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

* mathematically demanding
Distribution function

mass of molecules at location \( x \) at moment \( t \)
traveling with velocity \( \xi \)

\[
f(x, \xi, t)
\]

& its discrete counterpart

\[
f_i(x, t) \quad \text{with a velocity set} \quad c_i = (c_{ix}, c_{iy}, c_{iz})
\]

integrations become summations:

\[
\rho = \sum_i f_i \quad \rho u = \sum_i c_i f_i
\]
\[ \Delta t = 1 \quad \text{streaming: form lattice site to lattice site} \]

\[ \Delta x = 1 \]

\[ f^*_i (\mathbf{x}, t) = f_i (\mathbf{x}, t) + \Omega (\mathbf{x}, t) \]

\[ f_i (\mathbf{x} + \mathbf{c}_i, t + 1) = f^*_i (\mathbf{x}, t) \]
BGK

\[ \Omega_i(x, t) = \Omega_i(f) = -\frac{1}{\tau}(f_i - f_i^{eq}) \]

need a discrete version of the equilibrium distribution function

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

D2Q9
\[
\begin{align*}
w_0 &= 4/9 \\
w_{1-4} &= 1/9 \\
w_{5-8} &= 1/36 \\
c_s^2 &= 1/3
\end{align*}
\]
LBE to “Navier-Stokes”

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

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 \( p = c_s^2 \rho \quad \nu = c_s^2 \left( \tau - \frac{1}{2} \right) \)

if \( \rho \) were constant, this would be incompressible Navier-Stokes

....but \( \rho \) is not constant
(in)compressibility

\[ \rho \approx \text{constant if } Ma = \frac{|u|}{c_s} \ll 1 \]

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

\[ \text{Re} = \frac{\rho UL}{\mu} = \frac{UL}{\nu} \]

*the same in dimensionless variables
\[ \tilde{x} = x/L, \quad \tilde{y} = y/L, \quad \tilde{t} = tU/L, \quad \tilde{u} = u/U \]

\[ \tilde{u}(\tilde{x}, \tilde{y}, \tilde{t}) \]

designing an LB simulation
- choose \( U \) based on compressibility constraint
- choose \( L \) based on required resolution
- determine \( \nu \) to match Re
Coding

put some thought in your program \( e.g. \) streaming

\[ f_i(x + c_i, t + 1) = f_i^*(x, t) \]

for \( j = 1 \) to \( ny \)
for \( i = 1 \) to \( nx \)
  \( f(0, i, j) = f^*(0, i, j) \)
  \( f(1, i, j) = f^*(1, i-1, j) \)
  \( f(2, i, j) = f^*(2, i+1, j) \)
  \( f(3, i, j) = f^*(3, i, j-1) \)
  \( f(4, i, j) = f^*(4, i, j+1) \)
  \( f(5, i, j) = f^*(5, i-1, j-1) \)
  \( f(6, i, j) = f^*(6, i+1, j-1) \)
  \( f(7, i, j) = f^*(7, i+1, j+1) \)
  \( f(8, i, j) = f^*(8, i-1, j+1) \)
end
end

needs two large arrays

for \( j = ny \) to 1
for \( i = nx \) to 1
  \( f(2, i, j) = f(2, i+1, j) \)
  \( f(4, i, j) = f(4, i, j+1) \)
  \( f(7, i, j) = f(7, i+1, j+1) \)
  \( f(8, i, j) = f(8, i-1, j+1) \)
end
end

needs one large array
Boundary conditions

- Ghost cell framework
  - Fill ghost cells with the appropriate \( f^* \)
  - Then stream towards all “real” cells

All Cartesian-based – flat surfaces or staircases
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
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: \( \sum_i \Omega_i c_i = 0 \)  \( \sum_i \Omega_i c_i = F_\alpha \)

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

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

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

\[ \rho u_\alpha = \sum_i f_i c_\alpha + \frac{1}{2} F_\alpha \quad \rho = \sum_i f_i \]

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
Immersed boundary method – in eq’s

\[ w_j = \sum_i I(r_{ij})u_i \]
\[ F_j = \alpha F_j^{old} - \beta (w_j - v_j) \]
\[ F_i = I(r_{ij})F_j \]

\( i \) lattice point  \hspace{1cm} \( F_j \) marker point force
\( j \) marker point  \hspace{1cm} \( F_i \) lattice point force
\( u_i \) lattice velocity
\( w_j \) interpolated velocity at marker point
\( v_j \) desired velocity at marker point
\( \alpha, \beta \) empirical constants

another issue is the order in which we go over the marker points
An application: particle-resolved simulations

a small excursion into *three* dimensions

*suppose the marker points lie on a spherical surface*

\[ \sum \mathbf{F} \text{ is the force acting on the fluid to impose no-slip at the particle surface} \]

\[ \sum_{\text{all}} \mathbf{F} = -\mathbf{F}_{\text{f \rightarrow p}} \]

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

*particles have internal fluid*

\[ \mathbf{v}_j \text{ desired velocity at marker point} \]

\[ \mathbf{v}_j = \mathbf{u}_p + \mathbf{\Omega}_p \times (\mathbf{x}_j - \mathbf{x}_{cp}) \]
Dealing with internal fluid for rigid particle dynamics

\[ \rho_p V_p \frac{du_p}{dt} = \oint_S t dS + gV (\rho_p - \rho) \]

\[ \int f dV = -\oint_S t dS + \rho V_p \frac{du_p}{dt} \]

\[ (\rho_p - \rho) V_p \frac{du_p}{dt} = -\int f dV + gV (\rho_p - \rho) \]

\[ I \frac{d\omega_p}{dt} = M_h + \omega_p \times (I \omega_p) \]

\[ (\rho_p - \rho) I \frac{d\omega_p}{dt} = \rho_p S^{-1} \int_V [r - R_p] \times f dV + (\rho_p - \rho) \omega_p \times (I \omega_p) \]

- \( f \): body force on fluid (internal + external due to immersed boundary method)
- \( t \): traction on solid particle
- In a reference frame aligned with the principal axes of the particle
- \( S^{-1} \): coordinate transform
Simulation versus experiment

some dynamics

$\frac{u}{U_{p\infty}}$

$Re_p = 1.5$

$Re_p = 32$
An application: particle-resolved simulations
An application: particle-resolved simulations

liquid-solid fluidization

typically: 1 mm glass beads in water

body force on fluid

gravity

periodic boundary conditions
cross section through 3D domain
Inlet / outlet

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**

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

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

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

$\alpha > 0$ control algorithm $\alpha$ empirical

zero gradient $\partial/\partial x = 0$
Inlet / outlet – an application

cyclones for gas-solid separation

pressure field
(horizontal cross section)

\( \frac{(p-p_0)}{p_0} \)

-0.006

-0.042

0.1 \( R \)
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. \( u_{\alpha} u_{\beta} u_{\gamma} = O(u^3) \)

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 \textit{velocity moments} of the distribution function relax at different rates.

Velocity moments are linear combinations of \( f_i \)'s

\[
m = M \cdot f
\]

\[
f = (f_0, f_1 \ldots f_8)^{\text{trans}}
\]

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

A constant coefficient matrix.
From BGK to MRT

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

in vector form

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

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

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

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

define \( S = \omega I \)

\[ f(x + c_i, t + 1) - f(x, t) = -M^{-1}S \left[ m(x, t) - m^{eq}(x, t) \right] \]
From BGK to MRT — 2

\[ f(x + c_1, t + 1) - f(x, t) = -M^{-1}S[m(x, t) - m^{eq}(x, t)] \]

Now we can assign different relaxation rates to different velocity moments:

\[ S = \begin{bmatrix} \omega_0 & 0 & \ldots & 0 \\ 0 & \omega_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \omega_8 \end{bmatrix} \]
Velocity moments – D2Q9

“Gram-Schmidt procedure”

\[
M = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\
4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\
0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & -2 & 2 & 1 & -1 & 1 & -1 \\
0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
\end{bmatrix}
\]

0: \( \rho_{eq} = \rho \)

1: \( e_{eq} = \rho - 3\rho(u_x^2 + u_y^2) \)

2: \( \varepsilon_{eq} = 9\rho u_x^2 - 3\rho(u_x^2 + u_y^2) + \rho \)

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: \( p_{xx}^{eq} = \rho(u_x^2 - u_y^2) \)

8: \( p_{xy}^{eq} = \rho u_x u_y \)
Relaxation rates

\[ S = \text{diag}(0, \omega_e, \omega_e, 0, \omega_q, 0, \omega_q, \omega_\nu, \omega_\nu) \]

density and momentum have zero relaxation rates

we get closer to the Navier-Stokes eq.

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

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

“free” parameters

\[ \omega_e = \omega_q = 1 \]
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 $\mathbf{m} = \mathbf{M} \cdot f$
4. perform the collision $\mathbf{f}^* (\mathbf{x},t) = -\mathbf{M}^{-1} \mathbf{S} [\mathbf{m} (\mathbf{x},t) - \mathbf{m}^{eq} (\mathbf{x},t)]$
5. take care of boundary conditions
6. stream $f_i^* (\mathbf{x} + \mathbf{c}_i, t + 1) = f_i (\mathbf{x}, t)$
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
Large-eddy simulations of turbulence

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

Kolmogorov length scale

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

\[ \Delta \approx \eta_K \approx 3 \cdot 10^{-5} L \approx \frac{L}{3 \cdot 10^4} \]

\[ N \approx (3 \cdot 10^4)^3 \approx 3 \cdot 10^{13} \]

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

\[ \nu_{eddy} = (c_s \Delta)^2 \sqrt{2 S_{\alpha\beta} \overline{S}_{\alpha\beta}} \quad \text{with} \quad \overline{S}_{\alpha\beta} = \frac{1}{2} \left( \frac{\partial \overline{u}_\alpha}{\partial x_\beta} + \frac{\partial \overline{u}_\beta}{\partial x_\alpha} \right) \]
LES (in LBM)*

\[
S_{\alpha\beta} = \frac{1}{2} \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right)
\]

is readily available in LBM (at least in LBGK)

\[
\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \approx \frac{1}{\rho c_s^2 \tau} \sum_i c_{i\alpha} c_{i\beta} (f_i - f_i^{eq})
\]

note the overbars

start with a set of \(f_i\)'s on a lattice

1. determine \(\rho = \sum_i f_i\) \(\rho u = \sum_i c_i f_i\)

2. determine \(f_i^{eq}\) (needs density & velocity)

2a. determine \(S_{\alpha\beta}, \nu_{eddy}, \tau = 3(\nu_{eddy} + \nu) + \frac{1}{2}\)

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