Simulation of Granular Two-phase flow

Contributors to this work:

Hans Kuipers

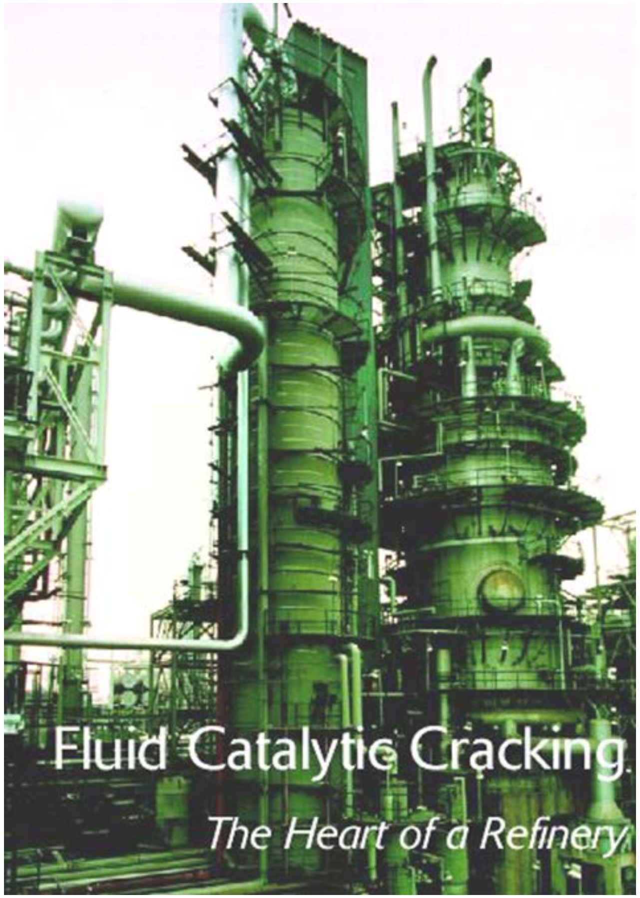
Renske Beetstra

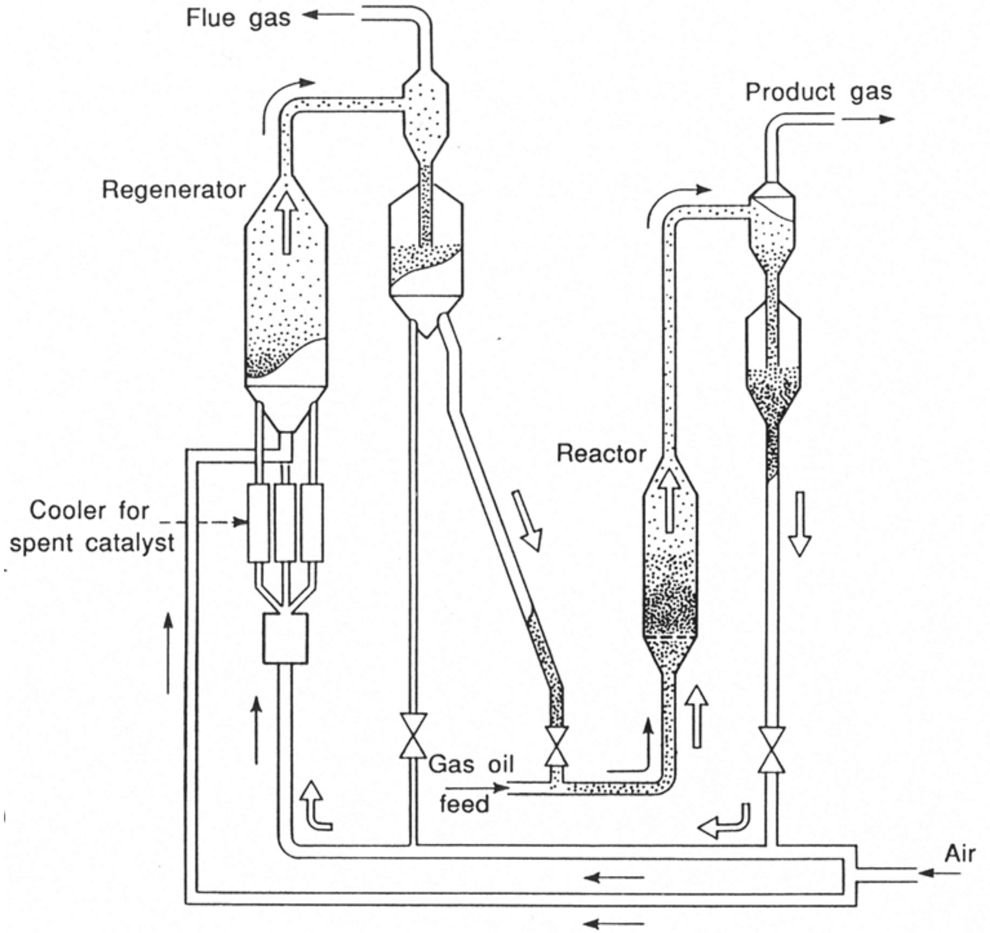
Christiaan Zeilstra

Sebastian Kriebitzsch

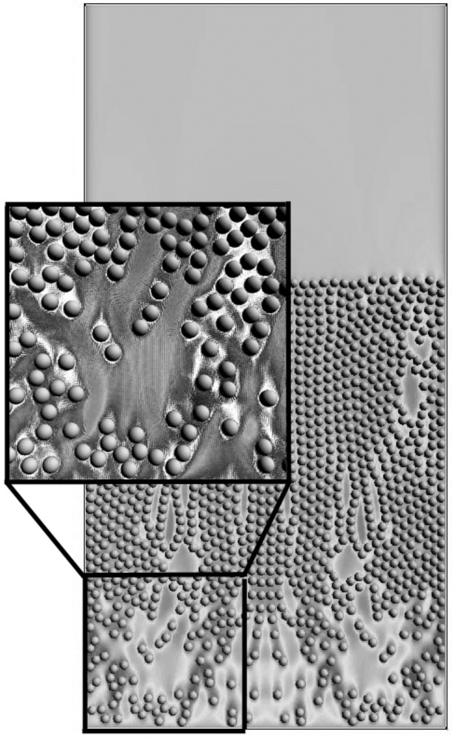
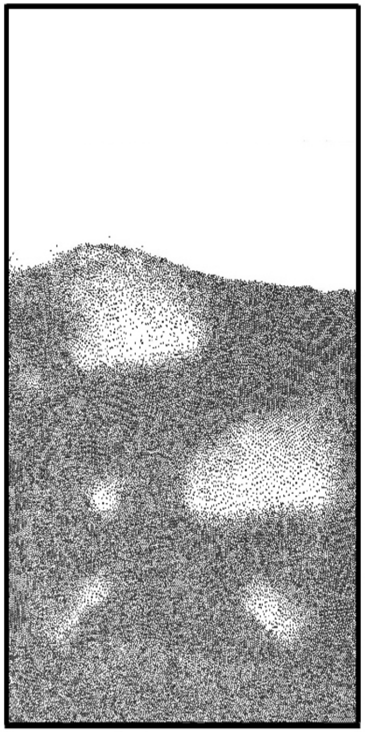
Simulation of Granular Two-phase flow

Important example: **gas fluidized beds**





Three basic models for two-phase granular flow:

DNS	DEM	TFM
		
Euler Lagrange	Euler Lagrange	Euler Euler

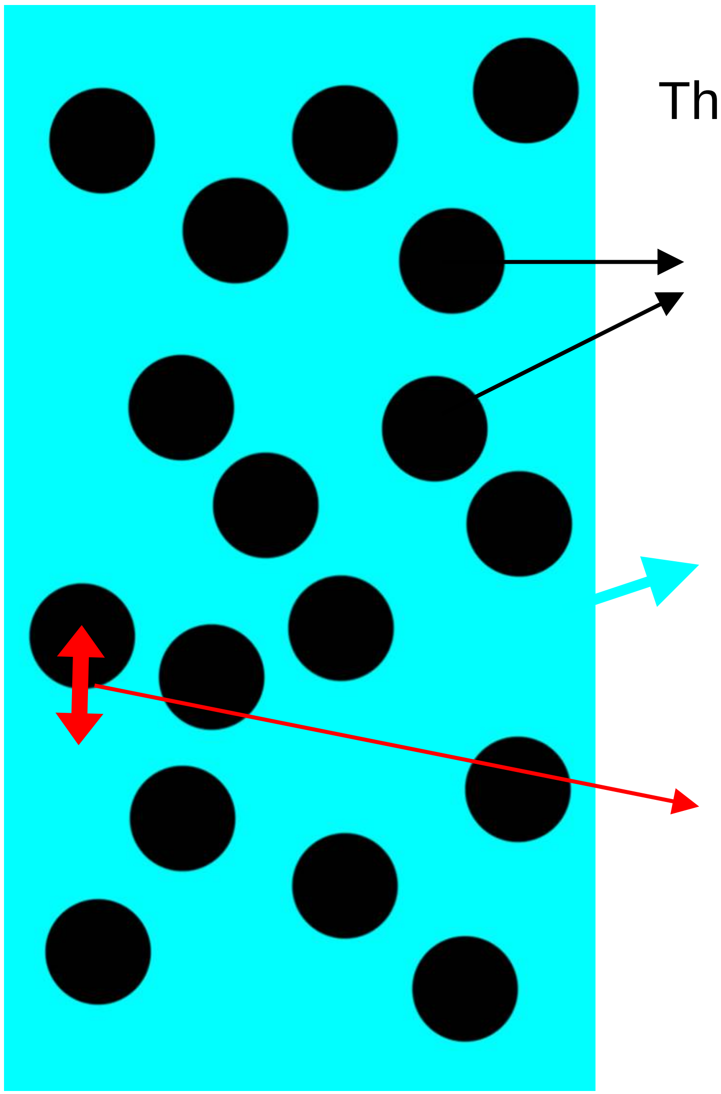
Outline

I. DNS models for granular two-phase flow

II. DEM models for granular two-phase flow

III. Example: Vibrated granular beds

I. DNS models for granular two-phase flow



Three issues:

How model the particles?

How to model the fluid phase?

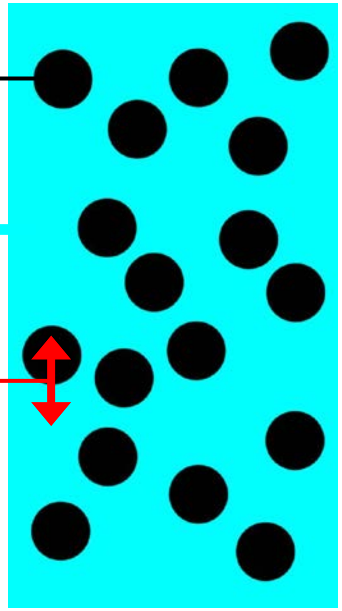
How to model the fluid-particle interaction?

I. DNS models for granular two-phase flow

1. Models for the solid phase

2. Models for the fluid phase (CFD, LBM)

3. Fully resolved fluid-particle interaction



1. Models for the solid phase

- Phase consists of individual particles → Lagrangian

- Methods borrowed from classical "molecular dynamics"

- Two different methods
  A. Soft-sphere model
  B. Hard-sphere model

1A. Solid Phase Models: Soft-Sphere

Position of particle a: $\mathbf{R}_a$

Newton's equation of motion: $$M \frac{d^2\mathbf{R}_a}{dt^2} = \mathbf{F}_{a,tot}$$

is integrated numerically:

$$\mathbf{R}_a(t+\delta t) = 2\mathbf{R}_a(t) - \mathbf{R}_a(t-\delta t) + \frac{\mathbf{F}_{a,tot}(t)}{M}\,\delta t^2$$

total force: $$\mathbf{F}_{a,tot} = \sum_b \mathbf{F}_{ab} + M\mathbf{g}$$

→ Time driven scheme

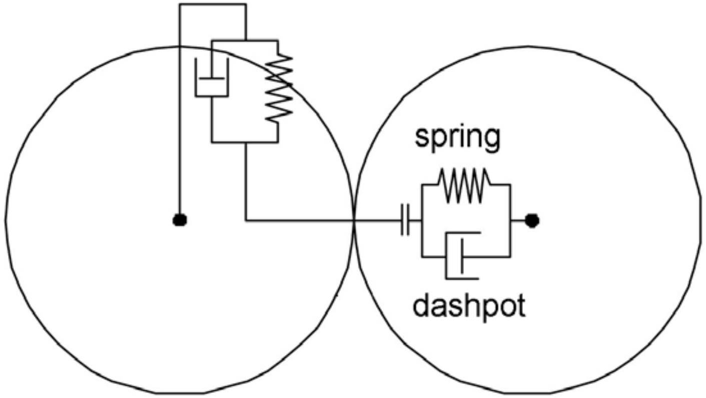
→ Interaction force $\mathbf{F}_{ab}$ follows from a continuous potential

→ "soft-sphere model"

1A. Solid Phase Models: Soft Sphere

Interaction force: $$\mathbf{F}_{ab} = \mathbf{F}_{ab}^{coll} + \mathbf{F}_{ab}^{el} + \mathbf{F}_{ab}^{coh}$$

- Collision force: Spring-dashpot model



$$\mathbf{F}_{ab,n}^{coll} = -k\,\delta\,\mathbf{n}_{ab} - \eta\,\mathbf{v}_{ab,n}$$

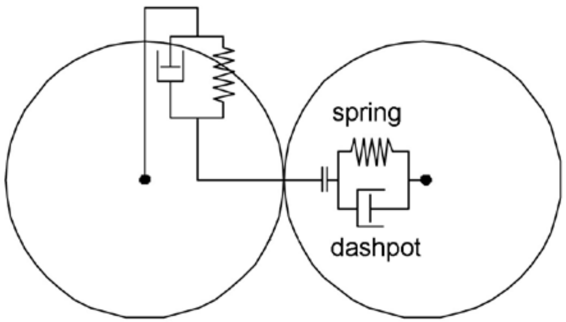
spring constant

damping coefficient

- Electrostatic force $$\mathbf{F}_{ab}^{el} = -\frac{q^2}{4\pi\epsilon}\frac{\mathbf{n}_{ab}}{R_{ab}^2}$$

- Cohesive force $$\mathbf{F}_{ab}^{coh} = \frac{Ad}{6}\frac{\mathbf{n}_{ab}}{S_{ab}^2}$$

1A. Solid Phase Models: Soft-Sphere



$\mathbf{R}_a(t)\ ,\mathbf{V}_a(t)$

soft-sphere model

Coulomb force etc.

$\mathbf{F}_{ab}^{\text{coll.}}$

$$M\ddot{\mathbf{R}}_a = \sum_b \mathbf{F}_{ab}^{\text{coll.}}$$

$\mathbf{R}_a(t+\delta t)$
$\mathbf{V}_a(t+\delta t)$

1B. Solid Phase Models: Hard-sphere

Simplified MD: hard-sphere model

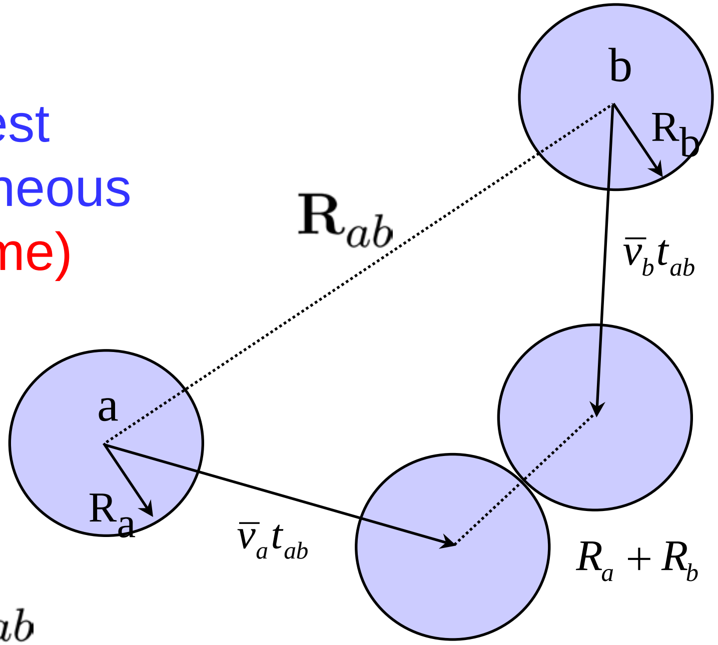
- Collision time between spheres can be calculated analytically:

$$t_{ab} = \frac{-\mathbf{R}_{ab} \cdot \mathbf{V}_{ab} - \sqrt{(\mathbf{R}_{ab} \cdot \mathbf{V}_{ab})^2 - V_{ab}^2 \left[R_{ab}^2 - 4R^2\right]}}{V_{ab}^2}$$

- Evolution in time: free-flight to nearest collision event followed by instantaneous binary collision (event driven scheme)

- Collision: change of momentum does not follow from forces, but is calculated via:

$$\Delta \vec{v}_a = \left(\frac{1+e}{2}\right) \frac{\mathbf{R}_{ab} \cdot \mathbf{V}_{ab}}{4\mathrm{R}^2} \mathbf{R}_{ab}$$



1B. Solid Phase Models: Hard-Sphere

Advantages of hard-sphere over soft-sphere

- Much faster for dilute systems
- Soft potential often “too soft” to model e.g. glass spheres

Disadvantages of hard-sphere over soft-sphere

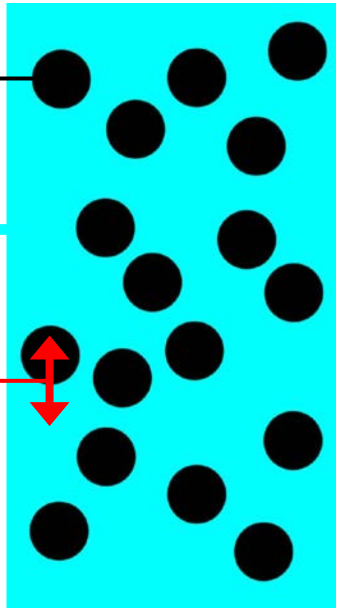
- HS breaks down for dense (close packed) systems
- Update not based on forces: more difficult to include other interactions

Outline

1. Models for the solid phase

2. Models for the fluid phase (CFD, LBM)

3. Fully resolved fluid-particle interaction



2. Models for the fluid phase (CFD, LBM)

- Continuum description of the phase → Eulerian

- Time evolution governed by Navier-Stokes (NS) equation

- Two basic methods for solving the NS equations on a grid

  A. Computational Fluid Dynamics

  B. Lattice Boltzmann Method

2. Fluid Phase Models: Computational Fluid Dynamics (CFD)

Basic idea: solve the set of differential equations:

$$\nabla \cdot \mathbf{u} = 0$$

$$\partial_t \mathbf{u} = -\frac{1}{\rho}\nabla P - \nabla \cdot \mathbf{u}\mathbf{u} - \nabla \cdot \tau$$

by finite difference methods

Closures for P and τ

$$P = \frac{RT}{M}\rho$$

$$\tau = -(\lambda - \frac{2}{3}\mu)(\nabla \cdot \mathbf{u})I + \mu((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T)$$

2. Fluid Phase Models: Computational Fluid Dynamics (CFD)

$$\nabla \cdot \mathbf{u} = 0$$

$$\partial_t \mathbf{u} = -\frac{1}{\rho}\nabla P - \nabla \cdot \mathbf{uu} - \nabla \cdot \tau$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} = -\frac{1}{\rho}\nabla P^{n+1} - \underbrace{\left[\nabla \cdot \mathbf{uu} + \nabla \cdot \tau\right]^n}$$

finite diff. form

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\delta t}{\rho}\nabla P^{n+1} - \delta t\, \mathbf{A}^n$$

$$\boxed{\hat{\mathbf{u}}^{n+1} = \mathbf{u}^n - \frac{\delta t}{\rho}\nabla \widehat{P}^{n+1} - \delta t\, \mathbf{A}^n}$$

tentative velocity

$$-\mathbf{u}^{n+1} + \hat{\mathbf{u}}^{n+1} = \frac{\delta t}{\rho}\nabla \left[P^{n+1} - \widehat{P}^{n+1}\right]$$

$$\Phi = P^{n+1} - \widehat{P}^{n+1}$$

taking $\nabla \cdot$

$$\underbrace{-\nabla \cdot \mathbf{u}^{n+1}}_{0} + \boxed{\nabla \cdot \hat{\mathbf{u}}^{n+1} = \frac{\delta t}{\rho}\nabla^2 \Phi}$$

$$\boxed{\begin{aligned} P^{n+1} &= \widehat{P}^{n+1} + \Phi \\ \mathbf{u}^{n+1} &= \hat{\mathbf{u}}^{n+1} + \frac{\delta t}{\rho}\nabla \Phi \end{aligned}}$$

Solution procedure to calculate variables at time $n+1$

Initial guess:

$$\boxed{\widehat{P}^{n+1} \Rightarrow \nabla \widehat{P}^{n+1}}$$

$$\boxed{\hat{\mathbf{u}}^{n+1} = \mathbf{u}^n - \frac{\delta t}{\rho} \nabla \widehat{P}^{n+1} - \delta t\, \mathbf{A}^n}$$

$$\boxed{\nabla \cdot \hat{\mathbf{u}}^{n+1} = \frac{\delta t}{\rho} \nabla^2 \Phi} \xrightarrow{\ \Phi\ } \boxed{\begin{aligned} P^{n+1} &= \widehat{P}^{n+1} + \Phi \\ \mathbf{u}^{n+1} &= \hat{\mathbf{u}}^{n+1} + \frac{\delta t}{\rho} \nabla \Phi \end{aligned}}$$

2. Fluid Phase Models: Computational Fluid Dynamics (CFD)

Finite differences in space: requires discretization of space

Divide space up in cells of $\delta l \cdot \delta l \cdot \delta l$

Define P_{ijk} as the pressure at the **center** of the cell $\{i, j, k\}$

Then for instance

$$\left(\frac{dP}{dx}\right)_{i+\frac{1}{2},j,k} = \frac{P_{i+1,j,k} - P_{i,j,k}}{\delta l}$$

Note: velocity is calculated
from an equation like:

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\delta t}{\rho}\nabla P^{n+1} - \frac{\delta t}{\rho}\mathbf{A}^n$$

→ Requires that velocities are defined at the **faces** of the cell

$$[u_x]_{i+\frac{1}{2},j,k} \qquad [u_y]_{i,j+\frac{1}{2},k} \qquad [u_z]_{i,j,k+\frac{1}{2}}$$

Then for instance

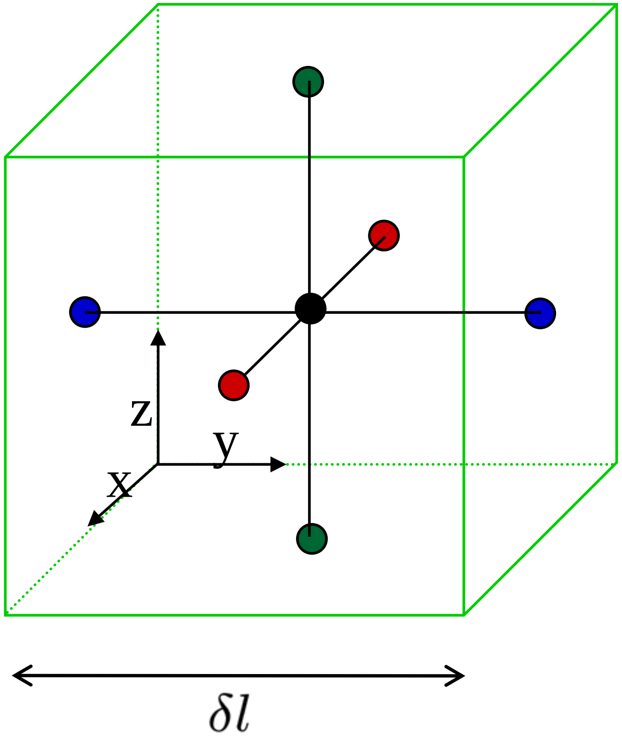
$$\left[\frac{du_x}{dx}\right]_{i,j,k} = \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\delta l}$$

is again defined at the cell **center**

2. Fluid Phase Models: Computational Fluid Dynamics (CFD)

Staggered grid

Define scalar variables at the cell centers, vector variables at the cell faces



- scalar variables $P, \ \rho$
- x-velocity component u_x
- y-velocity component u_y
- z-velocity component u_z

2. Fluid Phase Models: Computational Fluid Dynamics (CFD)

Resolution in time and space set by $\delta t, \delta l$

Any instability will originate from the explicit term in the velocity update

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\delta t}{\rho}\nabla P^{n+1} - \frac{\delta t}{\rho}\mathbf{A}^n \qquad \mathbf{A} = \nabla\cdot\rho\mathbf{u}\mathbf{u} + \nabla\cdot\tau$$

Stability condition from explicit treatment of

i) Convective term: $u_x + u_y + u_z < \dfrac{\delta l}{\delta t}$ *(Courant)*

ii) Stress term: $6\,\nu < \dfrac{(\delta l)^2}{\delta t}$

2. Fluid Phase Models: Lattice Boltzmann Method (LBM)

Hydrodynamic variables for the gas phase: $\rho(\mathbf{r})$ and $\mathbf{u}(\mathbf{r})$

These 4 variables can be captured by 1 variable: $f(\mathbf{r}, \mathbf{c})$

$$\begin{aligned} \int d\mathbf{c}\; f(\mathbf{r}, \mathbf{c}) &= \rho(\mathbf{r}) \\ \int d\mathbf{c}\; \mathbf{c}\; f(\mathbf{r}, \mathbf{c}) &= \rho(\mathbf{r})\mathbf{u}(\mathbf{r}) \end{aligned}$$

Time evolution of $f(\mathbf{r}, \mathbf{c})$: the Boltzmann Equation (BE)

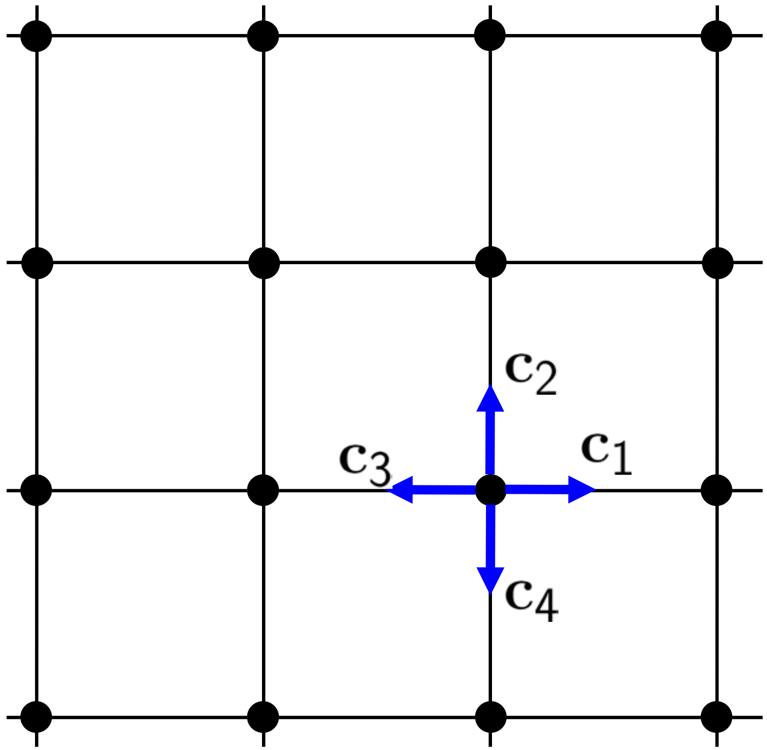
$$\boxed{\partial_t f + \mathbf{c}\cdot\nabla f = \mathcal{C}(f)} \qquad \mathcal{C}(f) = -\frac{f - f_{\text{eq}}(\rho, \mathbf{u})}{\tau}$$

$$\int d\mathbf{c}\; \boxed{} \Rightarrow \quad \partial_t \rho + \nabla\cdot(\rho\mathbf{u}) = 0$$

$$\int d\mathbf{c}\; \mathbf{c}\; \boxed{} \Rightarrow \quad \partial_t (\rho\mathbf{u}) + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) = -\nabla p - \nabla\cdot\tau$$

2. Fluid Phase Models: Lattice Boltzmann Method (LBM)

Discretization of coordinate and velocity space



$$f(\mathbf{r}, \mathbf{c}) \longrightarrow f_i(\mathbf{r})$$

$\mathbf{r}$ restricted to lattice sites

$\downarrow$

discrete velocities $\mathbf{c}_i$, such that $\mathbf{r} + \mathbf{c}_i \delta \mathrm{t}$ is located on a neighboring lattice site

For the 2-D square lattice:
4 velocities:

$$\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\} \qquad |\mathbf{c}_i| = \frac{\delta l}{\delta t}$$

2. Fluid Phase Models: Lattice Boltzmann Method (LBM)

Continuous

$$f(\mathbf{r}, \mathbf{c}, t)$$

$$\rho(\mathbf{r}, t) = \int d\mathbf{c}\, f(\mathbf{r}, \mathbf{c}, t)$$

$$\rho(\mathbf{r}, t)\, \mathbf{u}(\mathbf{r}, t) = \int d\mathbf{c}\, \mathbf{c}\, f(\mathbf{r}, \mathbf{c}, t)$$

$$\partial_t f + \mathbf{c} \cdot \nabla \mathrm{f} = \mathcal{C}(\mathrm{f})$$

$$\mathcal{C}(f) = -\frac{f - f^{\mathrm{eq}}(\rho, \mathbf{u})}{\tau}$$

Discrete

$$f_i(\mathbf{r}, t)$$

$$\rho(\mathbf{r}, t) = \sum_i f_i(\mathbf{r}, t)$$

$$\rho(\mathbf{r}, t)\, \mathbf{u}(\mathbf{r}, t) = \sum_i \mathbf{c}_i\, f_i(\mathbf{r}, t)$$

$$f_i(\mathbf{r} + \mathbf{c}_i \delta t, t + \delta t) = f_i(\mathbf{r}, t) + \mathcal{C}_i$$

$$\mathcal{C}_i = -\frac{f_i - f_i^{\mathrm{eq}}(\rho, \mathbf{u})}{\tau}$$

$$f_i^{\mathrm{eq}} = \rho\, e^{-(\mathbf{c}_\mathrm{i} - \mathbf{u})^2 / c_s^2}$$

Update in lattice-Boltzmann scheme:

$$\rho\mathbf{u} = \textstyle\sum_i \mathbf{c}_i f_i$$
$$\rho = \textstyle\sum_i f_i$$

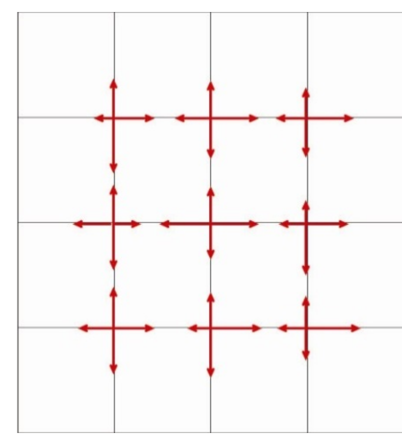
$$f_i(\mathbf{r},t) \longrightarrow \rho(\mathbf{r},t)\ ,\ \mathbf{u}(\mathbf{r},t) \longrightarrow \text{output}$$

$$f_i^{\text{eq}} = \rho e^{-(\mathbf{c}_\text{i}-\mathbf{u})^2/c_s^2}$$

$$f_i^{\text{eq}}(\mathbf{r},t)$$

$$f_i^*(\mathbf{r},t) \;=\; f_i(\mathbf{r},t) \;-\frac{1}{\tau}\left[f_i(\mathbf{r},t)-f_i^{\text{eq}}(\mathbf{r},t)\right]$$

$$f_i(\mathbf{r}+\mathbf{c}_i\delta t, t+\delta t) \;=\; f_i^*(\mathbf{r},t)$$



No iterations

All calculations are local

2. Fluid Phase Models: Lattice Boltzmann Method (LBM)

Advantages Lattice-Boltzmann:

- Easy to program
- Ideally suited for parallelization
- Simple boundary conditions
- Faster than CFD ?

Disavantages Lattice-Boltzmann:

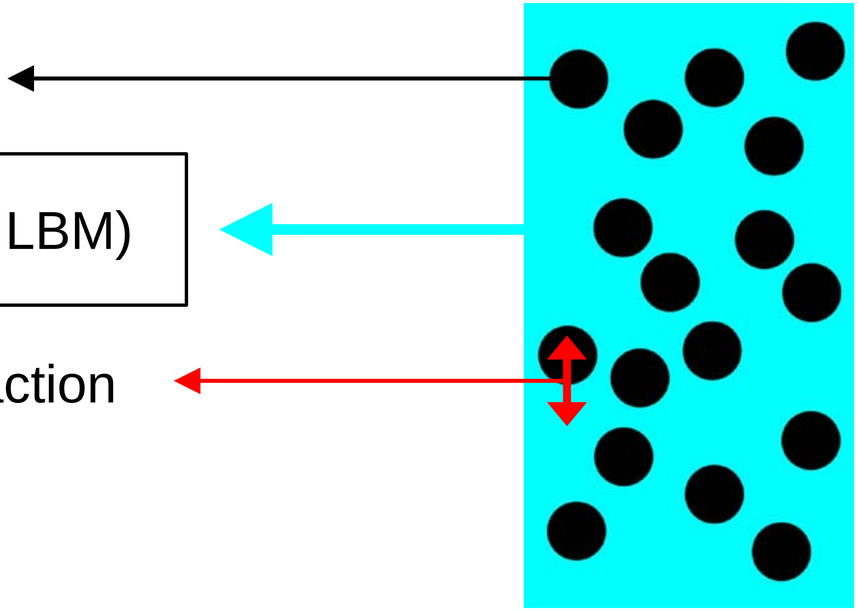
- Stability conditions not as clear as in CFD
- Conversion to SI units is less straightforward
- Not straightforward to include heat transer, and/or GLS flow

Outline

1. Models for the solid phase

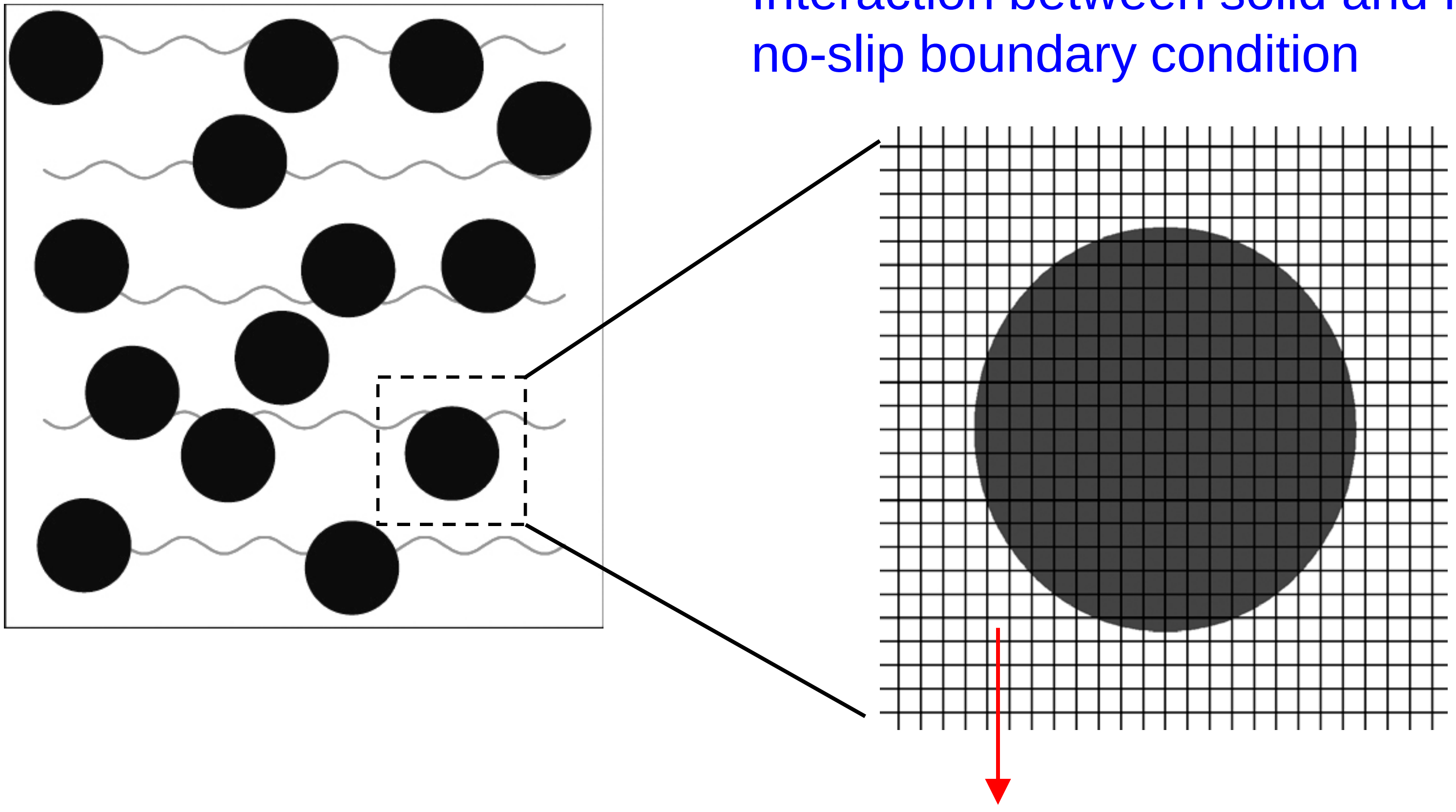
2. Models for the fluid phase (CFD, LBM)

3. Fully resolved fluid-particle interaction



3. Fully-resolved fluid-particle interaction

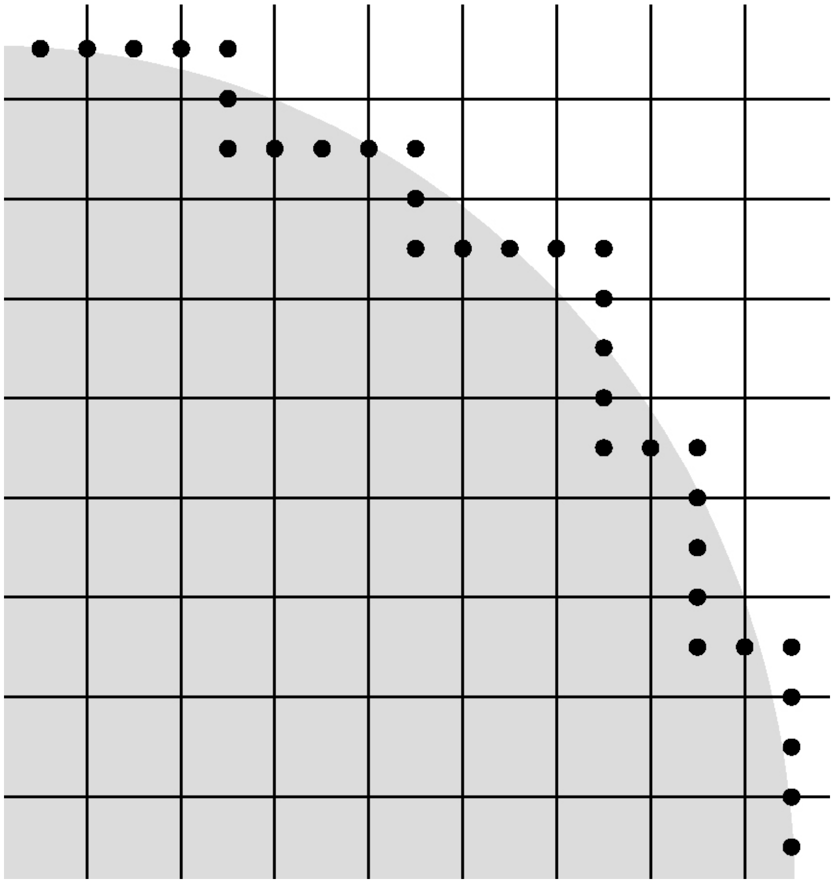
Interaction between solid and fluid:
no-slip boundary condition



LBM or CFD

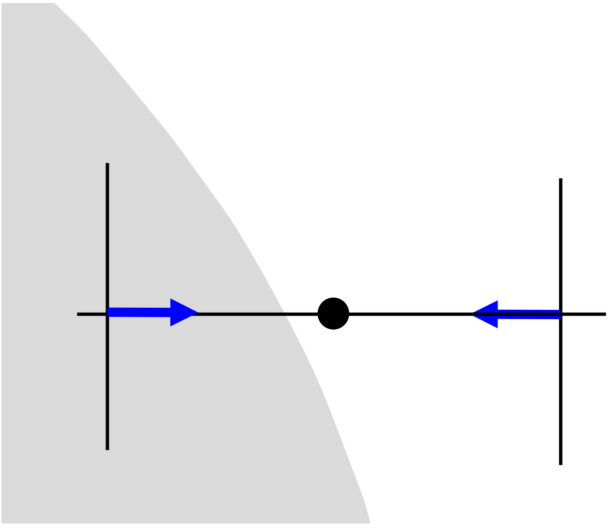
3. Fully resolved fluid-particle interaction: LBM

Resolved flow with LBM: Bounce Back at boundary nodes



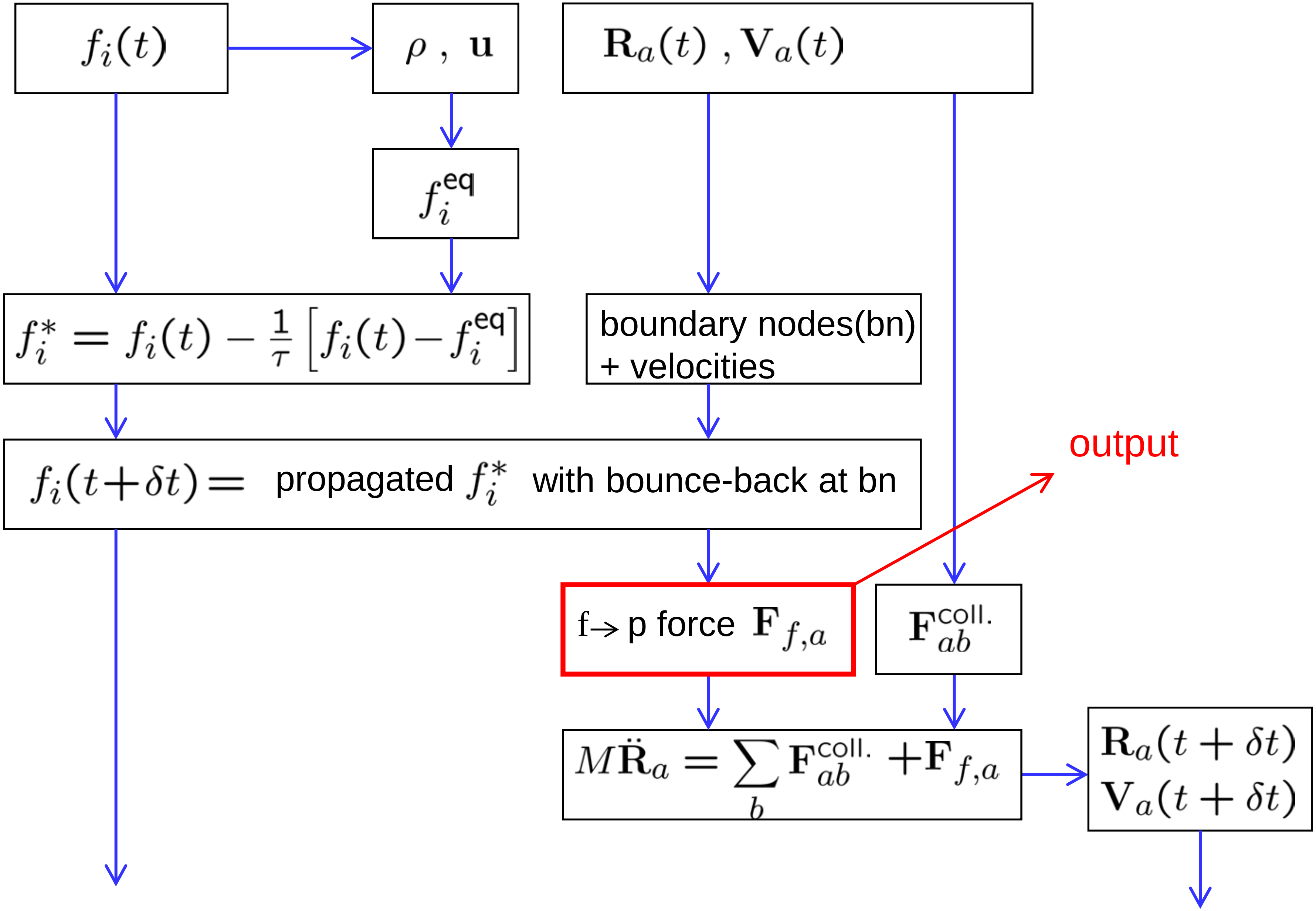
Define boundary node as point halfway an exterior and interior lattice site

In the propagation step: distribution "bounces back" at boundary nodes, and returns to its original site → average flow velocity is zero at boundary site



Update with boundary rules

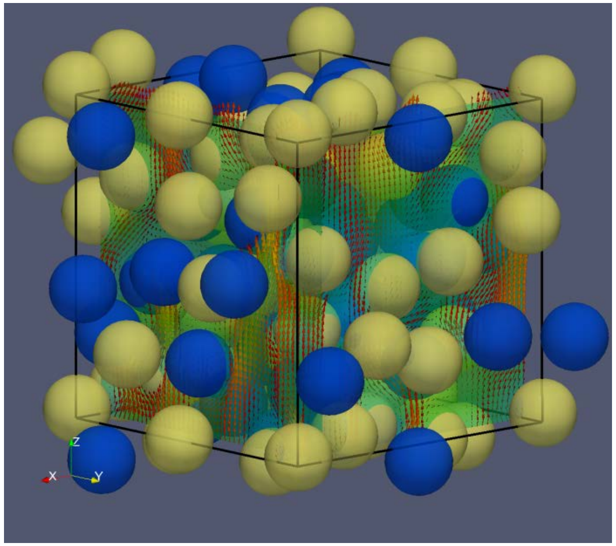
Fluid phase: Solid phase:

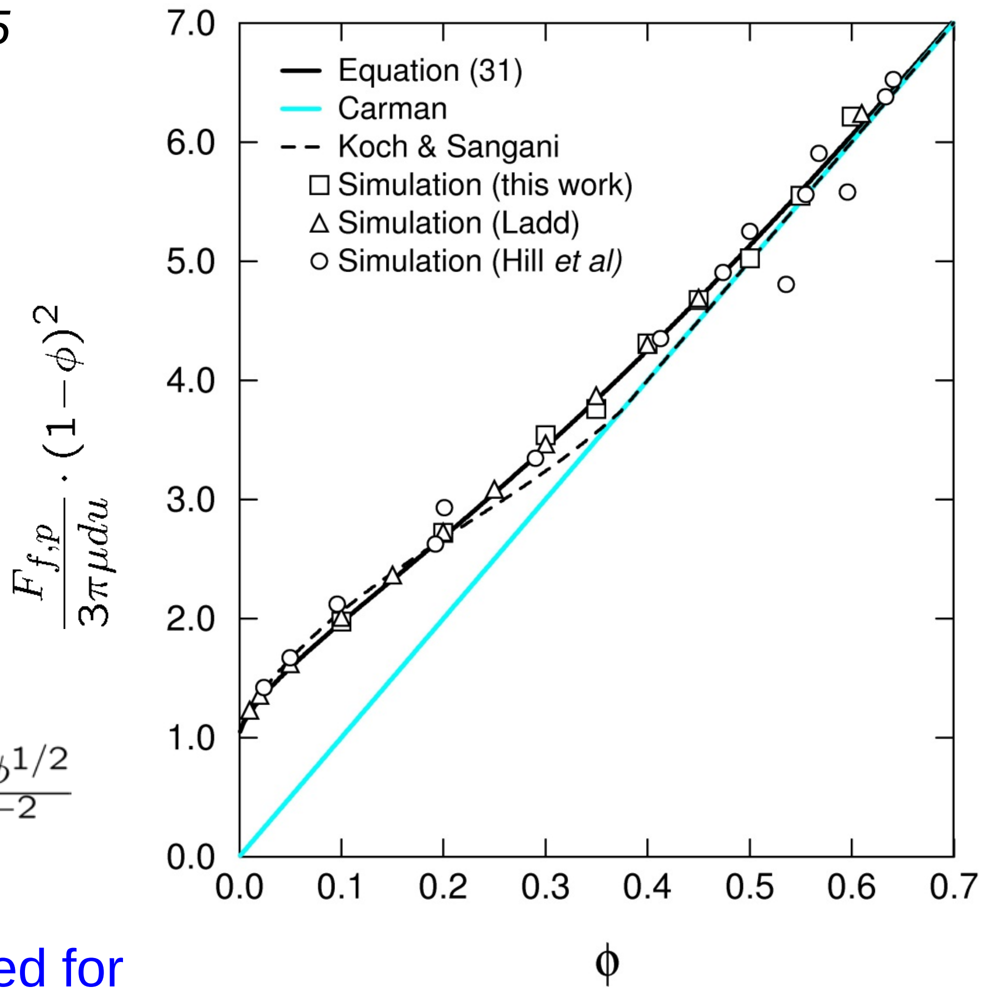


3. Fully resolved fluid-particle interaction: LBM

Fluid-particle force for a (infinite) random array (Stokes flow)

vdH, Beetstra & Kuipers, JFM 2005



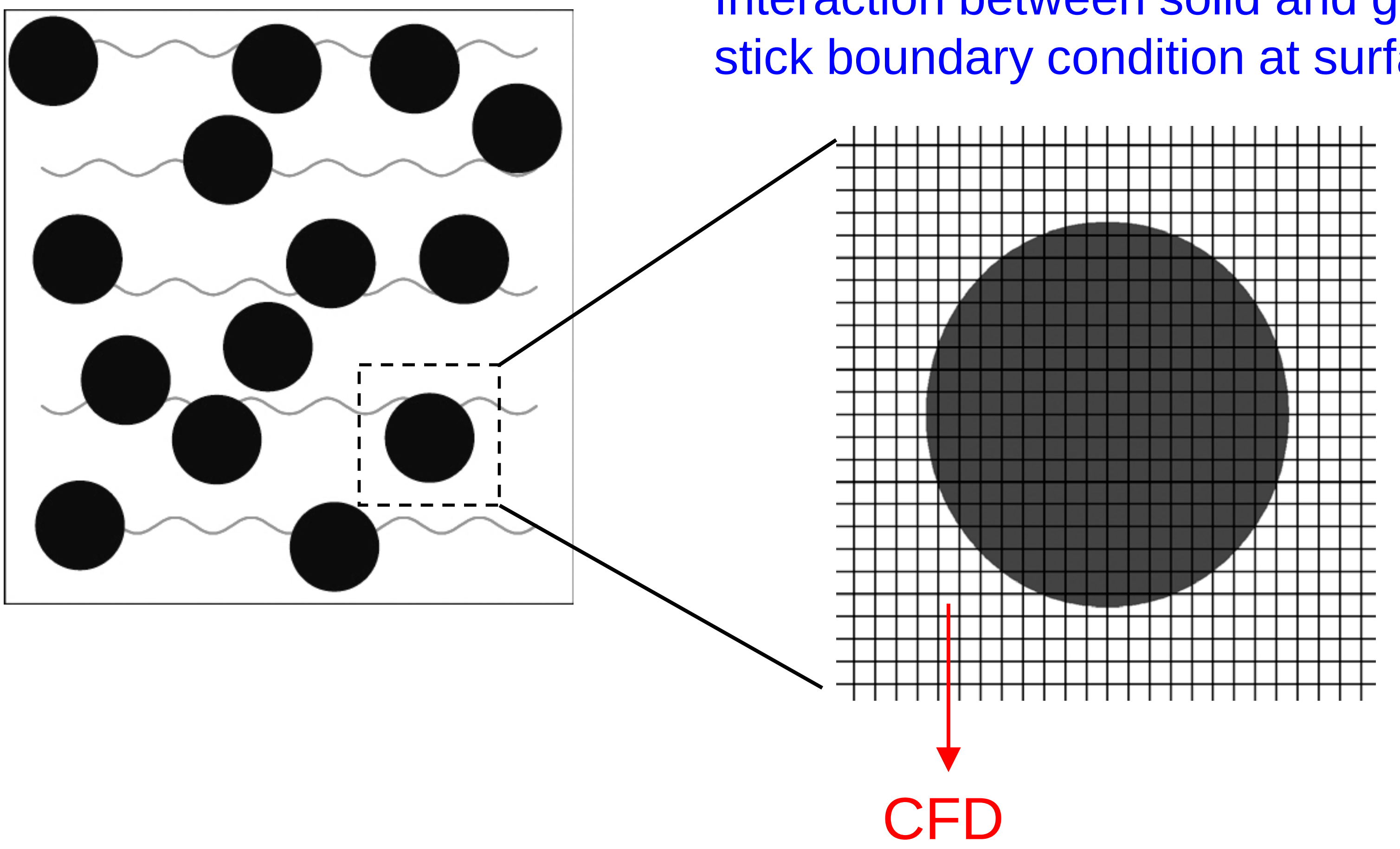


$$F(\phi) = \frac{F_{f,p}}{3\pi\mu d u} = \frac{10\phi}{(1-\phi)^2} + \frac{1+1.5\,\phi^{1/2}}{(1-\phi)^{-2}}$$

Relations have also been obtained for general Re and polydisperse systems

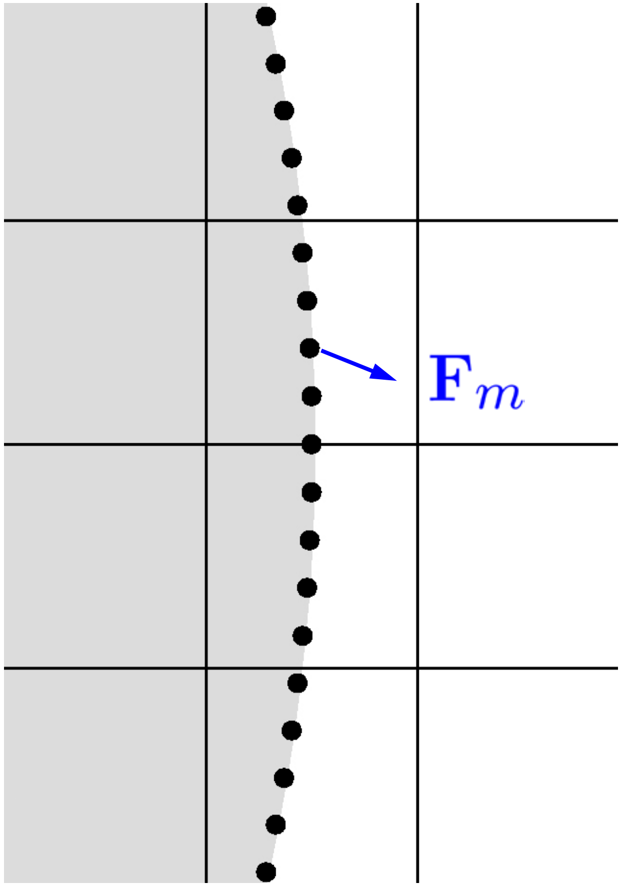
3. Fully resolved fluid-particle interaction: CFD

Interaction between solid and gas:
stick boundary condition at surface



3. Fully resolved fluid-particle interaction: CFD

Immersed Boundary Method (Uhlmann (2005)):



Define marker point on the surface, each of which applies a force $\mathbf{F}_m$ on the fluid, such that the velocity $\mathbf{u}_m$ of the fluid at the marker point is equal to the surface velocity $\mathbf{w}_m$

$$\mathbf{F}_m = C\,\delta l^3 \cdot \frac{\rho}{\delta t}(\mathbf{w}_m - \tilde{\mathbf{u}}_m)$$

Velocity from update <u>without</u> forcing

$$C = \frac{4\pi}{N}\left(\frac{R}{\delta l}\right)^2$$

3. Fully resolved fluid-particle interaction: CFD

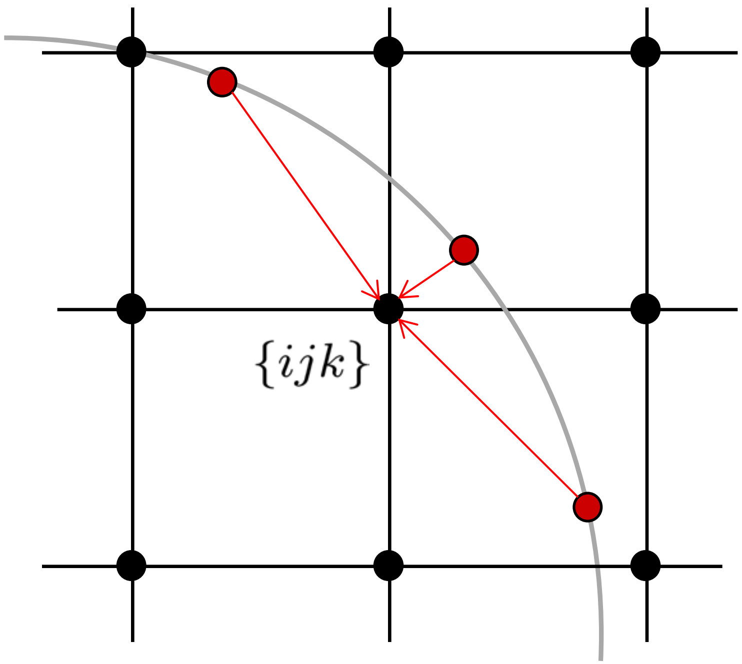
Force at marker point location:

$$\mathbf{F}_m = C\,\delta l^3 \cdot \frac{\rho}{\delta t}\,(\mathbf{w}_m - \tilde{\mathbf{u}}_m)$$

Total force density at $\{ijk\}$ from all force points in range of $\{ijk\}$

Update flow field on Eulerian grid:

$$\mathbf{u}_{ijk}^{n+1} = \mathbf{u}_{ijk}^{n} - \frac{\delta t}{\rho}(\nabla P^{n+1})_{ijk} - \frac{\delta t}{\rho}\mathbf{A}_{ijk}^{n} + \frac{\delta t}{\rho}\mathbf{f}_{ijk}$$



Required: Mapping

Lagrange → Euler

Euler → Lagrange

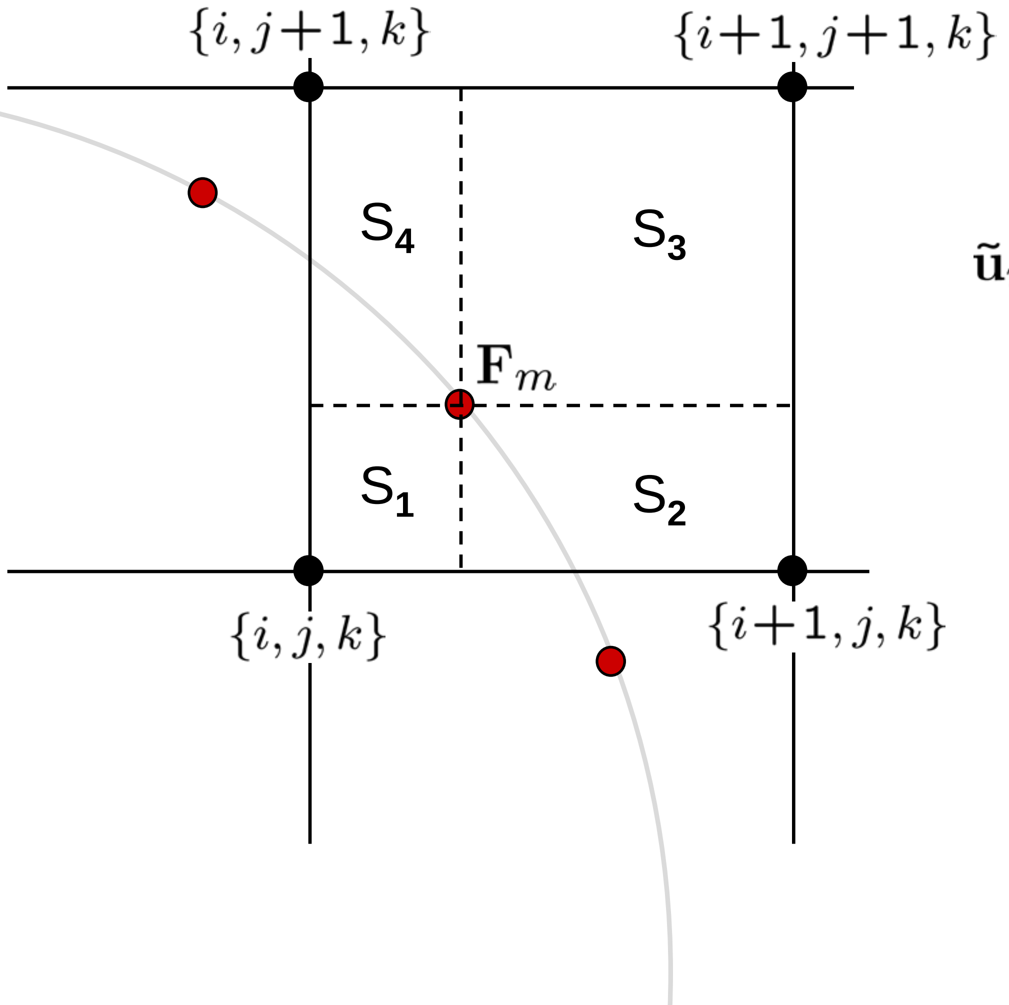
(note: for simplicity, we assume that the velocity is defined on gridpoints ijk (no staggered grid))

3. Fully resolved fluid-particle interaction: CFD

Euler → Lagrange mapping: Volume weighing

Basic idea shown in 2-D (surface weighing)

$$\mathbf{F}_m = C\,\delta l^3 \cdot \frac{\rho}{\delta t}\left(\mathbf{w}_m - \tilde{\mathbf{u}}_m\right)$$



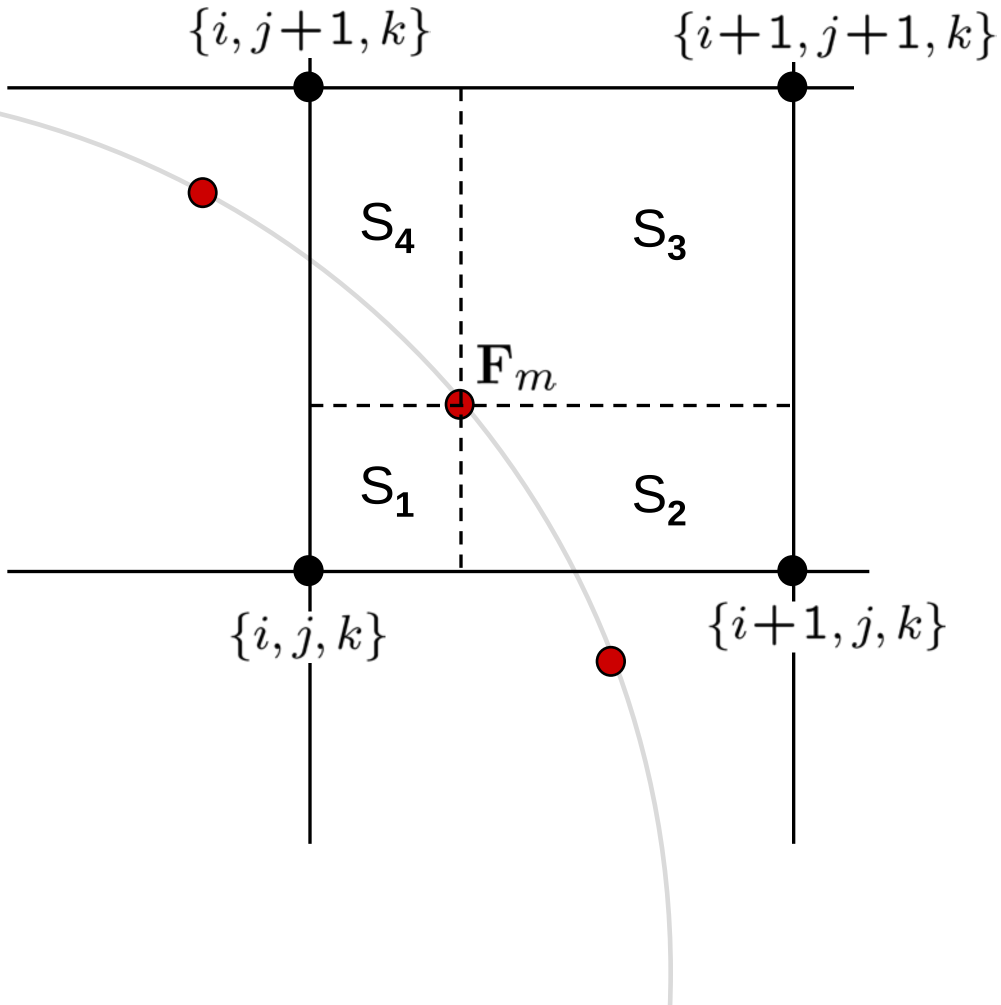
$$\tilde{\mathbf{u}}_m = \frac{N}{S_1}\tilde{\mathbf{u}}_{i,j,k} + \frac{N}{S_2}\tilde{\mathbf{u}}_{i+1,j,k} + \frac{N}{S_3}\tilde{\mathbf{u}}_{i+1,j+1,k} + \frac{N}{S_4}\tilde{\mathbf{u}}_{i,j+1,k}$$

$$\frac{N}{S_1} + \frac{N}{S_2} + \frac{N}{S_3} + \frac{N}{S_4} = 1$$

3. Fully resolved fluid-particle interaction: CFD

Lagrange → Euler mapping: Volume weighing

Basic idea shown in 2-D (surface weighing)



$$\mathbf{f}_{i,j,k} = \frac{N}{S_1} \cdot \frac{\mathbf{F}_m}{\delta l^3}$$

$$\mathbf{f}_{i+1,j,k} = \frac{N}{S_2} \cdot \frac{\mathbf{F}_m}{\delta l^3}$$

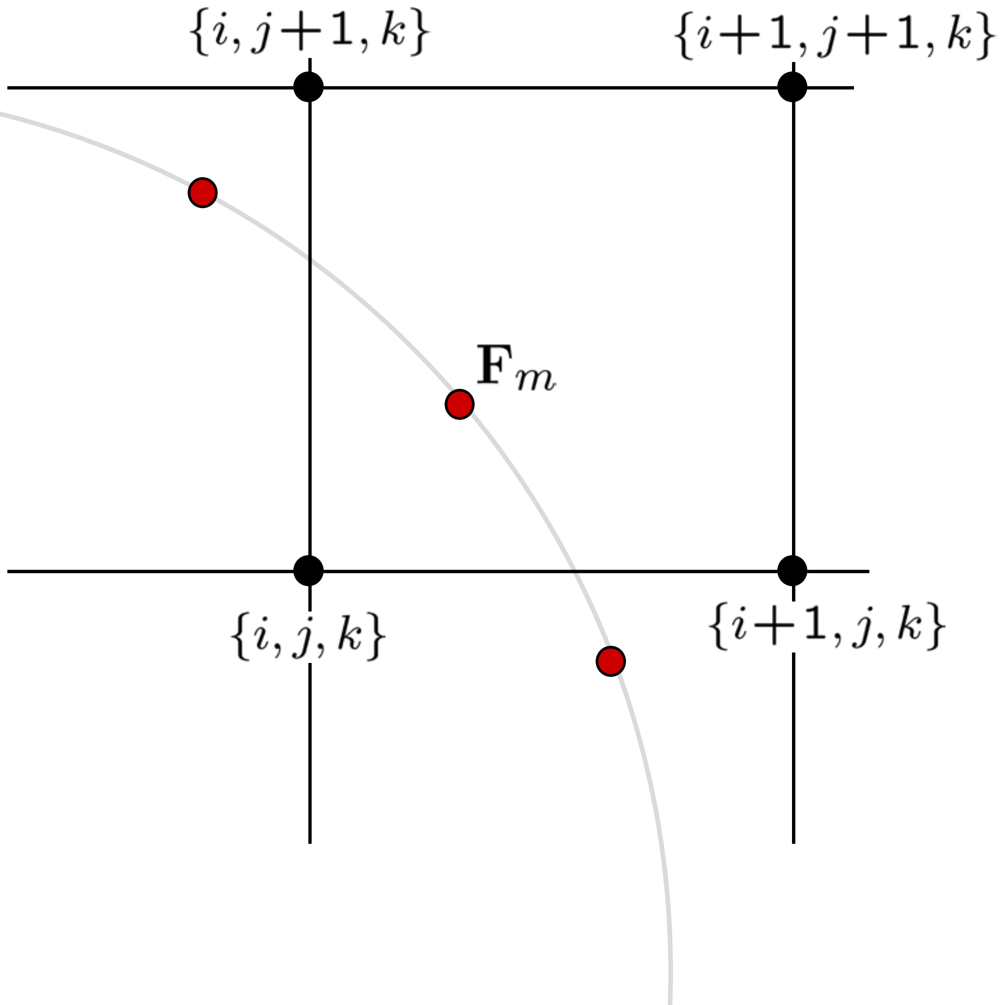
$$\mathbf{f}_{i+1,j+1,k} = \frac{N}{S_3} \cdot \frac{\mathbf{F}_m}{\delta l^3}$$

$$\mathbf{f}_{i,j+1,k} = \frac{N}{S_4} \cdot \frac{\mathbf{F}_m}{\delta l^3}$$

$$\frac{N}{S_1} + \frac{N}{S_2} + \frac{N}{S_3} + \frac{N}{S_4} = 1$$

3. Fully resolved fluid-particle interaction: CFD

Mapping can formally be written as:



Euler → Lagrange:

$$\tilde{\mathbf{u}}_m = \sum_{ijk} D(\mathbf{r}_{ijk} - \mathbf{r}_m) \cdot \tilde{\mathbf{u}}_{ijk}$$

Lagrange → Euler:

$$\tilde{\mathbf{f}}_{ijk} = \sum_{m} D(\mathbf{r}_{ijk} - \mathbf{r}_m) \cdot \frac{\mathbf{F}_m}{\delta l^3}$$

mapping is of course not restricted to volume-weighing

Solution procedure to calculate variables at time $n+1$ including IBM force $\mathbf{f}$

Initial guess:

$$\hat{P}^{n+1}_{ijk} \Rightarrow \nabla\hat{P}^{n+1}_{ijk}$$

$$\tilde{\mathbf{u}}^{n+1}_{ijk} = \mathbf{u}^{n}_{ijk} - \frac{\delta t}{\rho}\nabla\hat{P}^{n+1}_{ijk} - \frac{\delta t}{\rho}\mathbf{A}^{n}_{ijk}$$

Calculate velocity field without forcing

$$\tilde{\mathbf{u}}_{m} = \sum_{ijk} D(\mathbf{r}_{ijk} - \mathbf{r}_{m}) \cdot \tilde{\mathbf{u}}^{n+1}_{ijk}$$

Map velocity field to marker point locations

$$\mathbf{F}_{m} = C\,\delta l^{3} \cdot \frac{\rho}{\delta t}(\mathbf{w}_{m} - \tilde{\mathbf{u}}_{m})$$

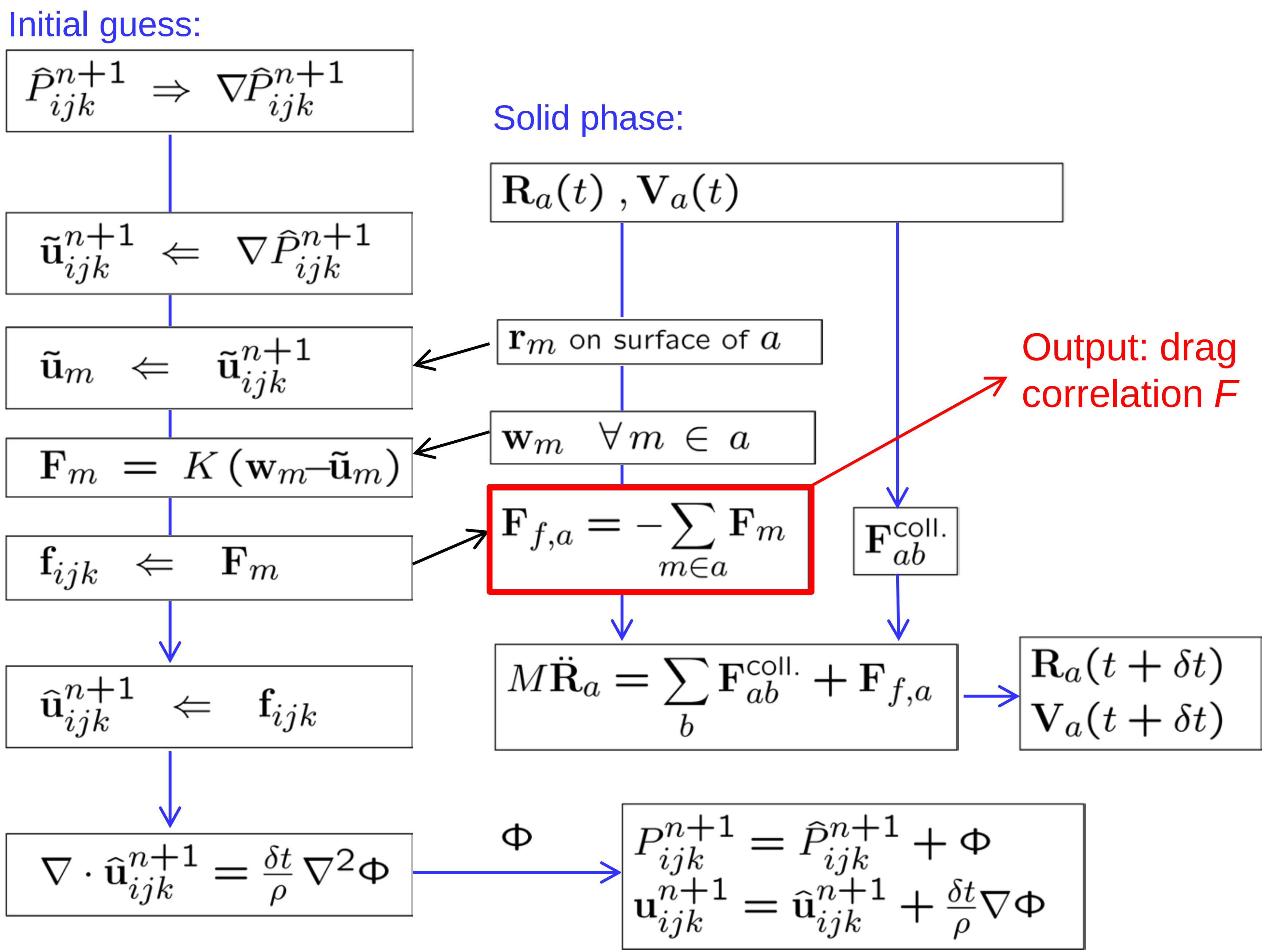
Calculate IBM force at marker point locations

$$\mathbf{f}_{ijk} = \sum_{m} D(\mathbf{r}_{ijk} - \mathbf{r}_{m}) \cdot \mathbf{F}_{m}/\delta l^{3}$$

Map IBM force to Eulerian grid

$$\hat{\mathbf{u}}^{n+1}_{ijk} = \mathbf{u}^{n}_{ijk} - \frac{\delta t}{\rho}\nabla\hat{P}^{n+1}_{ijk} - \frac{\delta t}{\rho}\mathbf{A}^{n}_{ijk} + \frac{\delta t}{\rho}\mathbf{f}_{ijk}$$

$$\nabla \cdot \hat{\mathbf{u}}^{n+1}_{ijk} = \frac{\delta t}{\rho}\nabla^{2}\Phi \xrightarrow{\Phi} \begin{aligned} P^{n+1}_{ijk} &= \hat{P}^{n+1}_{ijk} + \Phi \\ \mathbf{u}^{n+1}_{ijk} &= \hat{\mathbf{u}}^{n+1}_{ijk} + \frac{\delta t}{\rho}\nabla\Phi \end{aligned}$$



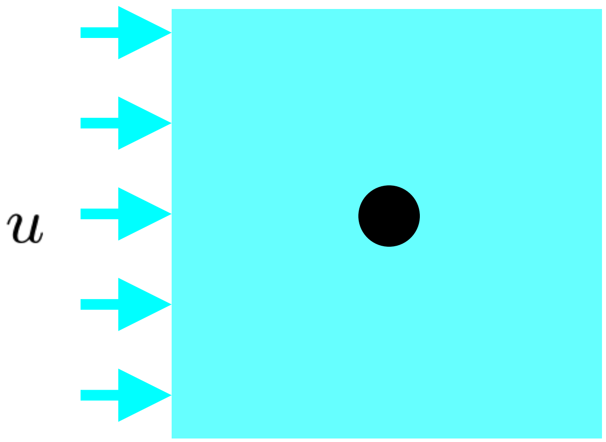
Initial guess:
$\widehat{P}_{ijk}^{n+1} \Rightarrow \nabla \widehat{P}_{ijk}^{n+1}$
$\tilde{\mathbf{u}}_{ijk}^{n+1} \Leftarrow \nabla \widehat{P}_{ijk}^{n+1}$
$\tilde{\mathbf{u}}_m \Leftarrow \tilde{\mathbf{u}}_{ijk}^{n+1}$
$\mathbf{F}_m = K(\mathbf{w}_m - \tilde{\mathbf{u}}_m)$
$\mathbf{f}_{ijk} \Leftarrow \mathbf{F}_m$
$\hat{\mathbf{u}}_{ijk}^{n+1} \Leftarrow \mathbf{f}_{ijk}$
$\nabla \cdot \hat{\mathbf{u}}_{ijk}^{n+1} = \frac{\delta t}{\rho} \nabla^2 \Phi$
Φ
Solid phase:
$\mathbf{R}_a(t)\ ,\mathbf{V}_a(t)$
$\mathbf{r}_m$ on surface of a
$\mathbf{w}_m \quad \forall m \in a$
$\mathbf{F}_{f,a} = -\sum_{m \in a} \mathbf{F}_m$
$\mathbf{F}_{ab}^{\text{coll.}}$
Output: drag correlation *F*
$M\ddot{\mathbf{R}}_a = \sum_b \mathbf{F}_{ab}^{\text{coll.}} + \mathbf{F}_{f,a}$
$\mathbf{R}_a(t+\delta t)$
$\mathbf{V}_a(t+\delta t)$
$P_{ijk}^{n+1} = \widehat{P}_{ijk}^{n+1} + \Phi$
$\mathbf{u}_{ijk}^{n+1} = \hat{\mathbf{u}}_{ijk}^{n+1} + \frac{\delta t}{\rho} \nabla \Phi$

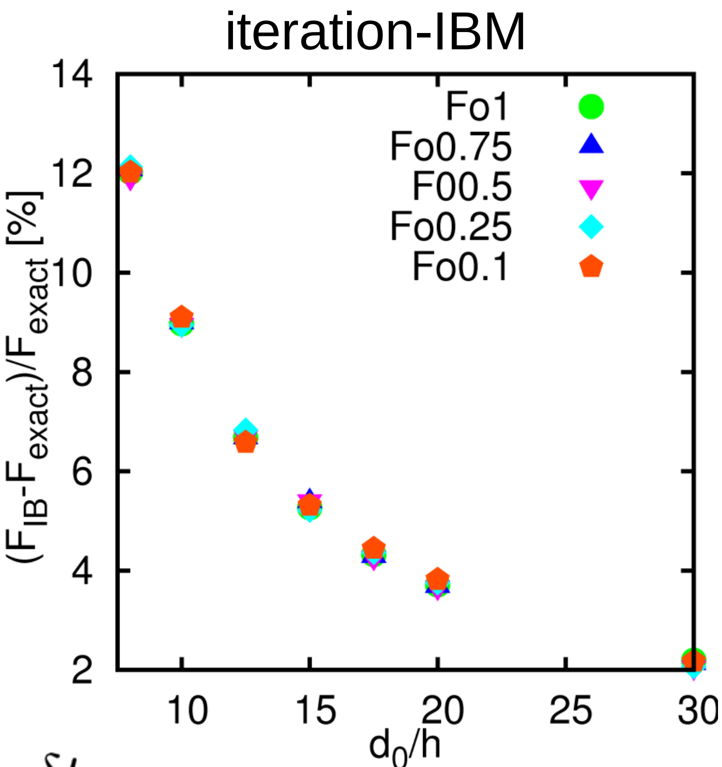
3. Fully resolved fluid-particle interaction: CFD

First test: comparison with the exact expression by Hasimoto for a dilute SQ infinite array:

$$\mathrm{F_{exact}} = \frac{1-\phi}{1-1.7601\phi^{1/3}+\phi-1.5593\phi^2}$$

$$\mathrm{F} = \frac{\mathbf{F}_{f,a}}{3\pi\mu d(\mathbf{u}-\mathbf{V}_a)}$$





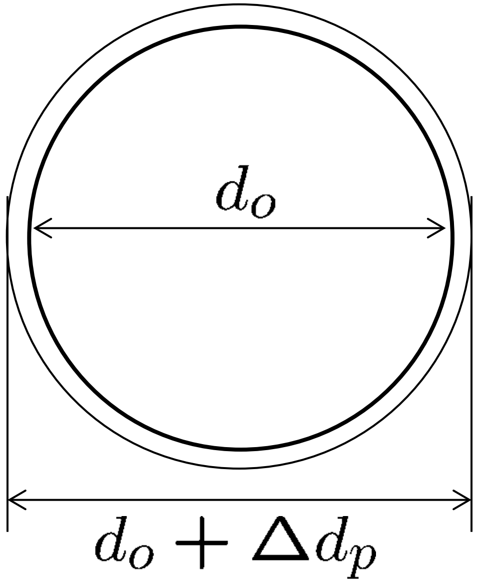
$$\mathrm{Fo} = \nu\frac{\delta t}{\delta l^2}$$

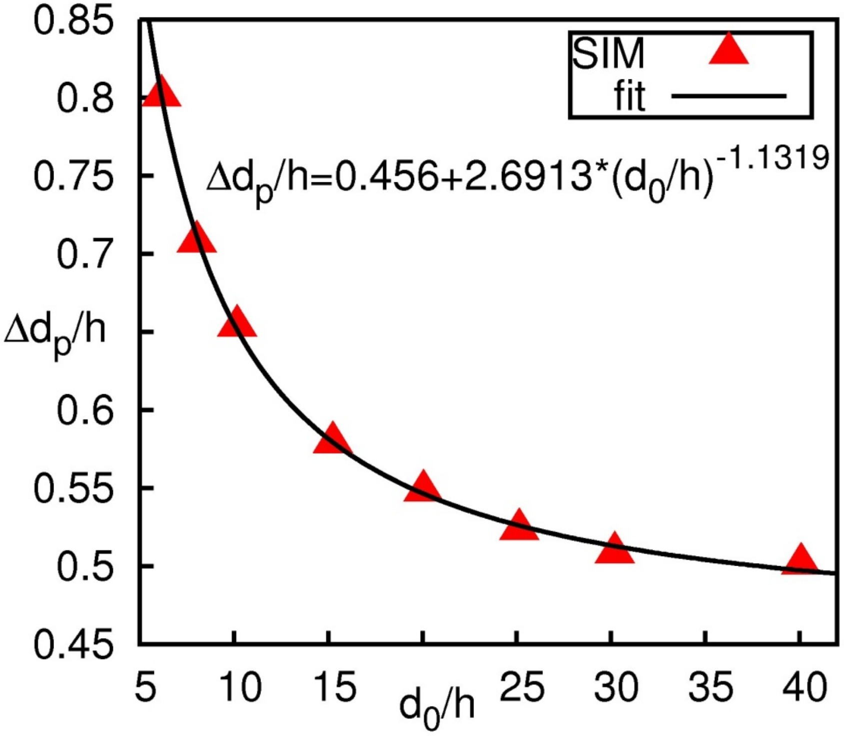
3. Fully resolved fluid-particle interaction: CFD

Idea:

Use Hasimoto for setting an effective hydrodynamic diameter (calibration)

$$F_{\text{exact}} = \frac{1-\phi}{1-1.7601\phi^{1/3}+\phi-1.5593\phi^2}$$

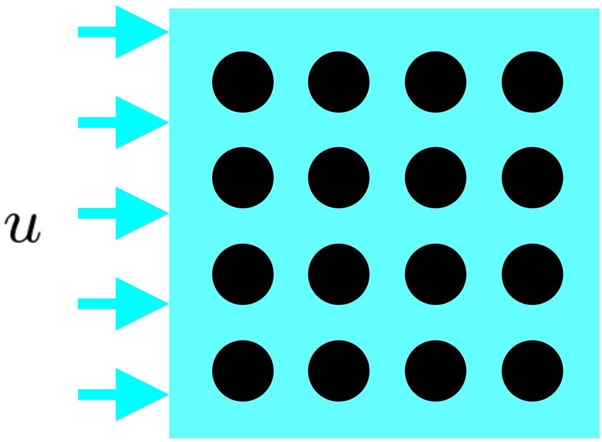


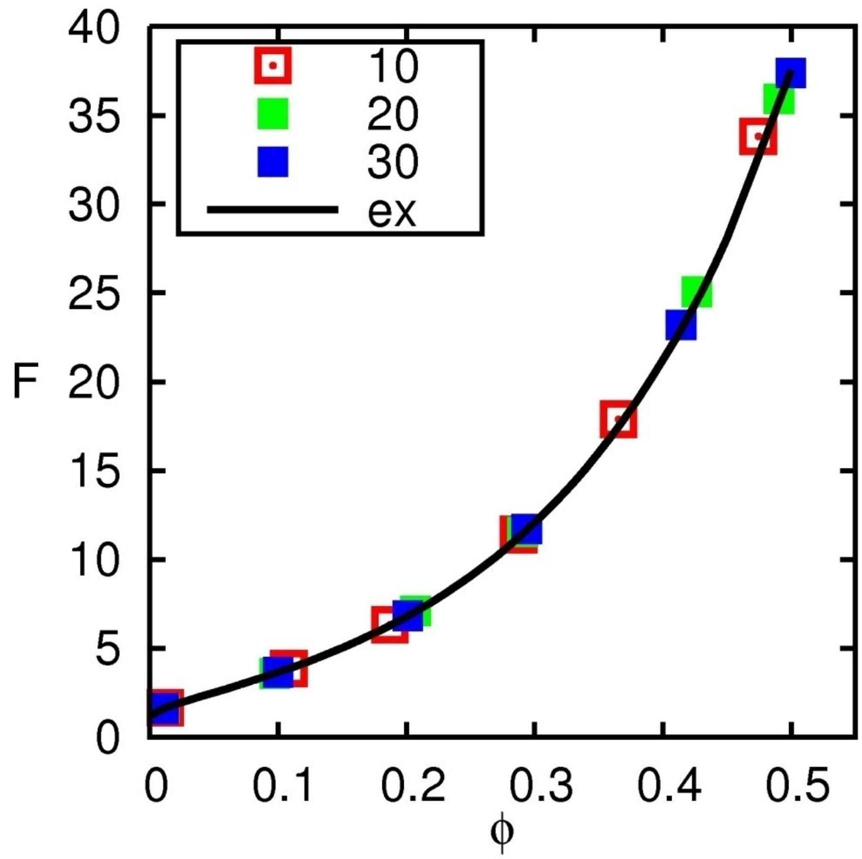


Note that also in LBM an effective diameter is used!

3. Fully resolved fluid-particle interaction: CFD

Validation: drag for a dense square array:



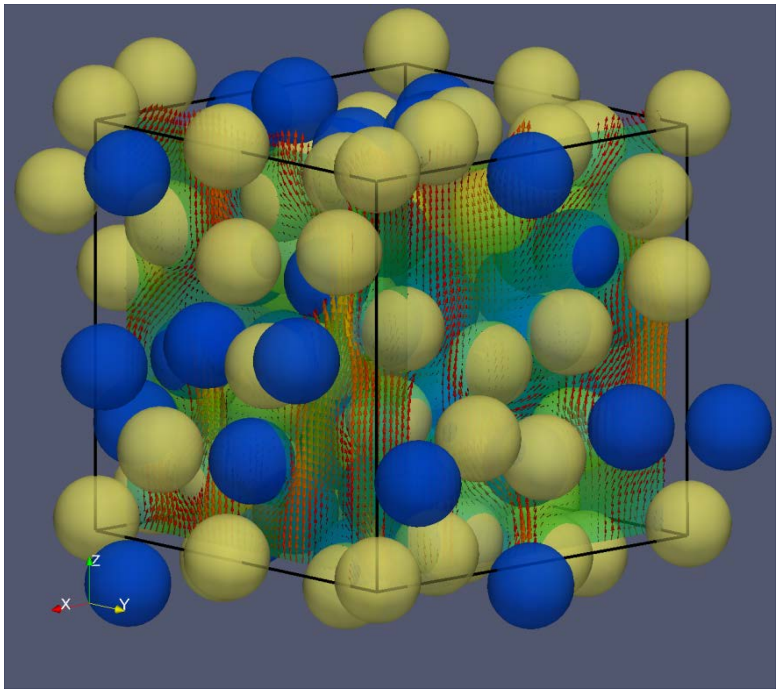


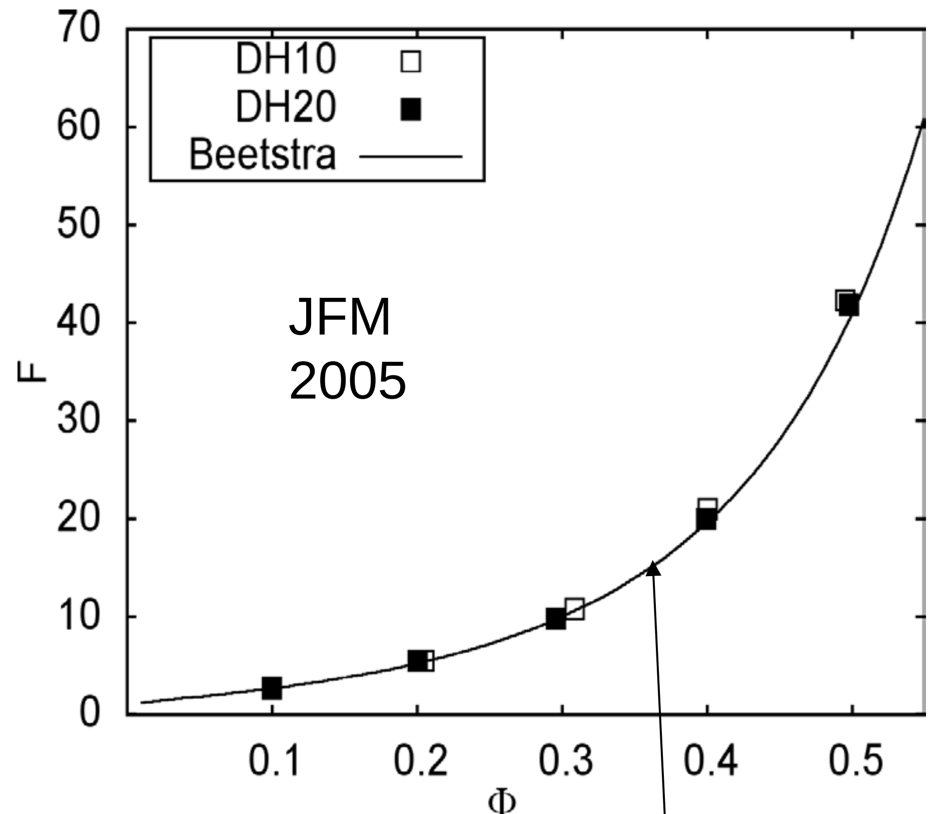
Without diameter correction

With diameter correction

3. Fully resolved fluid-particle interaction: CFD *vs* LBM

Comparison with LB results for the drag for a dense random array:

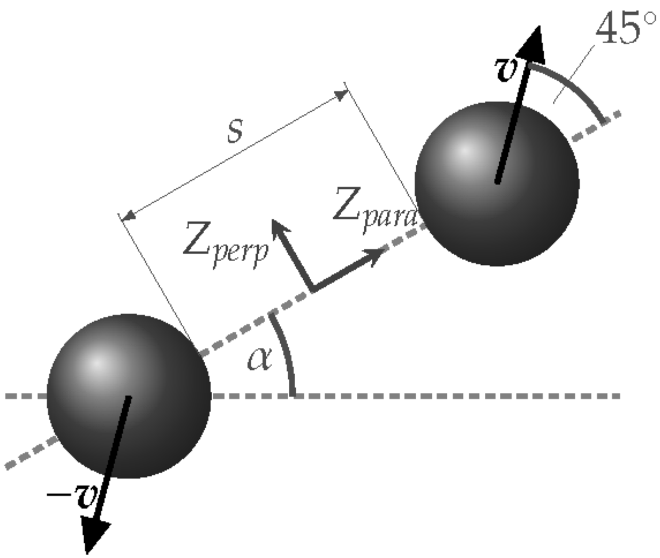


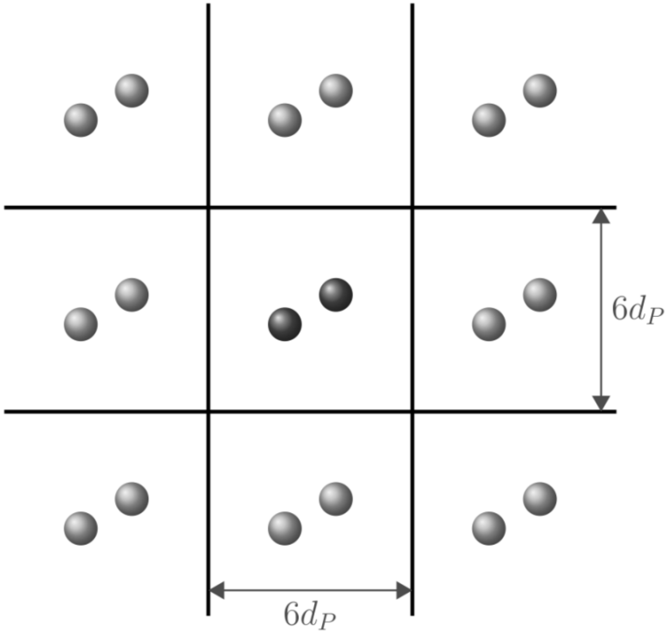


$$F(\phi) = \frac{F_{f,p}}{3\pi\mu d u} = \frac{10\phi}{(1-\phi)^2} + \frac{1+1.5\,\phi^{1/2}}{(1-\phi)^{-2}}$$

3. Fully resolved fluid-particle interaction: CFD *vs* LBM

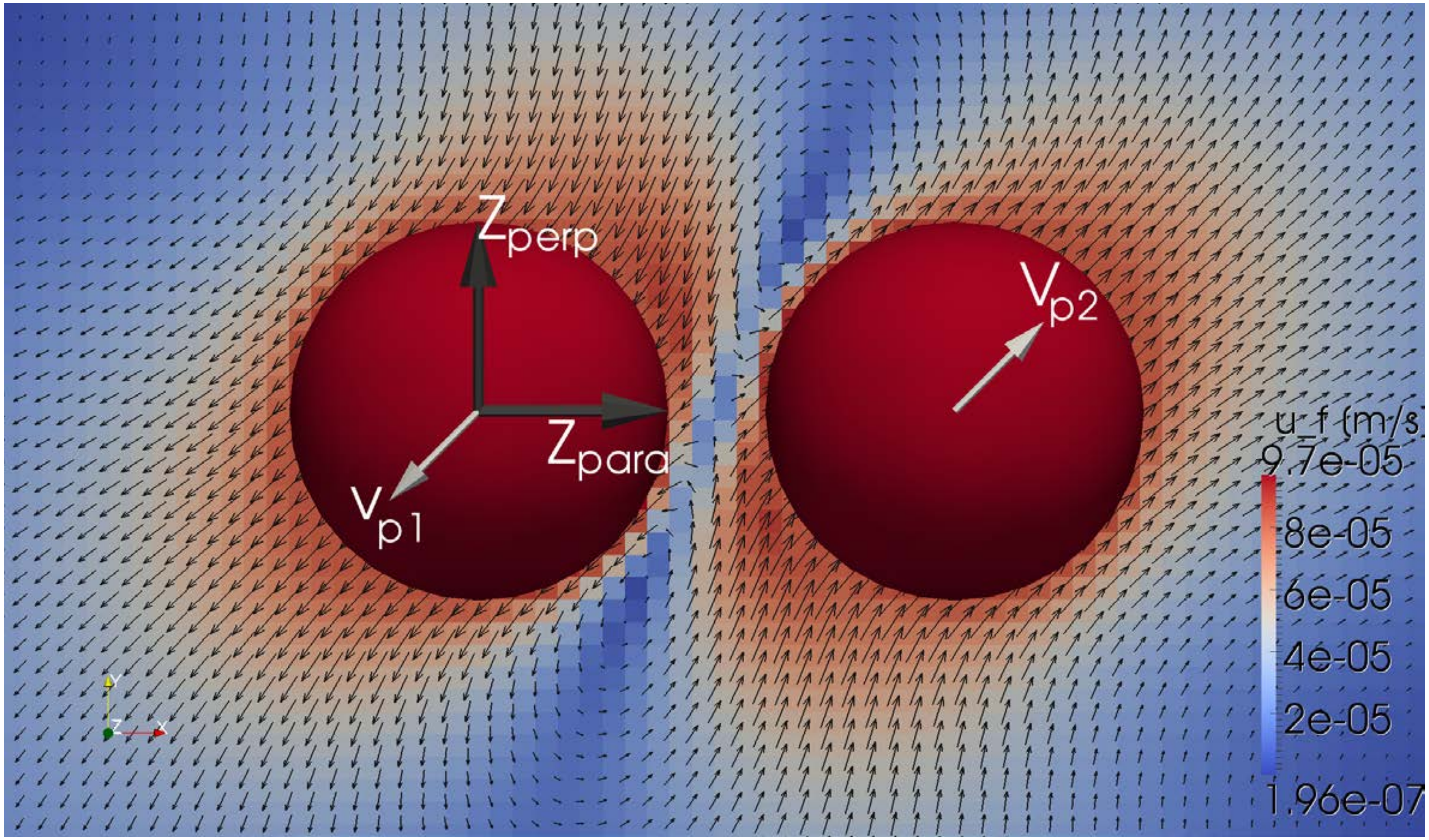
Interaction force between two particles in relative motion





- Particles fixed at their position
- Particles have equal, but opposite velocities, with Re << 1
- Surface-to-surface distance s varied
- Results compared with exact solution from multipole expansion of the Stokes eq.

3. Fully resolved fluid-particle interaction: CFD *vs* LBM

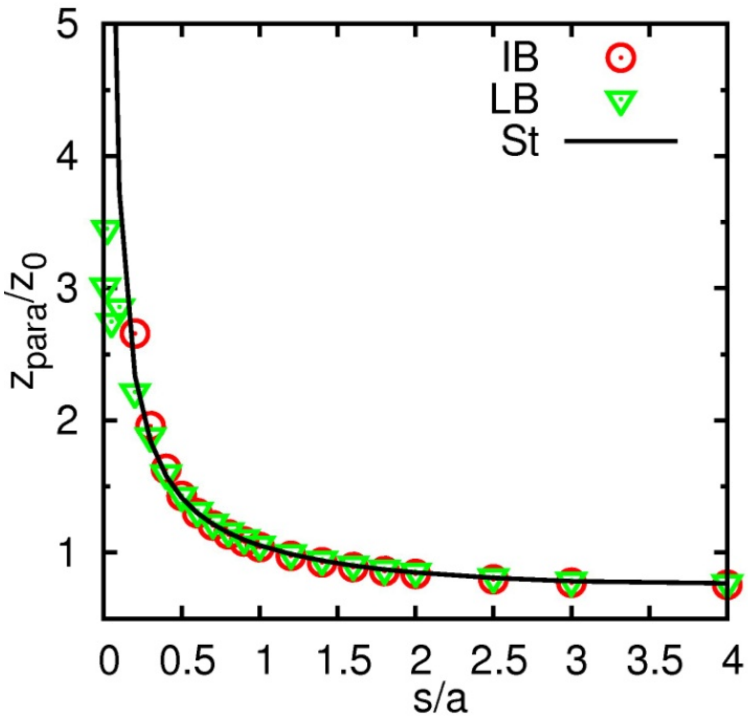


Thesis S.H.L. Kriebitzsch (2011)

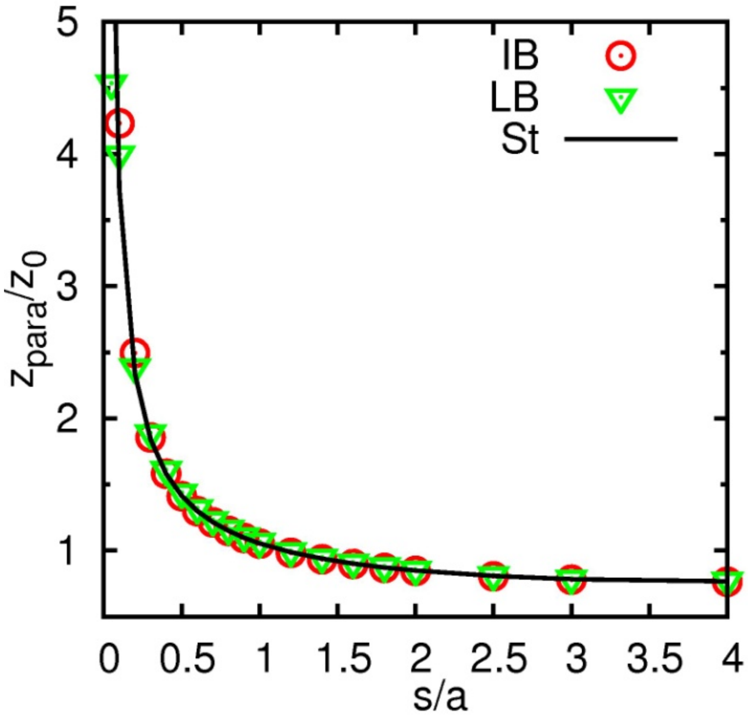
3. Fully resolved fluid-particle interaction: CFD *vs* LBM

Interaction force between two particles in relative motion

$d_o/\delta l = 9$



$d_o/\delta l = 17$

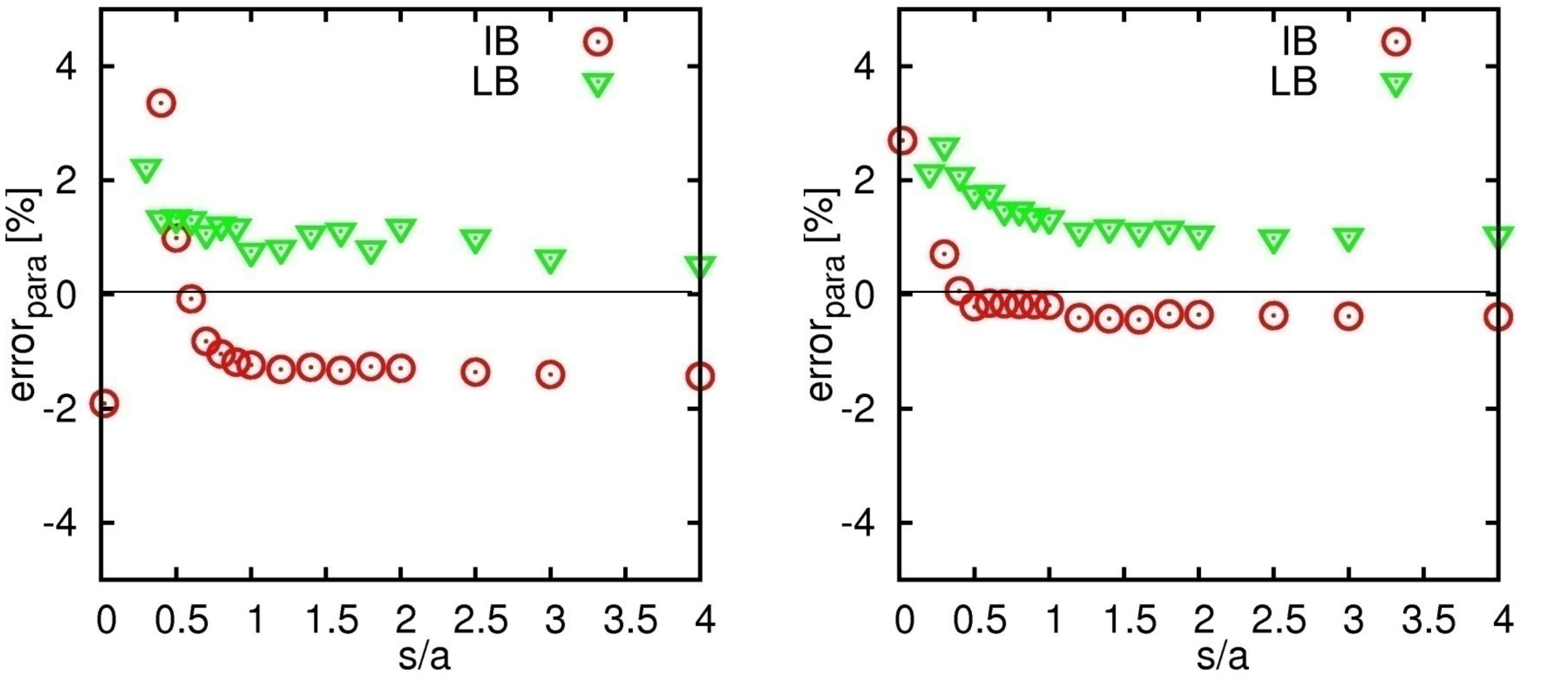


Thesis S.H.L. Kriebitzsch (2011)

3. Fully resolved fluid-particle interaction: CFD *vs* LBM

Interaction force between two particles in relative motion

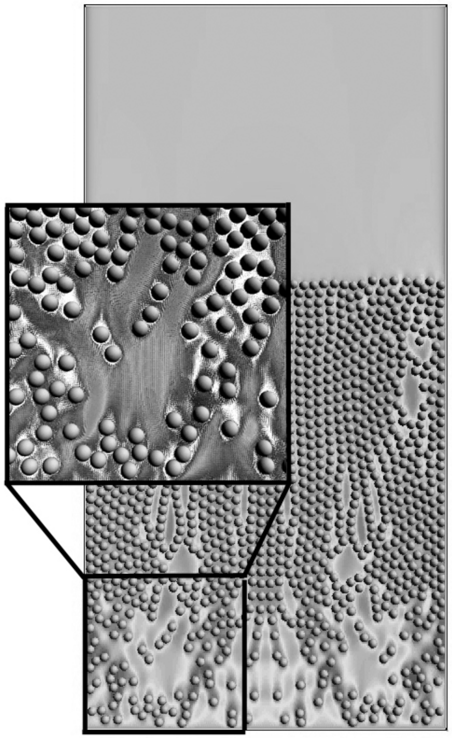
$d_o/\delta l =$ 9 $\qquad\qquad$ $d_o/\delta l =$ 17



Thesis S.H.L. Kriebitzsch (2011)

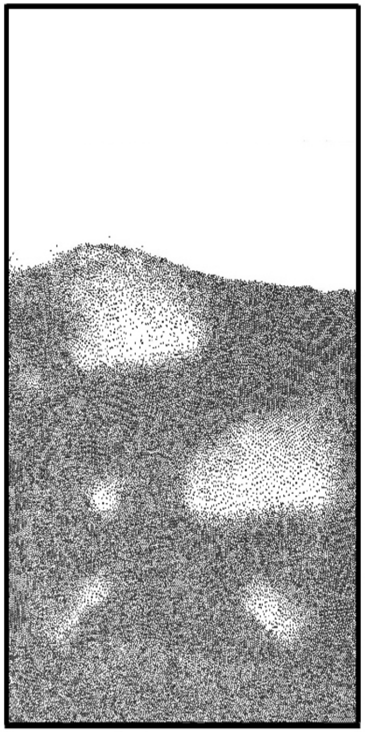
Three basic models for two-phase granular flow:

DNS



Euler
Lagrange

DEM



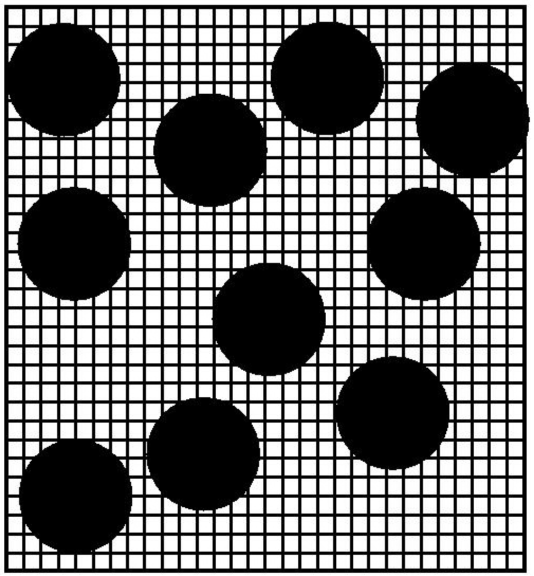
Euler
Lagrange

TFM

Euler
Euler

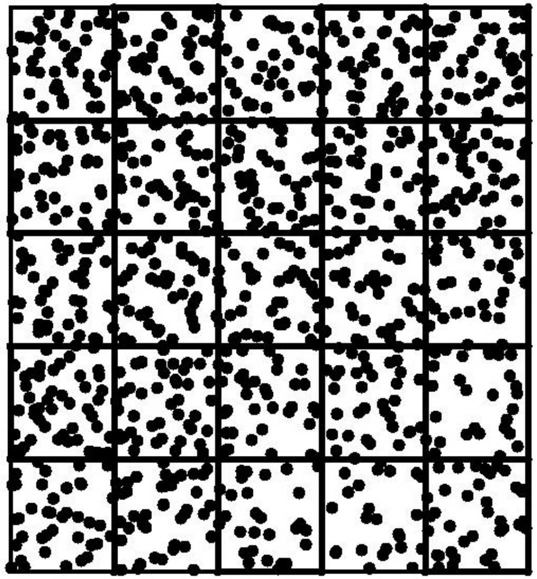
II. DEM models for granular two-phase flow

DNS



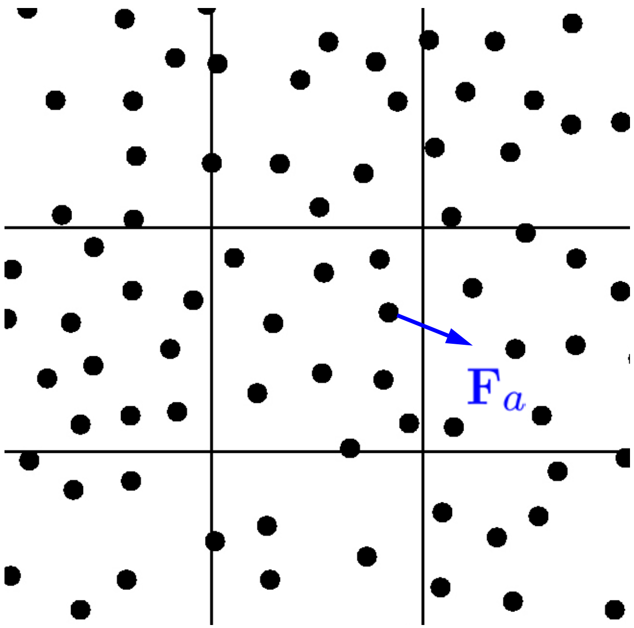
$\mathbf{F}_{f,a}$ follows from stick boundary conditions

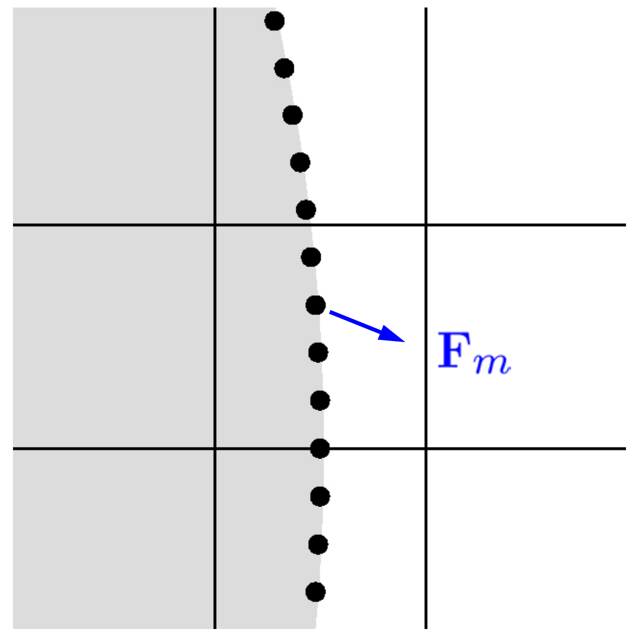
DEM



$\mathbf{F}_{f,a}$ estimated from relations based on the local ϕ

Unresolved flow: implementation similar to **resolved flow**



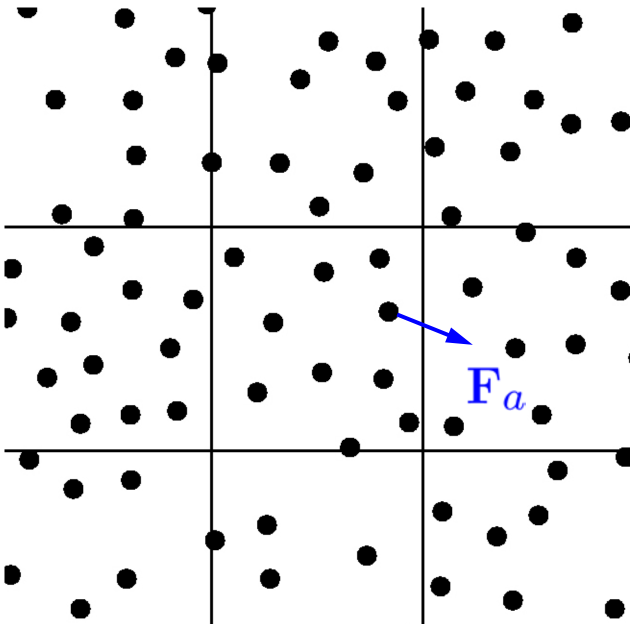


$$\begin{aligned}\mathbf{F}_a &= -\mathbf{F}_{f,a} \\ &= 3\pi\mu(\mathbf{v}_a - \mathbf{u}_a)\underbrace{F(\phi)}_{\frac{10\phi}{(1-\phi)^2} + \frac{1+1.5\,\phi^{1/2}}{(1-\phi)^{-2}}}\end{aligned}$$

$$\mathbf{F}_m = C\,\delta l^3 \cdot \frac{\rho}{\delta t}(\mathbf{w}_m - \tilde{\mathbf{u}}_m)$$

DEM models for granular two-phase flow

Unresolved flow: implementation similar to **resolved flow**

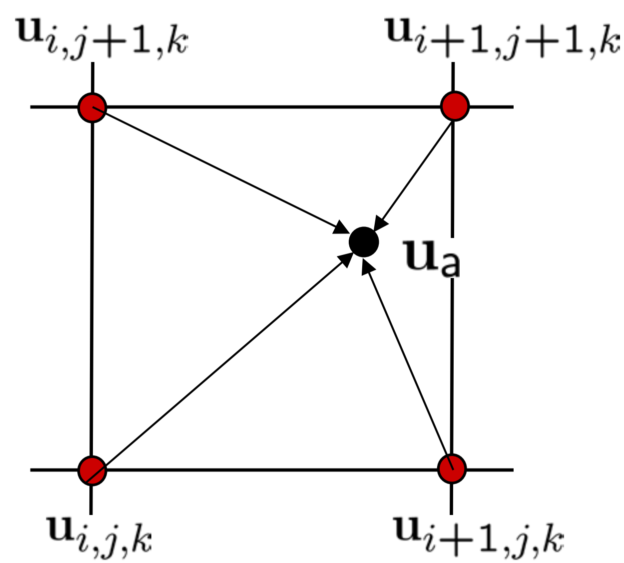


$$\mathbf{F}_a = -\mathbf{F}_{f,a}$$

$$= 3\pi\mu(\mathbf{v}_a - \mathbf{u}_a)\underbrace{F(\phi)}_{\frac{10\phi}{(1-\phi)^2} + \frac{1+1.5\,\phi^{1/2}}{(1-\phi)^{-2}}}$$

Note that $F(\phi)$ is the same for all particles that occupy the same cell.

To evaluate $\mathbf{u}_a$, again a Euler-Lagrange mapping is required.



Initial guess:

$$\widehat{P}^{n+1}_{ijk} \Rightarrow \nabla \widehat{P}^{n+1}_{ijk}$$

$$\tilde{\mathbf{u}}^{n+1}_{ijk} \Leftarrow \nabla \widehat{P}^{n+1}_{ijk}$$

$$\tilde{\mathbf{u}}_m \Leftarrow \tilde{\mathbf{u}}^{n+1}_{ijk}$$

$$\mathbf{F}_m = K\,(\mathbf{w}_m - \tilde{\mathbf{u}}_m)$$

$$\mathbf{f}_{ijk} \Leftarrow \mathbf{F}_m$$

$$\hat{\mathbf{u}}^{n+1}_{ijk} \Leftarrow \mathbf{f}_{ijk}$$

$$\nabla \cdot \hat{\mathbf{u}}^{n+1}_{ijk} = \frac{\delta t}{\rho} \nabla^2 \Phi$$

Solid phase:

$$\mathbf{R}_a(t)\ ,\mathbf{V}_a(t)$$

$\mathbf{r}_m$ on surface of a

$$\mathbf{w}_m \quad \forall m \in a$$

$$\boxed{\mathbf{F}_{f,a} = -\sum_{m \in a} \mathbf{F}_m}$$

Output: drag correlation F

$$\mathbf{F}^{\text{coll.}}_{ab}$$

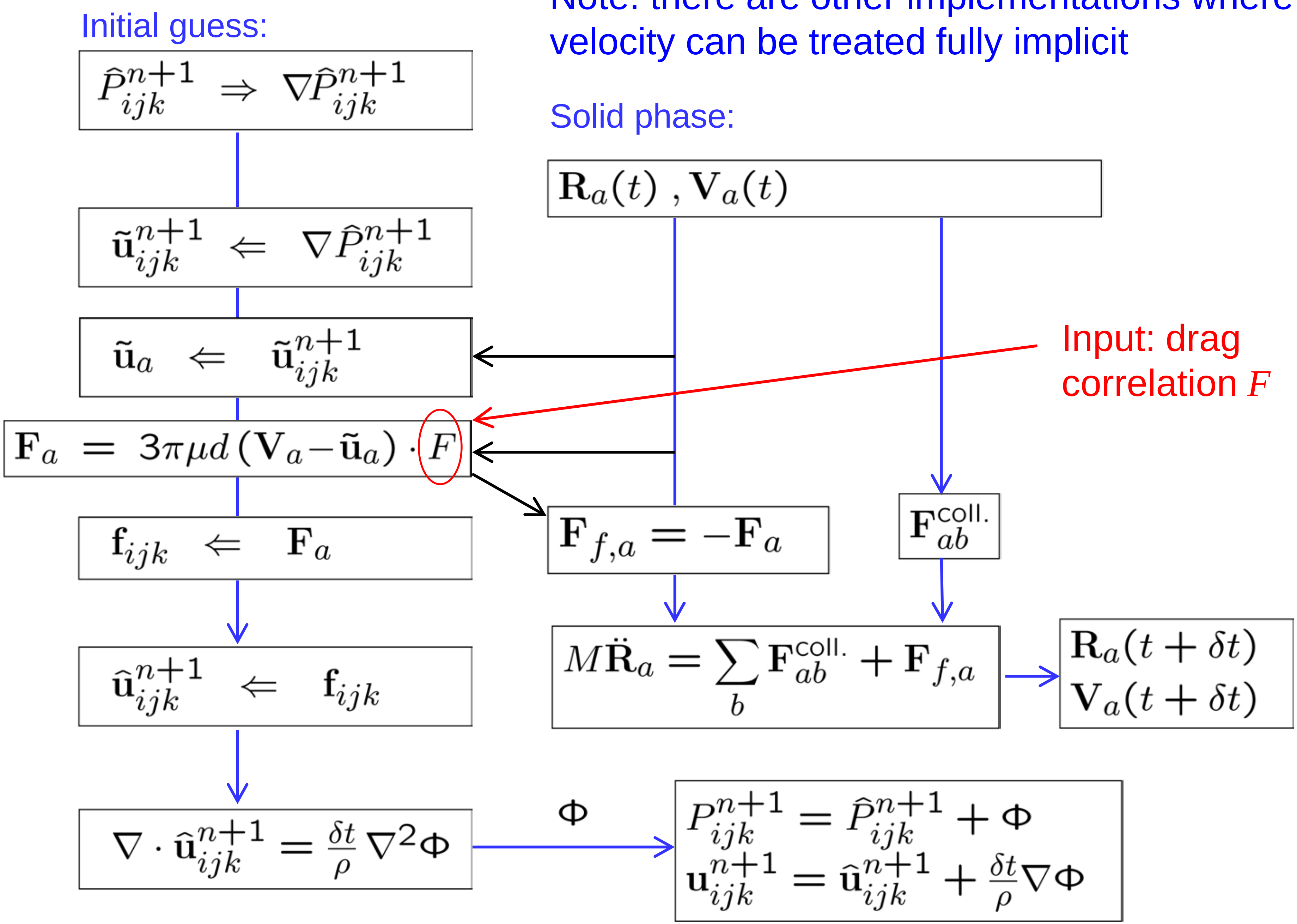
$$M\ddot{\mathbf{R}}_a = \sum_b \mathbf{F}^{\text{coll.}}_{ab} + \mathbf{F}_{f,a}$$

$$\mathbf{R}_a(t+\delta t)$$
$$\mathbf{V}_a(t+\delta t)$$

$$\xrightarrow{\Phi}$$

$$P^{n+1}_{ijk} = \widehat{P}^{n+1}_{ijk} + \Phi$$
$$\mathbf{u}^{n+1}_{ijk} = \hat{\mathbf{u}}^{n+1}_{ijk} + \frac{\delta t}{\rho} \nabla \Phi$$

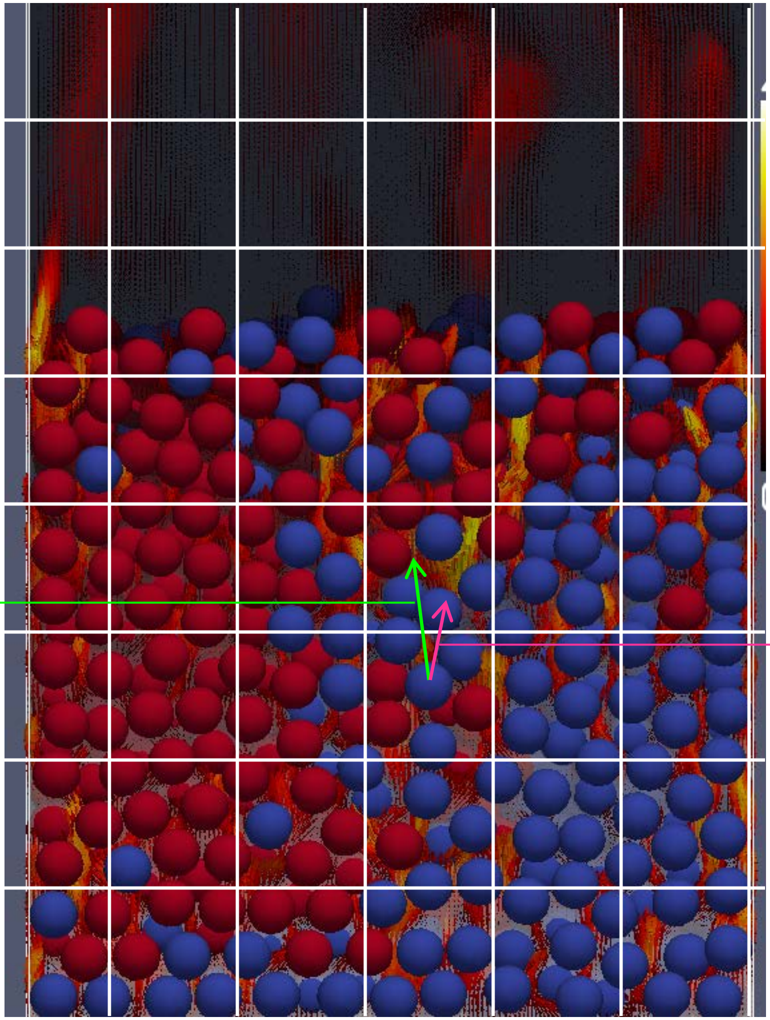
Note: there are other implementations where velocity can be treated fully implicit



DEM models for granular two-phase flow

Comparison DEM/DNS

Deviation ΔF of true force from DP force

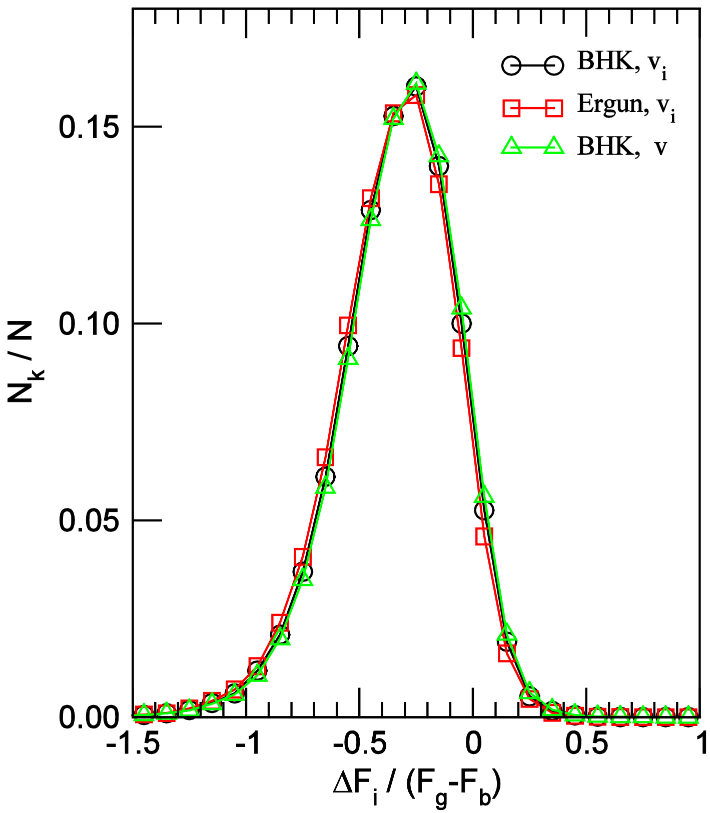


True force

Force as calculated in a DEM model from drag correlations

DEM models for granular two-phase flow

Comparison DEM/DNS



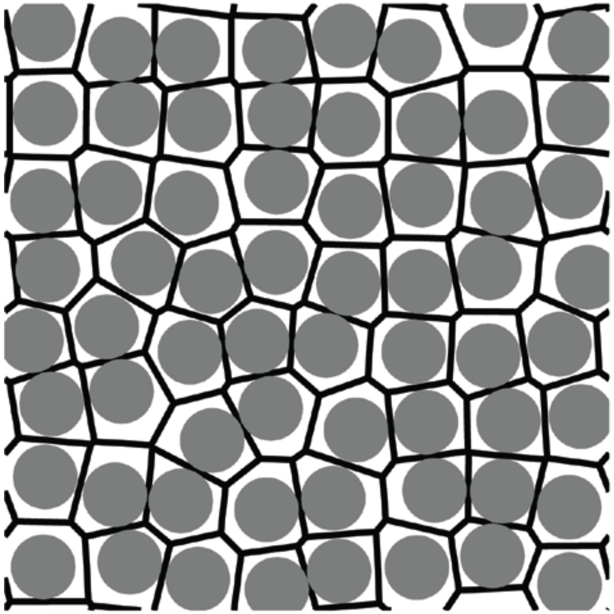
1.DEM drag is 33% lower than the average true drag force

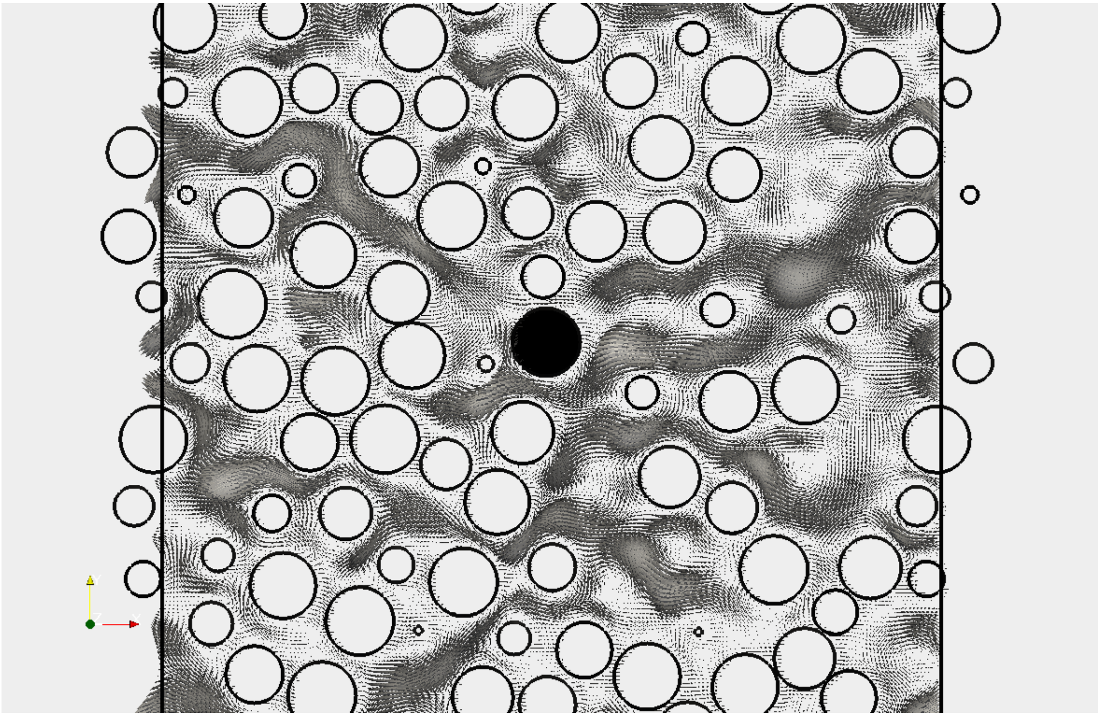
2. For 1 out of every 3 of the particles the individual force for deviates more than 25% with the DEM force.

DEM models for granular two-phase flow

Idea: use **individual** volume fractions ϕ_i

$$\vec{F}_{g,i} = 3\pi\mu d(\vec{u} - \vec{v}_i) \cdot F(\phi_i, \mathrm{Re})$$





DEM models for granular two-phase flow

Drag correlations derived for static system are not applicable to moving particles

Natural spreading in the fluid-particle drag is too large to capture with (advanced) drag models in DEM

Discrete element models are useful for obtaining insight, but should not be used for qualitative results.

III. Example: Effect of air on vibrated granular beds

Vibrated glass-bronze beds

E<u>qual-sized</u> bronze and glass spheres (100 μm)

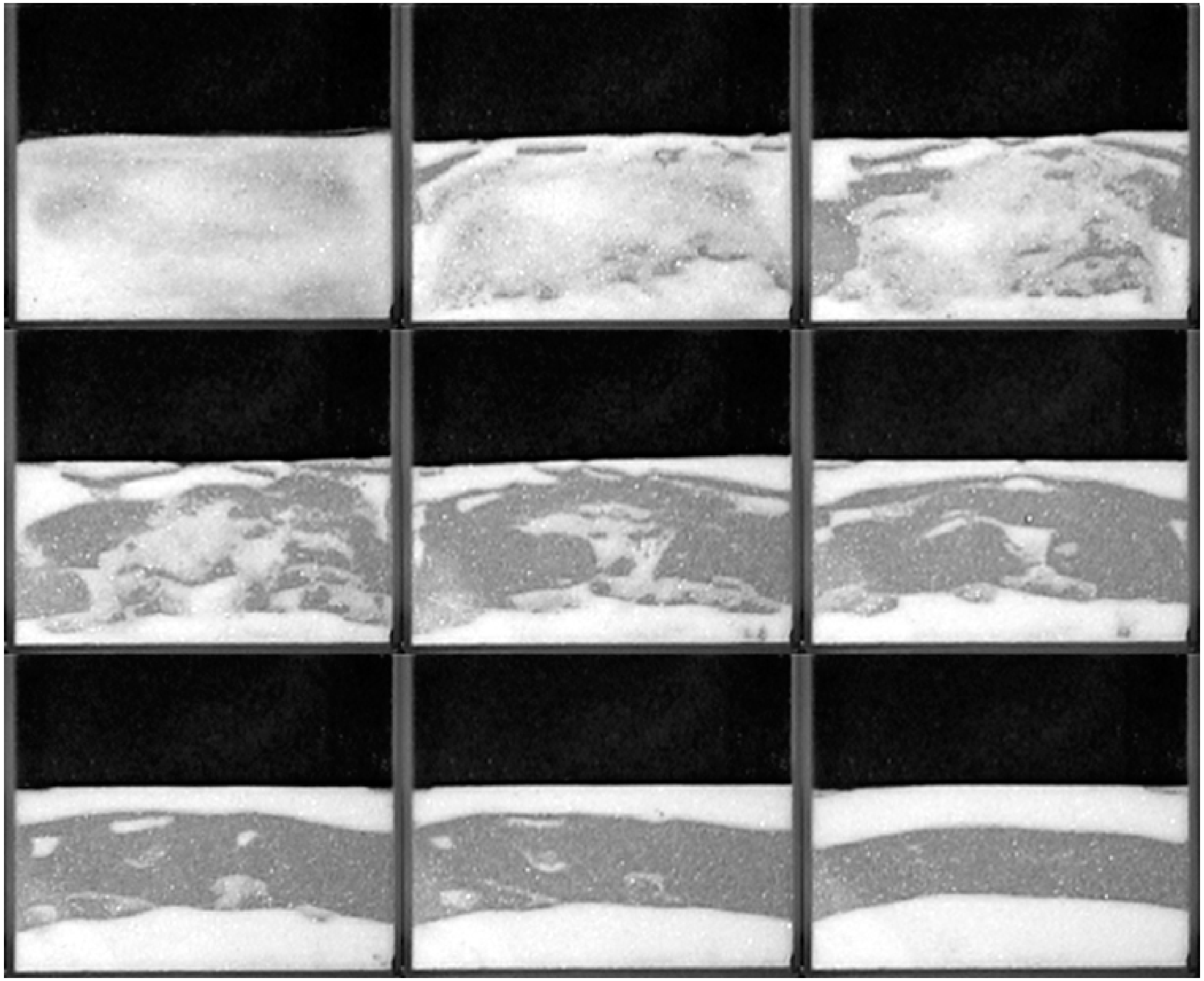
Experiments by Burtally, King and Swift (Science 2002)

Simulations:

- Particles: “molecular dynamics” with soft-sphere model
- Gas phase: computational fluid dynamics model
- Gas-Particle interactions: unresolved, empirical drag force
- System size: N_p = 25 000, W x H x D = 8 x 6 x 0.6 mm^3
- Parameters: f = 55 Hz, A = 1 mm $\Rightarrow \Gamma = \frac{A(2\pi f)^2}{g} = 12$

No air **Air**

Vibrated glass-bronze beds



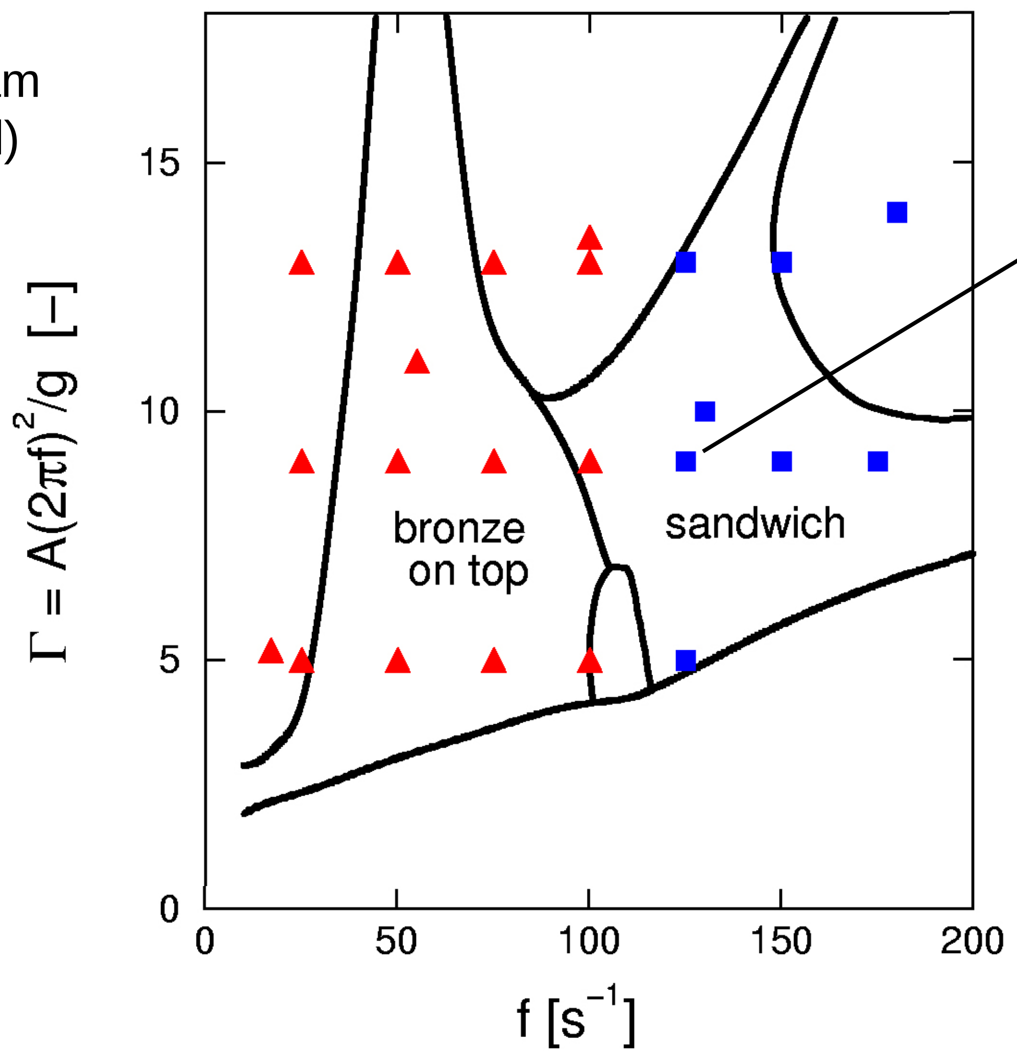
Burtally, King, Swift & Leaper, Gran.Mat. 2003

f = 50 Hz
f = 130 Hz

A = 1 mm
A = 0.07 mm

Vibrated glass-bronze beds

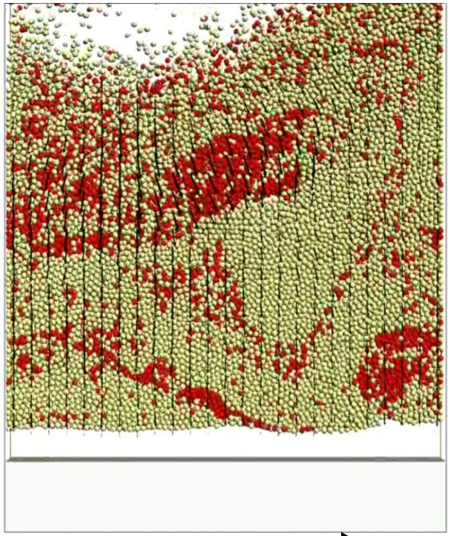
Experimental
phase diagram
(Burtally et al)

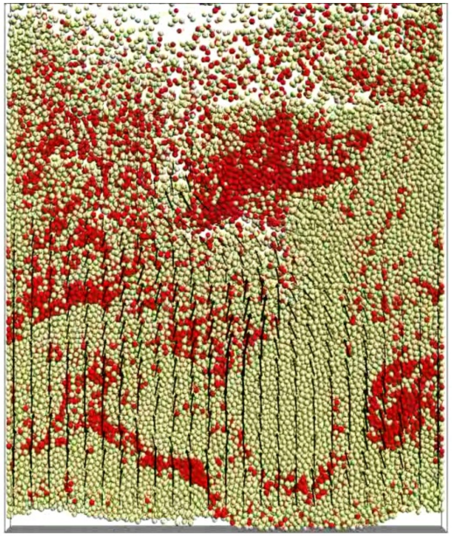


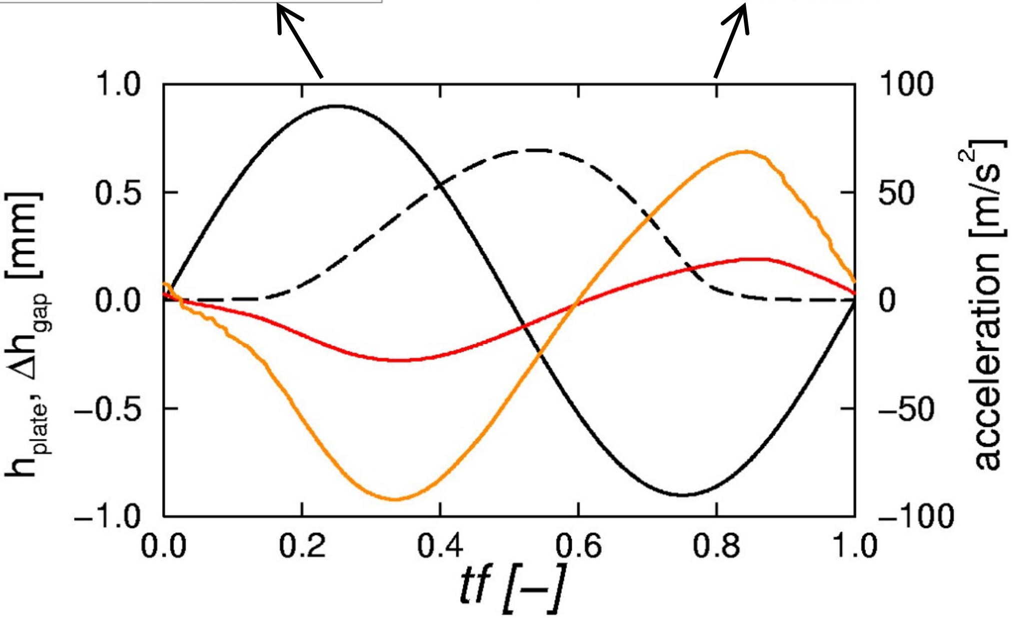
$f = 125$ Hz
$\Gamma = 9$

Vibrated glass-bronze beds

Why do the light particles sink to the bottom?







Vibrated glass-bronze beds

Sandwich formation:

- Convection plays an important role

- Sensitive of the particle-particle and partice wall friction

$\mu_{pp} \downarrow$ $\mu_{pw} \rightarrow$	0.0	0.1	0.4
0.0	Bottom	Bottom	Bottom
0.1	Middle (40 s)	**Middle** (10 s)	Top
0.4	Middle (20 s)	Middle (10 s)	Top

Concluding remarks

DEM simulations are a “cheap” way to incorporate the effect of air, however, they should only be used for getting qualitative insight

Almost all DNS simulations suffer from (large) grid resolution effects. The use of an effective diameter is essential for getting accurate results

Key features of the model:

With: Devaraj van der Meer
Ko van der Weele
Gabriel Caballero

- Particles: soft-sphere model, 0.5 mm diameter
- Gas phase: computational fluid dynamics model
- Gas-Particle interactions: unresolved, empirical drag drag
- System size: N_p = 14 000, W x H x D = 100 x 50 x 2.1 mm^3
- Parameters: f = 6.25 Hz, A = 10 mm $\Rightarrow \quad \Gamma = \frac{A(2\pi f)^2}{g} = 1.6$

No air **Air**

First documented by Da Vinci (1500) and Faraday (1831)

Mechanism for steady state heap