Master Thesis Experiment Report III : Inlet Architecture

Jérémie Guy

September 16, 2024

Abstract

Assessment of different input structures in the Lattice-Boltzmann system array.

1 Context

In this experiment, we will try different architectures for the inlet section of the system. As a recall from previous Experiment reports the inlet missing values (input into the system) are calculated following Zu-He method.

2 Problem

We have different possible architectures for the corners of the inlet border : either make the corners inlets or leave them as bounceback nodes.

3 Experiments

To assess the corner cases, we constructed a small 7x5 system with bounceback nodes on the top and bottom borders, an obstacle in the middle, the inlet on the right border and the inlet on the left border. Both cases are illustrated in Figures 1 and 2. The input velocity is a parabola (which corresponds to the expected velocity profile for a tube LB system). We will be running the simulation through 100 iterations.

We will monitor the PDFs of the whole system throughout the simulation, as well as the horizontal and vertical velocity of each lattice site for the following experiments. All the results have been compiled into two separate text files. The structure of the file for the second case is illustrated in figure 3

3.1 Experiment 1

3.1.1 Description

The first experiment follows the system illustrated in figure 1. Only the 3 centered leftmost nodes are defined as inlets. The execution is as in previous tests following the Zu-he method.

3.1.2 Results

The results have been compiled in a provided file.

3.2 Experiment 2

3.2.1 Description

The second experiment follows the system illustrated in figure 2. For this case, the corners have to be treated separately from the rest. The directions of the PDFs incoming for each lattice site are illustrated in Figure 4. The two corners have five missing directions: for the top left corner, we are missing the incoming PDfs of directions 0,1,2,5,8, and for the bottom left corner, we are missing the

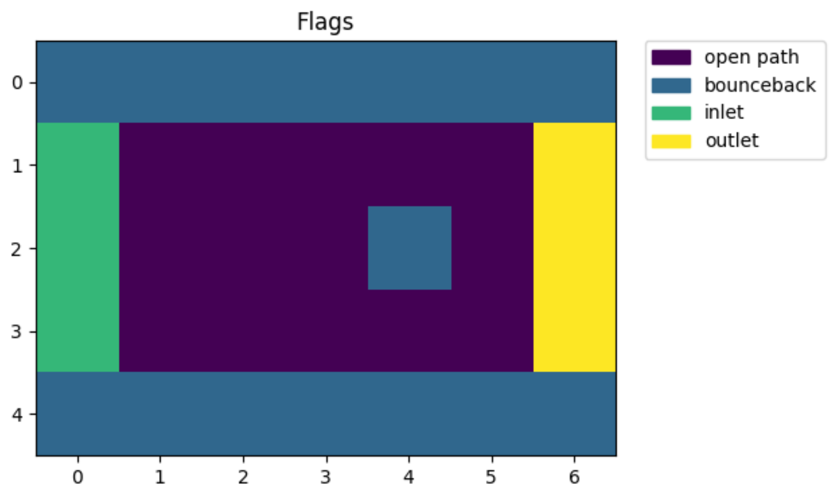


Figure 1: Corners as bounceback nodes

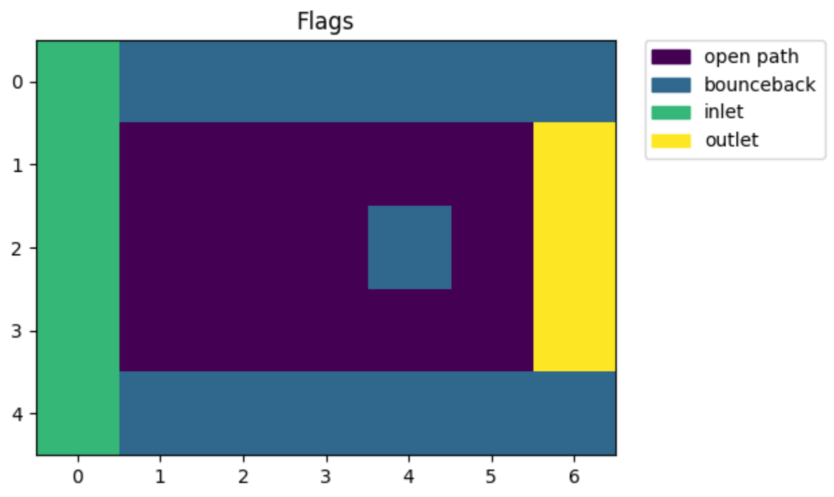


Figure 2: Corners as inlet nodes

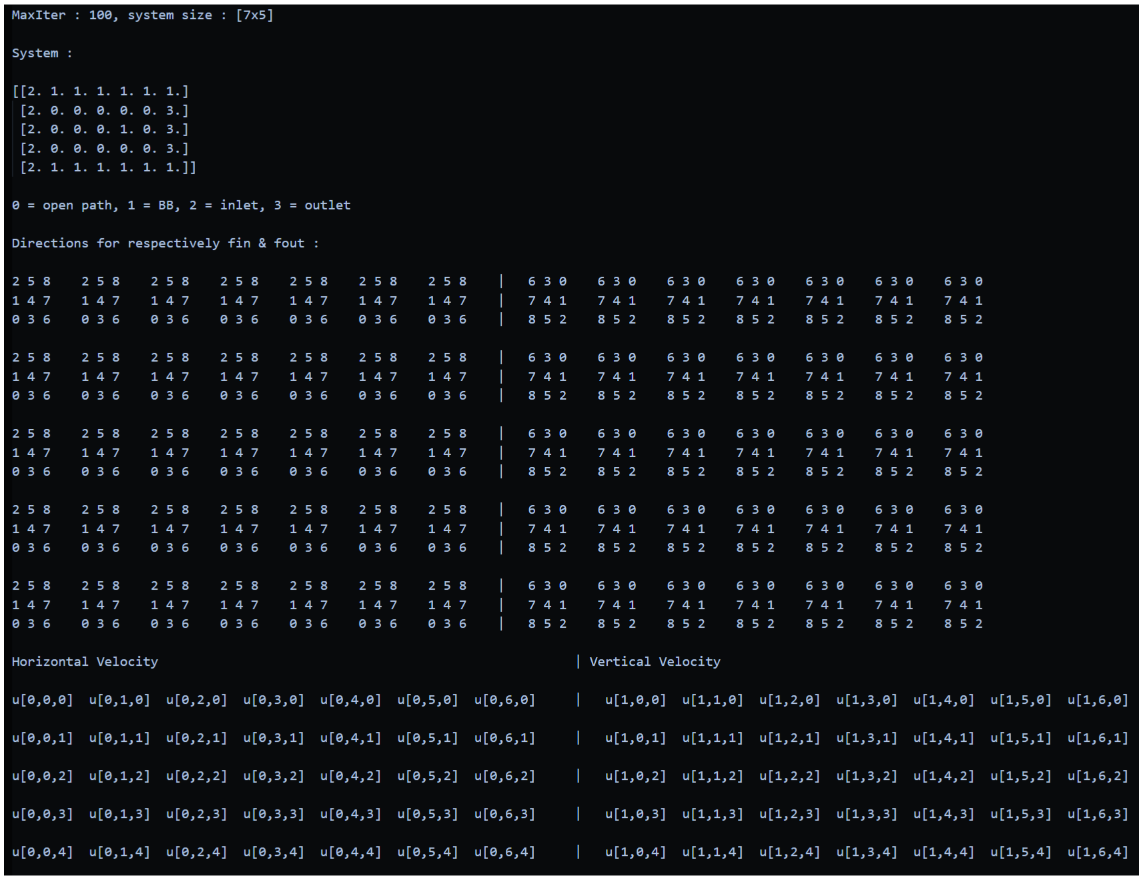
PDFs of directions 0,1,2,3,6.

The issue with the Zu-he method here is that we can only compute 3 missing directions of incoming PDFs. Instead, we artificially defined the missing directions for the corners. For example, for the upper corner, as illustrated in Figure 5, we compute the missing PDFs for the lattice site (0;1). We then take the values obtained and paste them in the missing directions of site (0;0). Figure 5 shows where the direction has been copied. To recall, $fin(i, x, y)$ is the PDFs input for a lattice site of coordinate (x;y) at direction i. The filled arrows show the PDFs calculated with Zu-He, and the dotted arrows show where the values have been copied. The colors indicate where every copied direction has been taken.

3.2.2 Results

4 Results

The results have been compiled in a provided file.



```
MaxIter : 100, system size : [7x5]

System :

[[2. 1. 1. 1. 1. 1. 1.]
 [2. 0. 0. 0. 0. 0. 3.]
 [2. 0. 0. 0. 1. 0. 3.]
 [2. 0. 0. 0. 0. 0. 3.]
 [2. 1. 1. 1. 1. 1. 1.]]

0 = open path, 1 = BB, 2 = inlet, 3 = outlet

Directions for respectively fin & fout :

2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   |   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0
1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   |   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1
0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   |   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2

2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   |   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0
1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   |   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1
0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   |   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2

2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   |   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0
1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   |   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1
0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   |   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2

2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   |   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0
1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   |   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1
0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   |   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2

2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   2 5 8   |   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0   6 3 0
1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   1 4 7   |   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1   7 4 1
0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   0 3 6   |   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2   8 5 2

Horizontal Velocity                                                   | Vertical Velocity

u[0,0,0]  u[0,1,0]  u[0,2,0]  u[0,3,0]  u[0,4,0]  u[0,5,0]  u[0,6,0]    |   u[1,0,0]  u[1,1,0]  u[1,2,0]  u[1,3,0]  u[1,4,0]  u[1,5,0]  u[1,6,0]

u[0,0,1]  u[0,1,1]  u[0,2,1]  u[0,3,1]  u[0,4,1]  u[0,5,1]  u[0,6,1]    |   u[1,0,1]  u[1,1,1]  u[1,2,1]  u[1,3,1]  u[1,4,1]  u[1,5,1]  u[1,6,1]

u[0,0,2]  u[0,1,2]  u[0,2,2]  u[0,3,2]  u[0,4,2]  u[0,5,2]  u[0,6,2]    |   u[1,0,2]  u[1,1,2]  u[1,2,2]  u[1,3,2]  u[1,4,2]  u[1,5,2]  u[1,6,2]

u[0,0,3]  u[0,1,3]  u[0,2,3]  u[0,3,3]  u[0,4,3]  u[0,5,3]  u[0,6,3]    |   u[1,0,3]  u[1,1,3]  u[1,2,3]  u[1,3,3]  u[1,4,3]  u[1,5,3]  u[1,6,3]

u[0,0,4]  u[0,1,4]  u[0,2,4]  u[0,3,4]  u[0,4,4]  u[0,5,4]  u[0,6,4]    |   u[1,0,4]  u[1,1,4]  u[1,2,4]  u[1,3,4]  u[1,4,4]  u[1,5,4]  u[1,6,4]
```

Figure 3: File structure for corners as inlet

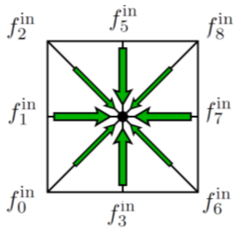


Figure 4: Directional PDFs input in a D2Q9 lattice, **Source :** [1]

4.1 Experiment 2

4.1.1 Description

There is still the ongoing problem of accumulated velocities increasing throughout the iterations on a BB node. The theory that we concluded behind this phenomena is the increasing input PDFs from Zu-He is responsible for a cumulative velocity. To test that, we computed the velocity differently on two chosen BB nodes.

Normally, the velocity is a macrovariable computed with the density ρ. ρ is given with the following formula :

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t) \tag{1}$$

With f_i the input PDFs for cell $\mathbf{x}$. This computes over the whole population for every direction. Instead, here, we compute ρ on a single BB node with only the relevant directions (facing the open path of the system, directions in which PDFs are bound to come before being bounced back). We chose 2 nodes to test it : $\mathbf{x1}$ = (1,0) and $\mathbf{x2}$ = (2,4). The first is at the bottom of the system, so we

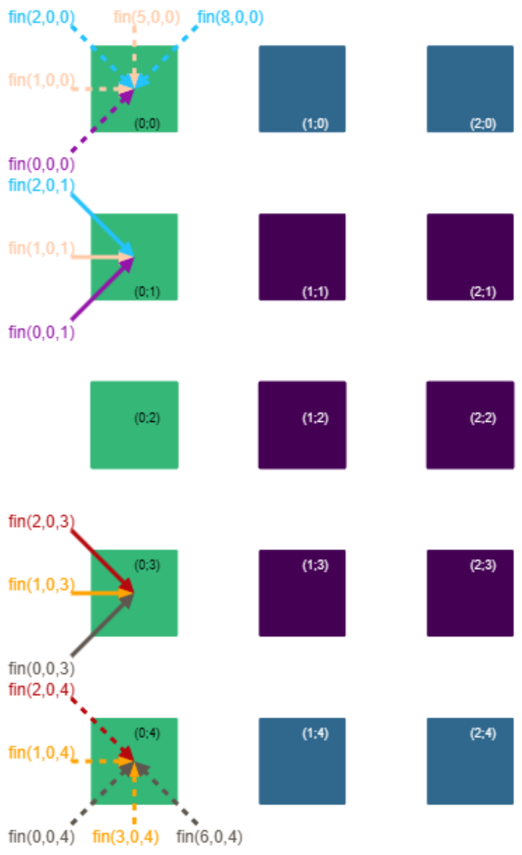


Figure 5: PDfs for the missing corners

only consider the directions 2, 5 & 8 (refer to Figure 4 for the PDFs directions). The second is at the top and the relevant directions are 0, 3 & 6. The equation becomes :

1. for x1 : $\rho(1,0) = fin[2,1,0] + fin[5,1,0] + fin[8,1,0]$
2. for x2 : $\rho(2,4) = fin[0,2,4] + fin[3,2,4] + fin[6,2,4]$

We then calculate the velocity using the density obtained. The velocity formula is as follows:

$$\mathbf{u}(\mathbf{x},t) = \frac{\sum_i f_i(\mathbf{x},t)\mathbf{e}_i}{\rho(\mathbf{x},t)} \tag{2}$$

With e_i the velocity vector as illustrated in Figure 6. This vector has 2 directions for each value, so we will separate them to isolate horizontal velocity $u[0,x]$ from vertical velocity $u[1,x]$. The computed velocities will then be, following the same logic as the density :

1. for u[0,x1] : $u(0,1,0) = \frac{fin[2,1,0]*e_2[0]+fin[5,1,0]*e_5[0]+fin[8,1,0]*e_8[0]}{\rho(1,0)}$
2. for u[1,x1] : $u(1,1,0) = \frac{fin[2,1,0]*e_2[1]+fin[5,1,0]*e_5[1]+fin[8,1,0]*e_8[1]}{\rho(1,0)}$
3. for u[0,x2] : $u(0,1,0) = \frac{fin[0,2,4]*e_0[0]+fin[3,2,4]*e_3[0]+fin[6,2,4]*e_6[0]}{\rho(1,0)}$
4. for u[1,x2] : $u(1,1,0) = \frac{fin[0,2,4]*e_0[1]+fin[3,2,4]*e_3[1]+fin[6,2,4]*e_6[1]}{\rho(1,0)}$

4.1.2 Results

We compiled the results over 500 iterations illustrated in Figure 7. We can observe that the horizontal velocities behave as expected, once the system stabilizes, there is no more continuous increase in velocity. This confirms the theory that the source is Zu-He. However, the vertical velocity seems to be a constant : some investigation is necessary.

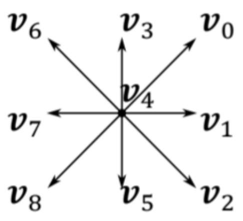
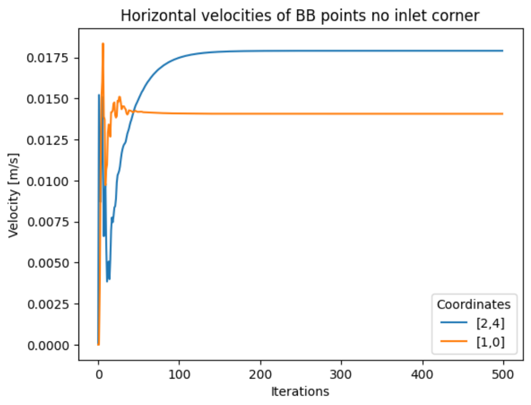
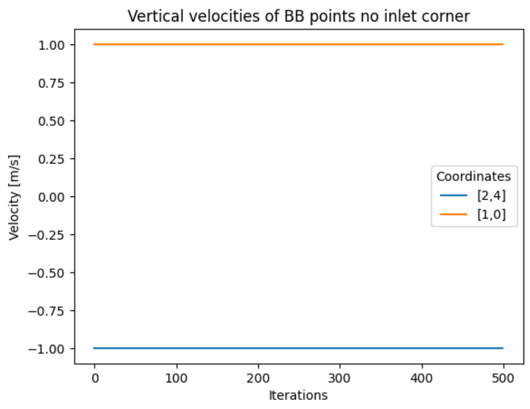


Figure 6: Velocity vector



(a) Horizontal Velocities



(b) Vertical Velocities

Figure 7: Horizontal & vertical velocities, no inlet corner

4.2 Experiment 3

4.2.1 Description

We will now repeat the exact same experiment as the previous one only this time changing the behavior of the inlet corners as described in previous sections.

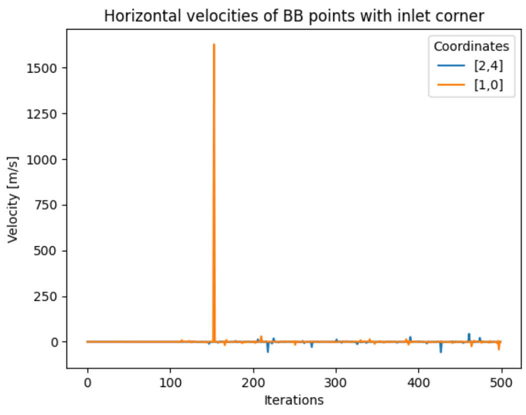
4.2.2 Results

The results here are less than satisfactory. There is a large increase in the velocity for the horizontal velocity on iteration 160, as illustrated in Figure 8a. We tried some debugging to understand the cause : it seems that there is a memory leak going on somewhere that causes the population to increase exponentially throughout the iterations. The spike here is just a side effect. This brings us to the conclusion that the inlet corner method used to artificially input PDFs values does not work as expected. More research will be necessary.

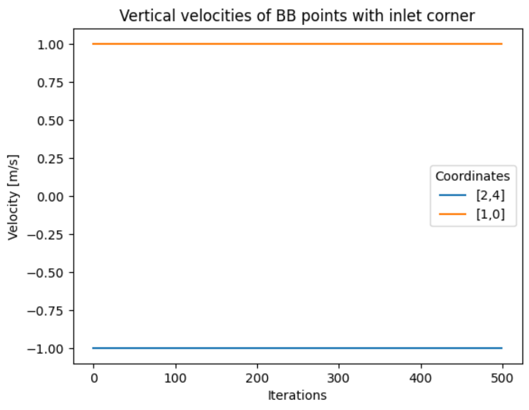
5 Conclusion

6 References

[1] Jonas Lätt. "La méthode de Boltzmann sur réseau pour la simulation des fluides". In: ().



(a) Horizontal Velocities



(b) Vertical Velocities

Figure 8: Horizontal & vertical velocities, with inlet corner