FEM Techniques for Particulate Flow

Stefan Turek, Decheng Wan

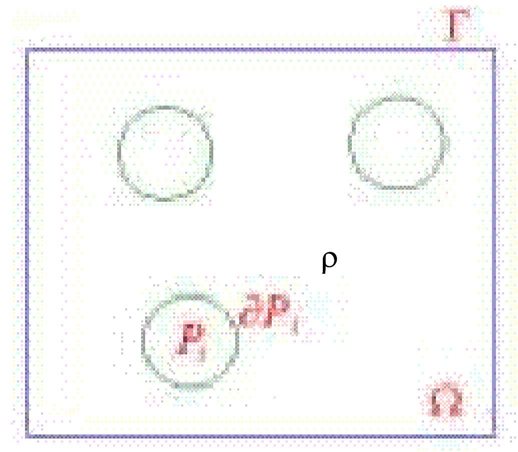
Institut für Angewandte Mathematik, Univ. Dortmund
http://www.mathematik.uni-dortmund.de/LS3
http://www.featflow.de

- Model for particulate flow
- Numerical techniques

Fluid – (Rigid) Solid Interfaces

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 $\Omega_p(t)$ 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,…,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 the **collision forces** (later).

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

2. **Explicit coupling**

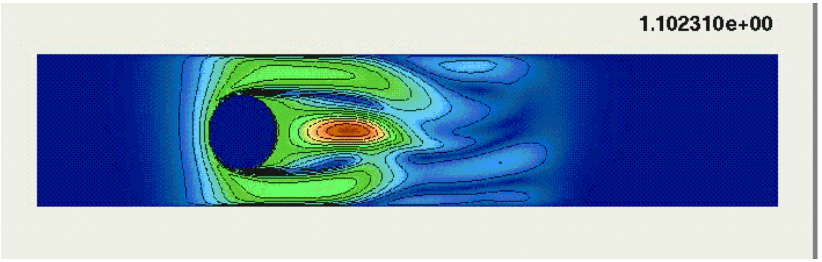
   t^n fluid $\rightarrow$ t^n force on solid $\rightarrow$ t^{n+1} solid $\rightarrow$ t^{n+1} fluid ....

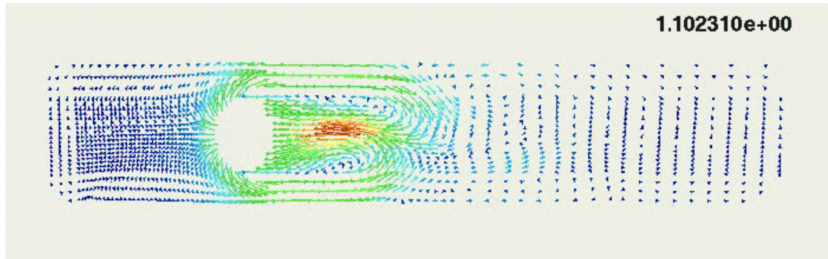
- FVM-fictitious domain methods (Duchanoy and Jongen(2003))
- **FEM-fictitious boundary methods** (Turek,Wan and Rivkind)

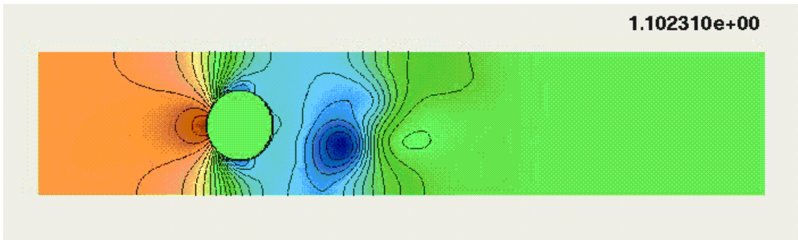
The ‘Fictitious Boundary Method (FBM)’

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) independent of ‘internal objects’

How to Calculate Forces with FBM?

Hydrodynamic forces and torque acting on the i-th particle

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

- FBM: Reconstruction of the shape is only first order accurate
  ➔ local grid adaptivity or alignment
  ➔ ”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.

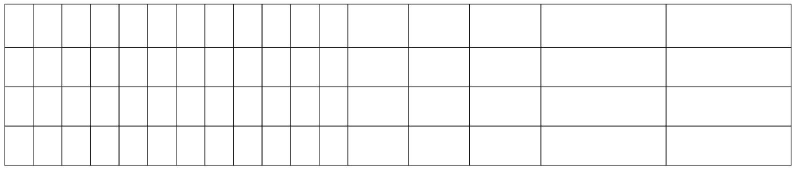
$$\mathbf{n}_p = \nabla \alpha_p$$

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

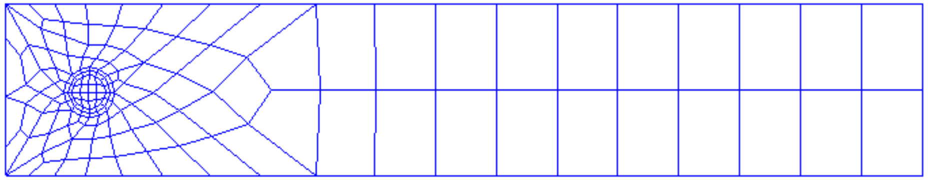
$$F_p = -\int_{\Gamma_p} \sigma \cdot \mathbf{n}_p \, d\,\Gamma_p = -\int_{\Omega_T} \sigma \cdot \nabla \alpha_p \, d\,\Omega_T$$

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

Evaluation of Force Calculations

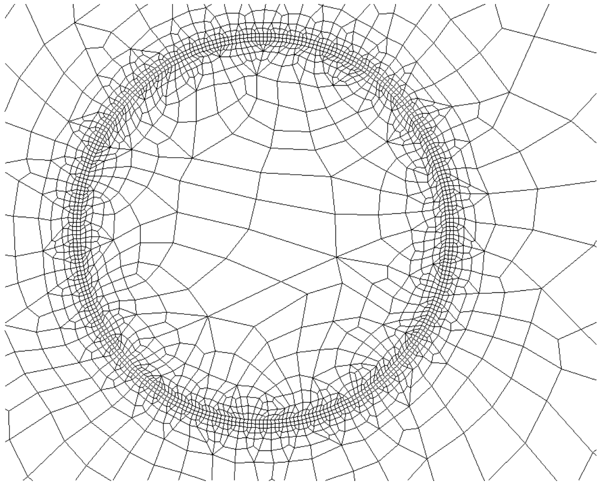


LEVEL 6 ≈ 280.000 elements



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

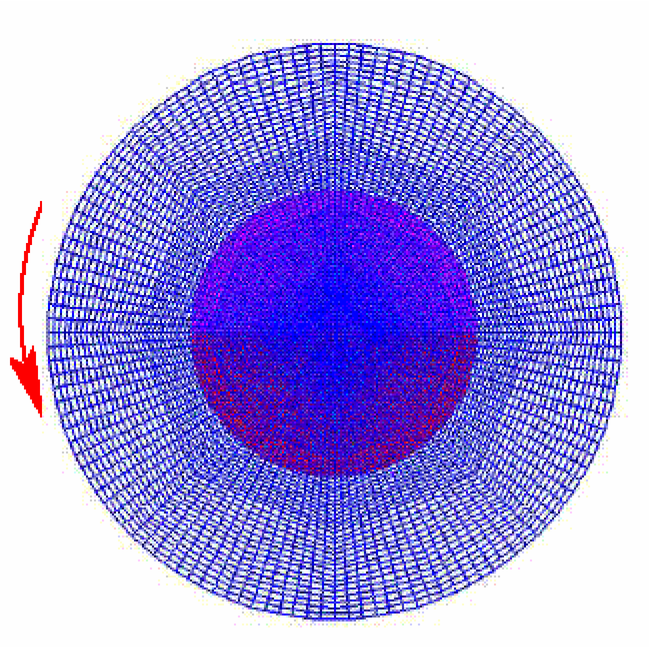
The algorithm for $t^n \rightarrow t^{n+1}$ consists of the following 4 substeps

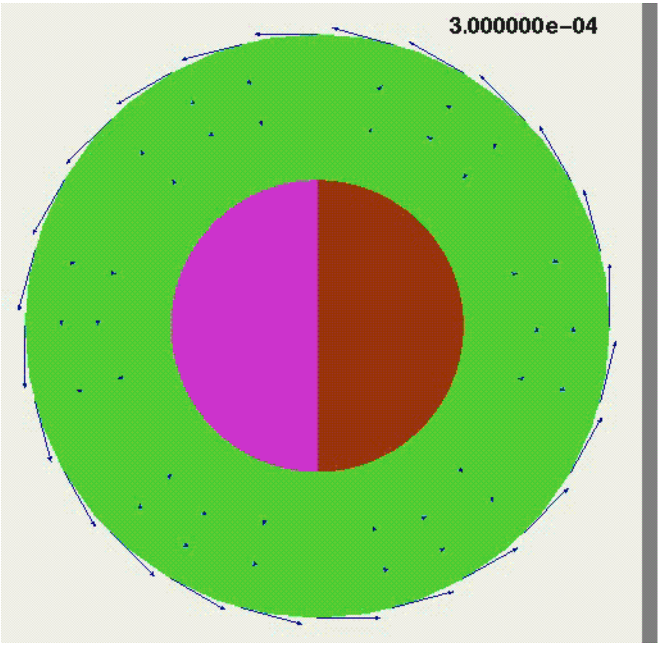
1. Fluid velocity and pressure : $NSE(\mathbf{u}_f^{n+1}, p^{n+1}) = BC(\Omega_p^n, \mathbf{u}_p^n)$
2. Calculate hydrodynamic forces: $\mathbf{F_p}^{n+1}$
3. Calculate velocity of particles: $\mathbf{u}_p^{n+1} = g(\mathbf{F_p}^{n+1})$
4. Update position of particles: $\Omega_p^{n+1} = f(\mathbf{u}_p^{n+1})$

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

Numerical Examples

'One particle in a rotating circular container'



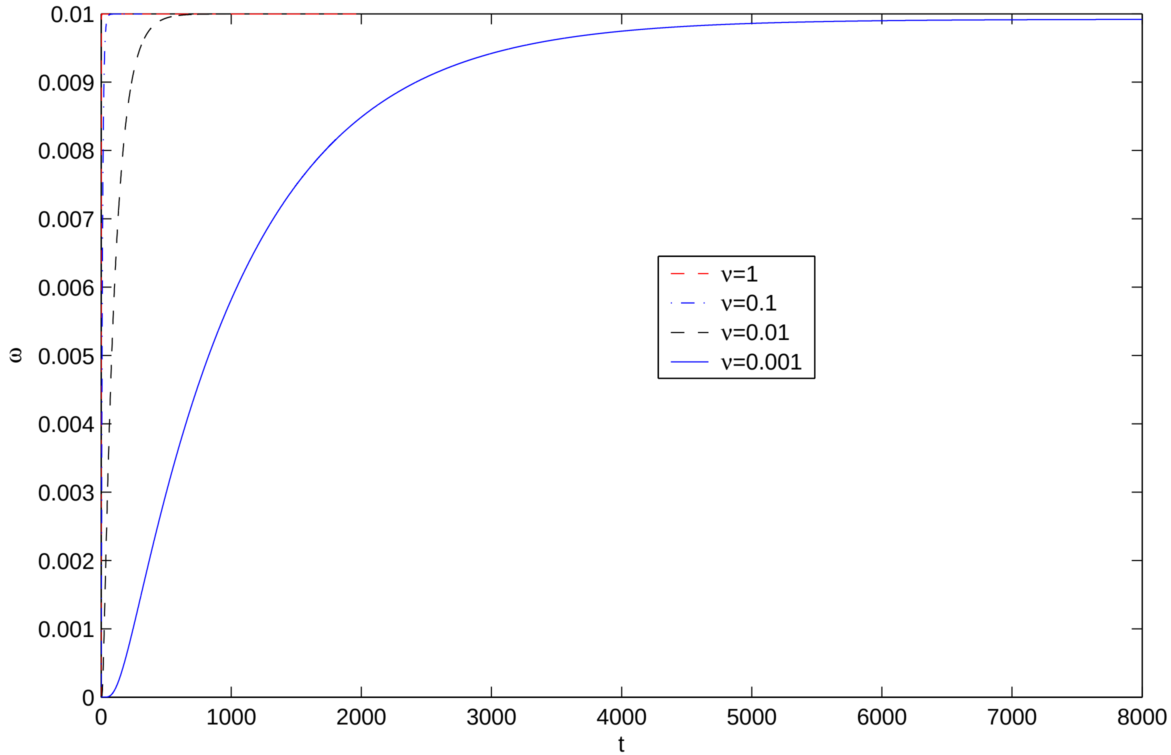


$\Omega = 0.01$

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

Numerical Examples

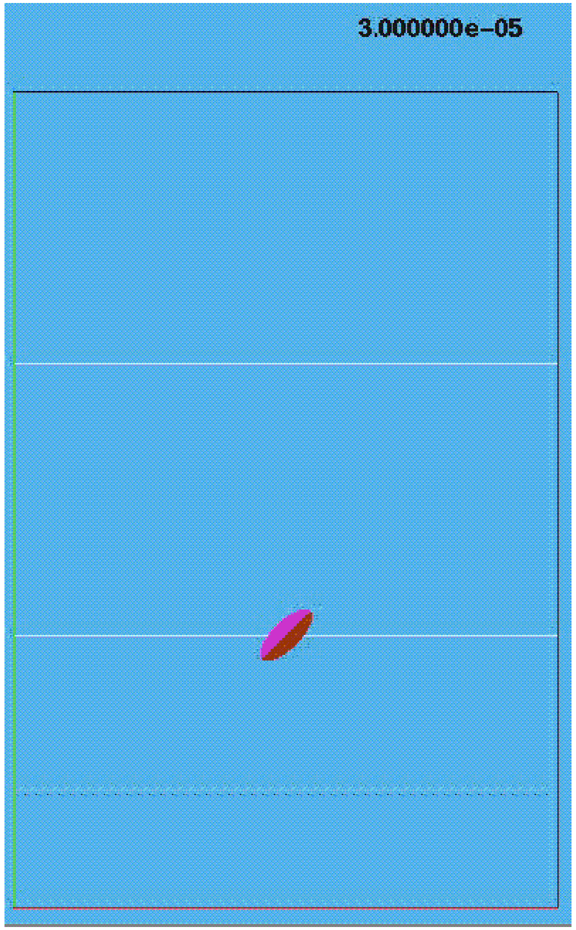
‘One particle in a rotating circular container’

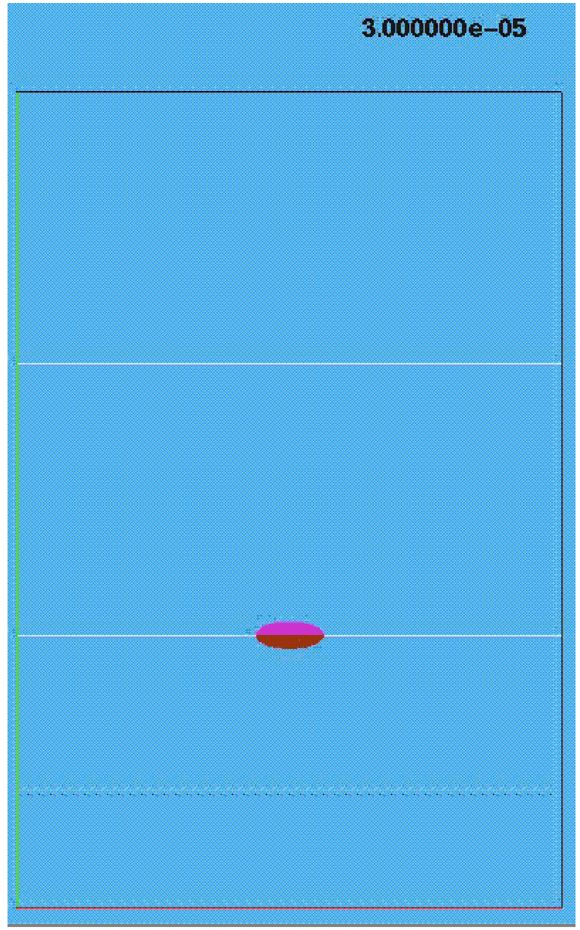


viscosity ν	Terminal angular velocity ω_p	Time reaching the steady state
0.001	0.0099185	7000.0
0.01	0.0099989	600.0
0.1	0.0099998	60.0
1.0	0.0099999	10.0

Numerical Examples

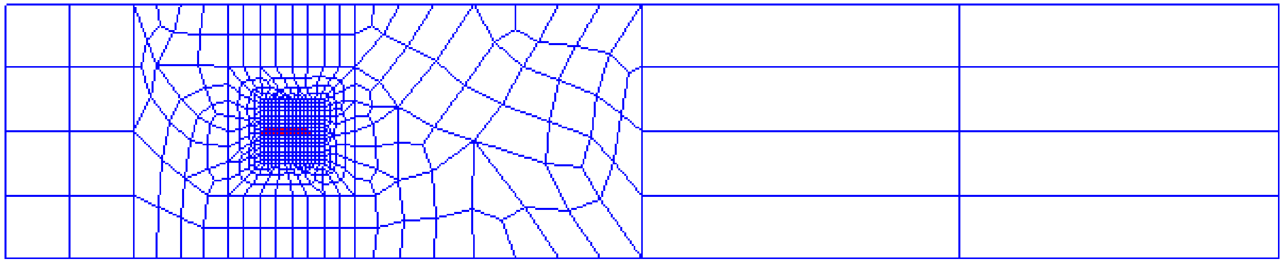
‘One ellipse falling in an (infinite) channel’

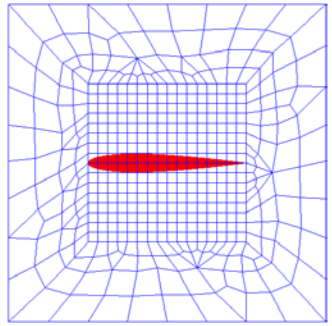


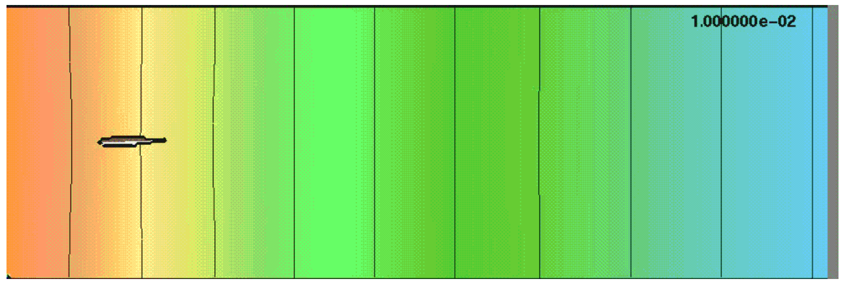


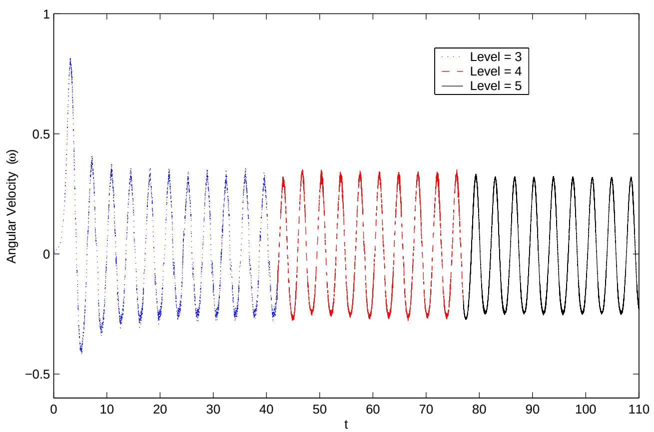
Numerical Examples

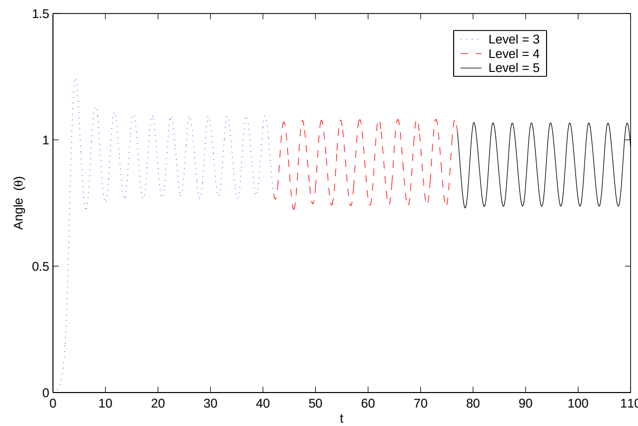
‘Viscous flow around a moving airfoil’ (Glowinski)



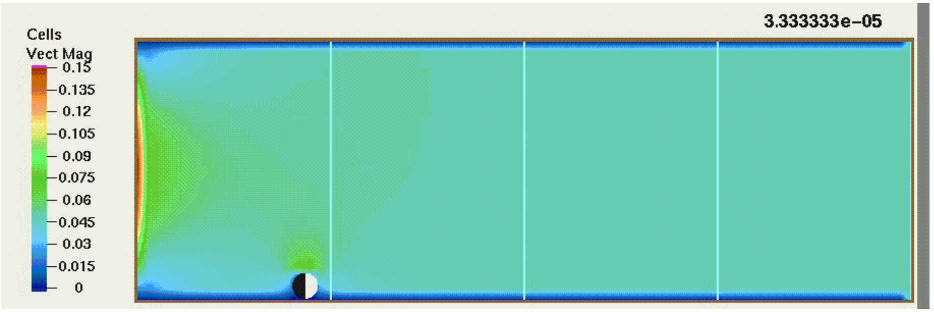


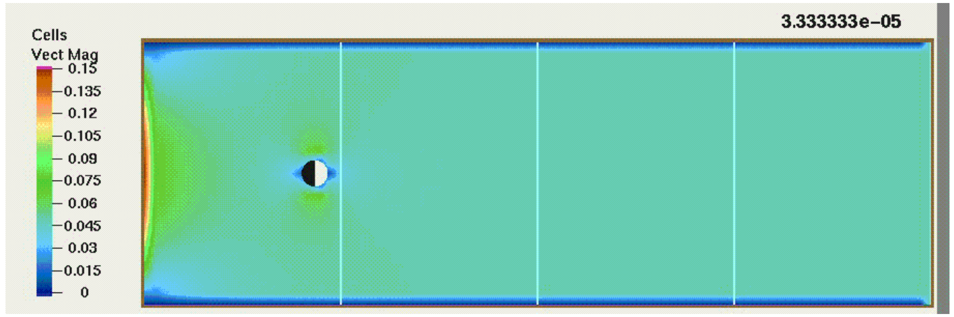






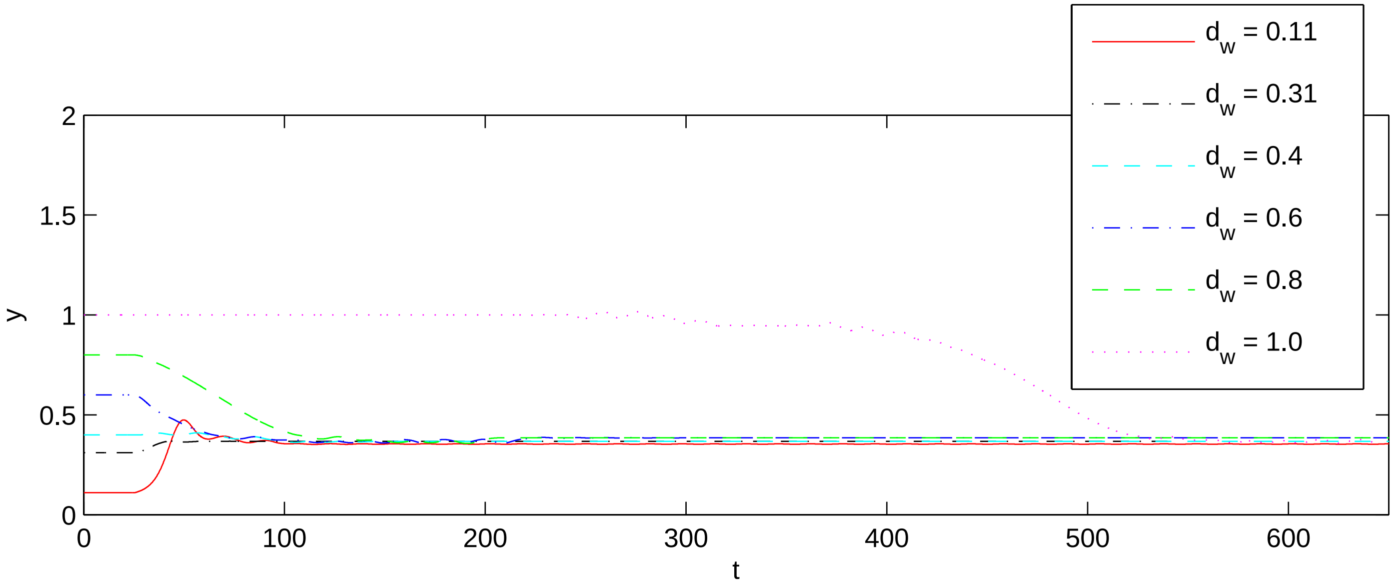
Lift-Off for Circle





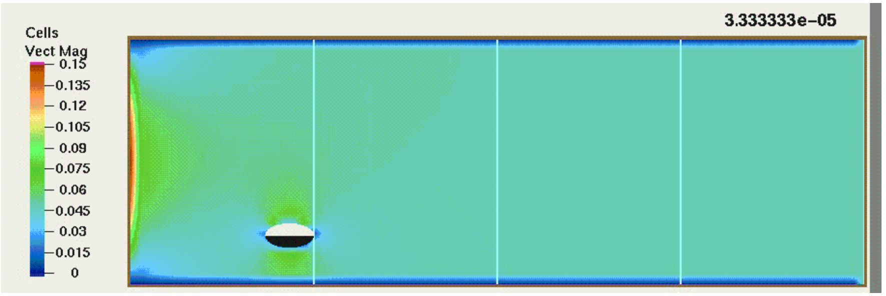
Velocity ($d_w = 0.1$)

Velocity ($d_w = 1.0$)

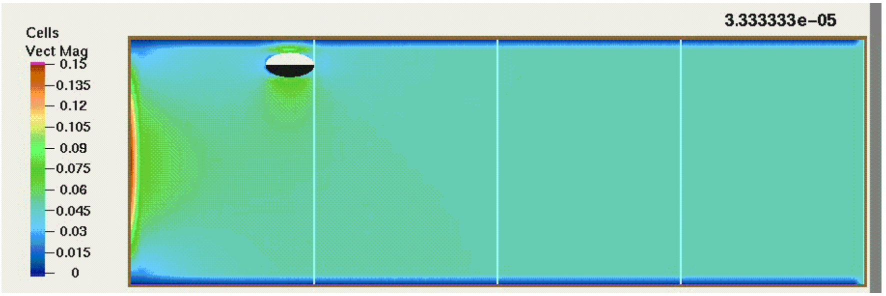


y of center of ball

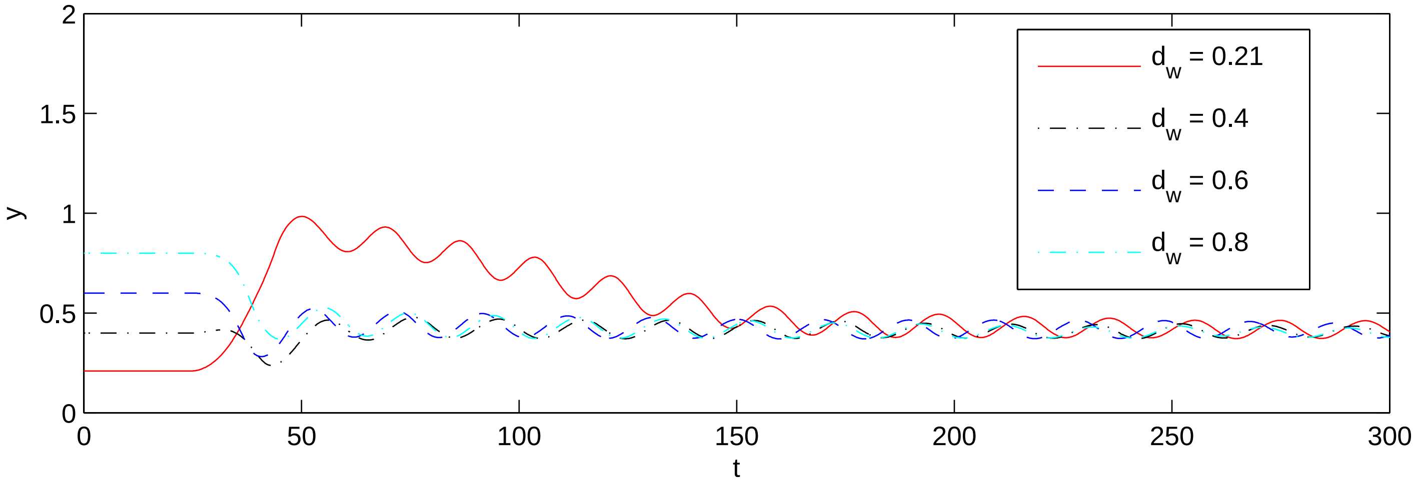
Lift-Off for Ellipse



Velocity ($d_w = 0.4$)



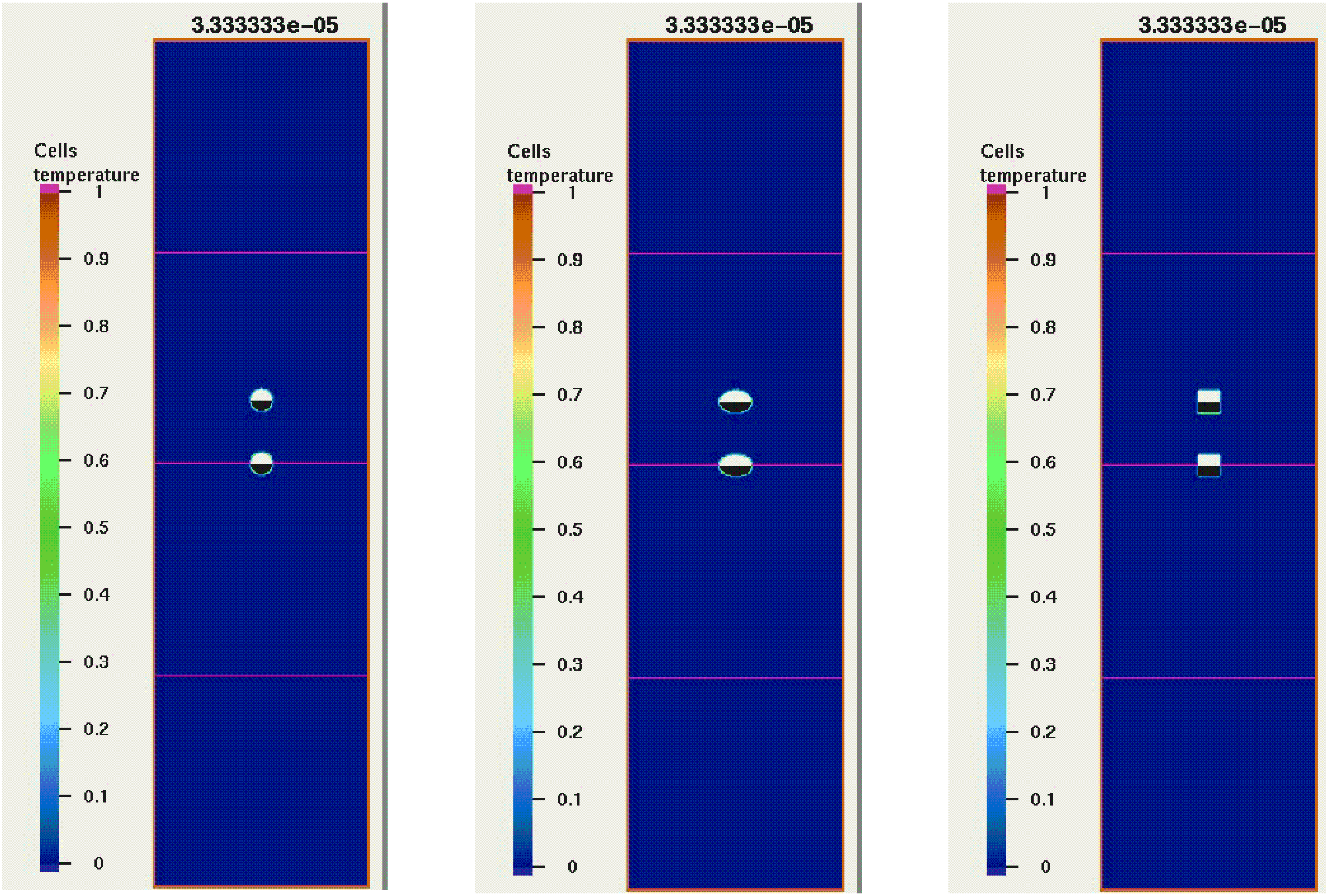
Velocity ($d_w = 1.8$)



y of center of ellipse

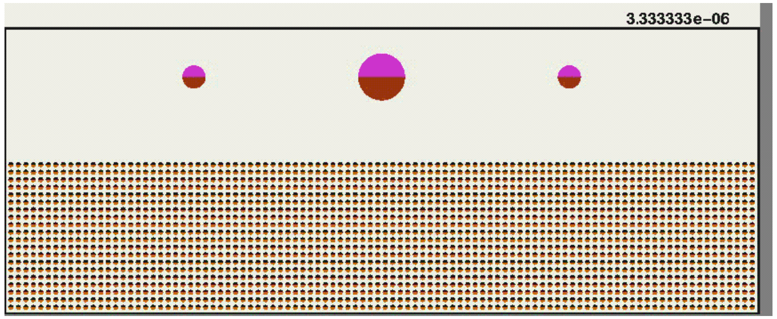
Numerical Examples

‘Kissing, Drafting, Thumbling’

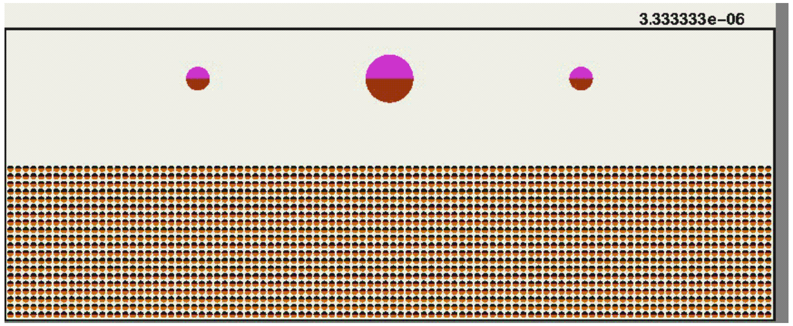


Numerical Examples

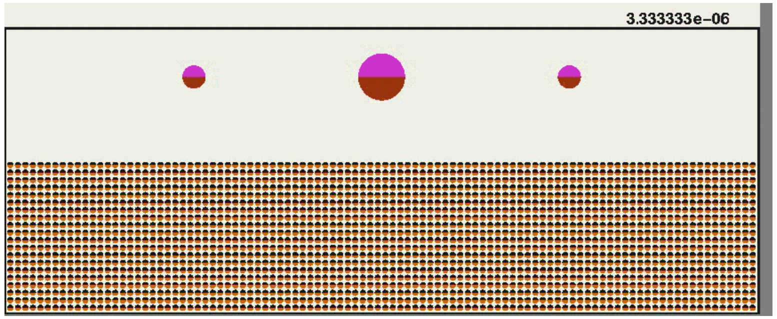
'Impact of heavy balls on 2000 small particles'



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



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

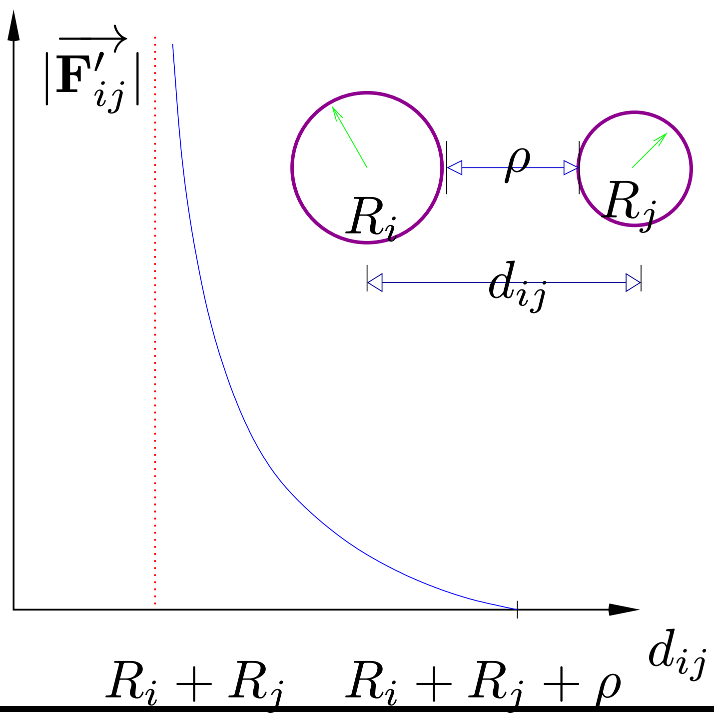


$$\rho_f = 1,\ \rho_{bd} = 2,\ \rho_{sp} = 20$$

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.



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

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:

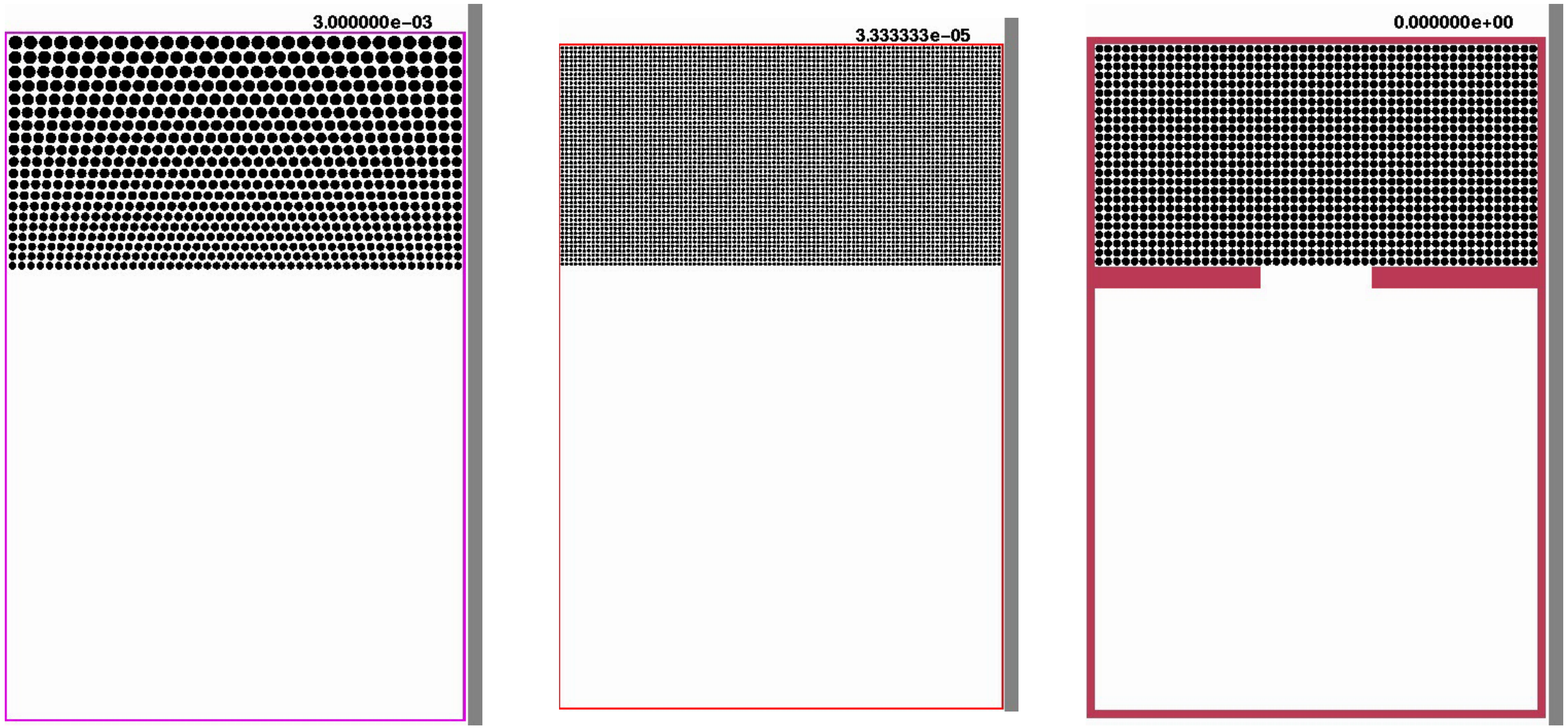
$$\mathbf{F}_{i,j}^{P} = \begin{cases} 0 & \text{for } d_{i,j} > R_i + R_j + \rho \\ \frac{1}{\epsilon_P}(\mathbf{X}_i - \mathbf{X}_j)(R_i + R_j + \rho - d_{i,j})^2 & \text{for } R_i + R_j \le d_{i,j} \le R_i + R_j + \rho \\ \frac{1}{\epsilon'_P}(\mathbf{X}_i - \mathbf{X}_j)(R_i + R_j - d_{i,j}) & \text{for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

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

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

Numerical Examples

‘Fluidization/Sedimentation of many particles’



Efficient Data Structures

$L3 \approx 220.000\, elements \qquad \approx 1.100.000\, d.o.f.s$

$L4 \approx 880.000\, elements \qquad \approx 4.400.000\, d.o.f.s$

$L5 \approx 3.530.000\, elements \qquad \approx 17.600.000\, d.o.f.s$

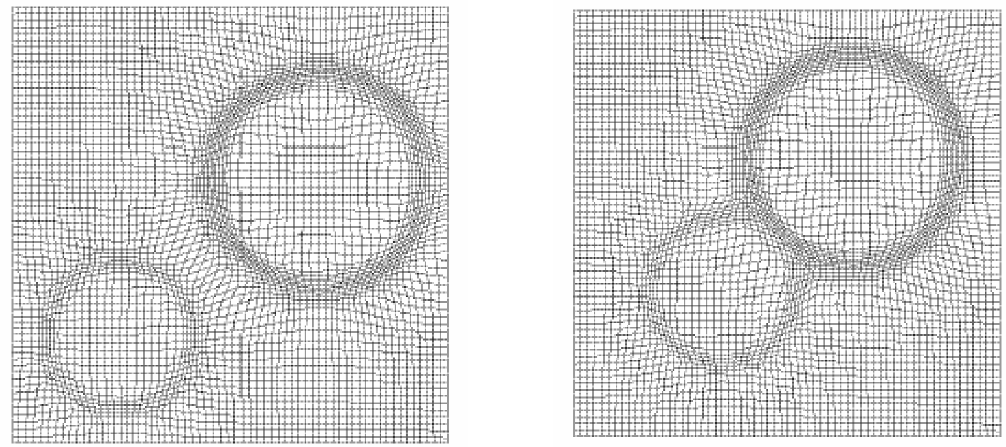
DEC/COMPAQ EV6, 833 MHz

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) ???

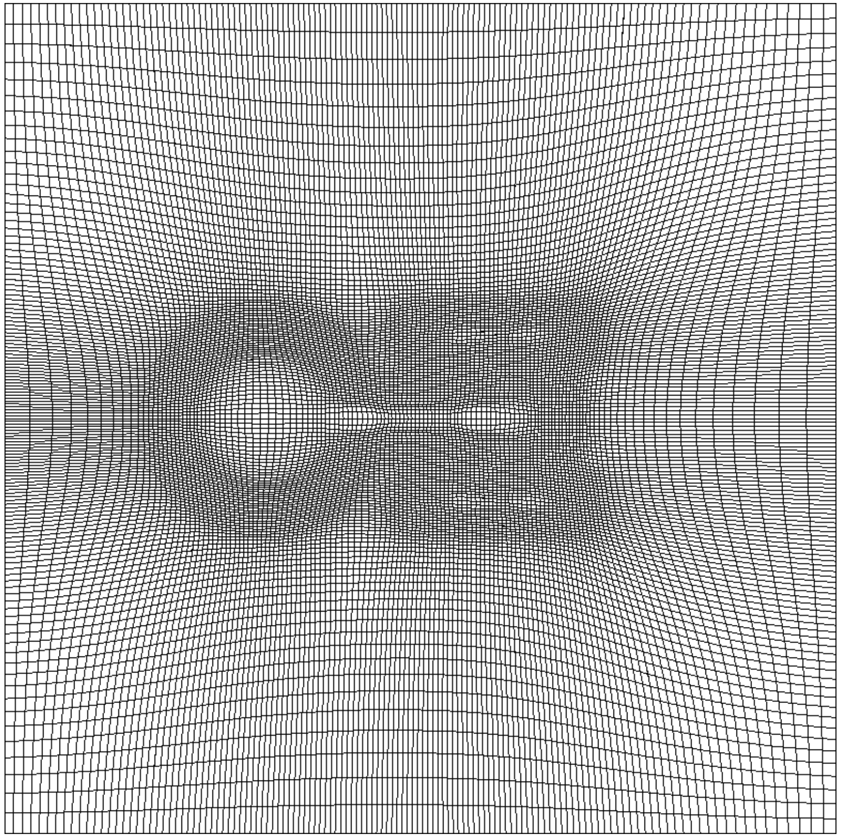
Challenges

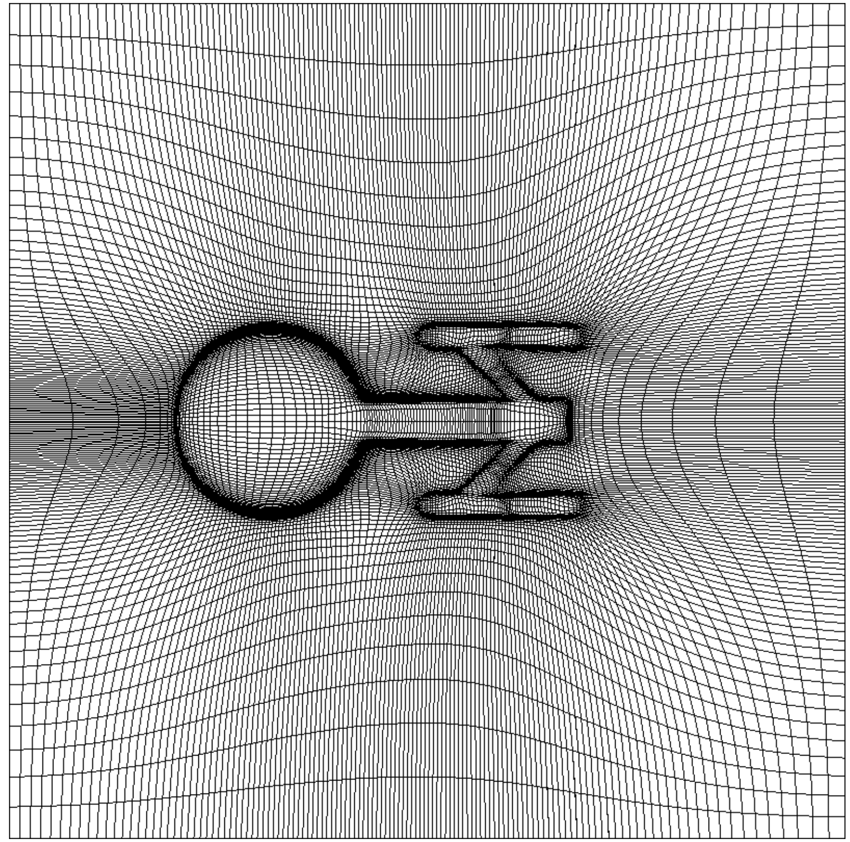
- Adaptive time stepping + dynamical adaptive grid alignment/ALE



- (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.

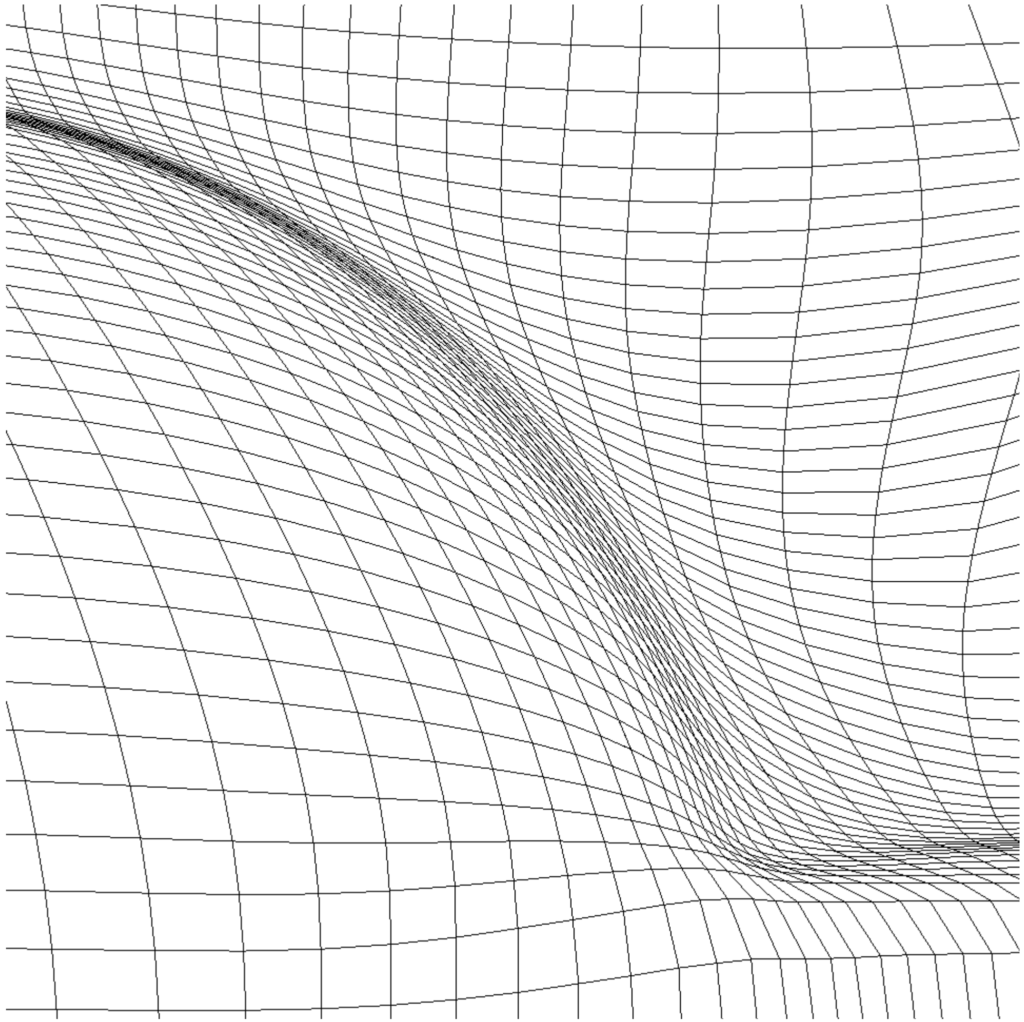
Example for Deformed Meshes





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

Example for Deformed Meshes



Exact control and smooth transitions

Last Example