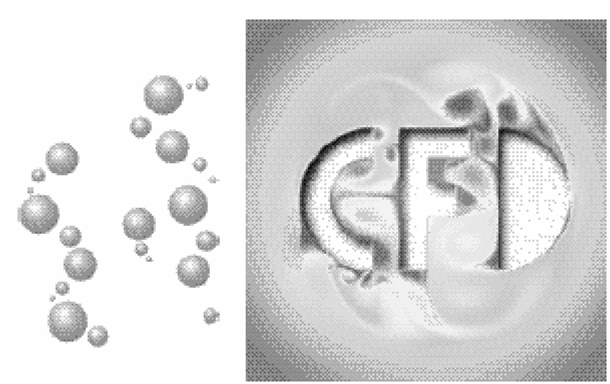
FEM Techniques for Particulate Flow

Stefan Turek

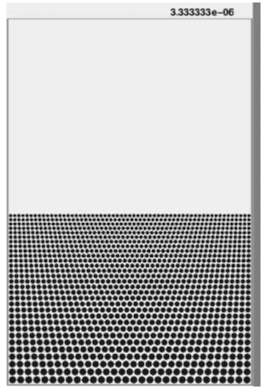
Institut für Angewandte Mathematik, Univ. Dortmund
http://www.mathematik.uni-dortmund.de/LS3
http://www.featflow.de

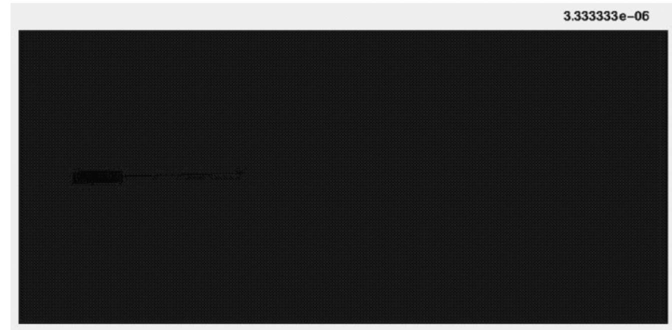
- Multiphase flow problems
- Model for particulate flow
- Numerical techniques

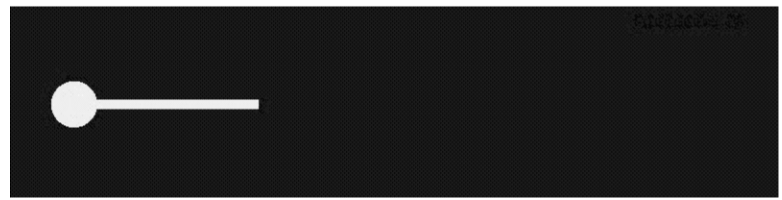


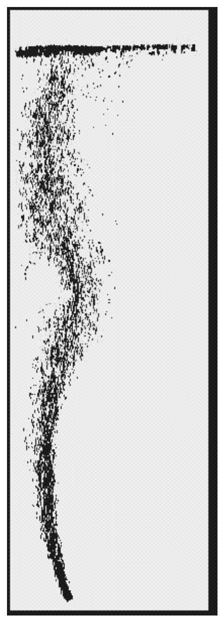
Multiphase Flow Problems

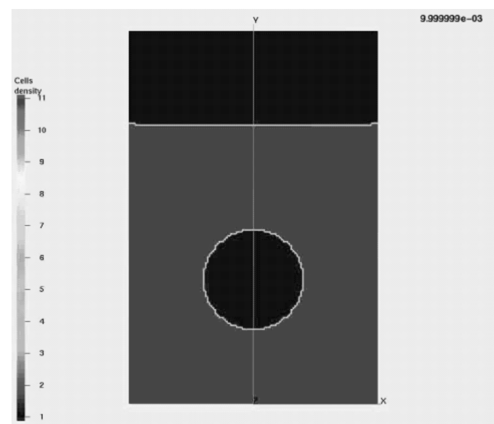
- ***Liquid – (rigid) solid interfaces***
- **Liquid – (elastic) solid interfaces**
- **Liquid – gas interfaces**
- **Liquid – liquid interfaces**







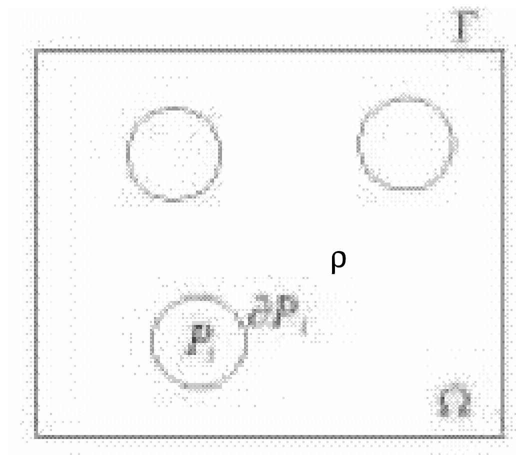




Fluid – (Rigid) Solid Interfaces

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 $\Omega_f(t)$

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

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

$$\sigma(X\,;t) = -pI + \mu\left[\nabla u + (\nabla u)^T\right]$$

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

$$M_p \frac{dU_p}{dt} = F_p + F_p' + (\Delta M_p)\, g; \qquad I_p \frac{d\omega_p}{dt} + \omega_p \times (I_p \omega_p) = T_p$$

with M_p the mass of the p-th particle (p=1,…,N);

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

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

and F'_p are the **collision forces** (later).

X_p is the position of the center of gravity of the p-th particle;

$\Gamma_p = \partial\,{}_p$ the boundary of the p-th particle;

n_p is the unit normal vector on the boundary Γ_p

Interaction between Particle and Fluid

No slip boundary conditions at interface $¡_p$ between particles and fluid
i.e., for any $X \; 2 \; ¡_p$, the velocity u(X) is defined by:

$$u(X) = U_p + !_p \; £ \; (X \; ¡ \; X_p)$$

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

$$\frac{dX_p}{dt} = U_p\,; \qquad \frac{d\mu_p}{dt} = !_p$$

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+! x 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

$\boxed{t^n\ \text{°uid}}$! $\boxed{t^n\ \text{force on solid}}$! $\boxed{t^{n+1}\ \text{solid}}$! $\boxed{t^{n+1}\ \text{°uid}}$ ....

- FVM-fictitious domain methods (Duchanoy and Jongen(2003))

- **FEM-fictitious boundary methods** (Turek,Wan and Rivkind)

Further Classification

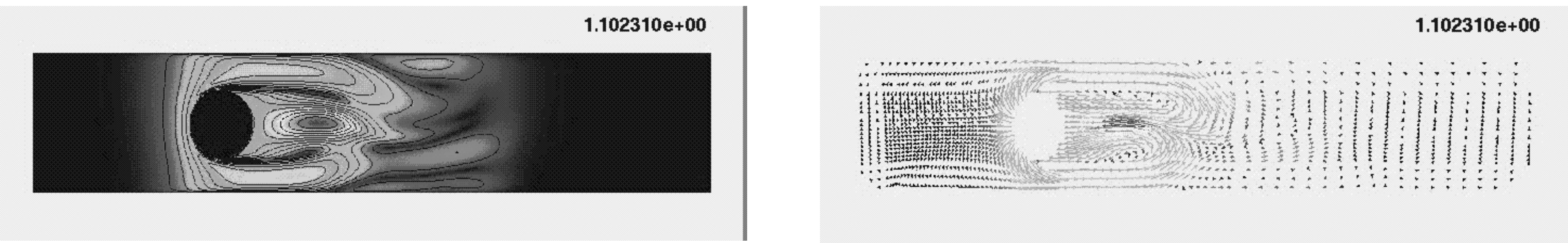
1. **Eulerian approach:** fixed meshes!
   Use a **fixed** mesh that covers the whole domain where the fluid may be present.
   - The distributed Lagrange multiplier/fictitious domain method
   - **FEM-fictitious boundary method**
   - FVM-fictitious domain method

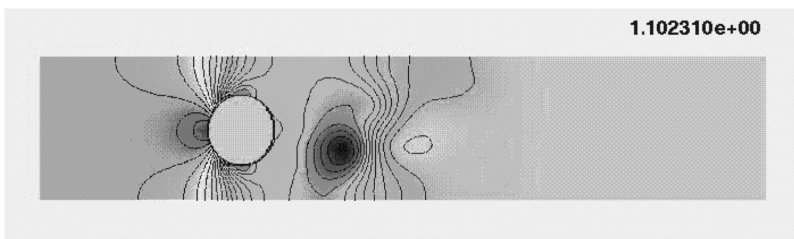
2. **Lagrangian approach:** moving meshes!
   Based on a moving mesh which follows the motion of the fluid boundary.
   - Arbitrary Lagrangian Eulerian (Hu,Joseph and Crochet (1992), Johnson and Tezduyar (1996))
   - Fat boundary method (Maury (2001))

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) independent of 'internal objects'

How to Calculate (Surface) Forces?

Hydrodynamic forces and torque acting on the i-th particle

$$F_i = -\int_{\partial P_i} \sigma \cdot n_i \, d\Gamma_i \, ; \qquad T_i = -\int_{\partial P_i} (X - X_i) \times (\sigma \cdot n_i) \, d\Gamma_i$$

- 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 n_p defined on the global grid.

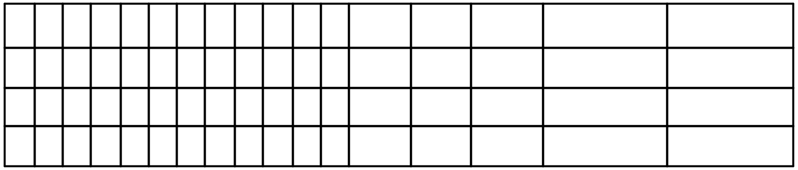
$$n_p = \nabla \alpha_p$$

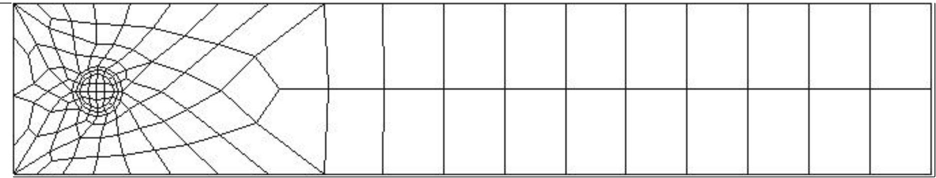
Force acting on the wall surface of the particles can be computed by

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

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

Evaluation of Force Calculations

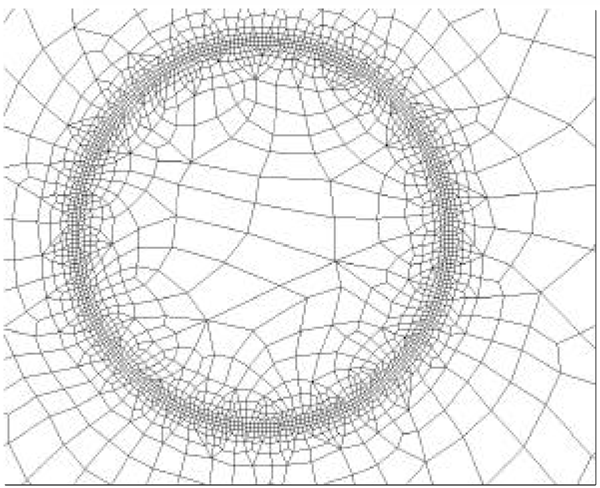




LEVEL 6 ¼ 280.000 elements

LEVEL 6 ¼ 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 ¼ 150.000 elements

(Explicit) Operator-Splitting Approach

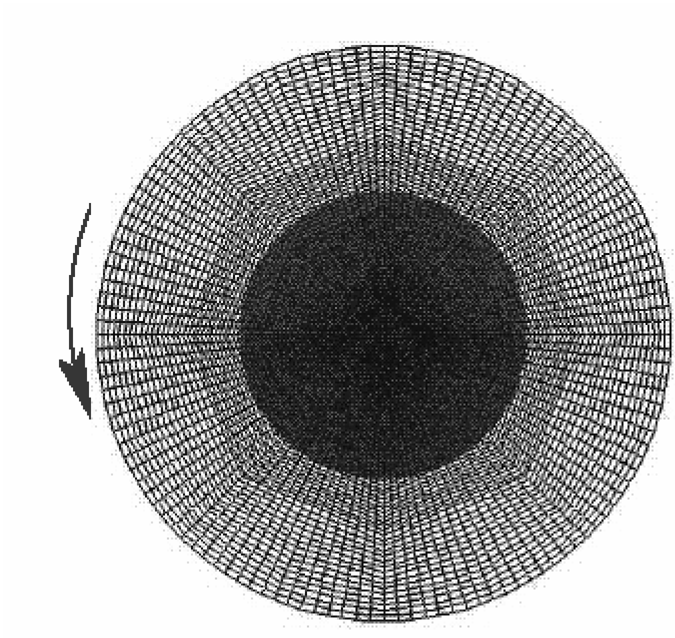
The algorithm for $t^n \; ! \; t^{n+1}$ consists of the following 4 substeps

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

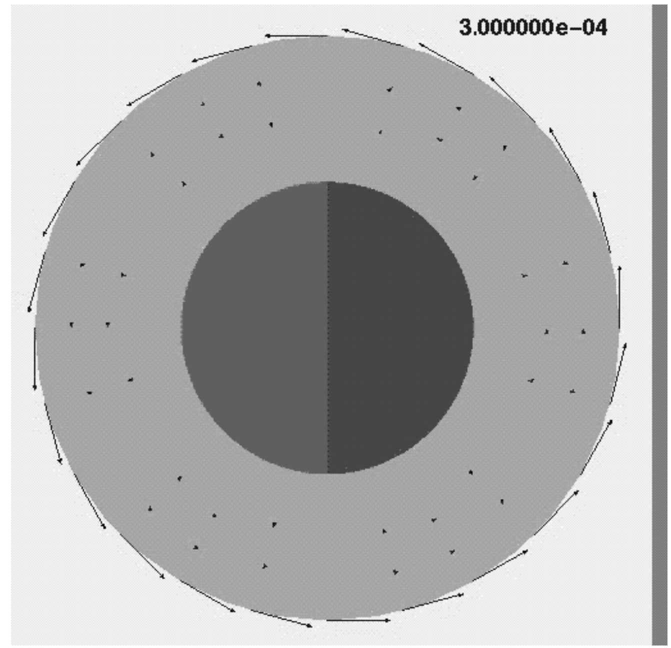
? Required: efficient calculation of hydrodynamic forces
? Required: efficient treatment of particle interaction (?)
? Required: fast (nonstationary) Navier-Stokes solvers (!)

Numerical Examples

‘One particle in a rotating circular container’



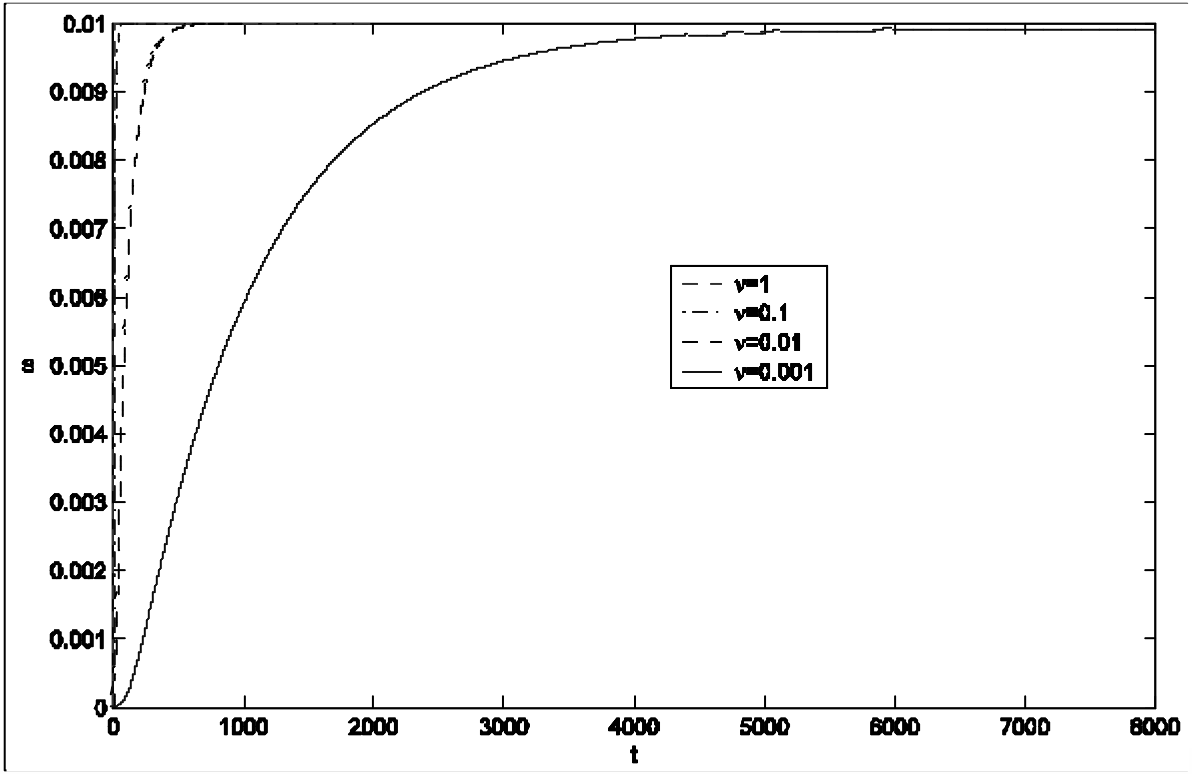
$\Omega = 0.01$



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

Numerical Examples

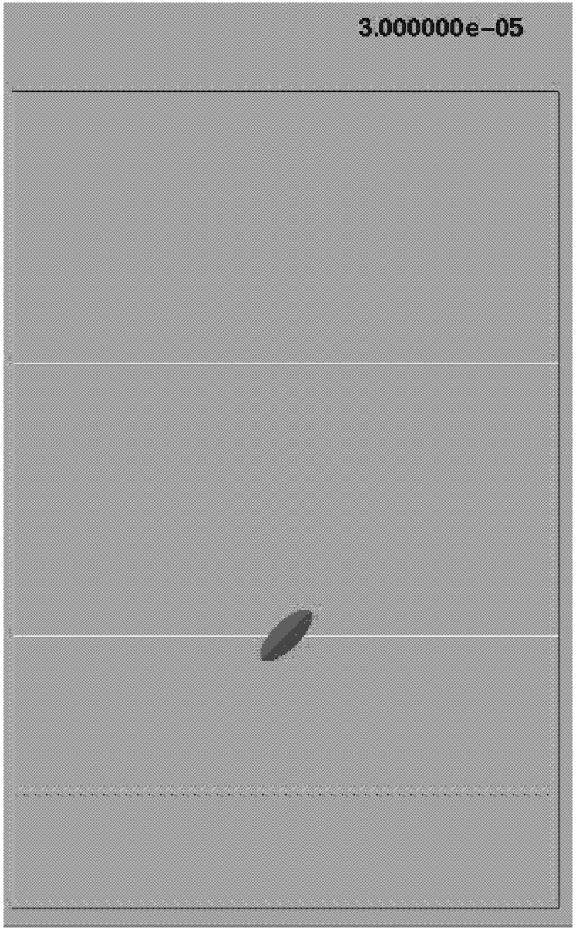
‘One particle in a rotating circular container’

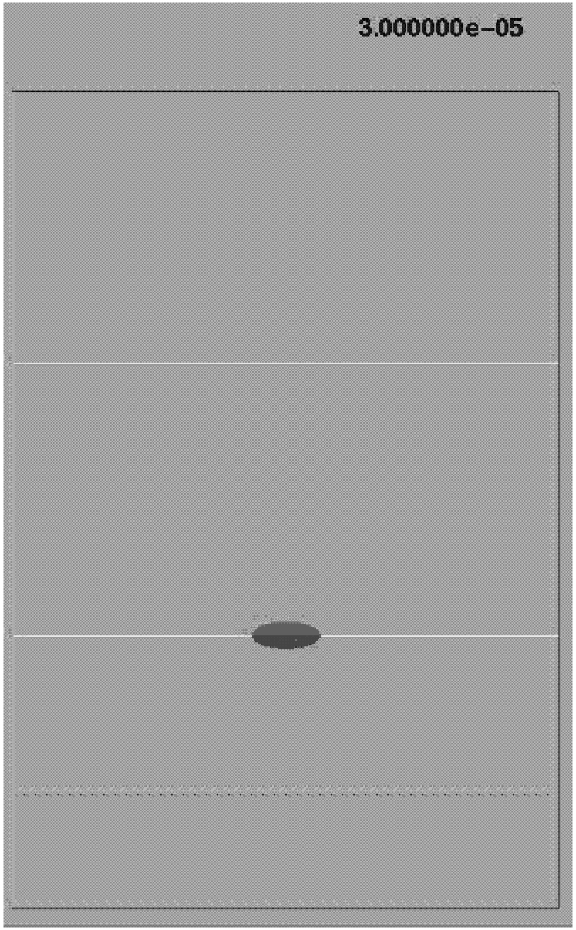


viscosity ν	Terminal angular velocity ω_p	Time reaching the steady state
0.001	0.0099185	7000.0
0.01	0.0099989	600.0
0.1	0.0099998	60.0
1.0	0.0099999	10.0

Numerical Examples

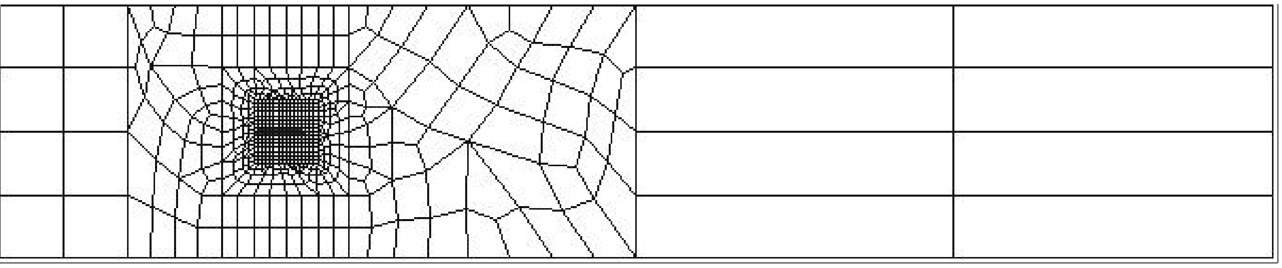
'One ellipse falling in an (infinite) channel'

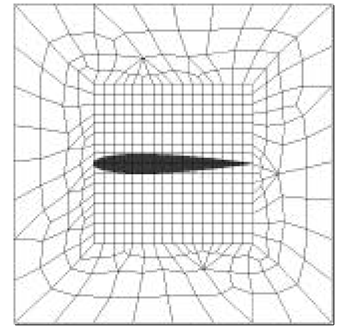


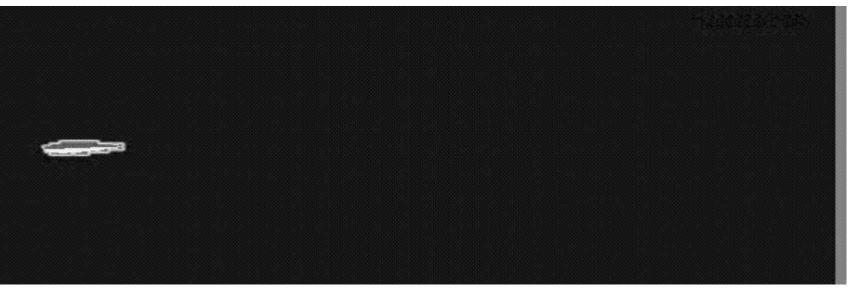


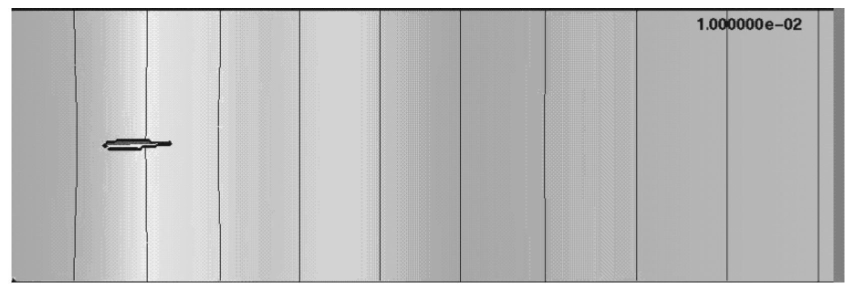
Numerical Examples

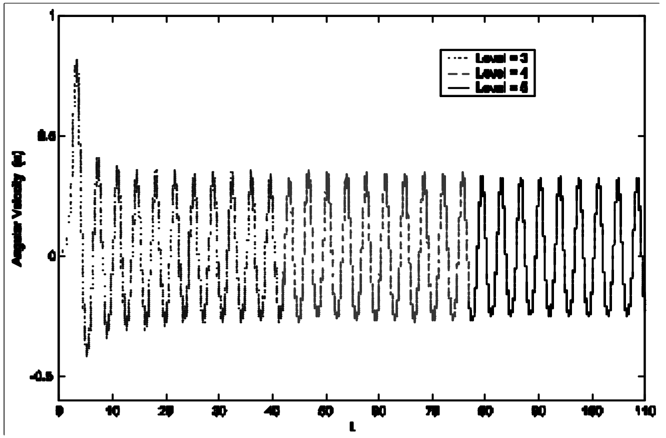
'Viscous flow around a moving airfoil' (Glowinski)

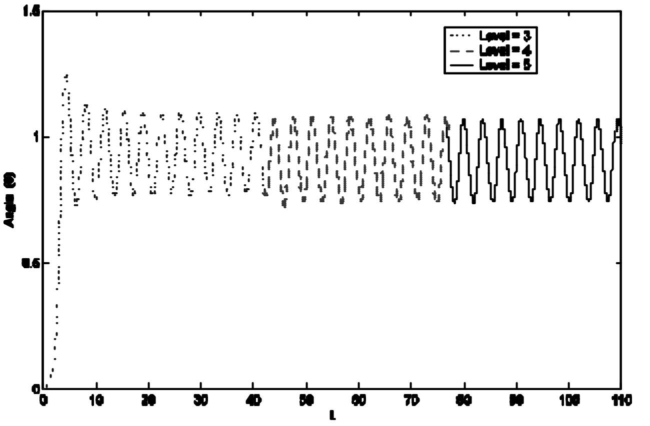






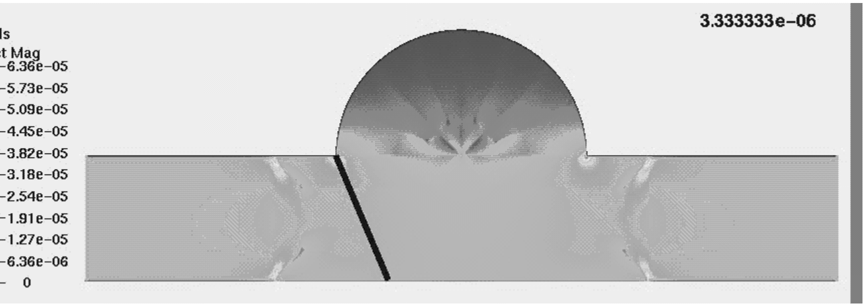


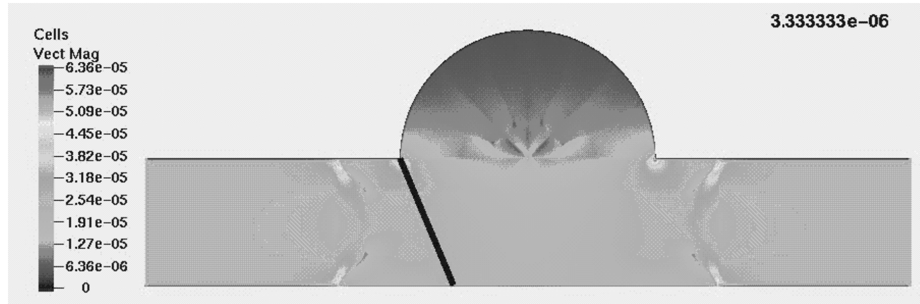




Numerical Examples

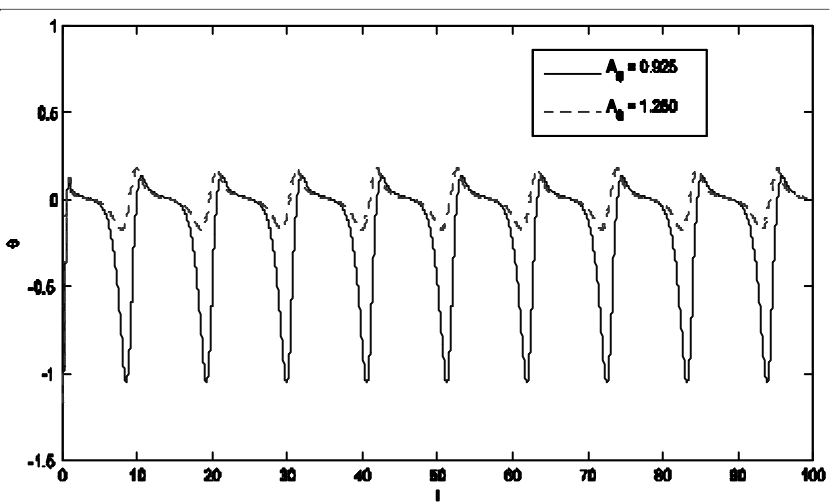
‘(Prototypical) Heart Valve’



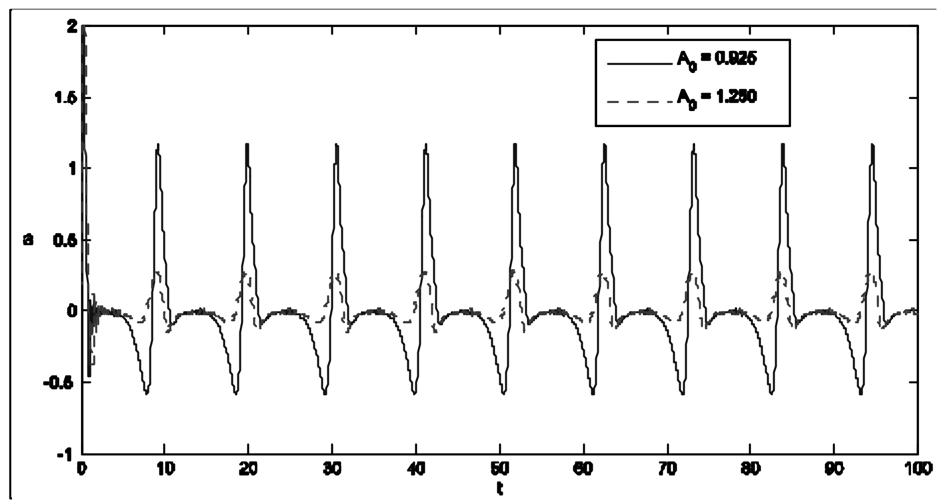


Velcoity (A_0 = 0:925) Velocity (A_0 = 1:250)

Inlet velocity $U = 9{:}828(A_0 + \sin(\frac{1{:}85\,t}{1/4}))$



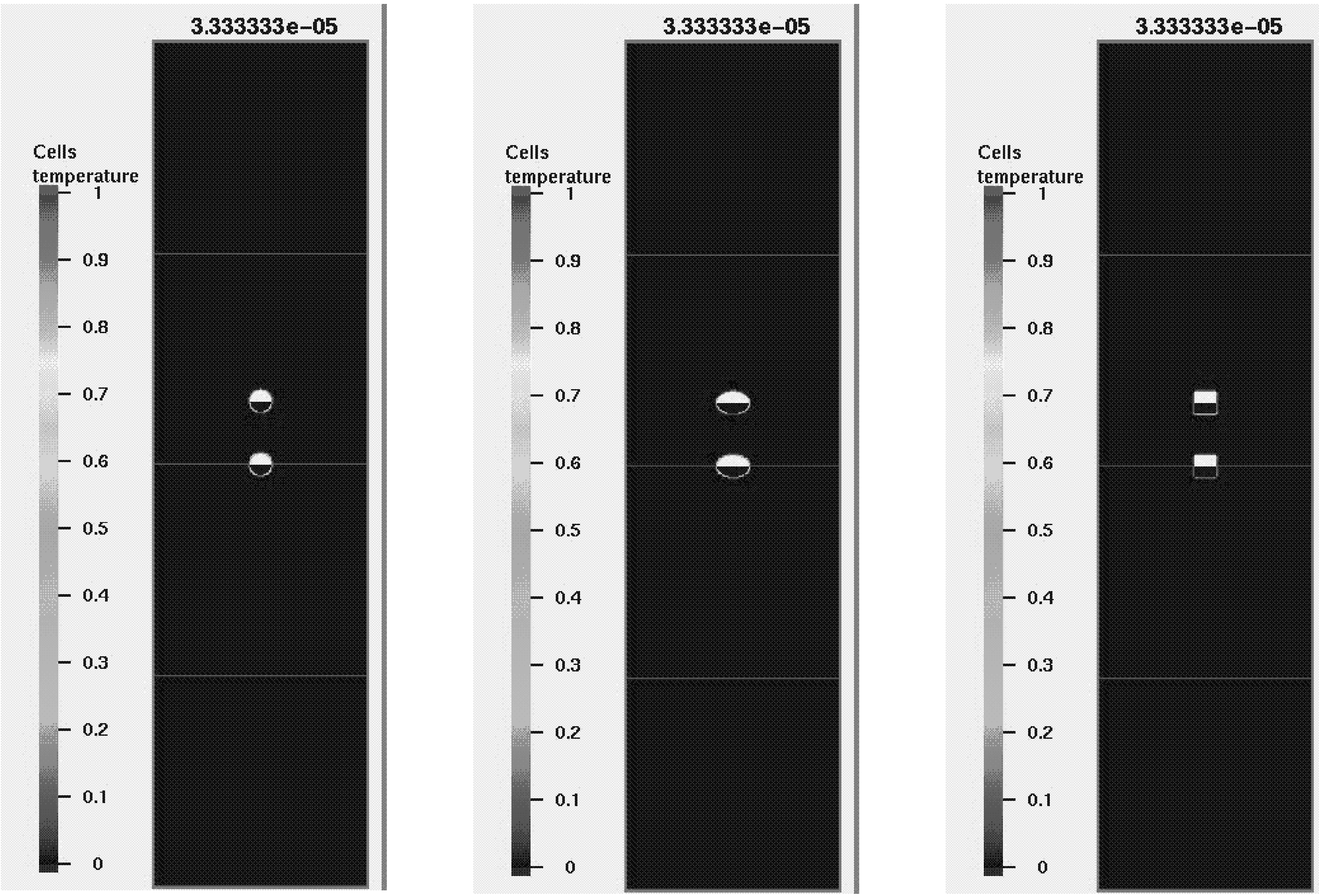
(a) angle



(b) angular velocity

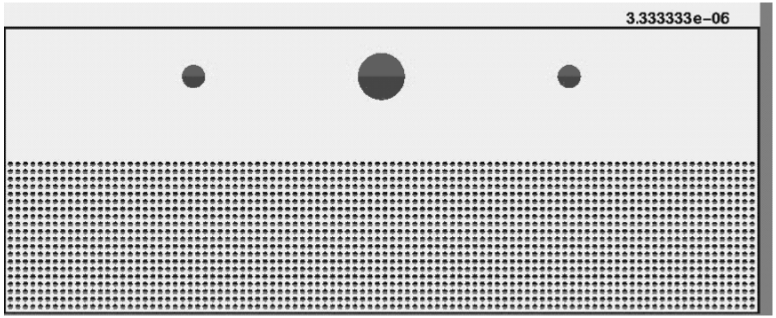
Numerical Examples

‘Kissing, Drafting, Thumbling’

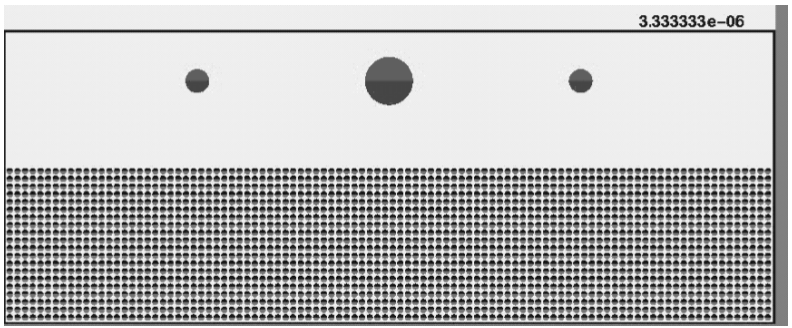


Numerical Examples

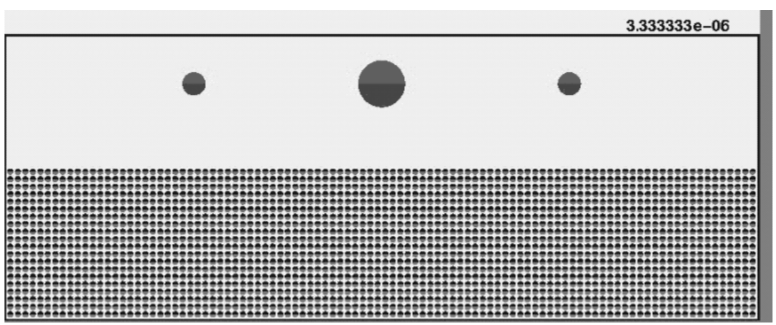
‘Impact of heavy balls on 2000 small particles’



½$_{r}$ = 1, ⅓$_{bd}$ = 2, ½$_{sp}$ = 1:1



½$_{r}$ = 1, ⅓$_{bd}$ = 2, ½$_{sp}$ = 2

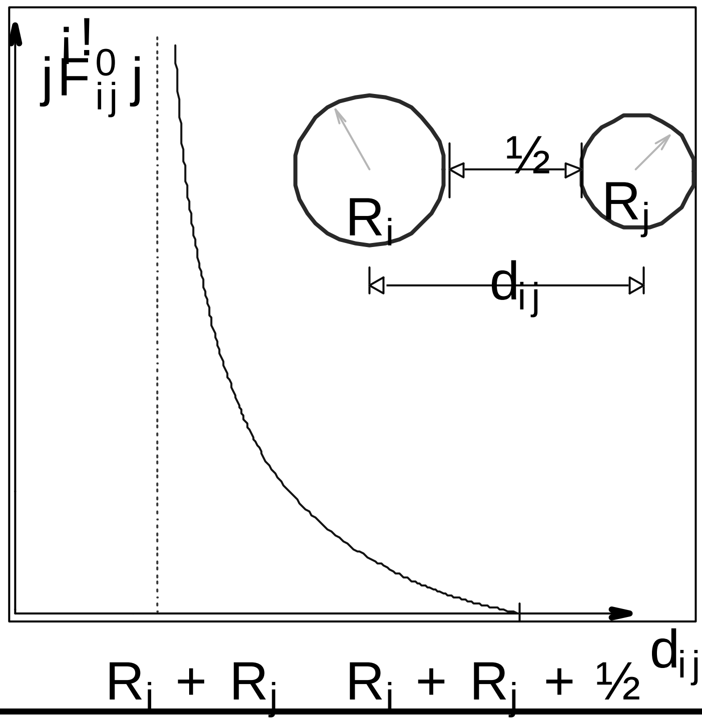


½$_{r}$ = 1, ⅓$_{bd}$ = 2, ½$_{sp}$ = 20

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.



$$
\begin{cases}
|\vec{F}^0_{ij}| = 0 & \text{if } d_{ij} \geq R_i + R_j + ½; \\
|\vec{F}^0_{ij}| = d_{ij} =\text{''} & \text{if } d_{ij} = R_i + R_j :
\end{cases}
$$

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:

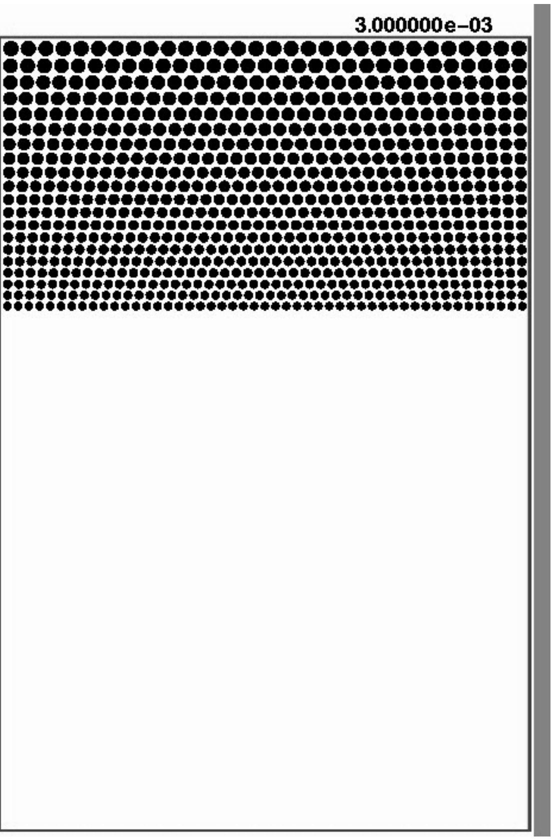
$$F^P_{i,j} = \begin{cases} 0 & \text{for } d_{i,j} > R_i + R_j + \rho \\ \frac{1}{\varepsilon_P}(X_i - X_j)(R_i + R_j + \rho - d_{i,j})^2 & \text{for } R_i + R_j \leq d_{i,j} \leq R_i + R_j + \rho \\ \frac{1}{\varepsilon'_P}(X_i - X_j)(R_i + R_j - d_{i,j}) & \text{for } d_{i,j} \leq R_i + R_j \end{cases}$$

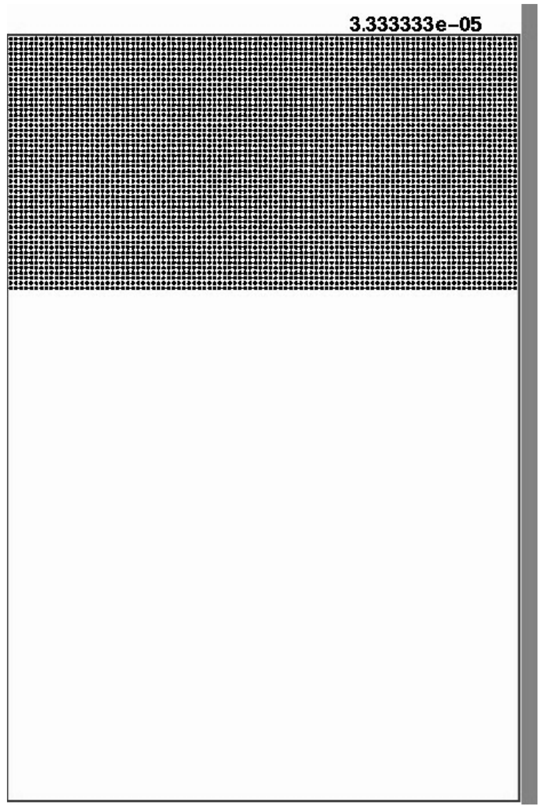
The total repulsive forces exerted on the i-th particle by the other particles and the walls can be expressed as follows:

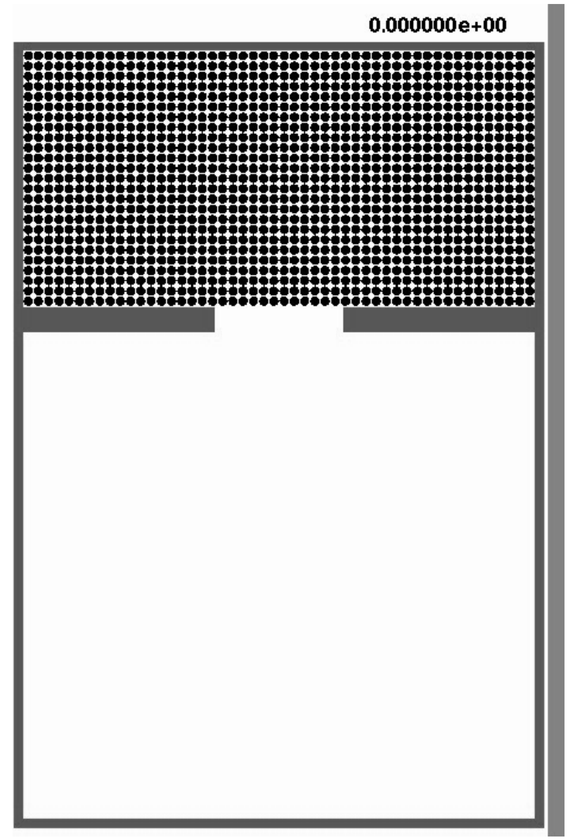
$$F'_i = \sum_{j=1, j \neq i}^{N} F^P_{i,j} + F^W_i$$

Numerical Examples

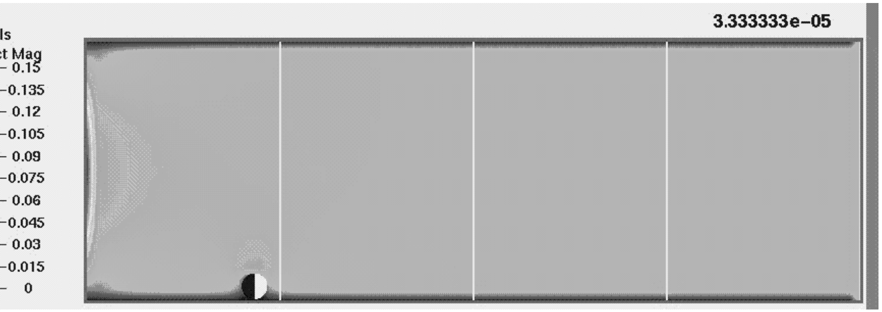
'Fluidization/Sedimentation of many particles'



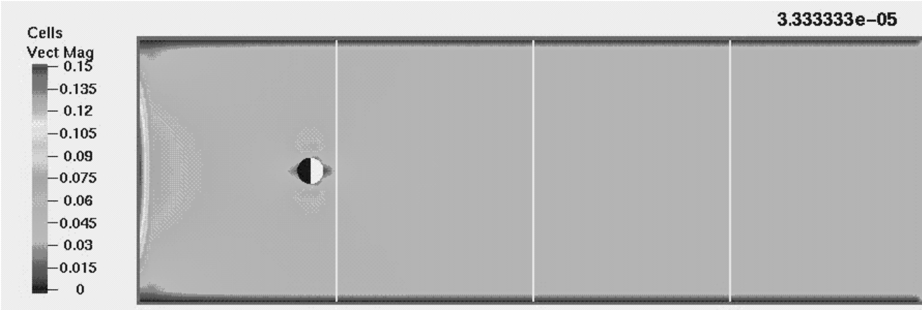




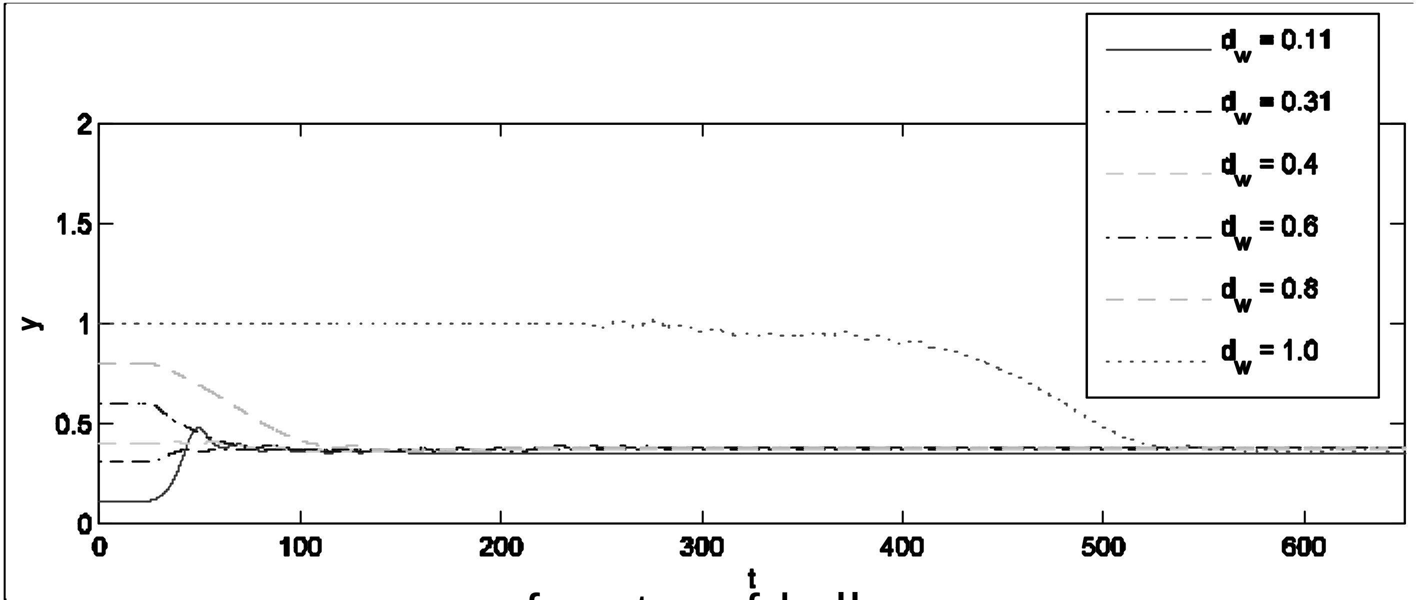
Lift-Off for Circle



Velocity (d_w = 0:1)

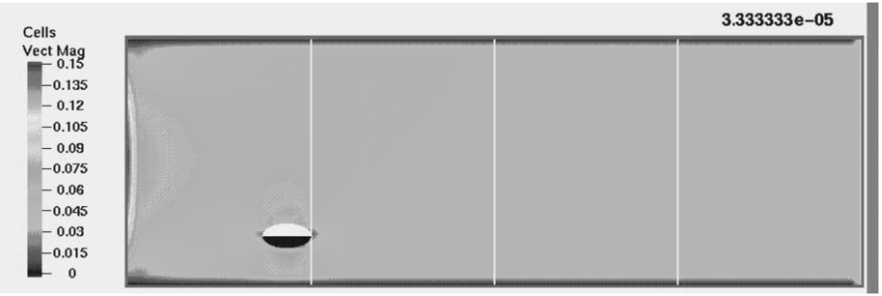


Velocity (d_w = 1:0)

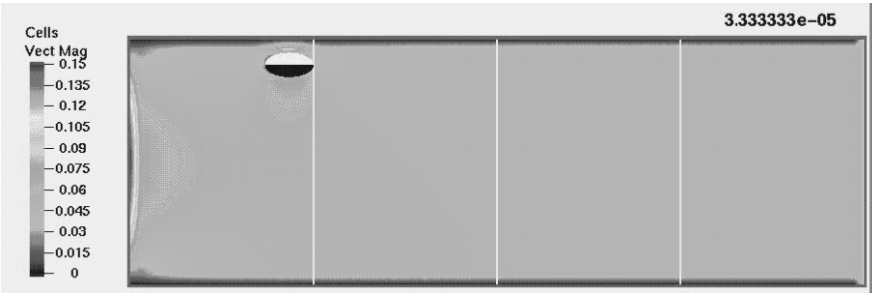


y of center of ball

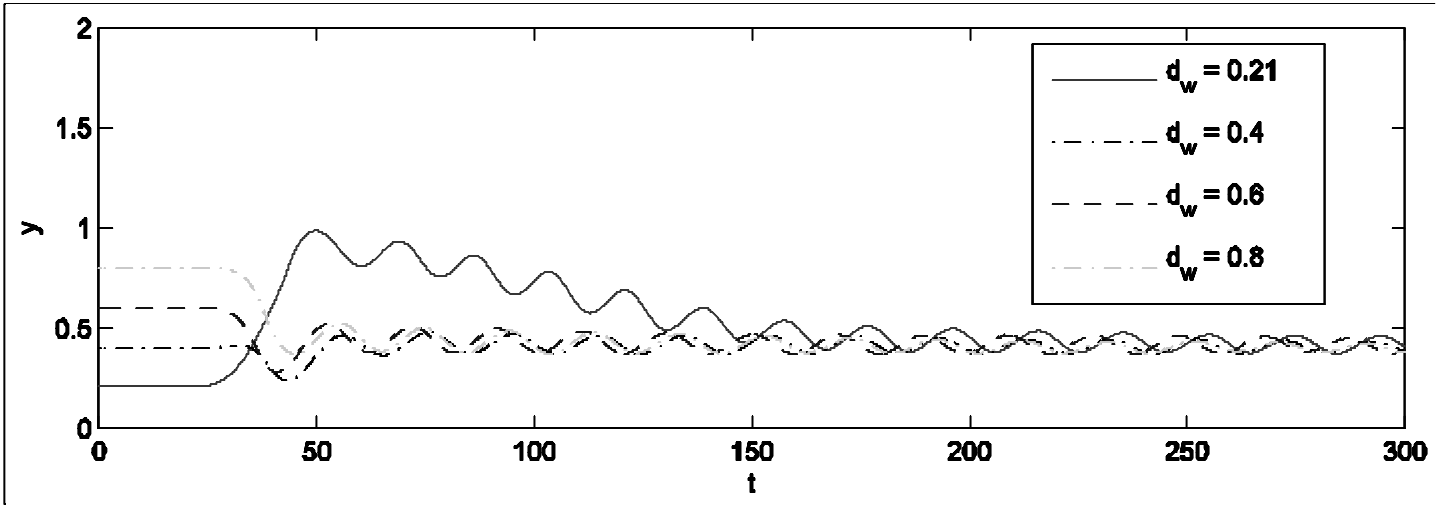
Lift-Off for Ellipse



Velocity (d_W = 0:4)



Velocity (d_W = 1:8)



y of center of ellipse

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 the 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 ¼ 220:000 elements ¼ 1:100:000 d:o:f :s

L4 ¼ 880:000 elements ¼ 4:400:000 d:o:f :s

L5 ¼ 3:530:000 elements ¼ 17:600:000 d:o:f :s

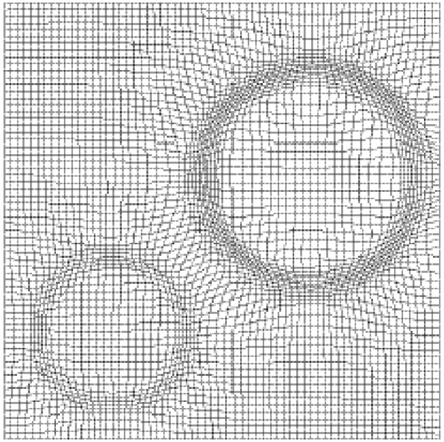
DEC/ COMPAQ EV6, 833 MHz

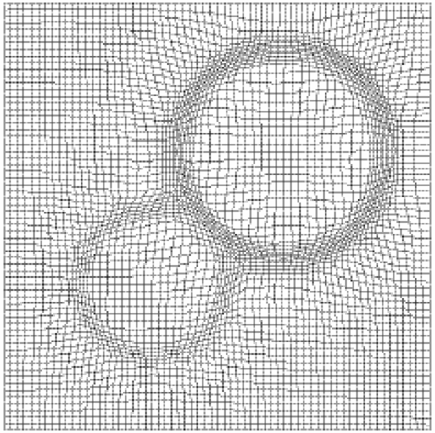
CPU (s)	'brute force'						'improved'					
#PART	= 10			= 1000			= 10			= 1000		
items	L=3	L=4	L=5	L=3	L=4	L=5	L=3	L=4	L=5	L=3	L=4	L=
NSE	17	88	440	16	80	**403**	17	95	423	17	83	**43**
Force	5	20	79	443	1771	**7092**	0	0	1	0	0	
Particle	1	5	25	20	82	**331**	0	3	14	1	5	**2**
Total	24	114	546	480	1934	**7827**	18	98	439	18	89	**46**

Next: E±cient °ow solver (for small ¢t) ???

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.

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 $\det \nabla \phi = f$

⟹ 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 ($t \in (0,1]$)

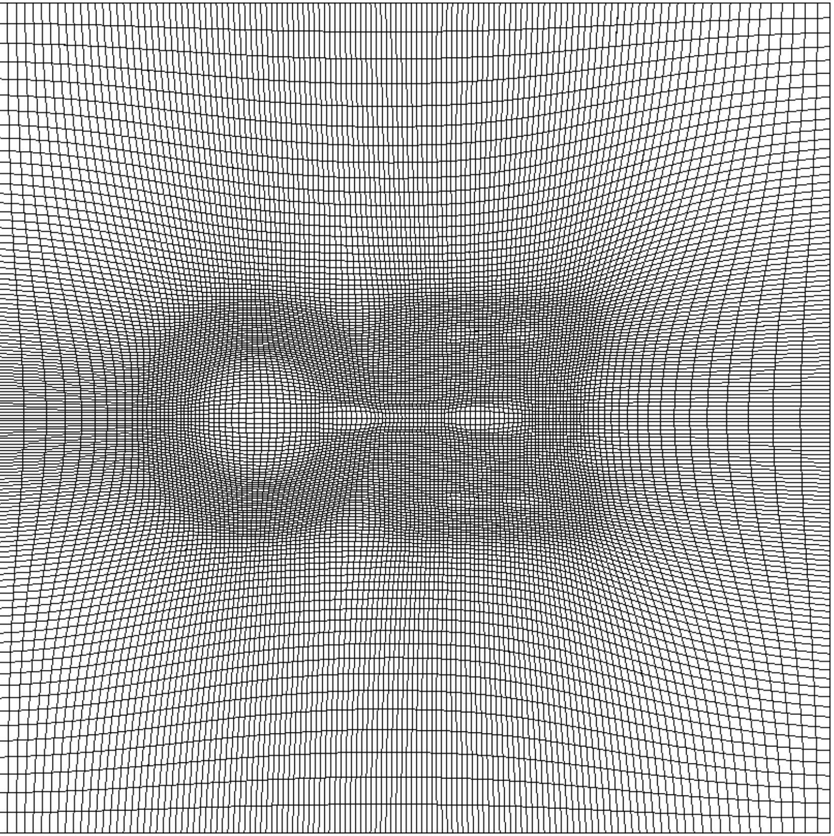
$$\Delta v(\xi, t) = -\frac{\partial}{\partial t}\left(\frac{1}{f(\xi,t)}\right), \quad \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0$$

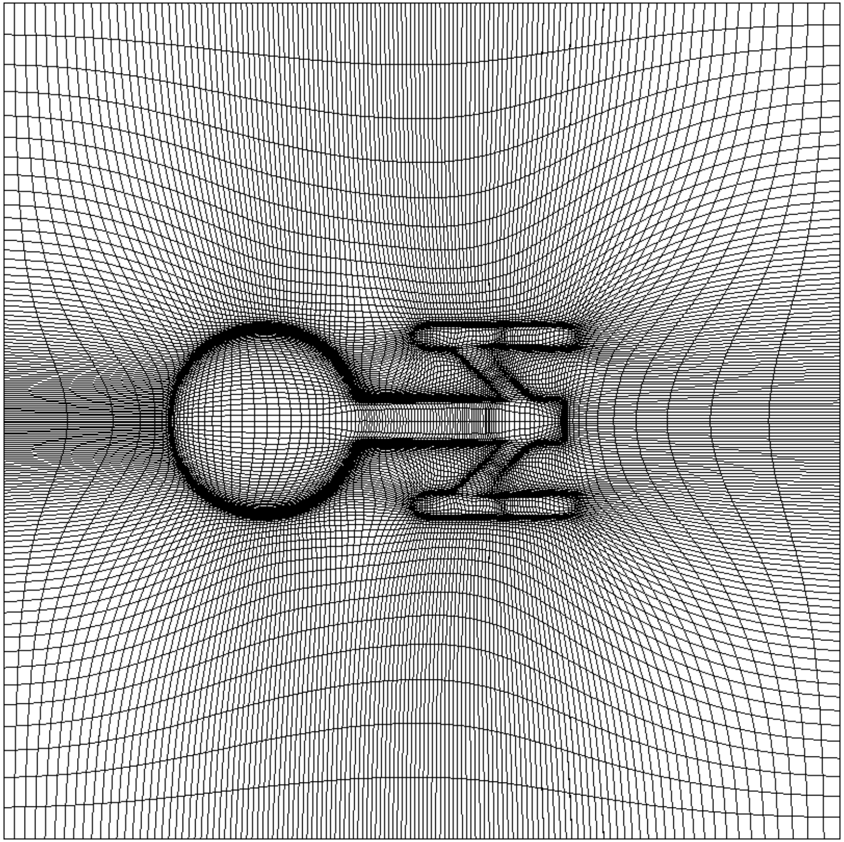
3. Solve the ODE system

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

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

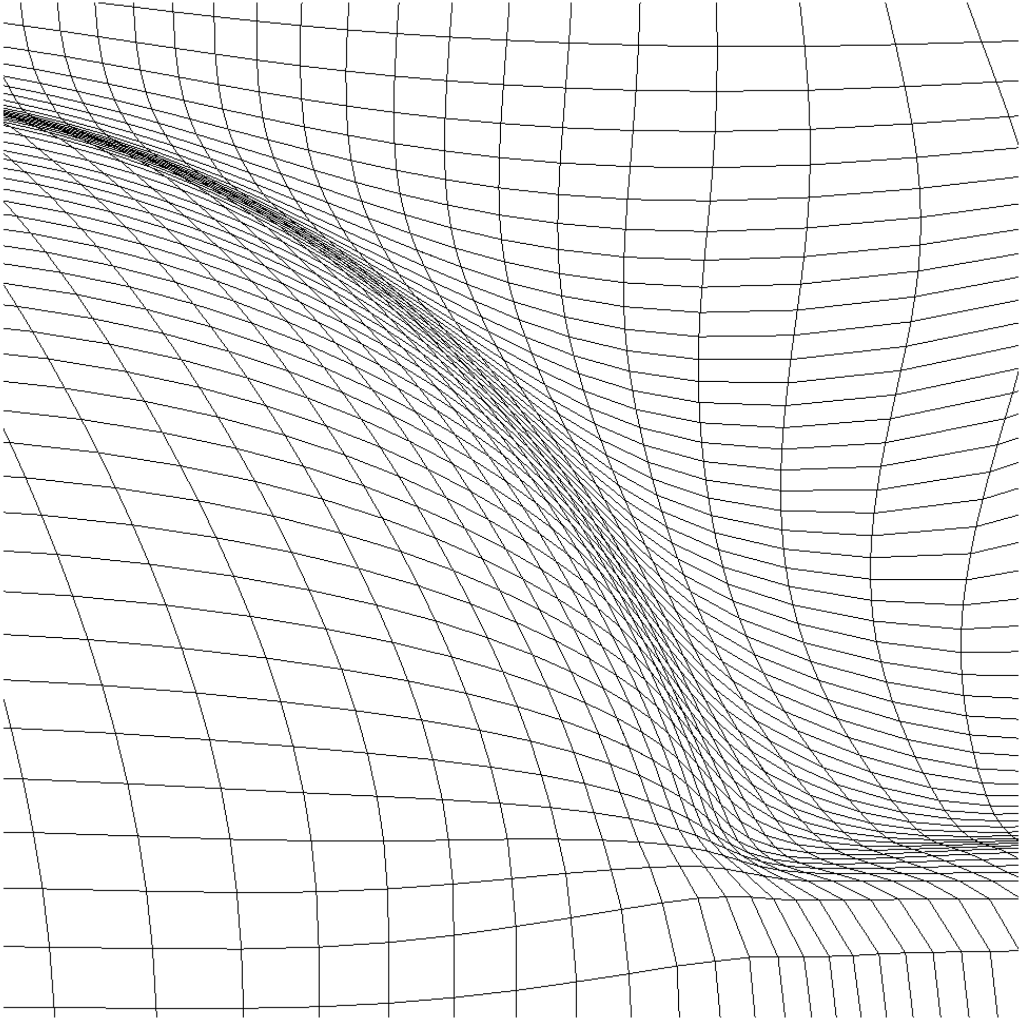
Example for Deformed Meshes





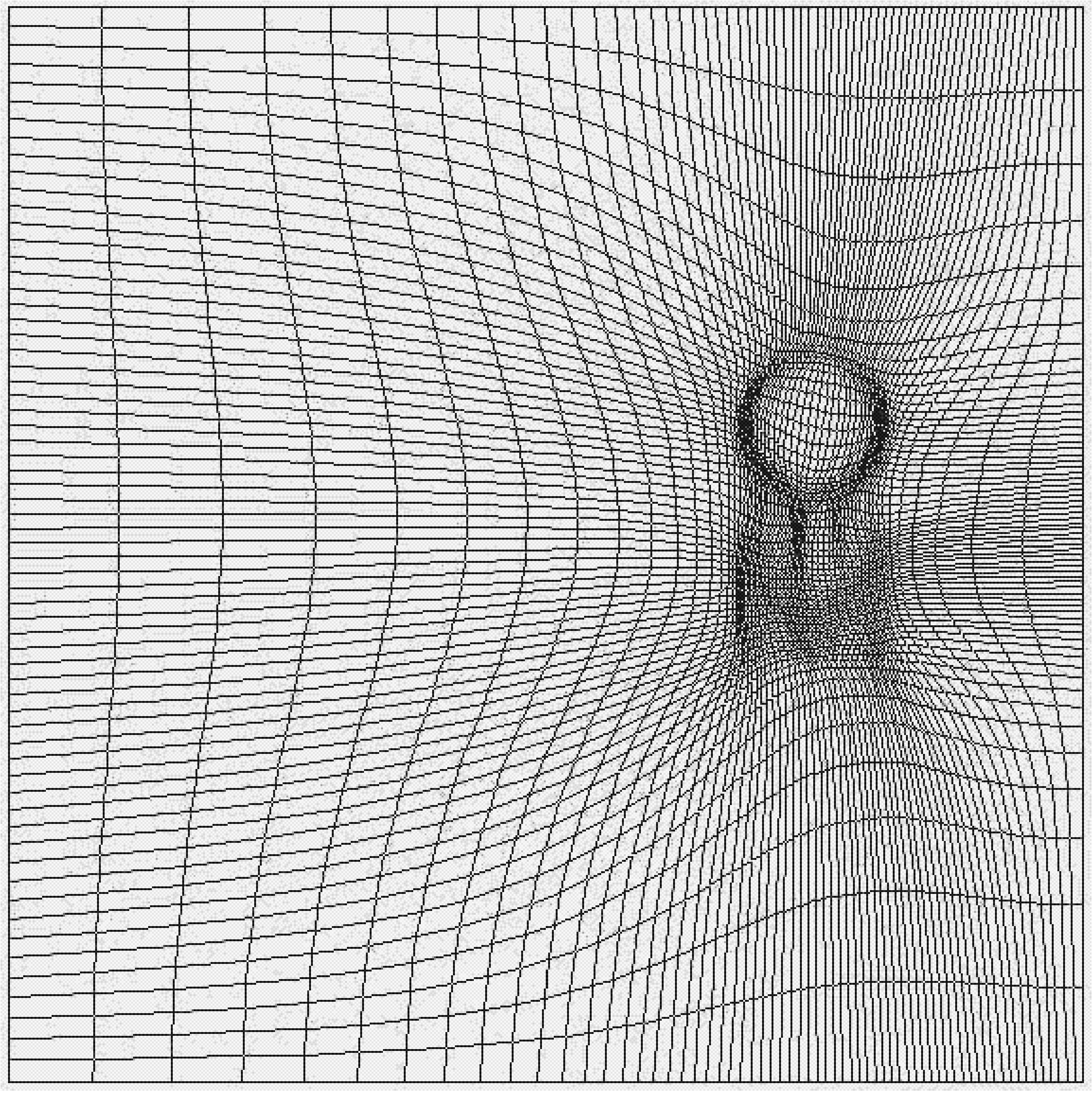
Grid deformation preserves the (local) logical structure of the grid

Example for Deformed Meshes



Exact control and smooth transitions

Last Example



(Really) Last Example