Modelling superquadric particles using Hybrid Fictitious Domain-Immersed Boundary method

Alexander Podlozhnyuk[1],
Federico Municchi[2],
Christoph Goniva[1,3],
Christoph Kloss[1,3]

Email:

[1,3]name.surname@dcs-computing.com

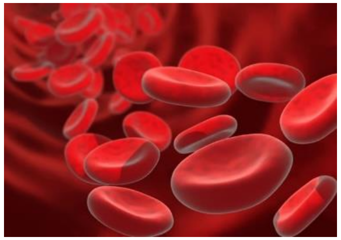
[2]fmunicchi@tugraz.at

DCS Computing GmbH
www.dcs-computing.com
Open source: www.cfdem.com
office@dcs-computing.com

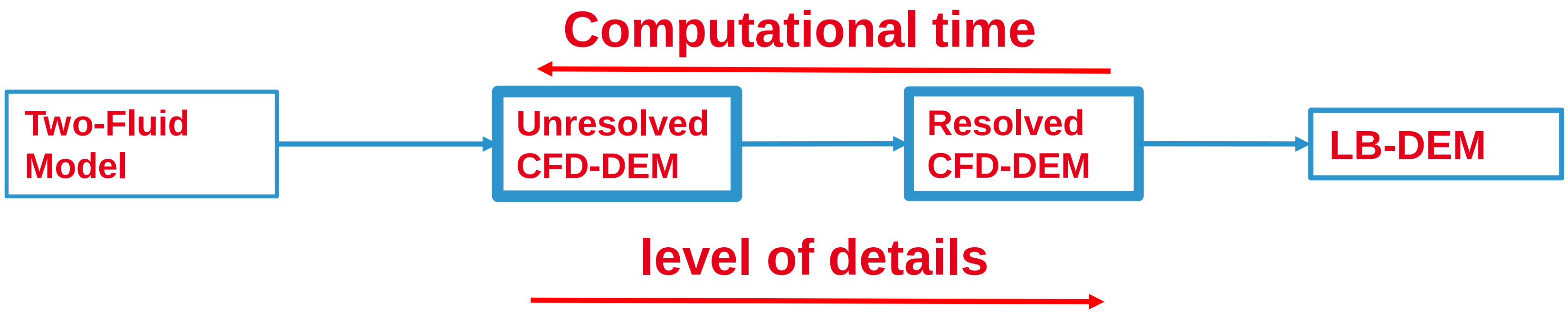
We love particles

Motivation

- Particles in reality are mostly non-spherical. Still many DEM ad CFD-DEM codes employ only perfect spheres



- Detailed information on solid and fluid velocities is difficult to achieve in experiments -> need to simulate
- Simulate at high level of details (resolved CFD-DEM) to apply the received knowledge at low level of details (unresolved CFD-DEM)



Objective

- Solve Navier-Stokes equations in complex domains:

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0 \\ \dfrac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p/\rho + \nu \Delta \boldsymbol{u} \\ +\text{boundary and initial conditions} \end{cases}$$

- Complex domains are particulate systems comprising spherical or **non-spherical** particles (much bigger than a CFD cell). Dirichlet boundary condition must be fulfilled:

$\boldsymbol{u}(\boldsymbol{x} \in \partial\Omega_i, t) = \boldsymbol{v}_i^{surf}$ treated by extra source term $\boldsymbol{f}_{\boldsymbol{IB}}$ in the momentum equation

- Particles can be static or moving:

$$m_i \frac{d^2 \boldsymbol{x}_i}{dt^2} = \boldsymbol{F}_{ext} + \boldsymbol{F}_{pp} + \boldsymbol{F}_{pw} + \boldsymbol{F}_{pf}$$

+ rotational motion

External force field (gravity, eletric field)

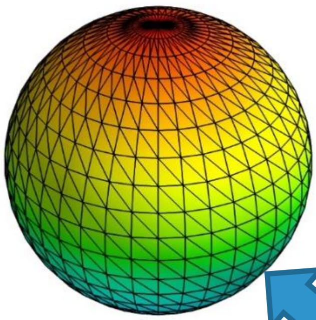
Particle-particle forces

Particle-wall forces

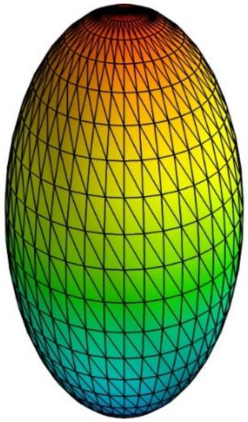
Particle-flow interaction

Superquadric shape

Superquadrics in Discrete Element Method:
Podlozhnyuk et al. „Efficient implementation of superquadric patrticles in DEM within an open-source framework“ (2016, submitted manuscript)



$a = b = c,$
$n_1 = n_2 = 2$



$a \neq b \neq c,$
$n_1 = n_2 = 2$

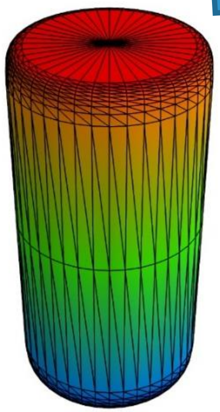
$$f(\boldsymbol{x}) \equiv \left(\left|\frac{x}{a}\right|^{n_2} + \left|\frac{y}{b}\right|^{n_2} \right)^{\frac{n_1}{n_2}} + \left|\frac{z}{c}\right|^{n_1} - 1 = 0$$

$$\boldsymbol{x} = (x, y, z)^T \; n_1 \geq 2, n_2 \geq 2^*$$

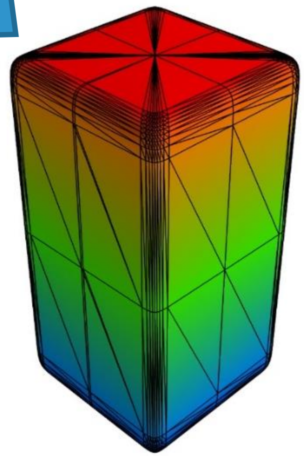
In local coordinate system!

$$F(\boldsymbol{X}) = f(\boldsymbol{A}^T \cdot (\boldsymbol{X} - \boldsymbol{X}_C))$$

In global (observer fixed) coordinate system!



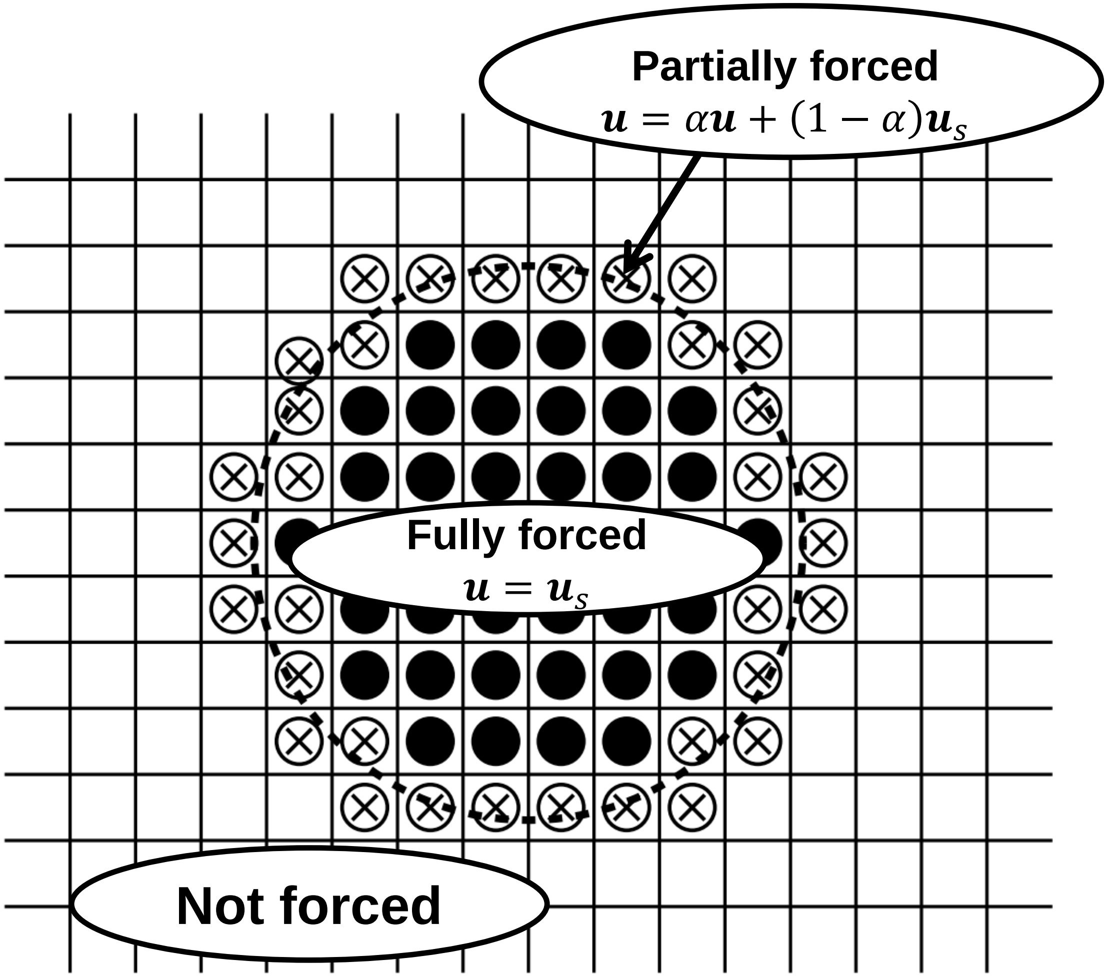
$n_1 \gg 2, n_2 = 2$



$n_1 \gg 2, n_2 \gg 2$

***A.H. Barr, Superquadrics and Angle-Preserving Transformations, IEEE Computer Graphics and Applications, vol.1, no. 1, pp. 11-23, January-March 1981**

Standard Immersed Boundary method* (cfdemSolverIB)



Sketch by F.Municchi, "A Hybrid Fictitious Domain-Immersed Boundary Method for the Direct Simulation of Heat and Mass Transport in Fluid-Particle Systems", **March 14 2016, LIGGGHTS and CFDEM coupling user meeting, Linz**

- Directly impose the fluid velocity inside a particle as on the sketch
- Make 1 step by PISO solver, find new pressure $\tilde{p}$ and velocity $\tilde{u}$.
- Correct new velocity $\tilde{u}$ such that it is divergence free:

$$u^{n+1} = \tilde{u} - \nabla\varphi$$

where $\Delta\varphi = \nabla \cdot \tilde{u}$

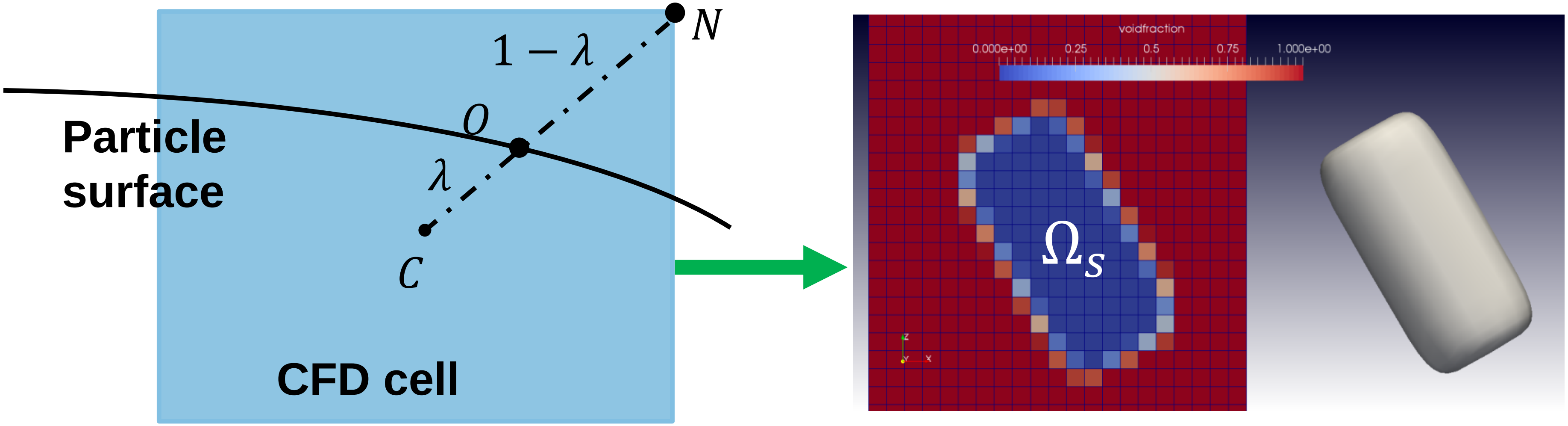
- Update pressure:

$$p^{n+1} = \tilde{p} + \rho\frac{\partial\varphi}{\partial t}$$

Hager A. , Kloss C. , Goniva C. , Towards an efficient immersed bounday method within an open-source framework, Proc. Oof the 8° conf. On CFD in Oil and Gas, Metallurgical and Process Industries, 2011

Force and void fraction calculation

- Requires calculation of the void fraction α

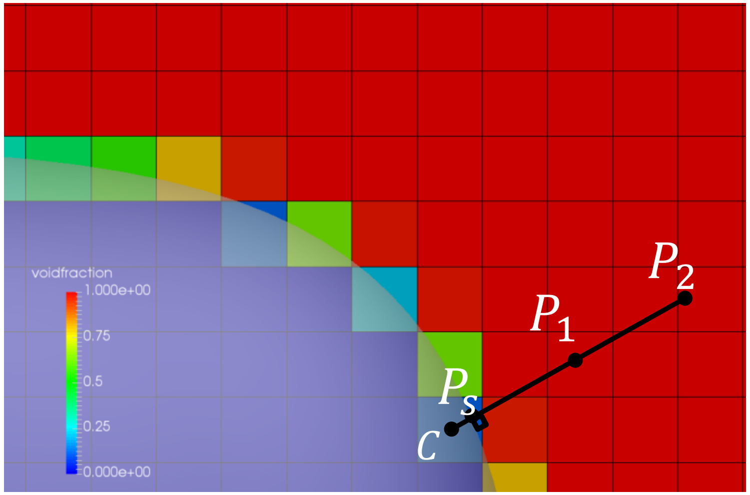


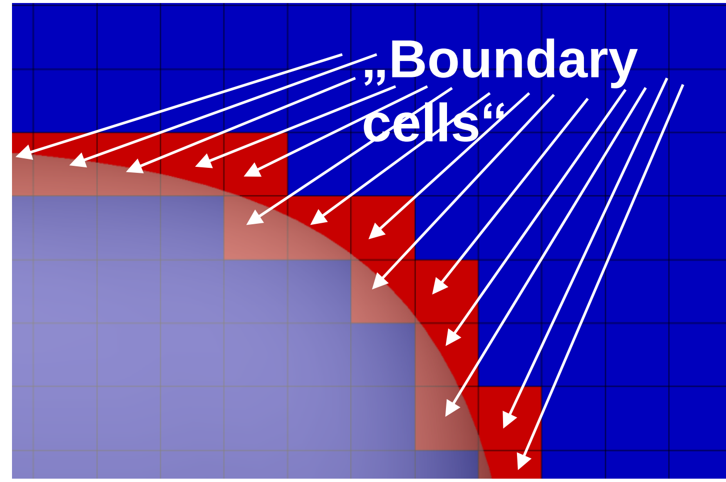
- Each intersection of center-node line segment with a particle surface gives $-\lambda/8$ to the void fraction field α.
- $\lambda = OC/CN$, where O is the intersection point. A few iterations required for superquadrics.
- Max error: 10%
- Calculate force acting on a particle:

$$\boldsymbol{F}^* = \int (-\nabla p + \mu \Delta \boldsymbol{u}) d\Omega_s \approx \sum (-\nabla p + \mu \Delta \boldsymbol{u})(1 - \alpha_i) V_i$$

A.A. Shirgaonkar, M.A. MacIver and N.A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, J. Comput. Phys., 228, (2009) 2366-2390

Hybrid Fictitious Domain-Immersed Boundary Method (HFDIBM)





- Dirichlet condition: $\boldsymbol{u}(P_S) = 0$, where P_S is the projection of the cell center onto the particle surface
- $\boldsymbol{u}(P_1)$ and $\boldsymbol{u}(P_2)$ are interpolated by OpenFOAM, „desired“ fluid velocity $\boldsymbol{u}_C^*$ at the cell center C is reconstructed by 2nd order interpolation using $\boldsymbol{u}(P_1)$, $\boldsymbol{u}(P_2)$, $\boldsymbol{u}(P_S)$.
- PISO-IB* forcing term (Newtonian acceleration):

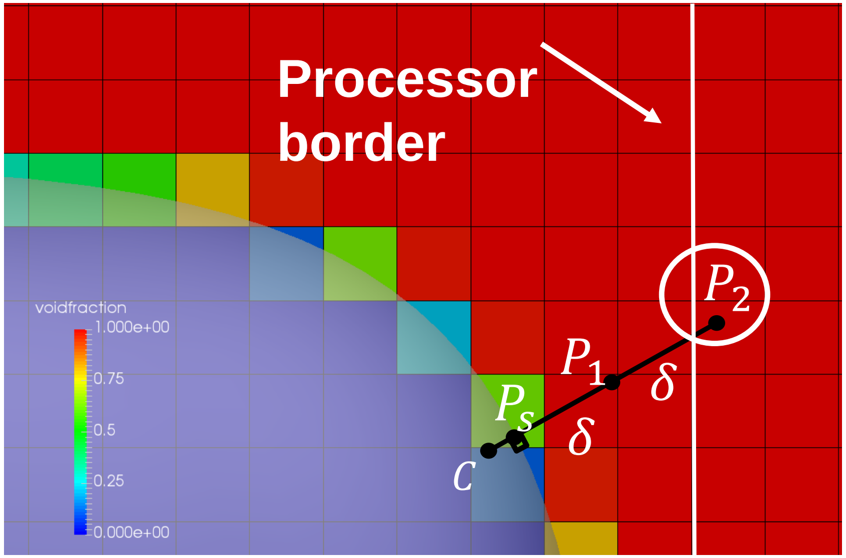
$$\boldsymbol{f}_{IB}^C = \frac{\boldsymbol{u}_C^* - \boldsymbol{u}_C}{\Delta t}$$

used explicitly in PISO-loops. $\boldsymbol{u}_C$ is the current fluid velocity at C. No additional velocity-correction loops are required.

- However, $\boldsymbol{f}_{IB}^C$ *is applied only to the „boundary cells“*

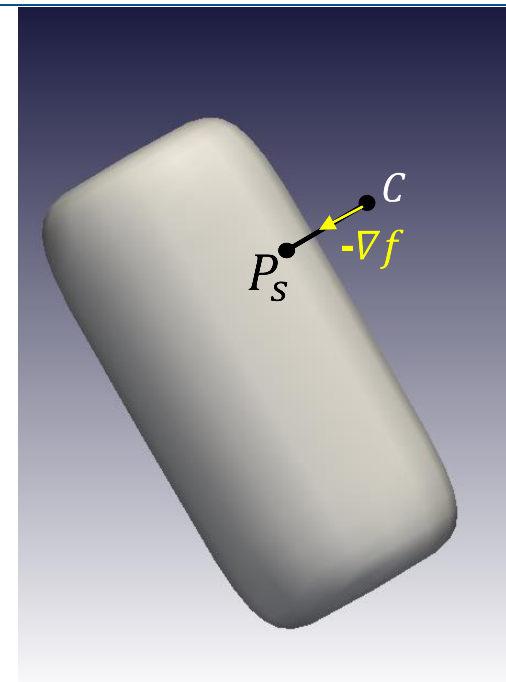
Blais B. , Lassaigne M. , Goniva C. , A semi-implicit immersed boundary method and its applications to viscous mixing, Computers and Chemical Engineering, 2016

HFDIBM – implementation details



Fully parallel:

- Interpolation points can be situated on different processors. Problem solved
- Particle can be divided by more than 2 processors on CFD side. Problem solved.



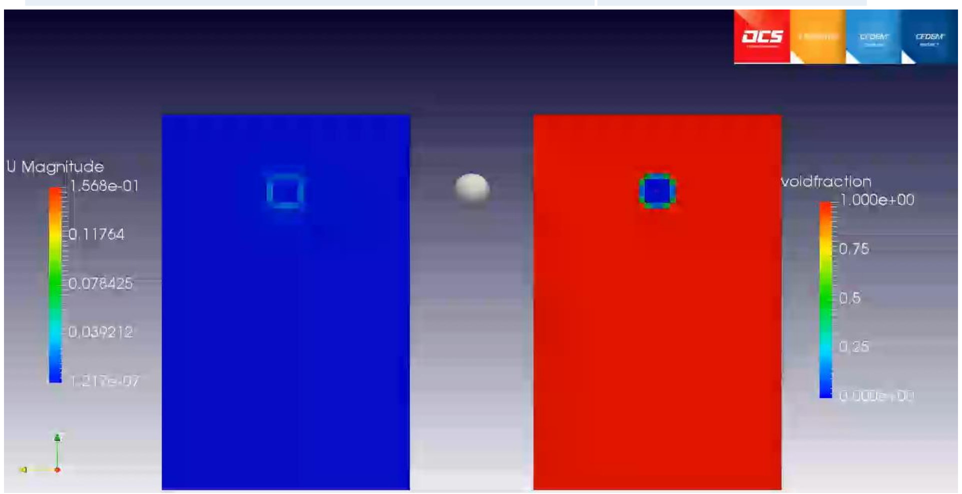
- Distance δ between interpolation points $\sim 1-2$ of mesh element size.
- Fluid-particle force:

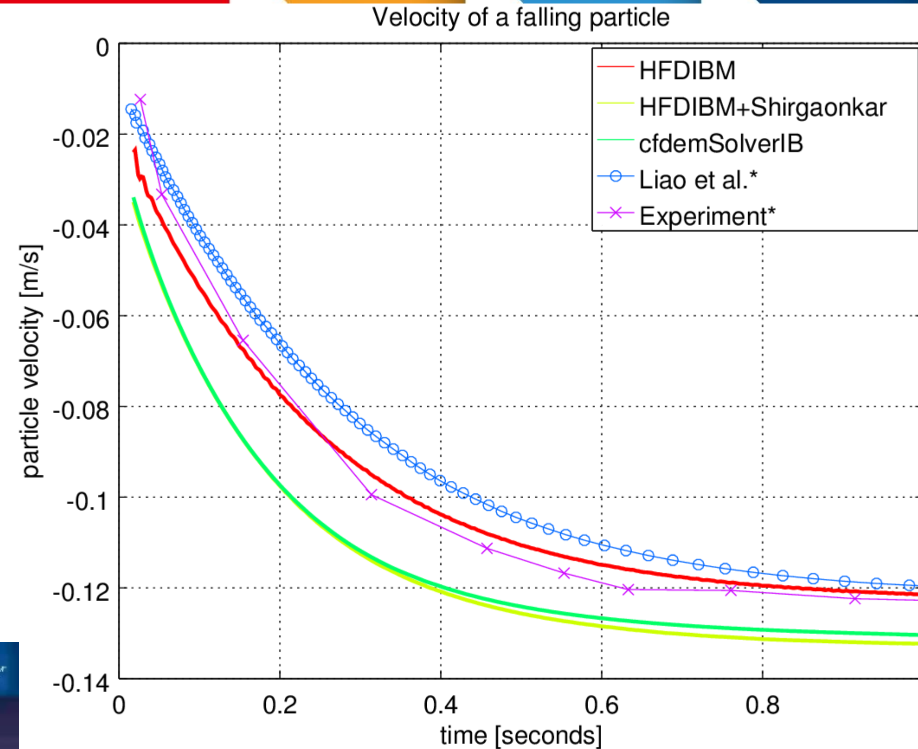
$$\boldsymbol{F} = \rho_{fluid} \sum_{i \in BC} \boldsymbol{f}_{IB}^{i} \, V_i$$

- To find P_S we assume that the gradient of the superquadric shape at C function is almost is almost orthogonal to the surface. Thus, just find the surface-line intersection point

Sedimentation of a single sphere

Parameter	Value
Fluid density, kg/m^3	960
Fluid viscosity, $10^{-3} N \cdot s/m^2$	58
Max Re	32
Sphere diameter, m	0.015
Domain size, $m \times m \times m$	$0.1 \times 0.1 \times 0.16$
Mesh size	$40 \times 40 \times 64$
Mesh refinement level	2
Cells per diameter	24
Domain decomposition, $N_x \times N_y \times N_z$	$2 \times 2 \times 1$
CFL number	0.2

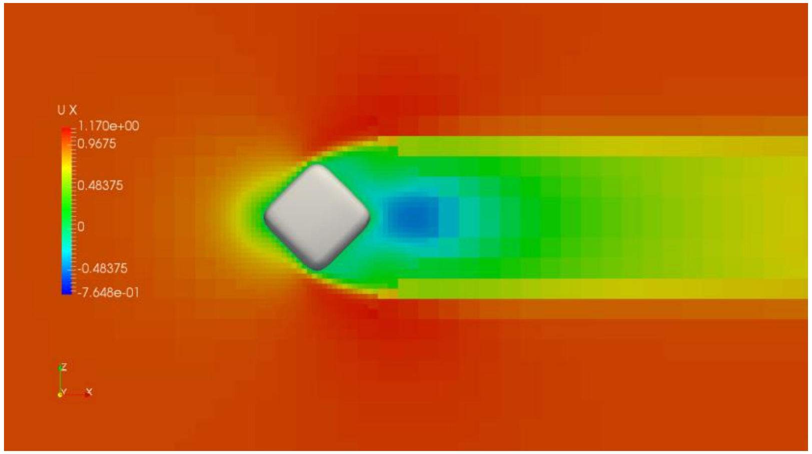


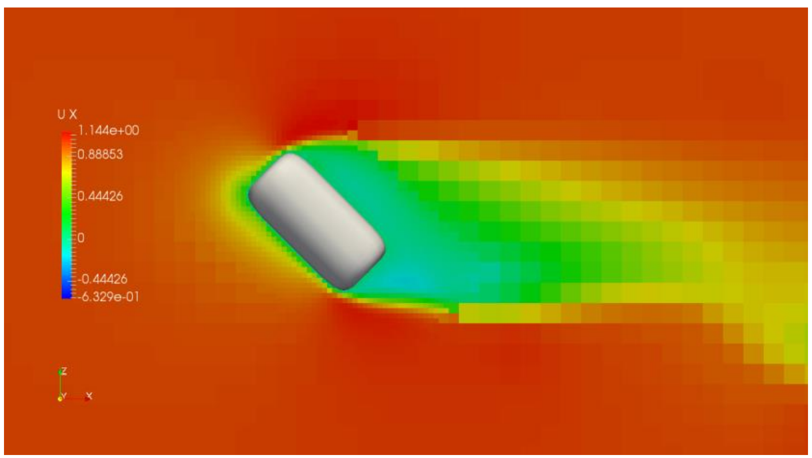


HFDIBM gives time-velocity curve in very good agreement with the experiment!

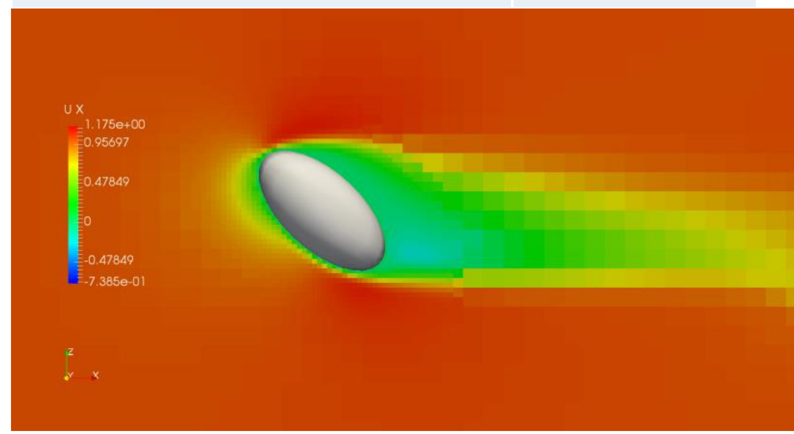
*Liao et al. , Simulating flows with moving rigid boundary using immersed-boundary method, 2010

Flow past a non-spherical particle





Parameter	Value
Fluid density, kg/m^3	1
Inlet velocity, m/s	1
Diameter of the volume equivalent sphere, m	1
Domain size, $m \times m \times m$	$12 \times 6 \times 6$
Mesh size	$60 \times 30 \times 30$
Mesh refinement level	2
Cells per diameter	20
Domain decomposition, $N_x \times N_y \times N_z$	$1 \times 2 \times 2$

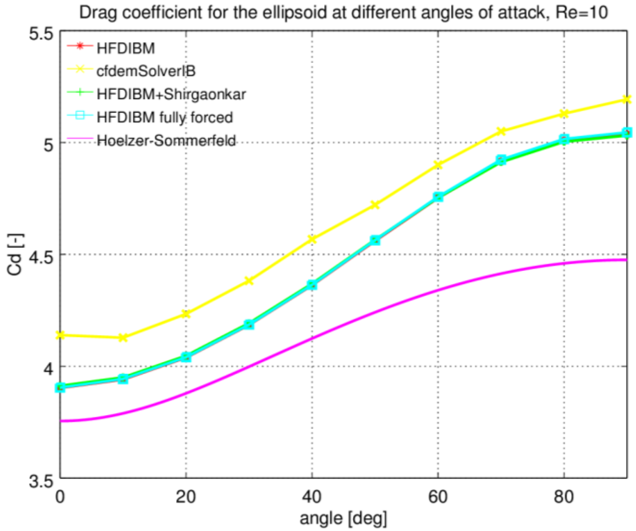


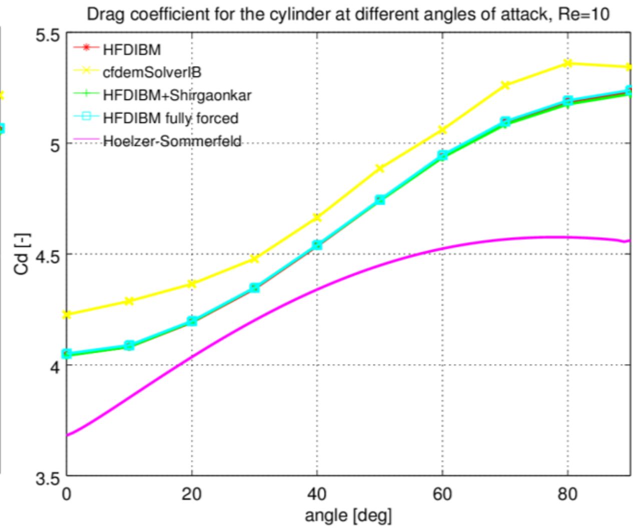
Compare with Hölzer-Sommerfeld* drag:

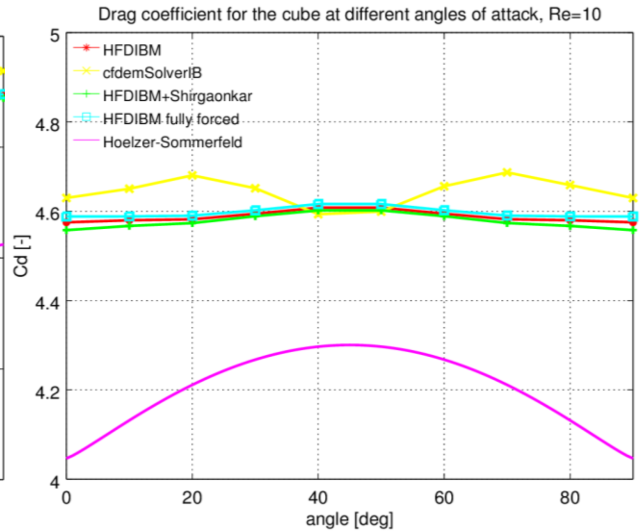
$$C_D = \frac{8}{\mathrm{Re_p}}\frac{1}{\sqrt{\Phi_\perp}} + \frac{16}{\mathrm{Re_p}}\frac{1}{\sqrt{\Phi}} + \frac{3}{\sqrt{\mathrm{Re_p}}}\frac{1}{\Phi^{3/4}} + 0.42 \cdot 10^{0.4(-\,log_{10}\,\Phi)}\frac{1}{\Phi_\perp}$$

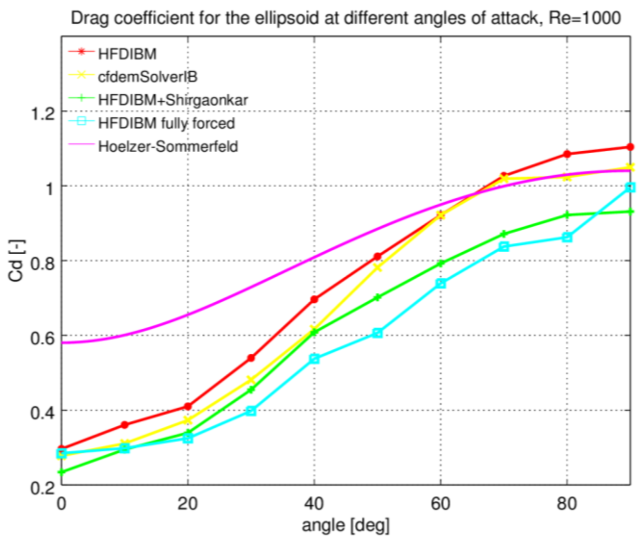
Hölzer and Sommerfeld,** ***New simple correlation formula for the drag coefficient of non-spherical particles, 2008

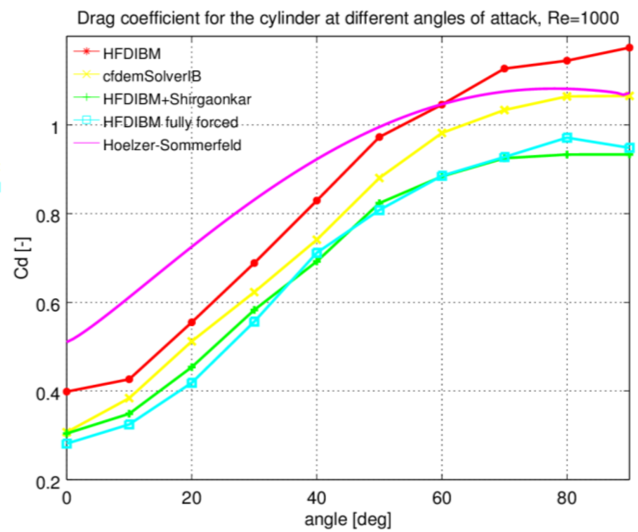
Comparison of drag coefficients

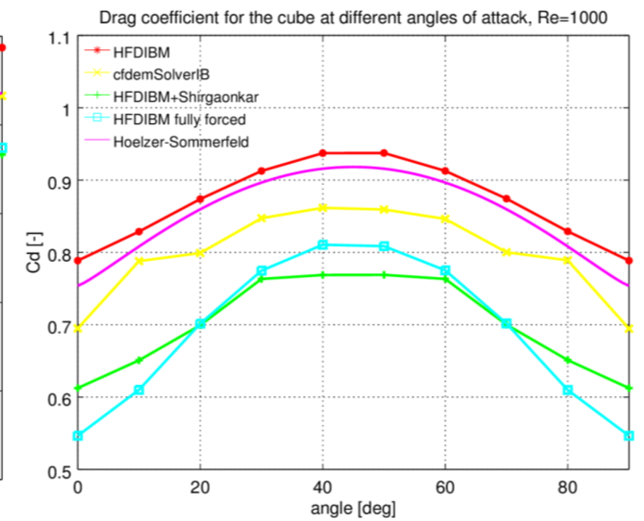




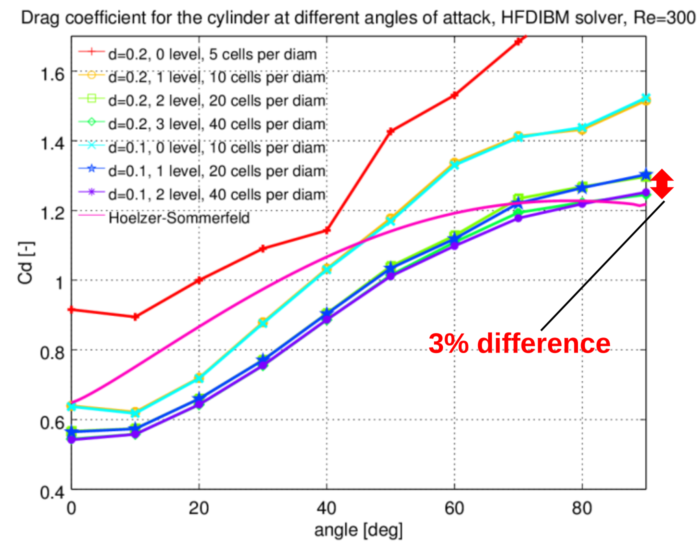


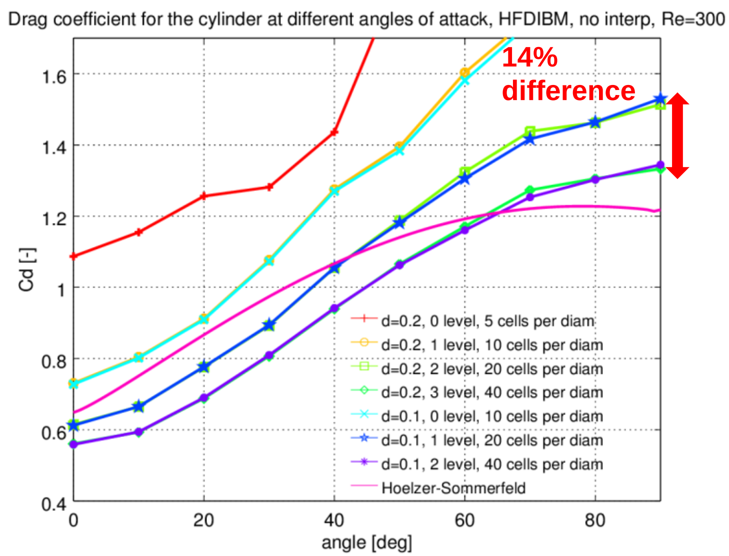






Mesh convergence





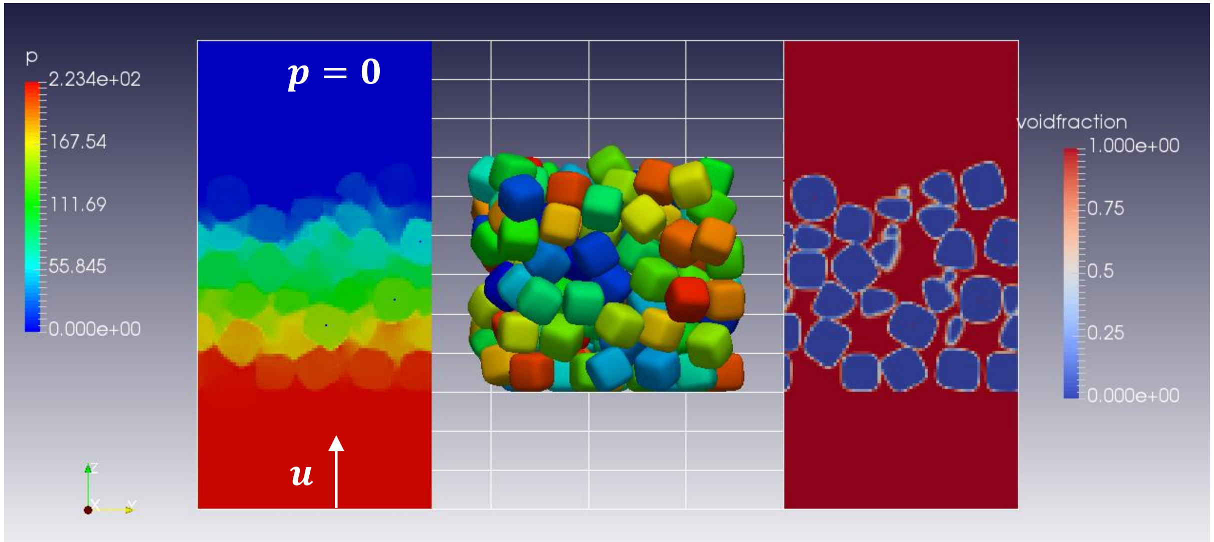
A cylinder at differen angles of attack

$Re = 300$

Mesh size (in blockMeshDict), d, m	Refinement level	Number of mesh elements	Cells per diameter
0.2	0	54000	5
0.2	1	~60500	10
0.2	2	~102000	20
0.2	3	~380000	40
0.1	0	432000	10
0.1	1	~473000	20
0.1	2	~752000	40

Optimal mesh size!
Finer meshes produce results with 3% difference requiring at least 4 times more computational time

Flow through a packe bed. Pressure drop



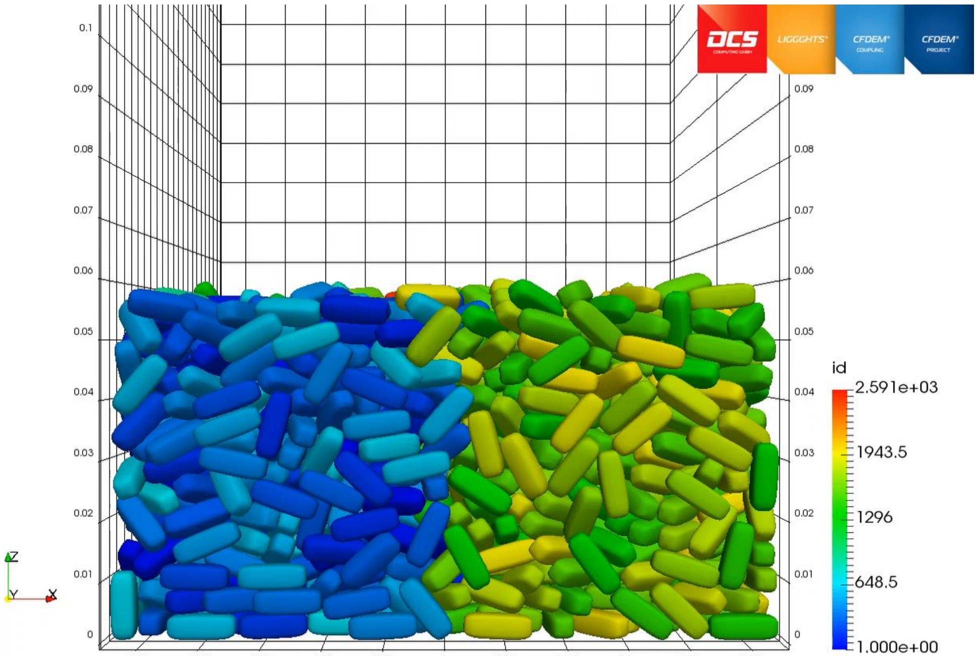
Parameter	Value
Number of particles	150
Domain size, $mm \times mm \times mm$	$12 \times 12 \times 24$
Particle size, $mm \times mm \times mm$	$1 \times 1 \times 1$
Mesh size, elements	$80 \times 80 \times 160$
Inlet velocity u, m/s	1
Viscosity μ, $Pa \cdot s$	0.0001
Bed height, mm	11 ± 0.5

Ergun equation:

$$\frac{\Delta p}{L} = \frac{150\mu(1-\varepsilon)^2}{d^2\varepsilon^3}u + \frac{1.75\rho(1-\varepsilon)}{d\varepsilon^3}u^2$$

- Ergun eqution gives $\Delta p = 273 \pm 80 Pa$
- HFDIBM gives $\Delta p = 221 Pa$
- cfdemSolver gives oscilating Δp, far from being within uncertainty interval given by Ergun equation
- HFDIBM with full body forcing behaves non-stable

Why do we need this

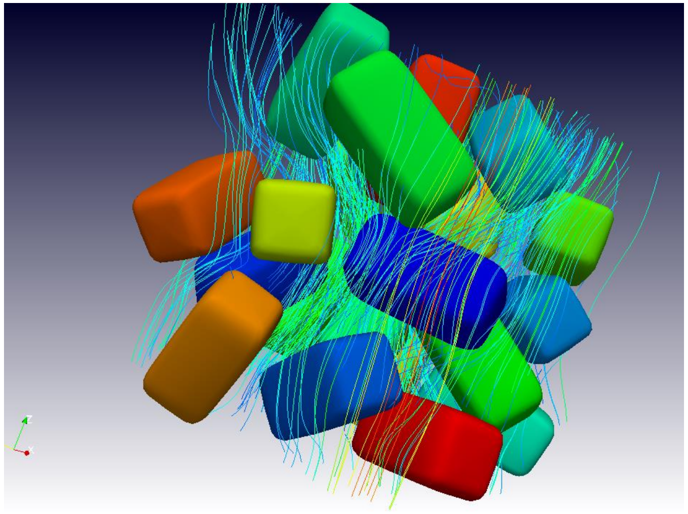


Example of unresolved CFD-DEM simulation: Fluidization

- We want to model flows through dens systems (e.g. fluidized beds) using unresolved CFD-DEM.
- Effect of tortuosity?

$$T = \frac{average\ streamline\ path}{Domain\ size}$$

$$T = \frac{< u >}{< \boldsymbol{u} \cdot \boldsymbol{n} >}$$



Goal: to „model“ a single CFD cell in **un**resolved CFD-DEM by means of resolved CFD-DEM

Find a new drag correlation for dense systems

Thank you for attention!