Lattice-Boltzmann Method in Acoustics – Prof. Andrey R. Silva

Lucas Schroeder - Mar. 25th, 2021

Exercise List IV
Not-aligned Walls

PART 1 – KÁRMÁN VORTEX STREET – PROBLEM DESCRIPTION

In the first part of this work, a 2DQ9 lattice-Boltzmann scheme will be used to simulate a 2D flow around a cylinder generating von Kármán vortex street. The scheme adopts the BGK collision operator with an arbitrary relaxation frequency $\omega = 1.88$ and it runs in a 2D lattice space of 302 by 152 cells as presented below. The cylinder has radius $a = 15\ latticeLenght$ and is centered at $(x, y) = (80, 75)$, and the pressure and velocity measurements are taken as the average of a line $100\ latticeLengths$ from the center of the cylinder.

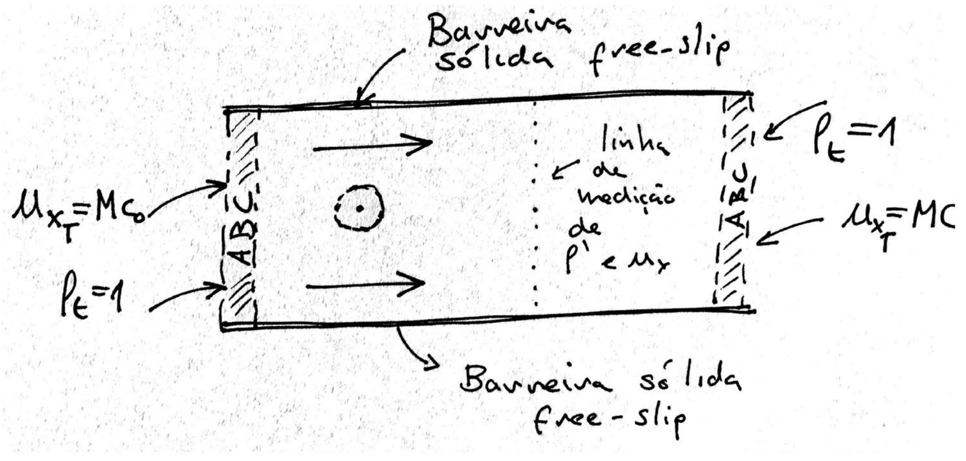

Barreira sólida free-slip
$\rho_t=1$
$u_{x_T}=Mc_0$
ABC
linha de medição de ρ' e u_x
ABC
$u_{x_T}=MC$
$\rho_t=1$
Barreira sólida free-slip

Figure 1 - Simulation case sketch

The object of study is the way we implement solid boundaries not aligned with the computational grid and assess its impact on the solution. Section 1.1 will show the procedure for the standard bounce-back method and Section 1.2 will introduce enhancement proposed by Bouzidi et at. (2003). We will evaluate the frequency of vertex shedding f_v and use it to calculate the Strouhal number, defined as $St = f_v D/U$, where D is the channel height and U is the nominal inlet velocity.

The computational domain is bounded by free-slip walls at the top and bottom, implemented using simple specular reflection. The inlet and outlet boundaries have absorbent boundary conditions (ABS) with target density of 1 and target velocity in x direction of Mach $M = 0.1$. As for the cylinder's wall, two BC will be tested, as discussed in the paragraph above. Figure 2 shows the position of the walls (thick black lines) and the ABC, colored according to the local σ value.

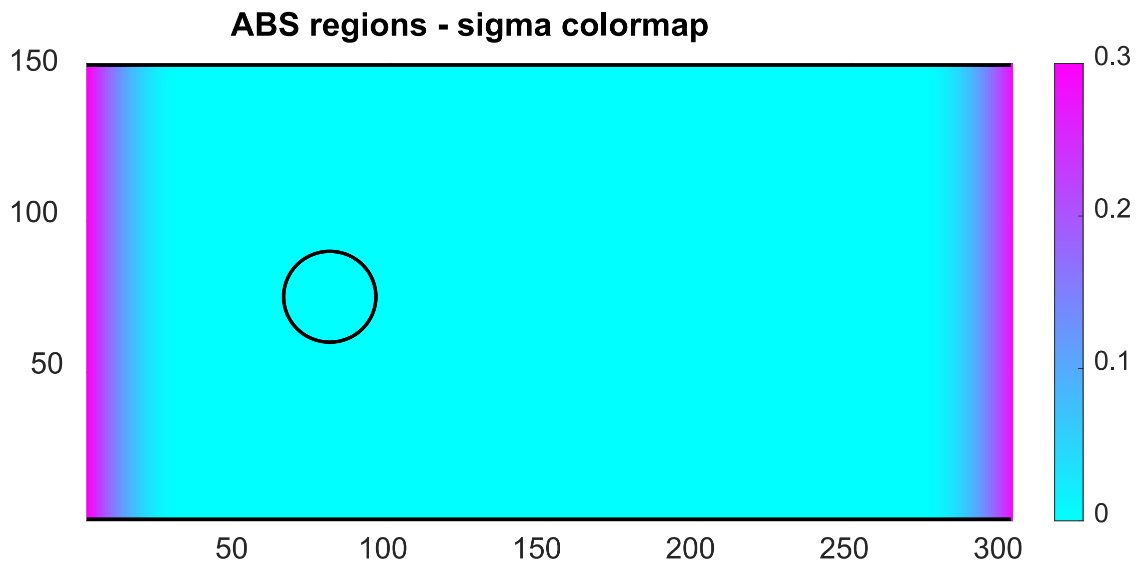
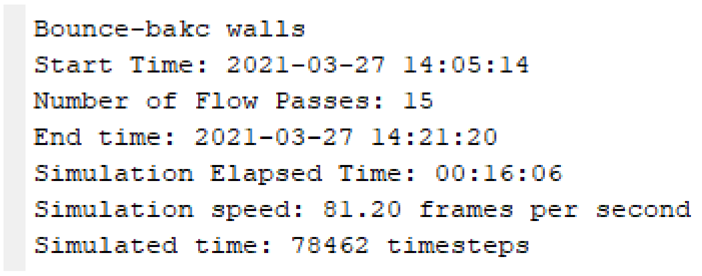


Figure 2 - Computational domain boundary conditions

1.1 Simple Bounce-Back

The first test will use a simple bounce-back scheme to implement the cylinder's wall. It takes some time for the flow to reach steady-state von Kármán vortex street, so we used a transient time of 15 flow passes until the velocity and pressure measurements became periodic. We did this only one time and saved the result in a Matlab .mat file that was used as seed for the next step of measurements. This procedure was useful because, as we can see in the figure below, it took 16 minutes to compute.



```
Bounce-bakc walls
Start Time: 2021-03-27 14:05:14
Number of Flow Passes: 15
End time: 2021-03-27 14:21:20
Simulation Elapsed Time: 00:16:06
Simulation speed: 81.20 frames per second
Simulated time: 78462 timesteps
```

Figure 3 - Simple bounce-back walls transient simulation log

Figure 4 shows the transient part of the simulation and how it clearly achieved periodic flow that could be used to perform the measurements.

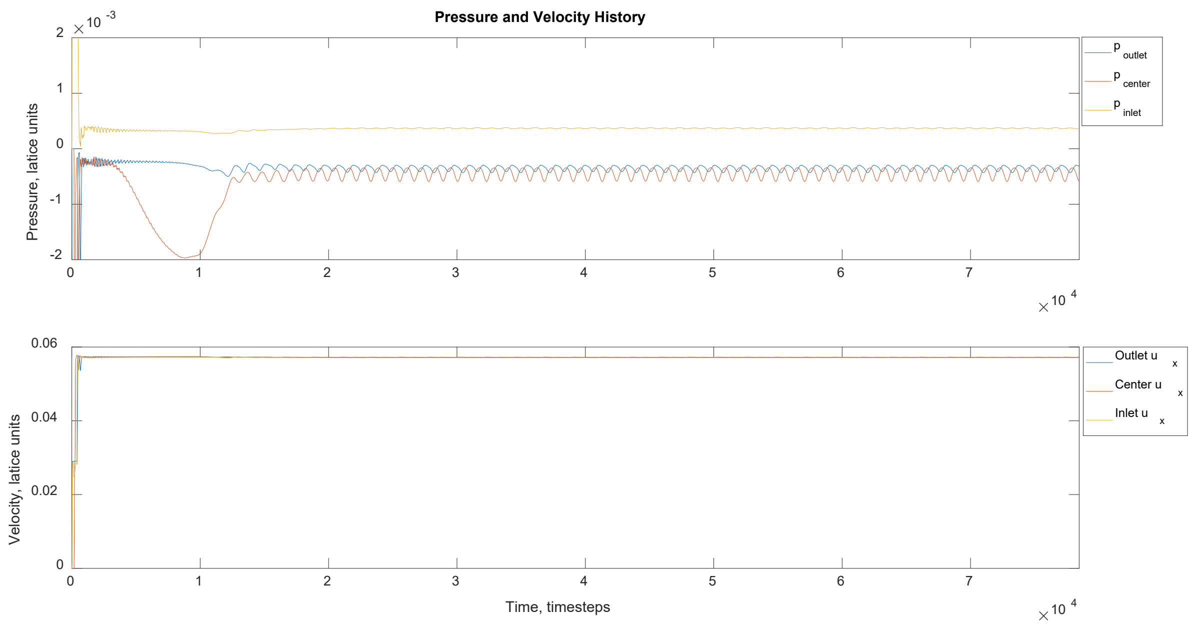


Figure 4 - Pressure and velocity history for 15 flow passes, until periodic flow was achieved

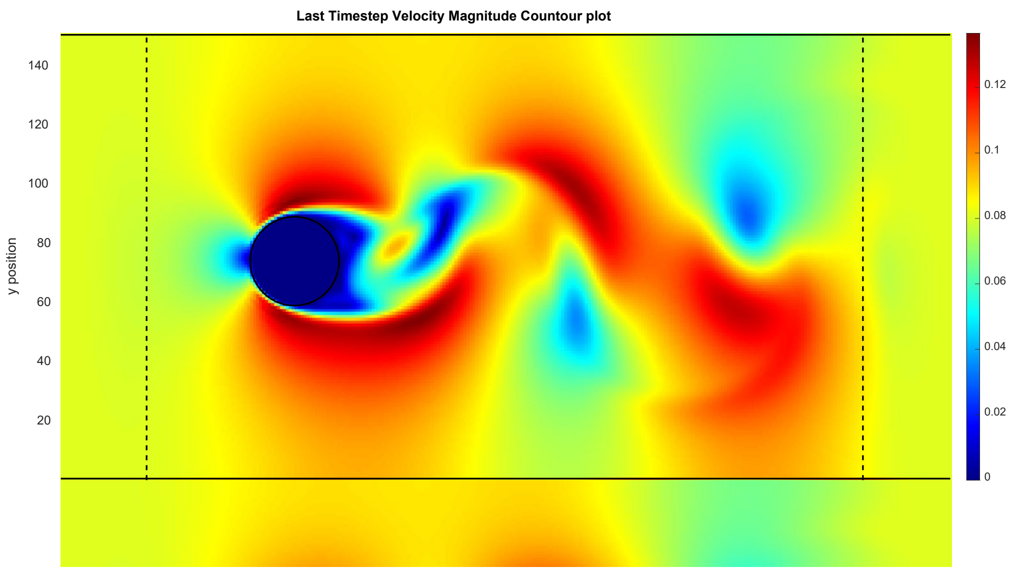


Figure 5 - Last timestep of the transient simulation, used as seed for the next step

Now that we know the von Kármán vortex street achieved steady state, we use the last saved distribution functions as seed and perform a quicker simulation with 4 flow passes.

```
Bounce-bakc walls
Start Time: 2021-03-27 15:38:57
Number of Flow Passes: 4
Seeding from previous results (seed_bounce_back.mat)
End time: 2021-03-27 15:43:12
Simulation Elapsed Time: 00:04:14
Simulation speed: 82.10 frames per second
Simulated time: 20924 timesteps
```

Figure 6 - Simple bounce-back walls periodic simulation log

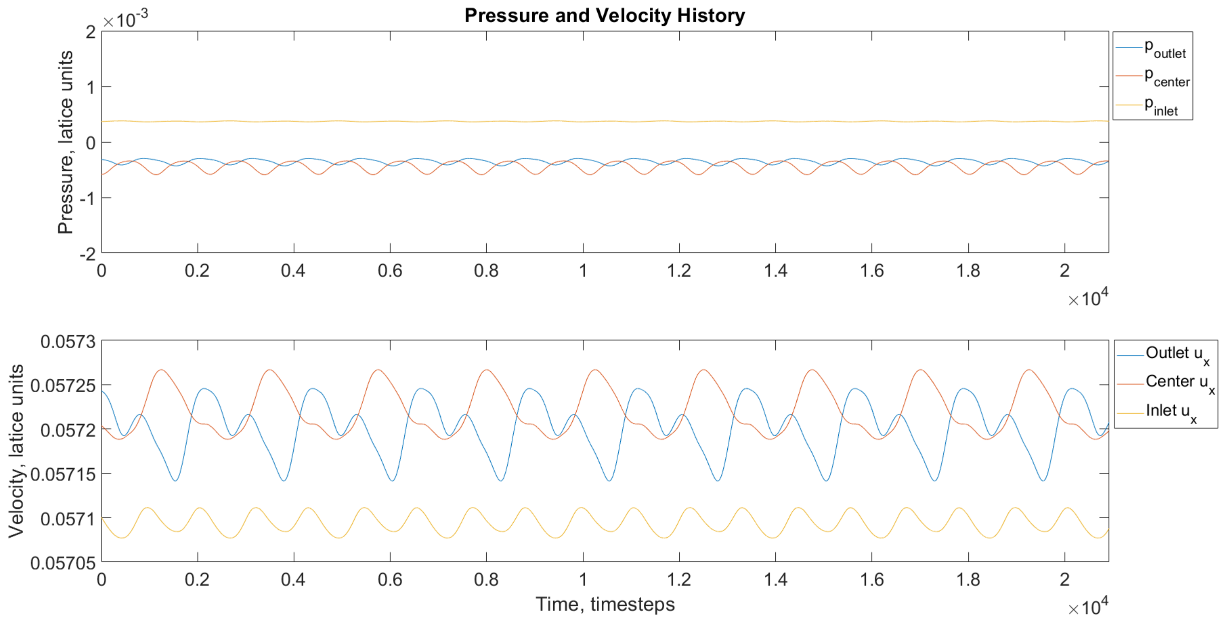


Figure 7 - Pressure and velocity history for 4 flow passes, periodic flow

The FFT of the density fluctuation showed that the vortex shedding frequency is $f_{char} = 9.0809 \cdot 10^{-4}\ \ 1/timestep$, resulting in a Strouhal number of $St = 0.4772$.

1.2 Bouzidi et al.

Simple bounce-back BC for rig wall has a big limitation to represent smooth real-world walls. Bouzidi et al. introduced in 2001 take into account the momentum from neighboring cells to estimate the effects of the real distance between a wall and its closest cells centers. First, they defined the ratio between the distances of the wall to the closest cells as $q = |AB|/|AC|$, as per Figure 8.

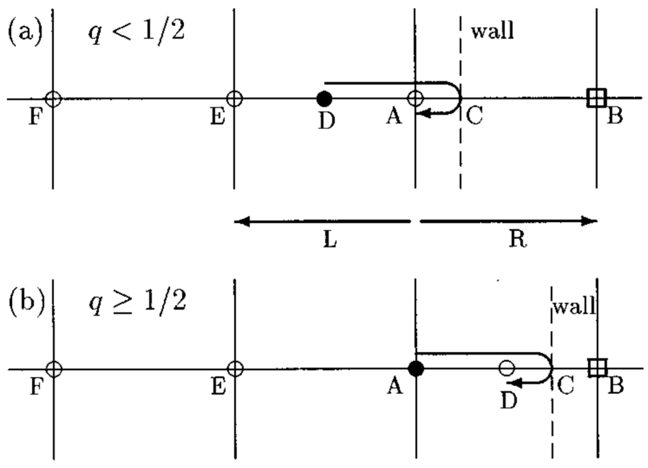


Figure 8 - Collision process with not-aligned wall. Source: Bouzidi et al. (2001)

Then, we must calculate the distribution function at the cells in the direction that crossed the walls. We look at two different cases:

a) $q < 1/2$

In this case, the populations that crossed the wall are reflected and end up at position D (Figure 8a). Using a quadratic interpolation for the moments of adjacent cells, it can be show that the distribution function in the cells closest to the walls are given by:

$$f_{i'}(\mathbf{r}_l, t+1) = q(2q+1) f_i^c(\mathbf{r}_l, t) + (1+2q) \times (1-2q) f_i^c(\mathbf{r}_l - \mathbf{c}_i, t) - q(1-2q) \times f_i^c(\mathbf{r}_l - 2\mathbf{c}_i, t), \quad q < \tfrac{1}{2},$$

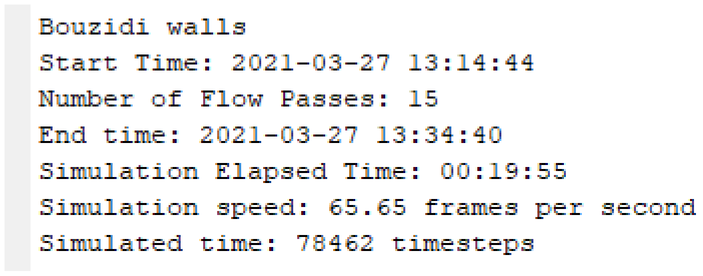
where f are the populations that crossed the wall in the direction i, and i' is the opposite direction of i.

b) $q \geq 1/2$

In this case, the populations that crossed the wall are reflected and end up at position D (Figure 8b). Here, the same procedure yields

$$f_{i'}(\mathbf{r}_l, t+1) = \frac{1}{q(2q+1)} f_i^c(\mathbf{r}_l, t) + \frac{(2q-1)}{q} f_{i'}^c(\mathbf{r}_l, t) + \frac{(1-2q)}{(1+2q)} f_{i'}^c(\mathbf{r}_l - \mathbf{c}_i, t), \quad q \geqslant \frac{1}{2}.$$

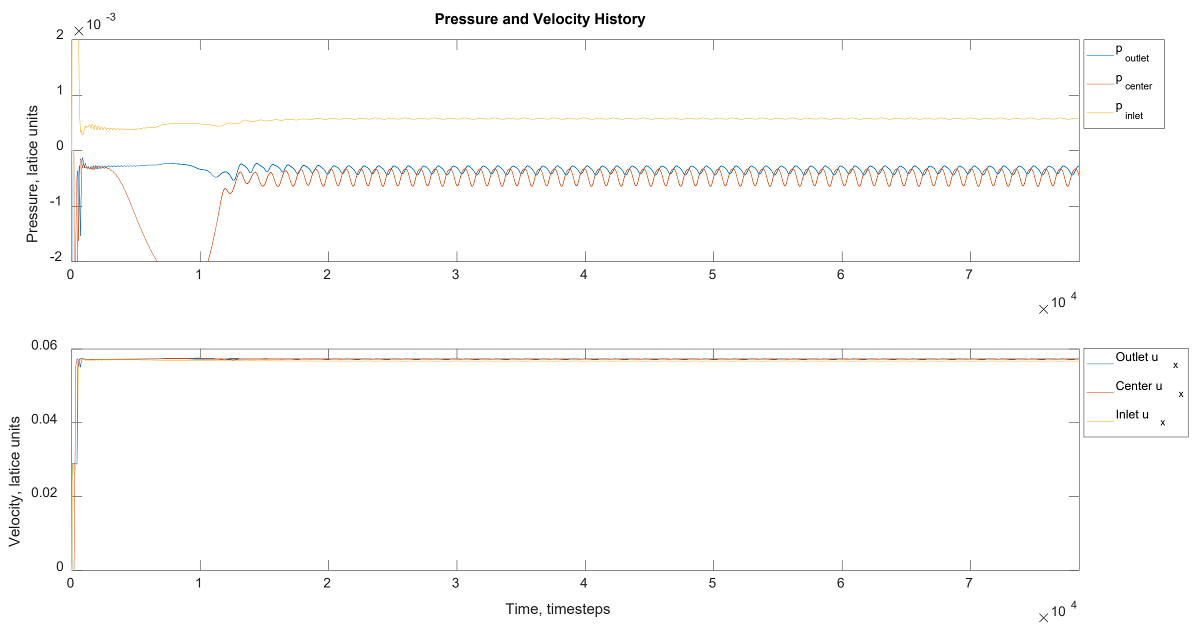
We now use the same procedure for transient time as we did in Section 1.1 for the transient time, until we reach periodic flow. Comparing the log for the simple bounce-back BC with the data from Figure 9, we can see that the code is significantly less efficient (-19% in frames per second).

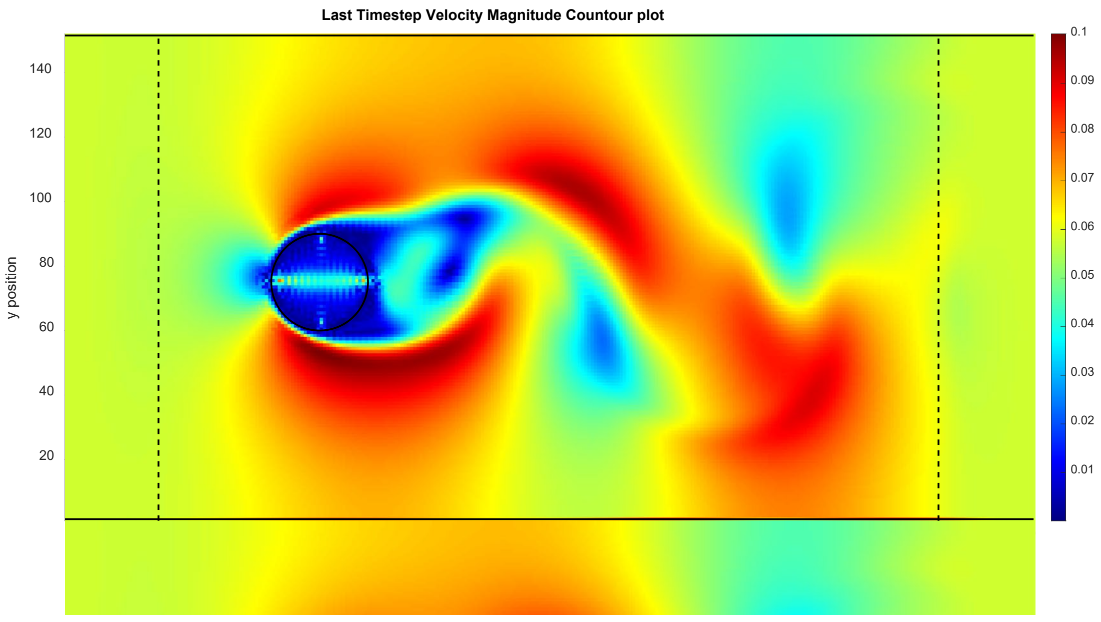


```
Bouzidi walls
Start Time: 2021-03-27 13:14:44
Number of Flow Passes: 15
End time: 2021-03-27 13:34:40
Simulation Elapsed Time: 00:19:55
Simulation speed: 65.65 frames per second
Simulated time: 78462 timesteps
```

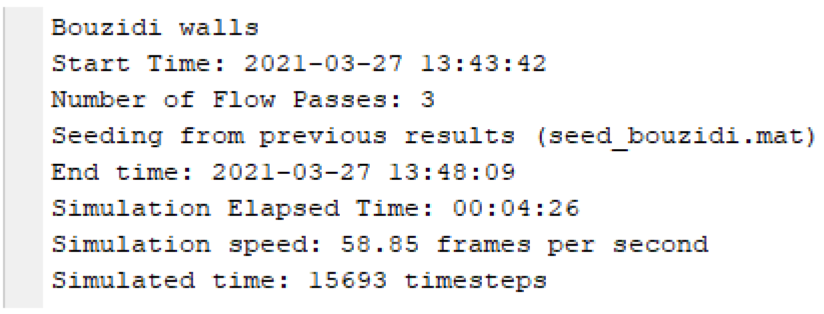
Figure 9 - Bouzidi walls transient simulation log

The following figures shows us that the periodic flow have actually been achieved.



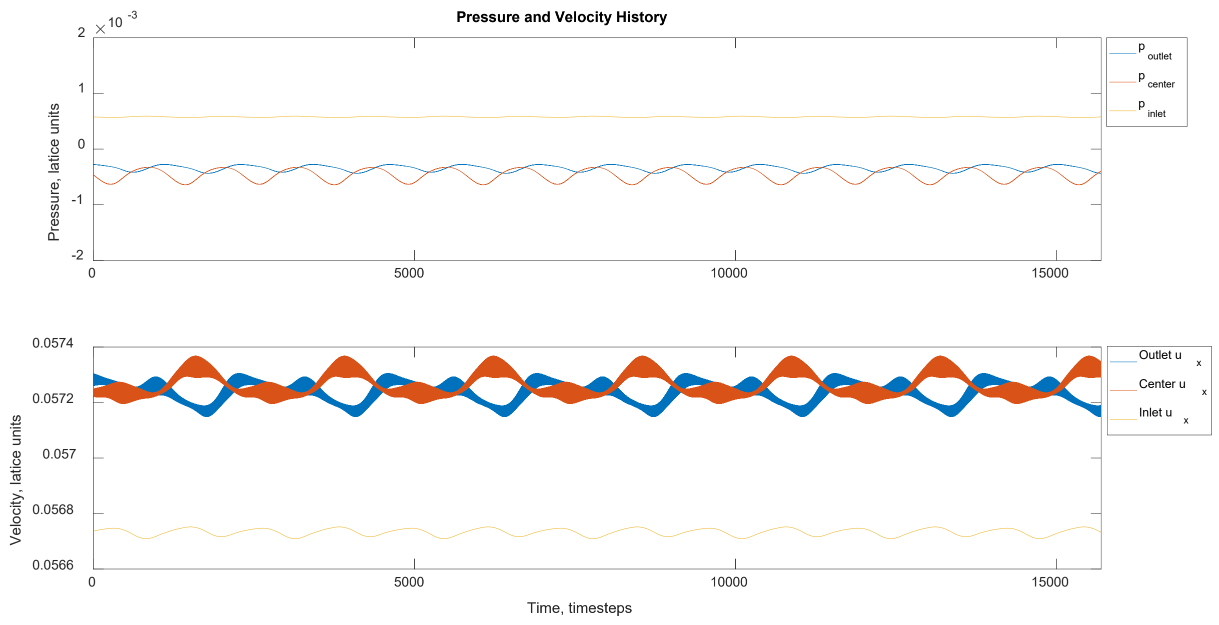


Now that we know the von Kármán vortex street achieved periodic flow, we can run a simulation using the last frame as seed, with much less flow pass (only 3).



```
Bouzidi walls
Start Time: 2021-03-27 13:43:42
Number of Flow Passes: 3
Seeding from previous results (seed_bouzidi.mat)
End time: 2021-03-27 13:48:09
Simulation Elapsed Time: 00:04:26
Simulation speed: 58.85 frames per second
Simulated time: 15693 timesteps
```

Figure 10 - Bouzidi walls periodic simulation log



The FFT of the density fluctuation showed that the vortex shedding frequency is $f_{char} = 8.9217 \cdot 10^{-4}\ \ 1/timestep$, resulting in a Strouhal number of $St = 0.4720$.

Comparing with results from simple bounce-back walls, we can see a small difference in the Strouhal number, only 1.1%, with a code 19% slower.

Part 2 – Moving String – Problem Description

In the second part of this work, a 2DQ9 lattice-Boltzmann scheme will be used to simulate a 2D fluid dynamic field around a bouncing string. The scheme adopts the BGK collision operator with an arbitrary relaxation frequency $\omega = 1.88$ and it runs in a 2D lattice space of 202 by 202 cells as presented below. The string is held static by two points $(x, y) = (50, 100)$ and $(x, y) = (150, 100)$, and has an initial tension, and vertical displacement $w(x, t)$. ABS's are used around the domain with thickness $D = 30\ latticeLengths$ (Figure 11).

Lallemand and Luo (2001) proposed a formulation for the interaction of moving walls and fluid by taking into account the momentum transfer into the fluid dynamic field. This is done by adding a term to the equations proposed by Bouzidi et al., which becomes

$$f_{\bar{\alpha}}(\mathbf{r}_j, t) = q(1+2q)f_\alpha(\mathbf{r}_j + \boldsymbol{e}_\alpha\delta_t, t) + (1-4q^2)f_\alpha(\mathbf{r}_j, t) - q(1-2q)f_\alpha(\mathbf{r}_j - \boldsymbol{e}_\alpha\delta_t, t) + 3w_\alpha(\boldsymbol{e}_\alpha \cdot \boldsymbol{u}_\mathrm{w}), \quad q < \frac{1}{2},$$

$$f_{\bar{\alpha}}(\mathbf{r}_j, t) = \frac{1}{q(2q+1)}f_\alpha(\mathbf{r}_j + \boldsymbol{e}_\alpha\delta_t, t) + \frac{(2q-1)}{q}f_{\bar{\alpha}}(\mathbf{r}_j - \boldsymbol{e}_\alpha\delta_t, t) - \frac{(2q-1)}{(2q+1)}f_{\bar{\alpha}}(\mathbf{r}_j - 2\boldsymbol{e}_\alpha\delta_t, t) + \frac{3w_\alpha}{q(2q+1)}(\boldsymbol{e}_\alpha \cdot \boldsymbol{u}_\mathrm{w}), \quad q \geqslant \frac{1}{2}.$$

where f are the populations that crossed the wall in the direction α, $\bar{\alpha}$ is the opposite direction of α, and $\boldsymbol{e}_\alpha \cdot \boldsymbol{u}_w$ is the velocity of the wall parallel to the α direction.

This scheme will be used to in two different ways. First, in a weakly coupled formulation, the displacement of the string due to initial conditions will affect the fluid dynamic field, but not the other way around. Then, in the strongly coupled formulation, the fluid dynamic field will produce forces that will affect the way the string moves. This will be calculated using the difference in pressure between both sides of the string with

$$F(x,y,t) = p(x,y-1,t)\Delta x^2 - p(x,y+1,t)\Delta x^2 = \big(\rho(x,y-1,t) - \rho(x,y-1,t)\big)c_s^2\Delta x^2 \,.$$

The string movement was modeled using

$$T\frac{\partial^2 w(x,t)}{\partial x^2} + F(x,t) = \rho\frac{\partial^2 w(x,t)}{\partial t^2}$$

where T is the tension in the string, F is the vertical force action upon the string, ρ is the string linear density and $w(x,t)$ is the vertical displacement. The simulation was made with $T = 0.333$ and $\rho = 160$, both in lattice units. The initial position is given by

$$w(x,t=0) = \frac{2}{25}x \quad \text{para} \quad 1 \le x \le 25$$

$$w(x,t=0) = -\frac{2}{75}x + \frac{8}{3} \quad \text{para} \quad 25 \le x \le 100$$

Figure 11 shows the initial position of the string, and the ABC, colored according to the local σ value.

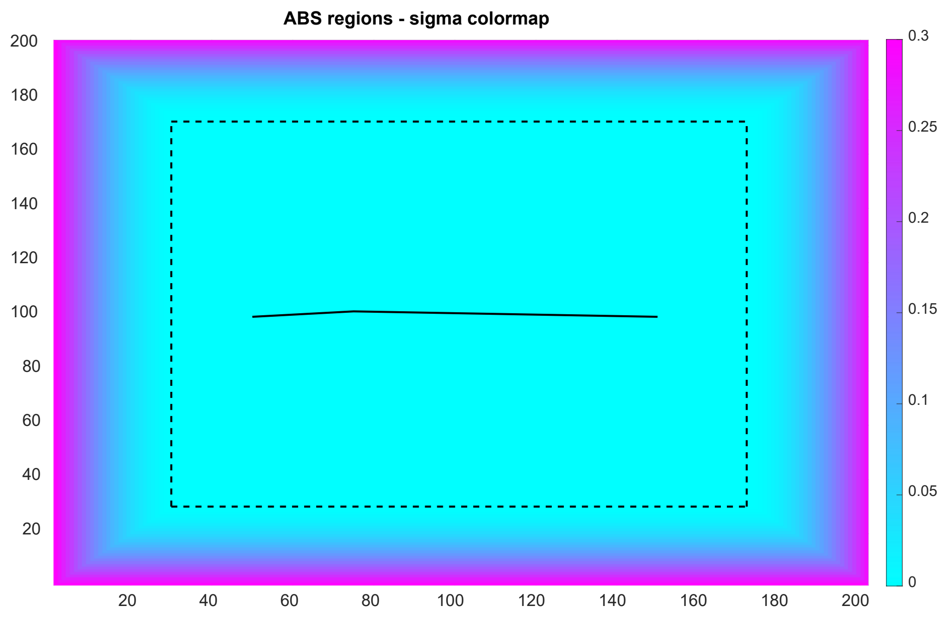
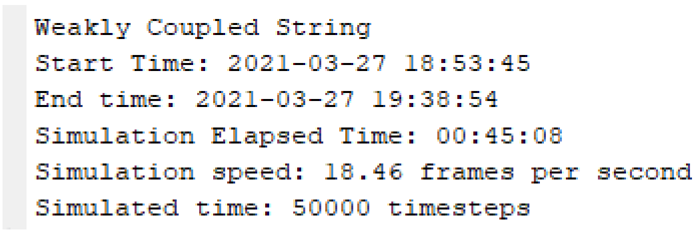


Figure 11 - Computational domain boundary conditions

2.1 Weakly Coupled

Since the fluid dynamic field do not affect the string movement, we can compute it separately and use it as input for the wall position at every timestep, setting the force upon string $F = 0$ during the whole simulation. For comparison purposes, we ran the simulation with this set up to access the simulation speed, shown in Figure 12. Since we need to compute what cells to apply the Lallemand and Luo equations at every timestep, the efficiency of the code drops drastically. The results are plotted in Figure 14.

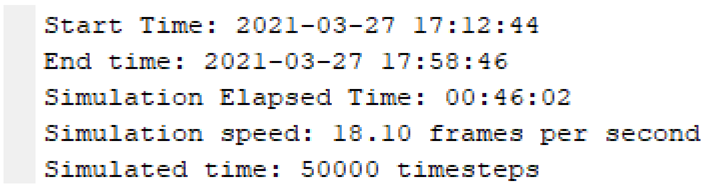


```
Weakly Coupled String
Start Time: 2021-03-27 18:53:45
End time: 2021-03-27 19:38:54
Simulation Elapsed Time: 00:45:08
Simulation speed: 18.46 frames per second
Simulated time: 50000 timesteps
```

Figure 12 - Weakly coupled string log

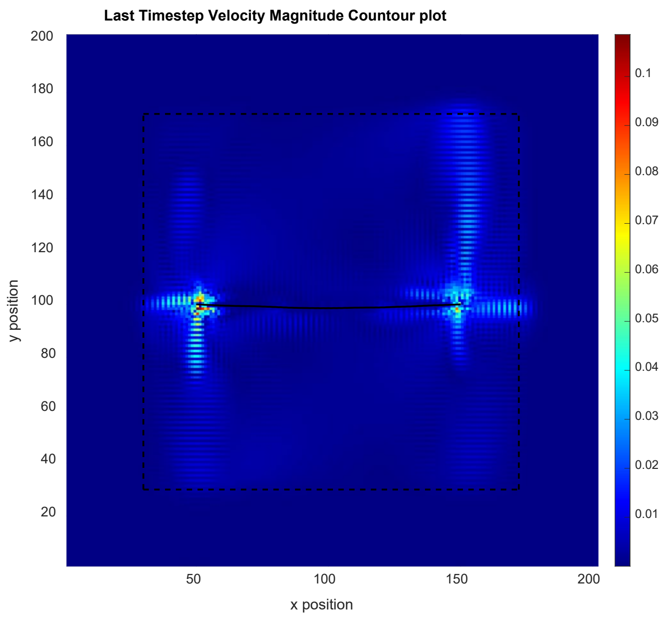
2.2 Strongly Coupled

Now, we take into account the influence of the fluid dynamic field upon the strig by calculating $F(x,t)$ at every timestep and solving the string equation at every timestep as well. This changes the way the string moves, as a consequence, it also changes how the fluid dynamic field evolves. Figure 13 show the log of the simulation, from which it can be seen that adding the computation of the forces and solving the string equation simultaneously does not change the simulation speed significantly, since the slower step is finding which cells are crossed by the string.



```
Start Time: 2021-03-27 17:12:44
End time: 2021-03-27 17:58:46
Simulation Elapsed Time: 00:46:02
Simulation speed: 18.10 frames per second
Simulated time: 50000 timesteps
```

Figure 13 - Strongly coupled string log.



The results are shown in Figure 14, where it can be seen that the string behavior diverges when the forces are taken into account. It is small at first, but this is most likely due to the high density of the string, which makes it harder to change its path, due to inertia.

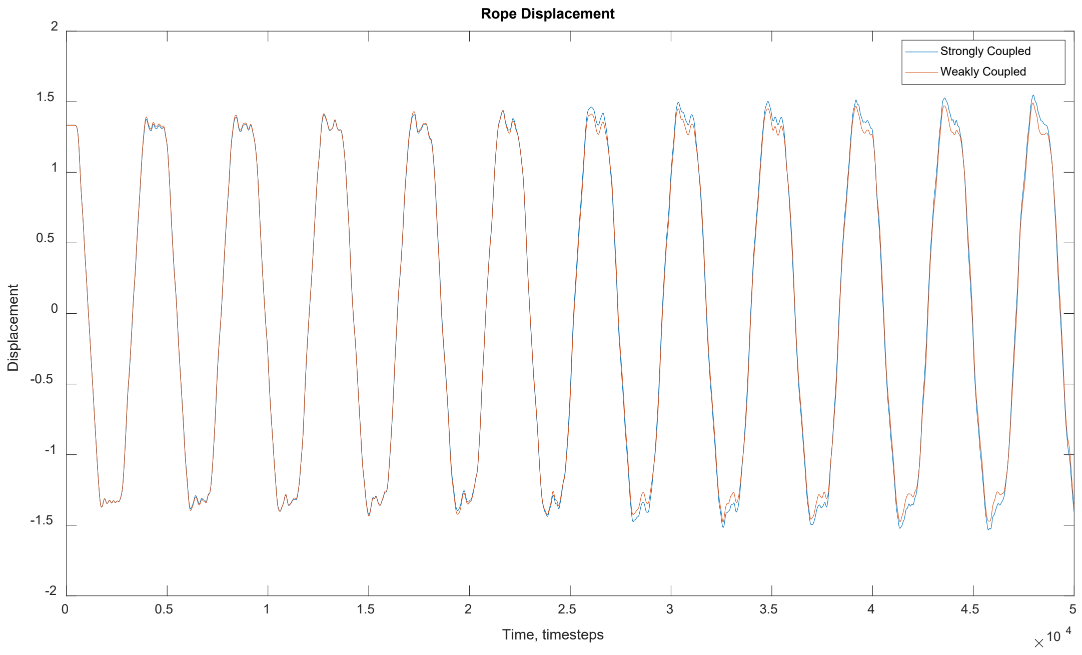


Figure 14 - History of the displacement of the central point of the string.