Five lectures & five sets of lecture notes

- Kinetic theory
  - Distribution functions*
  - Boltzmann equation*
  - Transport equations
- Lattice-Boltzmann (LB) method
  - Discrete space, time & velocity
  - An LB algorithm
  - Chapman-Enskog analysis*
- Practical aspects of the LB method
  - Dimensional analysis
  - Boundary conditions
  - Coding
- Forces, collision operators
- Multiphase flow
  - Free energy LBM & interfaces*
  - Volume-averaged Navier-Stokes equation

UNIVERSITY OF ABERDEEN

* mathematically demanding

Distribution function

mass of molecules at location **x** at moment *t* traveling with velocity ξ

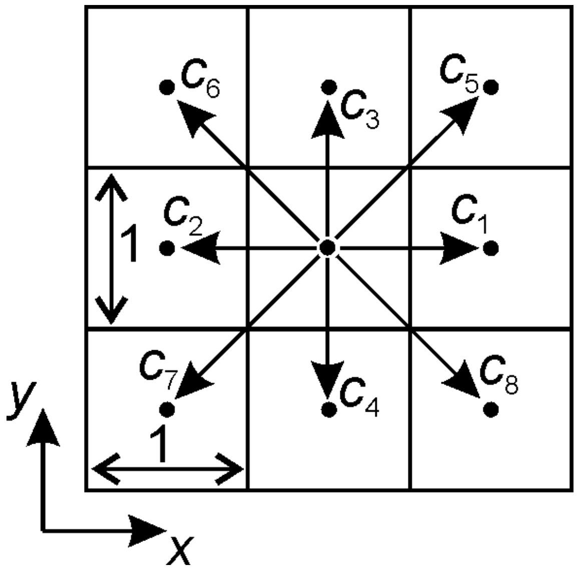
$$f\left(\mathbf{x}, \boldsymbol{\xi}, t\right)$$

& its discrete counterpart

$f_i\left(\mathbf{x}, t\right)$ with a velocity set $\mathbf{c_i} = \left(c_{ix}, c_{iy}, c_{iz}\right)$

integrations become summations: $\rho = \sum_i f_i \quad \rho\mathbf{u} = \sum_i \mathbf{c_i} f_i$

UNIVERSITY OF ABERDEEN



D2Q9

$$\Delta t = 1$$

$$\Delta x = 1$$

streaming: form lattice site to lattice site

collisions $$f_i^*(\mathbf{x},t) = f_i(\mathbf{x},t) + \Omega(\mathbf{x},t)$$

post-collision pre-collision collision operator

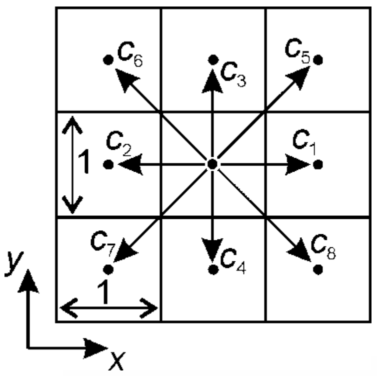
streaming $$f_i(\mathbf{x}+\mathbf{c_i},t+1) = f_i^*(\mathbf{x},t)$$

1495 UNIVERSITY OF ABERDEEN

BGK

$$\Omega_i(\mathbf{x},t) = \Omega_i(f) = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

need a discrete version of the equilibrium distribution function



$$f_i^{eq} = w_i \rho \left[1 + \frac{u_\alpha c_{i\alpha}}{c_s^2} + \frac{\left(u_\alpha c_{i\alpha}\right)^2}{2c_s^4} - \frac{u_\alpha u_\alpha}{2c_s^2}\right]$$

D2Q9

$$w_0 = 4/9 \quad w_{1-4} = 1/9 \quad w_{5-8} = 1/36 \quad c_s^2 = 1/3$$

UNIVERSITY OF ABERDEEN

LBE to “Navier-Stokes”

$$f_i\left(\mathbf{x}+\mathbf{c_i},t+1\right)=f_i\left(\mathbf{x},t\right)-\frac{1}{\tau}\left(f_i\left(\mathbf{x},t\right)-f_i^{eq}\left(\mathbf{x},t\right)\right)$$

Chapman-Enskog analysis

$$\frac{\partial}{\partial t}\left(\rho u_\beta\right)+\frac{\partial}{\partial x_\alpha}\left(\rho u_\alpha u_\beta\right)=-\frac{\partial p}{\partial x_\beta}+\nu\frac{\partial}{\partial x_\alpha}\left(\rho\left[\frac{\partial u_\alpha}{\partial x_\beta}+\frac{\partial u_\beta}{\partial x_\alpha}\right]\right)$$

with $p=c_s^2\rho \qquad \nu=c_s^2\left(\tau-\tfrac{1}{2}\right)$

if ρ were constant, this would be incompressible Navier-Stokes

….but ρ is not constant

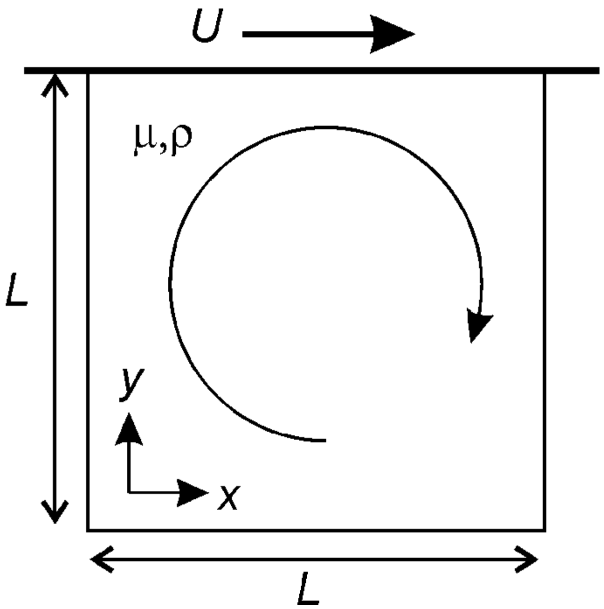
UNIVERSITY OF ABERDEEN

(in)compressibility

$$\rho \approx \text{constant if Ma}=|\mathbf{u}|/c_s \ll 1$$

keep flow velocities *in lattice units* well below speed of sound in *lattice units*

"Scaling"



$$Re = \frac{\rho U L}{\mu} = \frac{UL}{\nu}$$

two square lid-driven cavity flow systems (e.g. a physical one and an LB one) are the same* if they have the same Re

*the same in dimensionless variables

$$\tilde{x} = x/L,\ \tilde{y} = y/L,\ \tilde{t} = tU/L,\ \tilde{\mathbf{u}} = \mathbf{u}/U$$

$$\tilde{\mathbf{u}}\left(\tilde{x}, \tilde{y}, \tilde{t}\right)$$

designing an LB simulation

- choose *U* based on compressibility constraint
- choose *L* based on required resolution
- determine ν to match Re

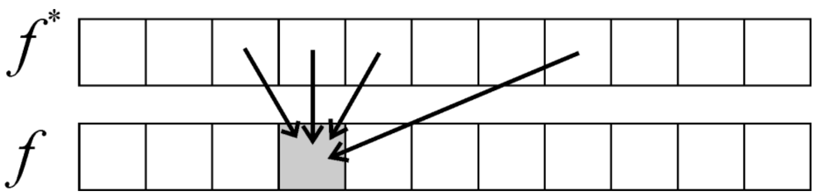
Coding

put some thought in your program *e.g. streaming*

$$f_i\left(\mathbf{x}+\mathbf{c_i},t+1\right)= f_i^*\left(\mathbf{x},t\right)$$

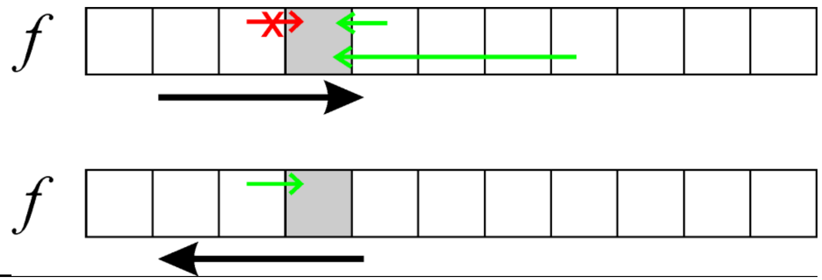
```
for j=1:ny
  for i=1:nx
    f(0,i,j)=fstar(0,i,j)
    f(1,i,j)=fstar(1,i-1,j)
    f(2,i,j)=fstar(2,i+1,j)
    f(3,i,j)=fstar(3,i,j-1)
    f(4,i,j)=fstar(4,i,j+1)
    f(5,i,j)=fstar(5,i-1,j-1)
    f(6,i,j)=fstar(6,i+1,j-1)
    f(7,i,j)=fstar(7,i+1,j+1)
    f(8,i,j)=fstar(8,i-1,j+1)
  end
end
```

needs two large arrays

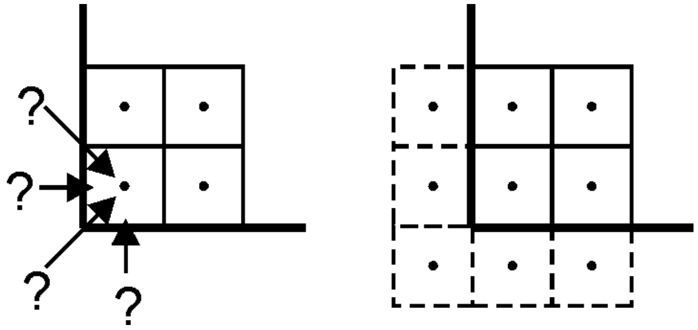



```
for j=1:ny
  for i=1:nx
    f(2,i,j)=f(2,i+1,j)
    f(4,i,j)=f(4,i,j+1)
    f(7,i,j)=f(7,i+1,j+1)
    f(8,i,j)=f(8,i-1,j+1)
  end
end
for j=ny:-1:1
  for i=nx:-1:1
    f(1,i,j)=f(1,i-1,j)
    f(3,i,j)=f(3,i,j-1)
    f(5,i,j)=f(5,i-1,j-1)
    f(6,i,j)=f(6,i+1,j-1)
  end
end
```

needs one large array

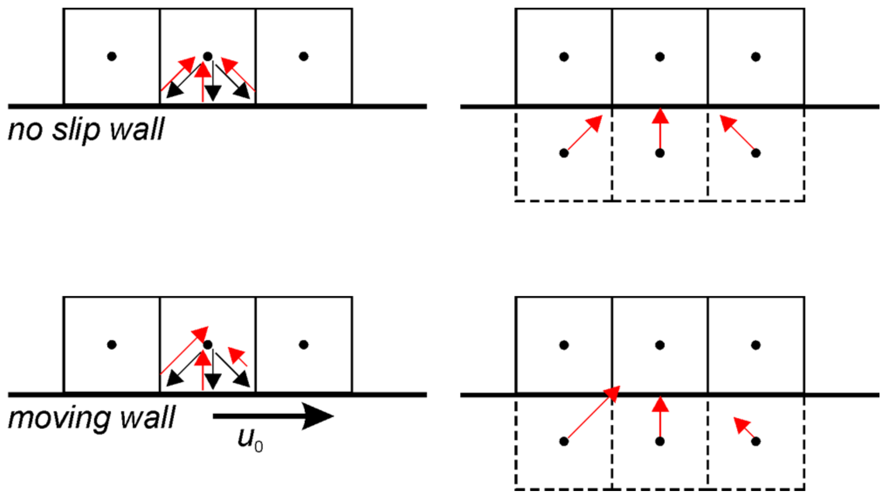


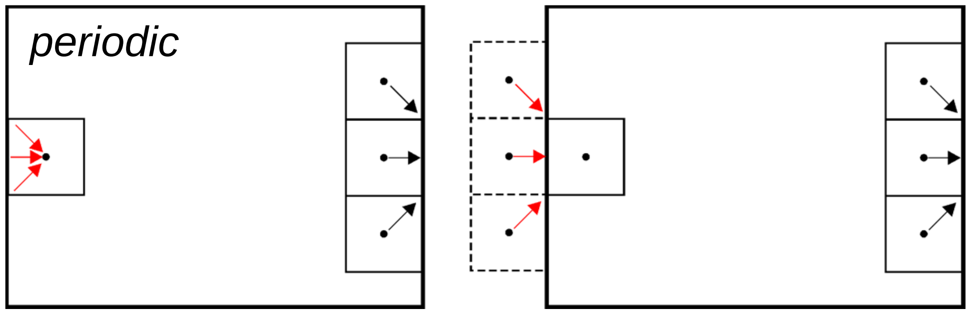
Boundary conditions



a ghost cell framework

- fill ghost cells with the appropriate *f**
- then stream towards all "real" cells





all Cartesian-based – flat surfaces or staircases

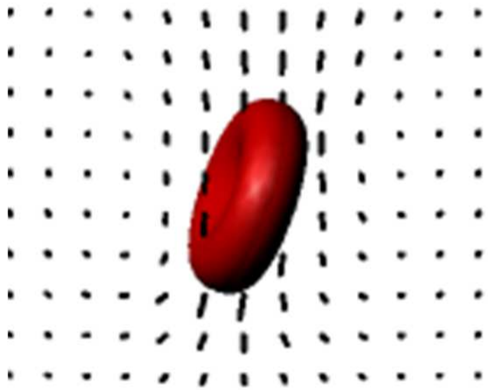
UNIVERSITY OF ABERDEEN

Immersed boundary conditions

want to do off-grid boundaries

immersed boundary method

can be implemented through forcing the fluid to a desired velocity at a desired (of lattice) location and so achieve **no-slip**



first need to know how to incorporate **forces** in LBM

UNIVERSITY OF ABERDEEN

Incorporating forces in LBGK

$$\rho \frac{\partial u_\alpha}{\partial t} + \rho u_\beta \frac{\partial u_\alpha}{\partial x_\beta} = -\frac{\partial p}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta}\left[\mu\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right)\right] + F_\alpha$$

options:

1. go via the collision operator no forces: $\sum_i \Omega_i c_{i\alpha} = 0$ with forces: $\sum_i \Omega_i c_{i\alpha} = F_\alpha$

2. include a new term in the LBE equation $f_i^* = f_i + \Omega_i + S_i$

$$S_i = \left(1 - \frac{1}{2\tau}\right) w_i \left(\frac{c_{i\alpha}}{c_s^2} + \frac{\left(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}\right)u_\beta}{c_s^4}\right) F_\alpha$$

this is pretty complicated, e.g. note the double summation convention

$$c_{i\alpha}c_{i\beta}u_\beta F_\alpha = c_{ix}c_{ix}u_x F_x + c_{iy}c_{ix}u_x F_y + c_{ix}c_{iy}u_y F_x + c_{iy}c_{iy}u_y F_y$$

Incorporating forces in LBGK – 2

this needs a “force correction” for momentum; density does not need a correction

$$\rho u_\alpha = \sum_i f_i c_\alpha + \tfrac{1}{2} F_\alpha \qquad \rho = \sum_i f_i$$

this all can be derived through Chapman-Enskog analysis

Immersed boundary method – in words

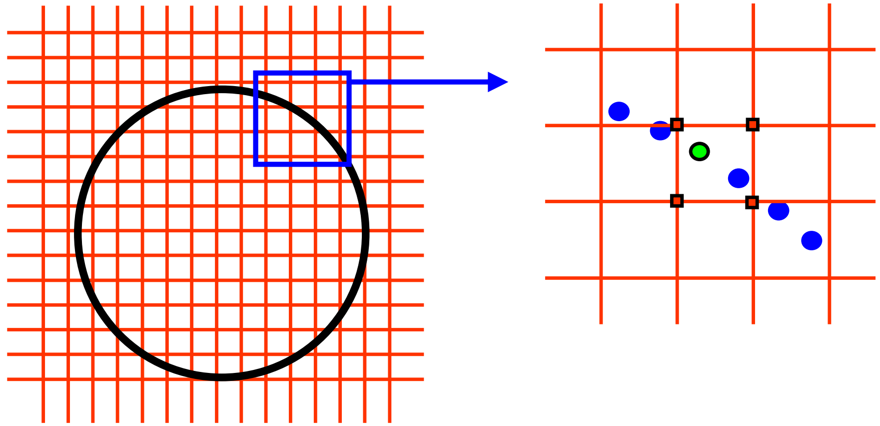
represent an off-grid surface through marker points

interpolate velocity to the marker points

determine the difference between interpolated velocity and the desired velocity at the marker point

calculate a force at the marker point that opposes the velocity difference

distribute the force over the surrounding lattice nodes



spacing marker points < 1

linear interpolation works well

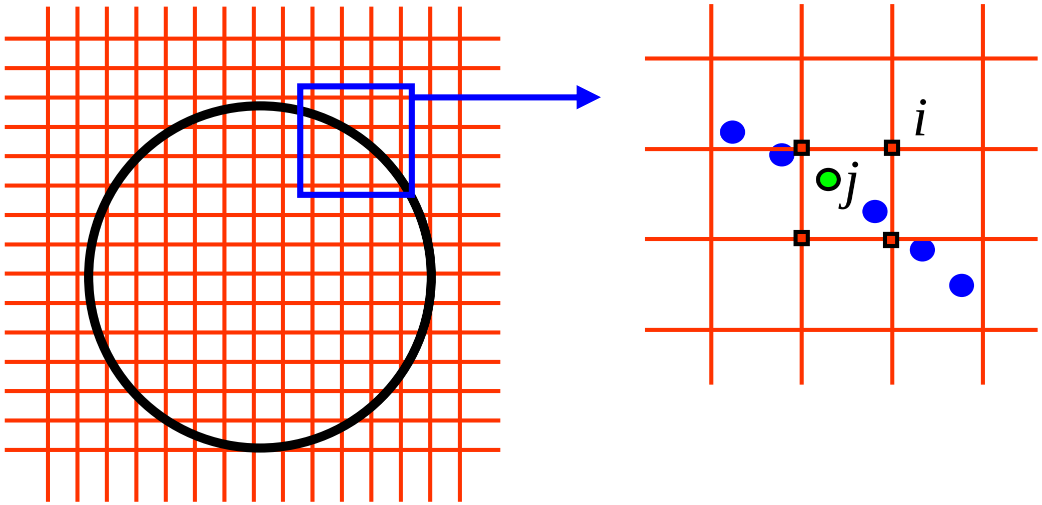
1495
UNIVERSITY OF ABERDEEN

Immersed boundary method – in eq's

$$\mathbf{w_j} = \sum_i I\left(\mathbf{r_{ij}}\right)\mathbf{u_i}$$

$$\mathbf{F_j} = \alpha\mathbf{F_j^{old}} - \beta\left(\mathbf{w_j} - \mathbf{v_j}\right)$$

$$\mathbf{F_i} = I\left(\mathbf{r_{ij}}\right)\mathbf{F_j}$$



i lattice point

j marker point

$\mathbf{u_i}$ lattice velocity

$\mathbf{w_j}$ interpolated velocity at marker point

$\mathbf{v_j}$ desired velocity at marker point

$\mathbf{F_j}$ marker point force

$\mathbf{F_i}$ lattice point force

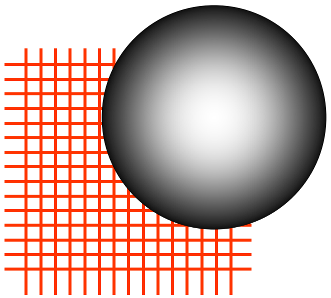
α β empirical constants

another issue is the order in which we go over the marker points

UNIVERSITY OF ABERDEEN

An application: particle-resolved simulations

a small excursion into *three* dimensions

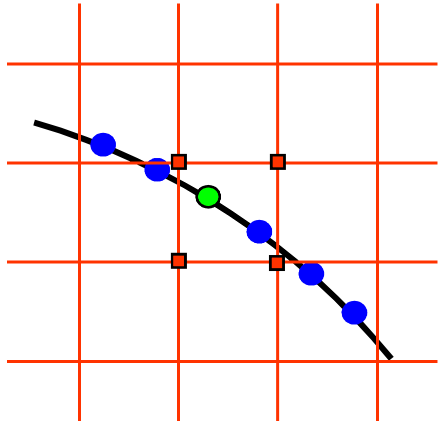


suppose the marker points lie on a spherical surface

$\sum_{all\,\bullet} \mathbf{F}$ is the force acting on the fluid to impose no-slip at the particle surface

$$\sum_{all\,\bullet} \mathbf{F} = -\,\mathbf{F}_{\mathbf{f}\rightarrow\mathbf{p}}$$

similarly $\mathbf{T}_{\mathbf{f}\rightarrow\mathbf{p}} = -\sum_{all\,\bullet} \mathbf{F} \times (\mathbf{r} - \mathbf{R}_\mathbf{p})$



particles have internal fluid

$\mathbf{v}_\mathbf{j}$ desired velocity at marker point

$$\mathbf{v}_\mathbf{j} = \mathbf{u}_\mathbf{p} + \boldsymbol{\Omega}_\mathbf{p} \times (\mathbf{x}_\mathbf{j} - \mathbf{x}_{\mathbf{cp}})$$

UNIVERSITY OF ABERDEEN

Dealing with internal fluid

for rigid particle dynamics

$$\rho_p V_p \frac{d\mathbf{u_p}}{dt} = \oint_S \mathbf{t}\, dS + \mathbf{g} V \left(\rho_p - \rho\right)$$

$$\int_V \mathbf{f}\, dV = -\oint_S \mathbf{t}\, dS + \rho V_p \frac{d\mathbf{u_p}}{dt}$$

$$\left(\rho_p - \rho\right) V_p \frac{d\mathbf{u_p}}{dt} = -\int_V \mathbf{f}\, dV + \mathbf{g} V \left(\rho_p - \rho\right)$$

f: body force **on** fluid (internal + external due to immersed boundary method

t traction **on** solid particle

$$\mathbf{I} \frac{d\boldsymbol{\omega}_\mathbf{p}}{dt} = \mathbf{M_h} + \boldsymbol{\omega}_\mathbf{p} \times \left(\mathbf{I} \boldsymbol{\omega}_\mathbf{p}\right)$$

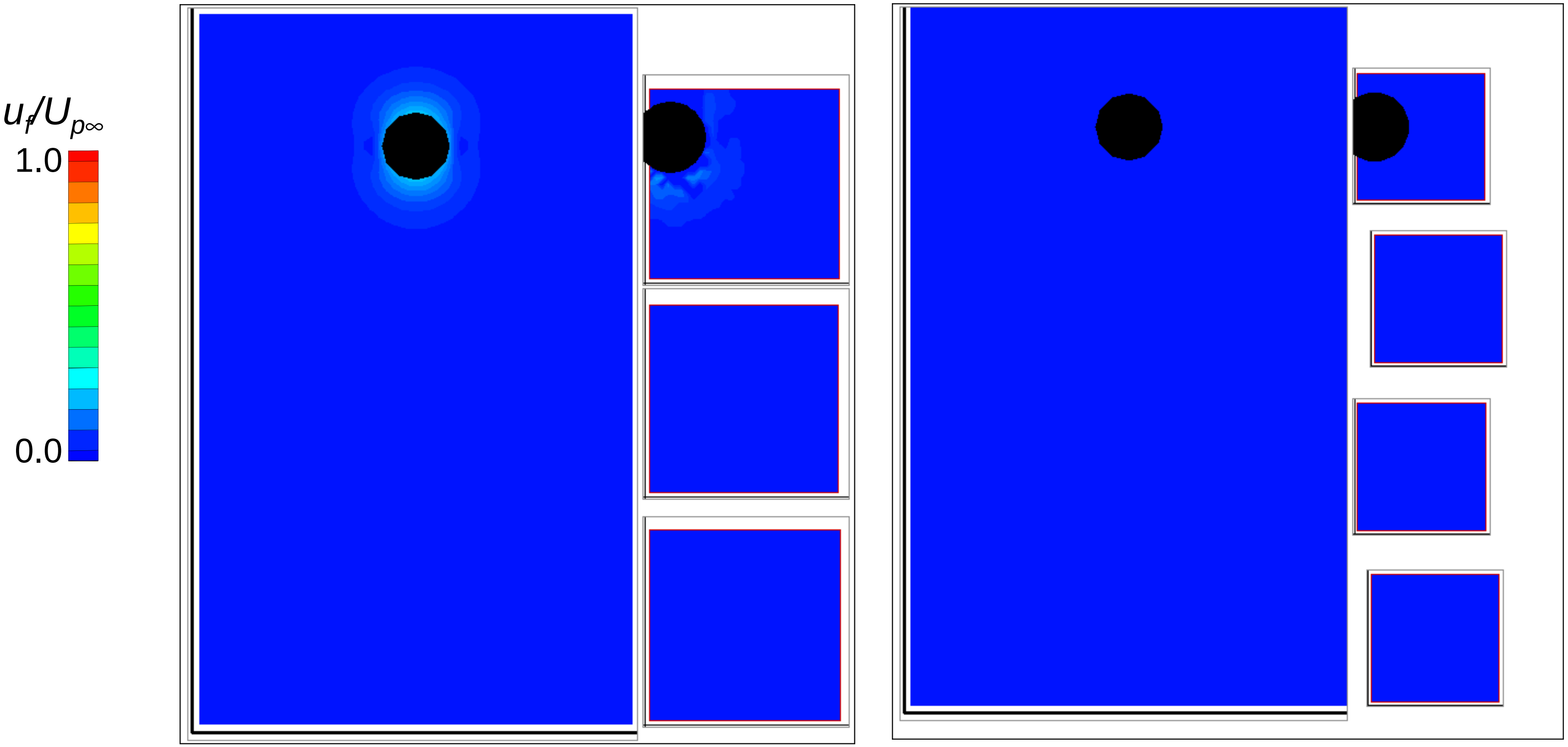
in a reference frame aligned with the principal axes of the particle

S^{-1} coordinate transform

$$\left(\rho_p - \rho\right) \mathbf{I} \frac{d\boldsymbol{\omega}_\mathbf{p}}{dt} = \rho_p \mathbf{S}^{-1} \int_V \left[\mathbf{r} - \mathbf{R_p}\right] \times \mathbf{f}\, dV + \left(\rho_p - \rho\right) \boldsymbol{\omega}_\mathbf{p} \times \left(\mathbf{I} \boldsymbol{\omega}_\mathbf{p}\right)$$

Simulation versus experiment

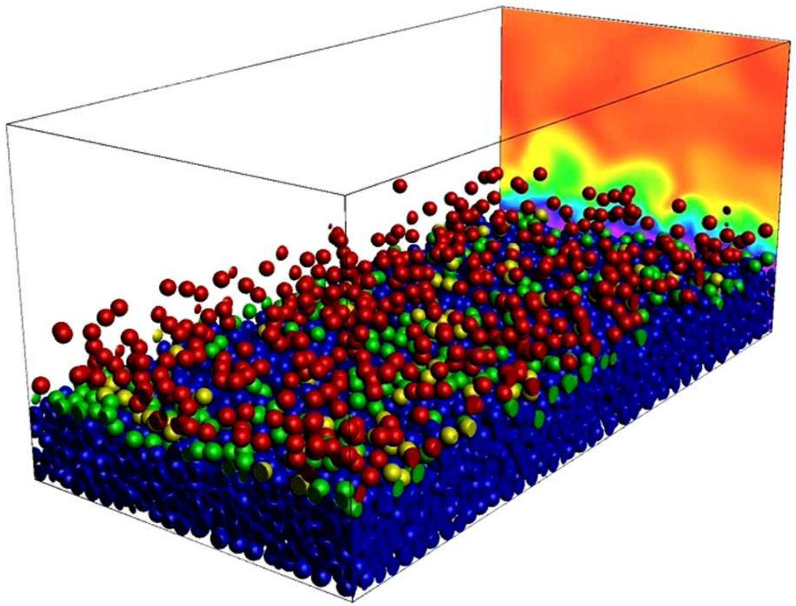
some dynamics



$Re_p = 1.5$ $Re_p = 32$

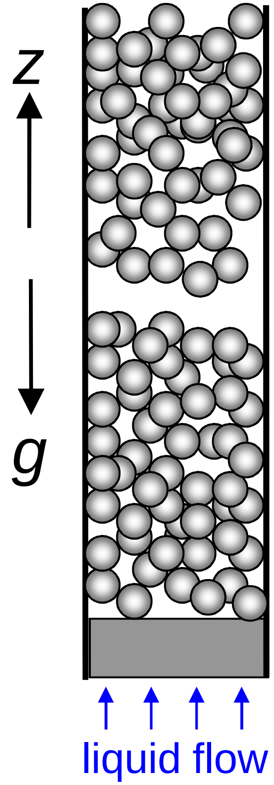
An application: particle-resolved simulations

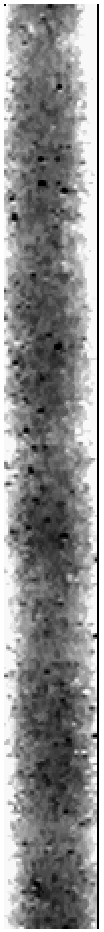
sediment transport



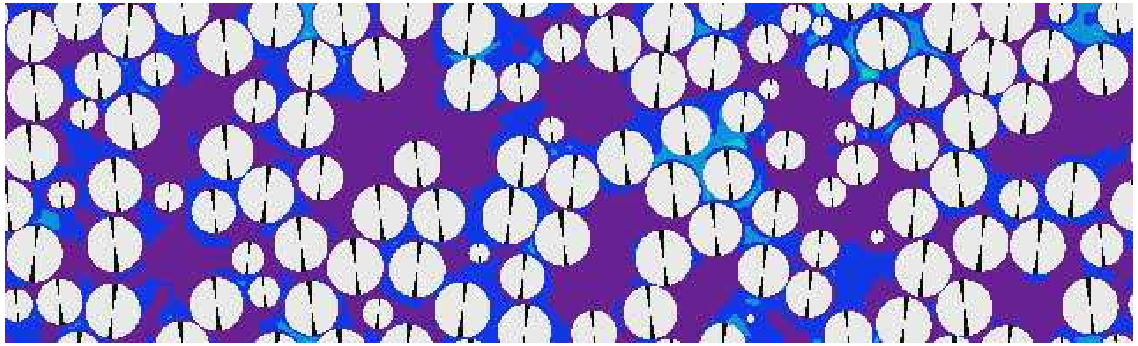
An application: particle-resolved simulations

typically: 1 mm glass beads in water





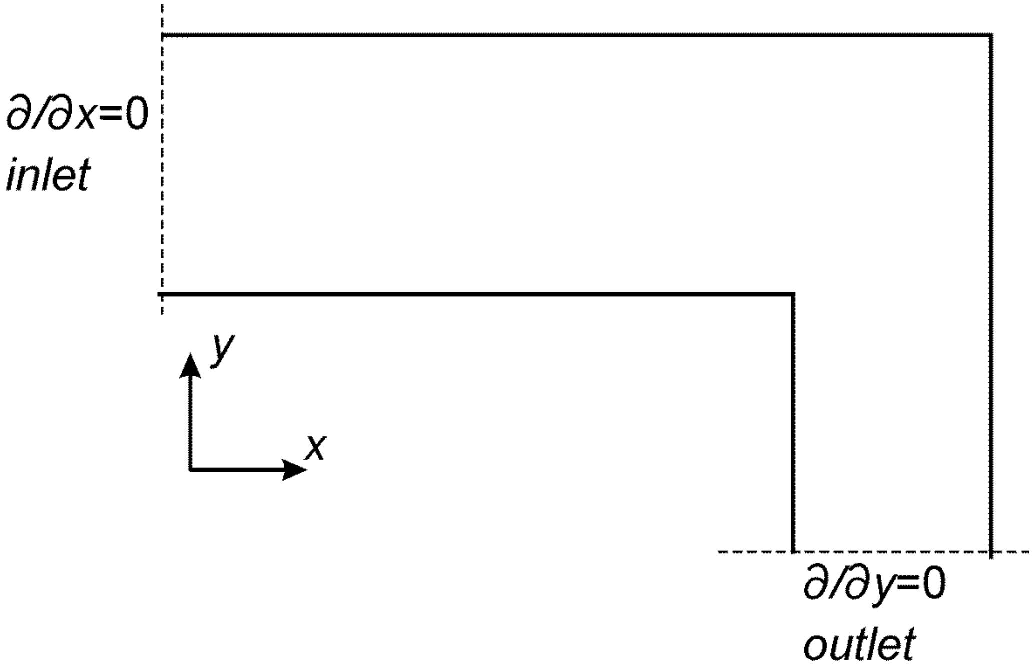
liquid-solid fluidization



body force on fluid → ← gravity

periodic boundary conditions
cross section through 3D domain

Inlet / outlet



$$* \phi_{m,in} = \int_{inlet} \rho u_x dy$$

$$** \phi_{m,out} = -\int_{outlet} \rho u_y dx$$

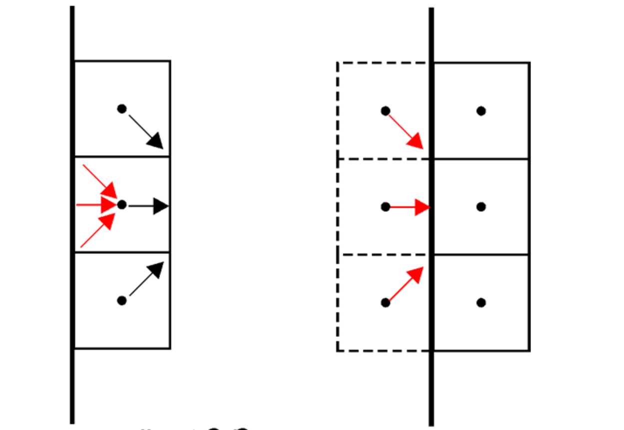
$$F_{outlet,y}^{(k+1)} = F_{outlet,y}^{(k)} + \alpha\left(\phi_{m,out} - \phi_{m,in}\right)$$

$\alpha > 0$ control algorithm α empirical

at the inlet: impose uniform velocity through IBM

every time step: calculate the mass influx*

apply a uniform force (in y-direction) that makes the mass outflux equal to the influx**

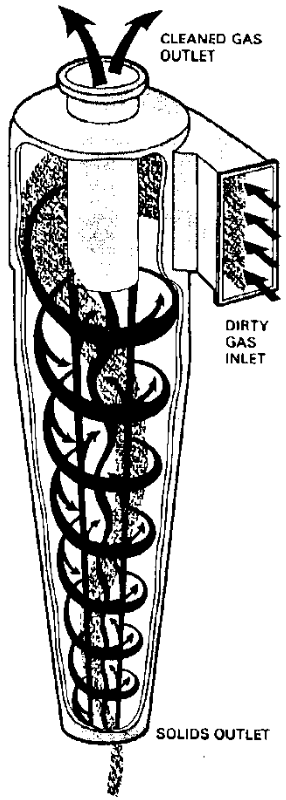


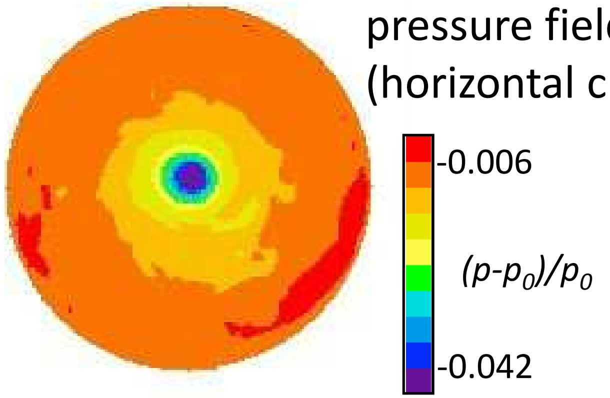
zero gradient ∂/∂x=0

UNIVERSITY OF ABERDEEN

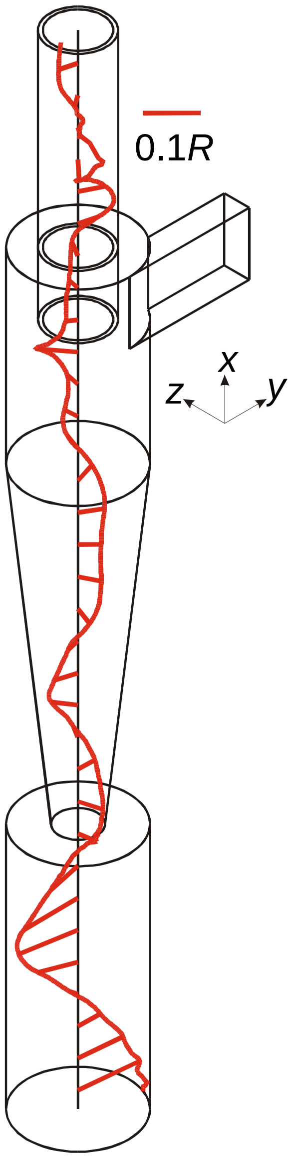
Inlet / outlet – an application

cyclones for gas-solid separation





pressure field
(horizontal cross section)



Revisit the collision operator

BGK

$$\Omega_i(f) = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

issues with BGK

- stability (at low viscosity)
- accuracy, e.g. $u_\alpha u_\beta u_\gamma = O\left(u^3\right)$

no a priori reason why all distribution functions would relax at the same rate, i.e. with the same time constant

Multiple Relaxation Time operator

let different ***velocity moments*** of the distribution function relax at different rates

velocity moments are linear combinations of f_i's

$$\mathbf{m} = \mathbf{M} \cdot \boldsymbol{f}$$

$$\boldsymbol{f} = \left(f_0, f_1 \cdots f_8 \right)^{trans}$$

$$\mathbf{M} = \begin{bmatrix} m_{00} & m_{01} & \cdots & m_{08} \\ m_{10} & m_{11} & \cdots & m_{18} \\ \vdots & & \ddots & \vdots \\ m_{80} & m_{81} & \cdots & m_{88} \end{bmatrix}$$

a constant coefficient matrix

From BGK to MRT

$$f_i(\mathbf{x}+\mathbf{c_i},t+1)-f_i(\mathbf{x},t)=-\omega\left[f_i(\mathbf{x},t)-f_i^{eq}(\mathbf{x},t)\right] \text{ with } \omega=1/\tau$$

in vector form

$$\boldsymbol{f}(\mathbf{x}+\mathbf{c_i},t+1)-\boldsymbol{f}(\mathbf{x},t)=-\omega\left[\boldsymbol{f}(\mathbf{x},t)-\boldsymbol{f}^{eq}(\mathbf{x},t)\right]$$

$$\boldsymbol{f}(\mathbf{x}+\mathbf{c_i},t+1)-\boldsymbol{f}(\mathbf{x},t)=-\mathbf{M}^{-1}\mathbf{M}\omega\left[\boldsymbol{f}(\mathbf{x},t)-\boldsymbol{f}^{eq}(\mathbf{x},t)\right]$$

$$\boldsymbol{f}(\mathbf{x}+\mathbf{c_i},t+1)-\boldsymbol{f}(\mathbf{x},t)=-\mathbf{M}^{-1}\omega\left[\mathbf{M}\boldsymbol{f}(\mathbf{x},t)-\mathbf{M}\boldsymbol{f}^{eq}(\mathbf{x},t)\right]$$

$$\boldsymbol{f}(\mathbf{x}+\mathbf{c_i},t+1)-\boldsymbol{f}(\mathbf{x},t)=-\mathbf{M}^{-1}\omega\left[\mathbf{m}(\mathbf{x},t)-\mathbf{m}^{eq}(\mathbf{x},t)\right]$$

define $\mathbf{S}=\omega\mathbf{I}$

$$\boldsymbol{f}(\mathbf{x}+\mathbf{c_i},t+1)-\boldsymbol{f}(\mathbf{x},t)=-\mathbf{M}^{-1}\mathbf{S}\left[\mathbf{m}(\mathbf{x},t)-\mathbf{m}^{eq}(\mathbf{x},t)\right]$$

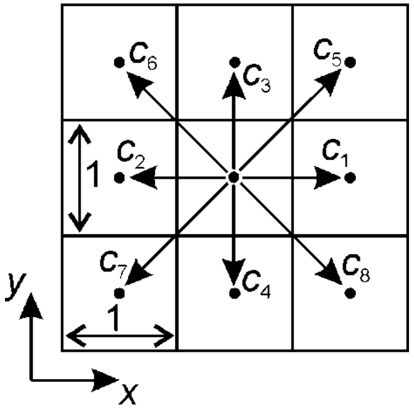
From BGK to MRT – 2

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c_i},t+1\right)-\boldsymbol{f}\left(\mathbf{x},t\right)=-\mathbf{M}^{-1}\mathbf{S}\left[\mathbf{m}\left(\mathbf{x},t\right)-\mathbf{m}^{eq}\left(\mathbf{x},t\right)\right]$$

now we can assign different relaxation rates to different velocity moments

$$\mathbf{S}=\begin{bmatrix}\omega_0 & 0 & \cdots & 0\\ 0 & \omega_1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \omega_8\end{bmatrix}$$

Velocity moments – D2Q9



"Gram-Schmidt procedure"

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & -2 & 2 & 1 & -1 & 1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix}$$

$$0: \rho^{eq} = \rho$$
$$1: e^{eq} = \rho - 3\rho\left(u_x^2 + u_y^2\right)$$
$$2: \varepsilon^{eq} = 9\rho u_x^2 - 3\rho\left(u_x^2 + u_y^2\right) + \rho$$
$$3: j_x^{eq} = \rho u_x$$
$$4: q_x^{eq} = 3\rho u_x^3 - \rho u_x$$
$$5: j_y^{eq} = \rho u_y$$
$$6: q_y^{eq} = 3\rho u_y^3 - \rho u_y$$
$$7: p_{xx}^{eq} = \rho\left(u_x^2 - u_y^2\right)$$
$$8: p_{xy}^{eq} = \rho u_x u_y$$

1495 UNIVERSITY OF ABERDEEN

Relaxation rates

$$\mathbf{S} = \text{diag}\left(0, \omega_e, \omega_\varepsilon, 0, \omega_q, 0, \omega_q, \omega_\nu, \omega_\nu\right)$$

$$0: \rho^{eq} = \rho$$
$$1: e^{eq} = \rho - 3\rho\left(u_x^2 + u_y^2\right)$$
$$2: \varepsilon^{eq} = 9\rho u_x^2 - 3\rho\left(u_x^2 + u_y^2\right) + \rho$$
$$3: j_x^{eq} = \rho u_x$$
$$4: q_x^{eq} = 3\rho u_x^3 - \rho u_x$$
$$5: j_y^{eq} = \rho u_y$$
$$6: q_y^{eq} = 3\rho u_y^3 - \rho u_y$$
$$7: p_{xx}^{eq} = \rho\left(u_x^2 - u_y^2\right)$$
$$8: p_{xy}^{eq} = \rho u_x u_y$$

density and momentum have zero relaxation rates

we get closer to the Navier-Stokes eq.

$$\frac{\partial}{\partial t}\left(\rho u_\beta\right) + \frac{\partial}{\partial x_\alpha}\left(\rho u_\alpha u_\beta\right) = -\frac{\partial p}{\partial x_\beta} + \frac{\partial}{\partial x_\alpha}\left(\mu\left[\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right] + \left(\mu_b - \tfrac{2}{3}\mu\delta_{\alpha\beta}\right)\frac{\partial u_\gamma}{\partial x_\gamma}\right)$$

$$p = c_s^2\rho \quad \mu = \rho c_s^2\left(\frac{1}{\omega_\nu} - \frac{1}{2}\right) \quad \mu_b = \rho c_s^2\left(\frac{1}{\omega_e} - \frac{1}{2}\right) - \frac{1}{3}\mu$$

“free” parameters

$$\omega_e = \omega_q = 1$$

1495 UNIVERSITY OF ABERDEEN

An LB – MRT algorithm

start with a set of f_i's on a lattice

1. determine $\rho = \sum_i f_i \;\; \rho\mathbf{u} = \sum_i \mathbf{c_i} f_i$
2. determine $\mathbf{m}^{eq}$ (needs density & velocity)
3. determine $\mathbf{m} = \mathbf{M} \cdot \boldsymbol{f}$
4. perform the collision $\boldsymbol{f}^*(\mathbf{x},t) = -\mathbf{M}^{-1}\mathbf{S}\left[\mathbf{m}(\mathbf{x},t) - \mathbf{m}^{eq}(\mathbf{x},t)\right]$
5. take care of boundary conditions
6. stream $f_i^*(\mathbf{x}+\mathbf{c_i}, t+1) = f_i(\mathbf{x},t)$

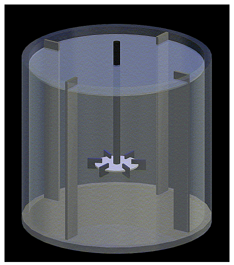
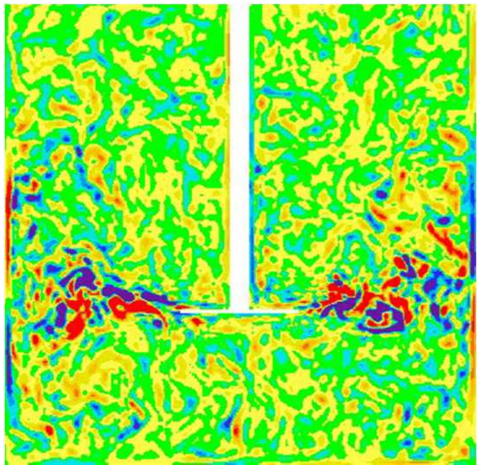
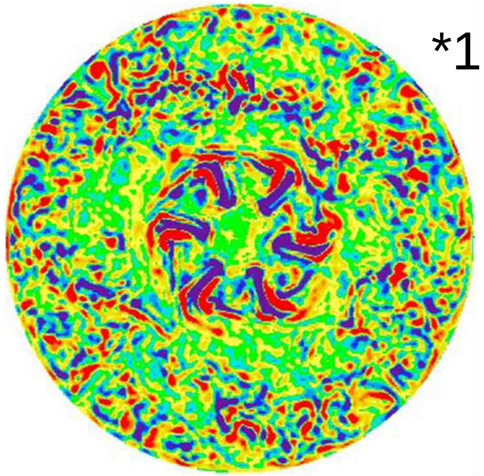
1495 UNIVERSITY OF ABERDEEN

Turbulence

why (not) perform turbulence simulations with the lattice-Boltzmann method

why not:
- uniform & cubic grid
  - no local grid refinement
- small time steps
  - no point in doing RANS with LBM

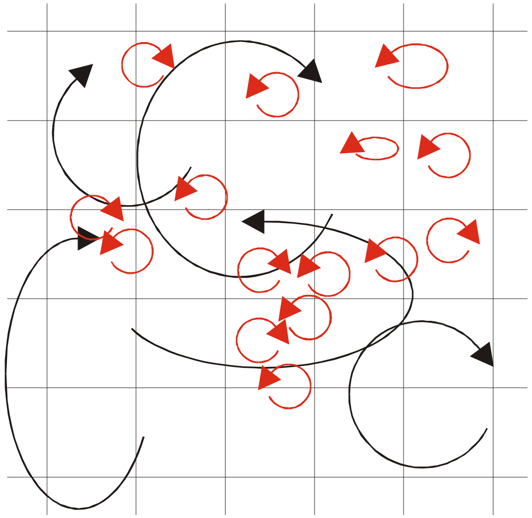
why ~~not~~:
- if only for fun*
- geometric flexibility
  - moving boundaries (IBM)
- "easy" to do large-eddy simulations

*1998

UNIVERSITY OF ABERDEEN

Large-eddy simulations of turbulence



unresolved eddies
resolved eddies

the trouble with (numerical simulations of) turbulence:
resolutions of the fine length (and time) scales

Kolmogorov length scale

$$\frac{\eta_K}{L} \propto \mathrm{Re}^{-3/4} \rightarrow \text{if } \mathrm{Re} = 10^6 \rightarrow \eta_K \approx 3 \cdot 10^{-5} L$$

$$\Delta \approx \eta_K \approx 3 \cdot 10^{-5} L \approx \frac{L}{3 \cdot 10^4} \qquad N \approx \left(3 \cdot 10^4\right)^3 \approx 3 \cdot 10^{13}$$

mitigate this issue through a subgrid-scale model & perform LES

$$\nu_{eddy} = \left(c_S \Delta\right)^2 \sqrt{2 \bar{S}_{\alpha\beta} \bar{S}_{\alpha\beta}} \quad \text{with } \bar{S}_{\alpha\beta} = \tfrac{1}{2}\left(\frac{\partial \bar{u}_\alpha}{\partial x_\beta} + \frac{\partial \bar{u}_\beta}{\partial x_\alpha}\right)$$

1495 UNIVERSITY OF ABERDEEN

LES (in LBM)*

$$\bar{S}_{\alpha\beta} = \tfrac{1}{2}\left(\frac{\partial \bar{u}_\alpha}{\partial x_\beta} + \frac{\partial \bar{u}_\beta}{\partial x_\alpha}\right)$$

is readily available in LBM (at least in LBGK)

$$\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \approx$$

$$\approx \frac{1}{\rho c_s^2 \tau} \sum_i c_{i\alpha} c_{i\beta} \left(f_i - f_i^{eq} \right)$$

note the overbars

start with a set of f_i's on a lattice

1. determine $\rho = \sum_i f_i \;\; \rho\mathbf{u} = \sum_i \mathbf{c_i} f_i$
2. determine f_i^{eq} (needs density & velocity)

2a. determine $S_{\alpha\beta}$, ν_{eddy}, $\tau = 3\left(\nu_{eddy} + \nu\right) + \tfrac{1}{2}$

3. perform the collision

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

5. take care of boundary conditions
6. stream $f_i^*(\mathbf{x}+\mathbf{c_i}, t+1) = f_i(\mathbf{x},t)$

* note that turbulence is inherently three-dimensional