Lattice Boltzmann Method for Moving Boundaries

Hans Groot

March 18, 2009

Outline

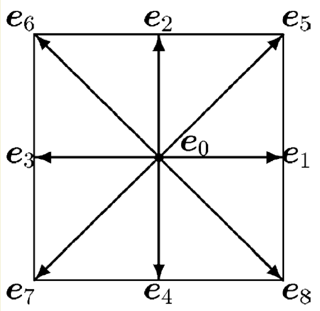
Lattice Boltzmann (LB) Equation

Define: nodes $\mathbf{x}_1, \ldots, \mathbf{x}_n$, discrete velocities $\mathbf{e}_0, \ldots, \mathbf{e}_m$

$$\underbrace{\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t+1) - \mathbf{f}(\mathbf{x}_j, t)}_{\text{advection}} = \underbrace{\mathcal{C}\, \mathbf{f}(\mathbf{x}_j, t)}_{\text{collision}} =: \mathbf{f}^c(\mathbf{x}_j, t)$$

- $\mathbf{f} = (f_0, \ldots f_m)^{\mathrm{T}}$: distribution
- $f_k(\mathbf{x}, t) = f(\mathbf{x}, \mathbf{e}_k, t)$:
  probability that particle is in state $(\mathbf{x}, \mathbf{e}_k)$ at time t
- $\mathcal{C}$: collision operator

D2Q9 Model



$$\delta x = 1, \quad \delta t = 1$$

Discrete velocities:

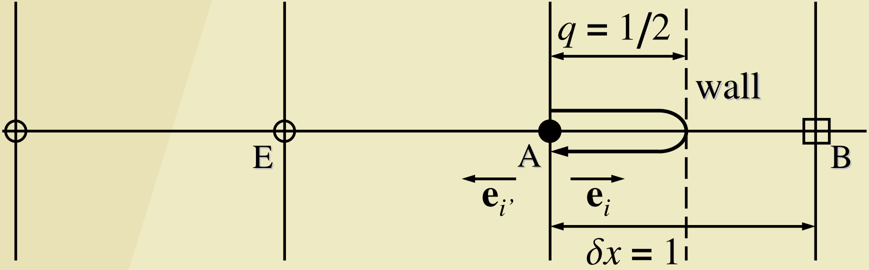
$$\mathbf{e}_k = \begin{cases} \big(0, 0\big), & k = 0, \\ \Big(\cos\big(\tfrac{1}{2}(k-1)\pi\big), \sin\big(\tfrac{1}{2}(k-1)\pi\big)\Big), & k = 1, 2, 3, 4, \\ \Big(\cos\big(\tfrac{1}{4}(2k-9)\pi\big), \sin\big(\tfrac{1}{4}(2k-9)\pi\big)\Big)\sqrt{2}, & k = 5, 6, 7, 8. \end{cases}$$

Previous seminar

Boundary Conditions in LB Methods:

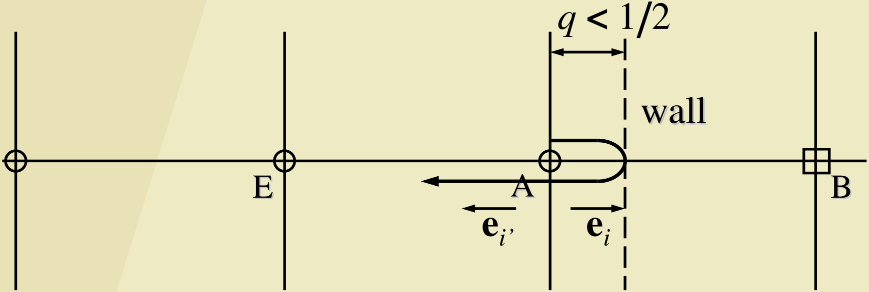
- General formulation of boundary conditions
- Treatment of boundaries:
  - ✓ periodic (infinite domain)
  - ✓ no-slip
  - ✓ free-slip
  - ✓ frictional slip, sliding walls
  - ✓ open (e.g. fluid inlet)
  - ✓ complex geometry

No-Slip: bounce-back

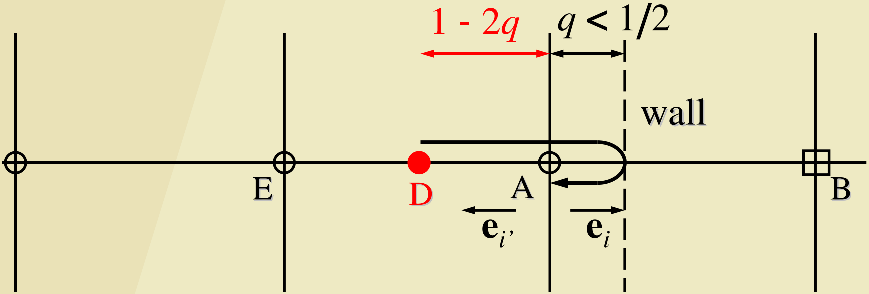


$$f_{i'}(\mathbf{x}_j, t+1) = f_i^c(\mathbf{x}_j, t)$$
$$\mathbf{e}_{i'} = -\mathbf{e}_i$$

No-Slip

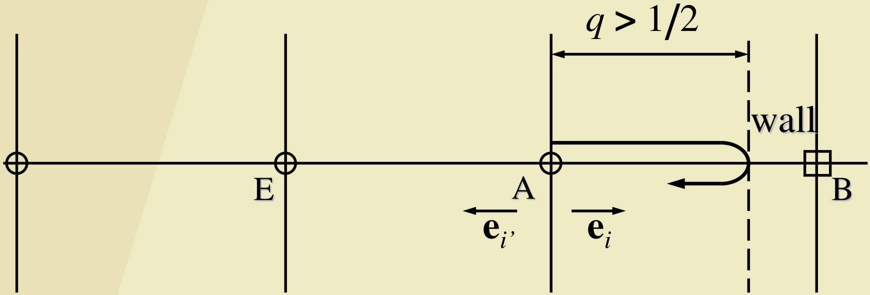


No-Slip: upwind quadratic interpolation

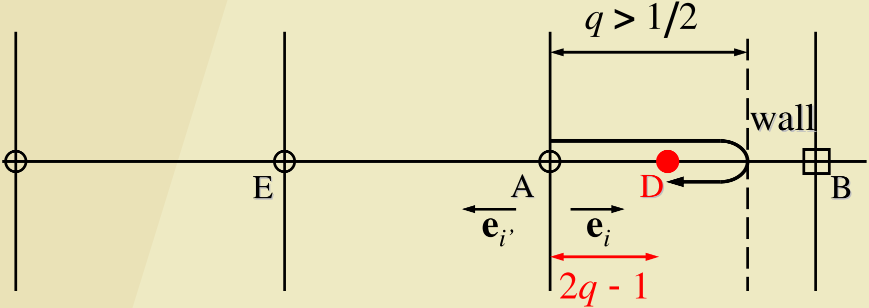


$$f_{i'}(\mathbf{x}_A, t+1) = q(2q+1)f_i^c(\mathbf{x}_A, t) + (1-2q)(1+2q)f_i^c(\mathbf{x}_A - \mathbf{e}_i, t) - q(1-2q)f_i^c(\mathbf{x}_A - 2\mathbf{e}_i, t) \qquad (q < \frac{1}{2})$$

No-Slip



No-Slip: downwind quadratic interpolation



$$f_{i'}(\mathbf{x}_A, t+1) = \frac{1}{q(2q+1)} f_i^c(\mathbf{x}_A, t) + \frac{2q-1}{q} f_{i'}^c(\mathbf{x}_A, t) - \frac{1-2q}{1+2q} f_{i'}^c(\mathbf{x}_A - \mathbf{e}_i, t) \qquad (q \geq \frac{1}{2})$$

Moving Boundary Conditions

$$\begin{aligned}\delta f_{i'}(\mathbf{x}_j, t+1) &= f_{i'}(\mathbf{x}_j, t+1) - \Big(q(2q+1) f_i^c(\mathbf{x}_j, t) \\ &\quad + (1-2q)(1+2q) f_i^c(\mathbf{x}_j - \mathbf{e}_i, t) \\ &\quad - q(1-2q) f_i^c(\mathbf{x}_j - 2\mathbf{e}_i, t) \Big) \quad (q < \frac{1}{2}) (1)\end{aligned}$$

$$\begin{aligned}\delta f_{i'}(\mathbf{x}_j, t+1) &= f_{i'}(\mathbf{x}_j, t+1) - \Big(\frac{1}{q(2q+1)} f_i^c(\mathbf{x}_j, t) \\ &\quad + \frac{2q-1}{q} f_{i'}^c(\mathbf{x}_j, t) \\ &\quad - \frac{1-2q}{1+2q} f_{i'}^c(\mathbf{x}_j - \mathbf{e}_i, t) \Big) \qquad (q \geq \frac{1}{2}) \ (2)\end{aligned}$$

Equilibrium Distribution at Moving Boundary

- first-order equilibrium distribution at boundary:

$$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

  - $\mathbf{e}_w$: velocity of boundary

- weight coefficients:
  - $f_i^0 = \{\frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}\}$
  - $\alpha_i = \{0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}\}$
- conservation ($\rho_0 = 1$):
  - ✓ mass: $\sum_i f_i^{(eq)} = 1$
  - ✓ momentum: $\sum_i \mathbf{e}_i f_i^{(eq)} = \mathbf{e}_w$

Equilibrium Distribution at Moving Boundary

- first-order equilibrium distribution at boundary:

$$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

  - $\mathbf{e}_w$: velocity of boundary

- weight coefficients:
  - $f_i^0 = \{\frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}\}$
  - $\alpha_i = \{0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}\}$

- conservation ($\rho_0 = 1$):
  - ✓ mass: $\sum_i f_i^{(eq)} = 1$
  - ✓ momentum: $\sum_i \mathbf{e}_i f_i^{(eq)} = \mathbf{e}_w$

Equilibrium Distribution at Moving Boundary

- first-order equilibrium distribution at boundary:

$$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

  - $\mathbf{e}_w$: velocity of boundary

- weight coefficients:
  - $f_i^0 = \{\frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}\}$
  - $\alpha_i = \{0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}\}$

- conservation ($\rho_0 = 1$):
  - ✓ mass: $\sum_i f_i^{(eq)} = 1$
  - ✓ momentum: $\sum_i \mathbf{e}_i f_i^{(eq)} = \mathbf{e}_w$

Moving Boundary Conditions

substitution $f_i \longrightarrow f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w,$

$$f_{i'} \longrightarrow f_i^0 - \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

$$(1) \Longrightarrow \delta f_{i'} = 2\alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

$$(2) \Longrightarrow \delta f_{i'} = \frac{2}{q(2q+1)} \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

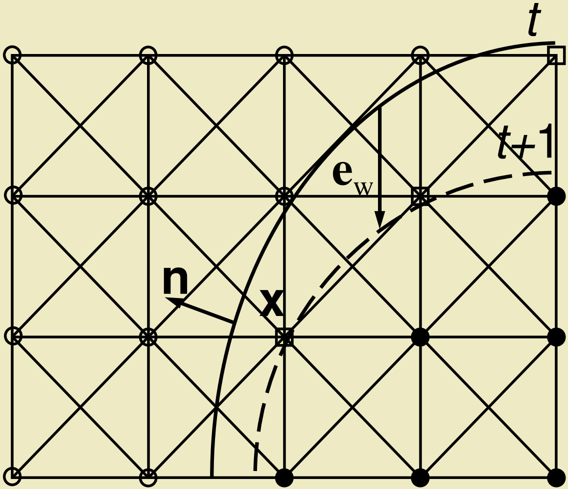
Moving Boundary Conditions

$$f_{i'}(\mathbf{x}_j, t+1) = q(2q+1)f_i^c(\mathbf{x}_j, t) + (1-2q)(1+2q)f_i^c(\mathbf{x}_j - \mathbf{e}_i, t) - q(1-2q)f_i^c(\mathbf{x}_j - 2\mathbf{e}_i, t) + 2\alpha_i \mathbf{e}_i \cdot \mathbf{e}_w, \quad (q < \frac{1}{2})$$

$$f_{i'}(\mathbf{x}_j, t+1) = \frac{1}{q(2q+1)} f_i^c(\mathbf{x}_j, t) + \frac{2q-1}{q} f_{i'}^c(\mathbf{x}_j, t) - \frac{1-2q}{1+2q} f_{i'}^c(\mathbf{x}_j - \mathbf{e}_i, t) + \frac{2}{q(2q+1)} \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w \quad (q \geq \frac{1}{2})$$

Moving Wall

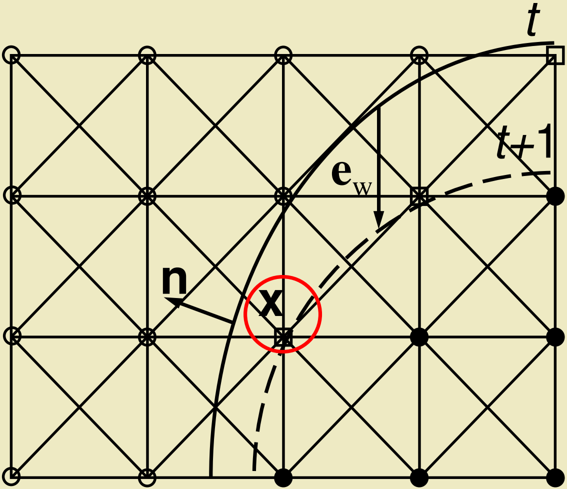
- ○ fluid
- □ solid→fluid
- ● solid



Solid nodes become fluid nodes

Moving Wall

- ○ fluid
- □ solid→fluid
- ● solid



Solid nodes become fluid nodes

Moving Wall

- Unknown distribution in solid→fluid nodes
- Different methods:
  - extrapolation:

$$\mathbf{f}(\mathbf{x}, t) = 3\mathbf{f}(\mathbf{x}', t) - 3\mathbf{f}(\mathbf{x}'', t) + \mathbf{f}(\mathbf{x}'' + \mathbf{e}_k, t)$$

$\mathbf{e}_k$ maximises $\mathbf{n} \cdot \mathbf{e}_k$

Moving Wall

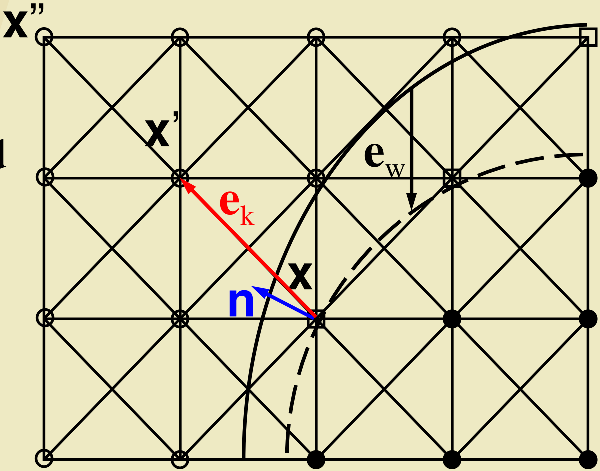
- Unknown distribution in solid→fluid nodes
- Different methods:
  1. extrapolation:

$$\mathbf{f}(\mathbf{x}, t) = 3\mathbf{f}(\mathbf{x}', t) - 3\mathbf{f}(\mathbf{x}'', t) + \mathbf{f}(\mathbf{x}'' + \mathbf{e}_k, t)$$

$\mathbf{e}_k$ maximises $\mathbf{n} \cdot \mathbf{e}_k$

Moving Wall

- ○ fluid
- □ solid→fluid
- ● solid



Moving Wall

- Unknown distribution in solid→fluid nodes
- Different methods:
  1. extrapolation:
     $$\mathbf{f}(\mathbf{x}, t) = 3\mathbf{f}(\mathbf{x}', t) - 3\mathbf{f}(\mathbf{x}'', t) + \mathbf{f}(\mathbf{x}'' + \mathbf{e}_k, t)$$
     $\mathbf{e}_k$ maximises $\mathbf{n} \cdot \mathbf{e}_k$
  2. equilibrium distribution function:
     $$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$
  3. systematically update distribution functions in non-fluid nodes with velocity $\mathbf{e}_w$.
- Methods produce similar results

Moving Wall

- Unknown distribution in solid→fluid nodes
- Different methods:
  1. extrapolation:
     $$\mathbf{f}(\mathbf{x},t) = 3\mathbf{f}(\mathbf{x}',t) - 3\mathbf{f}(\mathbf{x}'',t) + \mathbf{f}(\mathbf{x}''+\mathbf{e}_k,t)$$
     $\mathbf{e}_k$ maximises $\mathbf{n}\cdot\mathbf{e}_k$
  2. equilibrium distribution function:
     $$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$
  3. systematically update distribution functions in non-fluid nodes with velocity $\mathbf{e}_w$.
- Methods produce similar results

Moving Wall

- Unknown distribution in solid→fluid nodes
- Different methods:
  1. extrapolation:
     $$\mathbf{f}(\mathbf{x}, t) = 3\mathbf{f}(\mathbf{x}', t) - 3\mathbf{f}(\mathbf{x}'', t) + \mathbf{f}(\mathbf{x}'' + \mathbf{e}_k, t)$$
     $\mathbf{e}_k$ maximises $\mathbf{n} \cdot \mathbf{e}_k$
  2. equilibrium distribution function:
     $$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$
  3. systematically update distribution functions in non-fluid nodes with velocity $\mathbf{e}_w$.
- Methods produce similar results

Moving Wall

- Unknown distribution in solid→fluid nodes
- Different methods:
  1. extrapolation:

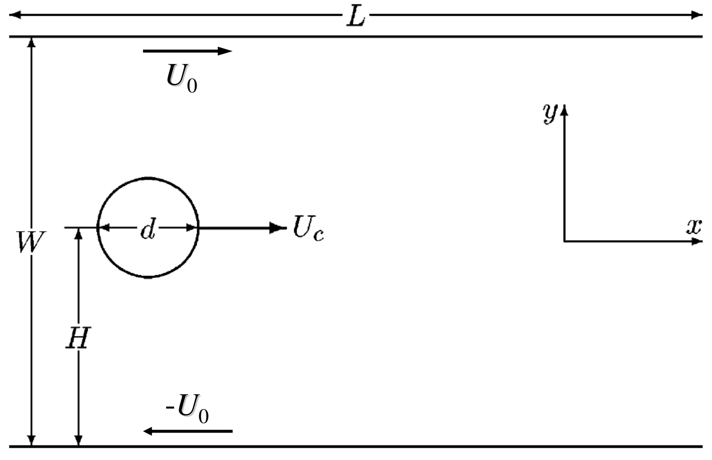
     $$\mathbf{f}(\mathbf{x}, t) = 3\mathbf{f}(\mathbf{x}', t) - 3\mathbf{f}(\mathbf{x}'', t) + \mathbf{f}(\mathbf{x}'' + \mathbf{e}_k, t)$$

     $\mathbf{e}_k$ maximises $\mathbf{n} \cdot \mathbf{e}_k$
  2. equilibrium distribution function:

     $$f_i^{(eq)} = f_i^0 + \alpha_i \mathbf{e}_i \cdot \mathbf{e}_w$$

  3. systematically update distribution functions in non-fluid nodes with velocity $\mathbf{e}_w$.
- Methods produce similar results

Cylinder in Transient Couette Flow



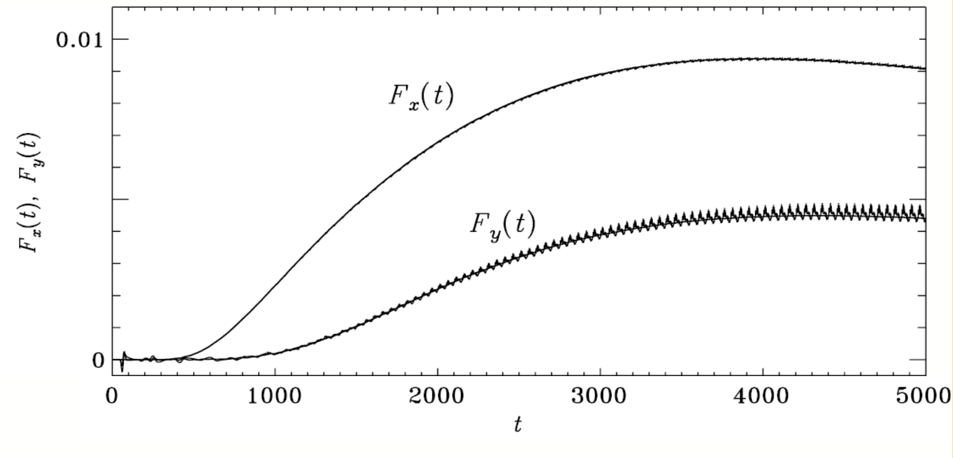
Typically: $d/W = 0.25$, $U_0 = 0.1$, $\mathrm{Re} = 11.36$

Two Reference Frames

- Reference frame at rest $\longleftrightarrow$ cylinder moving with speed U_c w.r.t. mesh
  - Moving bounday
- Reference frame moving with speed U_c $\longleftrightarrow$ cylinder fixed w.r.t. mesh

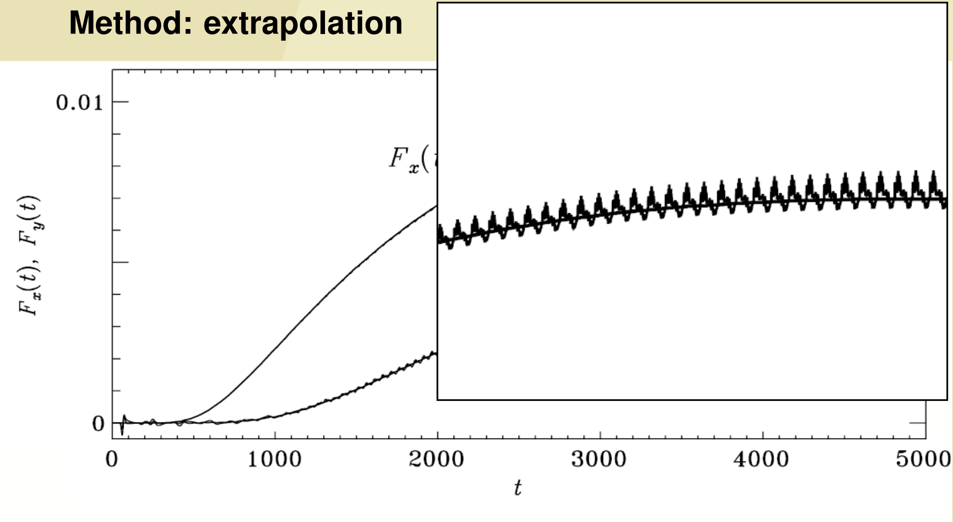
Total Force on Cylinder

Method: extrapolation



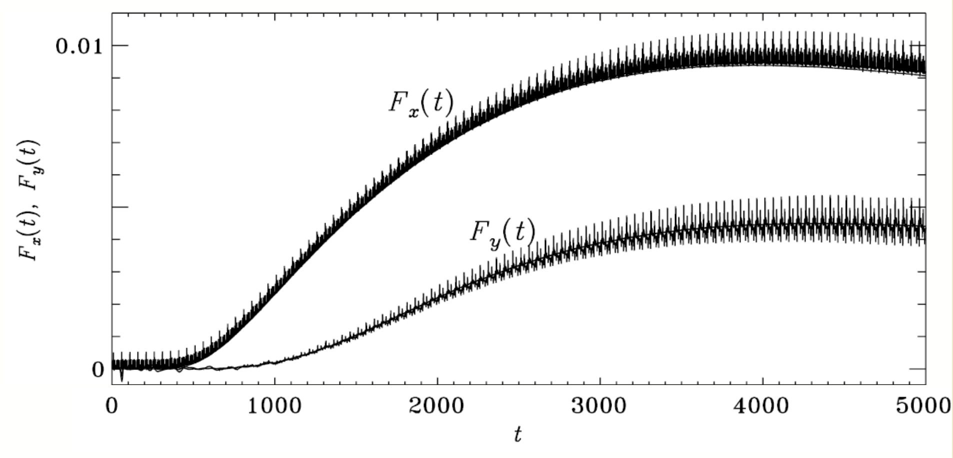
Total Force on Cylinder

Method: extrapolation



Total Force on Cylinder

Method: equilibrium distribution functions



Discussion

- Results for moving and fixed boundary in agreement with each other
- Spatial fluctuation in force
  - number of fluid nodes, hence volume, not conserved
  - error introduced by computing distribution in solid→fluid nodes
  - number of lattice lines (edges) varies

Collision-Advection Process

- Step 1: Compute Moments

$$\mathbf{m}(\mathbf{x}_j, t) = \mathcal{M}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 2: Relaxation

$$\mathbf{m}^c(\mathbf{x}_j, t) = \mathbf{m}(\mathbf{x}_j, t) - \mathbf{S}(\mathbf{m}(\mathbf{x}_j, t) - \mathbf{m}^{(eq)}(\mathbf{x}_j, t))$$

- Step 3: Compute post-collision distributions

$$\mathcal{C}\mathbf{f}(\mathbf{x}_j, t) = \mathcal{M}^{-1}\mathbf{m}^c(\mathbf{x}_j, t)$$

- Step 4: Advection

$$\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t + 1) = \mathbf{f}(\mathbf{x}_j, t) + \mathcal{C}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 5: Moving boundary

Collision-Advection Process

- Step 1: Compute Moments

$$\mathbf{m}(\mathbf{x}_j, t) = \mathcal{M}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 2: Relaxation

$$\mathbf{m}^c(\mathbf{x}_j, t) = \mathbf{m}(\mathbf{x}_j, t) - \mathbf{S}(\mathbf{m}(\mathbf{x}_j, t) - \mathbf{m}^{(\mathrm{eq})}(\mathbf{x}_j, t))$$

- Step 3: Compute post-collision distributions

$$\mathcal{C}\mathbf{f}(\mathbf{x}_j, t) = \mathcal{M}^{-1}\mathbf{m}^c(\mathbf{x}_j, t)$$

- Step 4: Advection

$$\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t + 1) = \mathbf{f}(\mathbf{x}_j, t) + \mathcal{C}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 5: Moving boundary

Collision-Advection Process

- Step 1: Compute Moments

$$\mathbf{m}(\mathbf{x}_j, t) = \mathcal{M}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 2: Relaxation

$$\mathbf{m}^c(\mathbf{x}_j, t) = \mathbf{m}(\mathbf{x}_j, t) - \mathbf{S}(\mathbf{m}(\mathbf{x}_j, t) - \mathbf{m}^{(\text{eq})}(\mathbf{x}_j, t))$$

- Step 3: Compute post-collision distributions

$$\mathcal{C}\mathbf{f}(\mathbf{x}_j, t) = \mathcal{M}^{-1}\mathbf{m}^c(\mathbf{x}_j, t)$$

- Step 4: Advection

$$\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t + 1) = \mathbf{f}(\mathbf{x}_j, t) + \mathcal{C}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 5: Moving boundary

Collision-Advection Process

- Step 1: Compute Moments

$$\mathbf{m}(\mathbf{x}_j, t) = \mathcal{M}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 2: Relaxation

$$\mathbf{m}^c(\mathbf{x}_j, t) = \mathbf{m}(\mathbf{x}_j, t) - \mathbf{S}(\mathbf{m}(\mathbf{x}_j, t) - \mathbf{m}^{(\mathrm{eq})}(\mathbf{x}_j, t))$$

- Step 3: Compute post-collision distributions

$$\mathcal{C}\mathbf{f}(\mathbf{x}_j, t) = \mathcal{M}^{-1}\mathbf{m}^c(\mathbf{x}_j, t)$$

- Step 4: Advection

$$\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t + 1) = \mathbf{f}(\mathbf{x}_j, t) + \mathcal{C}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 5: Moving boundary

Collision-Advection Process

- Step 1: Compute Moments

$$\mathbf{m}(\mathbf{x}_j, t) = \mathcal{M}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 2: Relaxation

$$\mathbf{m}^c(\mathbf{x}_j, t) = \mathbf{m}(\mathbf{x}_j, t) - \mathbf{S}(\mathbf{m}(\mathbf{x}_j, t) - \mathbf{m}^{(\mathrm{eq})}(\mathbf{x}_j, t))$$

- Step 3: Compute post-collision distributions

$$\mathcal{C}\mathbf{f}(\mathbf{x}_j, t) = \mathcal{M}^{-1}\mathbf{m}^c(\mathbf{x}_j, t)$$

- Step 4: Advection

$$\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t + 1) = \mathbf{f}(\mathbf{x}_j, t) + \mathcal{C}\mathbf{f}(\mathbf{x}_j, t)$$

- Step 5: Moving boundary

Step 1: Compute Moments

Linear transformation:

$$\mathbf{m} = \mathcal{M}\mathbf{f}, \qquad \mathbf{m} = (\rho, e, \epsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy})^{\mathrm{T}}$$

- ρ: density
- e: related to kinetic energy
- ϵ: related to kinetic energy square
- j_x, j_y: components of momentum density
- q_x, q_y: components of energy flux
- p_{xx}, p_{xy}: components of stress tensor

Transformation Matrix

Transformation from phase space to moment space

$$\mathcal{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}$$

Step 2: Relaxation

Relaxation equation

$$\mathbf{m}^c = \mathbf{m} - \mathbf{S}(\mathbf{m} - \mathbf{m}^{(\mathrm{eq})})$$

- $\mathbf{m}^c$: post-collision state
- $\mathbf{m}^{(\mathrm{eq})}$: equilibrium state
- $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_9)$: diagonal relaxation matrix
  - ✓ $\rho,\ j_x,\ j_y$ conserved $\longrightarrow\ s_1, s_4, s_6 = 0$
  - ✓ $e,\ \epsilon,\ q_x,\ q_y,\ p_{xx},\ p_{yy}$ non-conserved

    $\longrightarrow$ Lattice BGK model: $s_k = \frac{1}{\tau}, \quad k = 2, 3, 5, 7, 8, 9$

    - $\tau =$ relaxation time

Step 2: Relaxation

Relaxation equation

$$\mathbf{m}^c = \mathbf{m} - \mathbf{S}(\mathbf{m} - \mathbf{m}^{(\mathrm{eq})})$$

- $\mathbf{m}^c$: post-collision state
- $\mathbf{m}^{(\mathrm{eq})}$: equilibrium state
- $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_9)$: diagonal relaxation matrix
  - ✓ $\rho,\ j_x,\ j_y$ conserved $\longrightarrow\ s_1, s_4, s_6 = 0$
  - ✓ $e,\ \epsilon,\ q_x,\ q_y,\ p_{xx},\ p_{yy}$ non-conserved

    $\longrightarrow$ Lattice BGK model: $s_k = \frac{1}{\tau}, \quad k = 2, 3, 5, 7, 8, 9$

    - τ = relaxation time

Step 2: Relaxation

Relaxation equation

$$\mathbf{m}^c = \mathbf{m} - \mathbf{S}(\mathbf{m} - \mathbf{m}^{(\mathrm{eq})})$$

- $\mathbf{m}^c$: post-collision state
- $\mathbf{m}^{(\mathrm{eq})}$: equilibrium state
- $\mathbf{S} = \mathrm{diag}(s_1, \dots, s_9)$: diagonal relaxation matrix
  - ✓ $\rho,\ j_x,\ j_y$ conserved $\longrightarrow\ s_1, s_4, s_6 = 0$
  - ✓ $e,\ \epsilon,\ q_x,\ q_y,\ p_{xx},\ p_{yy}$ non-conserved

    $\longrightarrow$ Lattice BGK model: $s_k = \frac{1}{\tau}, \quad k = 2, 3, 5, 7, 8, 9$

    - τ = relaxation time

Equilibrium Distribution Functions

- Depend only on conserved moments ρ, j_x, j_y
- Kinetic theory for Maxwell molecules:

$$
\begin{aligned}
e^{(\mathrm{eq})} &= -2\rho + \frac{3}{\rho}(j_x^2 + j_y^2) \\
\epsilon^{(\mathrm{eq})} &= \rho - \frac{3}{\rho}(j_x^2 + j_y^2) \\
q_x^{(\mathrm{eq})} &= -j_x, \qquad q_y^{(\mathrm{eq})} = -j_y \\
p_{xx}^{(\mathrm{eq})} &= \frac{1}{\rho}(j_x^2 - j_y^2), \qquad p_{xy}^{(\mathrm{eq})} = \frac{1}{\rho} j_x j_y
\end{aligned}
$$

P. Lallemand and L.-S. Luo

Step 3: Compute Post-Collision Distributions

Post-collision distributions

$$\mathcal{C}\mathbf{f}(\mathbf{x}_j, t) = \mathcal{M}^{-1}\mathbf{m}^c(\mathbf{x}_j, t)$$

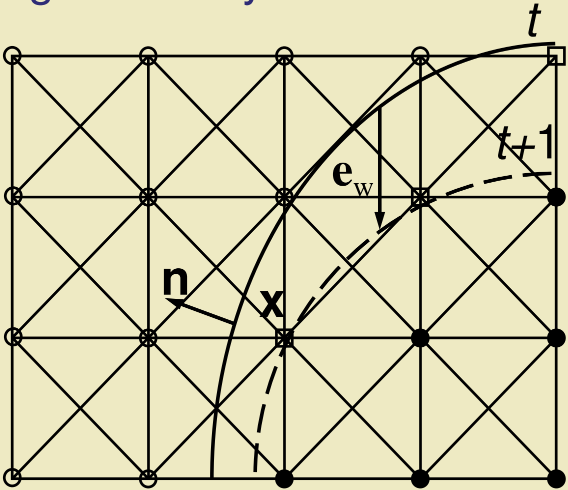
Step 4: Advection

Advected distributions

$$\mathbf{f}(\mathbf{x}_j + \mathbf{e}_k, t + 1) = \mathbf{f}(\mathbf{x}_j, t) + \mathcal{C}\mathbf{f}(\mathbf{x}_j, t)$$

Step 5: Moving Boundary

- ○ fluid
- □ solid→fluid
- ● solid



References

- M. Bouzidi, M. Firdaouss and P. Lallemand
  Momentum Transfer of a Boltzmann-Lattice Fluid with Boundaries
  2001.
- P. Lallemand and L.-S. Luo
  Theory of the Lattice Boltzmann Method: Dispersion, Dissipation, Isotropy, Galilean Invariance and Stability
  2000.
- P. Lallemand and L.-S. Luo
  Lattice Boltzmann Method for Moving Boundaries
  2003.