Computational Fluid Dynamics Report

Stefano Lusardi

June 2025

Contents

1 Lid Driven Cavity

The first simulation I will illustrate is the one concerning the lid driven cavity, so a 2D cavity filled with a fluid that has a free surface on which it is blowing a constant wind. The cavity has dimensions $1 \cdot 1m^2$ and the wind speed is $1m/s$, different Reynolds number are obtained acting on the viscosity ν.
The equations we should focus on to model the phenomenon are the Navier-Stokes equations in 2D for incompressible flow:

$$\partial u/\partial x + \partial v/\partial y = 0$$

$$\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} = -\frac{\partial p}{\partial y} + \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2})$$

NOTE: in this case I have already assumed $\rho = 1$ for simplicity.

1.1 $Re = 100$ on Matlab

The first implementation concerns a Matlab code using $dx = dy = 0.01$ and $dt = 0.001$. Since I actually run the openfoam code before, I already know that the steady state is reached at $T = 6s$, so I run the solver for 6000 iterations. To have $Re = 100 = UL/\nu$, $\nu = 0.01$.
A necessary check, in all the simulations, is the one regarding the stability and the Courant number.

$$\frac{dt}{dx^2 Re} \leq 0.25 \qquad \frac{dt}{dy^2 Re} \leq 0.25 \qquad dt(\frac{|U|}{dx} + \frac{|V|}{dy}) \leq 1$$

After having initialized all the needed variables and matrices, at each iteration the following steps are performed (note that each step is focusing on a time t):

1. **Predictor phase** that solves momentum equation without pressure through centered differences in space

```
ss_x=nu*dt/(dx^2);
ss_y=nu*dt/(dy^2);
s_x=dt/dx;
s_y=dt/dy;
u_star(i,j) = u(i,j,n-1) ...
    - s_x * ( (u(i+1,j,n-1)^2 - u(i-1,j,n-1)^2) / 2 ) ...
    - s_y * ( (u(i,j+1,n-1)*v(i,j+1,n-1) - u(i,j-1,n-1)*v(i,j-1,n-1)) / 2 )
         ...
    + ss_x * (u(i+1,j,n-1) - 2*u(i,j,n-1) + u(i-1,j,n-1)) ...
    + ss_y * (u(i,j+1,n-1) - 2*u(i,j,n-1) + u(i,j-1,n-1));

v_star(i,j) = v(i,j,n-1) ...
    - s_y * ( (v(i,j+1,n-1)^2 - v(i,j-1,n-1)^2) / 2 ) ...
    - s_x * ( (v(i+1,j,n-1)*u(i+1,j,n-1) - v(i-1,j,n-1)*u(i-1,j,n-1)) / 2 )
         ...
    + ss_x * (v(i+1,j,n-1) - 2*v(i,j,n-1) + v(i-1,j,n-1)) ...
    + ss_y * (v(i,j+1,n-1) - 2*v(i,j,n-1) + v(i,j-1,n-1));
```

At this point I impose on u^* and v^* the boundary conditions that are of no-slip type, excluding the top wall that of course has $u = 1$.

2. **Source computation**, based on the fact that $-\Delta\mathbf{P^n} = \frac{\nabla\vec{u^n} - \nabla\vec{u^*}}{\Delta t}$. Imposing mass conservation:

$$\mathbf{S} = \nabla\vec{u^*}/\Delta t = \Delta\mathbf{P^n}$$

```
S(i,j) = ((u_star(i+1,j) - u_star(i-1,j)) / (2*dx*dt) + (v_star(i,j+1) -
    v_star(i,j-1)) / (2*dy*dt)) * dx^2;
```

where I have already multiplied by $dx^2 = dy^2$ to simplify the successive steps.

3. **Poisson equation solution** through an implicit scheme and the solution of a linear system.

$$\frac{P^n_{i+1,j} - 2P^n_{i,j} + P^n_{i-1,j}}{\Delta x^2} + \frac{P^n_{i,j+1} - 2P^n_{i,j} + P^n_{i,j-1}}{\Delta y^2} = S_{i,j}$$

The first thing to do here is to change the format of $\mathbf{P}$ and $\mathbf{S}$ to become a vector instead of a matrix

$$p = (P_{11}, P_{12}, ..., P_{21}, P_{22}, ...)$$

and similarly for s. In this way I can resolve numerically the system

$$\mathbf{A}p = s$$

$$A = \begin{bmatrix} B & I & 0 & \cdots & 0 \\ I & B & I & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I & B \end{bmatrix}$$

$$B = \begin{bmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -4 \end{bmatrix}$$

and get back to $\mathbf{P}$. $\mathbf{I}$ is the identity matrix.
Here I also impose the homogeneous Neumann boundary conditions for pressure.

4. **Corrector phase** At this point I just need to put everything together considering

$$\frac{u^n - u^{n-1}}{\Delta t} = -\nabla P^n + adv^{n-1} + diff^{n-1}$$

that practically transaltes into

```
u(i,j,n) = u_star(i,j) - dt * (p(i+1,j,n) - p(i-1,j,n)) / (2*dx);
v(i,j,n) = v_star(i,j) - dt * (p(i,j+1,n) - p(i,j-1,n)) / (2*dy);
```

Let us have a look at some results.

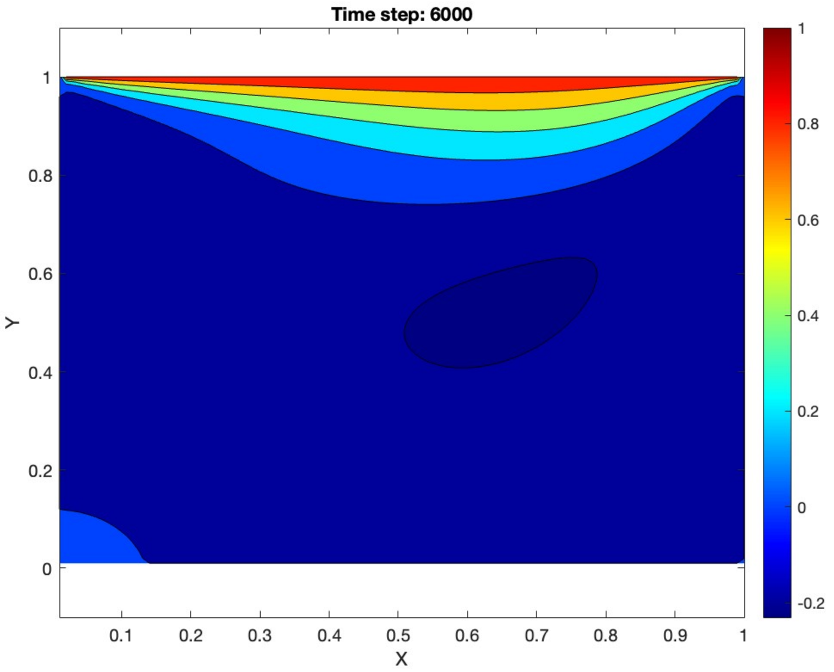
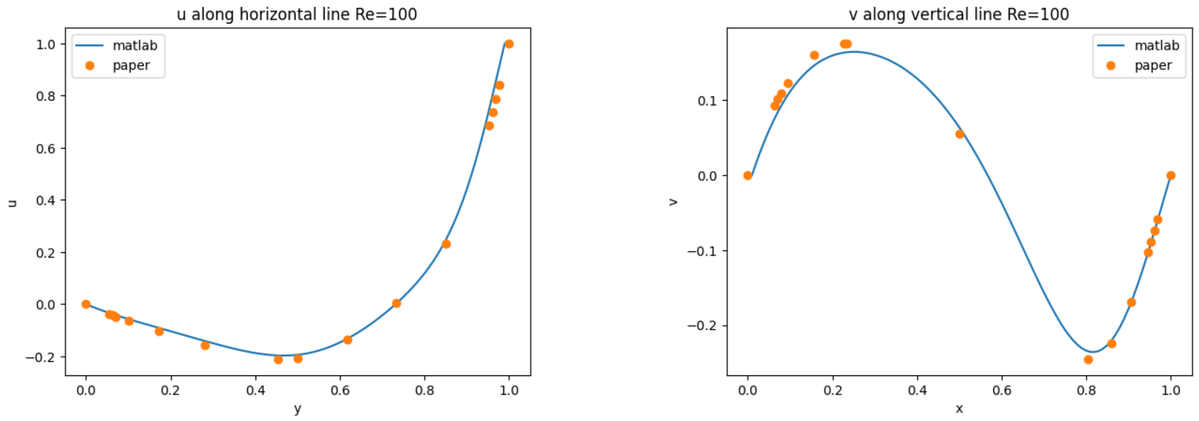


Figure 1: u field in contour plot at $T = 6s$ (steady state).

Here, we observe the formation of a vortex caused by the shearing motion of the top lid and the no-slip boundary conditions along the cavity walls. Also the shearing motion is well visible, as well as a small secondary vortex in the bottom left corner.
It is also interesting to look at the vertical velocity extracted in the middle of the cavity on an horizontal line and at the horizontal velocity extracted in the middle of the cavity on a vertical line. These results can be compared with the results in a paper by Ghia et al. [1]



(a) Comparison for u along horizontal line

(b) Comparison for v along vertical line

Figure 2: Results for Matlab, lines extracted respectively at $y = 0.5$, $x = 0.5$.

If you want to have a look at the complete code, as well as the way graphs are generated, and the data from openfoam, you can look here `https://github.com/onafetsidrasul/CFD`.

1.2 $Re = 100$ on openfoam

The same study case has been simulated also on openfoam, a very useful framework for CFD in C++ that allows to automatize and make the simulations a lot faster. The solver used is $icoFoam$.
In this case the grid has been implemented to have $200 \cdot 200$ cells and to be finer near borders and corners but coarser in the central part of the cavity (1/5 the number of grid points in the finer parts). All of the other parameters are kept as in the Matlab scenario.

As already pointed out, the study with openfoam allows to have a look in a more automatic way to when the steady state is reached. Indeed choosing a point and looking at pressure or velocity one can look at when it stabilizes.

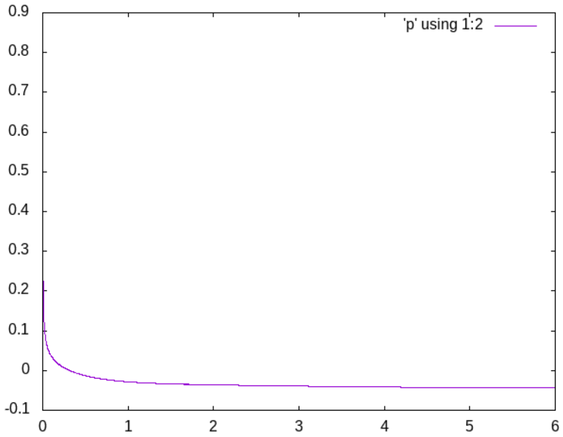


Figure 3: In abscissa is depicted the time, in ordinate p.

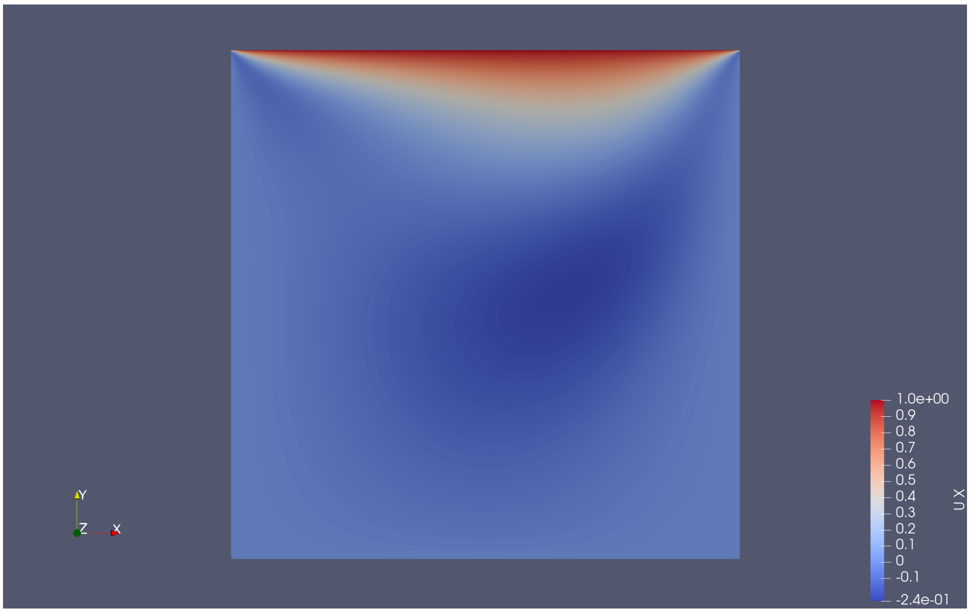


Figure 4: u field in contour plot at $T = 6s$ (steady state).

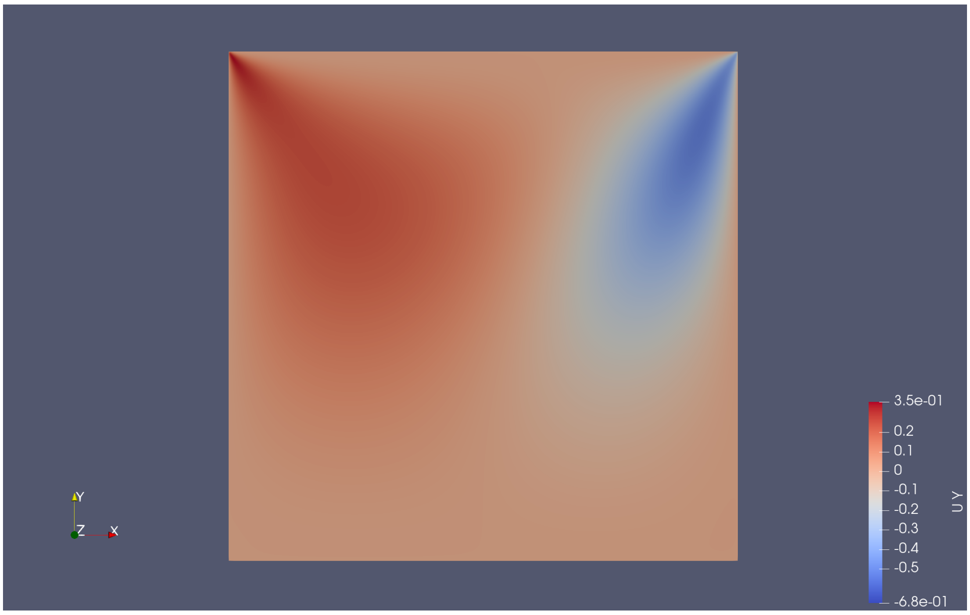
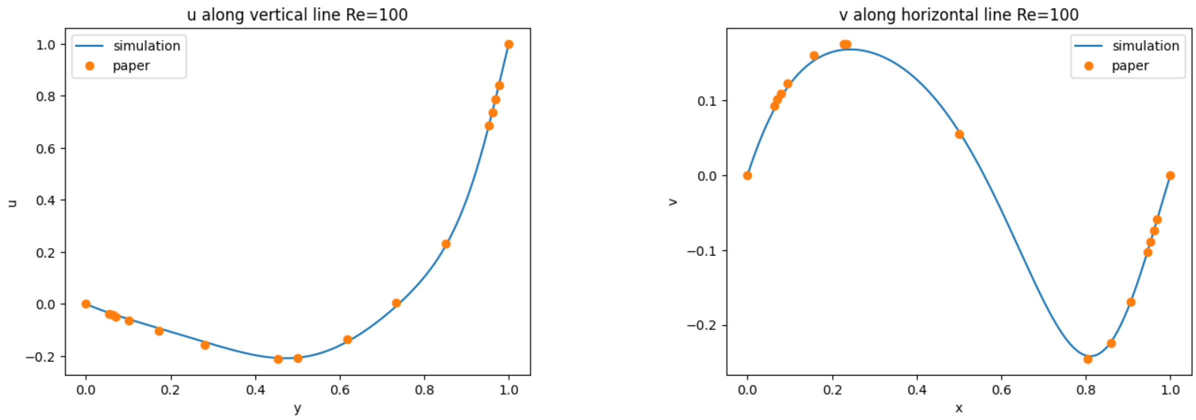


Figure 5: v field in contour plot at $T = 6s$ (steady state).



(a) Comparison for u along horizontal line

(b) Comparison for v along vertical line

Figure 6: Results for openfoam, lines extracted respectively at $y = 0.5$, $x = 0.5$.

The better results I am obtaining in this case compared to the Matlab one can be explained by the finer and more cleaver way in which the grid has been implemented. Despite this fact the computational time of the two differs of an order of magnitude (in hours) with the openfoam case that overperforms the Matlab one.

1.3 $Re = 1000$

The following case has been implemented changing $\nu = 0.001$, everything else is still the same.

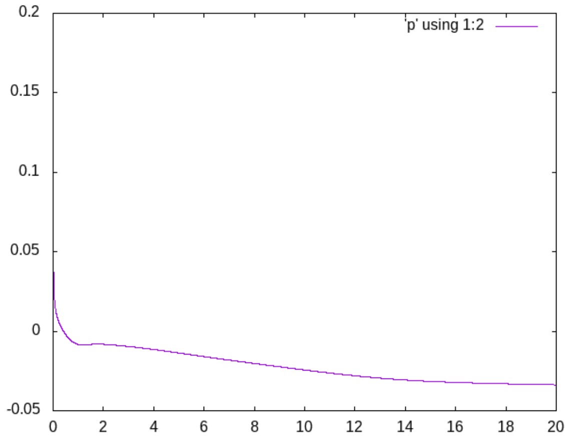
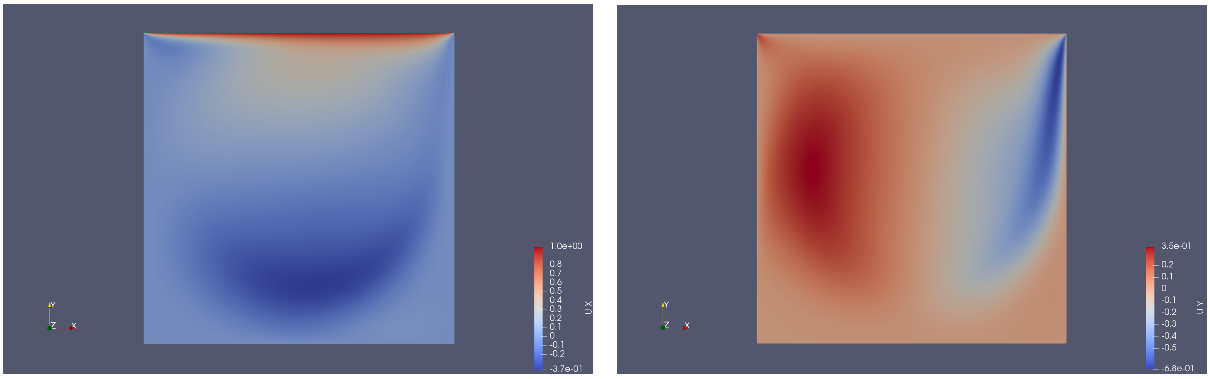


Figure 7: Check of steady state. In abscissa is depicted the time, in ordinate p.

As we can notice, the steady state is reached around $T = 20s$.

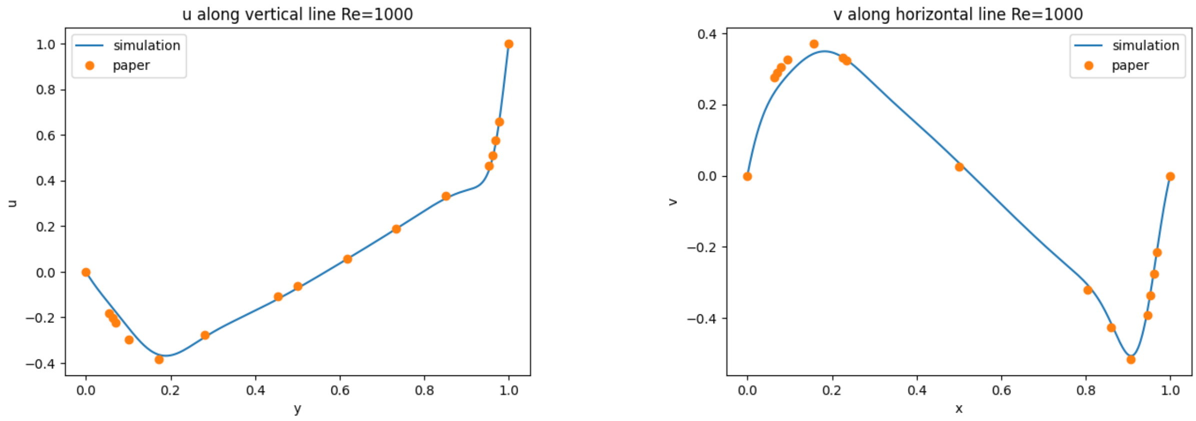


(a) u field in contour plot.

(b) v field in contour plot.

Figure 8: Results at steady state, $T = 20s$.

Comparing these results with the ones with lower Re, we notice a longer computational time needed, as the steady state requires more time to emerge, and a bigger vortex showing up.



(a) Comparison for u along horizontal line

(b) Comparison for v along vertical line

Figure 9: Results for openfoam, lines extracted respectively at $y = 0.5$, $x = 0.5$.

1.4 $Re = 3200$

The following case has been obtained changing $\nu = 0.0003125$.

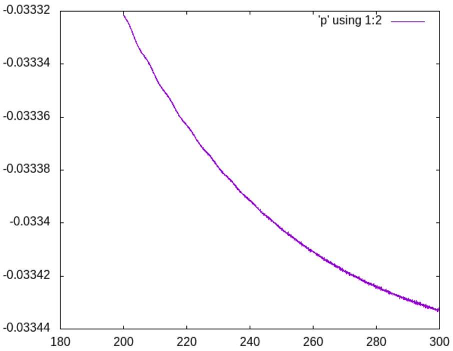
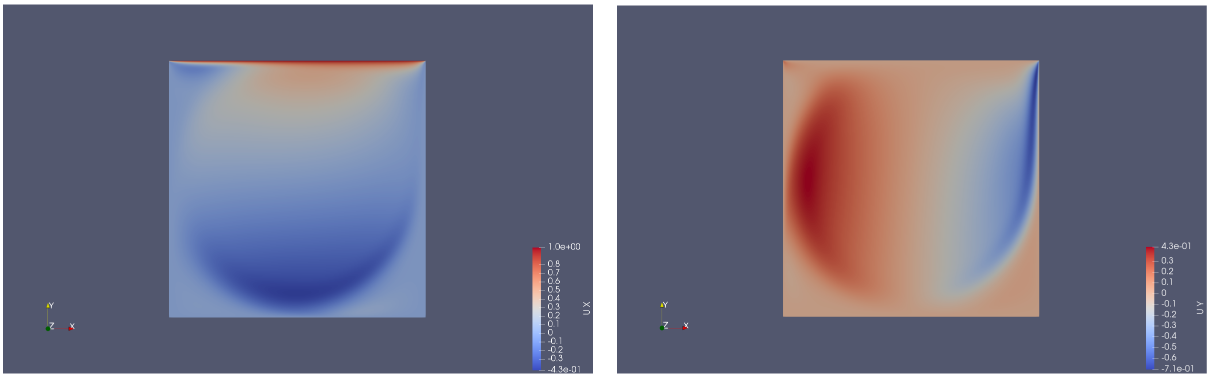


Figure 10: Check of steady state. In abscissa is depicted the time, in ordinate p.

Here I considered the steady state to be practically reached around $300s$ since we notice that between $200s$ and $300s$ p changes only in the 4^{th} decimal.

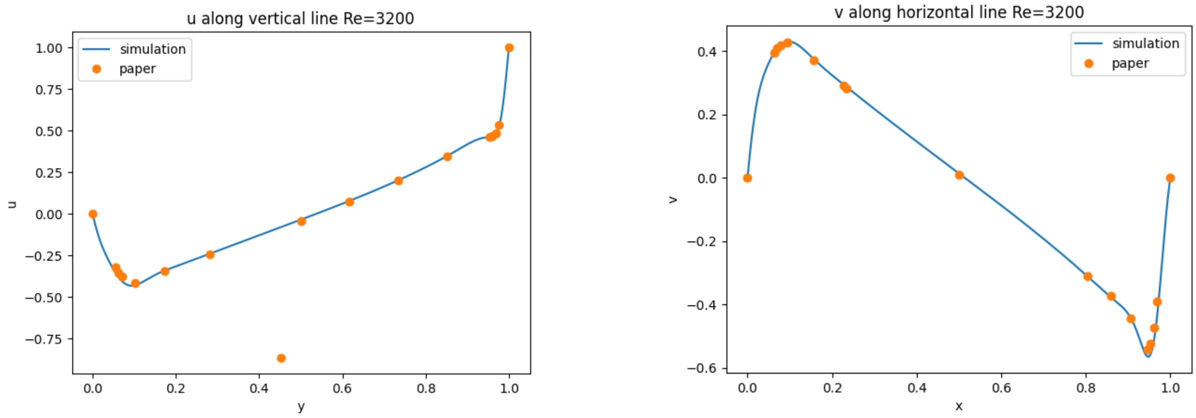


(a) u field in contour plot.

(b) v field in contour plot.

Figure 11: Results at steady state, $T = 300s$.

The considerations done above for the higher Re are still valid. Since ν is smaller we should not be surprised to see that the region directly affected by the shearing motion is less big.



(a) Comparison for u along horizontal line

(b) Comparison for v along vertical line

Figure 12: Results for openfoam, lines extracted respectively at $y = 0.5$, $x = 0.5$.

The point that the simulation is missing in (12a) is probably a typo of the paper.

2 Poiseuille flow

The second type of simulation I run is the one regarding Poiseuille flow, so the flow that happens in between 2 parallel plates distant $2b$ one from each other. Considering a 2D case, we know analitycally that extracting the velocity field u on a vertical line we will get a parabolic shape.
Indeed looking again at the Navier-Stokes equations

$$\partial u/\partial x + \partial v/\partial y = 0$$

$$\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} = -\frac{\partial p}{\partial y} + \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2})$$

and noticing that the 2 plates stay still with v that doesn't change, the last equation becomes

$$\frac{\partial p}{\partial y} = 0$$

And the first equation becomes

$$\frac{\partial u}{\partial x} = 0$$

Considering also the flow to be fully developed, and so, steady

$$\frac{\partial u}{\partial t} = 0$$

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} = 0$$

So, the second equation becomes

$$-\frac{dp}{dx} + \nu\frac{d^2 u}{dy^2} = 0$$

and integrating

$$B + Ay = -y^2/2\frac{dp}{dx} + \nu u$$

Applying no-slip bc $u(0) = u(2b) = 0$

$$u(y) = -\frac{y}{\nu}\frac{dp}{dx}(b - y/2)$$

The implementation on openfoam consider a pipe with $L = 3m$, $2b = 1m$. The grid has $300 \cdot 100$ cells with simple grading and $dt = 0.00025$. The solver in this case is set to be *pimpleFoam*, the *simulationType* is laminar, the *transportModel* is Newtonian with $\nu = 0.05$ (again $\rho = 1$).
The boundary conditions are of type no-slip for the top and bottom wall while for the inlet and outlet I have cyclic boundary conditions.

Even though I have set $u = 5m/s$, it is probable that this is not the u_{bulk} we want to insert in the parabola equation to get the unknown $\frac{dp}{dx}$, indeed this was leading to an out of scale result.
To overcome this issue I used the fact that the forces at the wall should be equal to 0

$$\frac{dp}{dx}dxdy = \frac{d\tau}{dy}dxdy$$

Simplifying and integrating I get

$$\tau(y) = \frac{dp}{dx}y + C$$

that imposing $\tau(0) = 0$ (at the center of the tube) implies

$$\tau(y) = \frac{dp}{dx}y$$

and at the boundary, for Newtonian fluids

$$\tau(b) = \frac{dp}{dx}b = \nu\frac{du}{dy}|_b$$

Then, what I did in practice was to look at the value of u in the first cell against the wall and, knowing its dimensions and that I have no-slip conditions

$$\frac{u_{wall} - 0}{dy/2} \approx \frac{du}{dy} \approx 49.9$$

$$\frac{dp}{dx} = \frac{\nu}{b}\frac{du}{dy} \approx \frac{0.05}{1/2}49.9 \approx 4.99$$

We can now see the results.

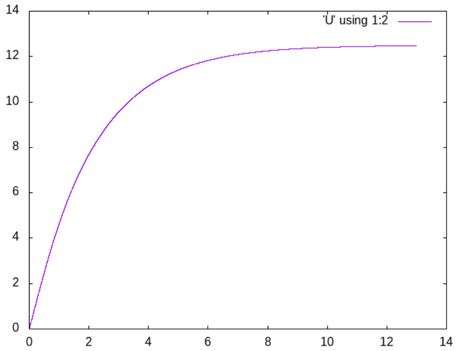


Figure 13: Check of steady state. In abscissa is depicted the time, in ordinate u.

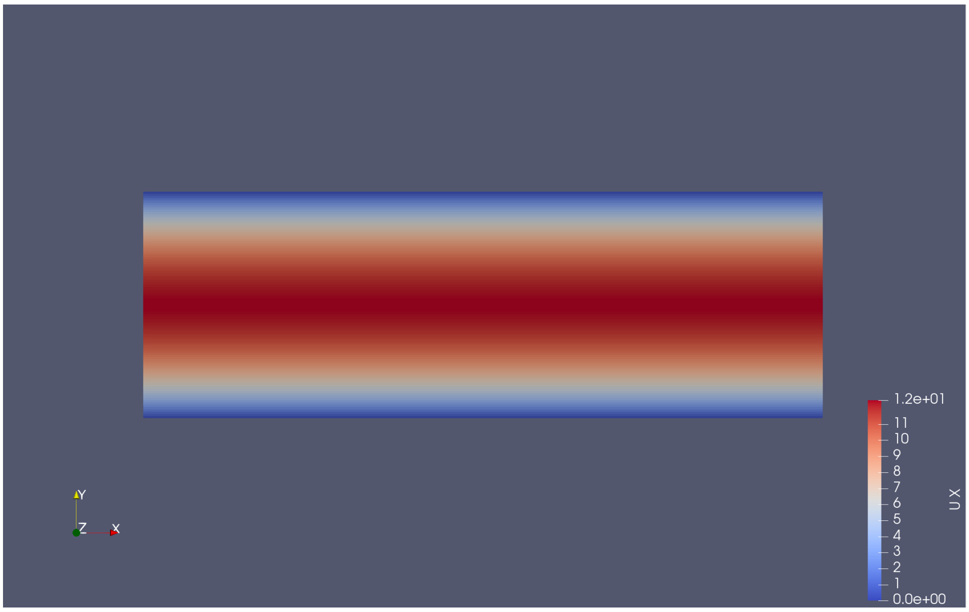


Figure 14: u field in contour plot at $T = 13s$ (steady state).

As we should expect the biggest velocity is reached in the center of the pipe while $u = 0$ at the wall.

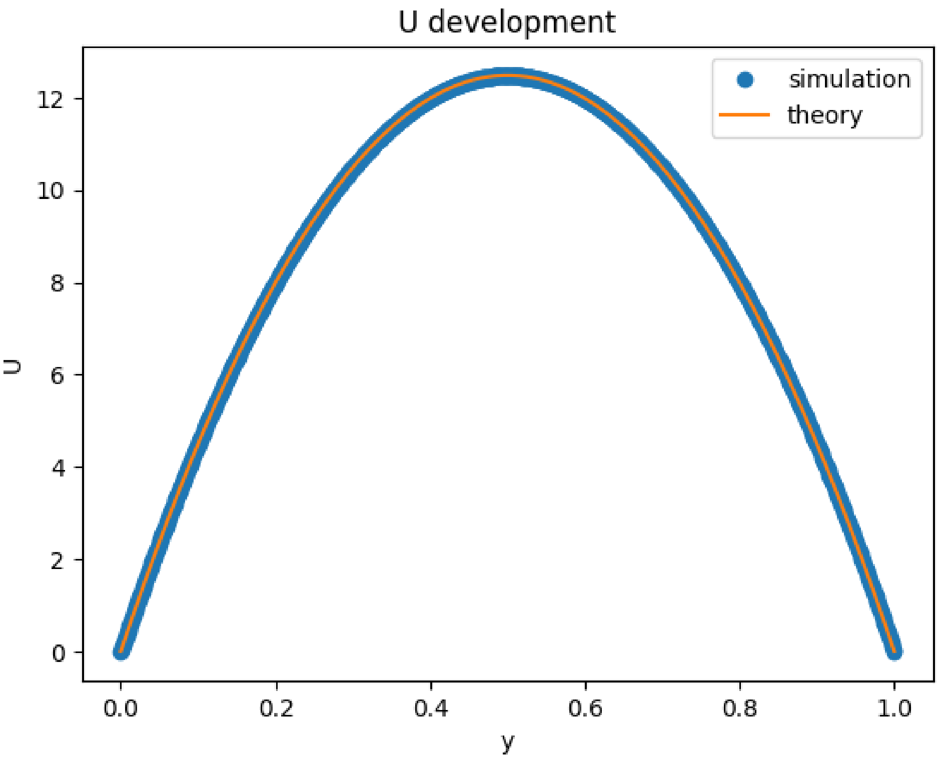


Figure 15: u extracted at $x = 1.5$ and $T = 13s$ compared with the analytical expression.

We see that there is practically a perfect superposition between the data extracted along a vertical line for u and the analytical expression I derived above.

3 Airfoil, boundary layer

A 2D airfoil is basically a section of a wing or a propeller.
When working with high Reynolds number the viscous effects generated on the foil are present only near the wall, in a region called boundary layer. The fact that there is the emergence of this region is related to the first Stokes problem, which considers a suddenly started plate that generates momentum diffusion with a diffusive layer thickness δ.
For this problem it is possible to find a non dimensional group

$$\eta = \frac{y}{2\sqrt{\nu t}}$$

and for $y = \delta$

$$\eta = \frac{\delta}{2\sqrt{\nu t}}$$

so that

$$\delta(t) \propto \sqrt{\nu t}$$

Trying to adapt this considerations to an airfoil I need to specify that what I am doing in this case is keeping the foil still and letting a wind with velocity U flow on it. That said I can consider the advection time

$$t = x/U$$

and

$$\delta \approx \sqrt{\frac{\nu x}{U}}$$

The most interesting part here is

$$\delta \propto \sqrt{x}$$

and fixing x

$$\delta \propto 1/\sqrt{Re}$$

Willing to find an expression for Navier-Stokes in this specific case we can rely on the work of Prandtl and Blasius.

- Prandtl was the first who studied it and from some simplifications coming from $\delta/L = \epsilon << 1$, with δ the thickness of the layer and L the length of the foil, he found

  $$\frac{\partial p}{\partial y} \approx O(\epsilon)$$

  and so

  $$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \nu\frac{\partial^2 u}{\partial y^2}$$

  Imposing the advection part to be *circa* the diffusion part and substituting the scale quantities

  $$U^2/L \approx \frac{\nu U}{\delta^2}$$

  we find again

  $$\delta \approx \sqrt{\frac{\nu L}{U}}$$

- Blasius carried on further simplifications considering a flat plate, a laminar flow, and reduced everything to an ODE thanks to self similar solutions. Thanks to him we can say that δ can be extracted looking at where

  $$u = 99\%U$$

Even though these results are related to a flat plate, they can be extended to the foil.
To model the foil I used NACA0012 profile with length 0.12, if you want to find more details on how also the mesh has been implemented look at the *meshgen.m* file in the repository and consider that *blockMeshDict* should be modified to have only integers in the number of cells and all the gradings set to one (for simplicity).
The boundary conditions are set to be of type fixedValue on the inlet with $u = 1$, no-slip on the foil, slip on the top and bottom of the domain, and zeroGradient at the outlet.

3.1 $Re = 1000$

The case of $Re = 1000$ is obtained with $\nu = 0.00012$. The solver is *icoFoam*.
As in the previous cases I can check that the steady state is reached by looking at the probe that in this case has been put above the foil.

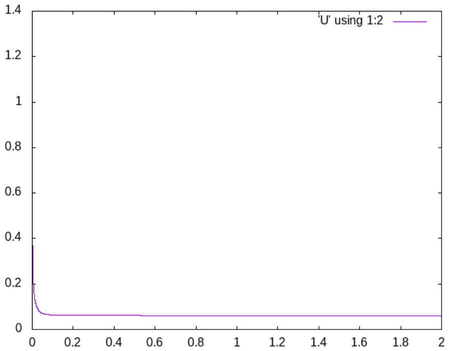


Figure 16: Check of steady state. In abscissa is depicted the time, in ordinate u.

What I can do now is to check the velocity profile and hence, the thickness of the boundary layer by looking at where $u = 99\%U$ where U has been extracted in the case with *slip* conditions on the foil.
The velocity profiles should be extracted at lines that are perpendicular to the foil but I did it just trying to look at the foil on the screen. Moreover, since the foil is curved, the distances that separate

the different lines are not very easy to calculate: here I report a scheme of how I computed them on Geogebra.

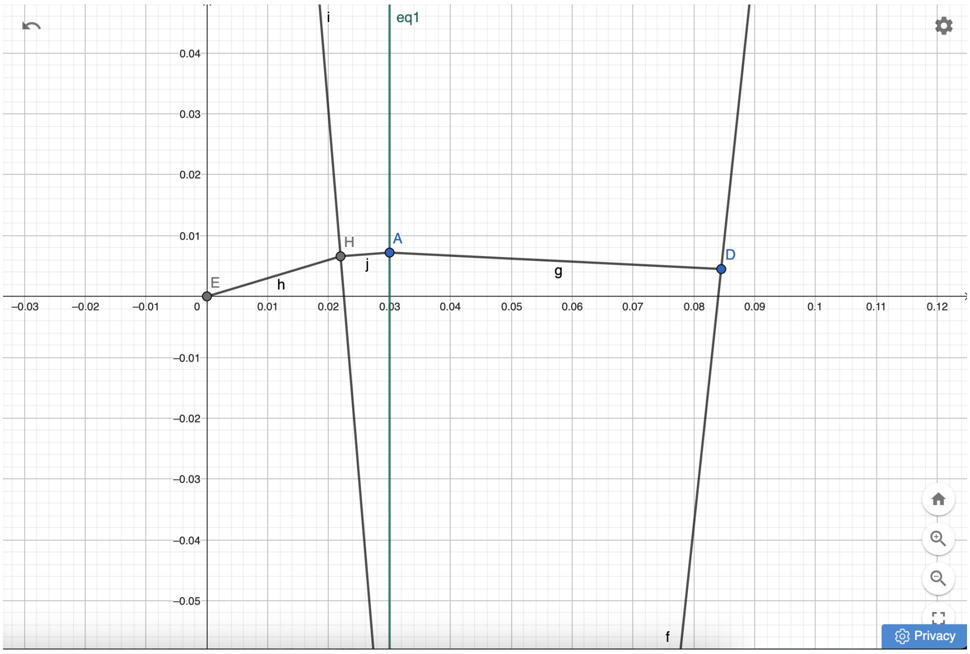


Figure 17: Scheme of the distances between the different extracted lines. $x = 0$ at the origin.

To have a more detailed explanation on how the the velocity profiles are considered in order to obtain δ, you can have a look at the Python notebook in the repository.
Looking at the single extracted δ values we notice that they are close to the theoretical value

$$\frac{0.12}{\sqrt{1000}} \approx 0.00379$$

Looking at the values at different x we notice the trend we expected ($\sqrt{x}$).

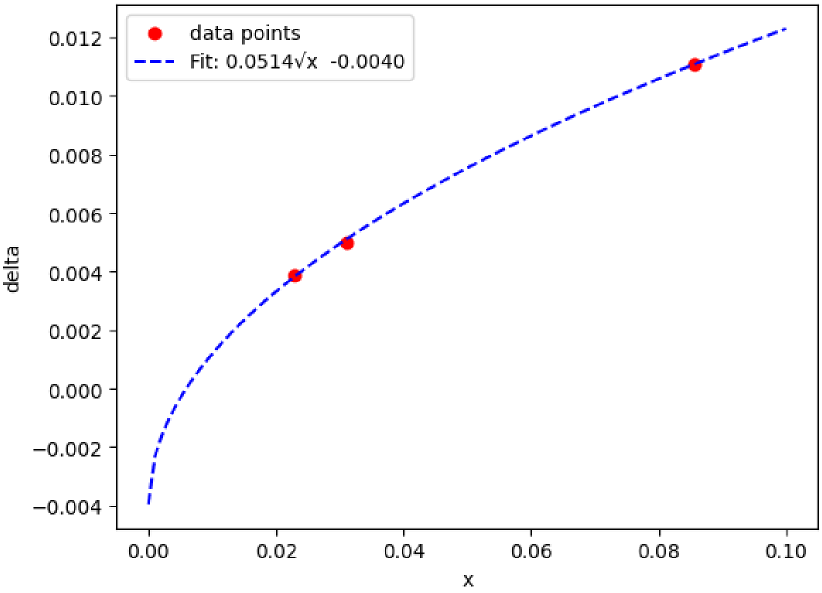


Figure 18: Thickness of boundary layer δ at different x.

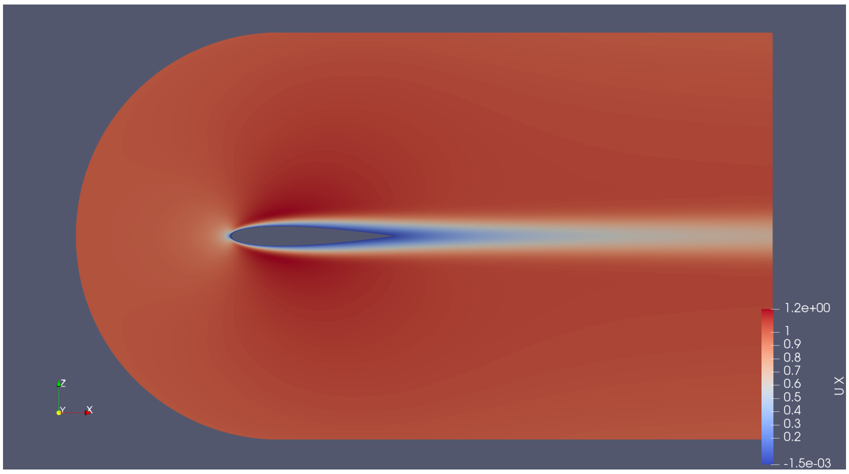


Figure 19: u field for the airfoil.

3.2 $Re = 4000$

Changing ν I obtain the equivalent case for $Re = 4000$. Here we should notice the same trend with the δ values that are half the values of the previous case. Indeed

$$\frac{\delta_{4000}}{\delta_{1000}} = \frac{\sqrt{1000}}{\sqrt{4000}} = 1/2$$

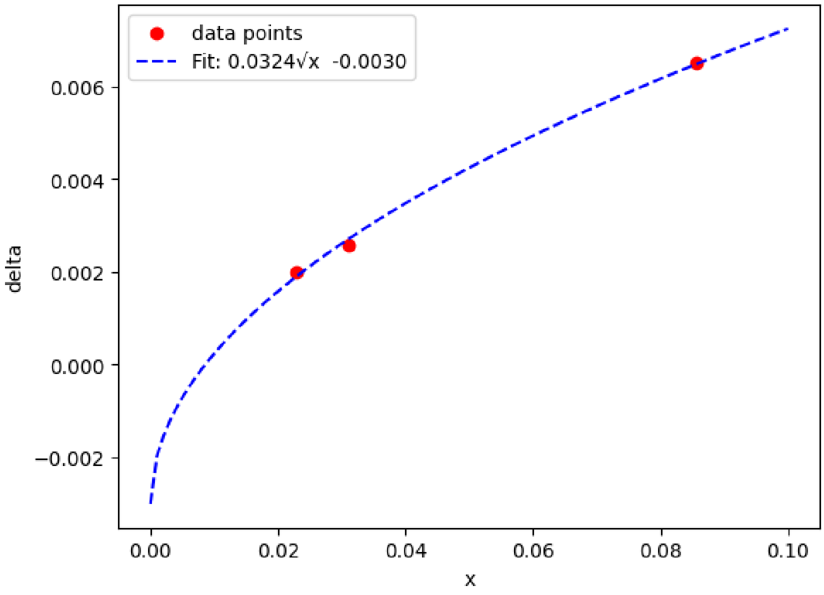


Figure 20: Thickness of boundary layer δ at different x.

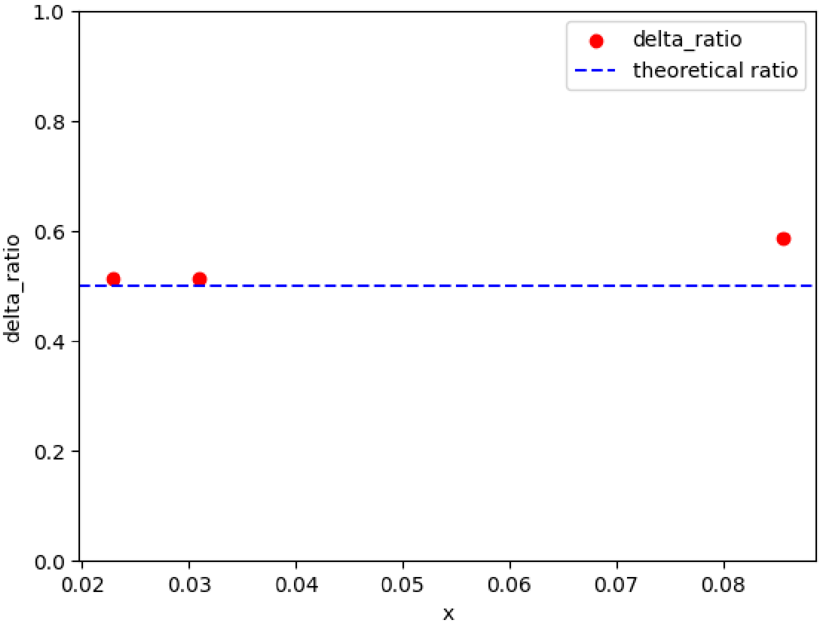


Figure 21: Visualization of $\delta_{4000}/\delta_{1000}$.

Fluctuations can be related to the fact that all the measurements have been done ”by eye”.

3.3 Different angle of attack $Re = 1000$

Changing the angle of attack ($\alpha = 0.2$) we can notice what is still a consequence of boundary layer theory, that is the recirculation of fluid due to a change of the sign in $\frac{\partial p}{\partial x}$. When this starts to happen we can locate the separation point starting from which we will have a turbulent behavior.
In the separation region, looking at the vertical velocity profile, we should be able to see both positive and negative u.

This simulation has been carried on with $dt = 0.00007$ to ensure stability. Since I do not need steady state, in this case I have simulated only $0.3s$.

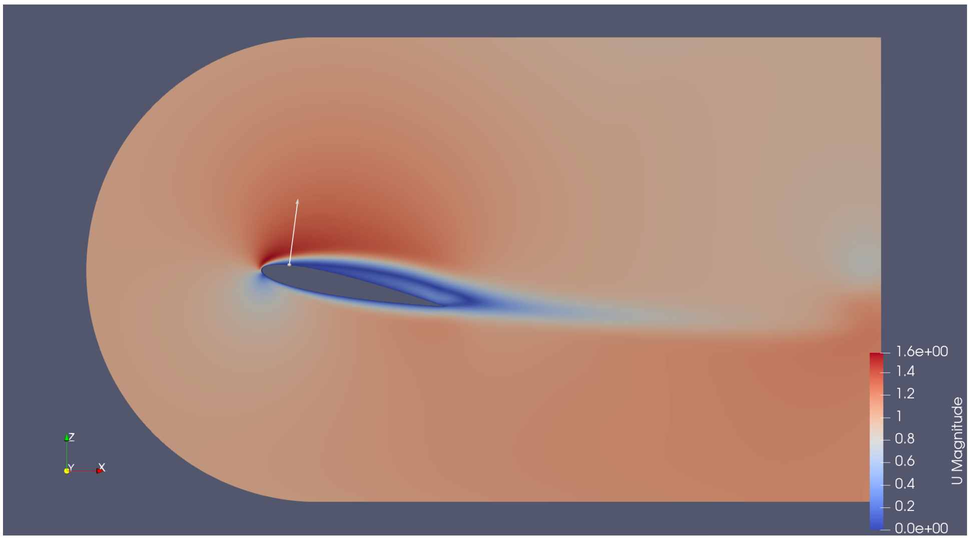


Figure 22: Velocity magnitude for tilted foil. The line highlights the beginning of separation region.

To better highlight the separation phenomenon and the velocity that changes its sign, I report here representations of the magnitude of the velocity (v, the vertical component is zero almost everywhere).

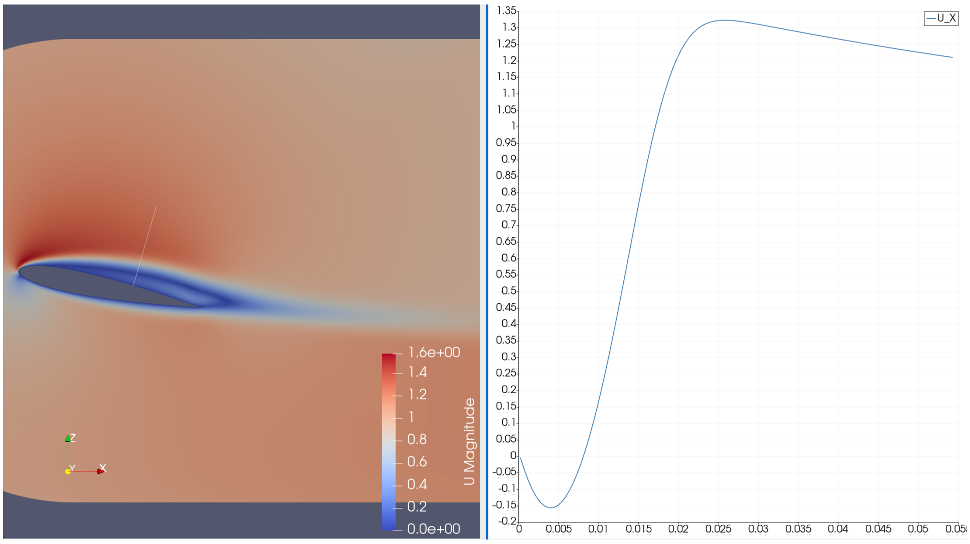


Figure 23: u profile along vertical line being both positive and negative.

Here you can clearly see the velocity in separation region being negative in the first region near the foil, going to zero, and going positive farther from the foil.

4 Airfoil, Bernoulli equation

Outside the boundary layer, considering the case of slip conditions, an another thing I can do is to check the validity of Bernoulli equation.
Bernoulli equation considers incompressible, steady, and non-viscous flow. Thanks to this assumptions the momentum equation becomes

$$u\frac{\partial u}{\partial s} = -\frac{1}{\rho}\frac{\partial p}{\partial s} - \frac{\partial}{\partial s}gz$$

where s is the coordinate following the trajectory of a single particle along the streamline.
Knowing that ρ is constant along s

$$\frac{\partial}{\partial s}(\frac{1}{2}u^2 + \frac{p}{\rho}z) = 0$$

To check this I extract a streamline near the foil (it will not matter since there are slip conditions) and calculate the supposed-constant value along it.

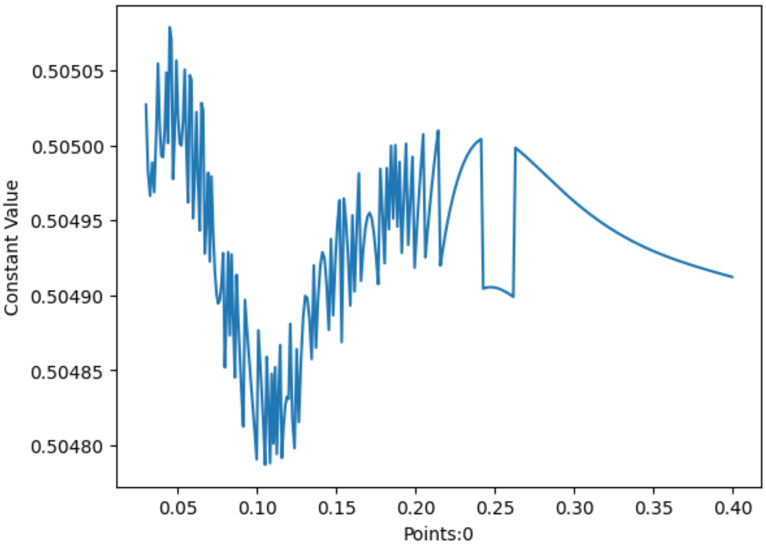


Figure 24: Values of the constant from Bernoulli equation along x of the streamline. $Re = 1000$

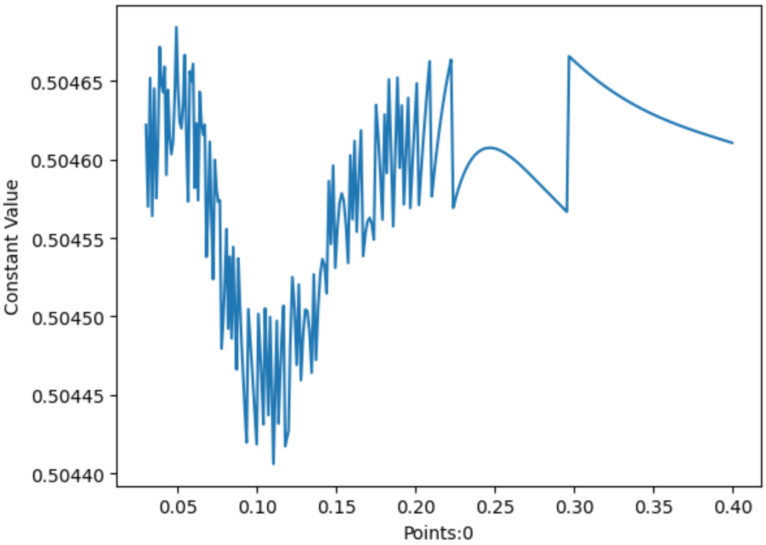


Figure 25: Values of the constant from Bernoulli equation along x of the streamline. $Re = 4000$

The small fluctuations we notice (around 4^{th} decimal) are probably due to numerical approximations.

References

[1] C. T. Shin U. Ghia K. N. Ghia. “High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method”. In: *Journal of Computational Physics* 48 (1982).