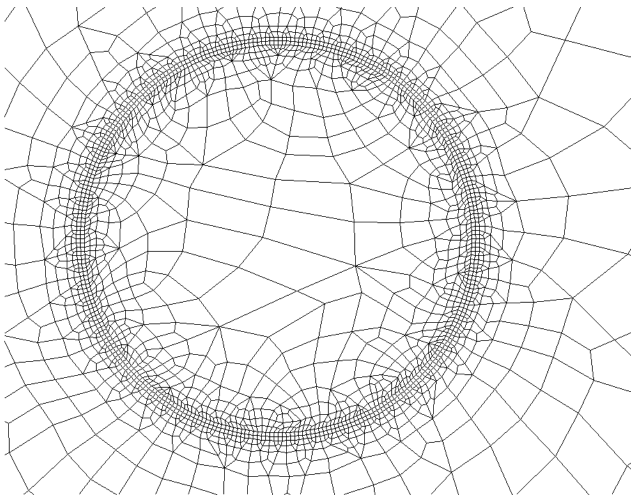


LEVEL 6 ≈ 280.000 elements



LEVEL 6 ≈ 150.000 elements

LEVEL	ch. mesh I	ch. mesh II	ch. mesh I	ch. mesh II
3	0.5529+01	0.5569+01	0.1216-01	0.2443-03
4	0.5353+01	0.5575+01	0.1074-01	0.0014-01
5	0.5427+01	0.5572+01	0.6145-02	0.0812-01
6	0.5501+01	0.5578+01	0.9902-02	0.1020-01
	C_d = 0.55795+01		C_l = 0.10618-01	



LEVEL	C_d	C_l
2	0.55201+01	0.1057-01
3	0.55759+01	0.1036-01
4	0.55805+01	0.1041-01

LEVEL 2 ≈ 10.000 elements

Application to Liquid-Solid Flow

Consider flow of N solid particles in a fluid with density ρ and viscosity μ

Denote by $\Omega_f(t)$ the domain occupied by the fluid at time t, and by $\Omega_p(t)$ the domain occupied by the particle p at time t

Fluid flow is modeled by the Navier-Stokes equations in $\Omega_f(t)$

$$\rho\left(\frac{\partial\,\mathbf{u}}{\partial\,t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)-\nabla\sigma=0\,,\qquad \nabla\cdot\mathbf{u}=0$$

where σ is the total stress tensor in the fluid phase, which is defined as

$$\sigma(X,\,t)=-p\,\mathbf{I}+\mu\left[\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right]$$

Model for Particle Motion

Motion of particles is modeled by the Newton-Euler equations, i.e., the translational velocities U_p and angular velocities ω_p of the p-th particle satisfy

$$M_p \frac{d\,U_p}{d\,t} = F_p + (\Delta M_p)\ \mathbf{g}\,, \qquad \mathbf{I}_p \frac{d\,\omega_p}{d\,t} + \omega_p \times (\mathbf{I}_p\,\omega_p) = T_p$$

with M_p the mass of the p-th particle $(p = 1, \cdots, N)$; $\mathbf{I}_p$ the moment of inertia tensor of the p-th particle; ΔM_p the mass difference between the mass M_p and the mass of the fluid occupying the same volume

F_p and T_p are the hydrodynamical forces and the torque at mass center acting on the p-th particle

$$F_p = \int_{\Gamma_p} \sigma \cdot \mathbf{n}_p\, d\,\Gamma_p\,, \qquad T_p = \int_{\Gamma_p} (X - X_p) \times (\sigma \cdot \mathbf{n}_p)\ d\,\Gamma_p$$

with X_p the position of the center of gravity of the p-th particle, $\Gamma_p = \partial\Omega_p$ the boundary of p-th particle, $\mathbf{n}_p$ is the unit normal vector on the boundary Γ_p

Interaction between Particle and Fluid

No-slip boundary conditions at interface Γ_p between particle and fluid, i.e., for any $X \in \Gamma_p$, the velocity $\mathbf{u}(X)$ is defined by

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

Position X_p of the p-th particle and its angle θ_p are obtained by integration of the kinematic equations

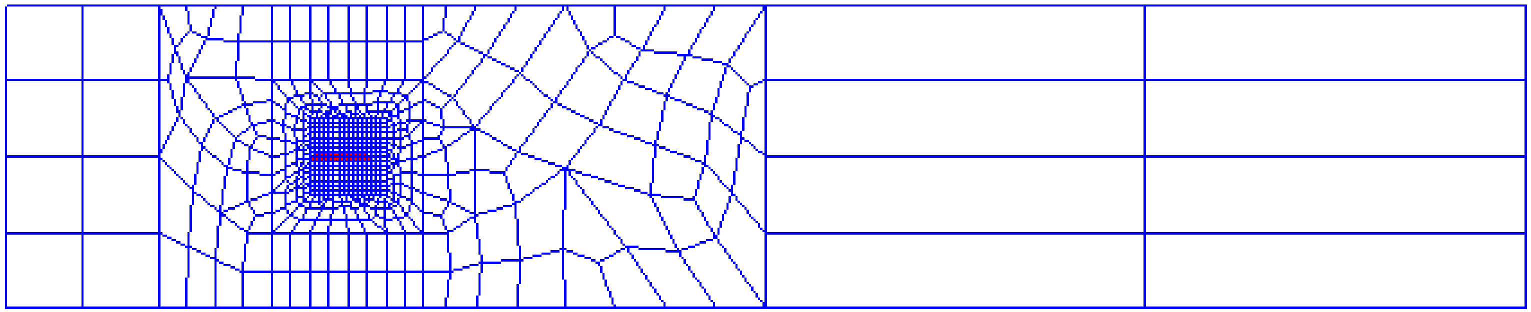
$$\frac{d\,X_p}{d\,t} = U_p\,, \qquad \frac{d\,\theta_p}{d\,t} = \omega_p$$

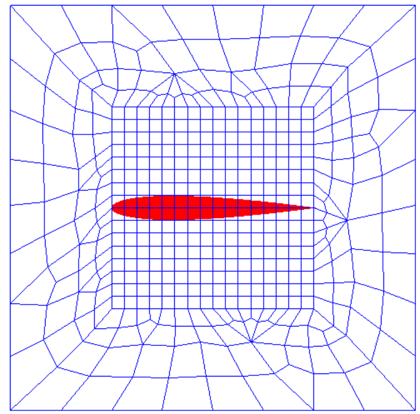
Operator-Splitting Approach

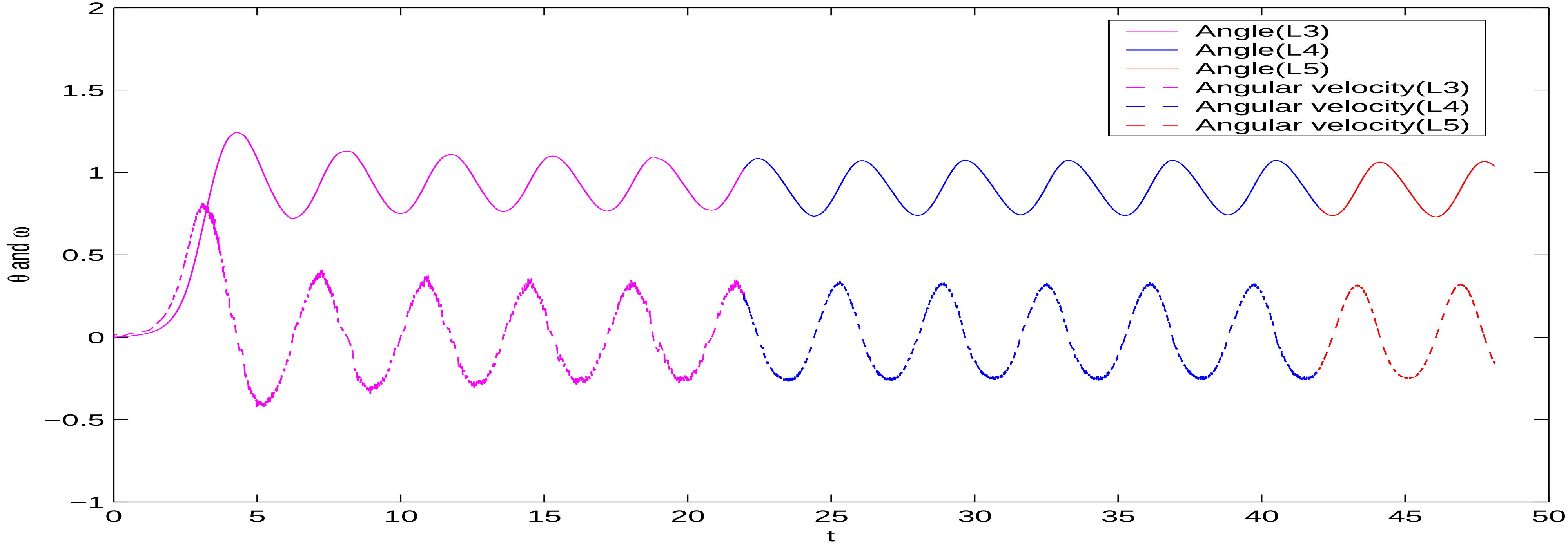
The algorithm for $t^n \rightarrow t^{n+1}$ consists of the following 3 substeps:

1. Fluid velocity and pressure: $NSE(\mathbf{u}_f^{n+1}, p^{n+1}) = BC(\Omega_p^n, \mathbf{u}_p^n)$
2. Calculate hydrodynamic forces: $\mathbf{F_p}^{n+1}$
3. Calculate velocity of particles: $\mathbf{u}_p^{n+1} = g(\mathbf{F_p}^{n+1})$
4. Update position of particles: $\Omega_p^{n+1} = f(\mathbf{u}_p^{n+1})$

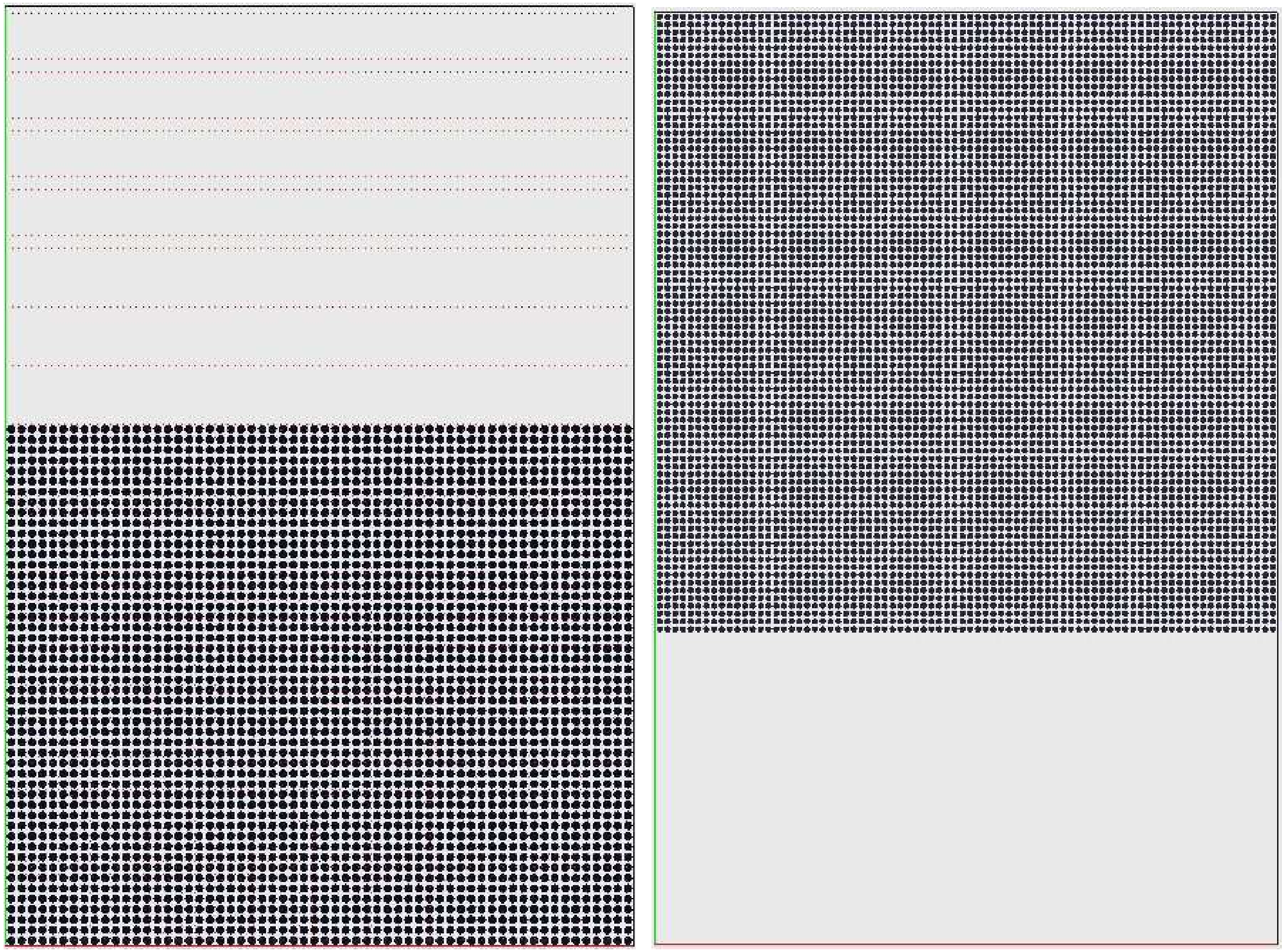
Rotating Airfoil (cf. Glowinski)







‘Balls raising/falling in a Box’



Challenges (for 'many' Particles)

- Adaptive time-stepping + locally adaptive grid alignement
- Accurate calculation of forces ($\rightarrow \alpha \in [0, 1]$) for 'complex' shapes
- Efficient data structures for treating the particles/particle forces
- (Better) Collision models
- Nonlinear fluids (*'Kissing, Drafting, Thumbling'*)
- 3D
- "100.000" particles...

Mathematical Key Techniques

(Special) Implicit FEM schemes in space/time, adaptivity/error control, FCT/TVD stabilization, hierarchical Newton/multigrid solvers,...

Hardware-oriented Numerics for PDE, High Performance Computing techniques,...

‘Higher (guaranteed) accuracy with less unknowns via hierarchical solvers with ‘optimal’ numerical complexity while exploiting the available huge sequential/parallel GFLOP/s rates’

Conclusions:

There is a huge potential...

There is much to do...:-)