FEM Techniques for Incompressible Flow Problems with Time-Dependent Interfaces

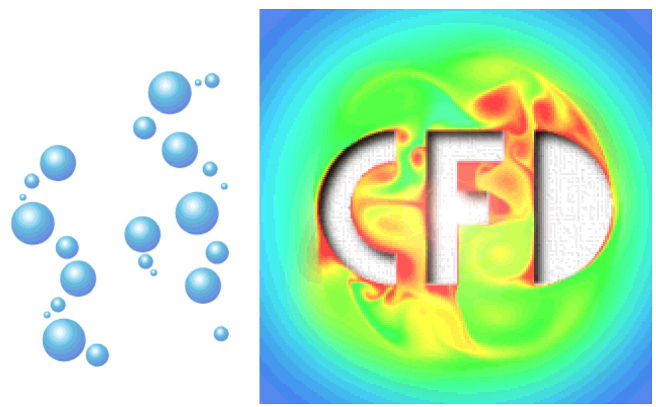
Stefan Turek

Institut für Angewandte Mathematik, Univ. Dortmund

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

`http://www.featflow.de`

- **Liquid - (rigid) solid interfaces**
- [Liquid - (elastic) solid interfaces]
- [Liquid - gas interfaces]
- [Liquid - liquid interfaces]



Fluid - (Rigid) Solid Interfaces

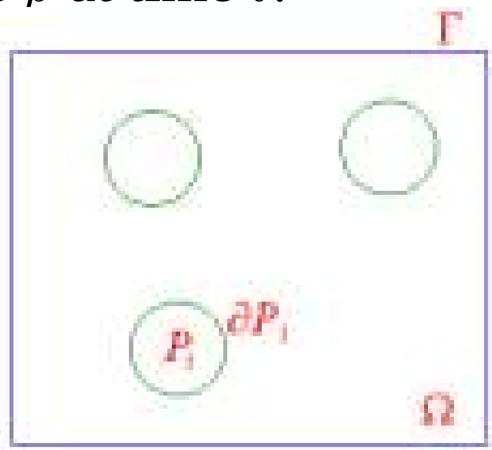
1. Sedimentation (e.g. sand flow in river)
2. Suspensions (fluidized beds)
3. Lubricated transport (e.g. coal slurries in water)
4. Hydraulic fracturing (separation process using cyclones)
5. Gas-liquid reactors (fragmentation/coalescence)
6. . . .

Aim: Exploit the high efficiency of operator-splitting techniques for highly (!!!) time-dependent configurations in FeatFlow

How far can we come with ”simple Mathematics” ???

Model for Fluid 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 modelled by the **Navier-Stokes equations** in $\Omega_f(t)$

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

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 (I)

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

$$M_p \frac{d\,U_p}{d\,t} = F_p + F_p' + (\Delta M_p)\ \mathbf{g}\,, \qquad \mathbf{I}_p \frac{d\,\omega_p}{d\,t} + \omega_p \times (\mathbf{I}_p\,\omega_p) = T_p$$

with M_p the mass of the p-th particle $(p = 1, \ldots, N)$;
$\mathbf{I}_p$ 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.

Model for Particle Motion (II)

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

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

and F_p' are **collision forces**.

X_p is the position of the center of gravity of the p-th particle;
$\Gamma_p = \partial\Omega_p$ the boundary of the 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 particles and fluid, i.e., for any $X \in \Gamma_p$, the velocity $\mathbf{u}(X)$ is defined by:

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

The **position** X_p of the p-th particle and its **angle** θ_p are obtained by integration of the kinematic equations:

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

Coupling between Fluid and Particle

1. Implicit coupling (”Distributed Lagrange Multiplier/Fictitious Domains”)

   Idea: Calculate the fluid on the complete fluid-solid domain; the solid domain is constrained to move with the rigid motion; mutual forces between solid and fluid are cancelled.

   - Body-force-DLM (*Glowinski*, *Pan*, *Hesla*, *Joseph and Périaux* (1999)): the constraint of rigid-body motion is represented by $\mathbf{u} = \mathbf{U} + \omega \times \mathbf{r}$
   - Stress-DLM (*Patankar*, *Singh*, *Joseph*, *Glowinski and Pan* (2000)): the constraint of rigid-body motion is represented by a stress field just as there is pressure in fluid

2. Explicit coupling

   $\boxed{t^n \text{ fluid}} \rightarrow \boxed{t^n \text{ force on solid}} \rightarrow \boxed{t^{n+1} \text{ solid}} \rightarrow \boxed{t^{n+1} \text{ fluid}} \ldots$

   - FVM-fictitious domain methods (*Duchanoy and Jongen* (2003))
   - **FEM-fictitious boundary methods** (*Turek*, *Wan and Rivkind*)

Further Classification

1. **Eulerian approach:** fixed meshes!
   Use a ”fixed” mesh that covers the whole domain where the fluid may be present.

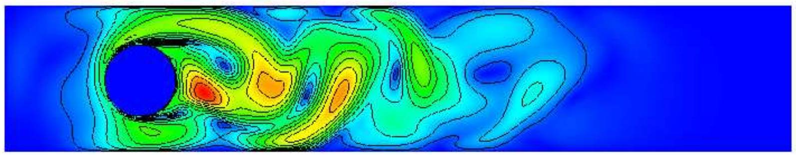
   - The distributed Lagrange multiplier/fictitious domain method
   - **FEM-fictitious boundary method**
   - FVM-fictitious domain method

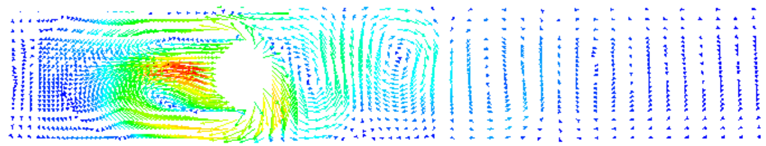
2. **Lagrangian approach:** moving meshes!
   Based on a moving mesh which follows the motion of the fluid boundary.

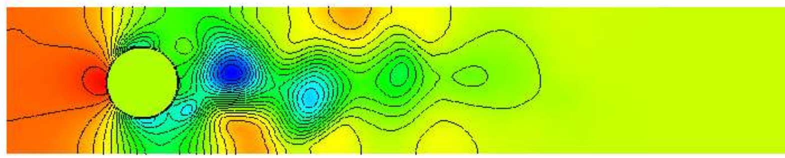
   - Arbitrary Lagrangian Eulerian (*Hu, Joseph and Crochet* (1992), *Johnson and Tezduyar* (1996))
   - Fat boundary method (*Maury* (2001))

The ‘Fictitious Boundary Method’

1. Describe fine-scale geometrical structures and time-dependent objects via (level-dependent) inner ”boundary points”!
2. Use projectors onto the ”right” b.c.’s in iterative components!







Computational mesh (can be) <u>independent</u> of ‘internal objects’

How to Calculate (Surface) Forces?

Hydrodynamic forces and torque acting on the i-th particle:

$$\mathbf{F}_i = -\int_{\partial P_i} \sigma \cdot \mathbf{n}_i \, d\Gamma_i \,, \qquad T_i = -\int_{\partial P_i} (\mathbf{X} - \mathbf{X}_i) \times (\sigma \cdot \mathbf{n}_i) \; d\Gamma_i$$

- Reconstruction of the shape is only first order accurate
  $\Rightarrow$ local grid adaptivity or alignment
  $\Leftarrow$ ”only” averaged/integral quantities are required
- But: The FBM can only decide ”INSIDE” or ”OUTSIDE”

‘Replace the surface integral by a volume integral’

Calculation of Hydrodynamic Forces

Define auxiliary 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:

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

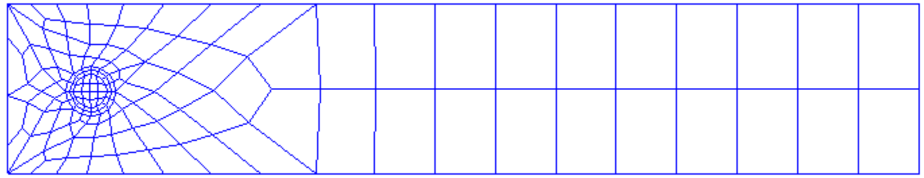
Force acting on the wall surface of the particles can be computed by

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

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

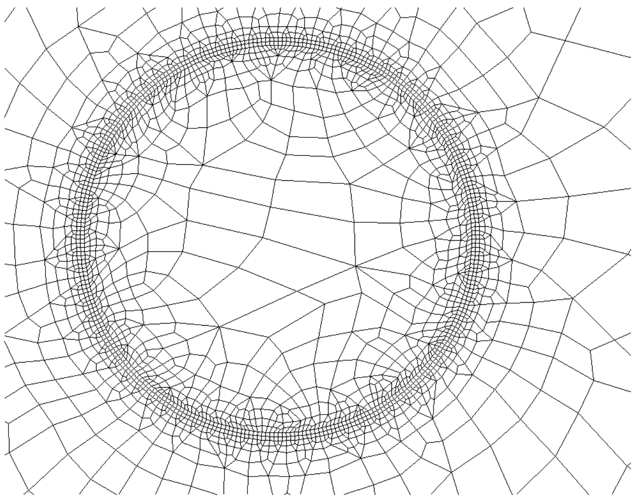
Evaluation of Force Calculations

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 4 ≈ 150.000 elements

(Explicit) Operator-Splitting Approach

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

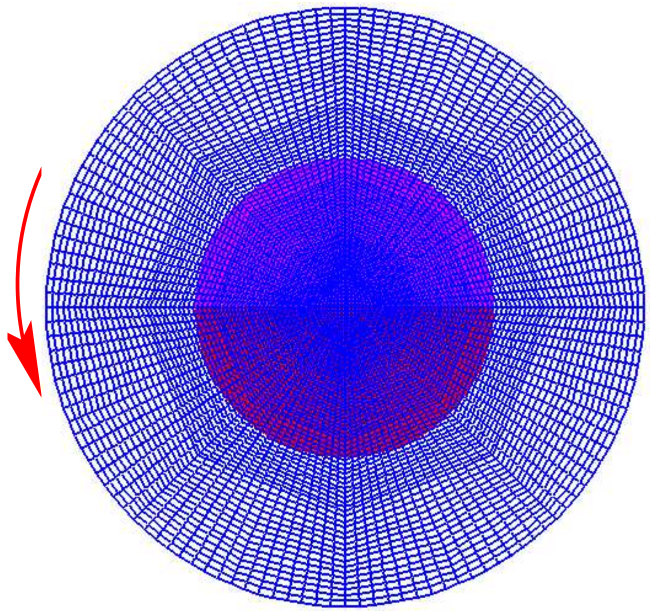
1. Fluid velocity and pressure: $NSE(\mathbf{u}_f^{n+1}, p^{n+1}) = BC(\Omega_p^n, \mathbf{u}_p^n)$
2. Calculate hydrodynamic forces: $\mathbf{F_p}^{n+1}$
3. Calculate velocity of particles: $\mathbf{u}_p^{n+1} = g(\mathbf{F_p}^{n+1})$
4. Update position of particles: $\Omega_p^{n+1} = f(\mathbf{u}_p^{n+1})$

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

Numerical Examples

'One particle in a rotating circular container'

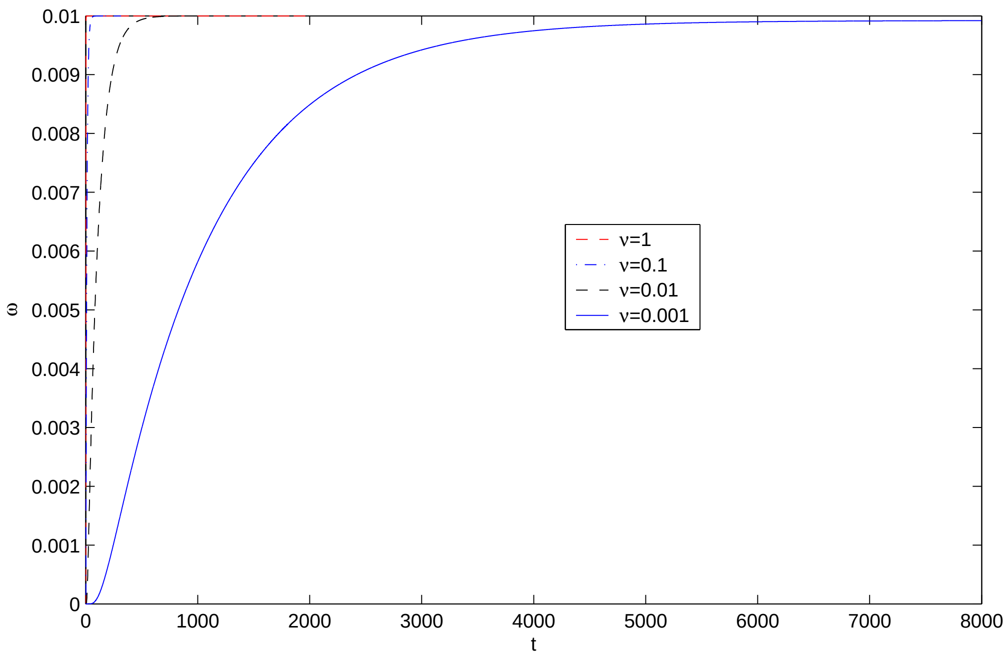
$\Omega = 0.01$



$$R_\Omega = 2.0, \quad R_p = 1.0$$

Numerical Examples

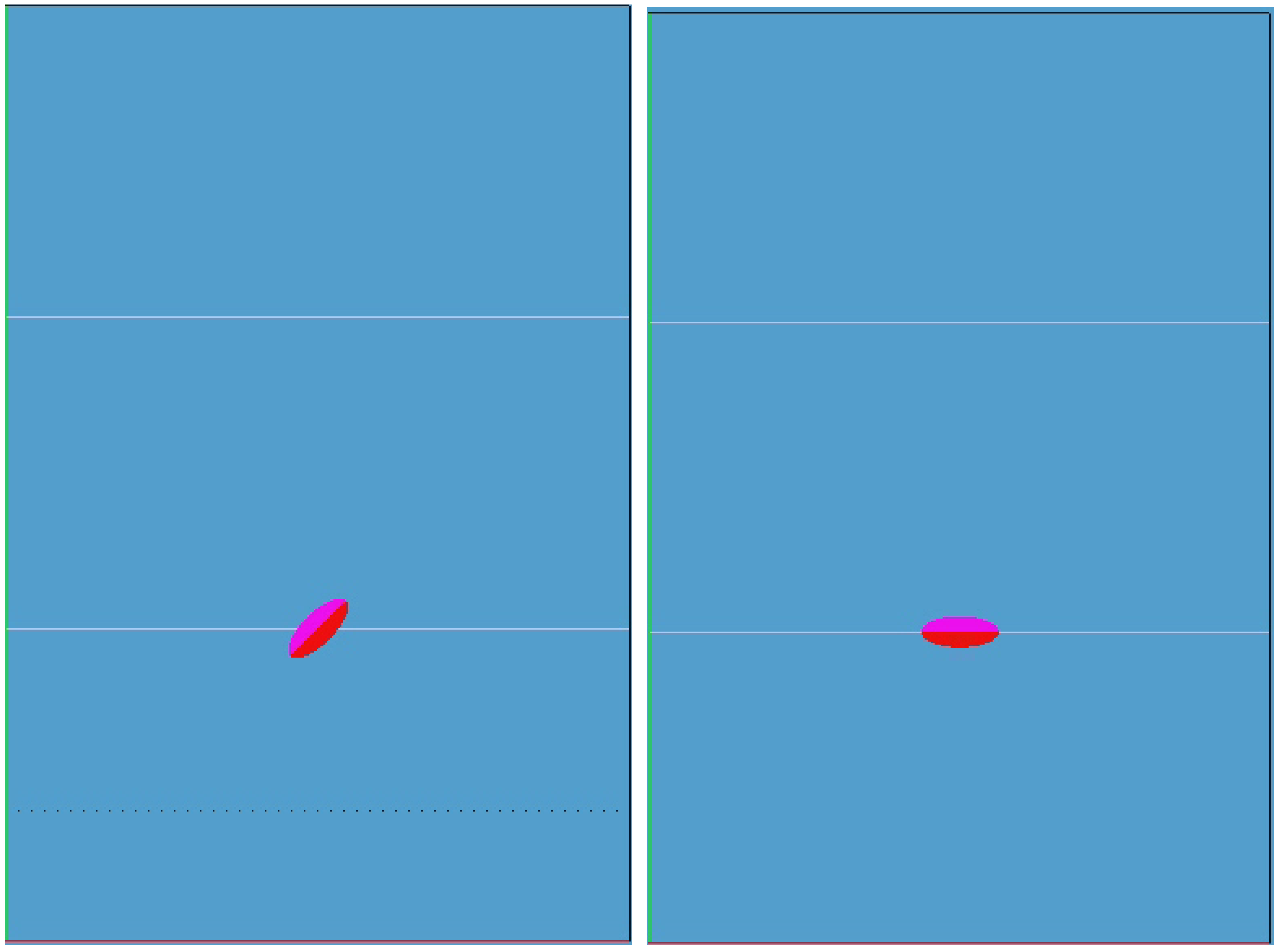
‘One particle in a rotating circular container’



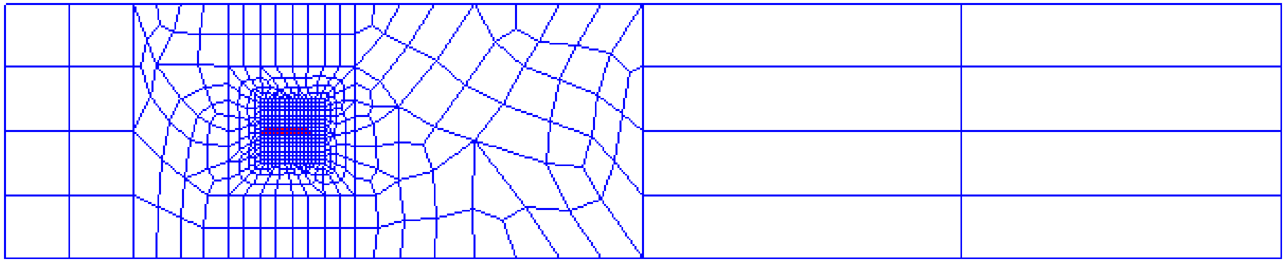
viscosity ν	Terminal angular velocity ω_p	Time reaching the steady state
0.001	0.0099185	7000.0
0.01	0.0099989	600.0
0.1	0.0099998	60.0
1.0	0.0099999	10.0

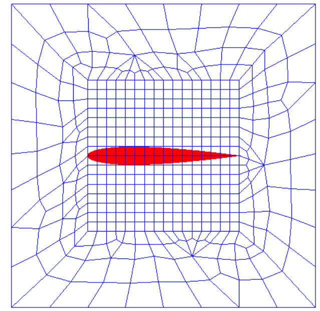
Numerical Examples

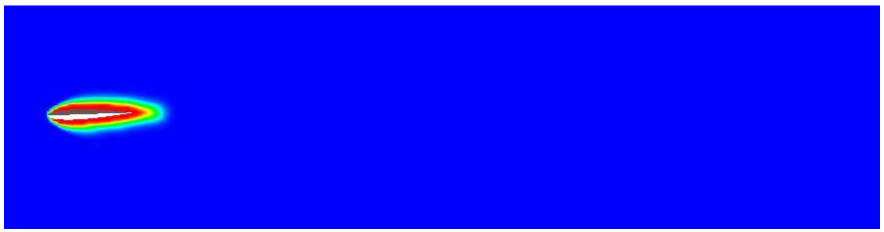
‘One ellipse falling in an (infinite) channel’

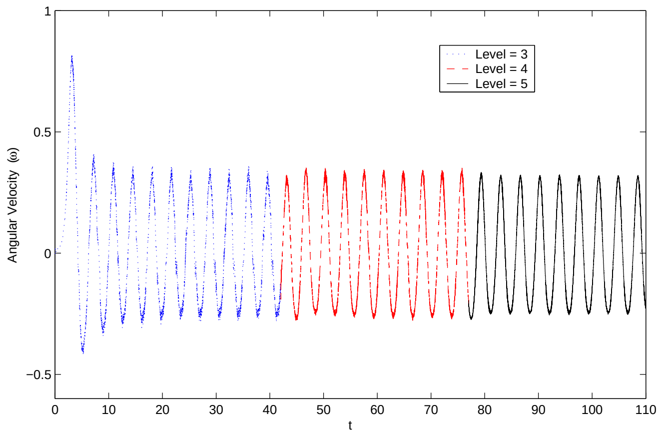


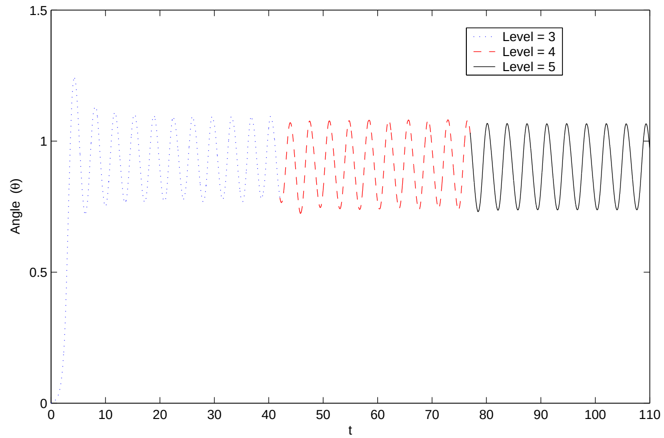
‘Viscous flow around a moving airfoil’ (Glowinski)





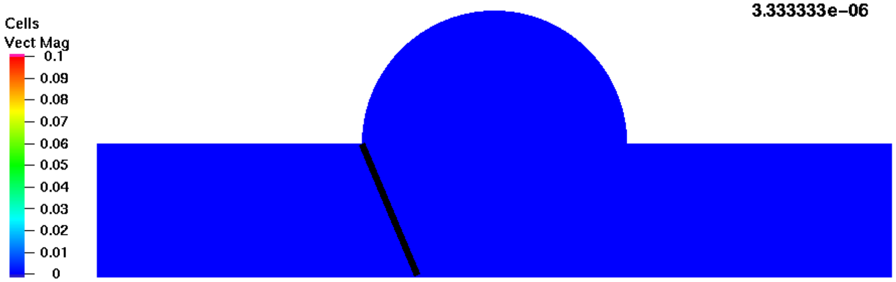


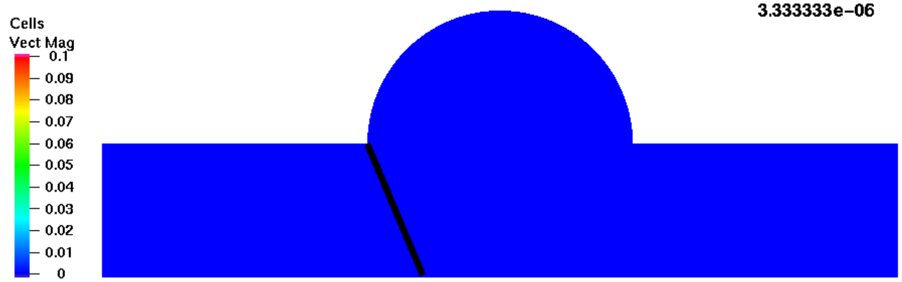




Numerical Examples

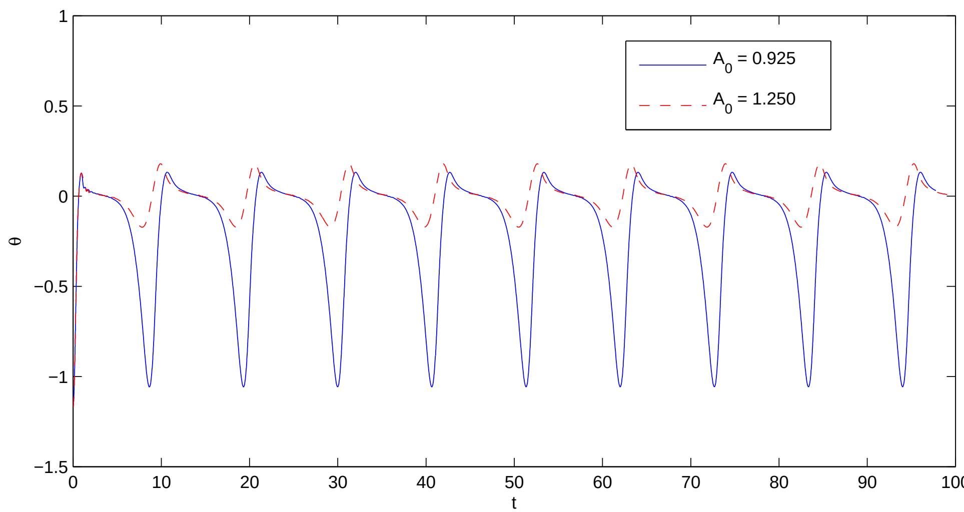
‘(Prototypical) Heart Valve’



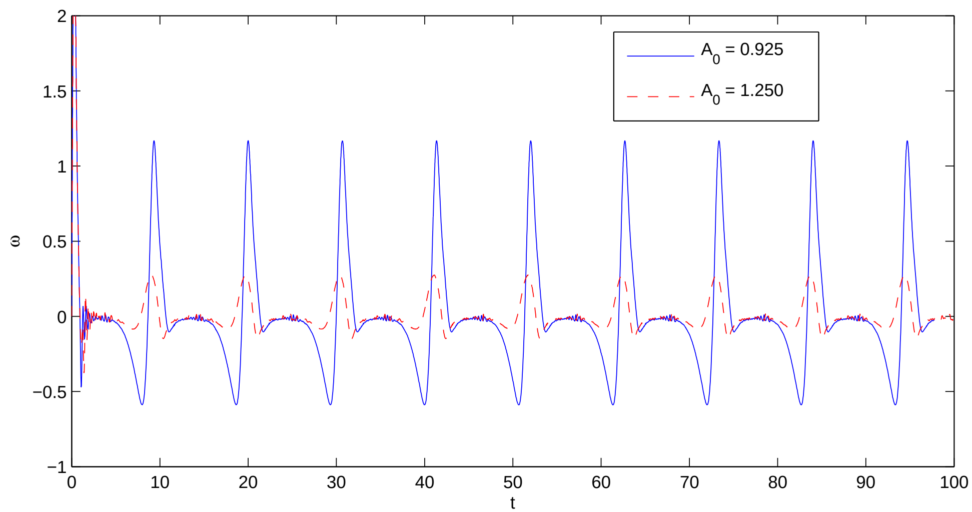


Velcoity $(A_0 = 0.925)$ Velocity $(A_0 = 1.250)$

Inlet velocity $U = 9.828(A_0 + \sin(\frac{1.85\,t}{\pi}))$



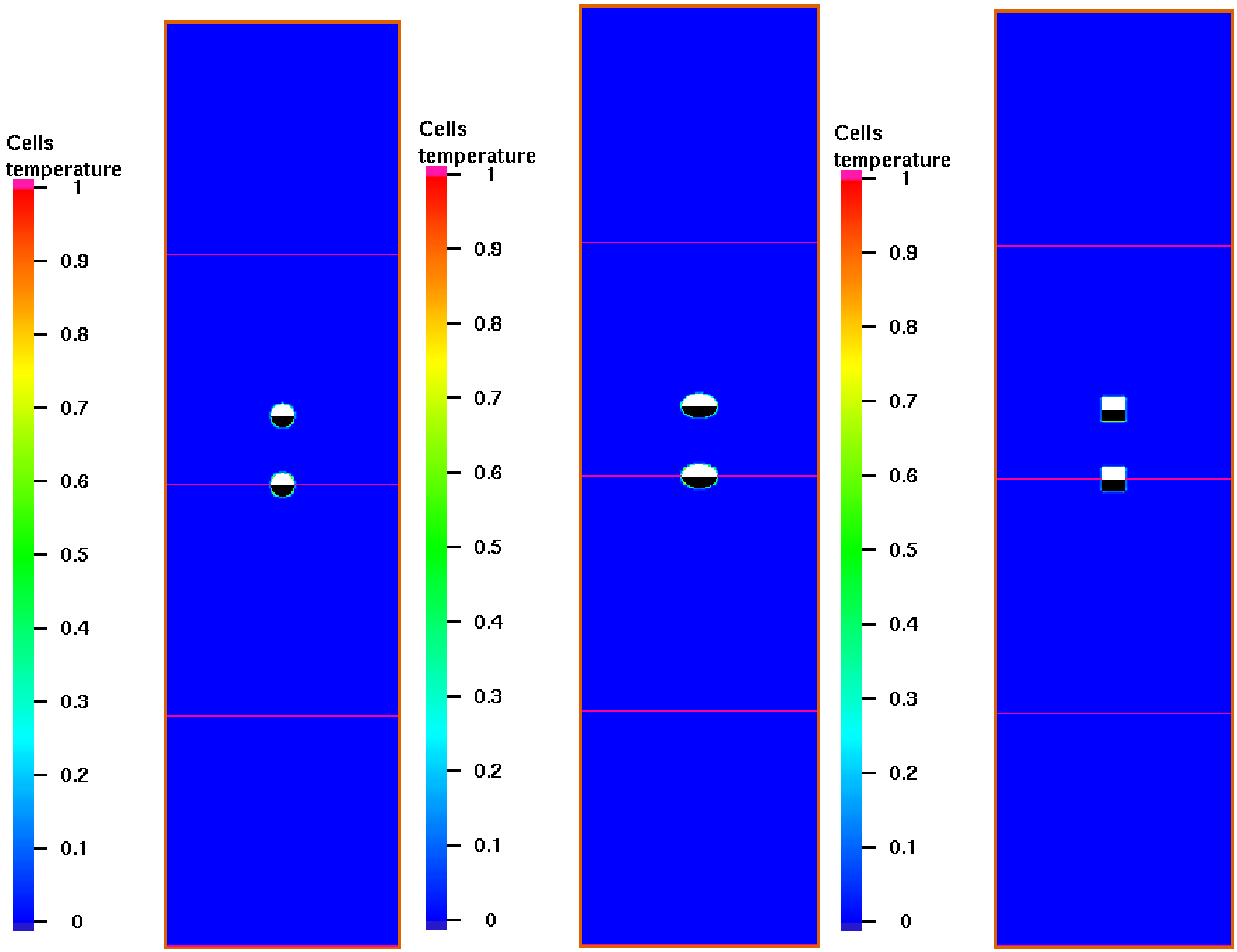
(a) angle



(b) angular velocity

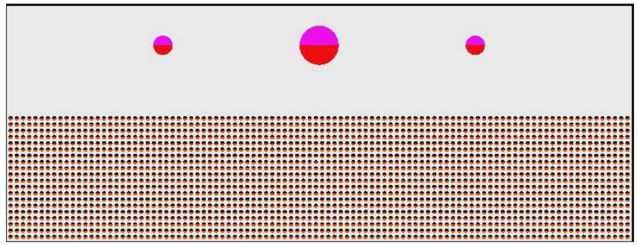
Numerical Examples

‘Kissing, Drafting, Thumbling’

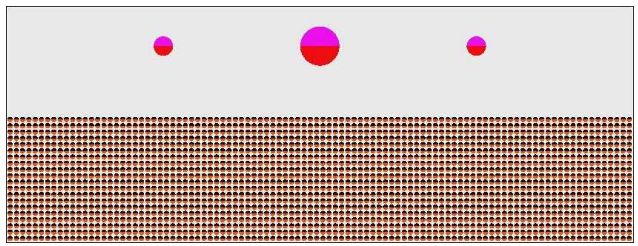


Numerical Examples

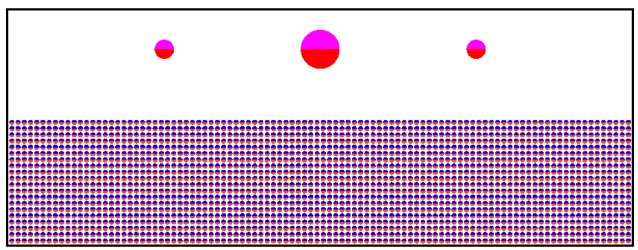
'Impact of heavy balls on 2000 small particles'



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



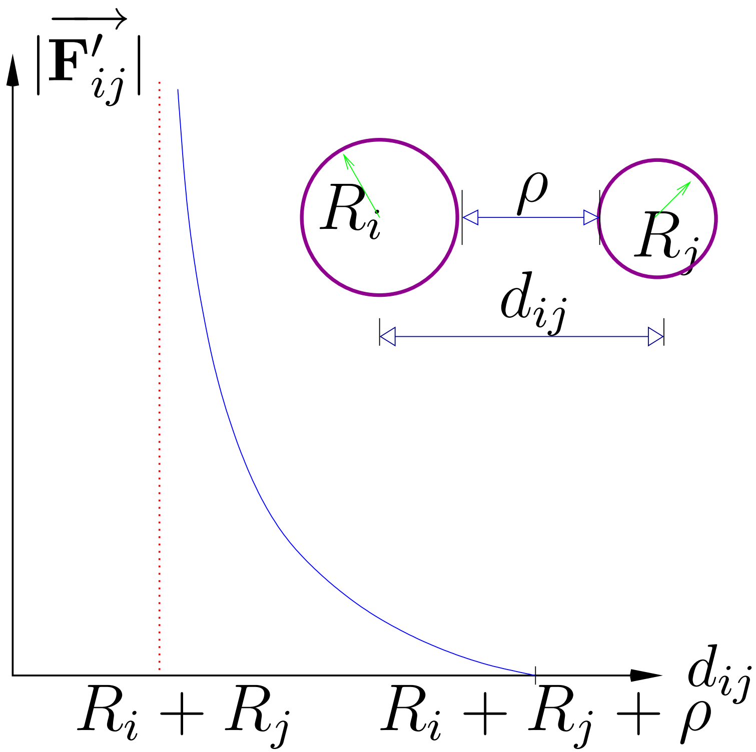
$\rho_f = 1,\ \rho_{bd} = 2,\ \rho_{sp} = 2$



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

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.



$$
\begin{cases}
|\overrightarrow{\mathbf{F}'_{ij}}| = 0 & \text{if } d_{ij} \geq R_i + R_j + \rho\,, \\
|\overrightarrow{\mathbf{F}'_{ij}}| = d_{ij}/\varepsilon & \text{if } d_{ij} = R_i + R_j\,.
\end{cases}
$$

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:

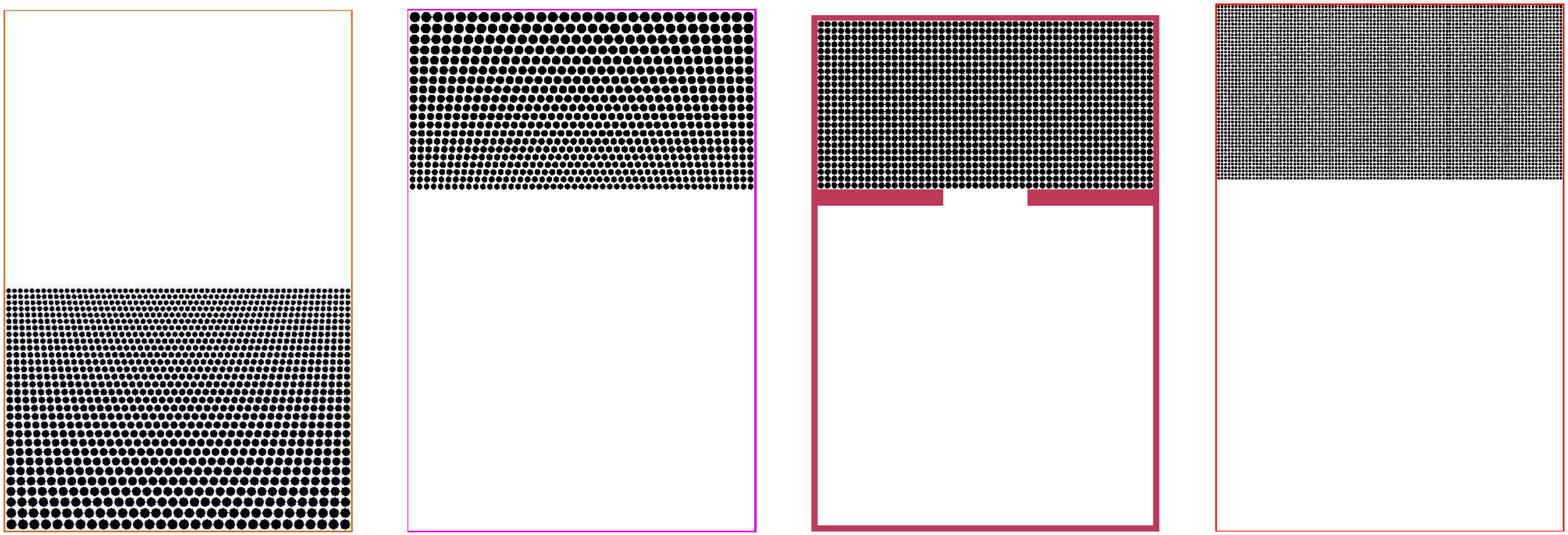
$$\mathbf{F}_{i,j}^{P} = \begin{cases} 0 & \text{for } d_{i,j} > R_i + R_j + \rho \\ \frac{1}{\epsilon_P}(\mathbf{X}_i - \mathbf{X}_j)(R_i + R_j + \rho - d_{i,j})^2 & \text{for } R_i + R_j \leq d_{i,j} \leq R_i + R_j + \rho \\ \frac{1}{\epsilon'_P}(\mathbf{X}_i - \mathbf{X}_j)(R_i + R_j - d_{i,j}) & \text{for } d_{i,j} \leq R_i + R_j \end{cases}$$

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

$$\mathbf{F}'_i = \sum_{j=1, j\neq i}^{N} \mathbf{F}_{i,j}^{P} + \mathbf{F}_i^{W}$$

Numerical Examples

‘Fluidization/Sedimentation of many particles’



Complete Algorithm

The complete algorithm /($t_n \to t_{n+1}$) for the coupled fluid-solid system can be summarized as follows:

1. Given the position and velocity of the particles at time t_n.
2. Set the fictitious boundary and its boundary condition for the fluid.
3. Solve the fluid equations to get the fluid velocity and the pressure.
4. Calculate the hydrodynamic forces acting on every particle.
5. Calculate the motion of the solid particles.
6. Check if collision happens and calculate collision forces.
7. Update the particle position and velocity by the collision forces.
8. Return to the first step ($n \to n+1$) and advance to the next time step.

Efficient Data Structures

$L3 \approx 220.000\, elements \qquad \approx 1.100.000\, d.o.f.s$
$L4 \approx 880.000\, elements \qquad \approx 4.400.000\, d.o.f.s$
$L5 \approx 3.530.000\, elements \qquad \approx 17.600.000\, d.o.f.s$

DEC/COMPAQ EV6, 833 MHz

CPU (s)	'brute force'					
#PART	$= 10$			$= 1000$		
items	L=3	L=4	L=5	L=3	L=4	L=5
NSE	17	88	440	16	80	**403**
Force	5	20	79	443	1771	**7092**
Particle	1	5	25	20	82	**331**
Total	24	114	546	480	1934	**7827**

Efficient Data Structures

$L3 \approx 220.000\, elements \qquad \approx 1.100.000\, d.o.f.s$
$L4 \approx 880.000\, elements \qquad \approx 4.400.000\, d.o.f.s$
$L5 \approx 3.530.000\, elements \qquad \approx 17.600.000\, d.o.f.s$

DEC/COMPAQ EV6, 833 MHz

CPU (s)	'brute force'						'improved'					
#PART	$= 10$			$= 1000$			$= 10$			$= 1000$		
items	L=3	L=4	L=5	L=3	L=4	L=5	L=3	L=4	L=5	L=3	L=4	L=5
NSE	17	88	440	16	80	**403**	17	95	423	17	83	**435**
Force	5	20	79	443	1771	**7092**	0	0	1	0	0	**1**
Particle	1	5	25	20	82	**331**	0	3	14	1	5	**21**
Total	24	114	546	480	1934	**7827**	18	98	439	18	89	**468**

Next: Efficient flow solver (for small Δt) ???

Multilevel Pressure Schur Complement

'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) decoupling of velocity and pressure'
'Newton/Multigrid solver for all scalar subproblems'

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

Key Ideas of GLOBAL MPSC Approaches:

$$\begin{bmatrix} S & kB \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \quad \Leftrightarrow \quad \begin{aligned} B^T S^{-1} B p &= \frac{1}{k} B^T S^{-1} \mathbf{g} \\ \mathbf{u} &= S^{-1}(\mathbf{g} - kBp) \end{aligned}$$

$\Rightarrow$ **Preconditioned Richardson scheme** for scalar problems:

$$\begin{aligned} p^l &= p^{l-1} - C^{-1}(Ap^{l-1} - b) \\ &= p^{l-1} - C^{-1}(B^T S^{-1} B p^{l-1} - \frac{1}{k} B^T S^{-1} \mathbf{g}) \end{aligned}$$

$\Rightarrow$ **Choice of (global) PSC preconditioner C^{-1}:**

$$C^{-1} = \sum \alpha_i \tilde{C}_i^{-1} \quad (\approx [B^T S^{-1} B]^{-1}) \quad , \quad S := \alpha M + \theta_1 \nu k L + \theta_2 k N(\tilde{\mathbf{U}})$$

'GOOD' SMOOTHER FOR SCALAR PROBLEM $A := B^T S^{-1} B$?

‘Additive’ GLOBAL PSC Preconditioners:

$$C^{-1} = \alpha_R P^{-1} + \alpha_D M_p^{-1}$$

Complete GLOBAL PSC Basic Iteration:

$$p^l = p^{l-1} - [\alpha_R P^{-1} + \alpha_D M_p^{-1}]\,(B^T S^{-1} B p^{l-1} - \frac{1}{k} B^T S^{-1} \mathbf{g})$$

Realization of 1 ”Discrete Projection” step:

$S\tilde{\mathbf{u}} = \mathbf{g} - kBp^n$ (**Burgers** problem with p^n and $\mathbf{u}^n$ given)

$f_p := \frac{1}{k} B^T \tilde{\mathbf{u}}$ $[= \frac{1}{k} B^T S^{-1} \mathbf{g} - B^T S^{-1} B p^n =$ **Residual** $(p^n)]$

$Pq = f_p$ ($\sim$ **‘Pressure-Poisson’**)

$$\Rightarrow \;\; p^{n+1} = p^n + \alpha_R q + \alpha_D M_p^{-1} f_p$$

Key Ideas of GLOBAL MPSC (cont.):

REACTIVE PRECONDITIONER $(\nabla \cdot I \nabla)^{-1}$:

- explicit calculation of $P := B^T M^{-1} B \Rightarrow \tilde{Q}_1/Q_0$: **5 (2D)/7 (3D)**
- P 'exact' discrete PSC preconditioner for $\Delta t \to 0$

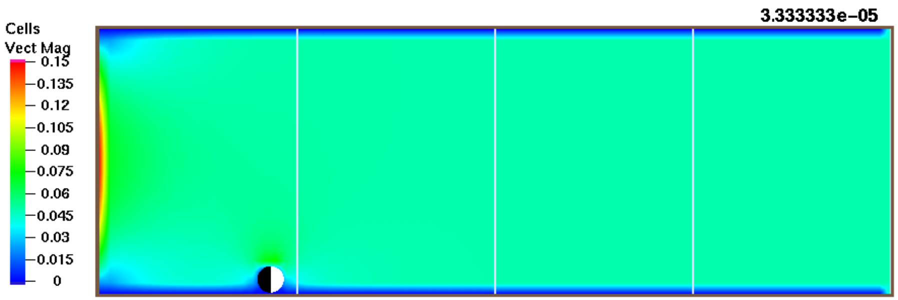
DIFFUSIVE PRECONDITIONER $(\nabla \cdot \Delta^{-1} \nabla)^{-1}$:

- explicit calculation of $B^T L^{-1} B$ impossible (L^{-1} **full !!!**)
- However: $\nabla \cdot \Delta^{-1} \nabla \sim I \implies B^T L^{-1} B \sim M_p$

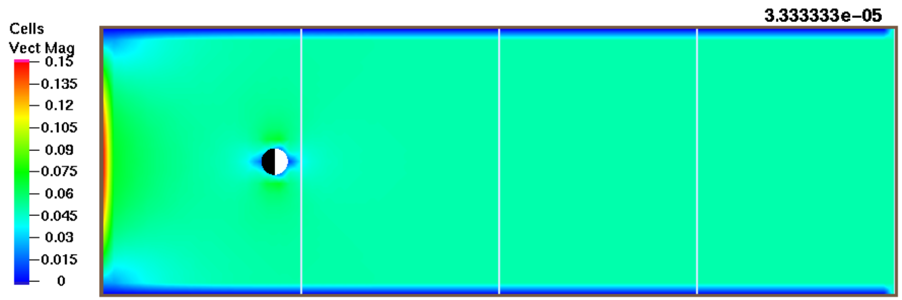
CONVECTIVE PRECONDITIONER $(\nabla \cdot (\mathbf{u} \cdot \nabla)^{-1} \nabla)^{-1}$:

- discrete construction $\longrightarrow$ **ILU ???**
- continuous construction $\to \nabla \cdot (\beta \cdot \nabla)^{-1} \nabla \sim$ **???**
- new techniques...

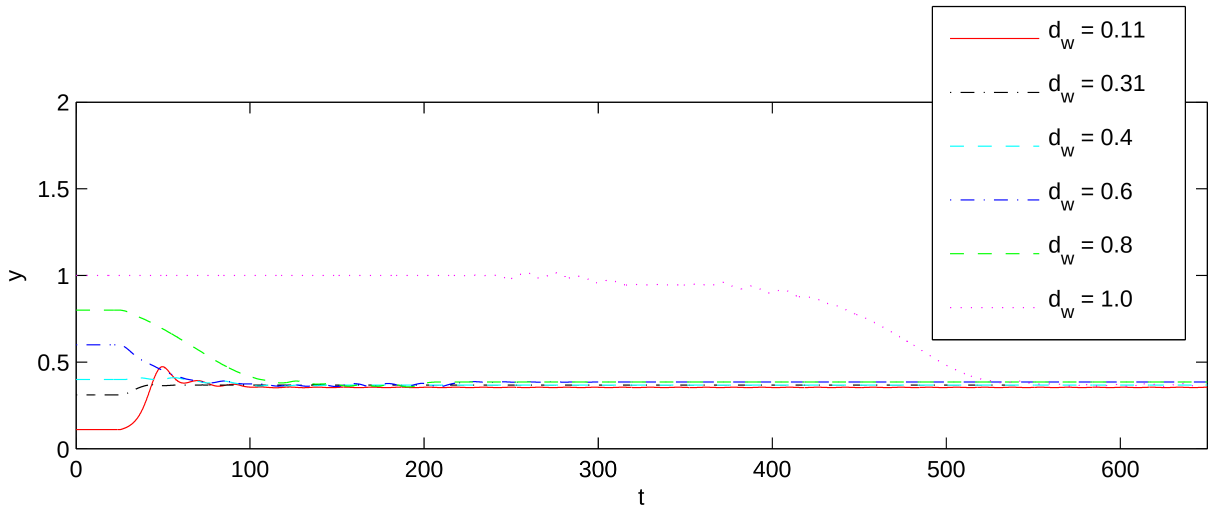
Lift-Off for Circle



Velocity $(d_w = 0.1)$

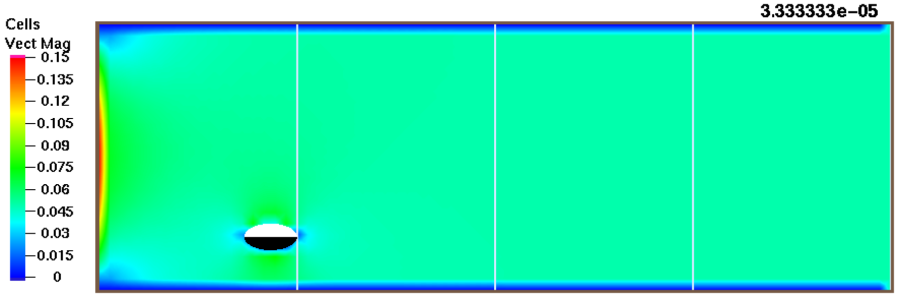


Velocity $(d_w = 1.0)$

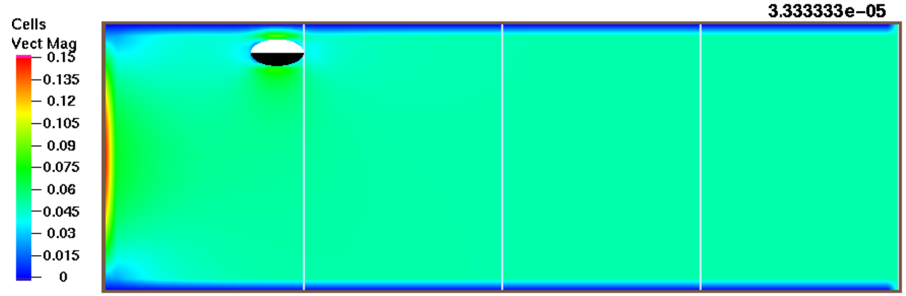


y of center of ball

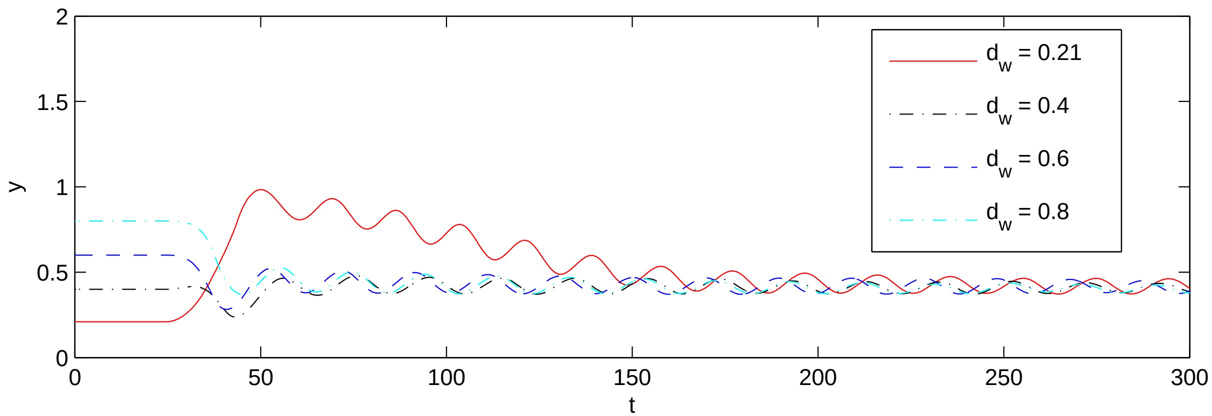
Lift-Off for Ellipse



Velocity $(d_w = 0.4)$

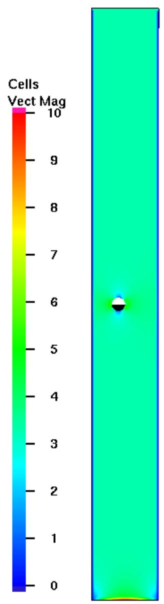


Velocity $(d_w = 1.8)$

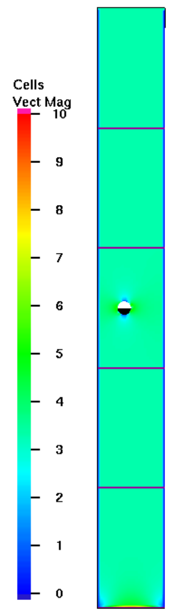


y of center of ellipse

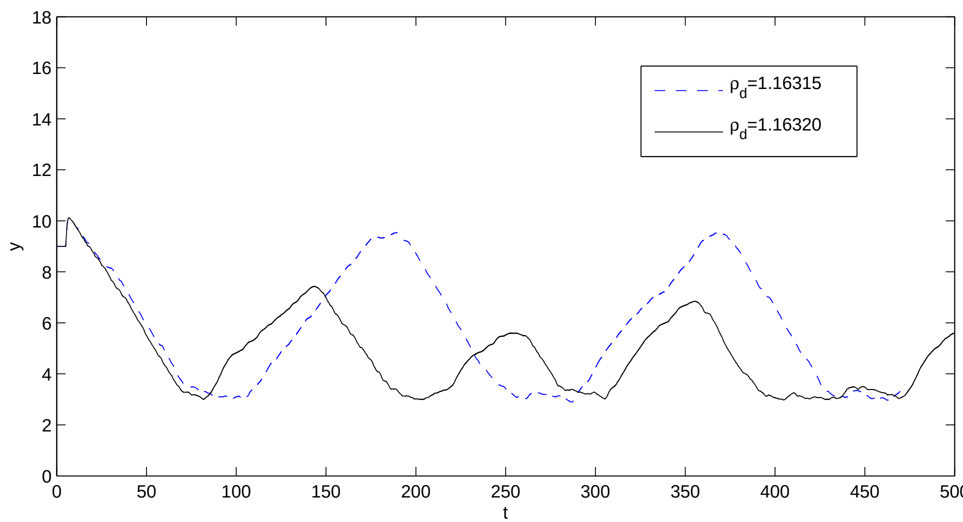
Current Aim (Dan Joseph)



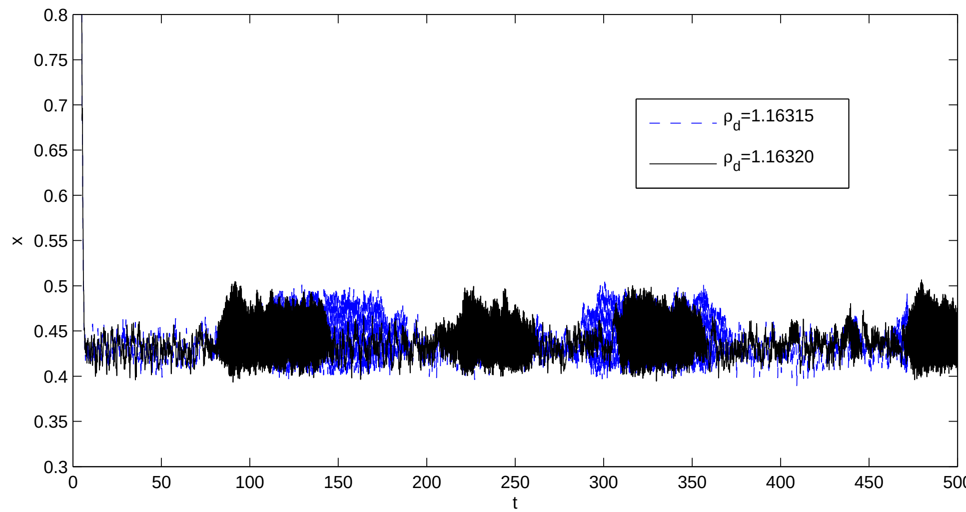
$(\rho_s = 1.16315)$



$(\rho_s = 1.16315)$ moving frame



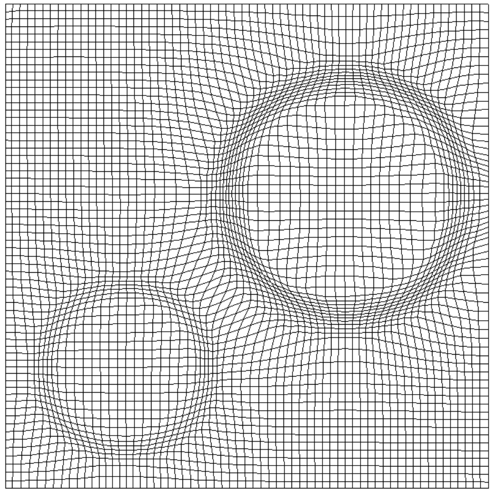
(a) y of center of ball

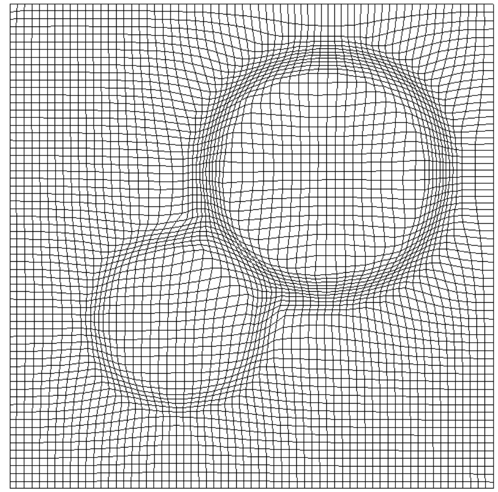


(b) x of center of ball

Challenges

- Adaptive time-stepping + dynamical adaptive grid alignment

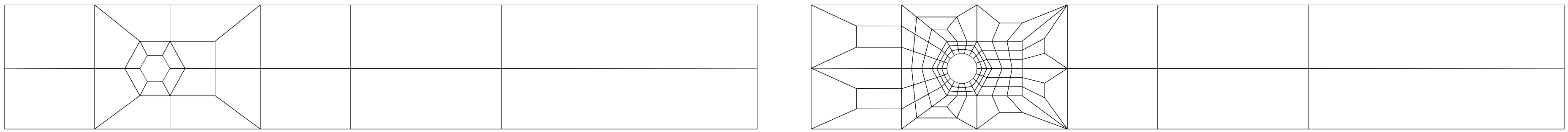




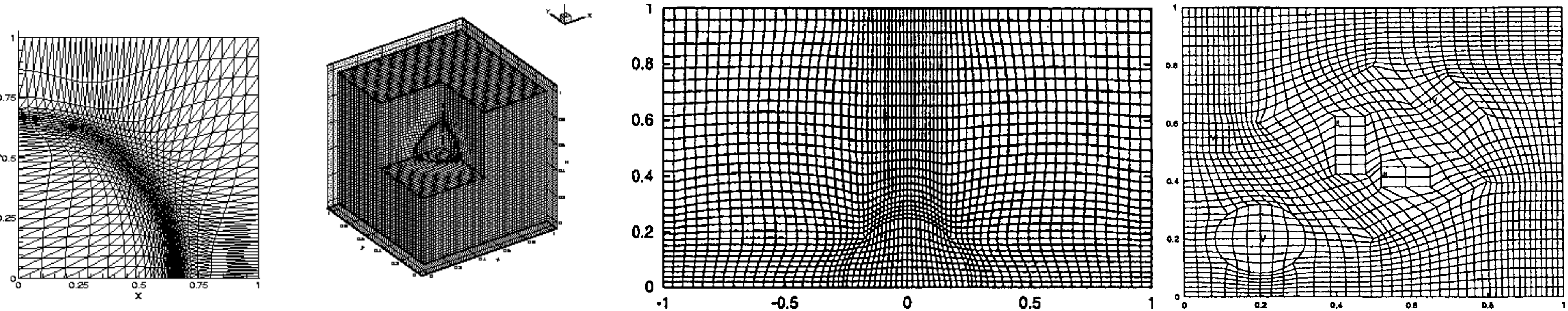
- (Better) Collision models/Repulsive forces
- Coupling with turbulence models
- Modelling of Break-up/Coalescence phenomena
- Deformable particles/fluid-structure interaction
- Analysis of viscoelastic effects
- Benchmarking and experimental validation for ”many” particles
- ”1.000.000” particles...

Concepts for Adaptive Meshing

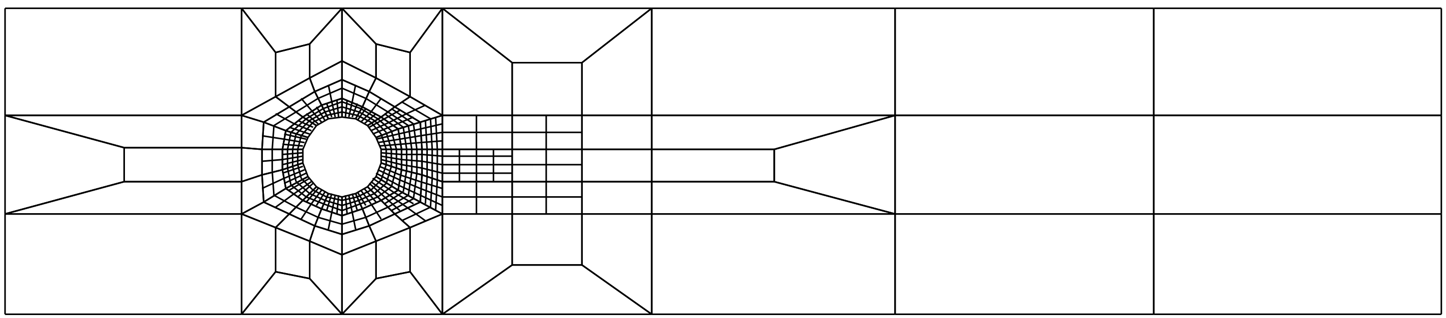
1) macro-oriented adaptivity



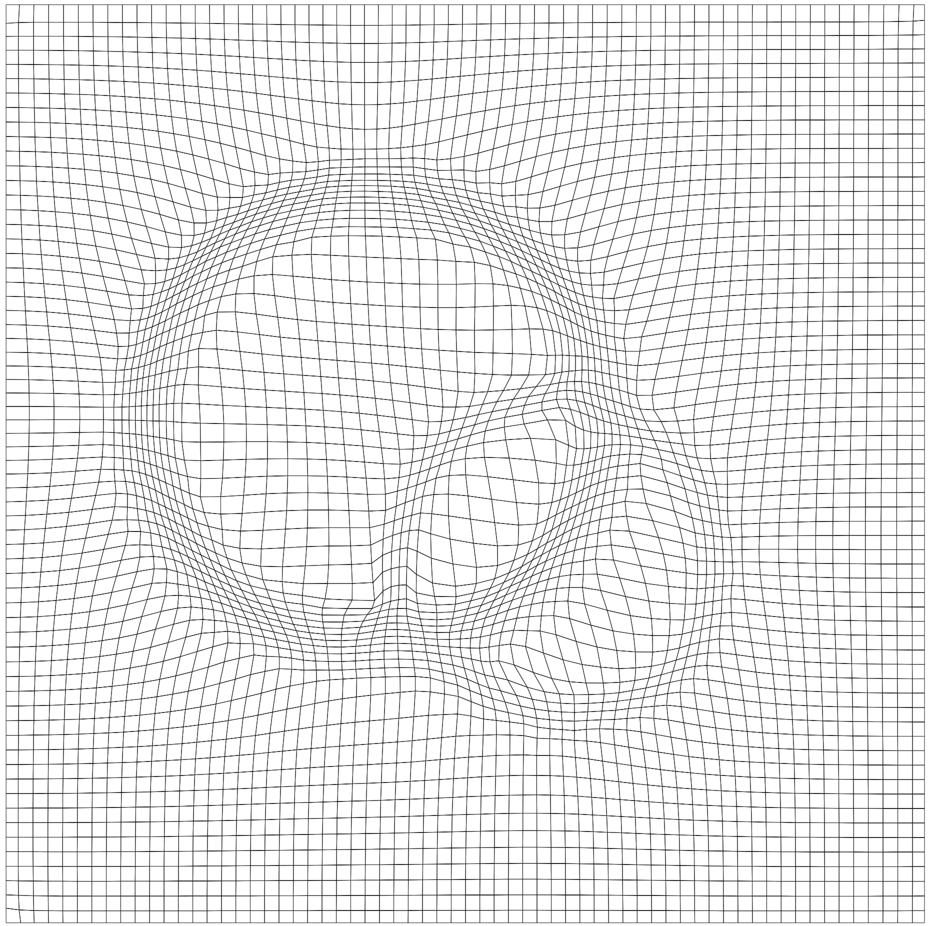
2) (patchwise) ‘deformation’ adaptivity



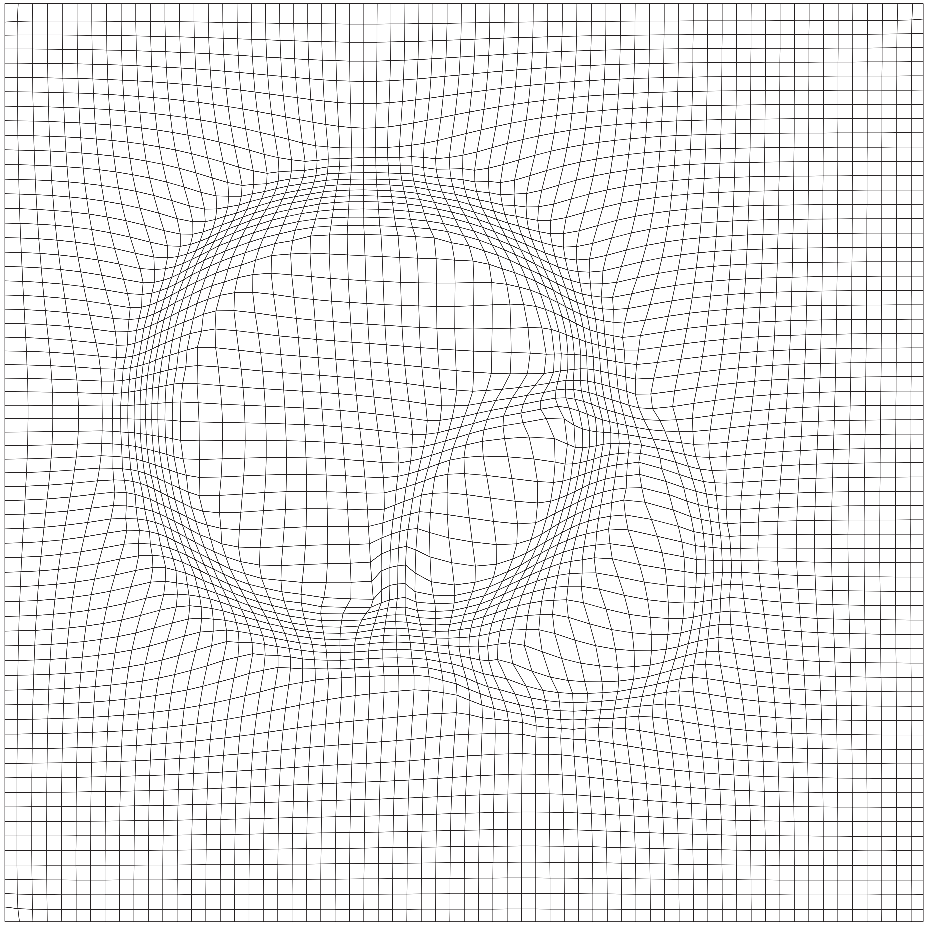
3) (patchwise) ‘local’ adaptivity



Example for Deformed Mesh



Example for Deformed Mesh



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

R-Adaptivity

1. **location based methods:**
   - Winslow’s method
   - Brackbill’s and Saltzman’s method
   - harmonic mapping

   disadvantages:

   (a) non-linear problems (demanding)

   (b) interaction of monitor function and grid not clear

2. **velocity based methods:**
   - MMPDE/GCL (Cao, Huang, Russell)
   - Deformation method (Liao et al.)

   advantages:

   (a) (several) Laplace problems on fixed mesh (fast)

   (b) monitor function “directly” from error distribution

   (c) mesh tangling prevented

Deformation Method (Moser/Liao)

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

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

1. compute monitor function $f(x,t) > 0, f \in \mathcal{C}^1$ and

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

Deformation Method (Moser/Liao)

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

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

1. compute monitor function $f(x,t) > 0, f \in \mathcal{C}^1$ and
$\int_\Omega f^{-1}(x,t)\,dx = |\Omega| \quad \forall t \in [0,1]$

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

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

Deformation Method (Moser/Liao)

idea : construct transformation $\phi, x = \phi(\xi, t)$ with $\det \nabla\phi = f$
$\Rightarrow$ local mesh area $\approx f$

1. compute monitor function $f(x,t) > 0, f \in \mathcal{C}^1$ and
$\int_\Omega f^{-1}(x,t)\, dx = |\Omega| \quad \forall t \in [0,1]$

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

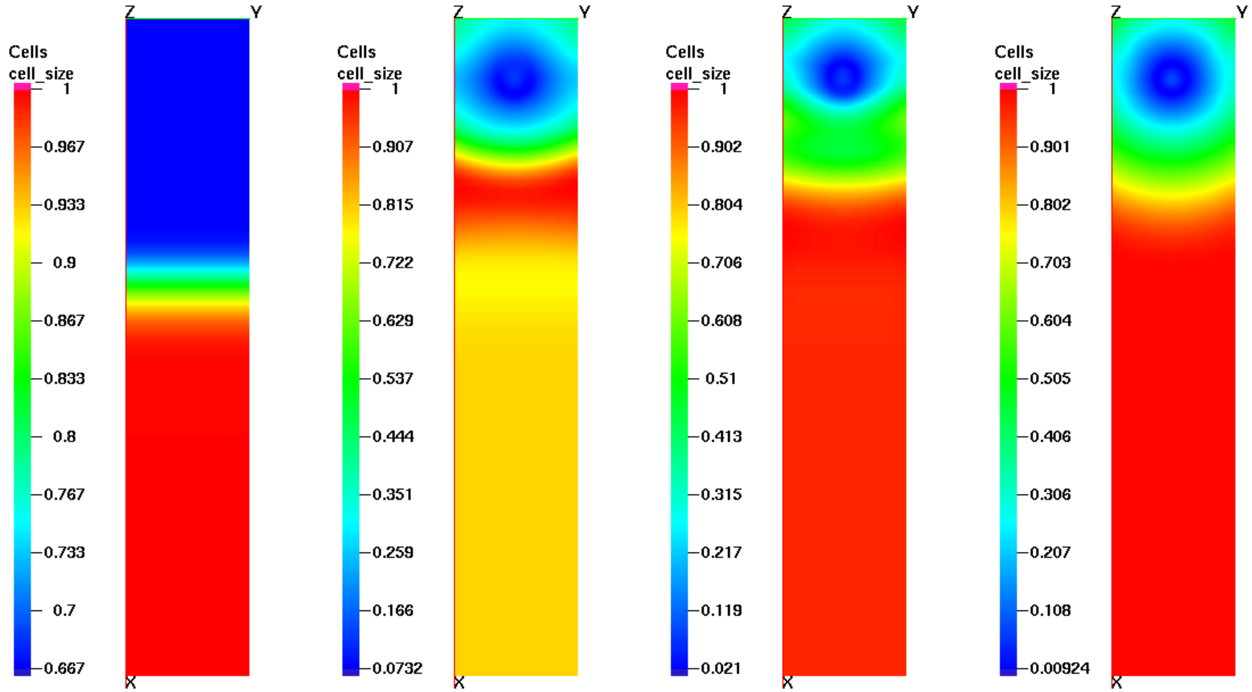
$$\Delta v(\xi,t) = -\frac{\partial}{\partial t}\left(\frac{1}{f(\xi,t)}\right), \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0$$

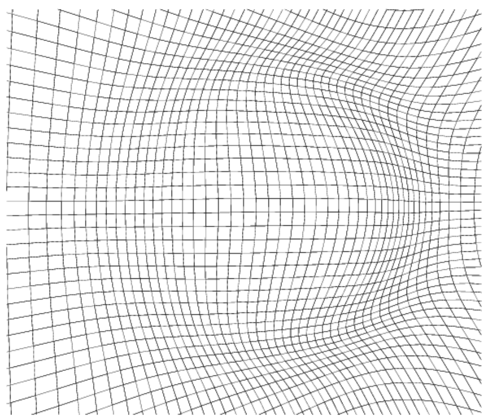
3. solve the ODE system

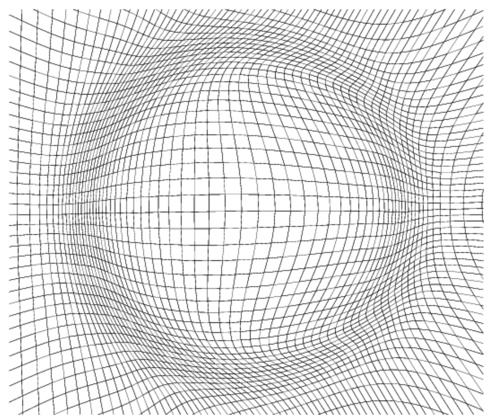
$$\frac{\partial}{\partial t}\phi(\xi,t) = f\Big(\phi(\xi,t),t\Big)\,\nabla v\Big(\phi(\xi,t),t\Big)$$

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

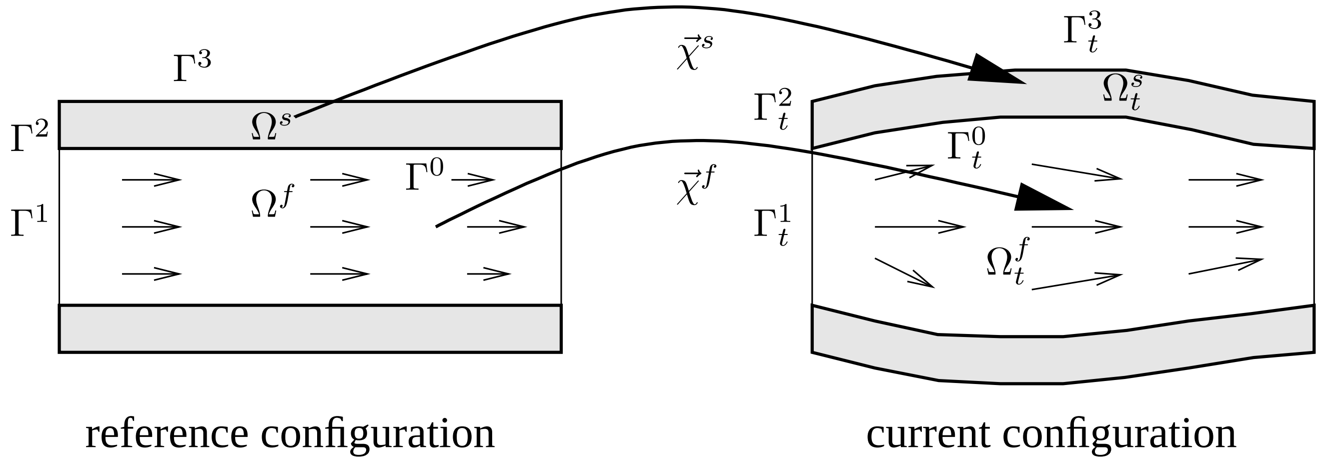
Dynamic Mesh Deformation







Fluid - (Elastic) Solid Interfaces



structure part

$$\boldsymbol{\chi}^s : \Omega^s \times [0,T] \mapsto \Omega_t^s$$

$$\boldsymbol{u}^s = \boldsymbol{\chi}^s(\boldsymbol{X},t) - \boldsymbol{X}, \quad \boldsymbol{v}^s = \frac{\partial \boldsymbol{u}^s}{\partial t}$$

$$\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}^s, \quad J = \det \boldsymbol{F}$$

fluid part

$$\boldsymbol{\chi}^f : \Omega^f \times [0,T] \mapsto \Omega_t^f$$

$$\boldsymbol{u}^f = \boldsymbol{\chi}^f(\boldsymbol{X},t) - \boldsymbol{X}$$

$$\boldsymbol{v}^f : \Omega_t^f \times [0,T] \mapsto \mathbb{R}^n$$

Governing Equations

structure part

$$\frac{\partial \boldsymbol{v}^s}{\partial t} = \operatorname{div}(J\boldsymbol{\sigma}^s \boldsymbol{F}^{-T}) + \boldsymbol{f} \qquad \text{in } \Omega^s$$

$$\det(\boldsymbol{I} + \nabla \boldsymbol{u}^s) = 1 \qquad \text{in } \Omega^s$$

$$\boldsymbol{u}^s = \boldsymbol{0} \qquad \text{on } \Gamma^2$$

$$\boldsymbol{\sigma}^s \boldsymbol{n} = \boldsymbol{0} \qquad \text{on } \Gamma^3$$

fluid part

$$\frac{\partial \boldsymbol{v}^f}{\partial t} + (\nabla \boldsymbol{v}^f)\boldsymbol{v}^f = \operatorname{div} \boldsymbol{\sigma}^f + \boldsymbol{f} \qquad \text{in } \Omega_t^f$$

$$\operatorname{div} \boldsymbol{v}^f = 0 \qquad \text{in } \Omega_t^f$$

$$\boldsymbol{v}^f = \boldsymbol{v}_0 \qquad \text{on } \Gamma_t^1$$

$$\boldsymbol{\sigma}^f \boldsymbol{n} = \boldsymbol{t} \qquad \text{on } \Gamma_t^1$$

interface conditions

$$\boldsymbol{v}^f = \boldsymbol{v}^s \qquad \text{on } \Gamma_0$$

$$\boldsymbol{\sigma}^f \boldsymbol{n} = \boldsymbol{\sigma}^s \boldsymbol{n} \qquad \text{on } \Gamma_0$$

Fully Coupled Formulation

$$\frac{\partial \boldsymbol{u}}{\partial t} = \begin{cases} \boldsymbol{v} & \text{in } \Omega^s \\ \Delta \boldsymbol{u} \text{ “mesh deformation operator”} & \text{in } \Omega^f \end{cases} \tag{1}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} = \begin{cases} -(\nabla \boldsymbol{v})\boldsymbol{F}^{-1}(\boldsymbol{v} - \frac{\partial \boldsymbol{u}}{\partial t}) + \frac{1}{J}\nabla\cdot(-Jp^f\boldsymbol{F}^{-T} + J\boldsymbol{\sigma}^f\boldsymbol{F}^{-T}) & \text{in } \Omega^f \\ \frac{1}{J\beta}\nabla\cdot(-Jp^s\boldsymbol{F}^{-T} + J\boldsymbol{\sigma}^s\boldsymbol{F}^{-T}) & \text{in } \Omega^s \end{cases} \tag{2}$$

$$0 = \begin{cases} \nabla\cdot(J\boldsymbol{v}\boldsymbol{F}^{-T}) & \text{in } \Omega^f \\ J-1 & \text{in } \Omega^s \end{cases} \tag{3}$$

$$\Downarrow$$

Sequence of discrete saddle point problems

$$\begin{pmatrix} M - \frac{k}{2}L^f & \frac{k}{2}M^s & 0 \\ \frac{1}{2}\frac{\partial N_2}{\partial u_h} + \frac{k}{2}\frac{\partial(N_1+S^s+S^f)}{\partial u_h} + k\frac{\partial B}{\partial u_h}p_h & M^s + \beta M^f + \frac{1}{2}\frac{\partial N_2}{\partial v_h} + \frac{k}{2}\frac{\partial(N_1+S_f^2)}{\partial v_h} & kB \\ B^{sT} + \frac{\partial B^{fT}}{\partial u_h}v_h & B^{fT} & 0 \end{pmatrix}$$

Fully Coupled Formulation

$$\frac{\partial \boldsymbol{u}}{\partial t} = \begin{cases} \boldsymbol{v} & \text{in } \Omega^s \\ \Delta \boldsymbol{u} \text{ "mesh deformation operator"} & \text{in } \Omega^f \end{cases} \tag{3}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} = \begin{cases} -(\nabla \boldsymbol{v})\boldsymbol{F}^{-1}(\boldsymbol{v} - \frac{\partial \boldsymbol{u}}{\partial t}) + \frac{1}{J}\nabla \cdot (-Jp^f \boldsymbol{F}^{-T} + J\boldsymbol{\sigma}^f \boldsymbol{F}^{-T}) & \text{in } \Omega^f \\ \frac{1}{J\beta}\nabla \cdot (-Jp^s \boldsymbol{F}^{-T} + J\boldsymbol{\sigma}^s \boldsymbol{F}^{-T}) & \text{in } \Omega^s \end{cases} \tag{3}$$

$$0 = \begin{cases} \nabla \cdot (J\boldsymbol{v}\boldsymbol{F}^{-T}) & \text{in } \Omega^f \\ J - 1 & \text{in } \Omega^s \end{cases} \tag{3}$$

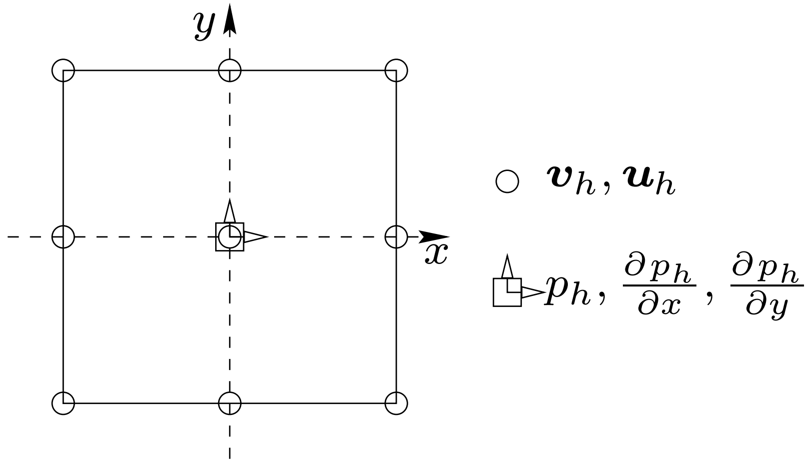
$$\Downarrow$$

Sequence of discrete saddle point problems

$$\begin{bmatrix} S_{\mathbf{uu}} & S_{\mathbf{uv}} & 0 \\ S_{\mathbf{vu}} & S_{\mathbf{vv}} & kB \\ c_{\mathbf{u}} B_s^T & c_{\mathbf{v}} B_f^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f_u} \\ \mathbf{f_v} \\ f_p \end{bmatrix}$$

Discretization in Space and Time

Discretization in space: FEM $Q_2/Q_2/P_1$



$$U_h = \{\boldsymbol{u}_h \in [C(\Omega_h)]^2, \boldsymbol{u}_h|_T \in [Q_2(T)]^2 \forall T \in \mathcal{T}_h, \boldsymbol{u}_h = \mathbf{0} \text{ on } \Gamma_1\},$$

$$V_h = \{\boldsymbol{v}_h \in [C(\Omega_h)]^2, \boldsymbol{v}_h|_T \in [Q_2(T)]^2 \forall T \in \mathcal{T}_h, \boldsymbol{v}_h = \mathbf{0} \text{ on } \Gamma_2\},$$

$$P_h = \{p_h \in L^2(\Omega_h), p_h|_T \in P_1(T) \forall T \in \mathcal{T}_h\}.$$

Discretization in time: Crank-Nicolson scheme with adaptive time-step selection

Solution of the Nonlinear Problem

- compute the Jacobian matrix via divided differences

$$\left[\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{X}}\right]_{ij} (\boldsymbol{X}^n) \approx \frac{[\boldsymbol{\mathcal{R}}]_i(\boldsymbol{X}^n + \varepsilon \boldsymbol{e}_j) - [\boldsymbol{\mathcal{R}}]_i(\boldsymbol{X}^n - \varepsilon \boldsymbol{e}_j)}{2\varepsilon},$$

- solve for $\dot{\boldsymbol{X}}$

$$\left[\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{X}}(\boldsymbol{X}^n)\right] \dot{\boldsymbol{X}} = \boldsymbol{\mathcal{R}}(\boldsymbol{X}^n)$$

- adaptive line search strategy

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^n + \omega \dot{\boldsymbol{X}} \qquad \omega \text{ such that } f(\omega) = \boldsymbol{\mathcal{R}}(\boldsymbol{X} + \omega \dot{\boldsymbol{X}}) \cdot \boldsymbol{X} \searrow$$

- BiCGStab/GMRes(m) with ILU(k) preconditioner or multigrid for the linear problems

Multigrid Solver

- standard geometric multigrid approach
- smoother by local MPSC-Ansatz (Vanka-like smoother)

$$\begin{bmatrix} \mathbf{u}^{l+1} \\ \mathbf{v}^{l+1} \\ p^{l+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{l} \\ \mathbf{v}^{l} \\ p^{l} \end{bmatrix} - \omega \sum_{\text{Patch } \Omega_i} \begin{bmatrix} S_{\mathbf{uu}|\Omega_i} & S_{\mathbf{uv}|\Omega_i} & 0 \\ S_{\mathbf{vu}|\Omega_i} & S_{\mathbf{vv}|\Omega_i} & kB_{|\Omega_i} \\ c_{\mathbf{u}} B^T_{s|\Omega_i} & c_{\mathbf{v}} B^T_{f|\Omega_i} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{def}^l_{\mathbf{u}} \\ \mathbf{def}^l_{\mathbf{v}} \\ def^l_p \end{bmatrix}$$

- full inverse of the local problems by optimized LAPACK (39×39 systems)
- alternatives: simplified local problems (3×3 systems) or ILU(k)
- combination with GMRES/BiCGStab methods possible
- full Q_2 and P_1^{disc} prolongation $\boldsymbol{P}$, restriction by $\boldsymbol{R} = \boldsymbol{P}^T$
- fast, robust and efficient Q_2 multigrid solvers available

Current Status

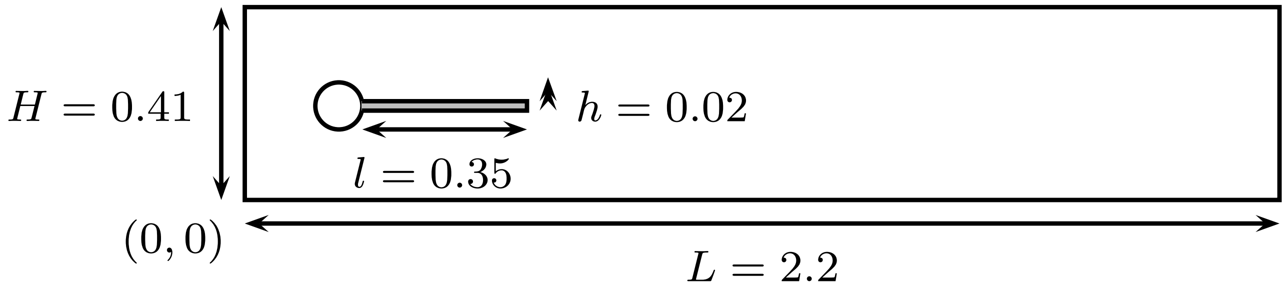
- Discretization
  - ✓ monolithic, fully coupled FEM (Q_2/P_1) for **viscous incompressible fluid** and **incompressible** or **compressible hyperelastic structure**
  - ✓ fully implicit 2nd order discretization in time (Crank-Nicholson, Fractional θ-step)
- Solver
  - ✓ Newton-like method for the coupled system (Jacobian matrix via divided differences)
  - ✓ **multigrid method** with Vanka-like smoother, preconditioned Krylov space (ILU(k)/GMRES(m)) or combination of both
- Further improvements
  - ❏ adaptive time step control
  - ❏ dynamically space-adapted mesh aligned with the structure ($\longrightarrow$ *deformation method*)
  - ❏ smoothers robust with respect to anisotropy
  - ❏ global MPSC/Discrete Projection solvers
  - ❏ accurate + robust + non-oscillatory stabilization for convection

Fluid-Structure Interaction Benchmark

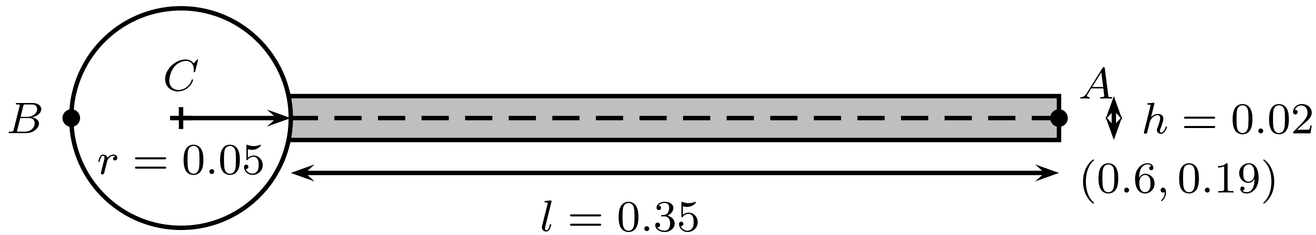
- based on the successful DFG *flow around cylinder*
- realistic materials
  - **incompressible Newtonian fluid**, laminar flow regime
  - **elastic solid**, large deformations (compressible + incompressible)
- setup with simple periodic oscillations + reasonable deformations
- computable configuration $\Rightarrow$ laminar flow, reasonable aspect ratios
- validation by experiments (Erlangen)

Computational Domain

- domain dimensions

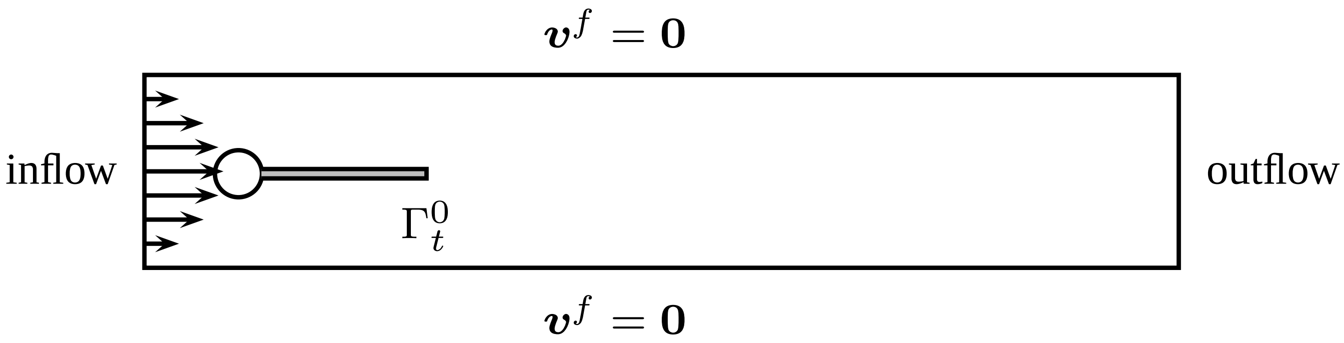


- detail of the submerged structure



- $A(t = 0) = (0.6, 0.2), \quad B = (0.15, 0.2), \quad C = (0.2, 0.2)$

Boundary and Initial Conditions



inflow parabolic velocity profile is prescribed at the left end of the channel

$$v^f(0,y) = 1.5\frac{y(H-y)}{\left(\frac{H}{2}\right)^2} = 1.5\frac{4.0}{0.1681}y(0.41-y),$$

outflow condition can be chosen by the user, assuming zero reference pressure (*stress free* or *do nothing*)

interface condition on Γ_t^0 is $\boldsymbol{v}^f = \boldsymbol{v}^s$ and $\boldsymbol{\sigma}^f\boldsymbol{n} = \boldsymbol{\sigma}^s\boldsymbol{n}$

otherwise the *no-slip* condition is prescribed for the fluid on the other boundary parts. i.e. top and bottom wall and cylinder

initial condition (suggestion)

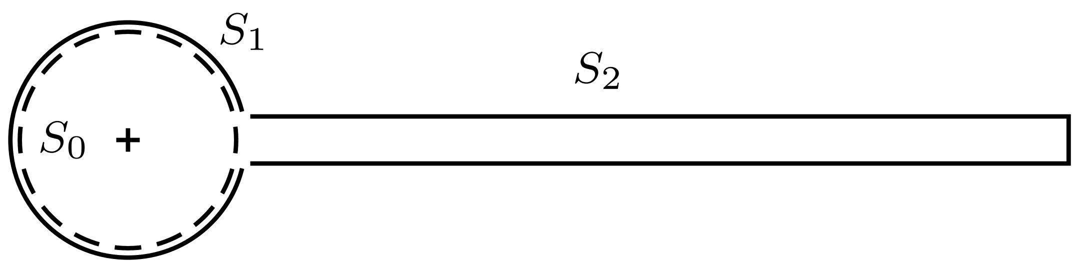
- zero velocity in the fluid and no deformation of the structure + smooth increase of the inflow profile

Quantities of Interest

- the position $A(t) = (x(t), y(t))$ of the end of the structure
- pressure difference between the points $A(t)$ and B

$$\Delta p^{AB}(t) = p^{B}(t) - p^{A(t)}(t)$$

- forces exerted by the fluid on the *whole* body, i.e. lift and drag forces acting on the cylinder and the structure together



$$(F_D, F_L) = \int_S \boldsymbol{\sigma} \boldsymbol{n} dS = \int_{S_1} \boldsymbol{\sigma}^f \boldsymbol{n} dS + \int_{S_2} \boldsymbol{\sigma}^{f|s} \boldsymbol{n} dSx = \int_{S_0} \boldsymbol{\sigma} \boldsymbol{n} dS$$

- frequency and maximal amplitude
- compare results for *one* full period and 3 different levels of spatial discretization h and 3 time step sizes Δt

Suggested Material Parameters

solid		fluid	
ϱ^s	density	ϱ^f	density
ν^s	Poisson ratio	ν^f	kinematic viscosity
μ^s	shear modulus		

parameter	polybutadiene & glycerine	polypropylene & glycerine
ϱ^s $[10^3 \frac{\mathrm{kg}}{\mathrm{m}^3}]$	0.91	1.1
ν^s	0.5	0.42
μ^s $[10^6 \frac{\mathrm{kg}}{\mathrm{ms}^2}]$	0.53	317
ϱ^f $[10^3 \frac{\mathrm{kg}}{\mathrm{m}^3}]$	1.26	1.26
ν^f $[10^{-3} \frac{\mathrm{m}^2}{\mathrm{s}}]$	1.13	1.13

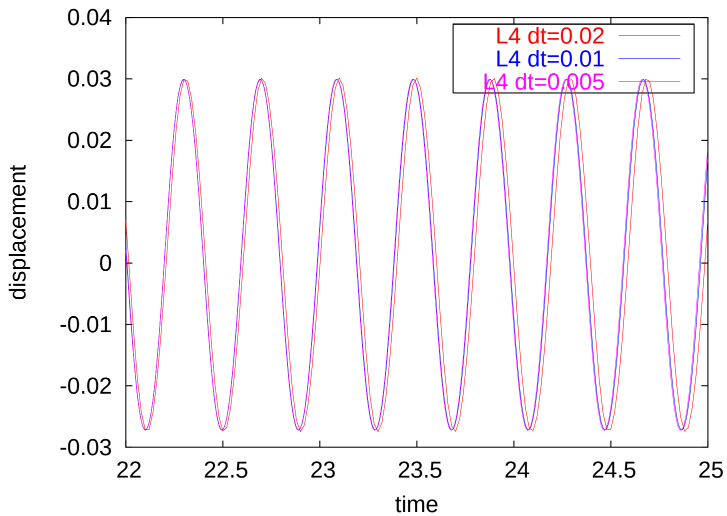
Some material combinations

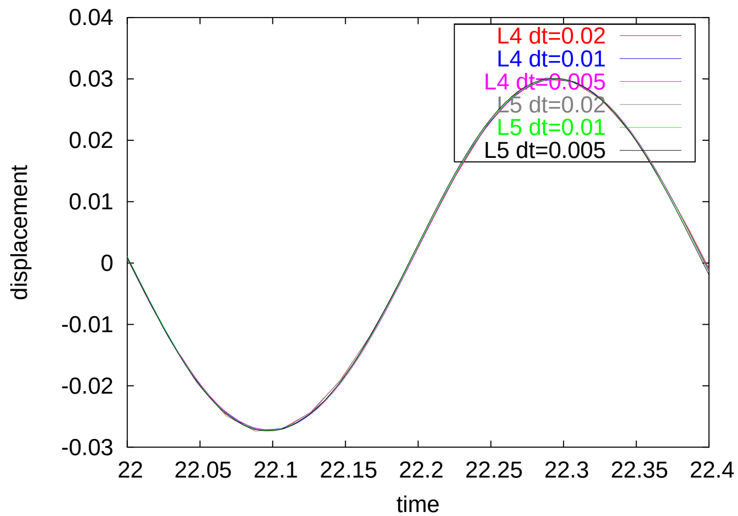
parameter	test 1	test 2	test 2a
ϱ^s $[10^3 \frac{\mathrm{kg}}{\mathrm{m}^3}]$	1	0.8	0.8
ν^s	0.5	0.4	0.4
μ^s $[10^6 \frac{\mathrm{kg}}{\mathrm{ms}^2}]$	0.5	100	0.5
ϱ^f $[10^3 \frac{\mathrm{kg}}{\mathrm{m}^3}]$	1	1	1
ν^f $[10^{-3} \frac{\mathrm{m}^2}{\mathrm{s}}]$	1	1	1

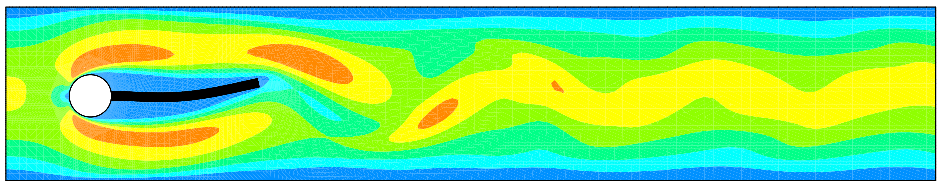
Suggested parameter settings for the tests

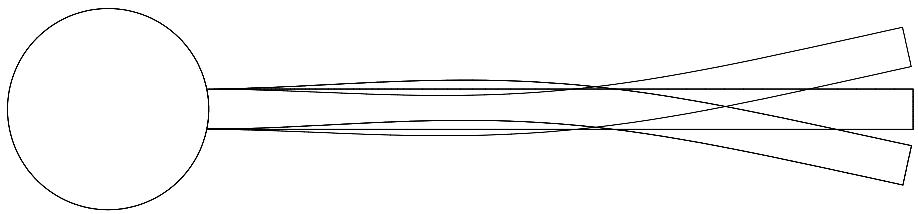
Test 1: Incompressible

$\varrho^s = 1 \times 10^3 \qquad \nu^s = 0.5 \qquad \mu^s = 0.5 \times 10^6 \qquad \varrho^f = 1 \times 10^3 \qquad \nu^f = 1 \times 10^3$



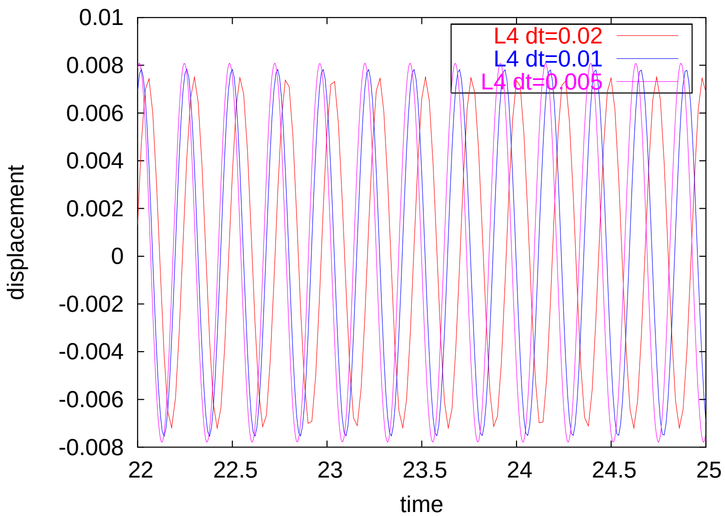


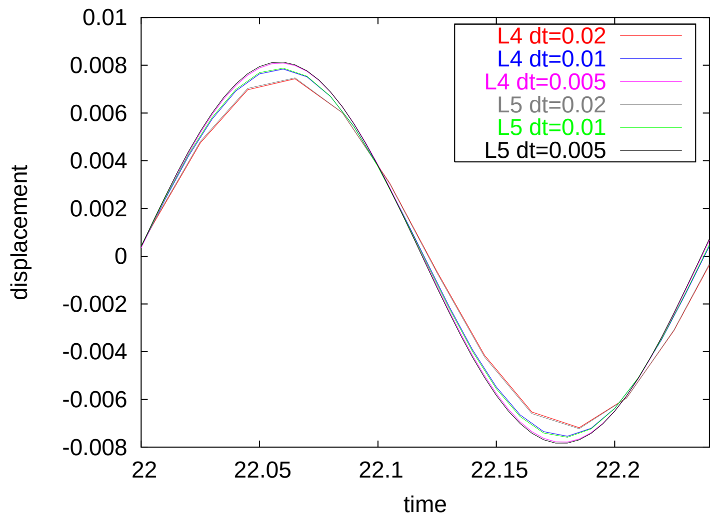


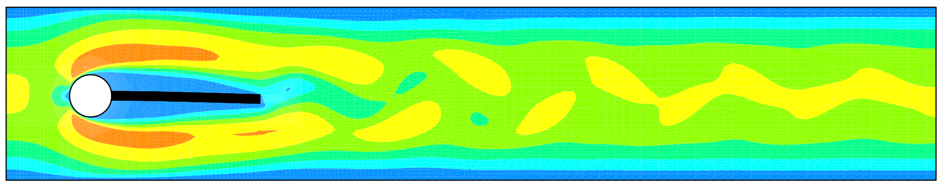


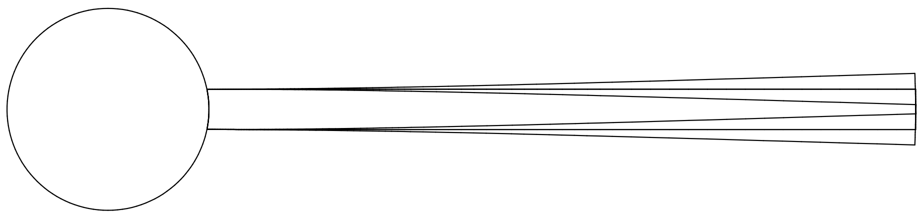
Test 2: Compressible (realistic)

$$\varrho^s = 0.8 \times 10^3 \qquad \nu^s = 0.4 \qquad \mu^s = 100.0 \times 10^6 \qquad \varrho^f = 1 \times 10^3 \qquad \nu^f = 1 \times 10^3$$



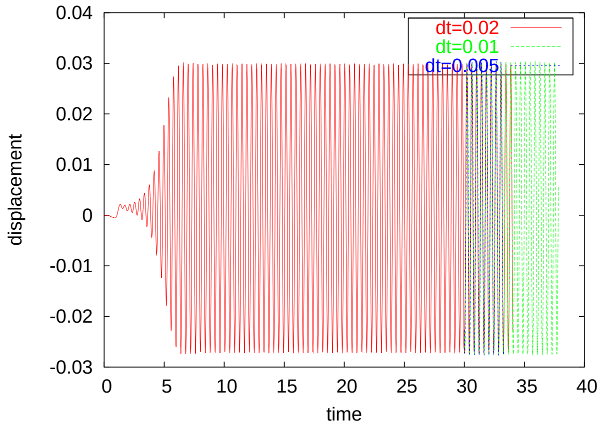


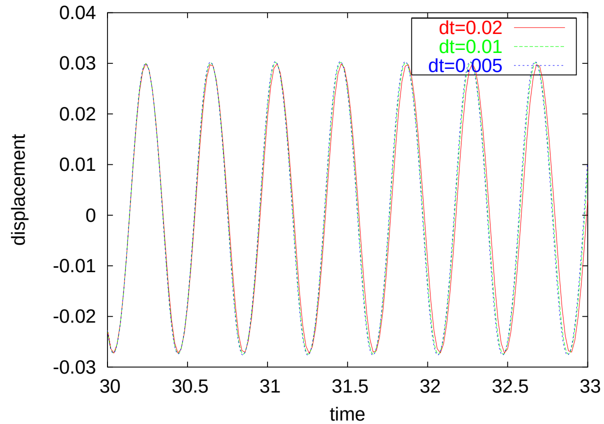


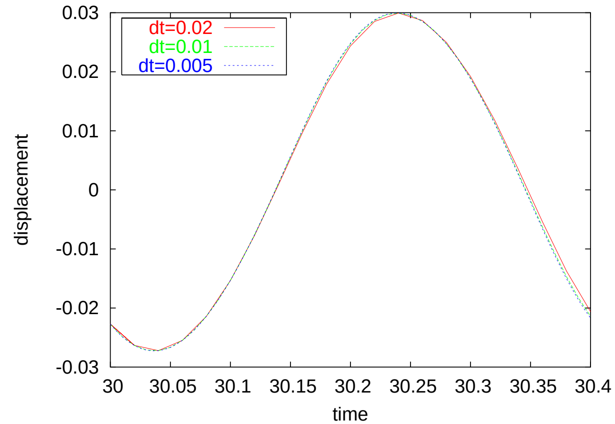


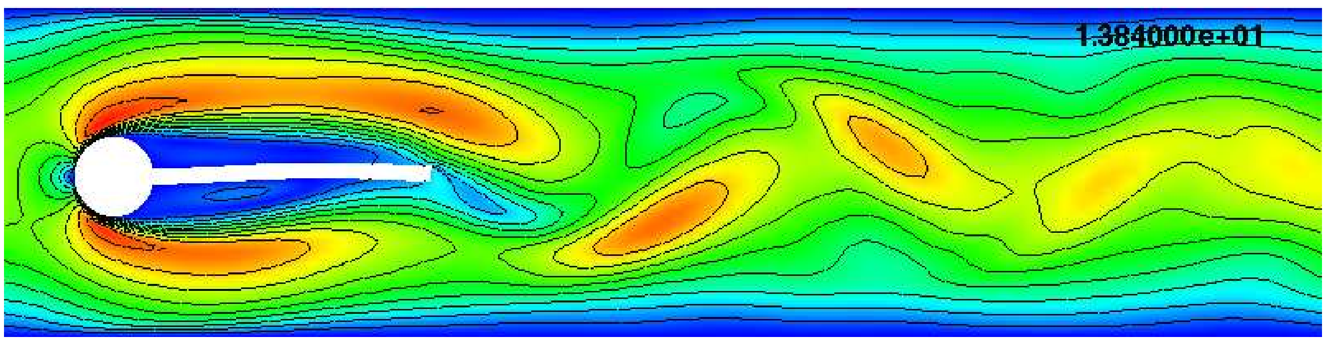
Test 2a: Compressible (numerical)

$$\varrho^s = 0.8 \times 10^3 \qquad \nu^s = 0.4 \qquad \mu^s = 0.5 \times 10^6 \qquad \varrho^f = 1 \times 10^3 \qquad \nu^f = 1 \times 10^3$$









Challenges and Problems

- Coupled (‘monolithic’) + fully implicit FEM discretization $\longrightarrow$ YES
- Efficient coupling/decoupling strategies for solver $\longrightarrow$ YES (?)
- Adaptive meshing/Error control of user-specific quantities $\longrightarrow$ YES (???)

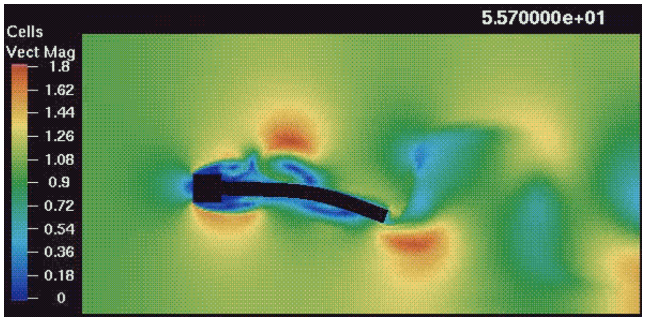
$\rightarrow$ **Prototypical for (multiphase) flow problems ?**

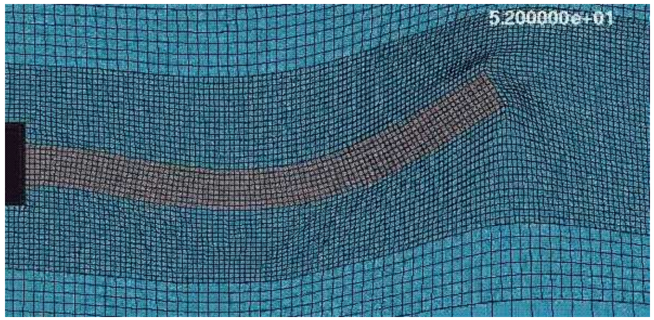
Challenges and Problems

- Coupled (‘monolithic’) + fully implicit FEM discretization $\longrightarrow$ YES
- Efficient coupling/decoupling strategies for solver $\longrightarrow$ YES (?)
- Adaptive meshing/Error control of user-specific quantities $\longrightarrow$ YES (???)

$\rightarrow$ **Prototypical for (multiphase) flow problems ?**

But: Problems with large deformations ? Remeshing of solid interfaces ?





Similar techniques (implicit reconstruction, accurate and oscillation-free transport solvers) as for multiphase flow ?