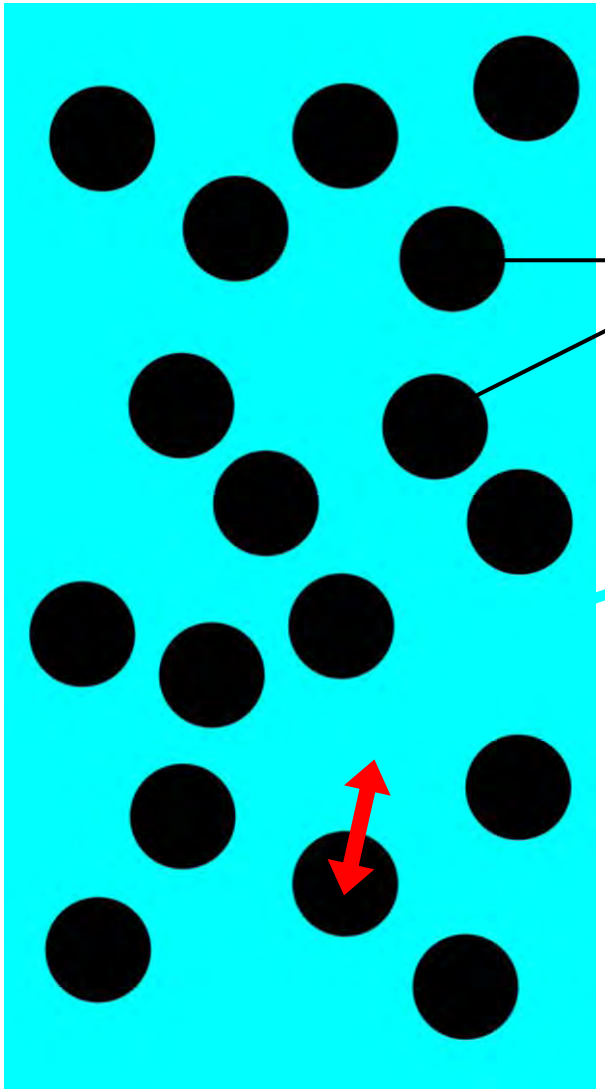
Eulerian methods for modeling gas flow in dense particle systems

Martin van der Hoef

Chemical Reaction Engineering
Department of Science & Technology
University of Twente

Modeling gas flow in granular systems



Three questions:

1) How model the particles?

   Molecular dynamics
   using soft-sphere model

2) How model the gas phase?

   First part of this lecture

3) How model the gas-particle interaction?

   Second part of this lecture

Outline

1. Models for single phase gas flow

A. Lagrangian Models
B. Eulerian models

2. Modeling gas-particle interaction

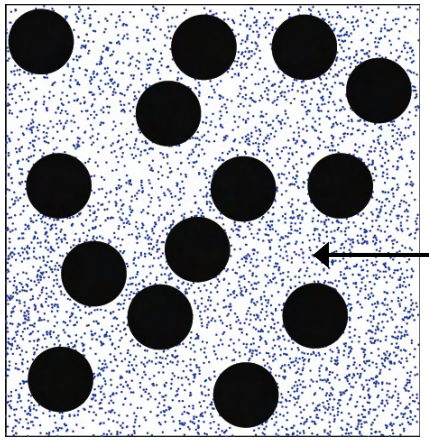
A. Fully resolved (DNS)
B. Unresolved (DEM)

3. Applications to Granular Matter research

A. Vibrated beds of bronze and glass spheres
B. Vibrated shallow beds

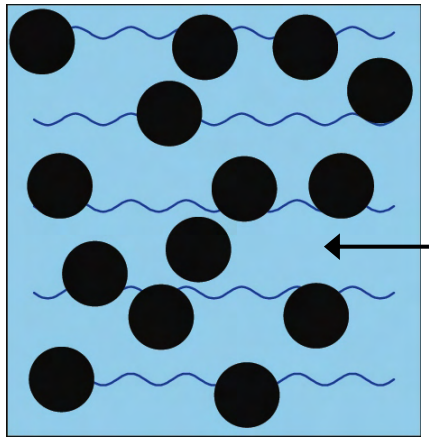
1. Models for single phase gas flow

A. Lagrangian models



- Molecular Dynamics
- Stochastic Rotation Dynamics
- Dissipative Particle Dynamics
- Lattice-Gas Cellular Automata

B. Eulerian models



- Lattice Boltzmann Model
- Computational Fluid Dynamics

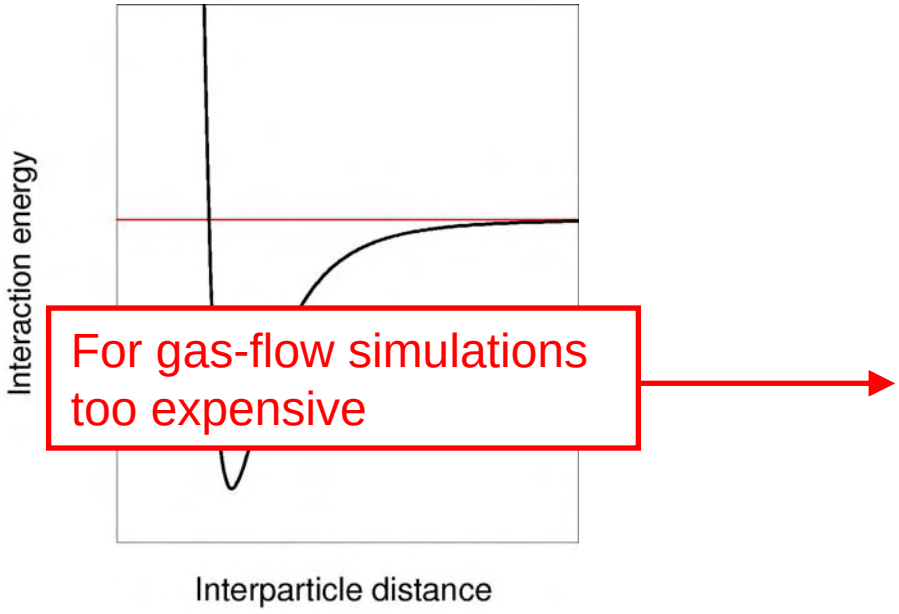
A. Lagrangian Models: Molecular Dynamics

Newton's equation of motion: $$m_i \frac{d^2\vec{r}_i}{dt^2} = \vec{F}_{i,tot}$$

is integrated numerically: $$\vec{r}_i(t+dt) = 2\vec{r}_i(t) - \vec{r}_i(t-dt) + \frac{\vec{F}_{i,tot}(t)}{m}\,\delta t^2$$

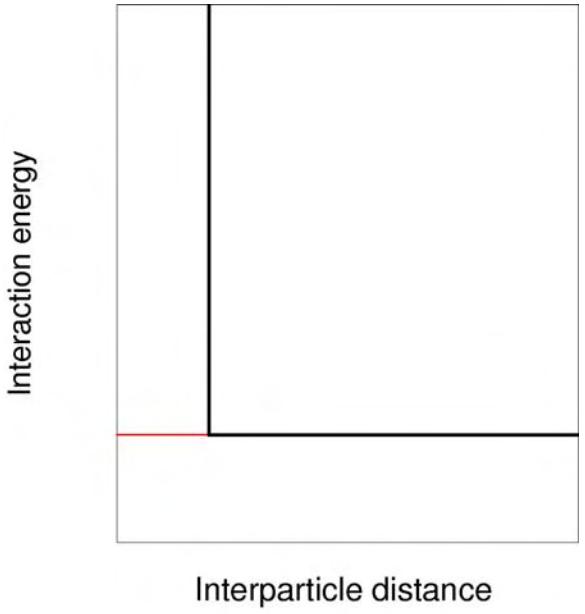
total force: $$\vec{F}_{i,tot} = \sum_j \vec{F}_{ij}^{inter} + m_i\vec{g}$$

Lennard-Jones potential



For gas-flow simulations too expensive

hard-sphere potential



A. Lagrangian Models: Molecular Dynamics

Hard-sphere model:

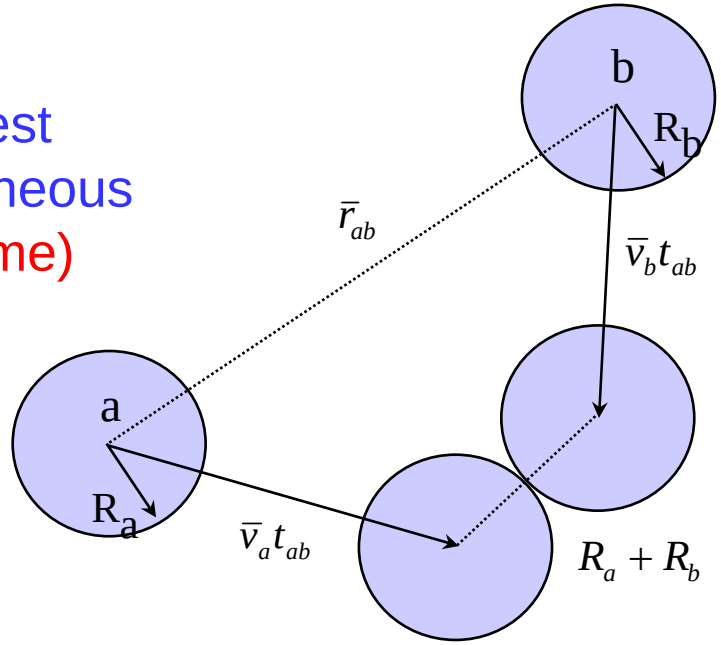
- Collision time between spheres can be calculated analytically:

$$t_{ab} = \frac{-\mathbf{r}_{ab}\cdot\mathbf{v}_{ab} - \sqrt{(\mathbf{r}_{ab}\cdot\mathbf{v}_{ab})^2 - v_{ab}^2\left[r_{ab}^2 - 4R^2\right]}}{v_{ab}^2}$$

- 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:

$$\Delta\mathbf{v}_a = \frac{\mathbf{r}_{ab}\cdot\mathbf{v}_{ab}}{4R^2}\,\mathbf{r}_{ab}$$



A. Lagrangian Models: Simplified MD models

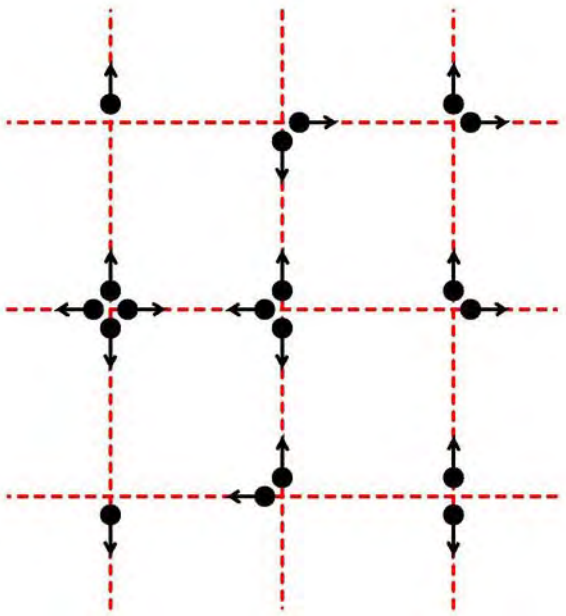
- Dissipative Particle Dynamics (DPD)
- Stochastic Rotation Dynamics (SRD)

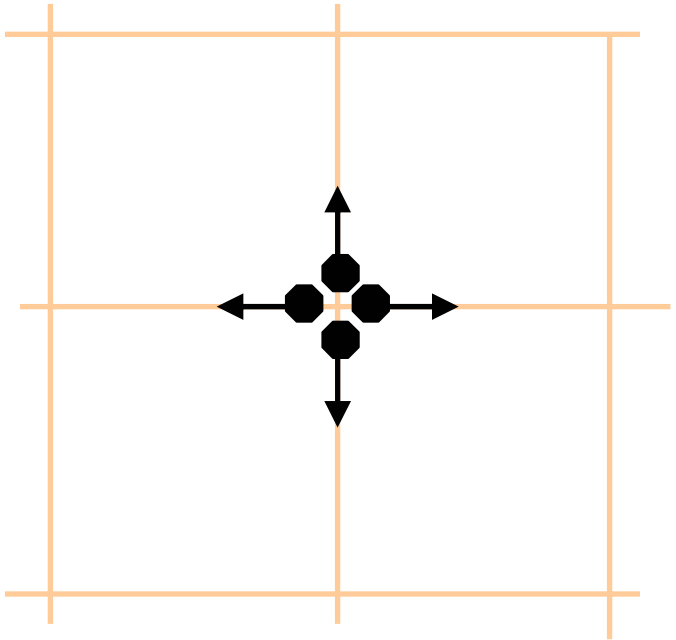
- Lattice-Gas Cellular Automata (LGCA)

A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

"be wise, discretise!"

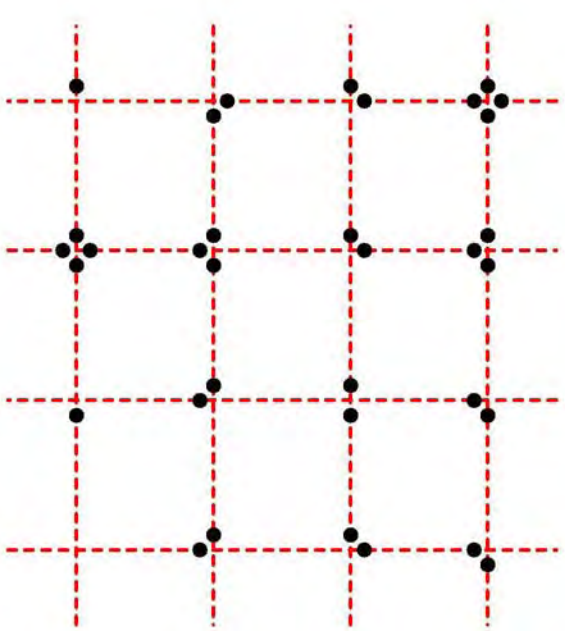
- Positions are restricted to lattice sites → discrete velocities
- No two particles with the same velocity allowed at one site
- Update: propagation and collision

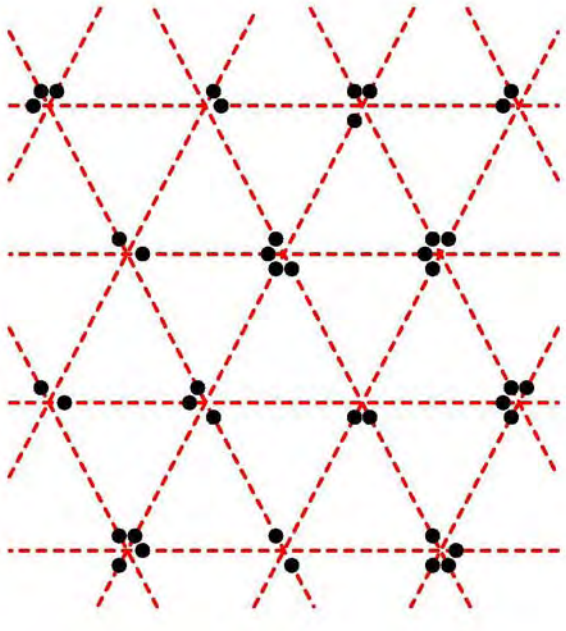




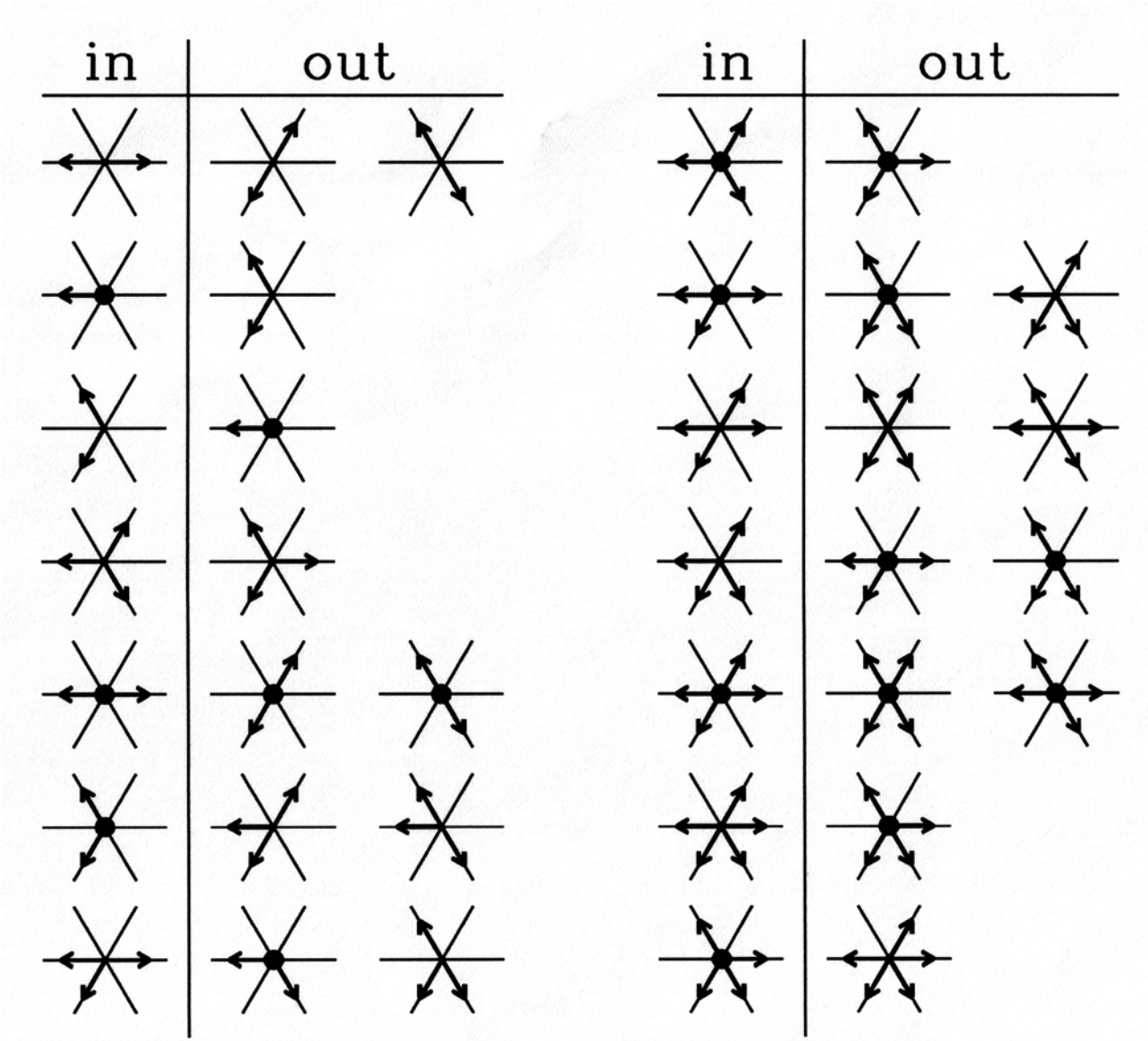
A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

Breakthrough: from square (1973) to hexagonal (1986)





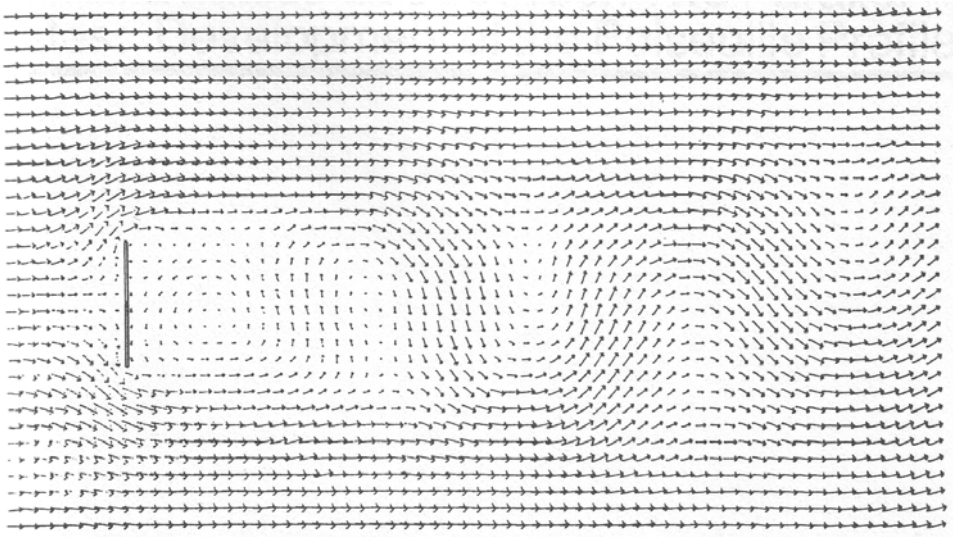
Frisch, Hasslacher & Pomeau, PRL 1986


in
out
in
out

A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

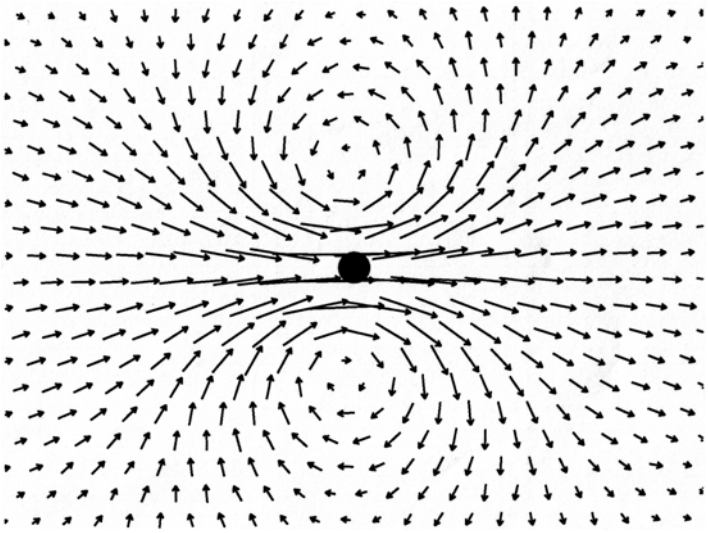
Flow past a plate

(Frisch, Hasslacher, Pomeau, PRL 1986)



Flow past a disc

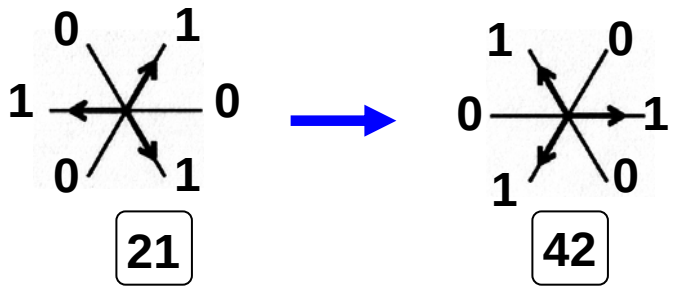
(Van der Hoef, Frenkel, & Ladd, PRL 1991)



A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

Key aspect of LGCA:

All bit manipulation
no floating point operations



Advantages:

- Extremely simple to program
- Fast
- Exact, no round-off errors
- Inherently stable

Disadvantages:

- Noisy
- Viscosity set by collision table (cannot be tuned)

A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

From lattice-gas to Navier-Stokes

Define velocity set $\{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_b\}$

Microstate of the system given by

the occupation number $n_i(\vec{r}, t) = \begin{cases} 1 & \text{if i-th link occupied} \\ 0 & \text{if i-th link empty} \end{cases}$

Local densities: $\hat{\rho}(\vec{r}, t) = \sum_i^b n_i(\vec{r}, t)$, $\hat{\rho}(\vec{r}, t)\, \hat{\vec{u}}(\vec{r}, t) = \sum_i^b \vec{c}_i\, n_i(\vec{r}, t)$

Collision rules defined by $\xi(n, n') = \begin{cases} 1 & \text{if } n \longrightarrow n' \\ 0 & \text{otherwise} \end{cases}$

Equation of motion: $n_i(\vec{r} + \vec{c}_i, t+1) = n_i(\vec{r}, t) + \Delta_i(n(\vec{r}, t))$

$$\sum_i \Delta_i(n) = 0 \,, \quad \sum_i \vec{c}_i \Delta_i(n) = 0 \qquad \overbrace{\sum_{s,s'} (s'_i - s_i)\xi(s, s') \prod_j n_j^{s_j} (1 - n_j)^{1 - s_j}}^{\Delta_i(n(\vec{r}, t))}$$

A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

Microscopic $\longrightarrow \langle \cdots \rangle \longrightarrow$ **Macroscopic**

$$n_i(\vec{r},t) \qquad\qquad f_i(\vec{r},t)$$

$$\hat{\rho}(\vec{r},t) = \sum_i n_i(\vec{r},t) \qquad\qquad \rho(\vec{r},t) = \sum_i f_i(\vec{r},t)$$

$$\hat{\rho}(\vec{r},t)\,\hat{\vec{u}}(\vec{r},t) = \sum_i \vec{c}_i\, n_i(\vec{r},t) \qquad\qquad \rho(\vec{r},t)\,\vec{u}(\vec{r},t) = \sum_i \vec{c}_i\, f_i(\vec{r},t)$$

$$\Delta_i(n) \qquad\qquad \Delta_i(f)$$

$$n_i(\vec{r}+\vec{c}_i,t+1) = n_i(\vec{r},t) + \Delta_i(n) \qquad\qquad f_i(\vec{r}+\vec{c}_i,t+1) = f_i(\vec{r},t) + \Delta_i(f)$$

$$\sum_i \Delta_i(n) = 0\,, \quad \sum_i \vec{c}_i \Delta_i(n) = 0 \qquad\qquad \sum_i \Delta_i(f) = 0\,, \quad \sum_i \vec{c}_i \Delta_i(f) = 0$$

A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

Equation of motion for f: $f_i(\vec{r}+\vec{c}_i, t+1) - f_i(\vec{r}, t) = \Delta_i(f)$

Taylor expansion (1st order)

$$\partial_t f_i + \vec{\nabla}\cdot\vec{c}_i f_i = \Delta_i(f)$$

$\sum_i$ $\qquad$ $\sum_i \vec{c}_i$

$$\partial_t \underbrace{\sum_i f_i}_{\rho} + \vec{\nabla}\cdot\underbrace{\sum_i \vec{c}_i f_i}_{\rho\vec{u}} = \underbrace{\sum_i \Delta_i(f)}_{0}$$

$$\partial_t \rho + \vec{\nabla}\cdot(\rho\vec{u}) = 0$$

$$\partial_t \underbrace{\sum_i \vec{c}_i f_i}_{\rho\vec{u}} + \vec{\nabla}\cdot\underbrace{\sum_i \vec{c}_i\vec{c}_i f_i}_{\bar{\bar{\Pi}}} = \underbrace{\sum_i \vec{c}_i\Delta_i(f)}_{0}$$

$$\partial_t (\rho\vec{u}) + \vec{\nabla}\cdot\bar{\bar{\Pi}} = 0$$

$$\bar{\bar{\Pi}} = \sum_i \vec{c}_i\vec{c}_i f_i = ?$$

A. Lagrangian Models: Lattice Gas Cellular Automata (LGCA)

$$f_i^{eq}(\rho,\vec{u}) = \frac{\rho}{b}\left[1 + \frac{(\vec{c}_i\cdot\vec{u})}{c_s^2} + \frac{(\vec{c}_i\cdot\vec{u})^2}{c_s^4} + \ldots.\right]$$

$$\underbrace{f_i = f_i^{eq} + f^{neq}}$$

$$\bar{\bar{\Pi}} = \sum_i \vec{c}_i\vec{c}_i\, f_i = \sum_i \vec{c}_i\vec{c}_i\, f_i^{eq} + \sum_i \vec{c}_i\vec{c}_i\, f_i^{neq}$$

$$= c_s^2\rho \underbrace{\left[\frac{1}{bc_s^2}\sum_i \vec{c}_i\vec{c}_i\right]}_{\bar{\bar{I}}} + \underbrace{\left[\frac{1}{bc_s^4}\sum_i \vec{c}_i\vec{c}_i\vec{c}_i\vec{c}_i\right]}_{\bar{\bar{\bar{\bar{I}}}}} : (\rho\vec{u}\vec{u}) + \sum_i \vec{c}_i\vec{c}_i\, f_i^{neq}$$

only for hexagonal lattice !!

$$= c_s^2\rho\,\bar{\bar{I}} + \rho\vec{u}\vec{u} + \underbrace{\sum_i \vec{c}_i\vec{c}_i\, f_i^{neq}}$$

$$\bar{\bar{\tau}} = -\mu\left[(\vec{\nabla}\vec{u}) + (\vec{\nabla}\vec{u})^T - \frac{2}{3}(\vec{\nabla}\cdot\vec{u})\,\bar{\bar{I}}\right]$$

B. Eulerian Models: Computational Fluid Dynamics (LGCA)

Basic idea: solve the set of differential equations:

$$\partial_t \rho + \vec{\nabla} \cdot \rho \mathbf{u} = 0$$

$$\partial_t \rho \mathbf{u} + \vec{\nabla} \cdot \rho \mathbf{u}\mathbf{u} = -\vec{\nabla} P - \vec{\nabla} \cdot \bar{\bar{\tau}}$$

by finite difference methods (CFD).

Closures for P and $\bar{\bar{\tau}}$

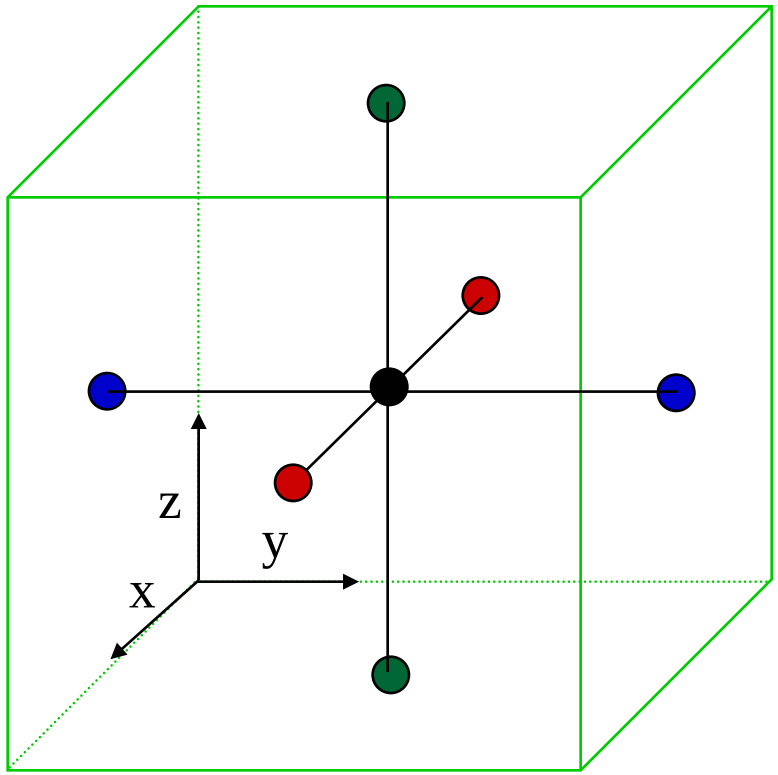
$$P = \frac{RT}{M}\rho$$

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

B. Eulerian Models: Computational Fluid Dynamics (LGCA)

Discretisation of space: define cells of volume $\delta l \cdot \delta l \cdot \delta l$

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

Notation: define variables at time n and cell $\mathbf{i} = \{i, j, k\}$ by

$$\rho_{\mathbf{i}}^{n} \qquad P_{\mathbf{i}}^{n} \qquad u_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}^{n}$$

with :

$$\mathbf{e}_x = (1,0,0)$$
$$\mathbf{e}_y = (0,1,0)$$
$$\mathbf{e}_z = (0,0,1)$$

$$h = \frac{1}{2}$$

B. Eulerian Models: Computational Fluid Dynamics (LGCA)

Finite difference of the mass continuity equation:

$$\partial_t \rho = -\vec{\nabla}\cdot\rho\mathbf{u}$$

$$\frac{\rho_{\mathbf{i}}^{n+1}-\rho_{\mathbf{i}}^{n}}{\delta t} = -\left[\vec{\nabla}\cdot\rho\mathbf{u}\right]_{\mathbf{i}}^{n+1}$$

$$\frac{\rho_{\mathbf{i}}^{n+1}-\rho_{\mathbf{i}}^{n}}{\delta t} = -\sum_{\beta}\frac{[\rho\mathbf{u}]_{\beta,\mathbf{i}+h\mathbf{e}_\beta}^{n+1}-[\rho\mathbf{u}]_{\beta,\mathbf{i}-h\mathbf{e}_\beta}^{n+1}}{\delta l}$$

$$\rho_{\mathbf{i}}^{n+1}-\rho_{\mathbf{i}}^{n} + \sum_{\beta}\frac{\delta t}{\delta l}\left([\rho\mathbf{u}]_{\beta,\mathbf{i}+h\mathbf{e}_\beta}^{n+1}-[\rho\mathbf{u}]_{\beta,\mathbf{i}-h\mathbf{e}_\beta}^{n+1}\right) = 0$$

B. Eulerian Models: Computational Fluid Dynamics (LGCA)

Finite difference of momentum continuity equation:

$$\partial_t \rho\mathbf{u} = -\vec{\nabla}P - \vec{\nabla}\cdot\rho\mathbf{u}\mathbf{u} - \vec{\nabla}\cdot\bar{\bar{\tau}}$$

$$\frac{[\rho\mathbf{u}]^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} - [\rho\mathbf{u}]^{n}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}}{\delta t} = -\left[\vec{\nabla}P\right]^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} - \left[\vec{\nabla}\cdot\rho\mathbf{u}\mathbf{u} + \vec{\nabla}\cdot\bar{\bar{\tau}}\right]^{n}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}$$

$$[\rho\mathbf{u}]^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} = -\delta t\left[\vec{\nabla}P\right]^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} \underbrace{- \delta t\left[\vec{\nabla}\cdot\rho\mathbf{u}\mathbf{u} + \vec{\nabla}\cdot\bar{\bar{\tau}}\right]^{n}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} + [\rho\mathbf{u}]^{n}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}}$$

$$[\rho\mathbf{u}]^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} = -\frac{\delta t}{\delta l}(P^{n+1}_{\mathbf{i}+\mathbf{e}_\alpha} - P^{n+1}_{\mathbf{i}}) + A^{n}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}$$

$$\mathbf{u}^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha} = \frac{-\frac{\delta t}{\delta l}(P^{n+1}_{\mathbf{i}+\mathbf{e}_\alpha} - P^{n+1}_{\mathbf{i}}) + A^{n}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}}{\rho^{n+1}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}}$$

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

1. First guess $P_{\mathbf{i}}^{n+1}$

$$\rho_{\mathbf{i}}^{n+1} = \frac{M}{RT} P_{\mathbf{i}}^{n+1}$$

$$\mathbf{u}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}^{n+1} = \frac{-\frac{\delta t}{\delta l}(P_{\mathbf{i}+\mathbf{e}_\alpha}^{n+1} - P_{\mathbf{i}}^{n+1}) + A_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}^{n}}{\rho_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}^{n+1}}$$

2. Test if $\rho_{\mathbf{i}}^{n+1}$ and $\mathbf{u}_{\alpha,\mathbf{i}+h\mathbf{e}_\alpha}^{n+1}$ satisfy conservation of mass eq.

$$\rho_{\mathbf{i}}^{n+1} - \rho_{\mathbf{i}}^{n} + \sum_{\beta} \frac{\delta t}{\delta l} \left([\rho\mathbf{u}]_{\beta,\mathbf{i}+h\mathbf{e}_\beta}^{n+1} - [\rho\mathbf{u}]_{\beta,\mathbf{i}-h\mathbf{e}_\beta}^{n+1} \right) = 0$$

no → try new $P_{\mathbf{i}}^{n+1}$ (Newton-Raphson procedure)

yes → go to $n+2$

B. Eulerian Models: Lattice Boltzmann Method (LBM)

Originate from the lattice-gas cellular automata

$$f_i(\vec{r}+\vec{c}_i, t+1) = f_i(\vec{r}, t) + \Delta_i(f)$$

$$\Delta_i(f) = \sum_{s,s'}(s'_i - s_i)\xi(s,s') \prod_j f_j^{s_j}(1-f_j)^{1-s_j}$$

3D: every update requires a double sum over $16\,10^6$ states!

Solution: linearize the collision operator about f^{eq}

$$\Delta_i(f) = \Delta_i(f^{eq}+(f-f^{eq})) = \underbrace{\Delta_i(f^{eq})}_{0} + \underbrace{\sum_{i=1}^{b}\left(\frac{\partial\Delta_i}{\partial f_j}\right)_{f_j=f_j^{eq}}}_{L_{ij}}(f_j-f_j^{eq})$$

only needs to be calculated once

B. Eulerian Models: Lattice Boltzmann Method (LBM)

Lattice Boltzmann equation (1988)

$$f_i(\vec{r}+\vec{c}_i, t+1) = f_i(\vec{r}, t) + \sum_{i=j}^{b} L_{ij}\,(f_j - f_j^{eq})$$

Lattice Boltzmann BGK equation (1992)

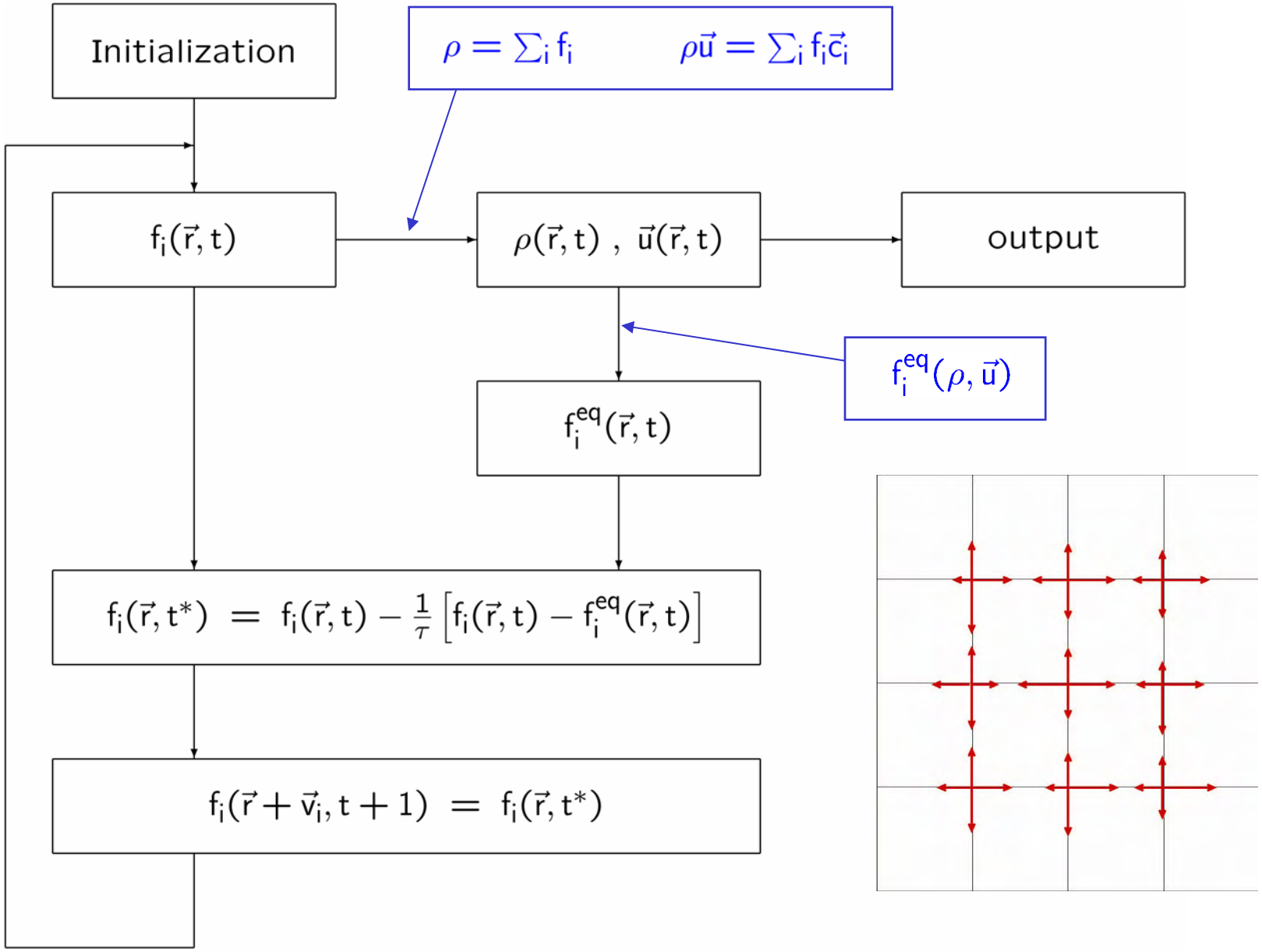
$$f_i(\vec{r}+\vec{c}_i, t+1) = f_i(\vec{r}, t) + \frac{1}{\tau}\,(f_i - f_i^{eq})$$

Note: continuous BGK equation (1954 !!)

$$\partial_t f + \vec{v}\cdot\vec{\nabla} f = \frac{1}{\tau}\,(f - f^{eq})$$

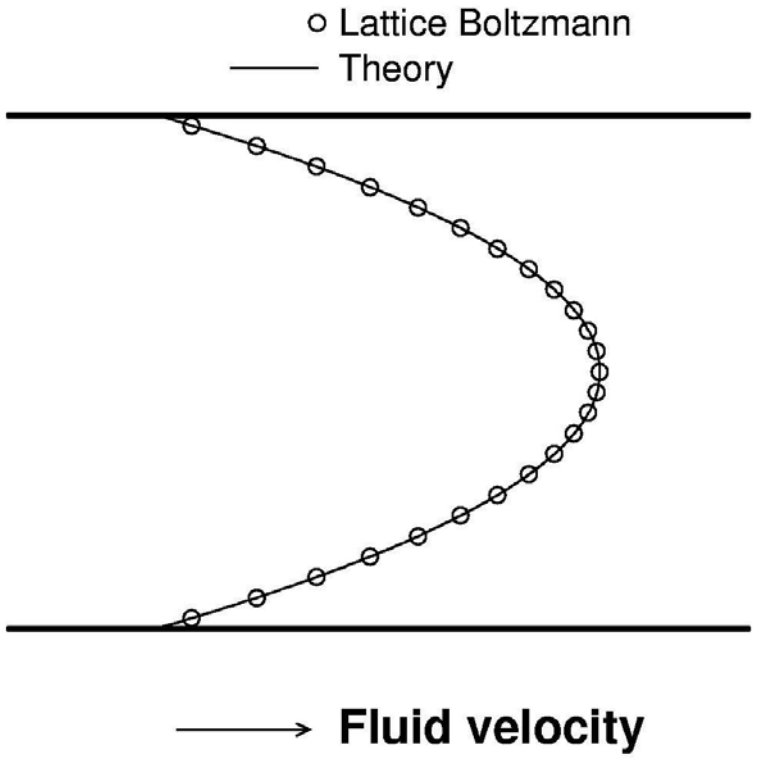
$\int d\vec{v}$ → $\partial_t \rho + \vec{\nabla}\cdot\rho\mathbf{u} = 0$

$\int \vec{v} d\vec{v}$ → $\partial_t \rho\mathbf{u} + \vec{\nabla}\cdot\rho\mathbf{u}\mathbf{u} = -\vec{\nabla}P - \vec{\nabla}\cdot\bar{\bar{\tau}}$


Initialization
$\rho = \sum_i f_i \qquad \rho\vec{u} = \sum_i f_i\vec{c}_i$
$f_i(\vec{r}, t)$
$\rho(\vec{r}, t)\ ,\ \vec{u}(\vec{r}, t)$
output
$f_i^{eq}(\rho, \vec{u})$
$f_i^{eq}(\vec{r}, t)$
$f_i(\vec{r}, t^*) = f_i(\vec{r}, t) - \frac{1}{\tau}\left[f_i(\vec{r}, t) - f_i^{eq}(\vec{r}, t)\right]$
$f_i(\vec{r} + \vec{v}_i, t + 1) = f_i(\vec{r}, t^*)$

B. Eulerian Models: Lattice Boltzmann Method (LBM)

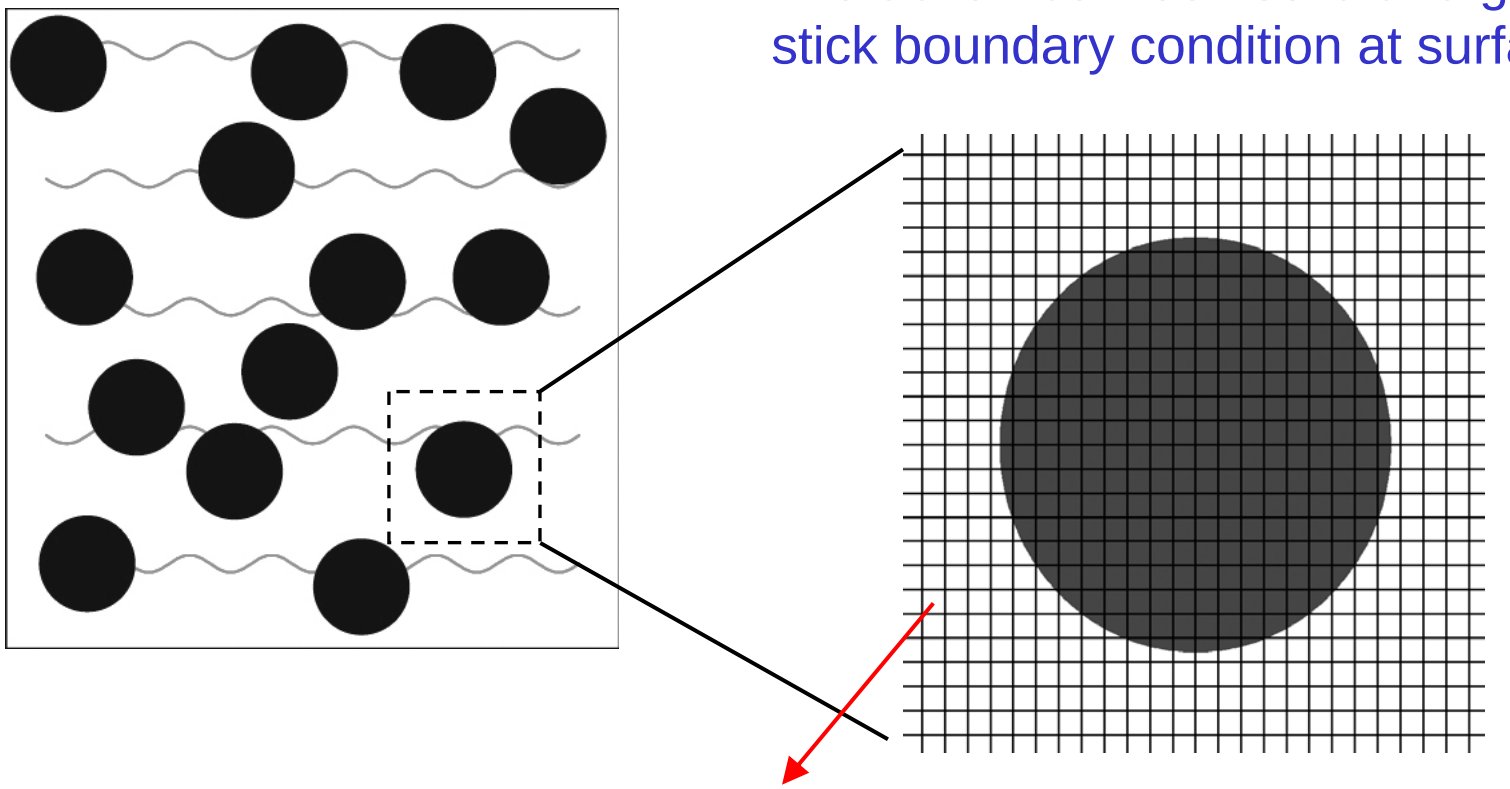
Example of single phase flow with LBM: Poiseuille flow



2. Modeling Gas-Particle Interaction

A. Resolved Discrete Particle Models

Interaction between solid and gas:
stick boundary condition at surface

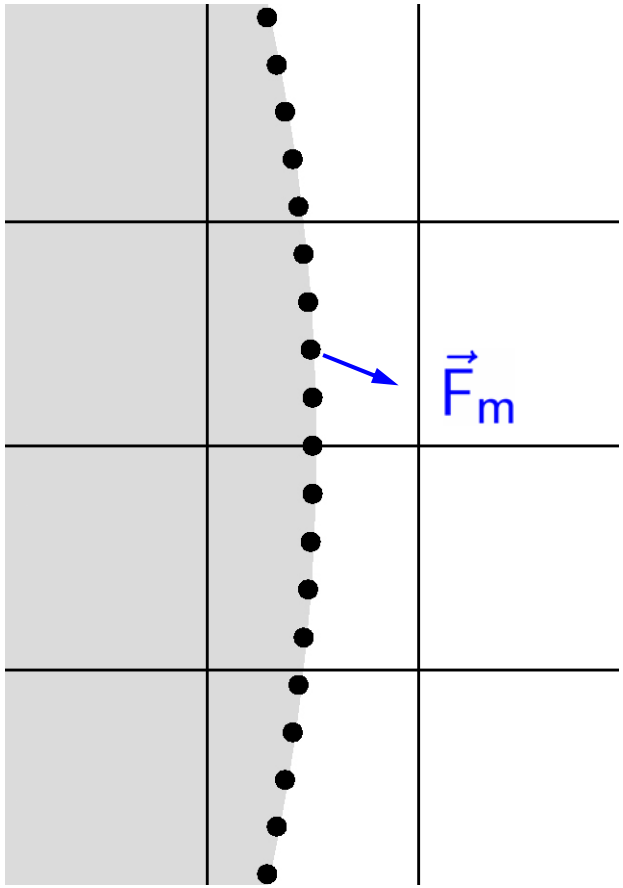


CFD or LB

A. Resolved Discrete Particle Models (CFD)

Resolved flow with CFD: Immersed Boundary Method

Define Lagrangian force points on the surface of the particle:



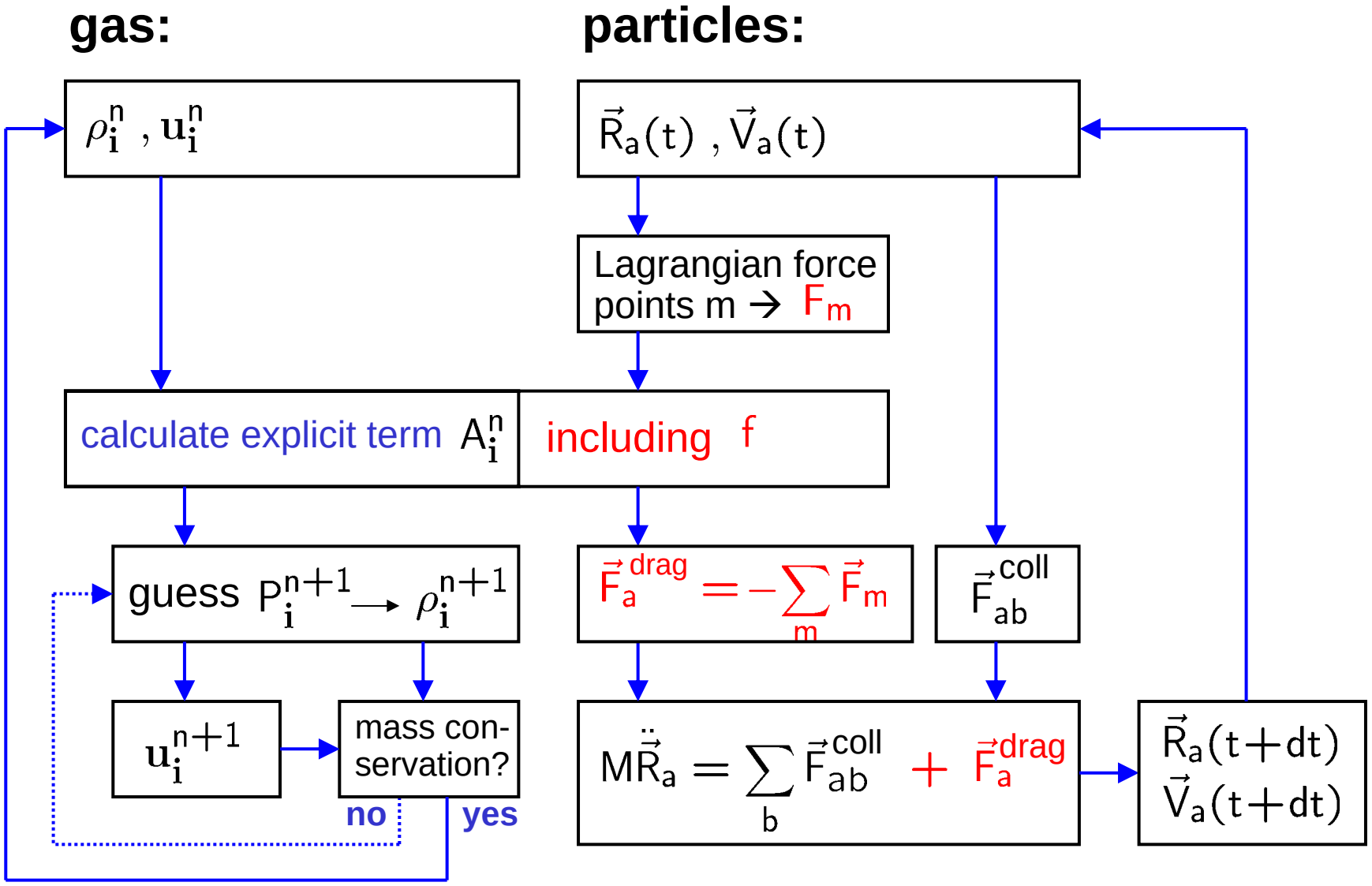
Each force point m applies a force $\vec{F}_m$ on the fluid, such that the local velocity of the fluid is equal to the local surface velocity

The sum of all force points results in a local force density $\vec{f}$ via a Lagrangian-Eulerian mapping

The momentum equation then becomes:

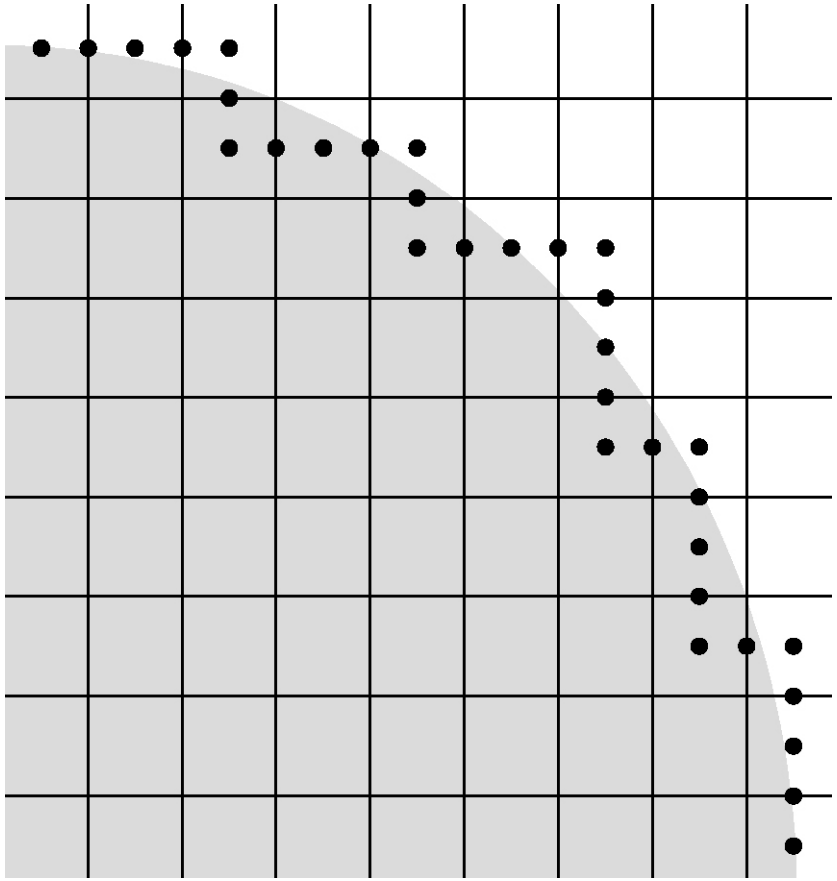
$$\partial_t \rho \vec{u} + \vec{\nabla} \cdot \rho \vec{u} \vec{u} = -\vec{\nabla} P - \vec{\nabla} \cdot \bar{\bar{\tau}} + \vec{f}$$

A. Resolved Discrete Particle Models (CFD)



A. Resolved Discrete Particle Models (LBM)

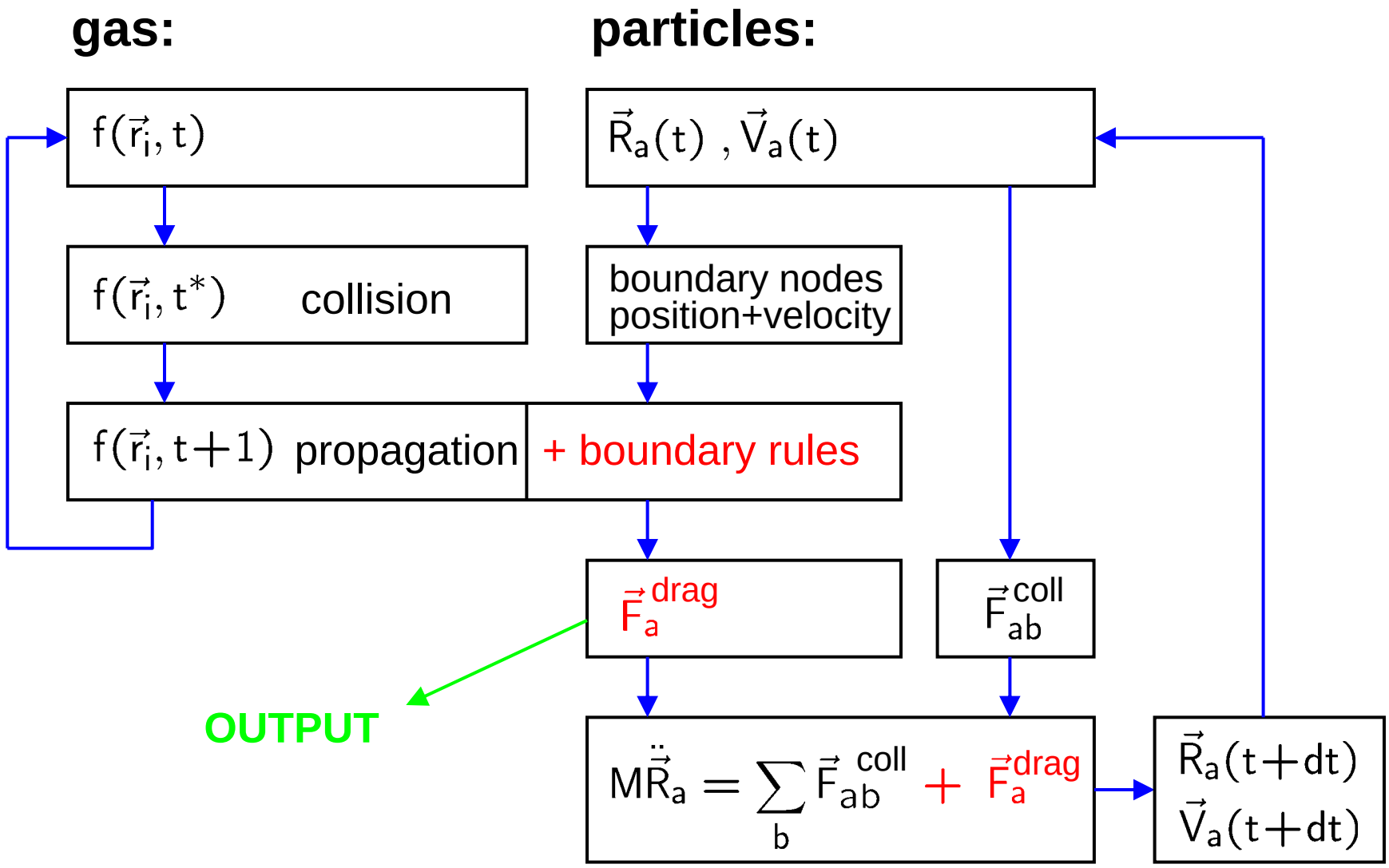
Resolved flow with LBM: Bounce Back at boundary nodes



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

A. Resolved Discrete Particle Models (LBM)



A. Resolved Discrete Particle Models

Results from resolved flow simulations

1. Fluidized bed simulation with 3600 particles using the CFD-IB method (J.A.M. Kuipers, 2008)

2. Sedimentation of 6144 particles using the LBM-BB method (A.J.C. Ladd, see http://ladd.che.ufl.edu)

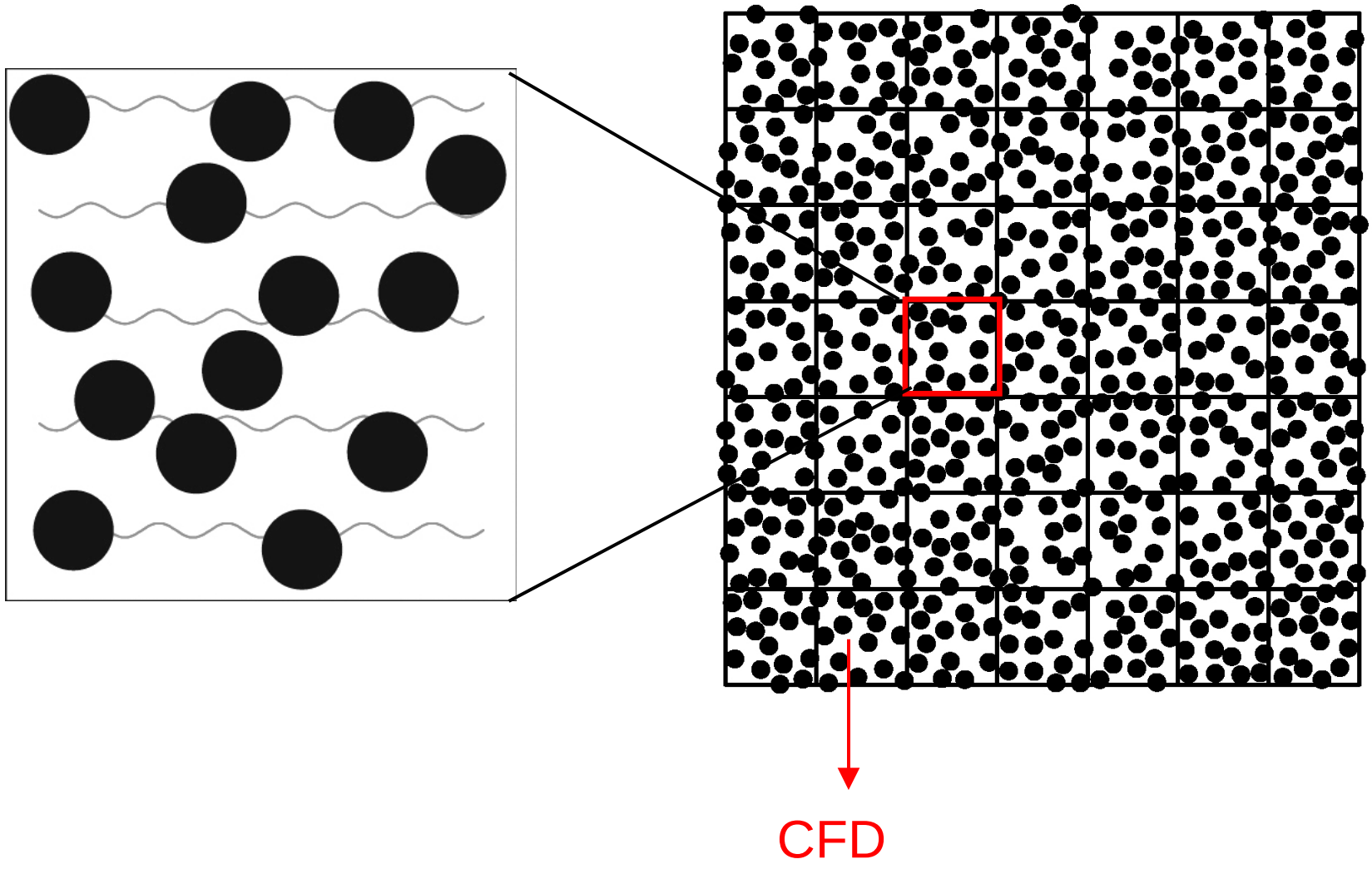
3. Measuring the gas-solid drag force using the LBM-BB method (Van der Hoef, Beetstra, Kuipers)

See next section

Conclusion

Fully resolved methods provide the most detailed level of modeling gas-particle flow, yet number of particles are limited to O(1000)

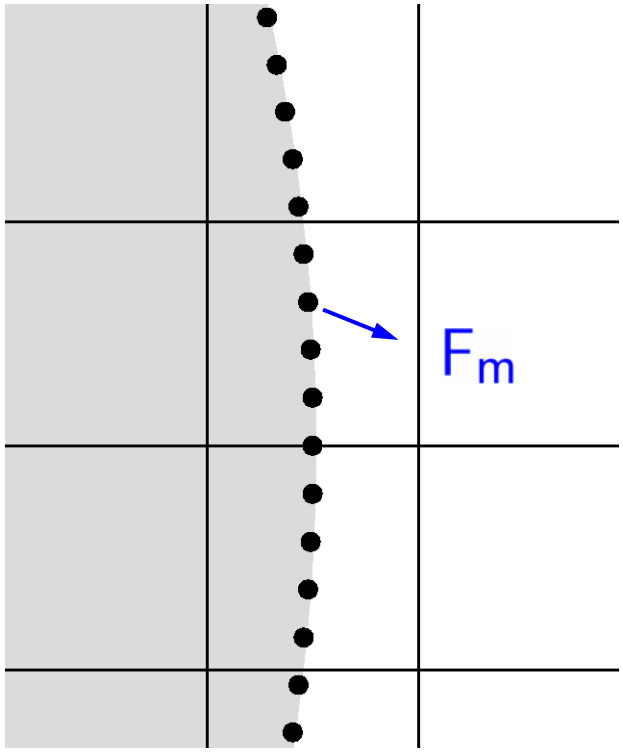
B. Unresolved Discrete Particle Models

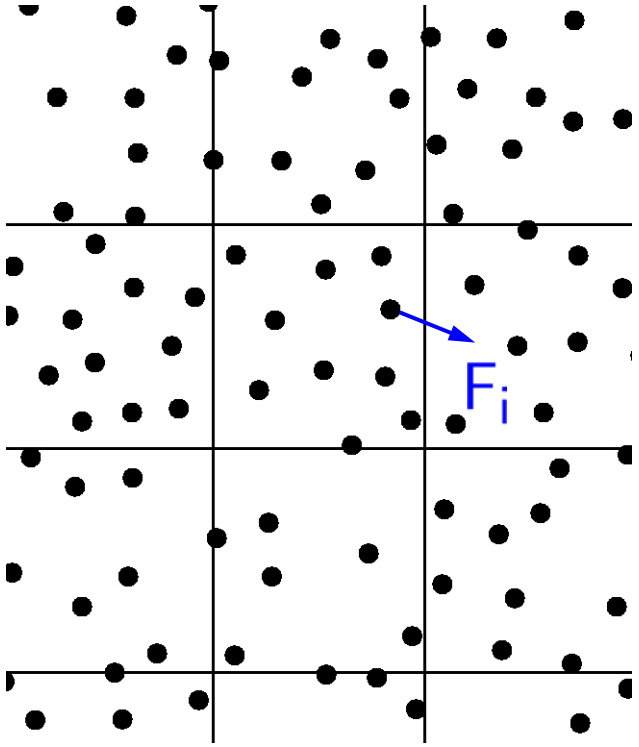


B. Unresolved Discrete Particle Models

Unresolved flow with CFD: implementation similar to resolved flow

Resolved flow





$$\partial_t \rho \mathbf{u} + \vec{\nabla} \cdot \rho \mathbf{u}\mathbf{u} = -\vec{\nabla} P - \vec{\nabla} \cdot \bar{\bar{\tau}} + \mathrm{f}$$

B. Unresolved Discrete Particle Models

$$\vec{F}_i = \frac{\beta V_i}{1-\varepsilon}(\vec{v}_i - \vec{u}) \qquad \vec{f} \equiv \beta\,(\vec{u} - \vec{v})$$

Gas phase:

$$\partial_t \varepsilon\rho\vec{u} + \vec{\nabla}\cdot\varepsilon\rho\vec{u}\vec{u} = -\vec{\nabla}P - \vec{\nabla}\cdot\,\varepsilon\bar{\bar{\tau}} + \vec{f} + (1-\varepsilon)\vec{\nabla}P$$

pressure force

Solid phase:

$$\frac{d}{dt} m\vec{v}_i = \sum_j \vec{F}_{ij} - \vec{F}_i + V_i\vec{\nabla}P$$

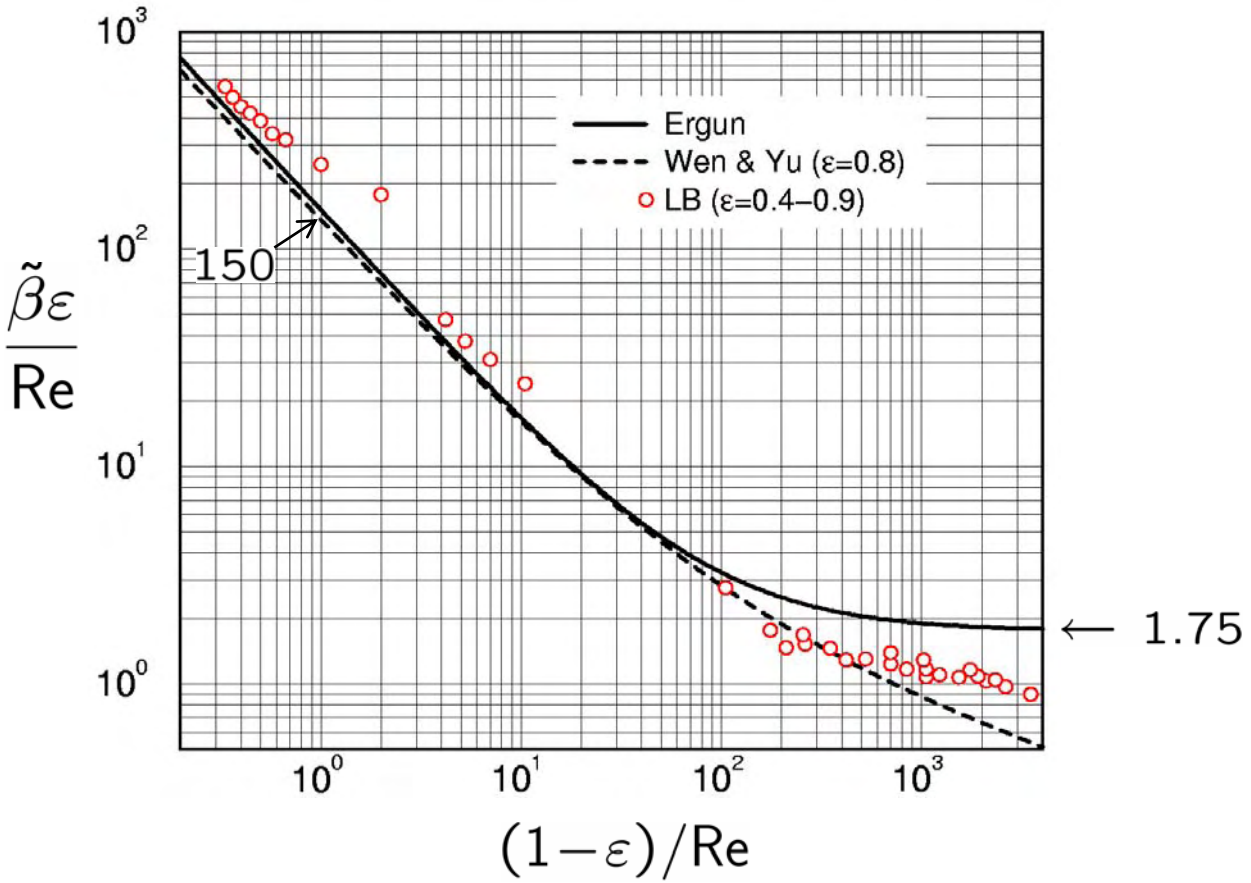
Required: correlations for β

B. Unresolved Discrete Particle Models

Monodisperse systems

$$\tilde{\beta} = \frac{\beta d^2}{\mu(1-\varepsilon)}$$

Ergun: $$\frac{\tilde{\beta}\varepsilon}{Re} = 150\left(\frac{1-\varepsilon}{Re}\right) + 1.75$$

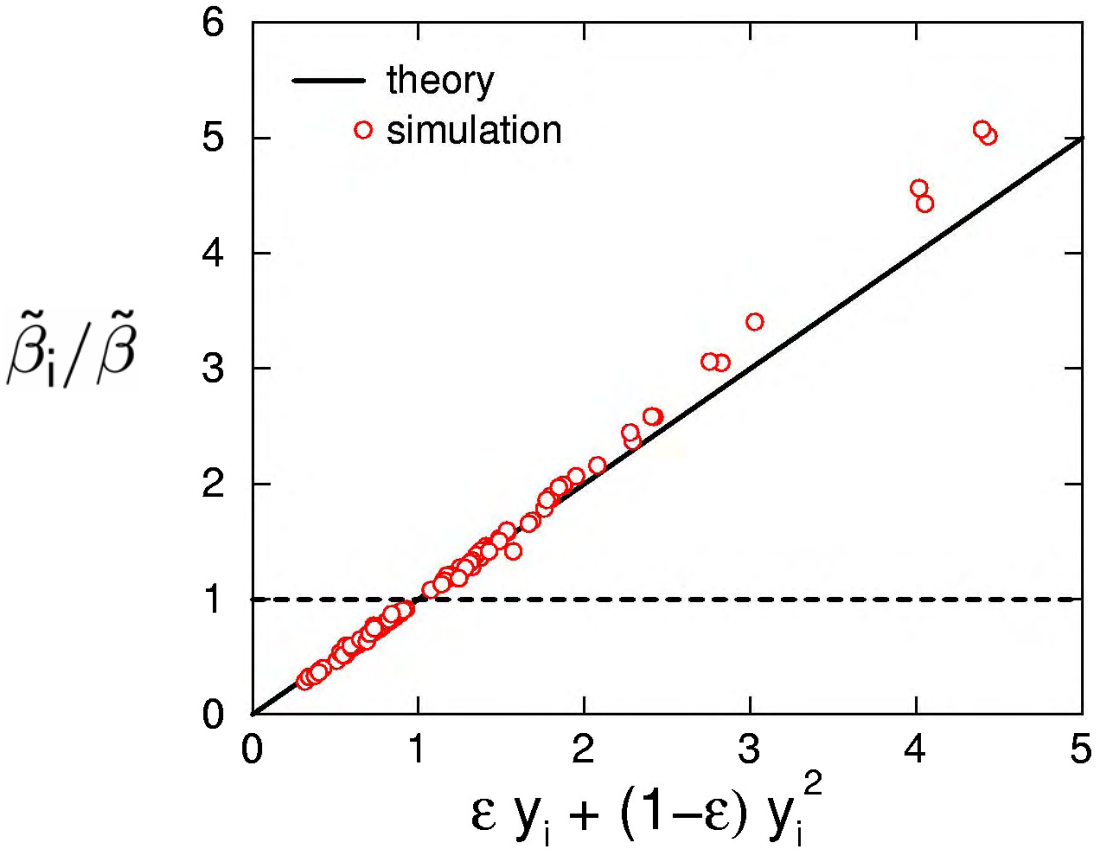


B. Unresolved Discrete Particle Models

Bidisperse systems

Currently assumed: $\tilde{\beta}_i = \tilde{\beta}$

Theory: $\tilde{\beta}_i/\tilde{\beta} = \varepsilon\, y_i + (1-\varepsilon)\, y_i^2$



$\left(y_i = \dfrac{d_i}{\langle d \rangle} \right)$

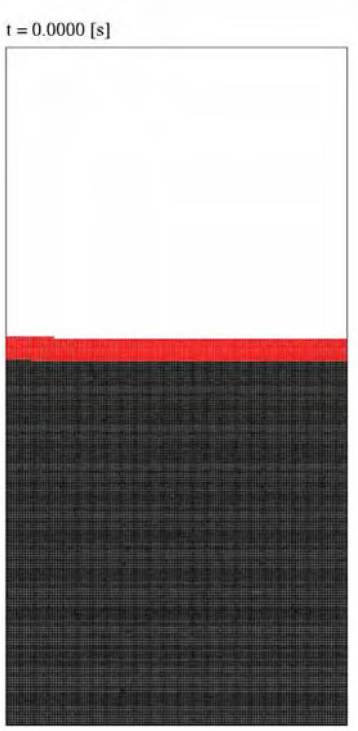
Re = 100

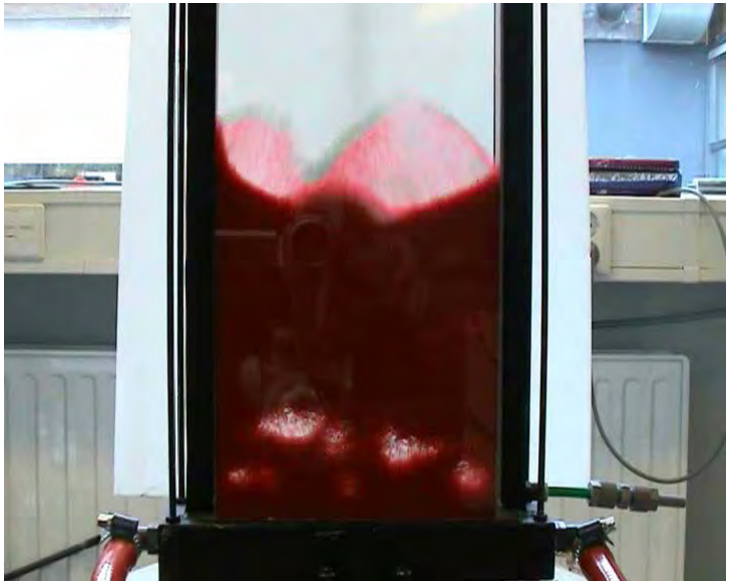
VDH, Beetstra & Kuipers,
J. Fluid Mech. 528 (2005)

B. Unresolved Discrete Particle Models

Applications of unresolved discrete particle models

I. Study of gas-fluidized beds





II. Study the effect of air in granular systems

3. Application to Granular Matter research

A. Vibrated beds of bronze and glass spheres

Christiaan Zeilstra, Hans Kuipers

B. Vibrated shallow beds

Henk-Jan van Gerner, Devaraj v.d. Meer, Ko v.d Weele

A. Density segregation of fine powders

Vibrated bed of <u>equal-sized</u> bronze and glass spheres (100 μm)

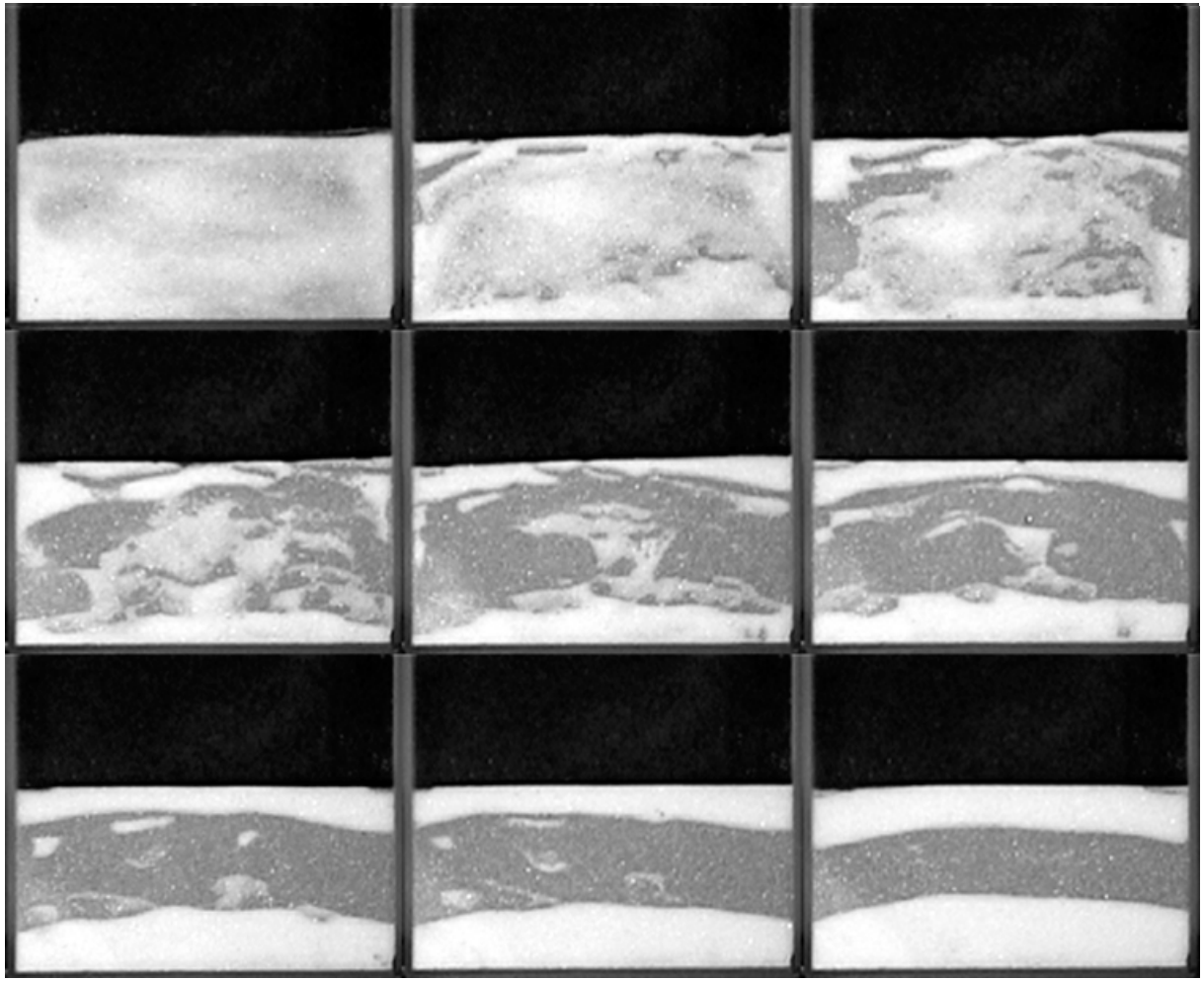
Experiments by Burtally, King and Swift (Science 2002)

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** ▶

A. Density segregation of fine powders



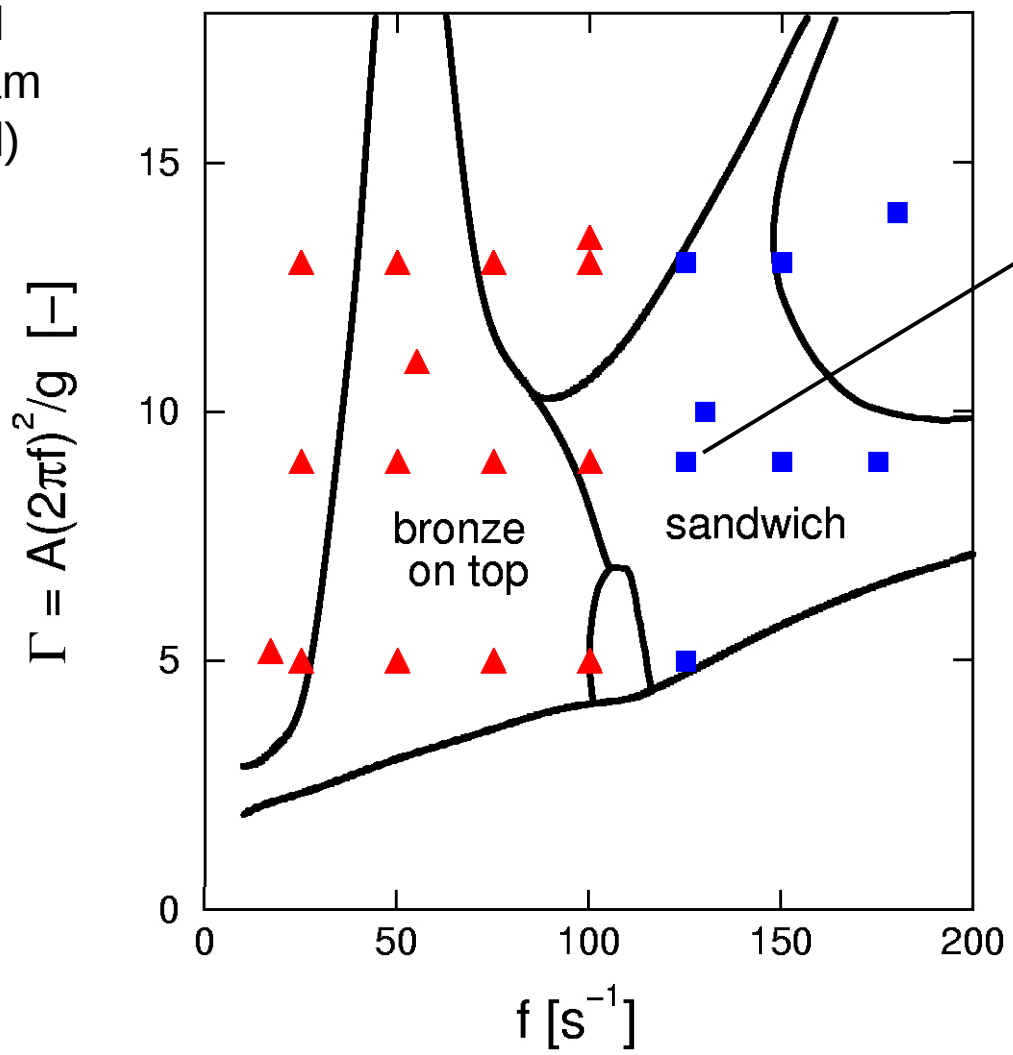
Burtally, King, Swift & Leaper, Gran.Mat. 2003

f = 50 Hz

A = 0.7 mm

A. Density segregation of fine powders

Experimental phase diagram (Burtally et al)

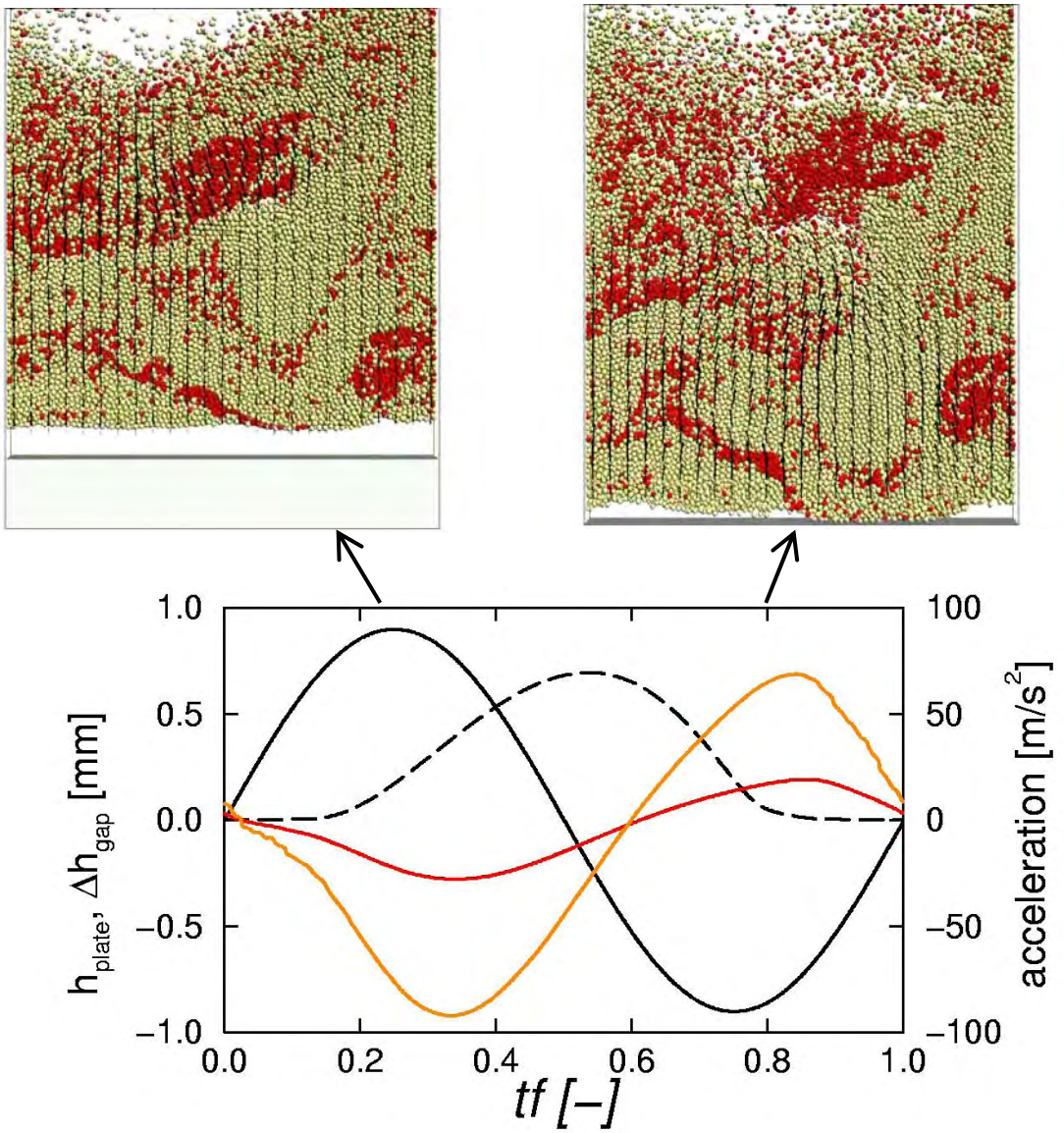


$f = 125$ Hz

$\Gamma = 9$

A. Density segregation of fine powders

Why do the light particles sink to the bottom?



A. Density segregation of fine powders

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

B. Faraday Heaping

with: **H-J van Gerner**
K. van der Weele
D. van der Meer

Simulations:

- 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 $\Rightarrow \Gamma = \frac{A(2\pi f)^2}{g} = 1.6$

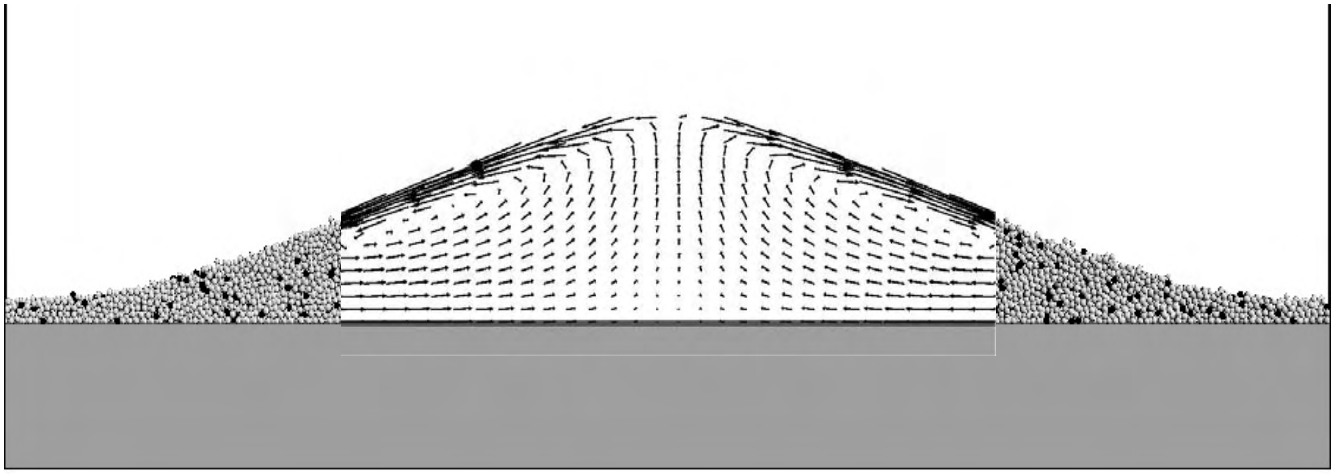
No air **Air** **Exp**

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

B. Faraday Heaping

Three rivaling mechanisms for steady state heap

1. Internal avalanche flow (Laroche et al., J. Phys., 1989)
2. Inward pressure gradient (Thomas and Squires, PRL, 1998)
3. Stability of inclined surfaces (Duran, PRL, 2000])



j

B. Faraday Heaping

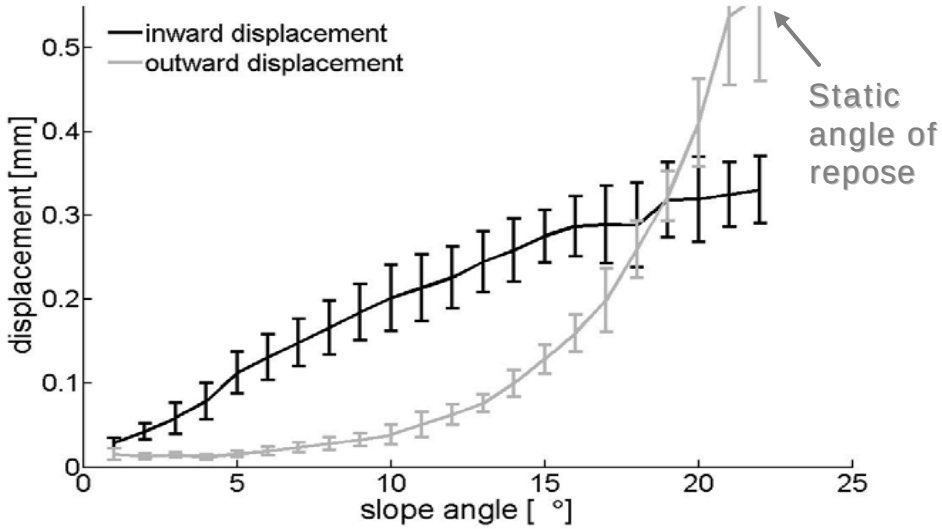
Conclusion for the steady state heap:

Our simulations confirm the pressure gradient mechanism proposed by Thomas & Squires.

How does the heap come into existence?

Two stages:

1. Fast initial stage where heaps are formed from a flat surface
2. Slow second stage where heaps merge:



Summary: overview of gas-solid models