

Lattice Boltzmann, Part 2

Fall Semester

Example

- Use D2Q9 incompressible model.
- Grid is 32×16 where $\Delta x = 1.0/16.0 = 0.0625$.
- Viscosity is $\nu = 1.0$. Relaxation parameter is $\omega = 0.8$.
-
- Pressure is $p = 1.1$ at left edge.
- Pressure is $p = 1.0$ at right edge.
- Velocity is $v = 0.0$ at top edge.
- Velocity is $v = 0.0$ at bottom edge.

Initial Conditions

- Pressure is $p = 1.05$ at all interior points.
- Velocity is $v = 0.0$ at all interior points.
- Distribution $g^1 = g^0$ at all interior points.
- Calculate equilibrium g^0 from v and p .

$$e_i = \langle x_i, y_i \rangle$$
$$s_i(v) = \hat{w}_i \left(3 \frac{e_i \cdot v}{c} + 4.5 \frac{(e_i \cdot v)^2}{c^2} - 1.5 \frac{v \cdot v}{c^2} \right)$$
$$g_i^0 = \{-4\sigma, \lambda, \lambda, \lambda, \lambda, \gamma, \gamma, \gamma, \gamma\} \frac{p}{c^2} + s_i(v)$$

Parameters

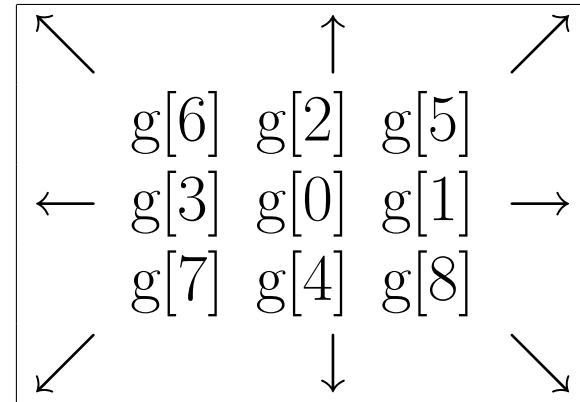
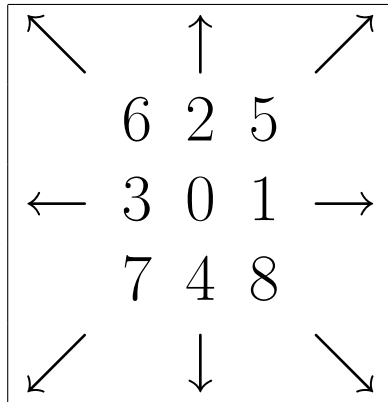
- $\Delta x = 0.0625$
- $\nu = 1.0$
- $\omega = 0.8$
-
- $\Delta t = \frac{(\Delta x)^2}{\nu} \cdot \frac{2-\omega}{6\omega}$
- $c = \Delta x / \Delta t$
- $\sigma = 5/12$
- $\lambda = 1/3$
- $\gamma = 1/12$

Main Loop

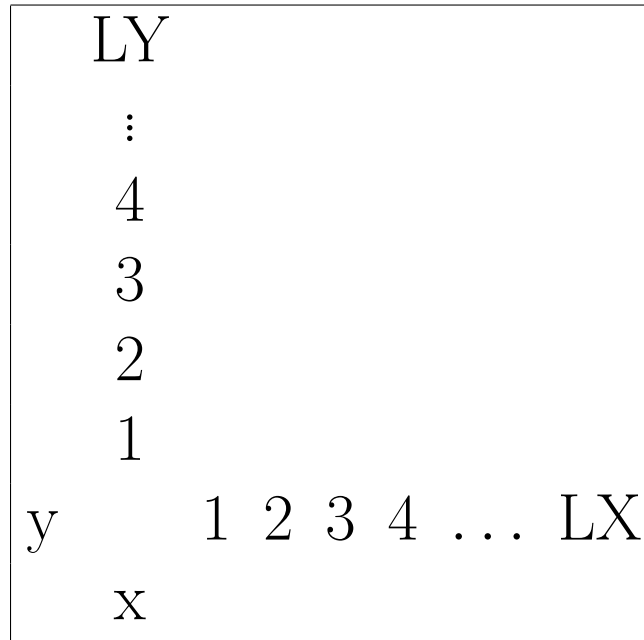
1. Collision.
2. Boundary conditions.
3. Streaming.
4. Update flow field.

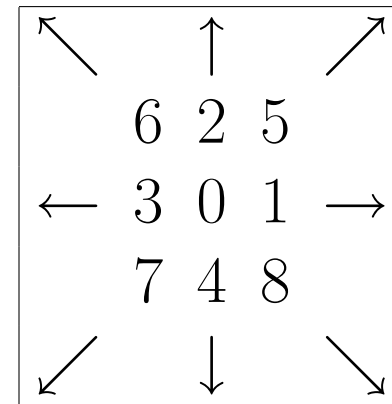
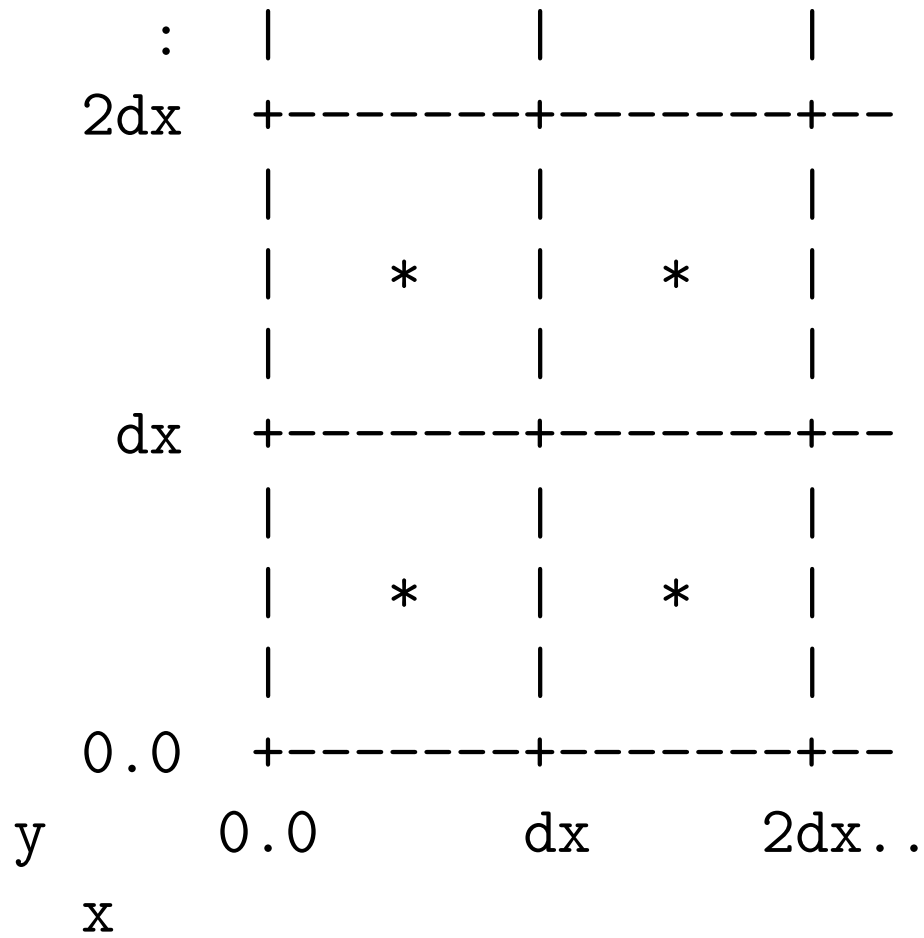
- Repeat until velocity stops changing:

$$\frac{\Sigma (v - v_{old})^2}{\Sigma v^2} < 10^{-6}$$

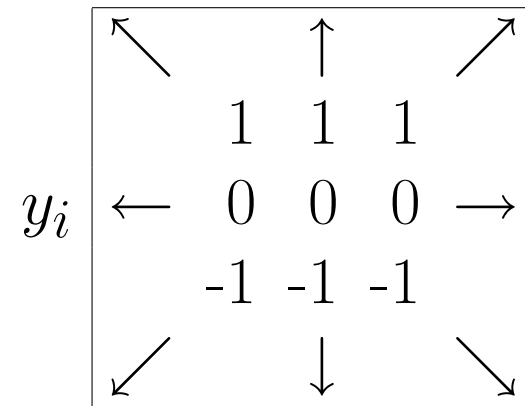
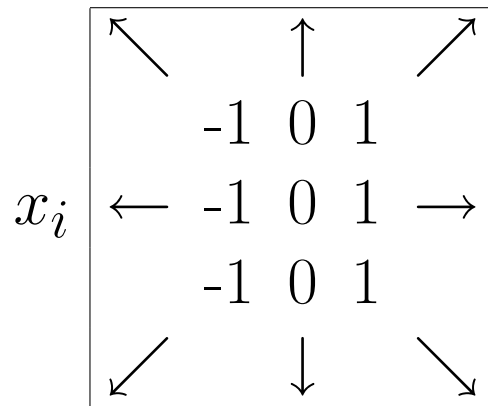
Distribution, g 

Grid





Vectors



Weights

$$w_i \begin{array}{c} \swarrow \quad \uparrow \quad \searrow \\ 1 \quad 4 \quad 1 \\ \leftarrow 4 \quad 16 \quad 4 \rightarrow \\ 1 \quad 4 \quad 1 \\ \swarrow \quad \downarrow \quad \searrow \end{array}$$

$$\hat{w}_i \begin{array}{c} \swarrow \quad \uparrow \quad \searrow \\ 1/36 \quad 1/9 \quad 1/36 \\ \leftarrow 1/9 \quad 4/9 \quad 1/9 \rightarrow \\ 1/36 \quad 1/9 \quad 1/36 \\ \swarrow \quad \downarrow \quad \searrow \end{array}$$

Flow Field

- Velocity components sum over each g .

$$v_x = \sum_i (cx_i g_i)$$

$$v_y = \sum_i (cy_i g_i)$$

- Incompressible. Pressure also a sum.

$$p = \frac{c^2}{4\sigma} \cdot \left(\frac{1.5\hat{w}_0 (v \cdot v)}{c^2} + \sum_{i \neq 0} g_i \right)$$

Main Loop

1. Collision.

- Calculate g^0 from v and p , then g^2 from g^1 and g^0 .

2. Boundary conditions.

- Specify values in g^2 .

3. Streaming.

- Update g^1 from g^2 .

4. Update flow field.

- Calculate v and p from g^1 .

Collision

- We already know g^1 and v and p .
- Calculate g^0 from v and p .

$$e_i = \langle x_i, y_i \rangle$$
$$s_i(v) = \hat{w}_i \left(3 \frac{e_i \cdot v}{c} + 4.5 \frac{(e_i \cdot v)^2}{c^2} - 1.5 \frac{v \cdot v}{c^2} \right)$$
$$g_i^0 = \{-4\sigma, \lambda, \lambda, \lambda, \lambda, \gamma, \gamma, \gamma, \gamma\} \frac{p}{c^2} + s_i(v)$$

- Then $g^2 = (1 - \omega) \cdot g^1 + \omega \cdot g^0$.
 - If $\omega > 1$ then this is an overrelaxation.

Boundary Conditions (1)

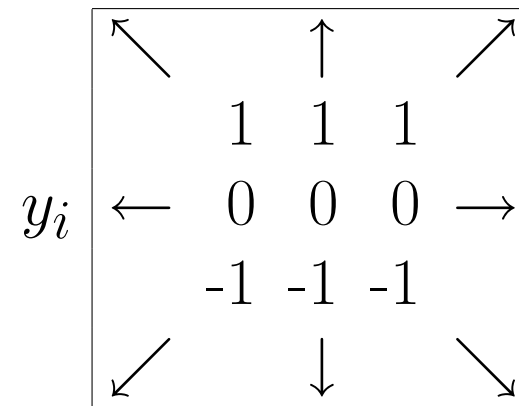
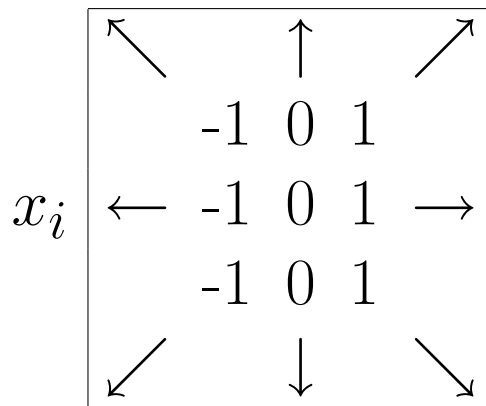
- Assume no internal walls for now.
-
- At left and right edge where pressure p is proscribed:
 - Set v from grid and p as specified.
-
- At top and bottom edge where velocity v is proscribed:
 - Set v as specified and p from grid.
-
- Regardless of direction go horizontal vertical only.

Boundary Conditions (2)

- Using the identified v and p , then find g .
-
- Set $g^2 = g + (1 - \omega) \cdot (g^1 - g^0)$.
-
- Only required for directions that will stream *into* the field.
-
- Trick
 - Put an extra row and column on each side of the grid.
 - Looping over actual field will access these extra cells.

Streaming

- Set g^1 from *directional* value of g^2 , so that only $g[0]$ stays put.
- Boundary cells only need values that stream *into* the field.



Update Flow Field

- Velocity components sum over each g .

$$v_x = \sum_i (cx_i g_i)$$

$$v_y = \sum_i (cy_i g_i)$$

- Incompressible. Pressure also a sum.

$$p = \frac{c^2}{4\sigma} \cdot \left(\frac{1.5\hat{w}_0 (v \cdot v)}{c^2} + \sum_{i \neq 0} g_i \right)$$

Lab Assignment: Visualize

- Plot y versus v_x at horizontal midline in Gnuplot.
-
- Plot $|v|$ contours in Matlab.

```
X=load('xfield.txt');  
Y=load('yfield.txt');  
V=load('vfield.txt');  
%  
contour(X,Y,V);  
colorbar;
```