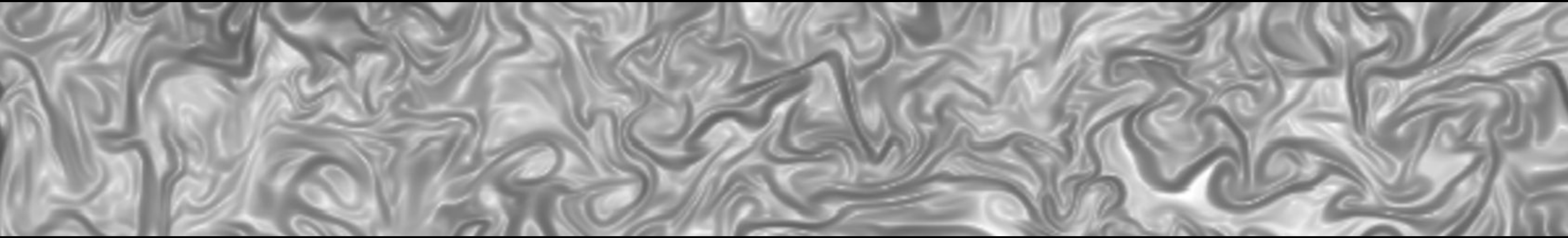
2D Wind Farm (Python) using Projection method for incompressible Navier Stokes

Project Documentation

Update: 15th August 24’



Governing Equations : 2D Incompressible Navier Stokes

$$\frac{du}{dx} + \frac{dv}{dy} = 0$$

Issue: There is no obvious way to couple pressure and velocity on the continuity equation.

$$\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy} = \frac{-1}{\rho}\frac{dp}{dx} + \nu\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right)$$

$$\frac{dv}{dt} + u\frac{dv}{dx} + v\frac{dv}{dy} = \frac{-1}{\rho}\frac{dp}{dy} + \nu\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)$$

Finding the way to couple velocity and pressure

Based on Laura Barba's lecture, velocity correction projection method.

We take the divergence of the momentum eq. by ... taking d/dx of 1) and d/dy of 2) and sum 'em.

$$1)\quad \frac{d}{dt}\left(\frac{du}{dx}\right) + \frac{du}{dx}\frac{du}{dx} + u\frac{d^2u}{dx^2} + \frac{dv}{dx}\frac{du}{dy} + v\frac{d^2u}{dxdy} = -\frac{1}{\rho}\frac{d^2p}{dx^2} + \nu\frac{d}{dx}(\nabla^2 u)$$

$$+$$

$$2)\quad \frac{d}{dt}\left(\frac{dv}{dy}\right) + \frac{du}{dy}\frac{dv}{dx} + u\frac{d^2v}{dydx} + \frac{dv}{dy}\frac{dv}{dy} + v\frac{d^2v}{dy^2} = -\frac{1}{\rho}\frac{d^2p}{dy^2} + \nu\frac{d}{dy}(\nabla^2 v)$$

$$\frac{d}{dt}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0} + \left(\frac{du}{dx}\right)^2 + \frac{du}{dy}\frac{dv}{dx} + \underbrace{u\frac{d^2u}{dx^2} + u\frac{d^2v}{dydx}}_{u\frac{d}{dx}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0}} + \frac{dv}{dx}\frac{du}{dy} + \left(\frac{dv}{dy}\right)^2 + \underbrace{v\frac{d^2u}{dxdy} + v\frac{d^2v}{dy^2}}_{v\frac{d}{dy}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0}}$$

$$= -\frac{1}{\rho}\left(\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2}\right) + \nu\underbrace{\left[\frac{d}{dx}\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) + \frac{d}{dy}\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)\right]}$$

$$\frac{d^3u}{dx^3} + \frac{d^3u}{dxdy} + \frac{d^3v}{dydx} + \frac{d^3v}{dy^2}$$

$$\frac{d^2}{dx^2}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0} + \frac{d^2}{dy^2}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0}$$

Apply continuity $\left(\frac{du}{dx} + \frac{dv}{dy} = 0\right)$

$$\left(\frac{du}{dx}\right)^2 + 2\left(\frac{du}{dy}\frac{dv}{dx}\right) + \left(\frac{dv}{dy}\right)^2$$

$$\frac{-1}{\rho}\left(\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2}\right) = \left(\frac{du}{dx}\right)^2 + 2\frac{du}{dy}\frac{dv}{dx} + \left(\frac{dv}{dy}\right)^2$$

$$\boxed{\nabla^2 p = -\rho}$$ Poisson Eq.

This poisson equation is supposed to ensure a divergence free velocity fields

The new governing equations

$$\left(\frac{d^2p}{dx^2}+\frac{d^2p}{dy^2}\right)=-\rho\left\{\left(\frac{du}{dx}\right)^2+2\frac{du}{dy}\cdot\frac{dv}{dx}+\left(\frac{dv}{dy}\right)^2\right\}$$

New Continuity eq. (Poisson eq.)

$$\frac{du}{dt}+u\frac{du}{dx}+v\frac{du}{dy}=\frac{-1}{\rho}\frac{dp}{dx}+\gamma\left(\frac{d^2u}{dx^2}+\frac{d^2u}{dy^2}\right)$$

X momentum

$$\frac{dv}{dt}+u\frac{dv}{dx}+v\frac{dv}{dy}=\frac{-1}{\rho}\frac{dp}{dy}+\gamma\left(\frac{d^2v}{dx^2}+\frac{d^2v}{dy^2}\right)$$

Y momentum

Let's discretize them

$$\frac{p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}}{\Delta x^2} + \frac{p^n_{i,j-1} - 2p^n_{i,j} + p^n_{i,j-1}}{\Delta y^2} = \rho\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\right)^2 - 2\left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\right)\left(\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x}\right) - \left(\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right)^2\right]$$

* 2nd order central difference

* 1st order central difference

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{\Delta t} + u^n_{i,j}\frac{u^n_{i,j} - u^n_{i-1,j}}{\Delta x} + v^n_{i,j}\frac{u^n_{i,j} - u^n_{i,j-1}}{\Delta y} = -\frac{1}{\rho}\frac{p^n_{i+1,j} - p^n_{i-1,j}}{2\Delta x} + \nu\left(\frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{\Delta x^2} + \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{\Delta y^2}\right)$$

* forward difference (time)

* Backward difference (space)

$$\frac{v^{n+1}_{i,j} - v^n_{i,j}}{\Delta t} + u^n_{i,j}\frac{v^n_{i,j} - v^n_{i-1,j}}{\Delta x} + v^n_{i,j}\frac{v^n_{i,j} - v^n_{i,j-1}}{\Delta y} = -\frac{1}{\rho}\frac{p^n_{i+1,j} - p^n_{i-1,j}}{2\Delta x} + \nu\frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{\Delta x^2} + \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{\Delta y^2}$$

And rearrange them in a explicit way

$$u_{ij}^{n+1} = u_{ij}^{n} - u_{ij}^{n}\frac{\Delta t}{\Delta x}\left(u_{ij}^{n} - u_{i-1j}^{n}\right) - v_{ij}^{n}\frac{\Delta t}{\Delta y}\left(u_{ij}^{n} - u_{ij-1}^{n}\right) - \frac{\Delta t}{\rho 2\Delta x}\left(p_{i+1j}^{n} - p_{i-1j}^{n}\right) + \nu\left(\frac{\Delta t}{\Delta x^2}\left(u_{i+1j}^{n} - 2u_{ij}^{n} + u_{i-1j}^{n}\right) + \frac{\Delta t}{\Delta y^2}\left(u_{ij+1}^{n} - 2u_{ij}^{n} + u_{ij-1}^{n}\right)\right)$$

$$v_{ij}^{n+1} = v_{ij}^{n} - u_{ij}^{n}\frac{\Delta t}{\Delta x}\left(v_{ij}^{n} - v_{i-1j}\right) - v_{ij}^{n}\frac{\Delta t}{\Delta y}\left(v_{ij}^{n} - v_{ij-1}^{n}\right) - \frac{\Delta t}{\rho 2\Delta y}\left(p_{ij+1}^{n} - p_{ij-1}^{n}\right) + \nu\left(\frac{\Delta t}{\Delta x^2}\left(v_{i+1j}^{n} - 2v_{ij}^{n} + v_{i-1j}^{n}\right) + \frac{\Delta t}{\Delta y^2}\left(v_{ij+1}^{n} - 2v_{ij}^{n} + v_{ij-1}^{n}\right)\right)$$

$$p_{ij}^{n} = \frac{\left(p_{i+1j}^{n} + p_{i-1j}^{n}\right)\Delta y^2 + \left(p_{ij+1}^{n} + p_{ij-1}^{n}\right)\Delta x^2}{2\left(\Delta x^2 + \Delta y^2\right)}$$

$$- \frac{\rho \Delta x^2 \Delta y^2}{2\left(\Delta x^2 + \Delta y^2\right)}\left[\boxed{\frac{1}{\Delta t}\left(\frac{u_{i+1j} - u_{i-1j}}{2\Delta x} + \frac{v_{ij+1} - v_{ij-1}}{2\Delta y}\right)} - \frac{u_{i+1j} - u_{i-1j}}{2\Delta x}\,\frac{u_{i+1j} - u_{i-1j}}{2\Delta x} - 2\,\frac{u_{ij+1} - u_{ij-1}}{2\Delta y}\,\frac{v_{i+1j} - v_{i-1j}}{2\Delta x} - \frac{v_{ij+1} - v_{ij-1}}{2\Delta y}\,\frac{v_{ij+1} - v_{ij-1}}{2\Delta y}\right]$$

¿Where does this term come from?

The algorithm (fractional step method)

1) Get pressure field by solving Poisson Eq. iterating.

2) Solve momentum equation explicitly

Previous algorithm (Matthias method)

Compute intermediate velocity solving momentum eq. avoiding the pressure gradient.

$$\frac{u^* - u^n}{\Delta t} = -(u^n \cdot \nabla) \cdot u^n + \nu \nabla^2 u^n$$

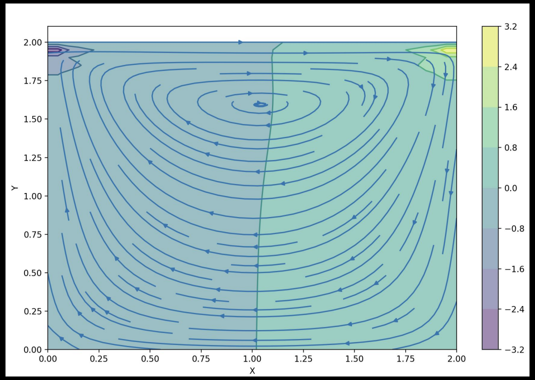
(convection + viscosity)
‖
¡ Burgers !

- Solve poisson equation thru an iteration → get pressure field
  - → Use u^* and v^*
- Obtain next step velocity by correcting the intermediate vel

$$u^{n+1} = u^* - \frac{\Delta t}{\rho} \nabla p^{n+1}$$

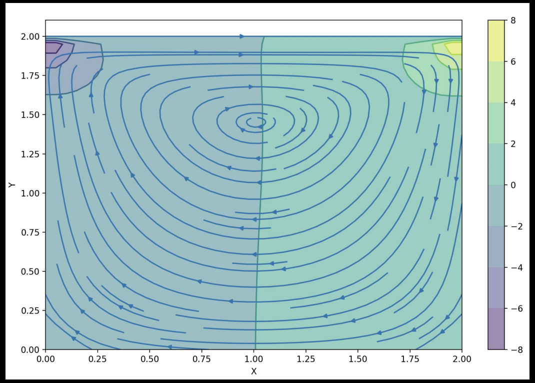
Cavity Flow

nu= 0.1



Steady State. Convergence is reached after 500 iterations.
The maximum absolute divergence is 7, regardless of the # iterations, and the summation of the divergence at all cells can be considered zero (e-16). This suggests that while the global (summed) divergence is near zero, locally there are regions with significant divergence.
Does it make any sense?
The poisson solver ain't doing anything.. I only need to compute the pressure once to obtain the same results regardless of how many more iterations.

nu= 0.5

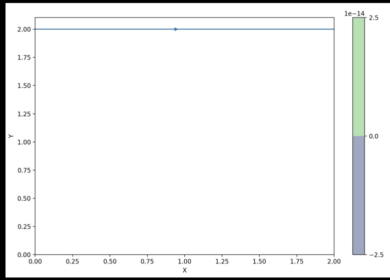


# Iterations (Poisson Solver)	Sum_Div	Max_Div
1	0.031	2.346
2	0.036	2.352
5	0.019	2.350
15	0.016	2.350
50	0.0145	2.350
100	0.0141	2.350

Max Div converges!

Sum Div converges!

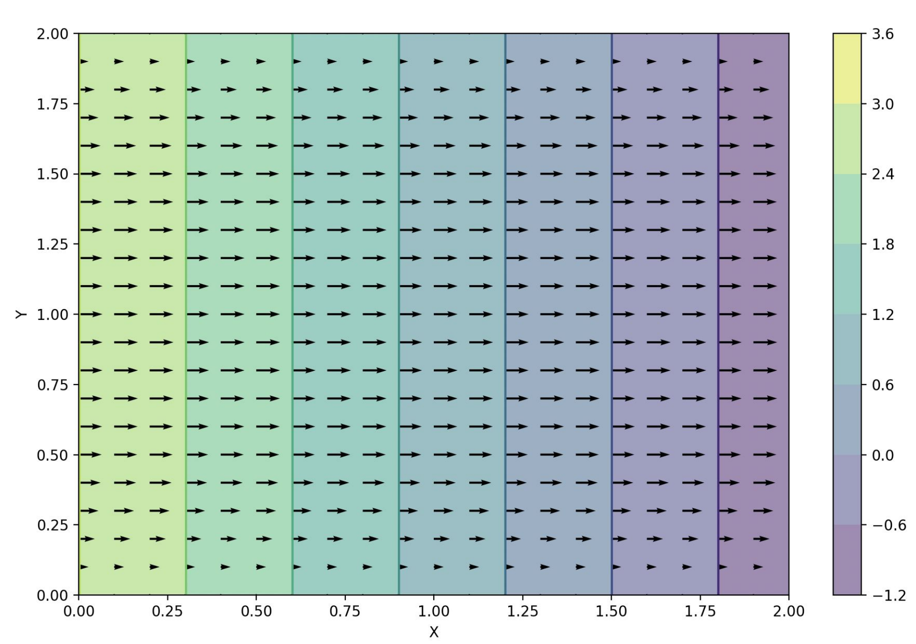
nu>0.6



Simulation crashes!!!
Why?

Channel Flow Neumann

- Substitute periodic BC's by Neuman BCs: dp/dx=0 at x=0,Xmax.
- Remove forcing term from Momentum eq.
- BC's:
    - dp/dx=F at x=0,Xmax (forces the flow at inlet and outlet)
    - dp/dy=0 at y=0, Ymax. (do nothing on p at walls)
    - u,v=0 at y=0,Ymax (non-slip)
    - du/dx=0 at x=0,Xmax (do nothing)



Sensitivity analysis.

- dt:
    - dt=0.003. Divergence grows with time. Solution makes sense until it blows up at around 1100.
    - dt= 0.002. Stable solution (good at iteration 200000!) Divergence stays zeroish.
- Poisson solver:

Iterations	Div	P_change
5	Large	blows up
15	Null	Grows Steady e-7
50	Null	Grows Steady e-7

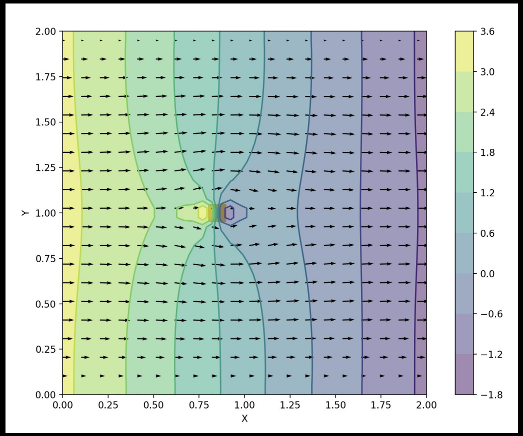
The steady state solution is reached after a single iteration! Then anything else changes if stable.

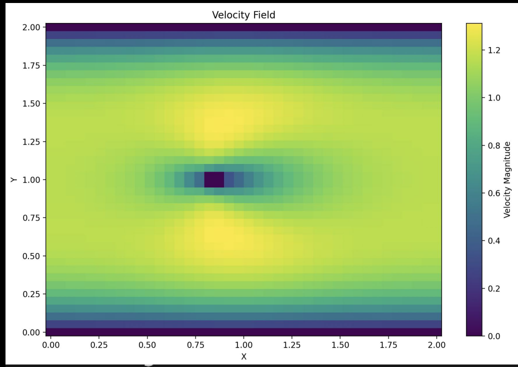
Might due to the BC dp/dx=F

Right, Peter?

Channel Flow with Square

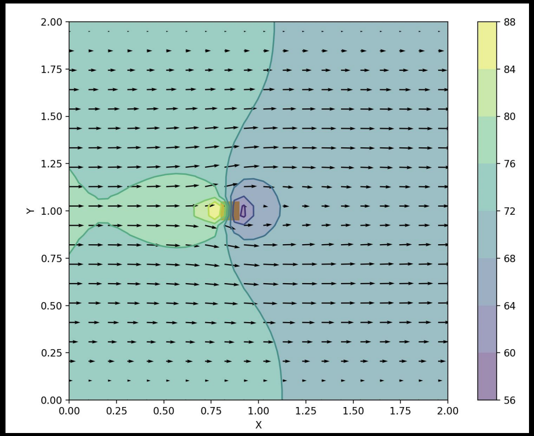
200 iterations

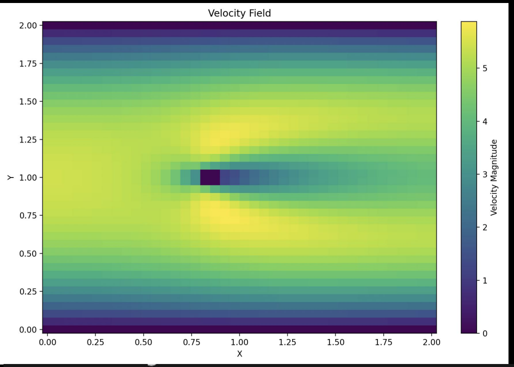





```
Max divergence:  2.7705924280282352
Sum divergence:  26.24037121867807
```

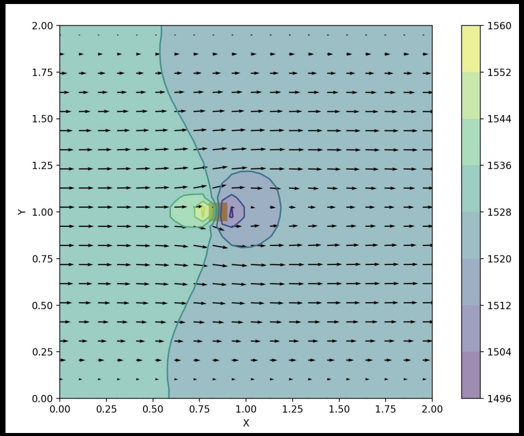
2000 iterations

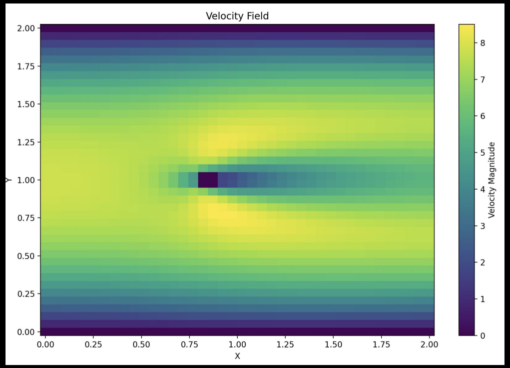





```
Max divergence:  16.92594778230421
Sum divergence:  166.79328286896322
```

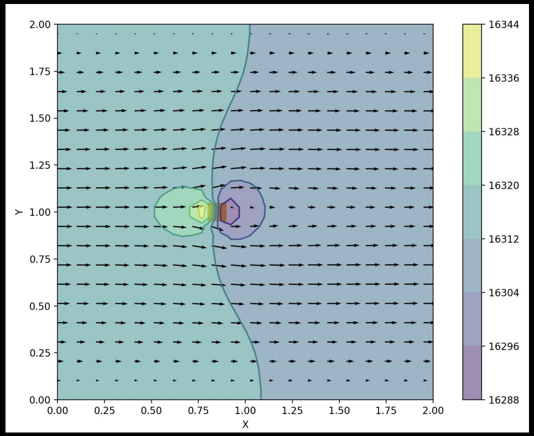
20000 iterations

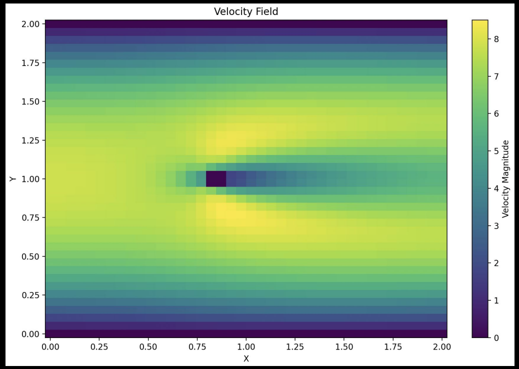





```
Max divergence:  27.566107859731265
Sum divergence:  266.53878731388414
```

200000 iterations





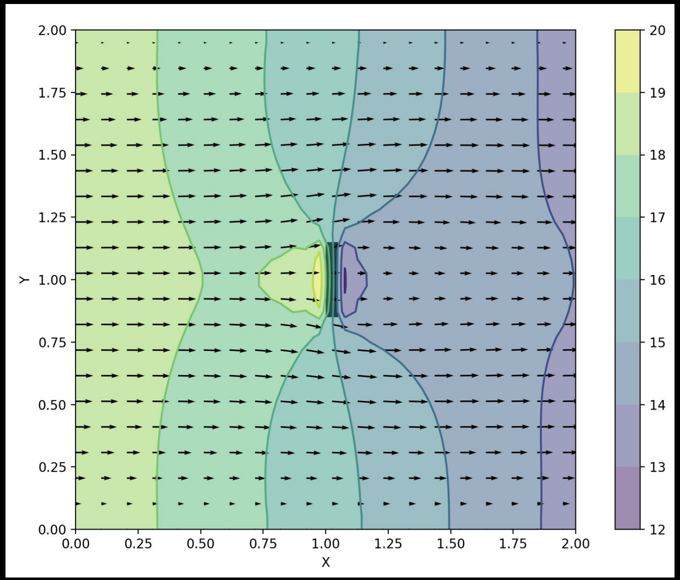

```
Max divergence:  27.566342128334014
Sum divergence:  266.5410032235841
```

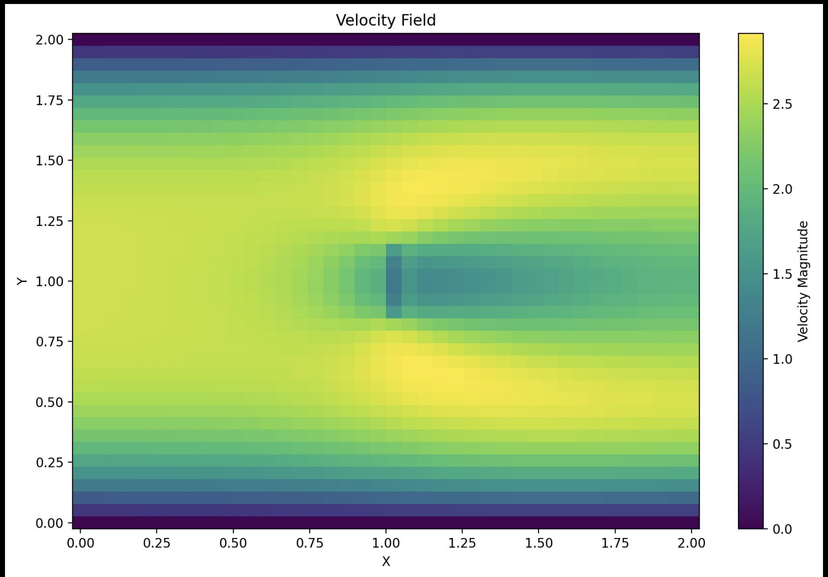
Divergence rises until the steady state is reached, when it stops growing (velocities do not evolve no more). Is this normal? Is it too high?

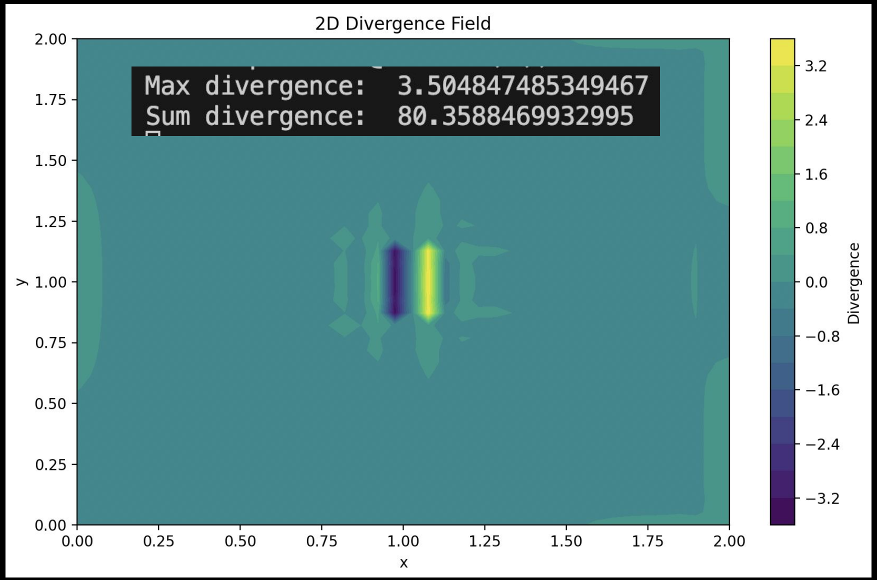
Channel Flow with Turbine (body force)

Applied the body force to the momentum eq itself.
Start using a randomly calibrated force magnitude.

Issues:
The force has a strong effect at the first steps but then it does not. So I need to set a weak force to run for a few steps and a strong force to







15 iterations, 800 steps, F=-2,

Optimizing the Poisson Solver

The solver needs more iterations for smaller steps and less for larger steps because it gets many iterations during the steps. So depending on the number of steps we need fewer.

- Channel flow, 20000 steps: 2 iterations is enough -> e-14 div.
- Channel flow, 20 steps: 15000 iterations to get e-14 div.

Channel with square

For more poisson iterations, the divergence gets concentrated near the obstacle, with higher maximum divergence values but the summation of the divergence is smaller, being closer to zero in the rest of the domain. Is it normal that this happens? Shouldn't the divergence be zero at the spots close to the object too?

It seems that, for a coarser mesh, it is normal that the divergence is high near irregularities because the small features are unresolved.

Can I solve the Poisson Equation using a direct method instead of using an iteration?

Plantear el problema y estudiar la mejor manera de resolver la ecuación:

- Direct methods: Matrix Inversion, LU decomposition.
- Iterative methods: Jacobi, Gauss-Seidel, successive over-relaxation, conjugate gradient, multigrid methods.
- Fast Fourier transforms.

Diffusive time step condition

Patankar discusses the discretization of the diffusion term and the associated stability criteria for explicit time-stepping schemes.

T_P^0 as a neighbor of T_P in the time direction.) Indeed, for this coefficient to be positive, the time step Δt would have to be small enough so that a_P^0 exceeds $a_E + a_W$. For uniform conductivity and $\Delta x = (\delta x)_e = (\delta x)_w$, this condition can be expressed as

$$\Delta t < \frac{\rho c (\Delta x)^2}{2k} . \qquad (4.39)$$

If this condition is violated, physically unrealistic results could emerge, because the negative coefficient implies that a higher T_P^0 results in a lower T_P. Equation (4.39) is the well-known stability criterion for the explicit scheme. It is interesting to note that we have been able to derive this from physical arguments based on one of our basic rules. The troublesome feature about condition (4.39) is that, as we reduce Δx to improve the spatial accuracy, we are forced to use a much smaller Δt.

I could take this principle and extrapolate it to viscous diffusion, substituting the thermal diffusivity by the viscous diffusivity coefficient. But… Does this make any sense? I need to verify whether the units make sense and the condition escalates accordingly.
I also need to extrapolate to the 2D domain.

$$\Delta t \leq \min\left(\frac{\Delta x^2}{2\nu}, \frac{\Delta y^2}{2\nu}\right)$$

Adding it to the CFL condition, I would get something like this:

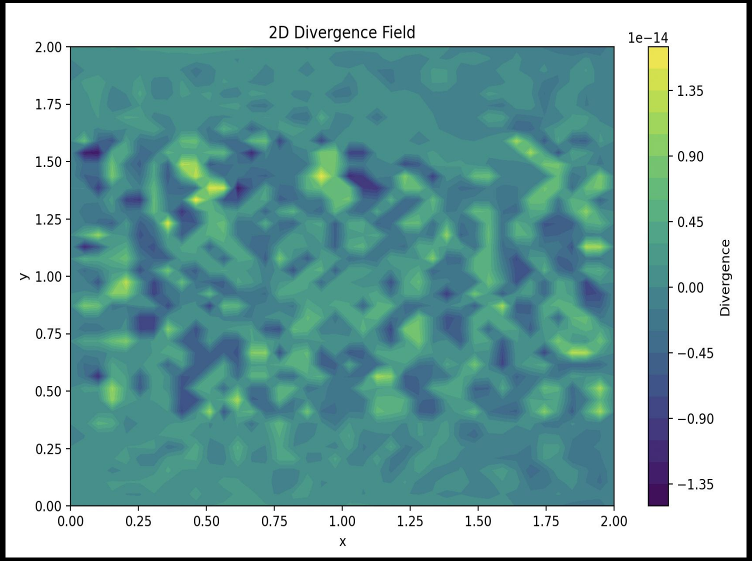
$$\Delta t \leq \min\left(\frac{\Delta x}{u_{\max}}, \frac{\Delta y}{v_{\max}}, \frac{\Delta x^2}{2\nu}, \frac{\Delta y^2}{2\nu}\right)$$

The condition is intended to ensure that the time step is small enough to capture the effects of diffusion without introducing numerical instability.

Divergence maps

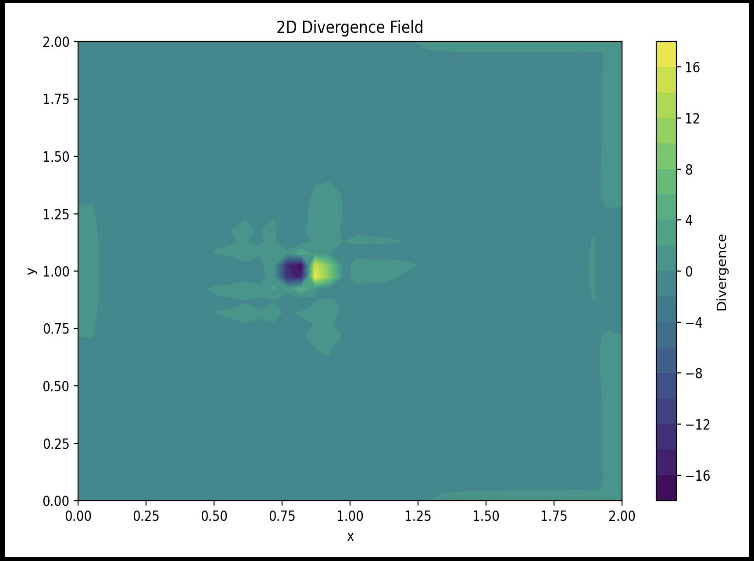
Central Difference Scheme for the interior cells.
Backward and Forward Difference Scheme for the boundary cells. (This is how I can compute all cells)

Channel Flow



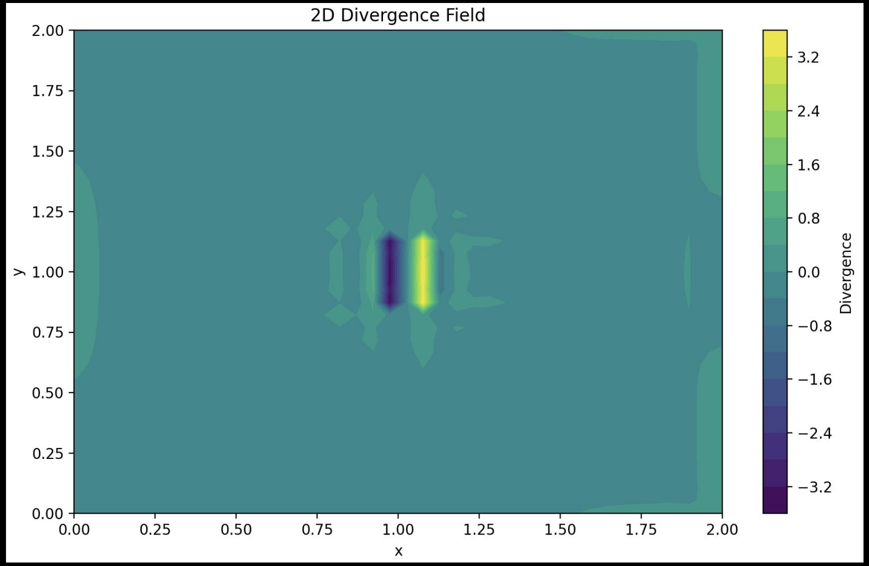
Zero divergence!

Channel Flow with square



Divergence is pretty high at the square limits.

Channel Flow with turbine (Axial Induction)



Channel Flow with Peter proposed BC’s

BC’s:

Inlet:

u=2, v=0

dP/dx=0

Outlet:

du/dx=0, dy/dx=0

dp/dx=0

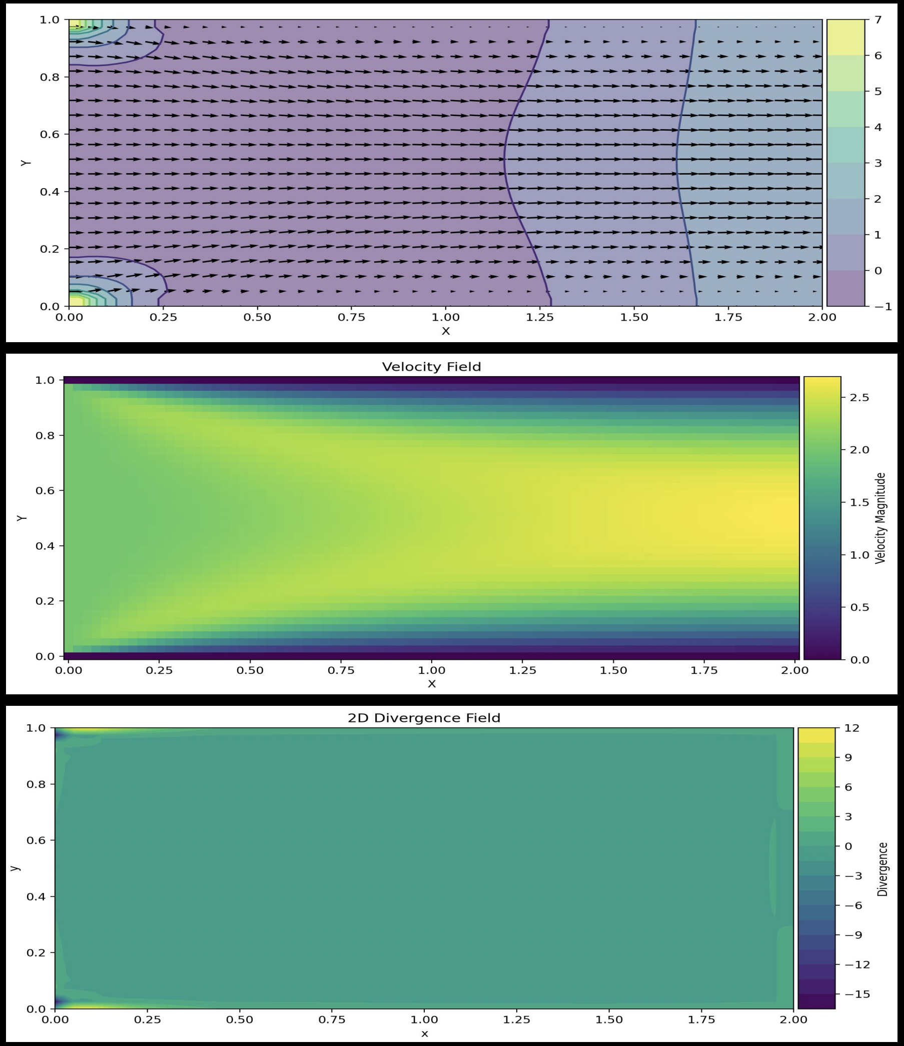
Top and bottom:

u=0, v=0 (non-slip)

dp/dy=0

Observations:

- The poisson solver has been optimized to #5 iterations.
- Divergence seems to be zero everywhere except of the entrance close to the edges, where it is pretty high.
- Solution looks realistic.



Channel Flow (Peter BC’s with Square)

BC’s:

Inlet:

u=2, v=0

dP/dx=0

Outlet:

du/dx=0, dy/dx=0

dp/dx=0

Top and bottom:

u=0, v=0 (non-slip)

dp/dy=0

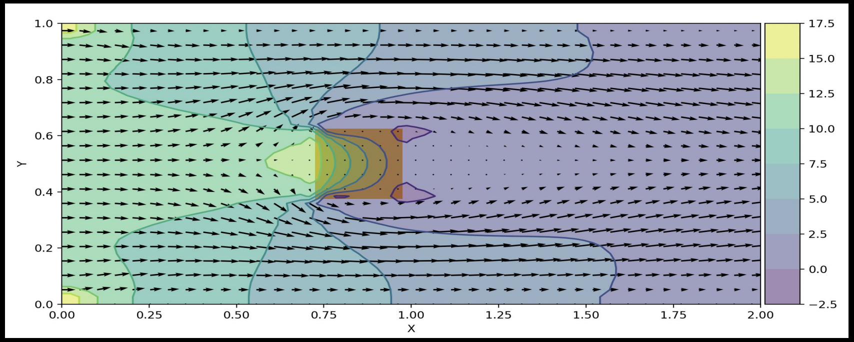
Observations:

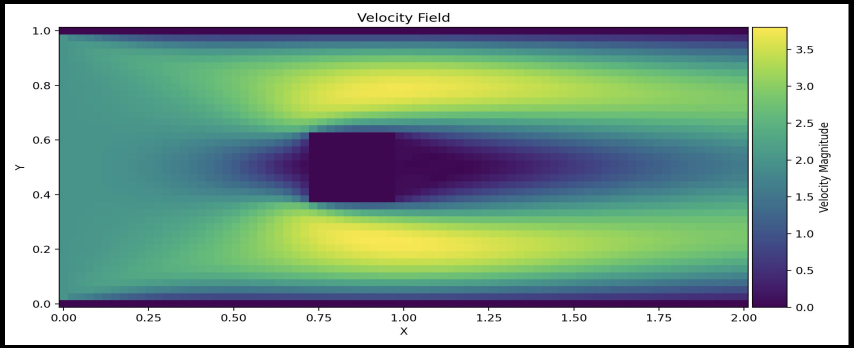
- The poisson solver has been optimized to #5 iterations.
- Divergence is high, why? Possibly:

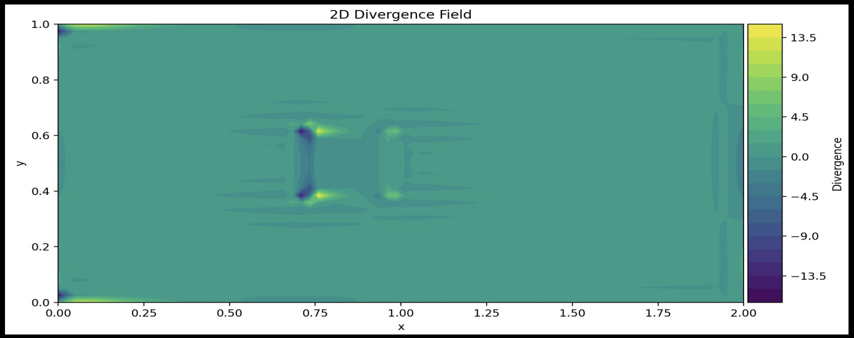
  A coarse grid might not adequately resolve the flow features near the boundaries, resulting in higher numerical divergence.

Suggestions:

Refining the mesh could be interesting.

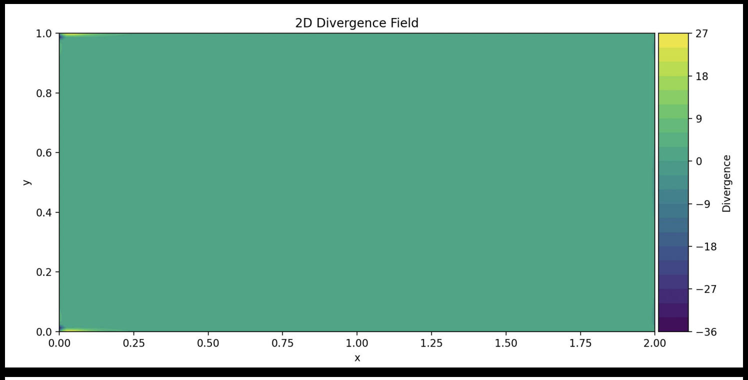


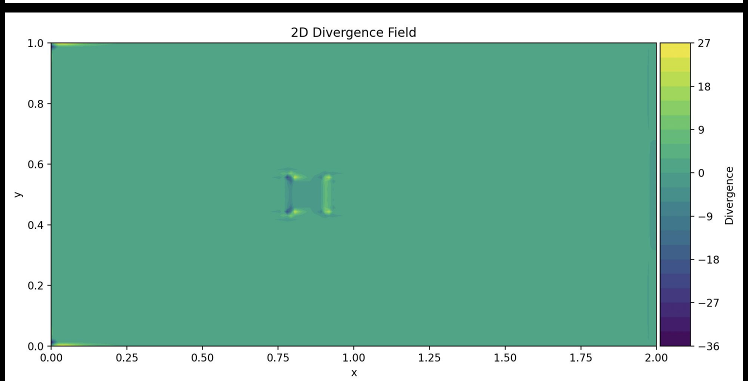




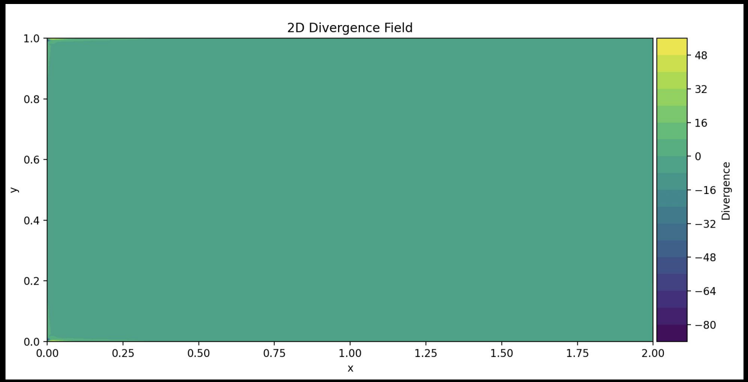
Refining the mesh, see how it affects to the divergence

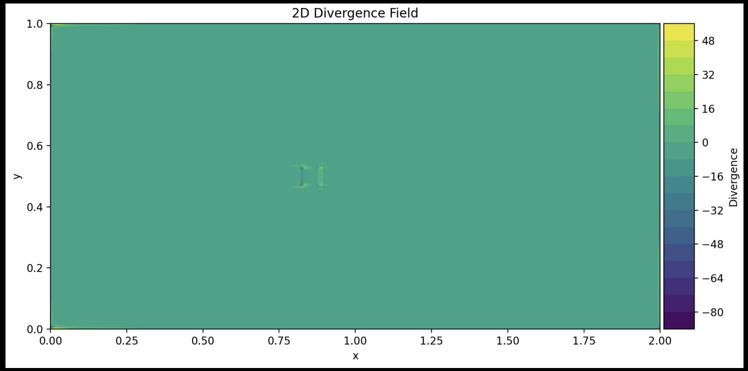
160x80



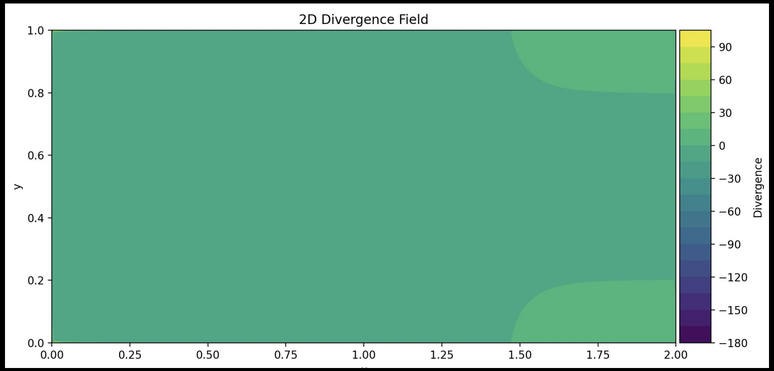


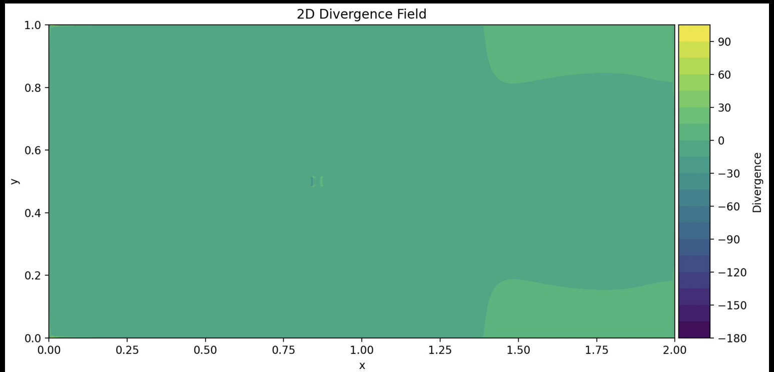
320x160





640x320





Divergence gets more concentrated on corners and edges, but the maximum value of divergence rises.

Poiseuille analytical solution

Fully Developed Laminar Pipe Flow

Perhaps the most useful exact solution of the Navier-Stokes equation is for incompressible flow in a straight circular pipe of radius R, first studied experimentally by G. Hagen in 1839 and J. L. Poiseuille in 1840. By *fully developed* we mean that the region studied is far enough from the entrance that the flow is purely axial, $v_z \neq 0$, while v_r and v_θ are zero. We neglect gravity and also assume axial symmetry—that is, $\partial/\partial\theta = 0$. The equation of continuity in cylindrical coordinates, Eq. (4.12*b*), reduces to

$$\frac{\partial}{\partial z}(v_z) = 0 \qquad \text{or} \qquad v_z = v_z(r) \qquad \text{only}$$

The flow proceeds straight down the pipe without radial motion. The r-momentum equation in cylindrical coordinates, Eq. (D.5), simplifies to $\partial p/\partial r = 0$, or $p = p(z)$ only. The z-momentum equation in cylindrical coordinates, Eq. (D.7), reduces to

$$\rho v_z \frac{\partial v_z}{\partial z} = -\frac{dp}{dz} + \mu \nabla^2 v_z = -\frac{dp}{dz} + \frac{\mu}{r}\frac{d}{dr}\left(r\frac{dv_z}{dr}\right)$$

The convective acceleration term on the left vanishes because of the previously given continuity equation. Thus the momentum equation may be rearranged as follows:

$$\frac{\mu}{r}\frac{d}{dr}\left(r\frac{dv_z}{dr}\right) = \frac{dp}{dz} = \text{const} < 0 \qquad (4.136)$$

This is exactly the situation that occurred for flow between flat plates in Eq. (4.132). Again the "separation" constant is negative, and pipe flow will look much like the plate flow in Fig. 4.12*b*.

Equation (4.136) is linear and may be integrated twice, with the result

$$v_z = \frac{dp}{dz}\frac{r^2}{4\mu} + C_1 \ln(r) + C_2$$

where C_1 and C_2 are constants. The boundary conditions are no slip at the wall and finite velocity at the centerline:

$$\text{No slip at } r = R: \quad v_z = 0 = \frac{dp}{dz}\frac{R^2}{4\mu} + C_1\ln(R) + C_2$$

$$\text{Finite velocity at } r = 0: \quad v_z = \text{finite} = 0 + C_1\ln(0) + C_2$$

To avoid a logarithmic singularity, the centerline condition requires that $C_1 = 0$. Then, from no slip, $C_2 = (-dp/dz)(R^2/4\mu)$. The final, and famous, solution for fully developed *Hagen-Poiseuille flow* is

$$\boxed{v_z = \left(-\frac{dp}{dz}\right)\frac{1}{4\mu}(R^2 - r^2)} \qquad (4.137)$$

Flow Due to Pressure Gradient between Two Fixed Plates

Case (*b*) is sketched in Fig. 4.12*b*. Both plates are fixed ($V = 0$), but the pressure varies in the x direction. If $v = w = 0$, the continuity equation leads to the same

conclusion as case (*a*)—namely, that $u = u(y)$ only. The x-momentum equation (4.130) changes only because the pressure is variable:

$$\mu\frac{d^2u}{dy^2} = \frac{\partial p}{\partial x} \qquad (4.132)$$

Also, since $v = w = 0$ and gravity is neglected, the y- and z-momentum equations lead to

$$\frac{\partial p}{\partial y} = 0 \qquad \text{and} \qquad \frac{\partial p}{\partial z} = 0 \qquad \text{or} \qquad p = p(x) \text{ only}$$

Thus the pressure gradient in Eq. (4.132) is the total and only gradient:

$$\mu\frac{d^2u}{dy^2} = \frac{dp}{dx} = \text{const} < 0 \qquad (4.133)$$

Why did we add the fact that dp/dx is *constant*? Recall a useful conclusion from the theory of separation of variables: If two quantities are equal and one varies only with y and the other varies only with x, then they must both equal the same constant. Otherwise they would not be independent of each other.

Why did we state that the constant is *negative*? Physically, the pressure must decrease in the flow direction in order to drive the flow against resisting wall shear stress. Thus the velocity profile $u(y)$ must have negative curvature everywhere, as anticipated and sketched in Fig. 4.12*b*.

The solution to Eq. (4.133) is accomplished by double integration:

$$u = \frac{1}{\mu}\frac{dp}{dx}\frac{y^2}{2} + C_1 y + C_2$$

The constants are found from the no-slip condition at each wall:

$$\text{At } y = \pm h: \quad u = 0 \quad \text{or} \quad C_1 = 0 \quad \text{and} \quad C_2 = -\frac{dp}{dx}\frac{h^2}{2\mu}$$

Thus the solution to case (*b*), flow in a channel due to pressure gradient, is

$$u = -\frac{dp}{dx}\frac{h^2}{2\mu}\left(1 - \frac{y^2}{h^2}\right) \qquad (4.134)$$

Poiseuille 2 plates analytical solution

This Poiseuille solution, also known as the Hagen-Poiseuille equation, describes the velocity profile of a steady, incompressible, laminar flow of a Newtonian fluid between 2 fix plates with no-slip boundary conditions. At the fully developed region.

$$u(y) = \frac{1}{2\mu}\frac{\partial p}{\partial x}(h^2 - y^2)$$

where nu: dynamic viscosity

dp/dx = pressure gradient (at the fully developed region)

h: half-distance between plates

y: varies from -h to h. (the origin is at the middle!) (h=H/2)

How to automatically set the length to reach a fully developed flow?

I can find out the length of the pipe where the flow gets fully developed, that way I do not have to compute extra cells. There are several ways to do this:

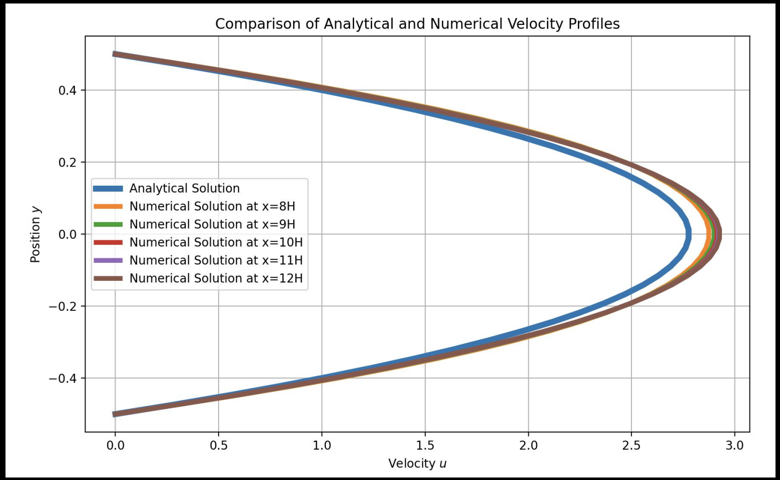
1. **Theoretically. Find the theoretical length where the flow gets fully developed. See Jeremy notes.**
2. **Practically. Do a test plotting the profiles at several lengths and find out where the flow gets fully developed, then reduce the pipe length in further simulations.**

Laminar: $\frac{L_e}{D} \approx 0.06\, Re$ Turbulent: $\frac{L_e}{D} \approx 4.4\, Re^{1/6}$

I'll use Re=30 (laminar), H=1 (D),
L_e= 0.06 * 30 * 1 = 1.8

This is shorter than what we had been seen..

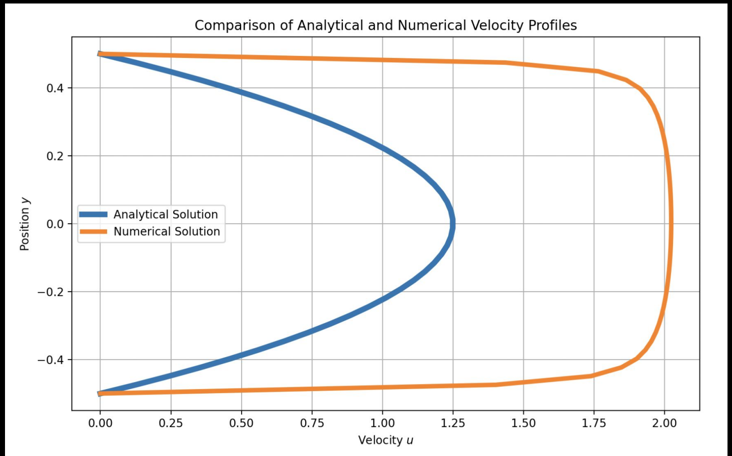
Practically…
I can take x=12H as converged.
It does not correspond to the theoretical result!!!



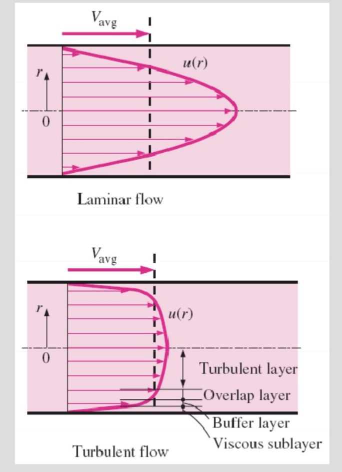
And… Why does it not correspond to the magnitude of the analytical result? Think that I am using a nu=0.0075 for the analytical result and a nu=0.5 for the numerical simulation!

Poiseuille Analytical solution - Numerical solution

I set p=2 at x=0 and p=0 at x=L. Then run my solver for a large #steps and compute the analytical Poiseuille solution.



The profiles do not match…
Maybe it is not fully developed?
Make pipe longer!!!
Maybe my flow is not laminar?
Check Re!!! Indeed, you could set the velocity in function of a defined Re.

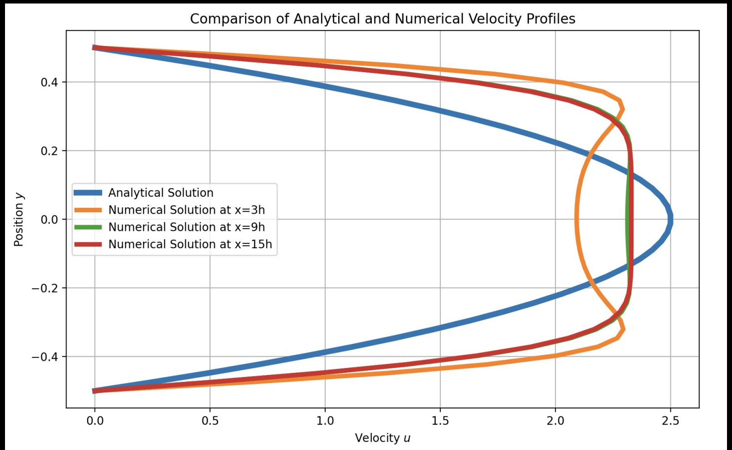


What velocity BCs are you using?

Colin said that I might need a length of 10-20 h to get a fully developed flow!! Let's try with 15h… And different BCs:

- p=1 at x=0
- p=0 at x=L

Still not dp/dx=constant but this is not possible because if p does not update divergence can't be eliminated… Any better suggestion??



My solution u(x) profile looks like a turbulent flow… Maybe I have a turbulent flow?? Let's check the Reynolds number:

Re=14

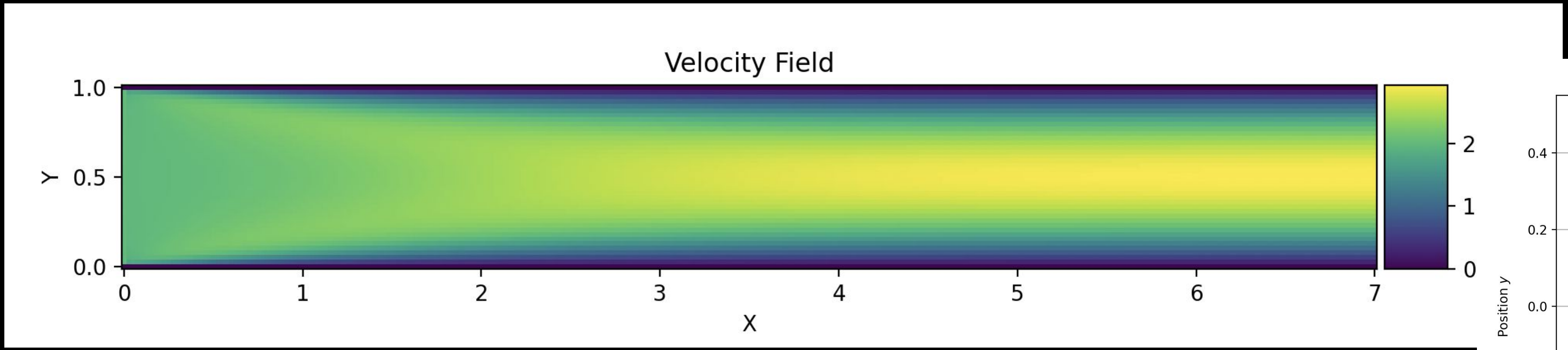
(<2300, laminar)

We need to reach the fully develop flow!!

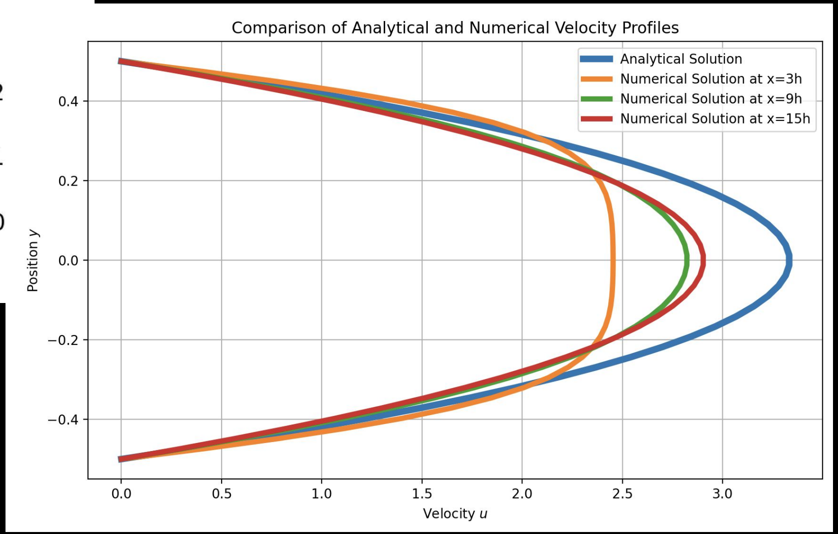
The reason why my profile is not parabolic is that the flow is not getting fully developed.
How can I get a fully developed flow?

- **I need to run the solver for longer.**
- **Rising the viscosity should lead into a faster development of the flow, since the velocity diffuses faster.**
  But the diffusion stability condition could reduce the timestep, leading into a not so faster convergence.
- **Let's just rise the viscosity, manually set the time step and run a long simulation, see what happens.**

nu=0.5, dt=0.00001, 1000000 steps, 60x600



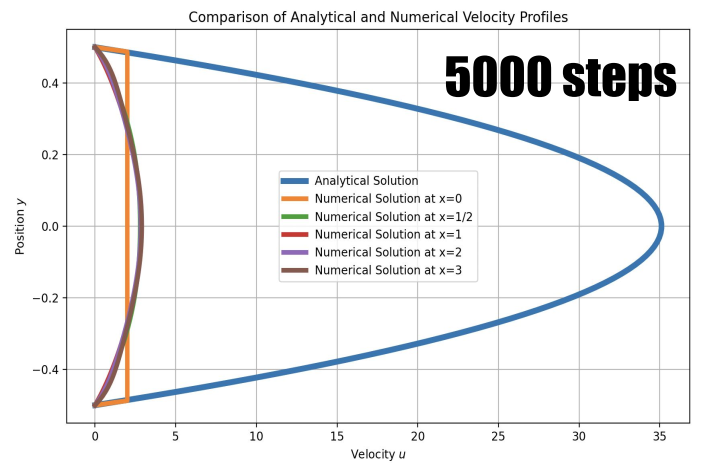
Bug alert!: x= 3,6 and 10 *H

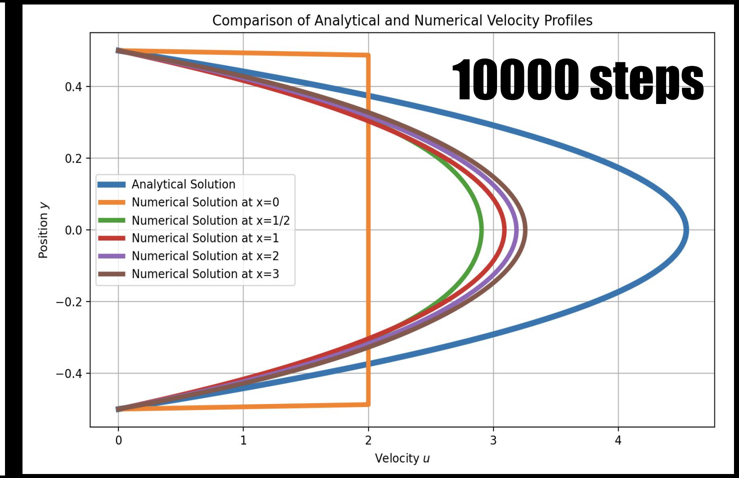


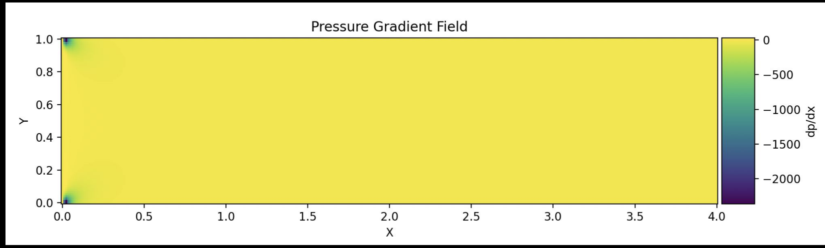
Something is wrong with this.. Fully developed flow should be reached earlier on the channel.

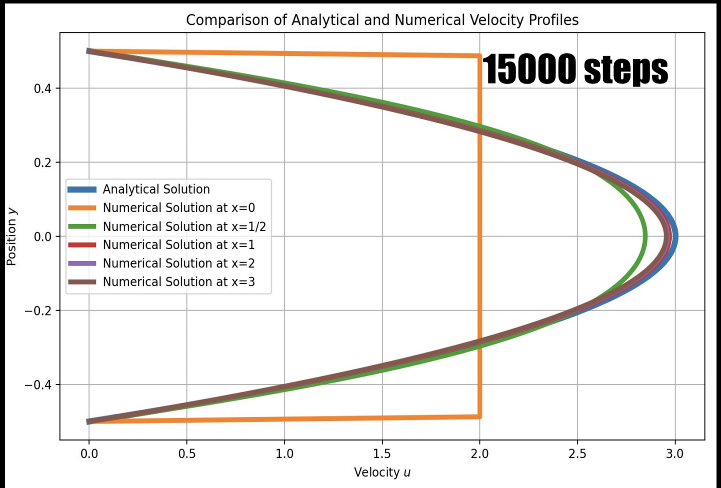
Poiseuille validation

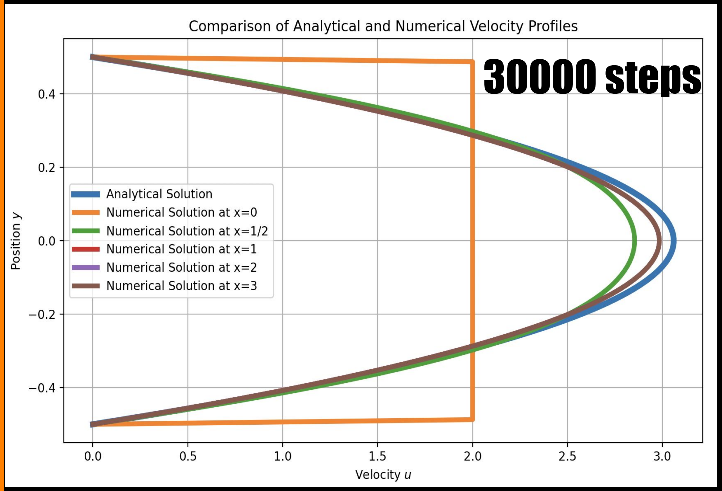
nu=0.5 80*320











nu=0.5

Steady State

The flow seems to be fully developed from the x=1 (x=H).

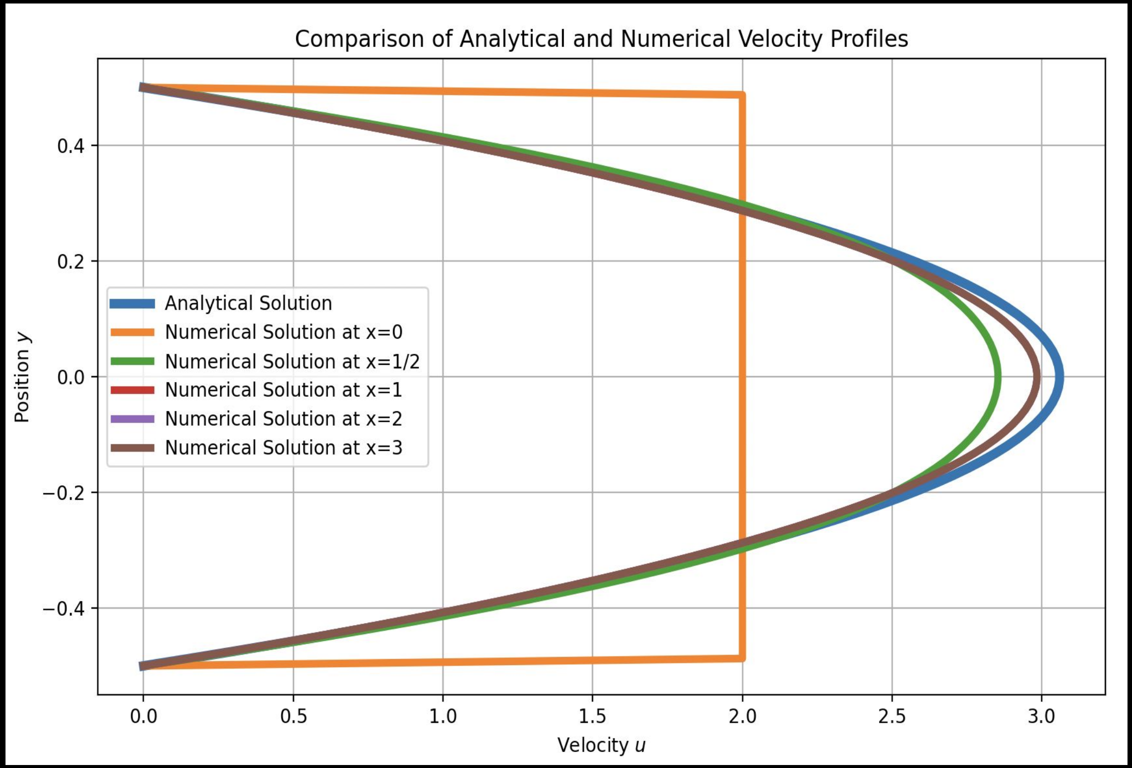
The solution is very close to the analytical one.. Why not exactly?

- Approximation error due to discretization? Then it would converge when increasing the resolution…
- Approximation error when computing dpdx.
- Needs to be a longer pipe and run for longer.

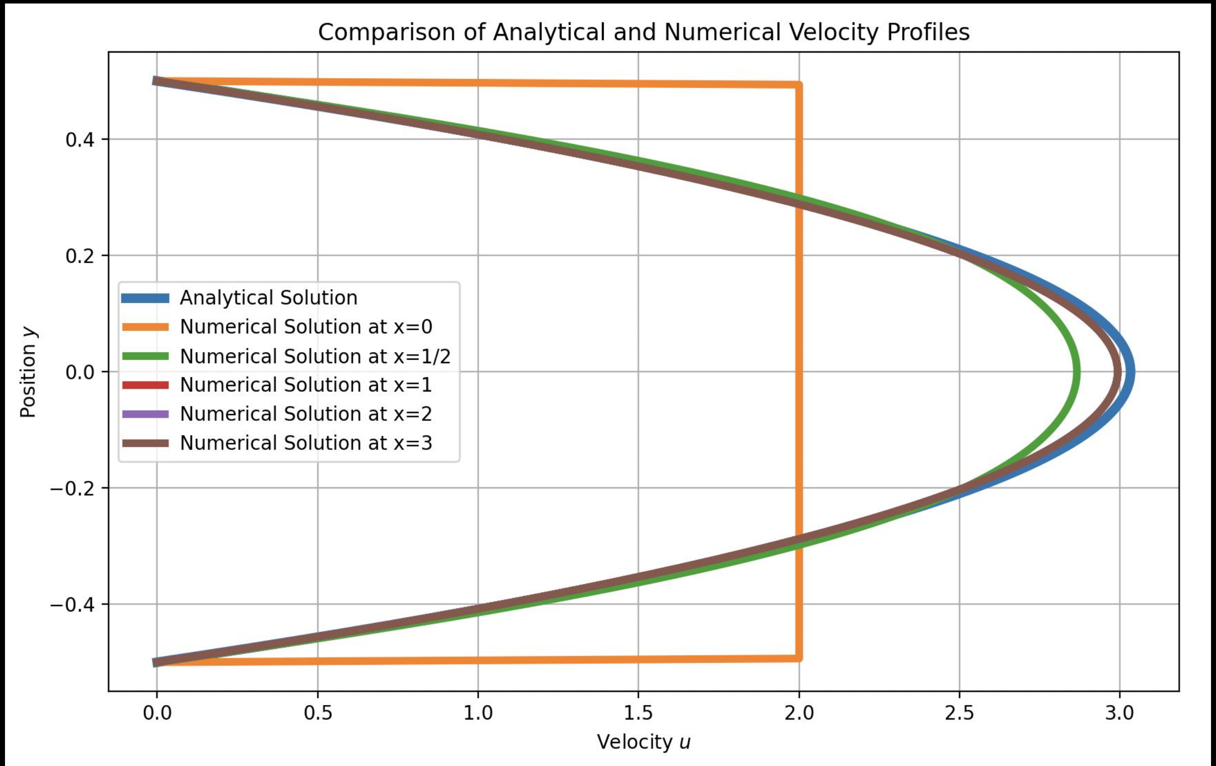
```
dpdx= (sim.p[int(L/2), -2]-sim.p[int(L/2), -1])/sim.dx
```

Poiseuille validation - Increase resolution (numY=160)

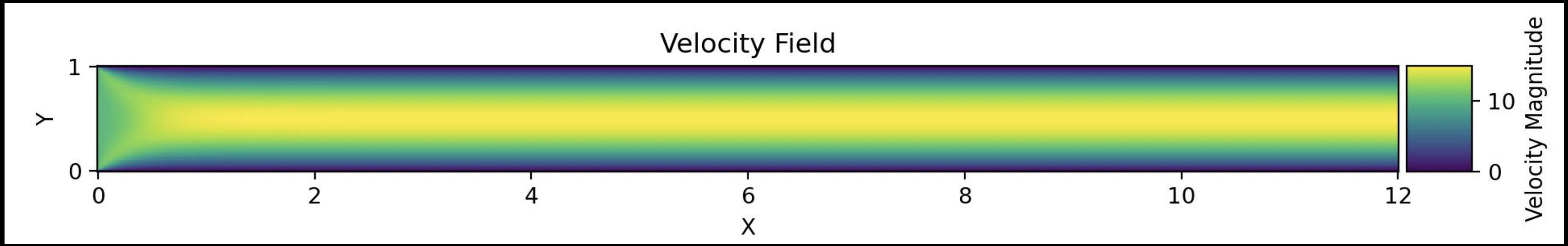
80*320

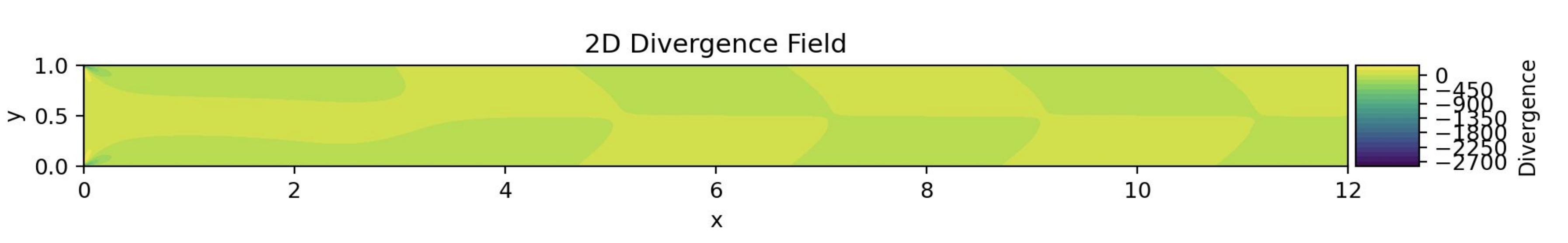


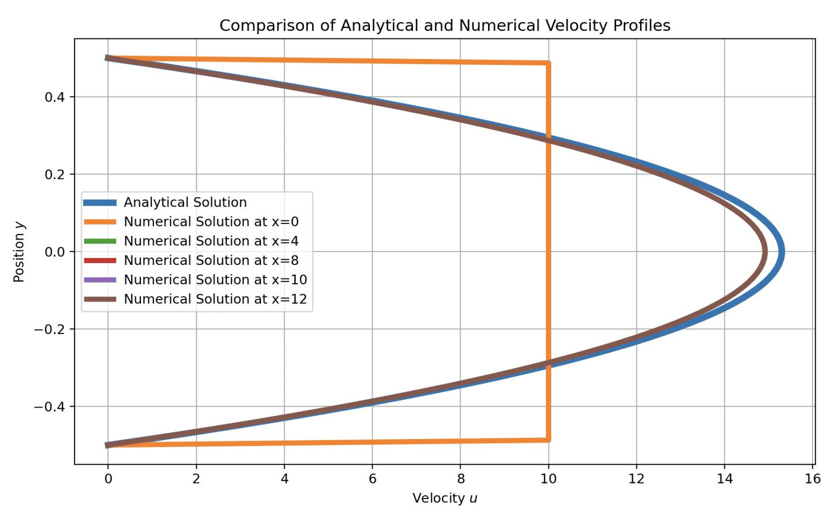
160*640



Poiseuille validation - More steps, Longer pipe



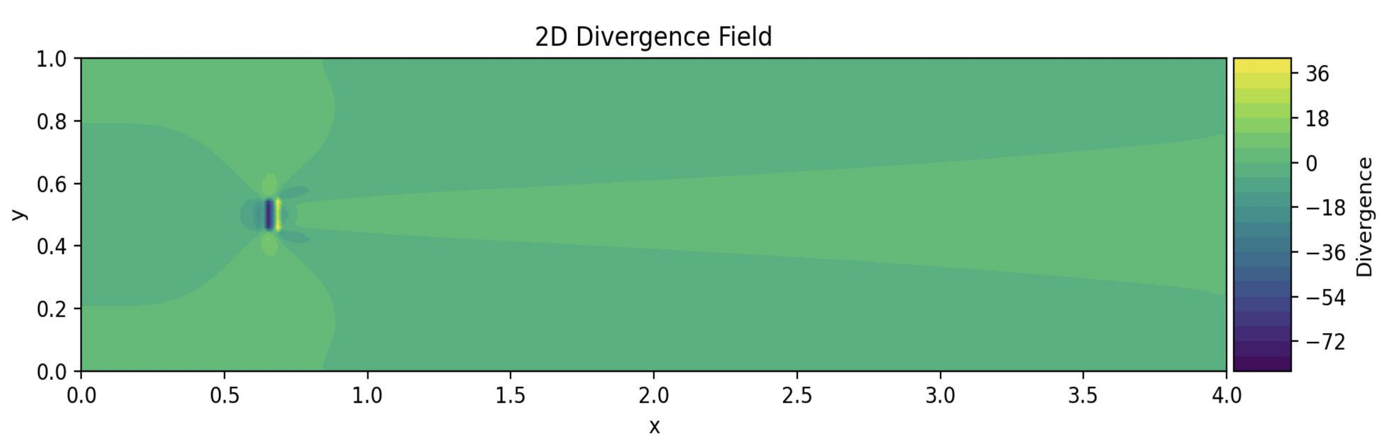


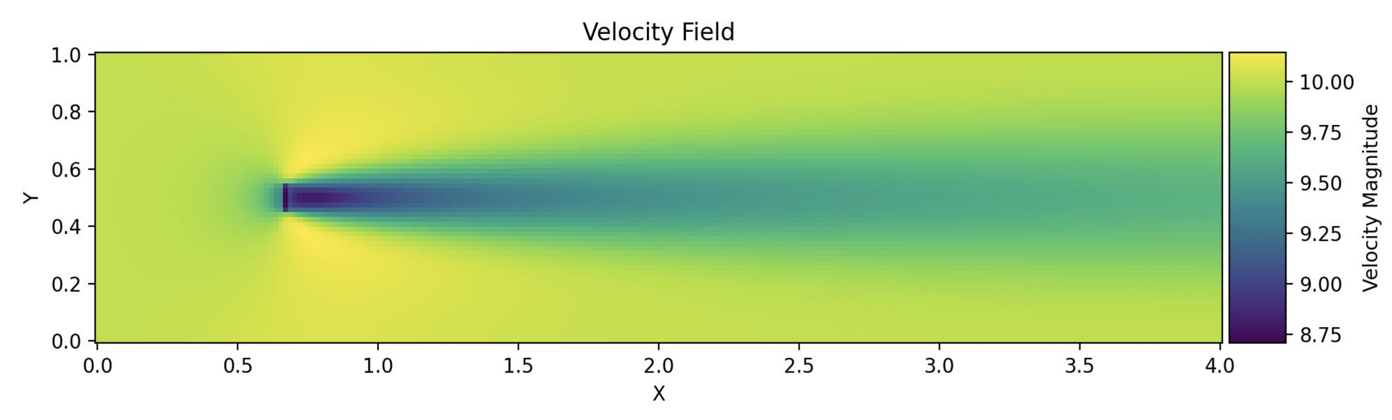


Channel Flow with body force F and no walls.

BC's:
- **Velocity.** **Inlet: Dirichlet** **Outlet, Top, Bottom: Neumann**
- **Pressure.** **Inlet, Top, Bottom: Neumann** **Outlet: Dirichlet**

nu=0.05
P iterations= 5
F= -1000
inVel= 10
60x240





Applied the axial induction body force (F=-1000) to the momentum eq itself.
No changes on the Poisson equation (because the divergence of a constant force is zero).

The divergence is high on the edges of the turbines!!!

The reason must be that we are not accounting for the F divergence on the Poisson equation.

Let's try to account for that.

Adding divergence of F term to Poisson Eq.

$$\frac{du}{dx} + \frac{dv}{dy} = 0$$

I'll start with just horizontal axial induction.

$$\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy} = \frac{-1}{\rho}\frac{dp}{dx} + \nu\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) + F$$

$$\frac{dv}{dt} + u\frac{dv}{dx} + v\frac{dv}{dy} = \frac{-1}{\rho}\frac{dp}{dy} + \nu\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)$$

Take sum of divergences

This is not null because F is not constant because it is only applied to certain cells

1) $$\overset{0}{\frac{d}{dt}\left(\frac{du}{dx}\right)} + \overbrace{\frac{du}{dx}\frac{du}{dx} + u\frac{d^2u}{dx^2}}^{1} + \overbrace{\frac{dv}{dx}\frac{du}{dy} + v\frac{d^2u}{dxdy}}^{2} = -\frac{1}{\rho}\overset{3}{\frac{d^2p}{dx^2}} + \nu\overset{4}{\frac{d}{dx}}(\nabla^2 u) + \frac{dF}{dx} + \frac{dF}{dy}$$

$+$

2) $$\frac{d}{dt}\left(\frac{dv}{dy}\right) + \frac{du}{dy}\frac{dv}{dx} + u\frac{d^2v}{dydx} + \frac{dv}{dy}\frac{dv}{dy} + v\frac{d^2v}{dy^2} = -\frac{1}{\rho}\frac{d^2p}{dy^2} + \nu\frac{d}{dy}(\nabla^2 v)$$

$$\frac{d}{dt}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0} + \left(\frac{du}{dx}\right)^2 + \frac{du}{dy}\frac{dv}{dx} + \underbrace{u\frac{d^2u}{dx^2} + u\frac{d^2v}{dydx}}_{u\frac{d}{dx}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0}} + \frac{dv}{dx}\frac{du}{dy} + \left(\frac{dv}{dy}\right)^2 + \underbrace{v\frac{d^2u}{dxdy} + v\frac{d^2v}{dy^2}}_{v\frac{d}{dy}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0}} = -\frac{1}{\rho}\left(\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2}\right) + \nu\underbrace{\left[\frac{d}{dx}\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) + \frac{d}{dy}\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)\right]}$$

$$\frac{d^3u}{dx^3} + \frac{d^3u}{dxdy} + \frac{d^3v}{dydx} + \frac{d^3v}{dy^2}$$

$$\frac{d^2}{dx^2}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0} + \frac{d^2}{dy^2}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0}$$

Apply continuity $\left(\frac{du}{dx} + \frac{dv}{dy} = 0\right)$

$$\left(\frac{du}{dx}\right)^2 + 2\left(\frac{du}{dy}\frac{dv}{dx}\right) + \left(\frac{dv}{dy}\right)^2$$

$$\boxed{\frac{-1}{\rho}\left(\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2}\right) = \left(\frac{du}{dx}\right)^2 + 2\frac{du}{dy}\cdot\frac{dv}{dx} + \left(\frac{dv}{dy}\right)^2 - \frac{dF}{dx} - \frac{dF}{dy}}$$

let's discretize it...

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

$$\rho\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\right)^2 - 2\overset{3}{\left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\right)}\left(\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x}\right) - \left(\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right)^2 - \frac{F_{i,j+1} - F_{i,j-1}}{2\Delta y} - \frac{F_{i+1,j} - F_{i-1,j}}{2\Delta x}\right]$$

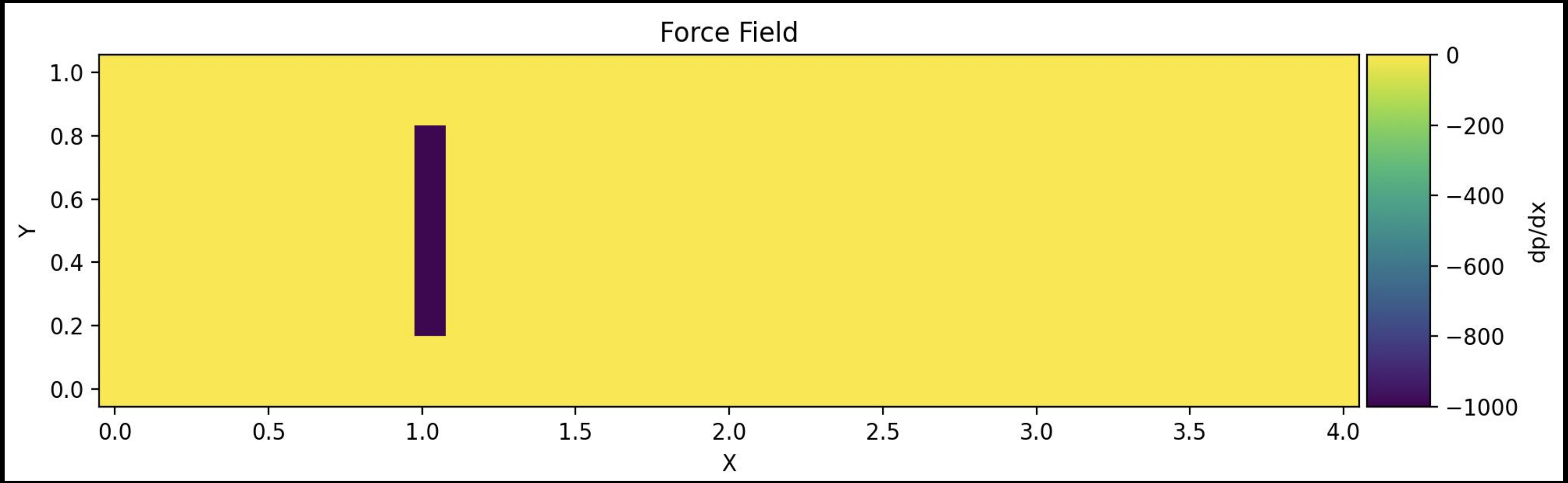
And re-arrange

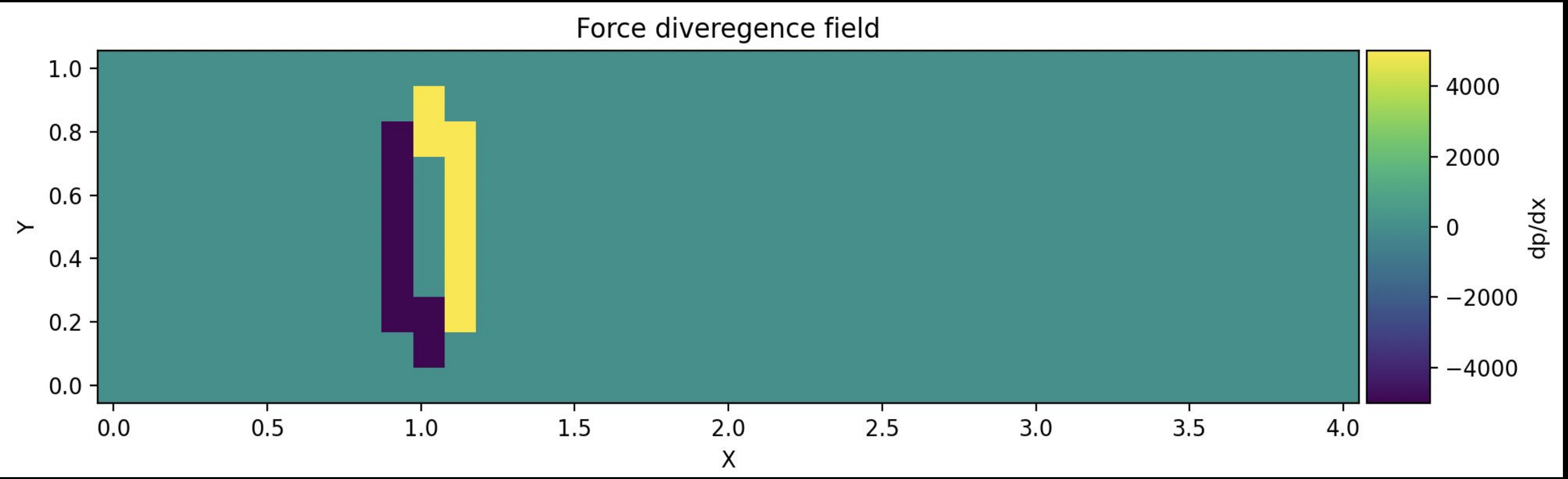
$$p^n_{ij} = \frac{(p^n_{i+1,j} + p^n_{i-1,j})\Delta y^2 + (p^n_{i,j+1} + p^n_{i,j-1})\Delta x^2}{2(\Delta x^2 + \Delta y^2)}$$

$$-\frac{\rho\Delta x^2\Delta y^2}{2(\Delta x^2 + \Delta y^2)}\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} - 2\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} - \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} - \frac{F_{i,j+1} - F_{i,j-1}}{2\Delta y} - \frac{F_{i+1,j} - F_{i-1,j}}{2\Delta x}\right]$$

$$\boxed{-\frac{F_{i,j+1} - F_{i,j-1}}{2\Delta y} - \frac{F_{i+1,j} - F_{i-1,j}}{2\Delta x}}$$

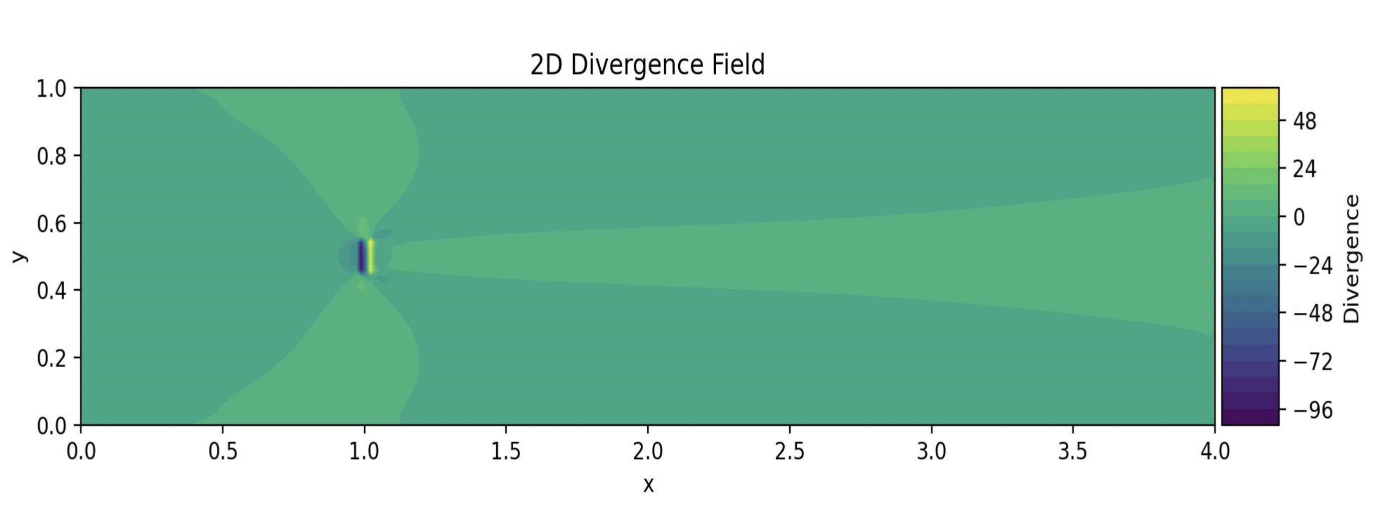
Lets see how F and the divergence of the F look like…

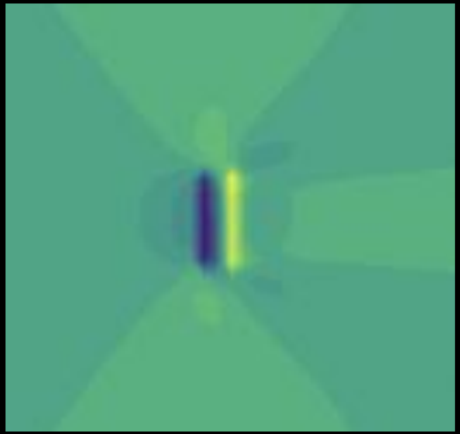




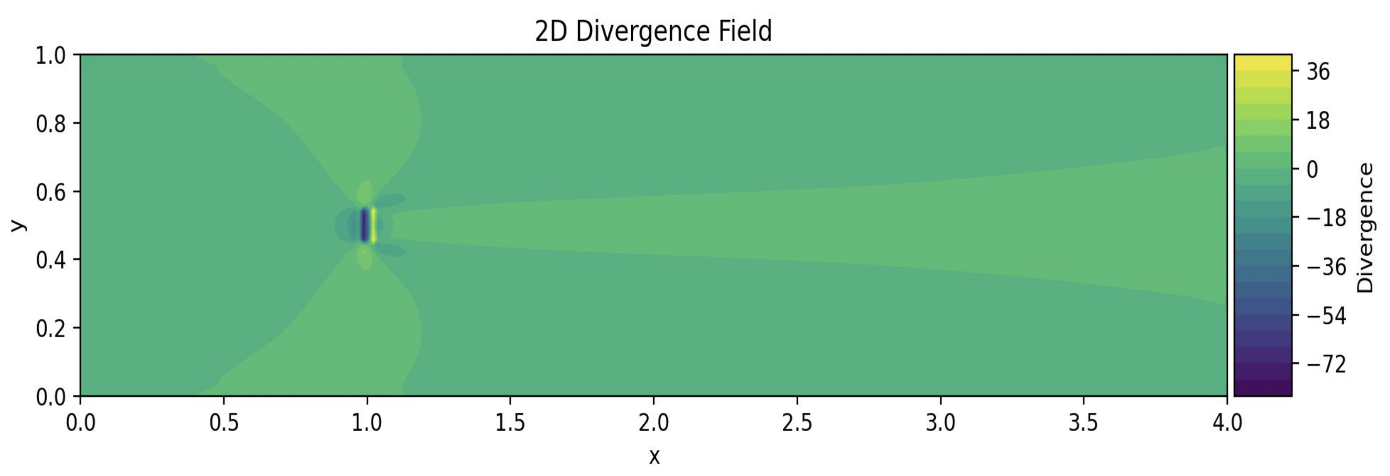
Add new term to the Poisson eq, solutions:

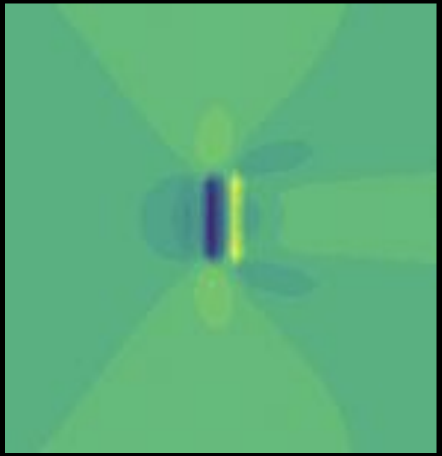
With F term





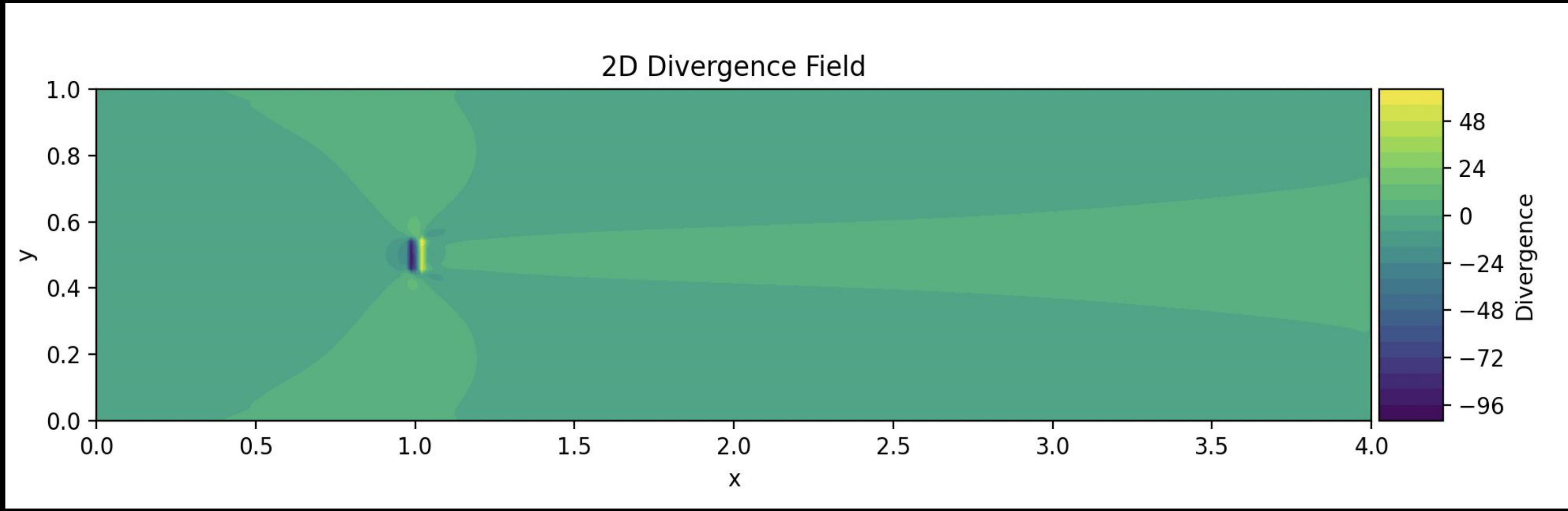
Without F term

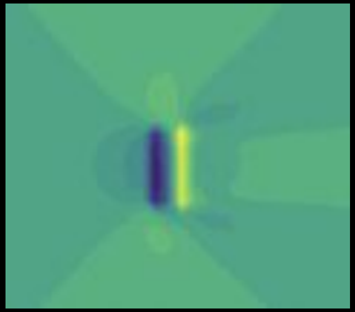




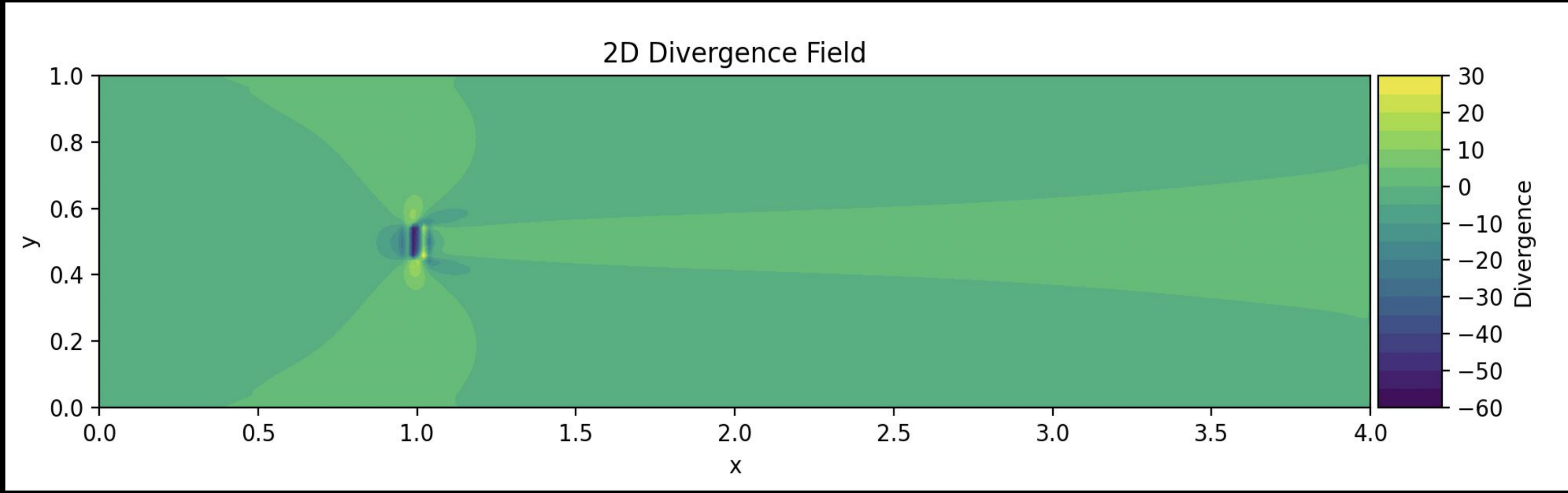
Am I using the right sign?

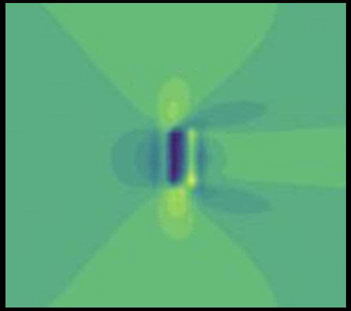
Negative F term





Positive F term

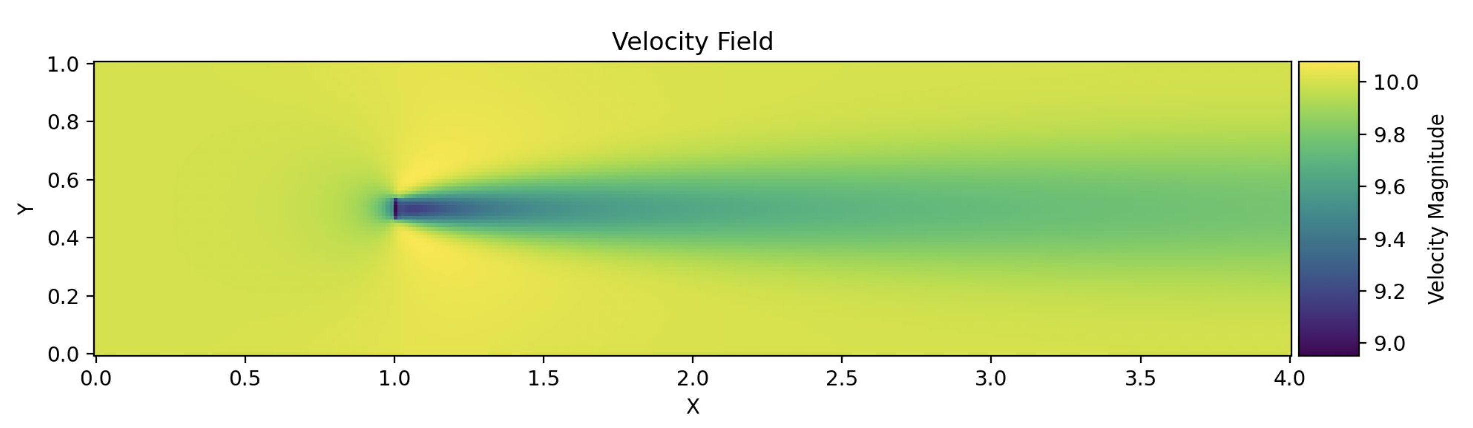


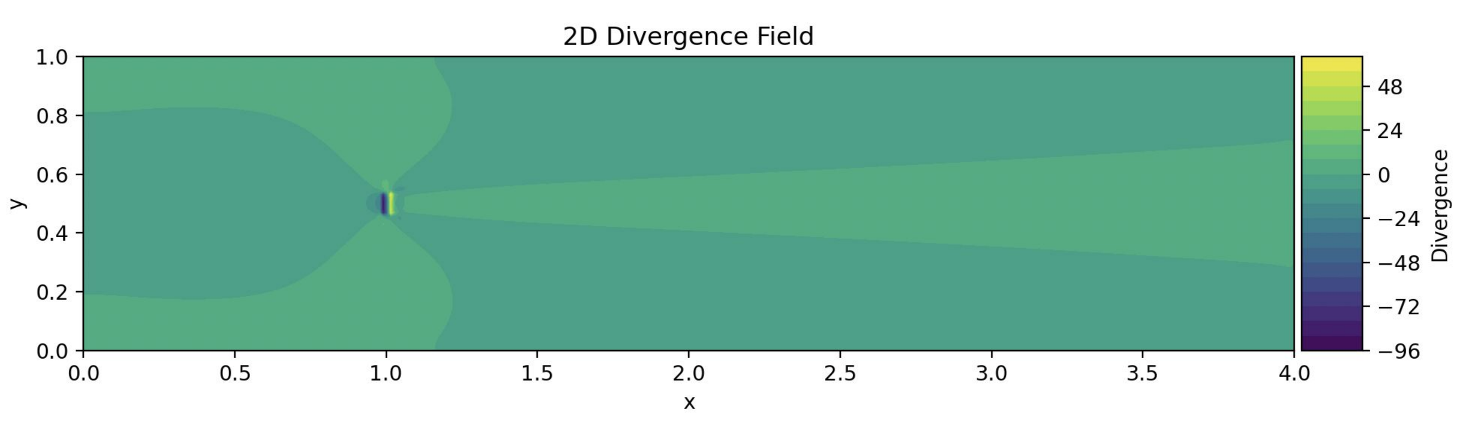


F turbine, no walls, New Poisson

nu=0.05
P iterations= 5
F= -1000
inVel= 10
80x240

BC's:
- **Velocity.** **Inlet: Dirichlet** **Outlet, Top, Bottom: Neumann**
- **Pressure.** **Inlet, Top, Bottom: Neumann** **Outlet: Dirichlet**





Adding Axial Induction term to the x-momentum eq.

momentum eq. with axial induction body force term

$$\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy} = \frac{-1}{\rho}\frac{dp}{dx} + \nu\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) - \frac{1}{2}\cdot\rho\cdot A\cdot C_t'\cdot \bar{u}^2$$

Discretizing

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{dt} + u_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{dx} + v_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{dy} = -\frac{1}{\rho}\frac{p_{i+1,j}^{n} - p_{i-1,j}^{n}}{2dx}$$

$$+ \nu\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{dx^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{dy^2}\right) - \frac{1}{2}\cdot\rho\cdot A\cdot C_t'\left(u_{i,j}^{n}\right)^2$$

And putting on explicit form

$$u_{i,j}^{n+1} = u_{i,j}^{n} - u_{i,j}^{n}\frac{dt}{dx}\left(u_{i,j}^{n} - u_{i-1,j}^{n}\right) - v_{i,j}^{n}\frac{dt}{dy}\left(u_{i,j}^{n} - u_{i,j-1}^{n}\right) - \frac{dt}{\rho 2dx}\left(p_{i+1,j}^{n} - p_{i-1,j}^{n}\right)$$

$$+ \nu\left(\frac{dt}{dx^2}\left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right) + \frac{dt}{dy^2}\left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right)\right)$$

$$- dt\left(\frac{1}{2}\rho A C_t'\left(u_{i,j}^{n}\right)^2\right)$$

IMPLEMENTATION:

```
if AI == True:
    BODY_FORCE= - self.dt*0.5*self.rho* A * Ct * un[1:-1, 1:-1] *un[1:-1, 1:-1]
```

Deriving the Poisson eq from the axial induction momentum eqs

New NS Equations:

I'll start with just horizontal axial induction.

$$\frac{du}{dx} + \frac{dv}{dy} = 0$$

$$\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy} = -\frac{1}{\rho}\frac{dp}{dx} + \nu\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) - \frac{1}{2}\cdot\rho\cdot A_n\cdot C_t\cdot \bar{u}^2$$

$$\frac{dv}{dt} + u\frac{dv}{dx} + v\frac{dv}{dy} = -\frac{1}{\rho}\frac{dp}{dy} + \nu\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)$$

Take sum of divergences

1) $$\frac{d}{dt}\left(\frac{du}{dx}\right) + \frac{du}{dx}\frac{du}{dx} + u\frac{d^2u}{dx^2} + \frac{dv}{dx}\frac{du}{dy} + v\frac{d^2u}{dxdy} = -\frac{1}{\rho}\frac{d^2p}{dx^2} + \nu\frac{d}{dx}(\nabla^2 u) - \rho A C_t\, u\frac{du}{dx}$$

\+

2) $$\frac{d}{dt}\left(\frac{dv}{dy}\right) + \frac{du}{dy}\frac{dv}{dx} + u\frac{d^2v}{dydx} + \frac{dv}{dy}\frac{dv}{dy} + v\frac{d^2v}{dy^2} = -\frac{1}{\rho}\frac{d^2p}{dy^2} + \nu\frac{d}{dy}(\nabla^2 v)$$

* Divergence of F.

$$\nabla F = \frac{F_x}{dx} + \cancel{\frac{F_y}{dy}} = \frac{F_x}{dx} = \frac{d}{dx}\left(\frac{1}{2}\rho A C_t \bar{u}^2\right) = \frac{1}{2}\rho A C_t\frac{d}{dx}u^2 = \frac{1}{2}\rho A C_t\, 2u\frac{du}{dx} = \rho A C_t\, u\frac{du}{dx}$$

$$\frac{d}{dt}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0} + \left(\frac{du}{dx}\right)^2 + \frac{du}{dy}\frac{dv}{dx} + \underbrace{u\frac{d^2u}{dx^2} + u\frac{d^2v}{dydx}}_{u\frac{d}{dx}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0}} + \frac{dv}{dx}\frac{du}{dy} + \left(\frac{dv}{dy}\right)^2 + \underbrace{v\frac{d^2u}{dxdy} + v\frac{d^2v}{dy^2}}_{v\frac{d}{dy}\underbrace{\left(\frac{du}{dx}+\frac{dv}{dy}\right)}_{=0}} = -\frac{1}{\rho}\left(\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2}\right) + \nu\left[\frac{d}{dx}\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) + \frac{d}{dy}\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)\right] - \rho A C_t\, u\frac{du}{dx}$$

Apply continuity $\left(\frac{du}{dx} + \frac{dv}{dy} = 0\right)$

$$\frac{d^3u}{dx^3} + \frac{d^3u}{dxdy} + \frac{d^3v}{dydx} + \frac{d^3v}{dy^2}$$

$$\frac{d^2}{dx^2}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0} + \frac{d^2}{dy^2}\underbrace{\left(\frac{du}{dx} + \frac{dv}{dy}\right)}_{=0}$$

$$\left(\frac{du}{dx}\right)^2 + 2\left(\frac{du}{dy}\frac{dv}{dx}\right) + \left(\frac{dv}{dy}\right)^2$$

$$\boxed{-\frac{1}{\rho}\left(\frac{d^2p}{dx^2} + \frac{d^2p}{dy^2}\right) = \left(\frac{du}{dx}\right)^2 + 2\frac{du}{dy}\frac{dv}{dx} + \left(\frac{dv}{dy}\right)^2 + \rho A C_t\, u\frac{du}{dx}}$$

Now, let's discretize it

$$\frac{p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}}{\Delta x^2} + \frac{p^n_{i,j-1} - 2p^n_{i,j} + p^n_{i,j-1}}{\Delta y^2} = \rho\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\right)^2 - 2\left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\right)\left(\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x}\right) - \left(\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right)^2 + \rho A C_t\cdot u^n_{i,j}\frac{u^n_{i,j} - u^n_{i-1,j}}{\Delta x}\right]$$

Backward Difference

And re-arrange into an explicit form

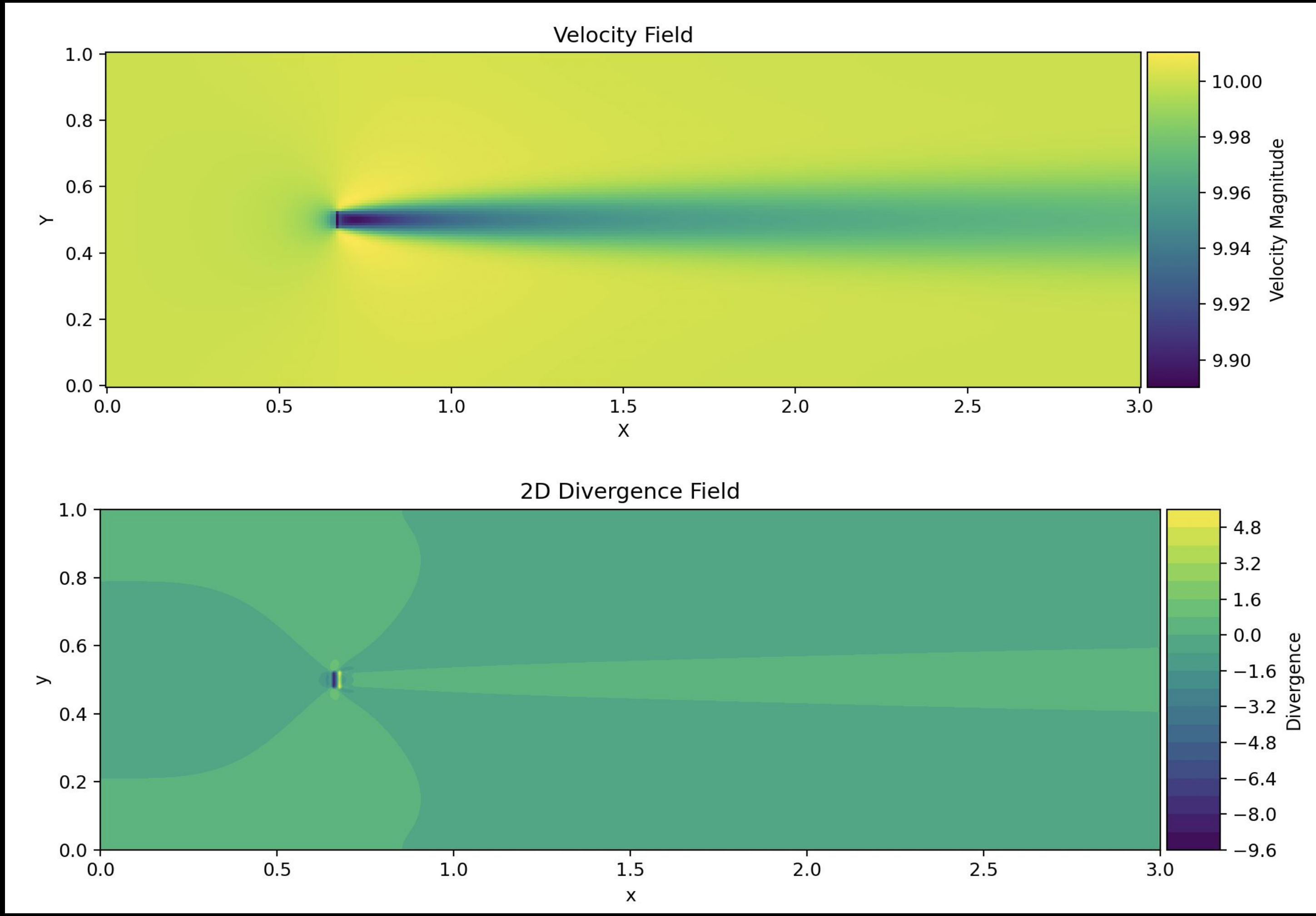
$$p^n_{i,j} = \frac{(p^n_{i+1,j} + p^n_{i-1,j})\Delta y^2 + (p^n_{i,j+1} + p^n_{i,j-1})\Delta x^2}{2(\Delta x^2 + \Delta y^2)}$$

$$- \frac{\rho\Delta x^2\Delta y^2}{2(\Delta x^2 + \Delta y^2)}\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} - 2\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} - \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} + \rho A C_t\, u^n_{i,j}\frac{u^n_{i,j} - u^n_{i-1,j}}{\Delta x}\right]$$

Now, it is time to implement it!

```python
def build_b_AI(self, rho, dt, u, v, dx, dy, A):
    Ct= 3/4
    b = np.zeros_like(self.u)
    b[1:-1, 1:-1] = (1 / dt *
                    ((u[1:-1, 2:] - u[1:-1, 0:-2]) /
                    (2 * dx) + (v[2:, 1:-1] - v[0:-2, 1:-1]) / (2 * dy)) -
                    ((u[1:-1, 2:] - u[1:-1, 0:-2]) / (2 * dx))**2 -
                    2 * ((u[2:, 1:-1] - u[0:-2, 1:-1]) / (2 * dy) *
                        (v[1:-1, 2:] - v[1:-1, 0:-2]) / (2 * dx))-
                        ((v[2:, 1:-1] - v[0:-2, 1:-1]) / (2 * dy))**2
                        + self.AI[1:-1, 1:-1]*rho*A*Ct*u[1:-1, 1:-1]*(u[1:-1, 1:-1]-u[1:-1, 0:-2])/dx)
    return b
```

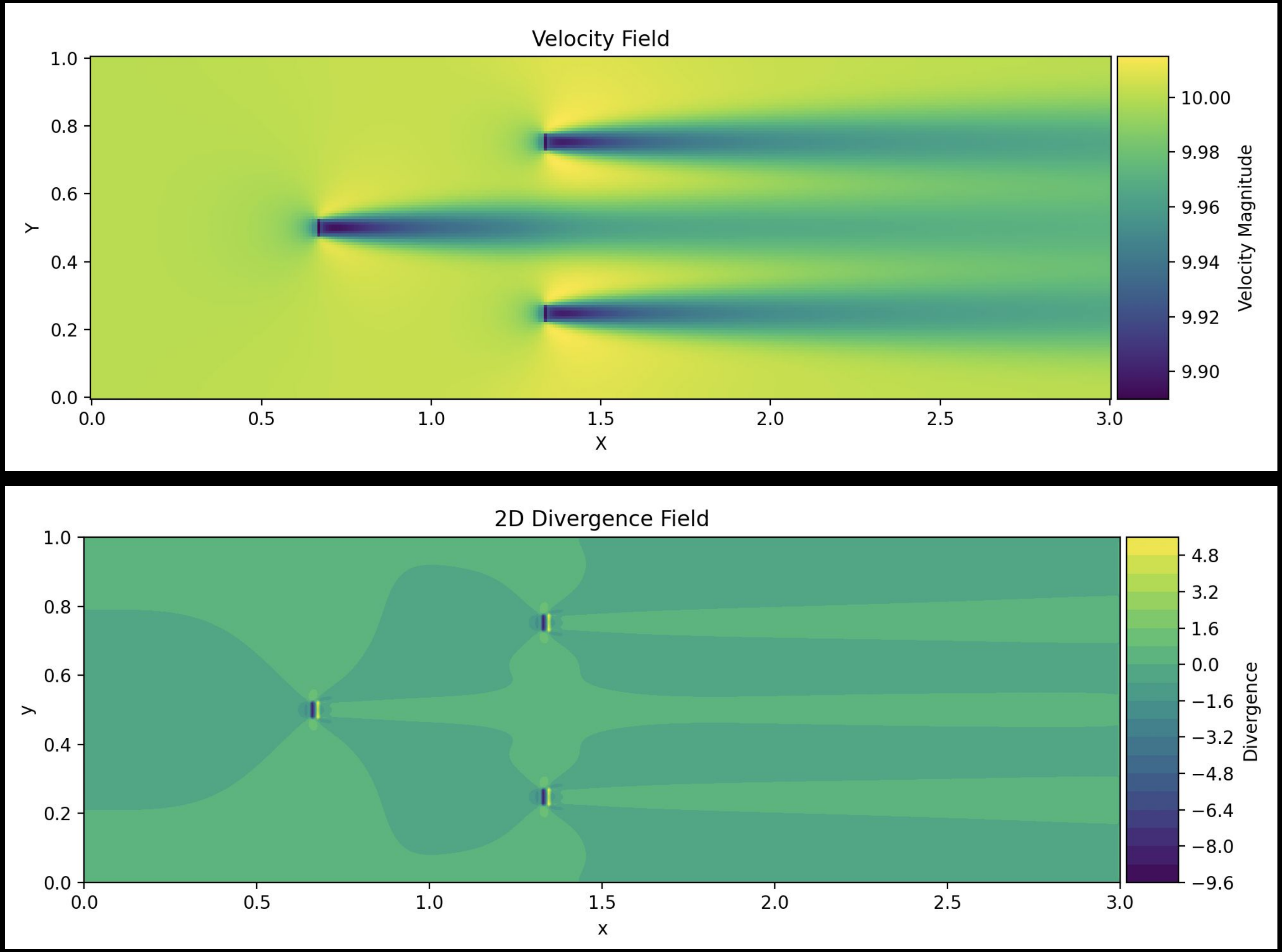
Single (axial induction) Turbine



Effect looks good but the magnitude of the force since to be off (too small).

This is the same that happened on the Javascript solver.

3 turbine wind farm (axial induction)



New Projection Method: Fractional Step

I went over a Guermond's Projection Method overview.

I will start with the simplest algorithm, proposed by Chorin.

The algorithm:

① $$\frac{du}{dt} + (u\cdot\nabla)u = \nu\nabla^2 u \qquad \begin{cases} \dfrac{du}{dt} + u\dfrac{du}{dx} + v\dfrac{du}{dy} = \nu\dfrac{d^2u}{dx^2} + \nu\dfrac{d^2u}{dy^2} \\ \dfrac{dv}{dt} + u\dfrac{dv}{dx} + v\dfrac{dv}{dy} = \nu\dfrac{d^2v}{dx^2} + \nu\dfrac{d^2v}{dy^2} \end{cases}$$

solve momentum eq

¡WITHOUT pressure TERM!

② $$\nabla^2 p = \frac{\rho}{\Delta t}\nabla\cdot u^* \qquad \left\{ \frac{d^2p}{dx^2} + \frac{d^2p}{dy^2} = \frac{\rho}{\Delta t}\left(\frac{du}{dx} + \frac{dv}{dy}\right) \right.$$

solve poisson eq.

③ $$U = U - \frac{\Delta t}{\rho}\nabla p \qquad \begin{cases} u = u^* - \dfrac{\Delta t}{\rho}\dfrac{dp}{dx} \\ v = v - \dfrac{\Delta t}{\rho}\dfrac{dp}{dy} \end{cases}$$

CORRECT VELOCITIES.

ELSEVIER
Available online at www.sciencedirect.com
SCIENCE @ DIRECT®
Comput. Methods Appl. Mech. Engrg. 195 (2006) 6011–6045
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

An overview of projection methods for incompressible flows

J.L. Guermond [a,b,*,1], P. Minev [c,2], Jie Shen [d,3]

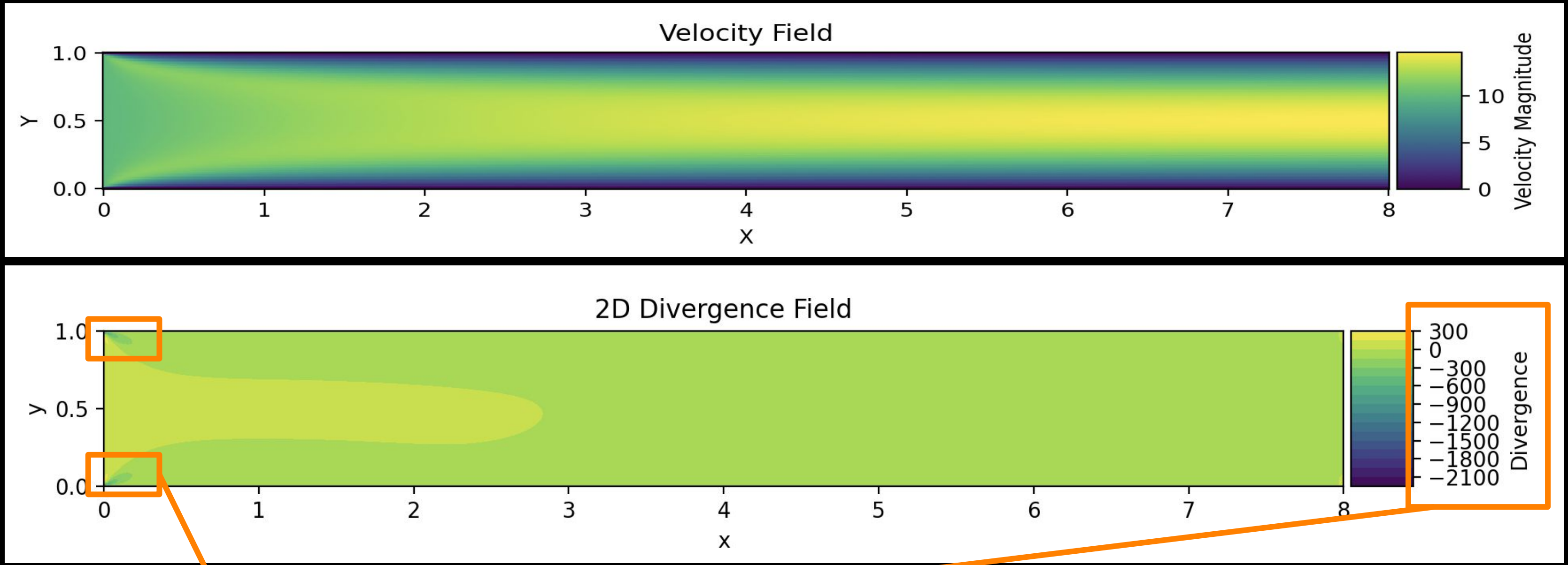
[a] LIMSI (CNRS-UPR 3251), BP 133, 91403 Orsay, France
[b] Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA
[c] Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Canada T6G 2G1
[d] Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA

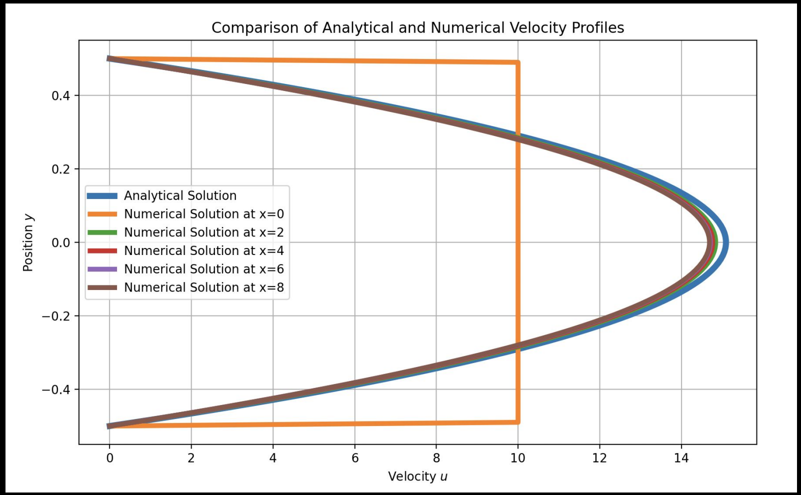
Received 10 June 2004; received in revised form 22 August 2005; accepted 21 October 2005

First solutions (pipe with walls)

BC's:
- **Velocity.** **Inlet, Top, Bottom: Dirichlet** **Outlet: Neumann**
- **Pressure.** **Inlet, Top, Bottom: Neumann** **Outlet: Dirichlet**

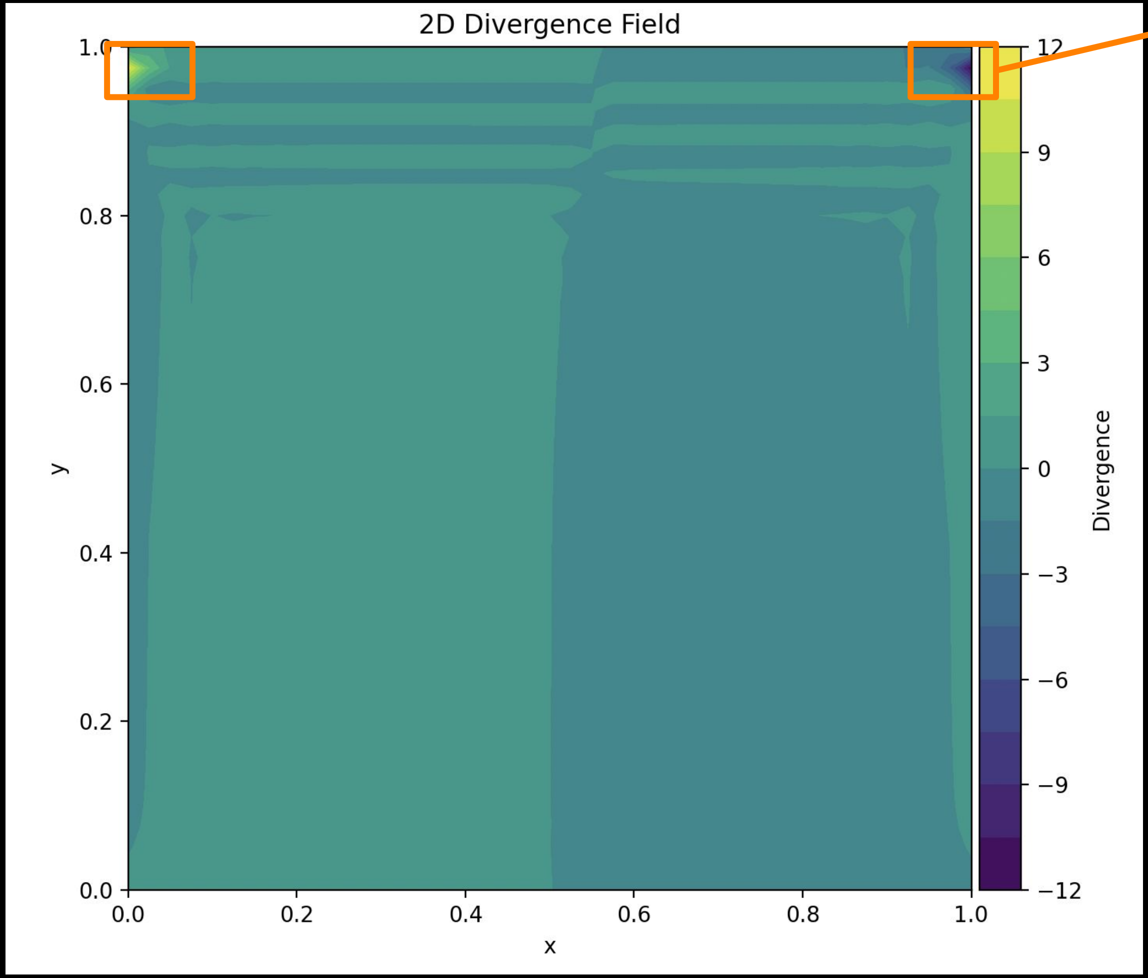
nu=0.05
P iterations= 5
inVel= 10
80x640
steady state



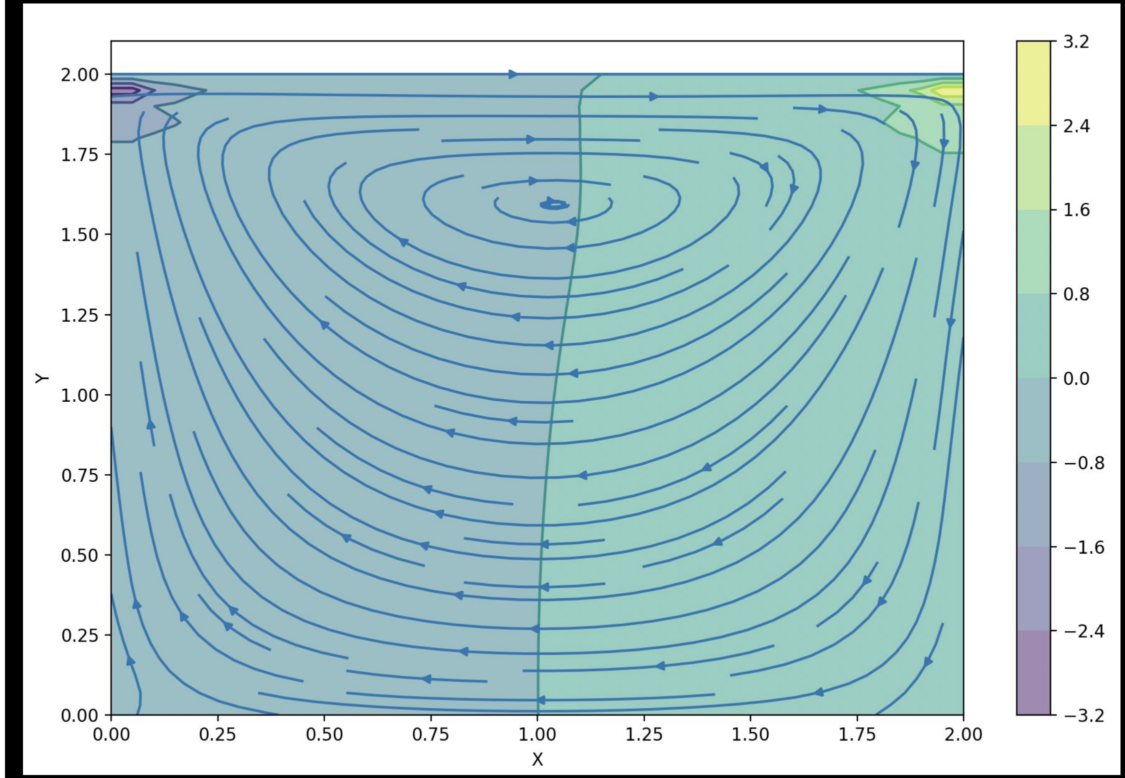


Big problem; Divergence is too high at the entrance close to the wall !!!!!

Why I can’t get rid of divergence? Let’s check with Cavity flow.



Lorena’s simulation contains divergence too!



Let’s fix the divergence issue.

Potential Issues:

Inlet Boundary Condition (Velocity and Pressure):

- I was specifying a Dirichlet boundary condition for the velocity at the inlet. However, I’m using a zero-gradient (Neumann) boundary condition for pressure at the inlet. This combination can lead to inconsistency because the pressure gradient at the inlet should be compatible with the imposed velocity profile, to satisfy the momentum equation. When the pressure gradient is not correctly specified, it can result in incorrect acceleration, leading to high divergence. dp/dx=0 won't work, for sure

Suggested Boundary Conditions:

B)

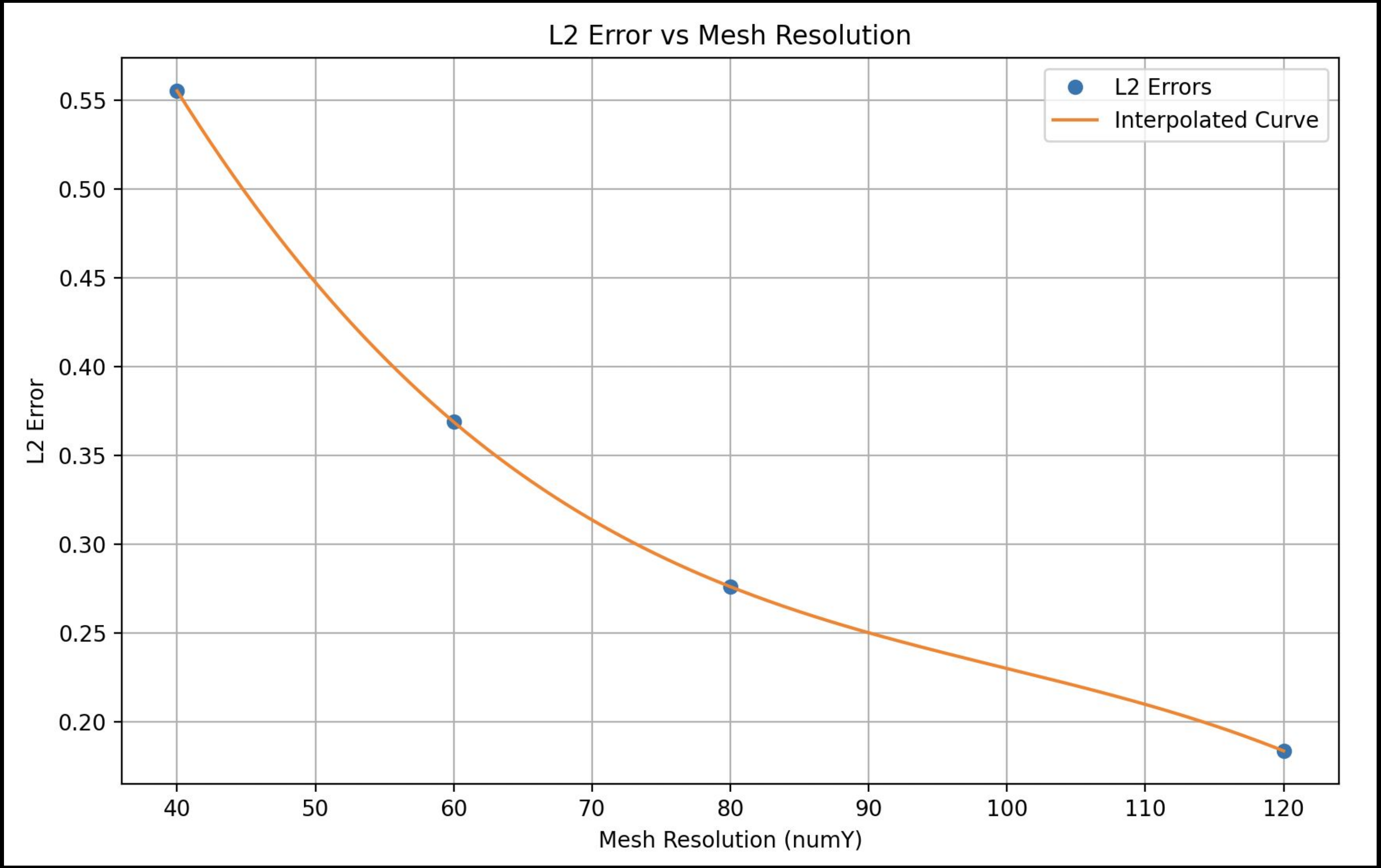
- **Velocity:** Keep the Dirichlet boundary condition for velocity.
- **Pressure:** Instead of using a zero-gradient condition, you might Keep the Dirichlet condition for pressure at the outlet, but ensure that the specified pressure is consistent with the expected flow profile. Fixed pressure gradient or a pressure value at the inlet.

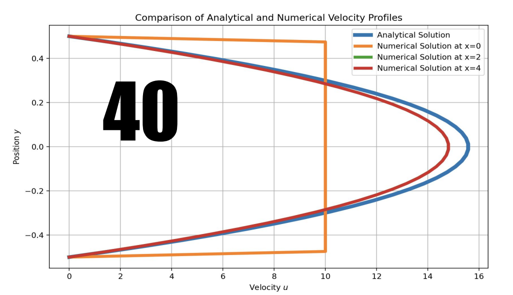
A) Proposed by Peter and Colin

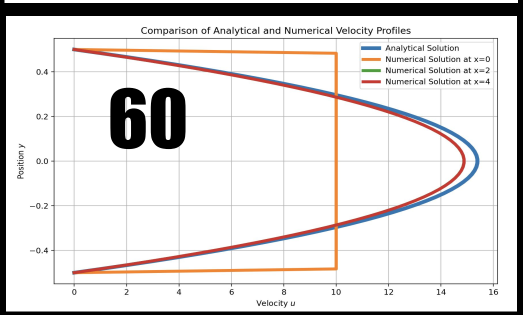
Velocity: Let the velocity at the inlet change. That is, using a Neumann BC?

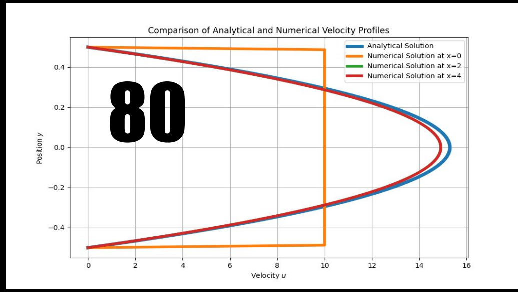
Pressure: Inlet pressure defined by the Poiseuille equation. I don’t understand how to do this..

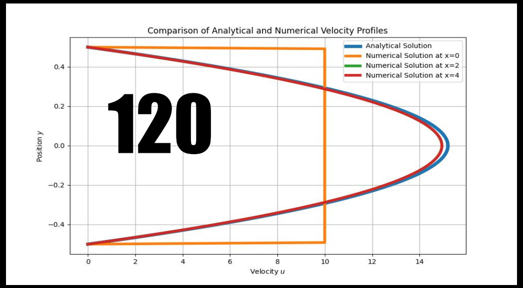
Mesh Refinement: L2 error.



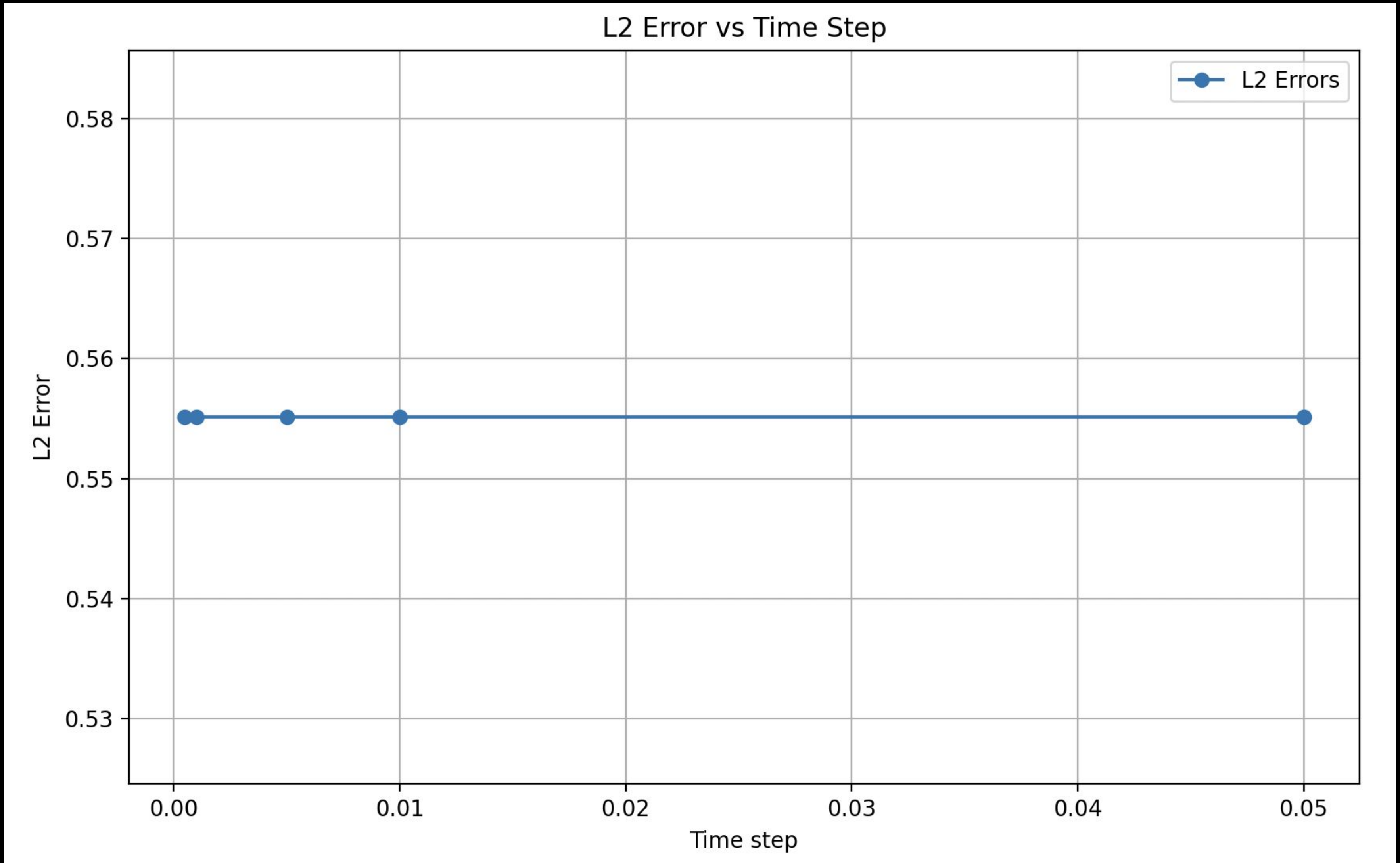








Time Step Refinement

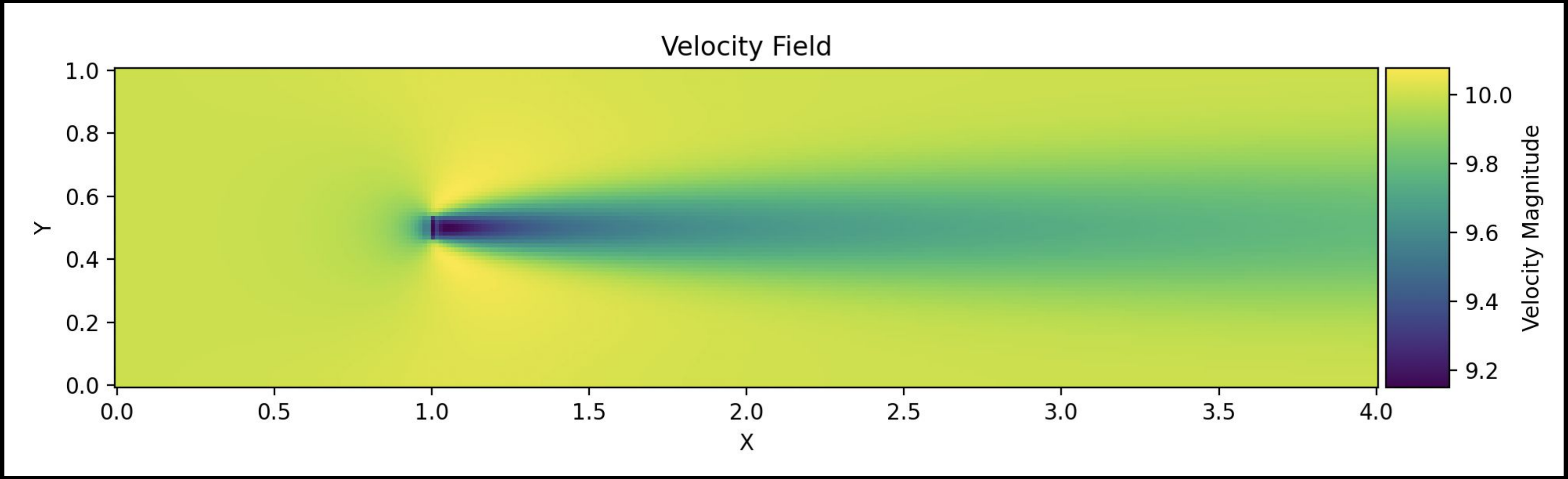


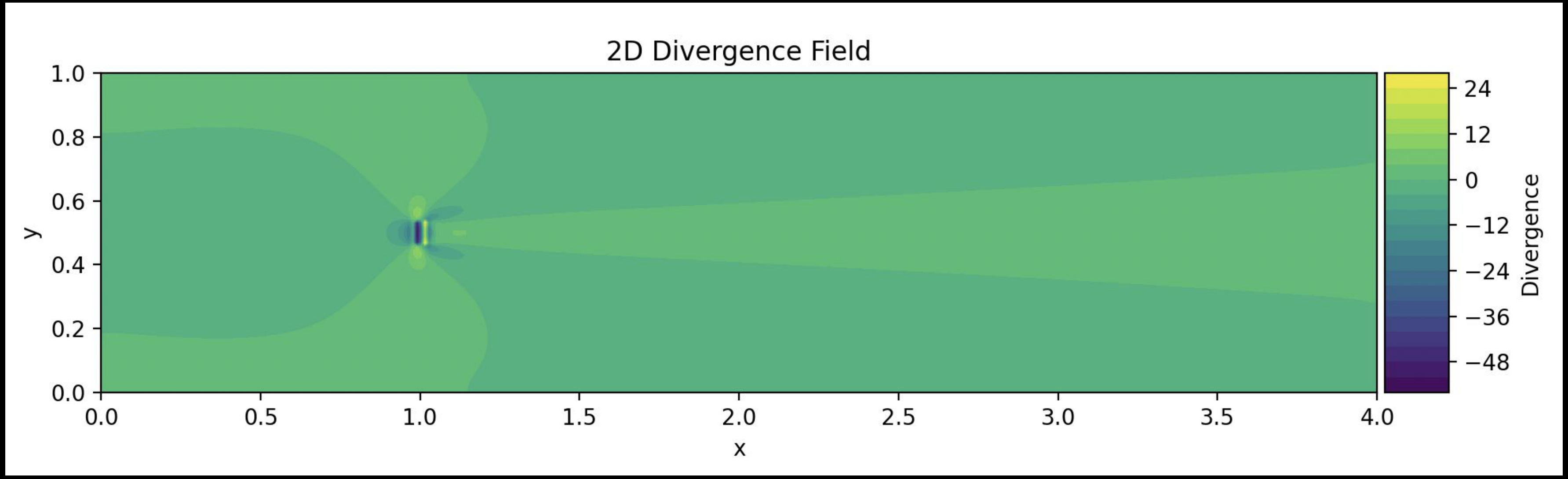
The simulation does not improve **at all** when reducing the time step.

I guess the larger timestep is simply sufficient.

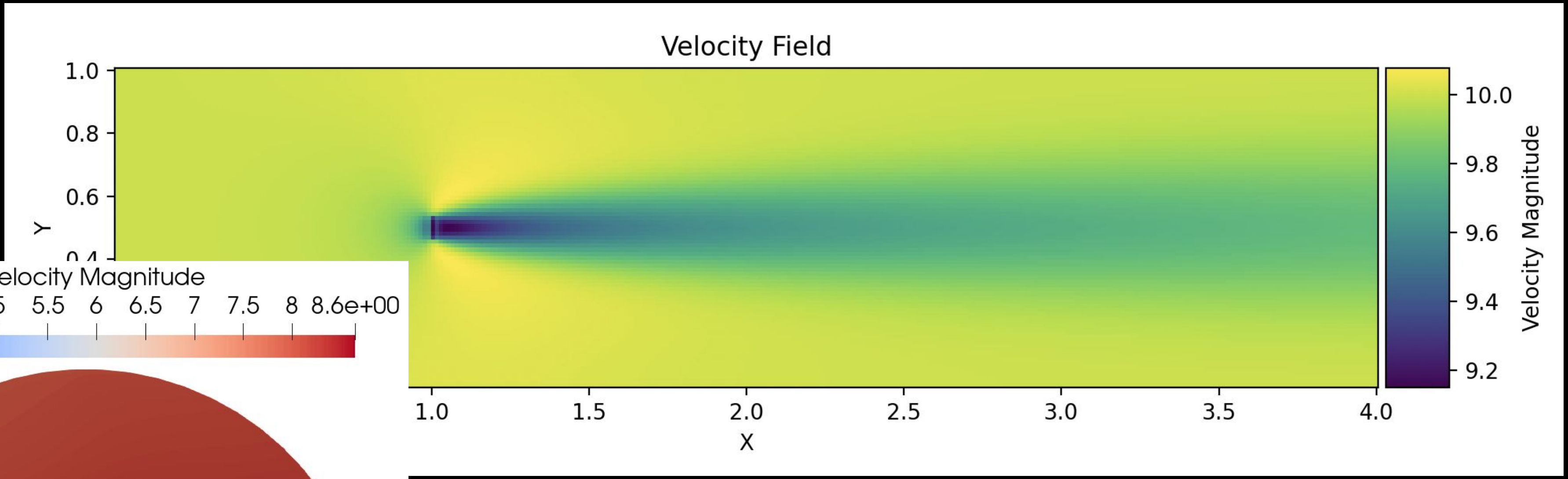
Colin:
Being laminar and steady state, this could be ok.

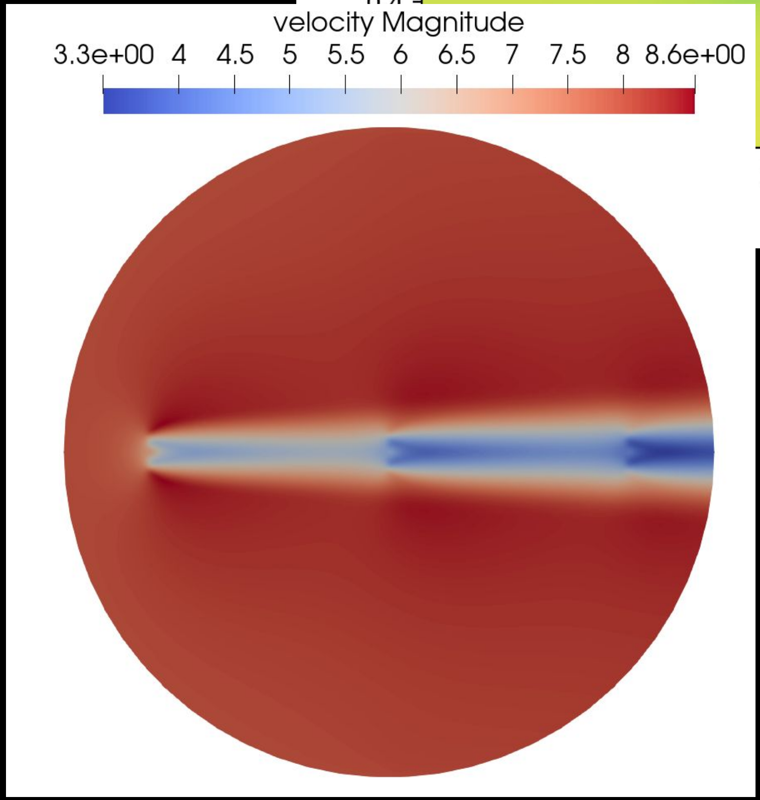
Let’s remove walls and add a turbine…

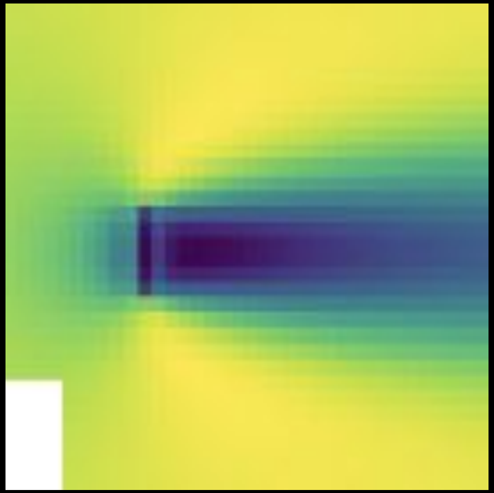


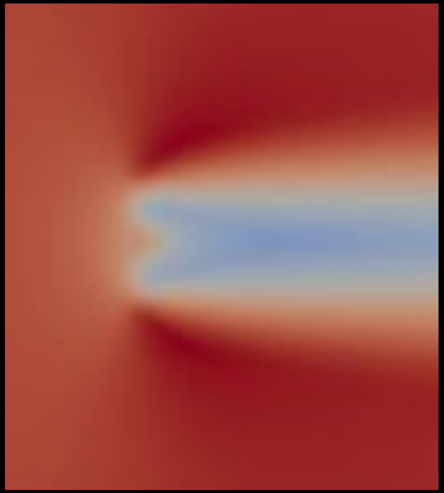


Comparison with WindSE - Velocity field

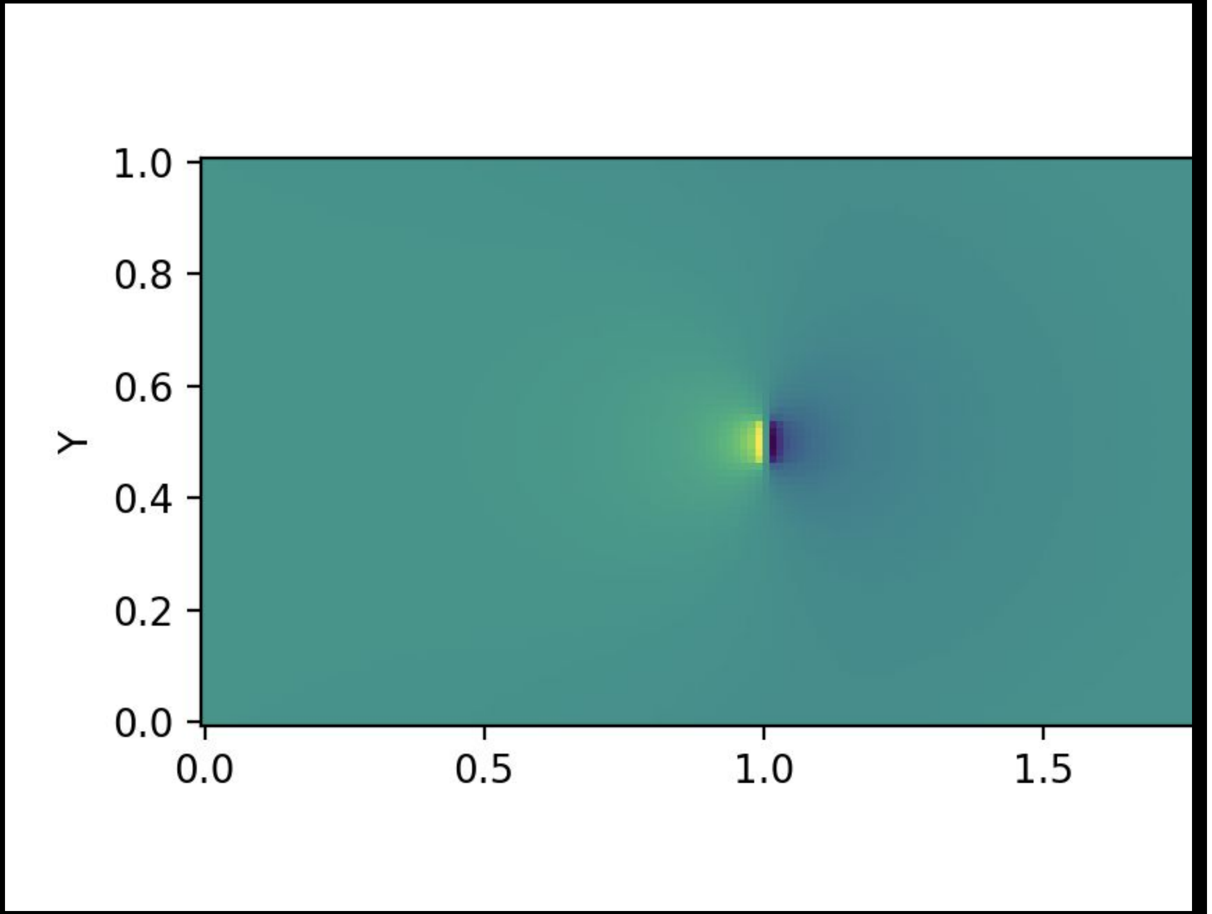


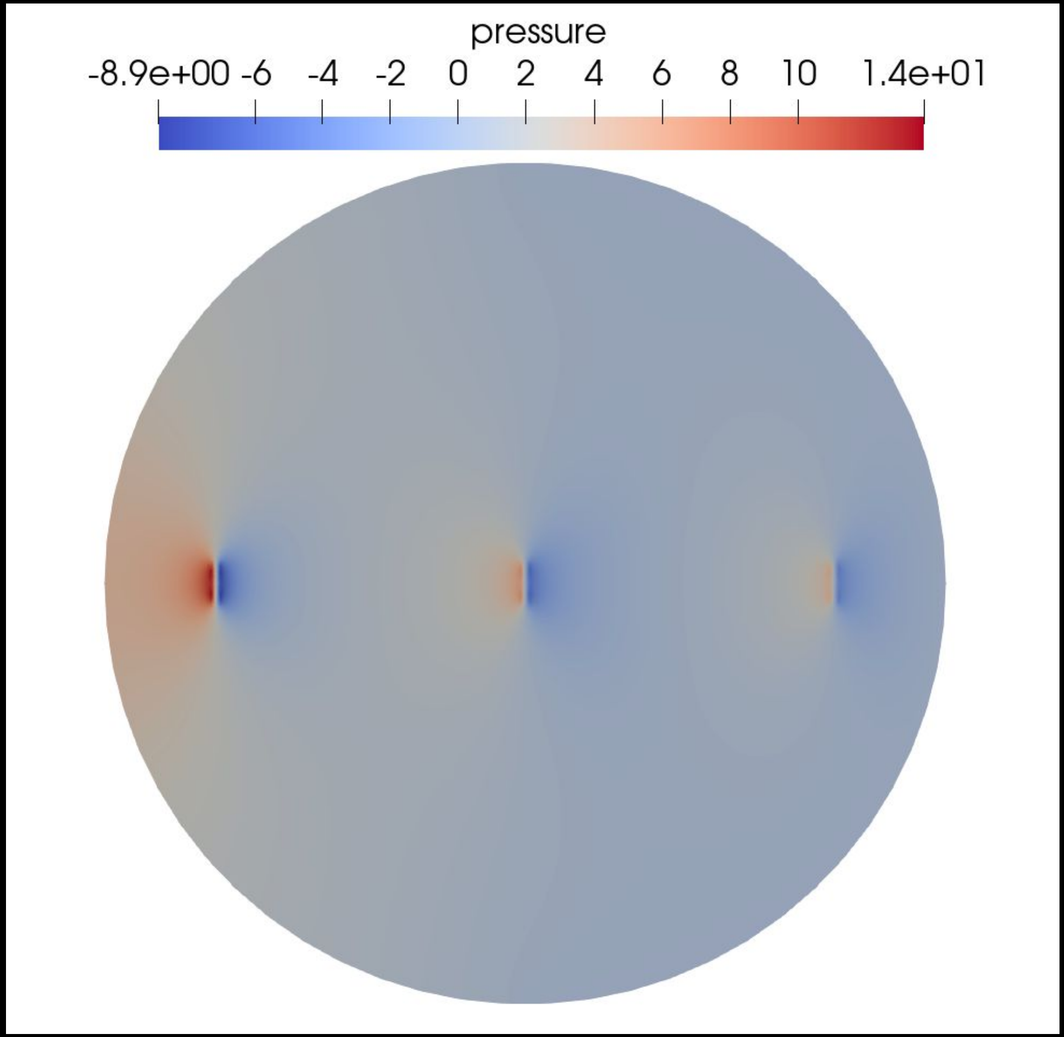




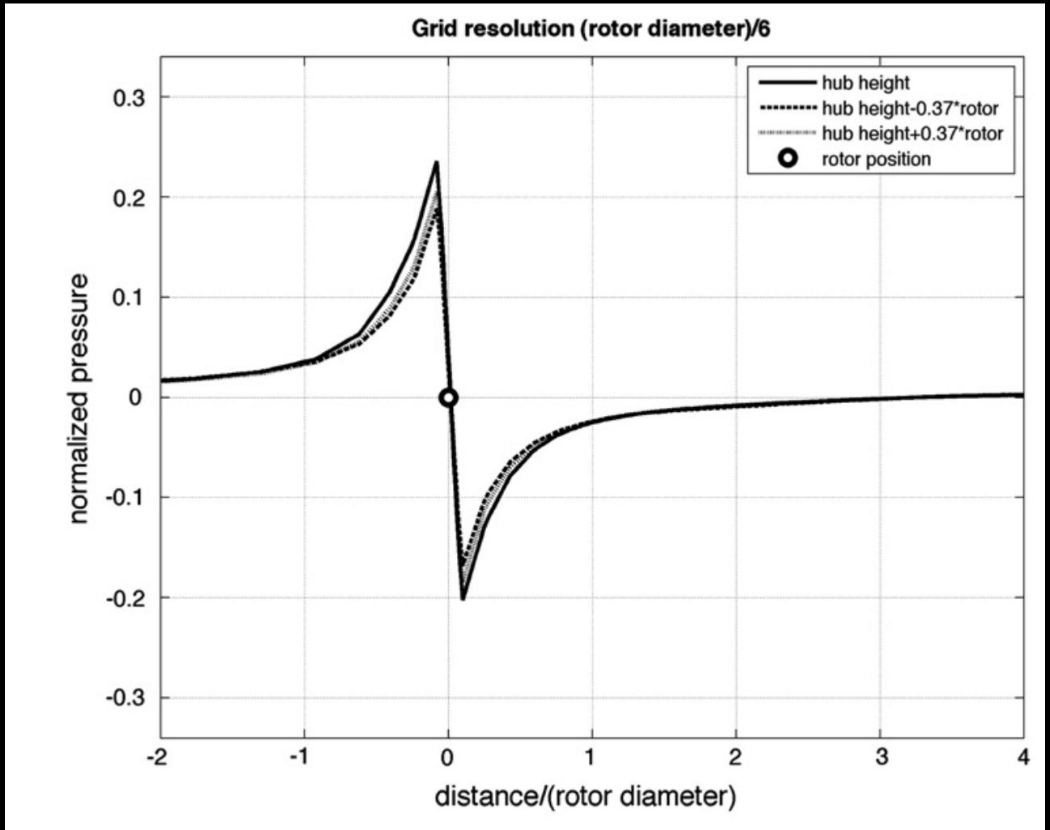


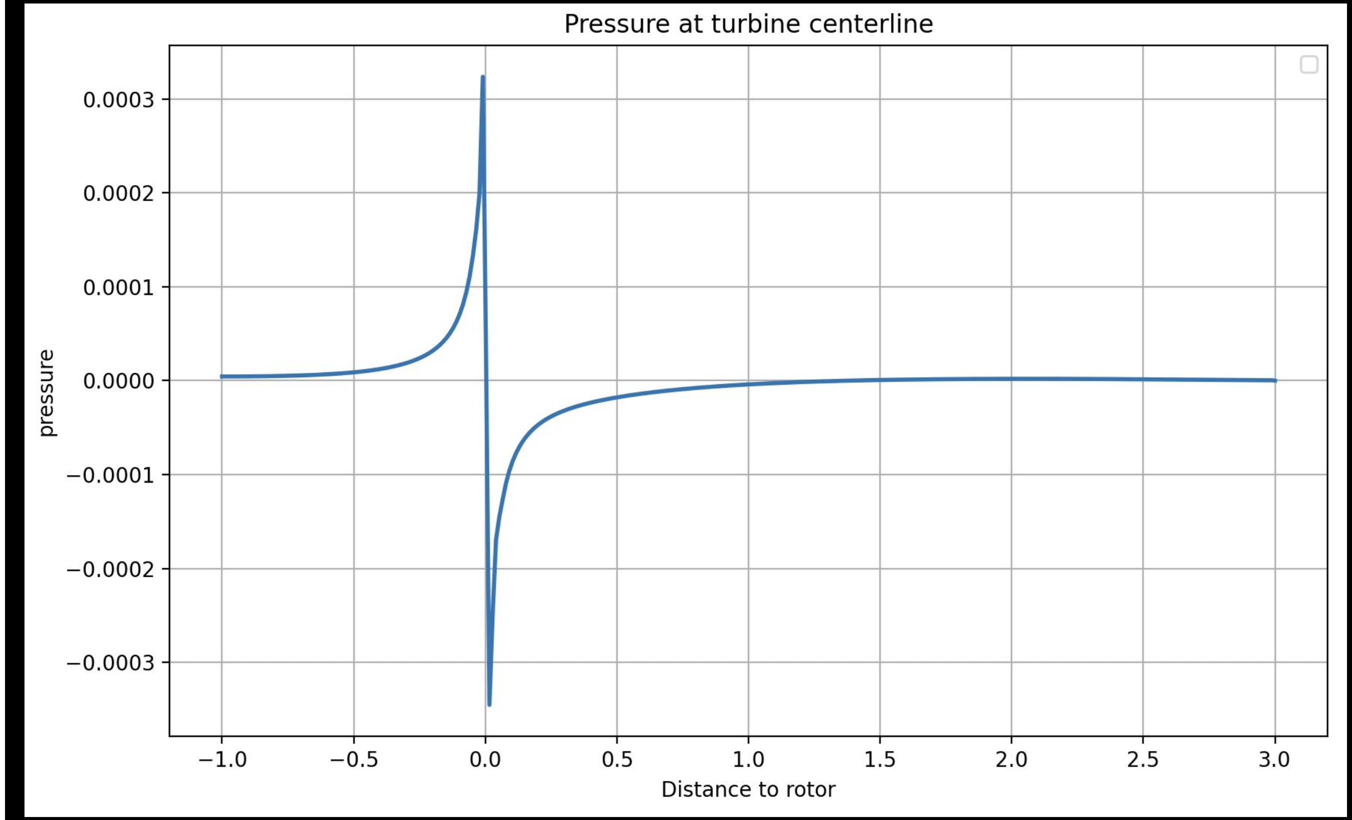
Comparison with WindSE - Pressure field





Comparison with Castellani’s paper - Pressure around rotor





Normalize over the rotor diameter.

Turbine wake modelling

Analytical	Fast and simple calculations	Wake losses calculated downstream the wind field
Actuator disc	Average calculation effort Contextual wind field and wakes simulations Coarse mesh	Effects of rotation cannot be simulated Extremely simplified geometry

Would an analytical solution be sufficient for my purpose? Probably not because I want to capture the effect on the upwind flow.

Actuator disc theory.

The actuator disc model disregards fundamental characteristics of wind turbine aerodynamics because it represents any rotor by a distributed action over the flow rather than the complex, vortical structure that a real rotor creates in the flow. Despite its simplicity, it is often a useful representation to reproduce wind deficits and wake losses.

Thrust coefficient. Depends on the thrust coefficient curve of the aerogenerator.

The streamlines passing through the disc are affected by the power extraction and a wind velocity deficit so that the wind speed on the rotor v_r is lower than the undisturbed speed v_∞. The theory represents this mechanism by introducing the axial induction factor:

$$v_r = (1 - a) * v_\infty \qquad (1)$$

Actuator disc theory

The actuator disc model disregards fundamental characteristics of wind turbine aerodynamics because it represents any rotor by a distributed action over the flow rather than the complex, vortical structure that a real rotor creates in the flow. Despite its simplicity, it is often a useful representation to reproduce wind deficits and wake losses.

Thrust coefficient. Depends on the thrust coefficient curve of the aerogenerator.

The streamlines passing through the disc are affected by the power extraction and a wind velocity deficit so that the wind speed on the rotor v_r is lower than the undisturbed speed v_∞. The theory represents this mechanism by introducing the axial induction factor:

$$v_r = (1 - a) * v_\infty \qquad (1)$$

An alternative extremely simplistic approach, the one that I initially proposed, consists on simply updating the velocity at the rotor using the axial induction factor.

The thrust coefficient for the whole rotor C_T is defined as:

$$C_T = \frac{S_a}{\frac{1}{2}\rho \cdot v_\infty^2 \cdot A}. \qquad (2)$$

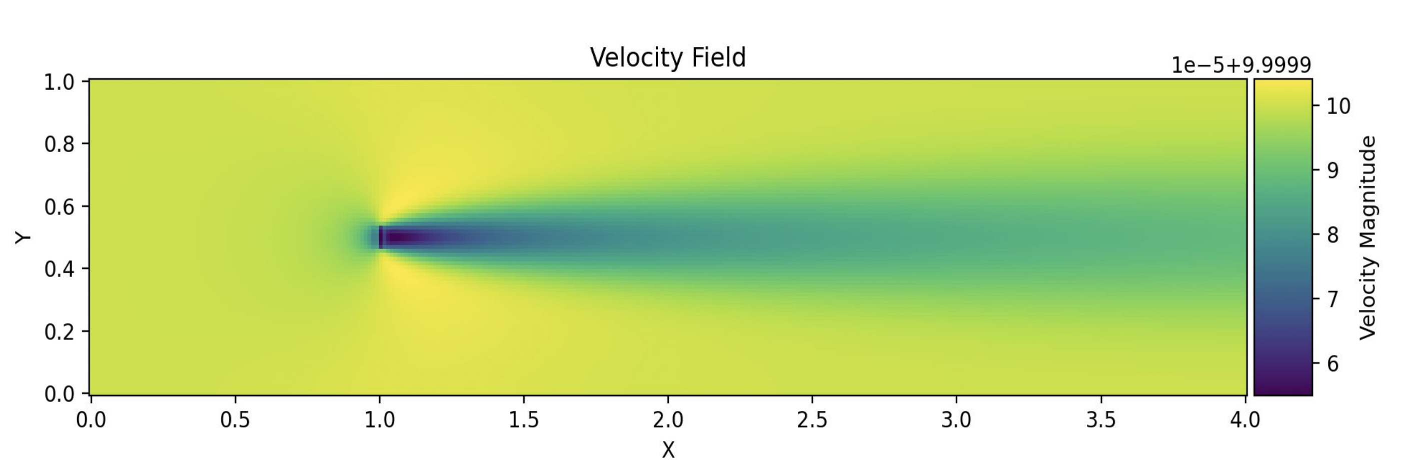
where S_a is the thrust force, ρ is the air density, v_∞ is the free stream velocity at the hub height and A is the rotor swept area.

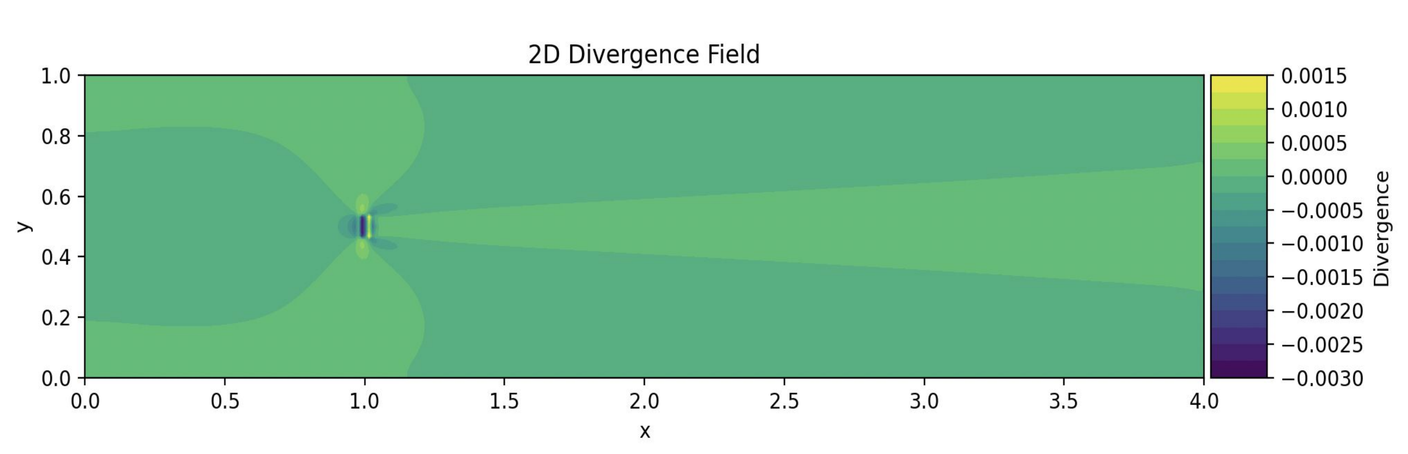
In this way it is possible to estimate the force that has to be introduced against the wind on the rotor surface (that is modelled as a group of cells on which the kinetic energy is extracted) for each different wind flow condition.

I have been using the velocity at the rotor location (not the unaffected one) to compute the axial induction force.
This could be the reason why the effect seems to be lighter than expected!!!!!!!
Let’s try to simply use the inlet velocity to compute a constant force based on actuator disk theory!!

Axial Induction from Actuator Disk Theory

For the first time, I am computing the axial induction force as a constant force based on the inlet velocity.





Divergence is so small!!! Good new but… Why?

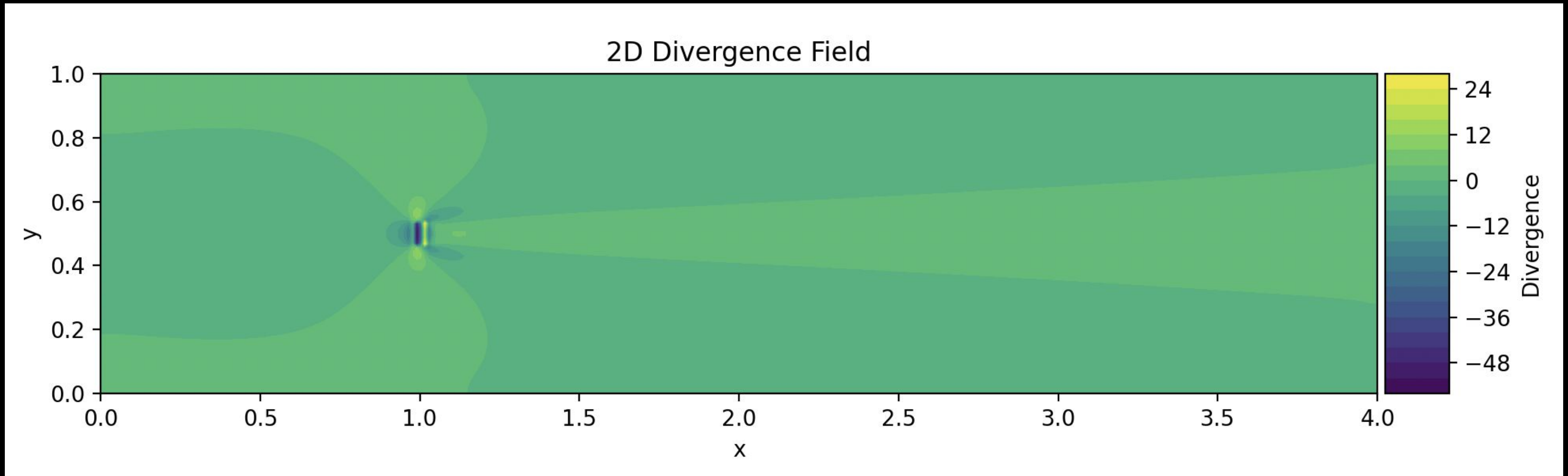
I got a realistic force magnitude for the first time!!! How do I know this? Let's use the axial induction factor a=⅓ (based on Betz limit). Then, the velocity at the rotor should be around 6.66. It's closer but now accurate. Think more about this..

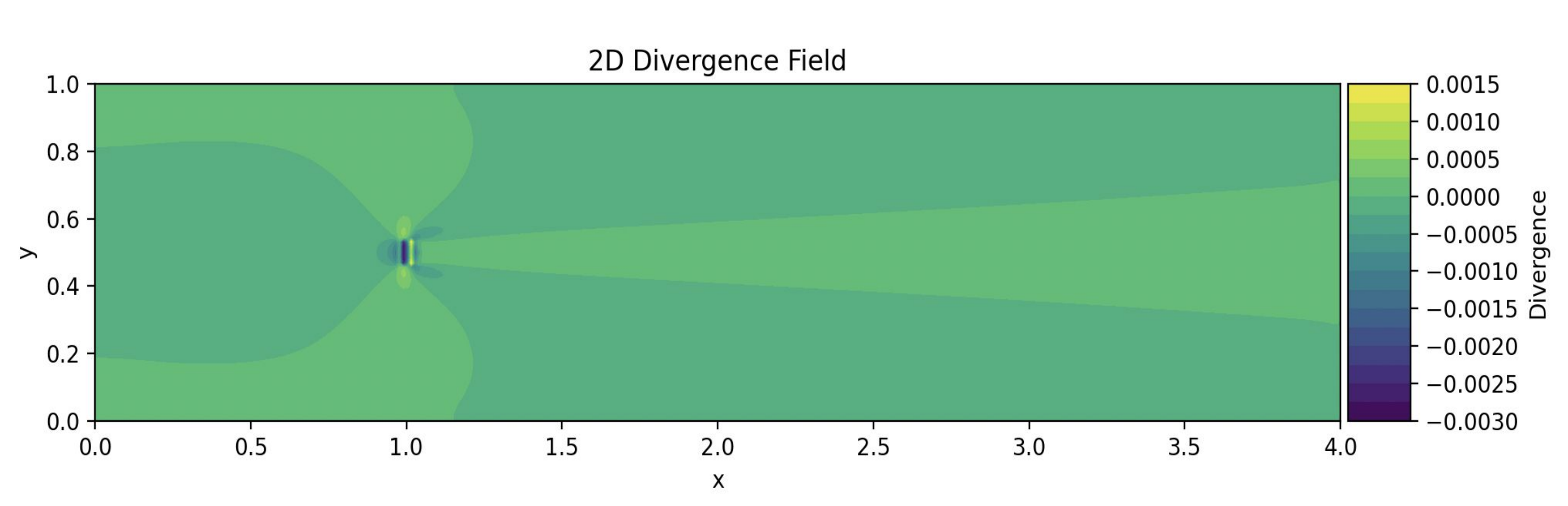
Maybe I am not using the proper Ct?
Go back to Peter notes about actuator disc theory.

The next question is.. using this technique.. which velocity am I going to use for the downstream turbine's AI force???

I need to implement a new method that takes the mean velocity from cells that are upstream of each of the turbines!!

Constant Force vs Actuator Disk Theory - Divergence






```
Adding F turbine...
F force=  -1000
```

The force I was previously using has such a large magnitude that it creates a huge divergence..

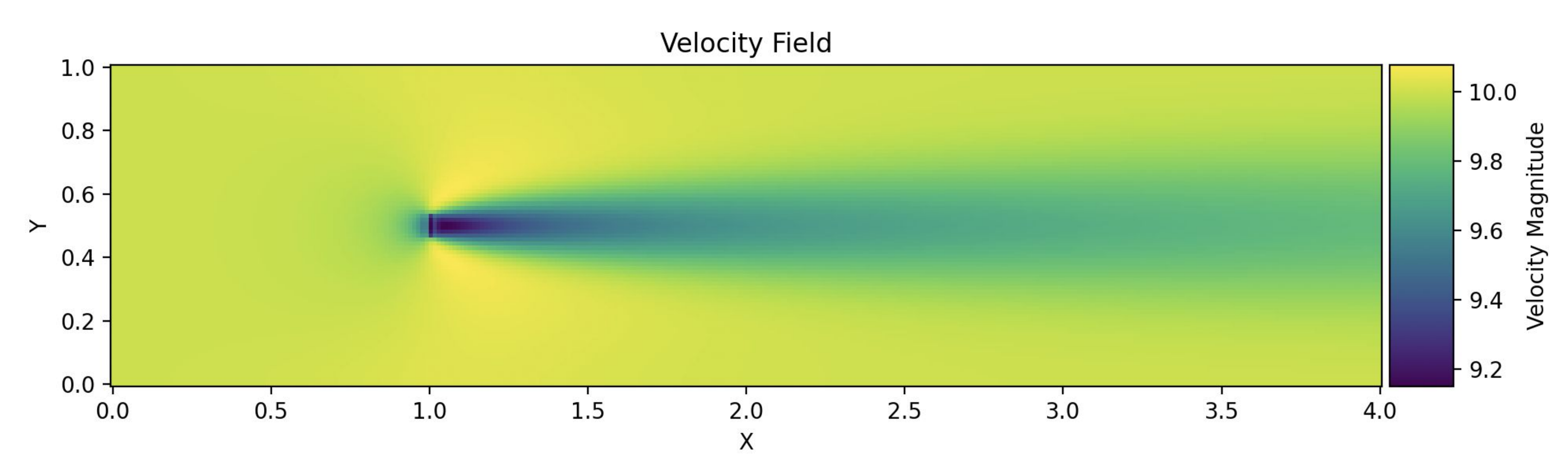
The axial induction force (from actuator disk theory) is very small and the diveregence created is negligible.

Let's see how does this affect the velocities at the wake.

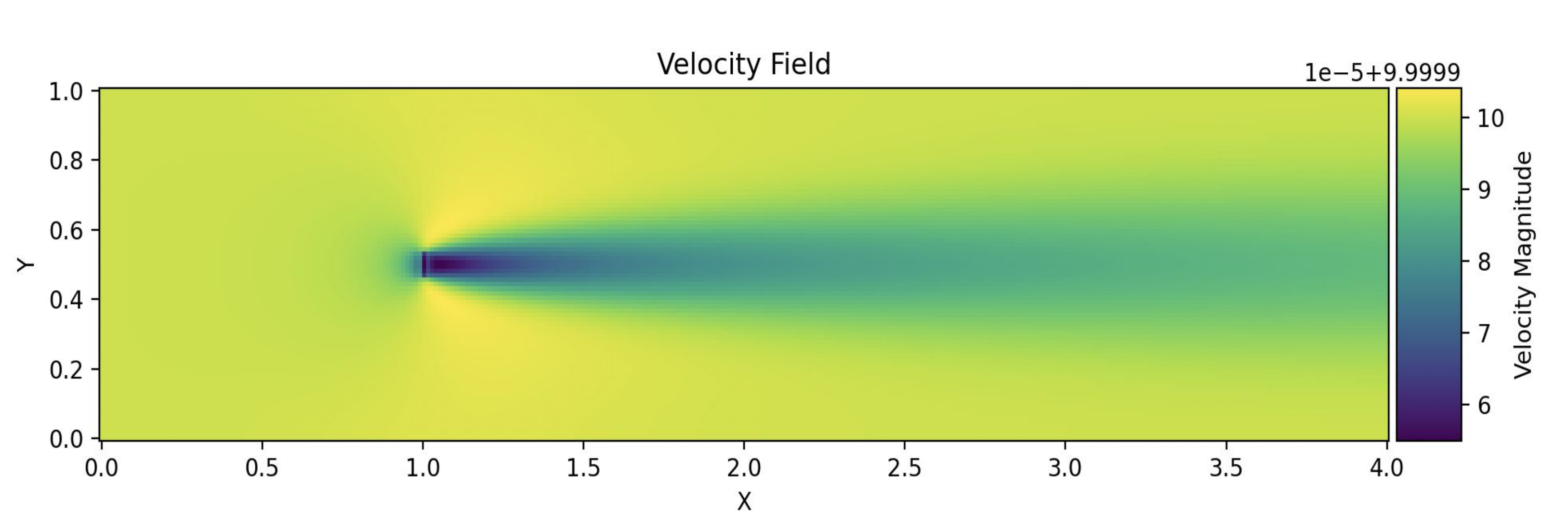
```
Adding AI turbine...
AI force=  -0.05887500
```

Constant Force vs Actuator Disk Theory - Velocities

F



AI



Surprisingly, the effect on the wake is stronger on the AI case.

The velocity at the wake is pretty much realistic, based on the Betz limit theory:

ADD CALCULATIONS HERE

Is it acceptable to have some divergence close to the edges???

For incompressible flow, the continuity equation must be satisfied everywhere. If your solution shows divergence near the edges of the turbine, it implies that the incompressibility condition is not being fully met in those regions. Why?

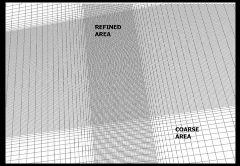
Numerical errors, grid resolution issues.

The edges of the actuator disk might have sharp gradients that are difficult to resolve accurately.

Acceptable? This could result in non-physical flow behavior and impact the accuracy of your simulation…

What can I do:

Refine mesh close to the turbines.



Re-evaluate how I am applying the axial induction force. Replace uniform by gaussian?

Re-evaluate pressure Poisson solver

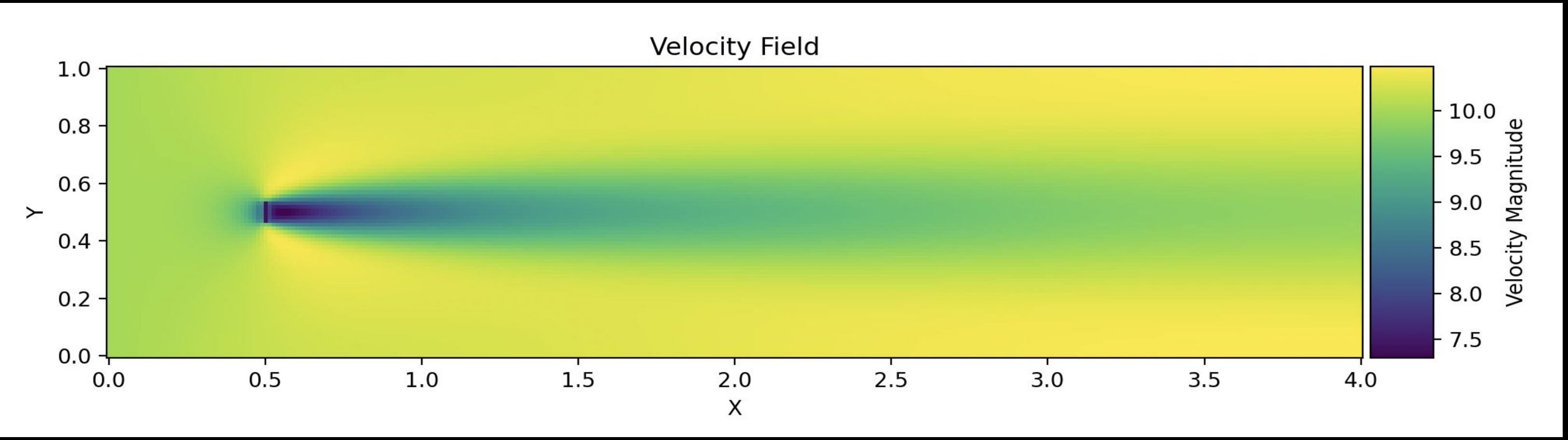
But the divergence is not really at the edges but at the front and back of the rotor.

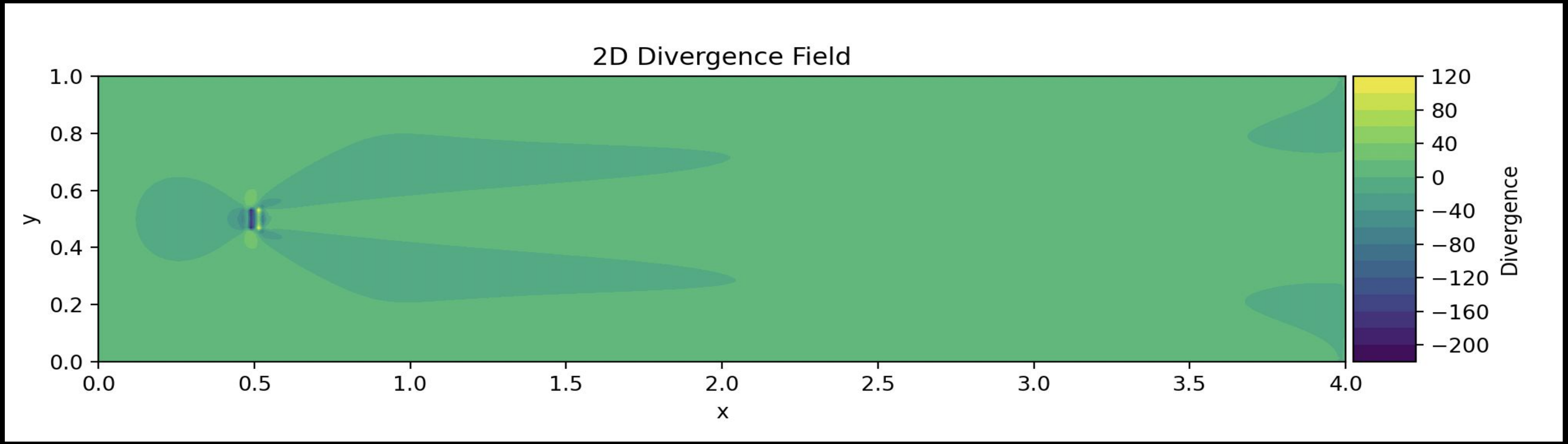
Interesting related papers:

- [An application of the actuator disc model for wind turbine wakes calculations]
-

Axial Induction from Actuator Disk Theory Alternative Method

This method consists of simply replacing the velocities using the axial induction factor at the rotor cells.



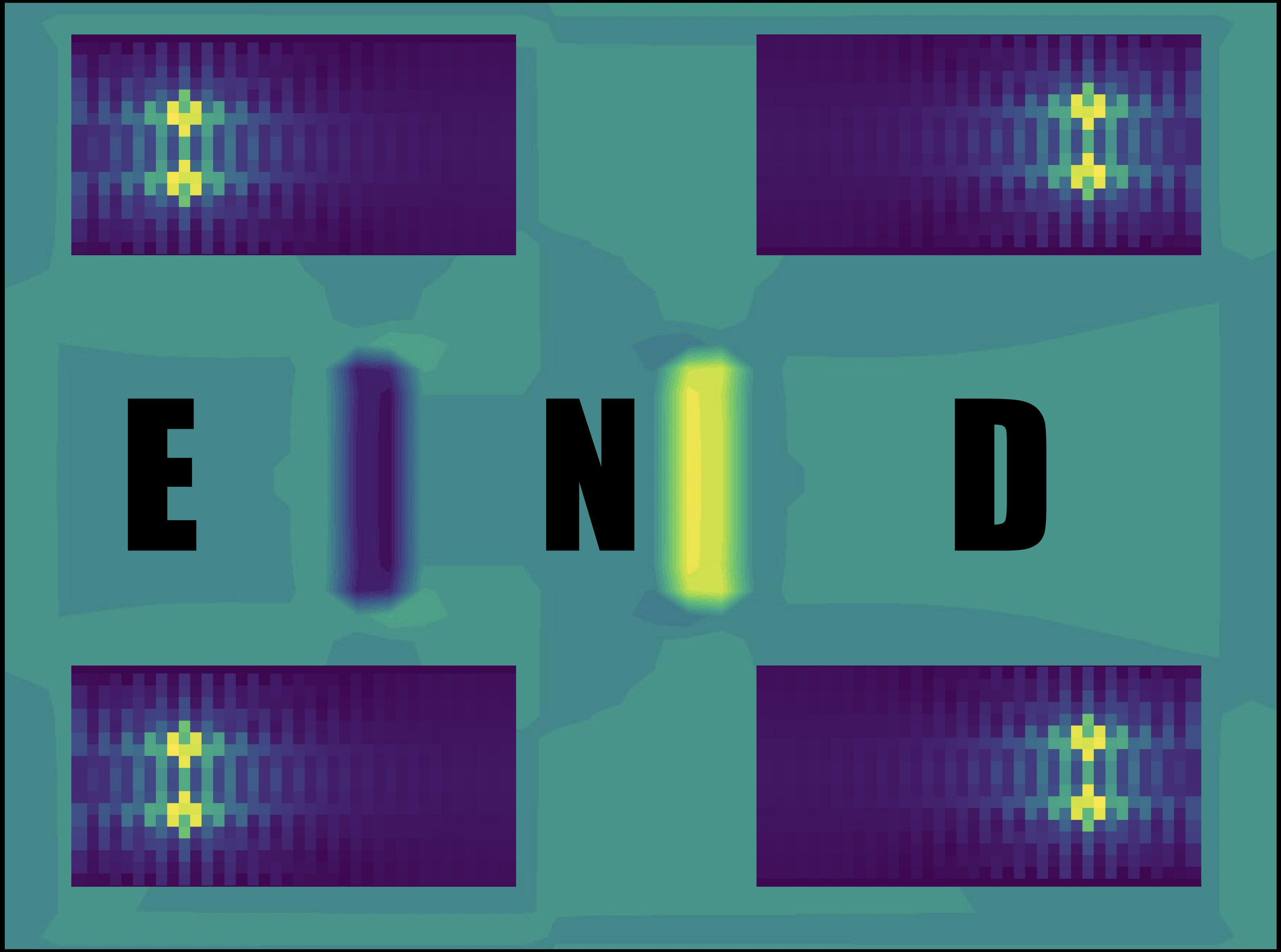


Divergence seems to be higher but the velocity profile makes sense.

I initially tried to crop the velocity by a factor of (1-a) but this led to unrealistic results (really low velocities) because it retrofeeds itself.

In this run, I updated the velocity at the rotor plane by a velocity of inVel*(1-a).

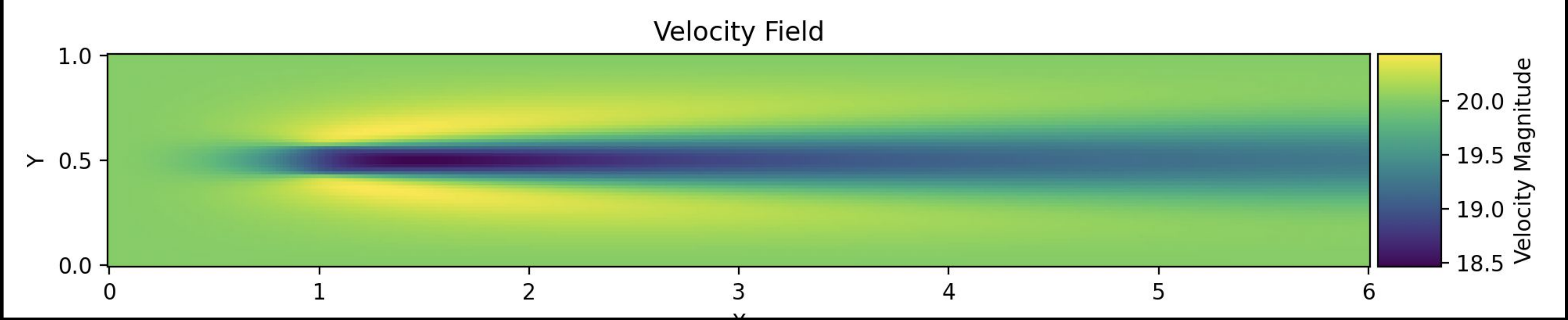
This might be a bad method due to the high divergence that it implies.

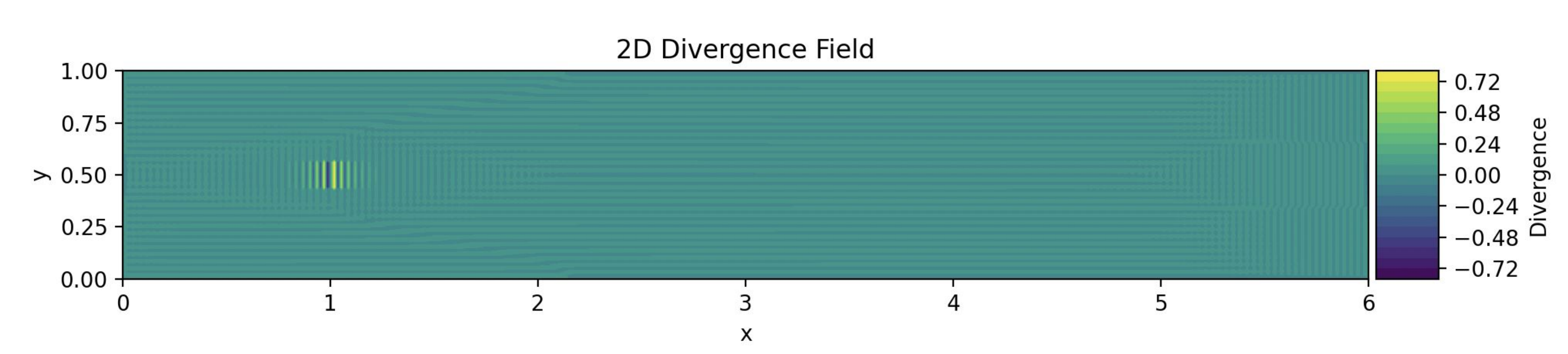

END

Dismissed slides

Channel Flow with body force F and no walls. Longer pipe.

Longer pipe —---> Domain: 60x640

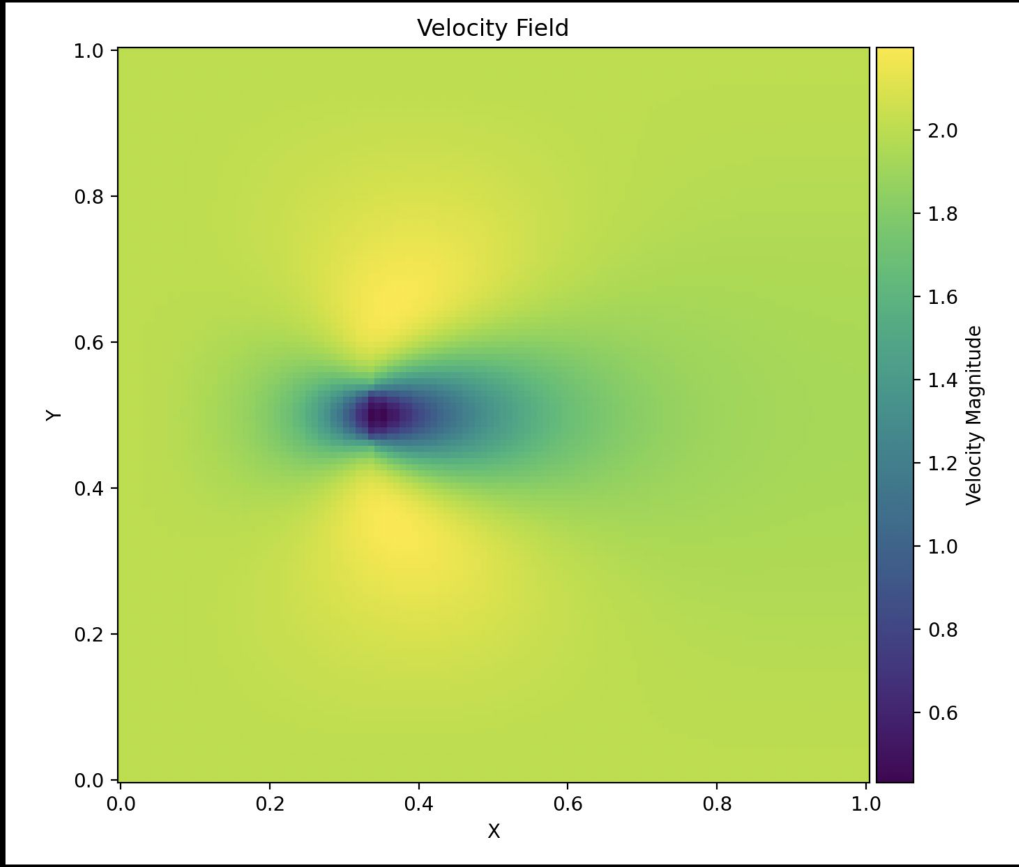


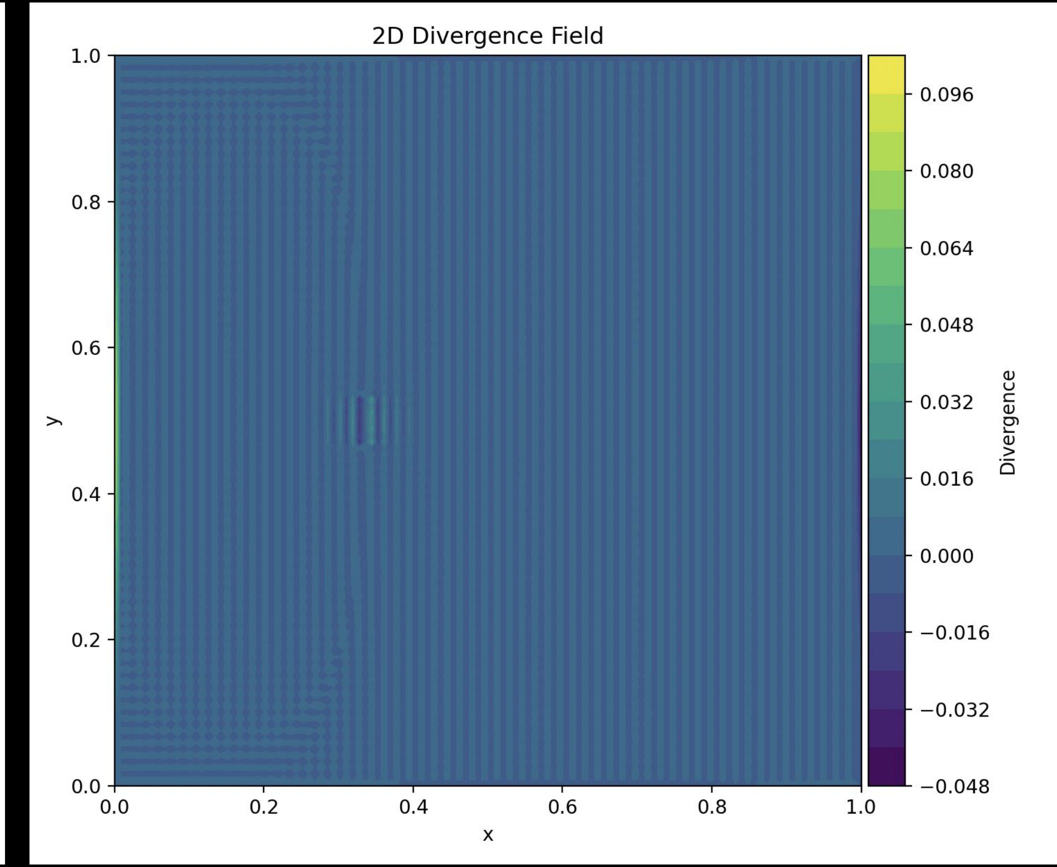


No walls, Single (constant force) Turbine

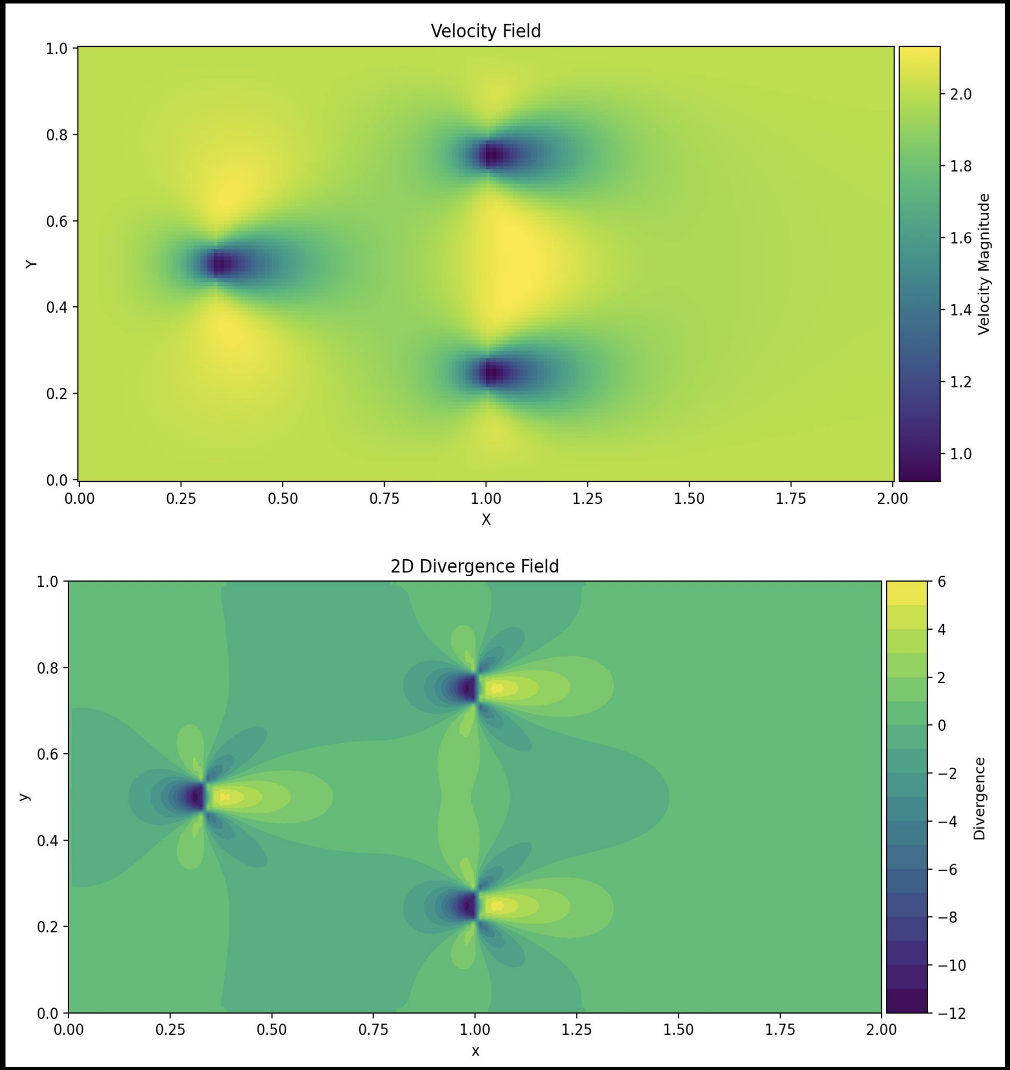
BC's:
- **Velocity.** **Inlet: Dirichlet** **Outlet, Top, Bottom: Neumann**
- **Pressure.** **Inlet, Top, Bottom: Neumann** **Outlet: Dirichlet**

nu=0.05
steps= 1000000
P iterations= 5
F= -1000
120x120
inVel = 2





Three turbine wind farm



BC's:
- Velocity.
  - Inlet: Dirichlet
  - Outlet, Top, Bottom: Neumann
- Pressure.
  - Inlet, Top, Bottom: Neuman
  - Outlet: Dirichlet

nu=0.05
steps= 500000
P iterations= 5
F= -1000
inVel= 20
120x240

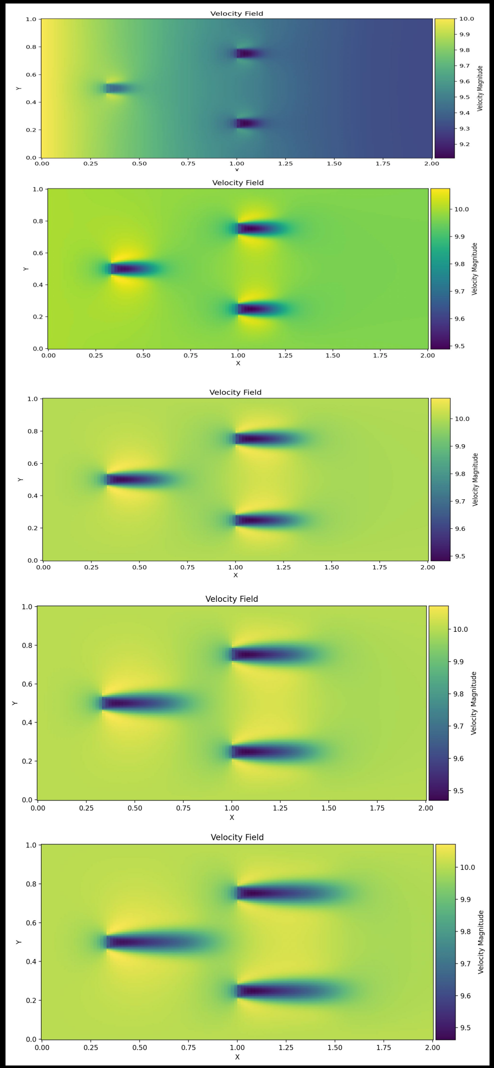
Divergence is high!
Diffusion looks really high too…

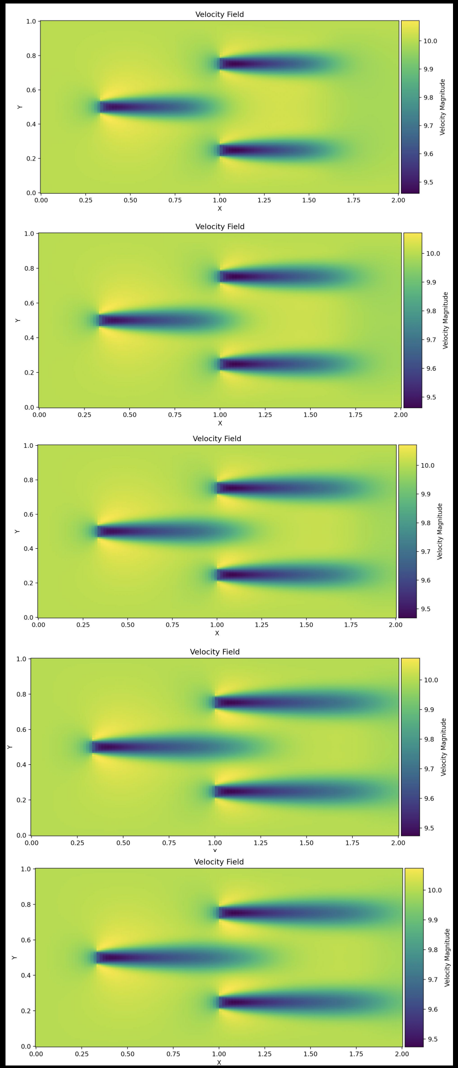
Three turbine wind farm

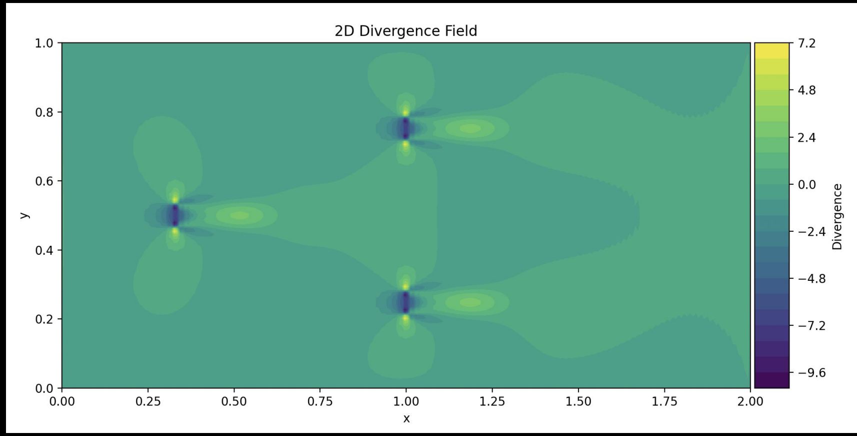
BC's:

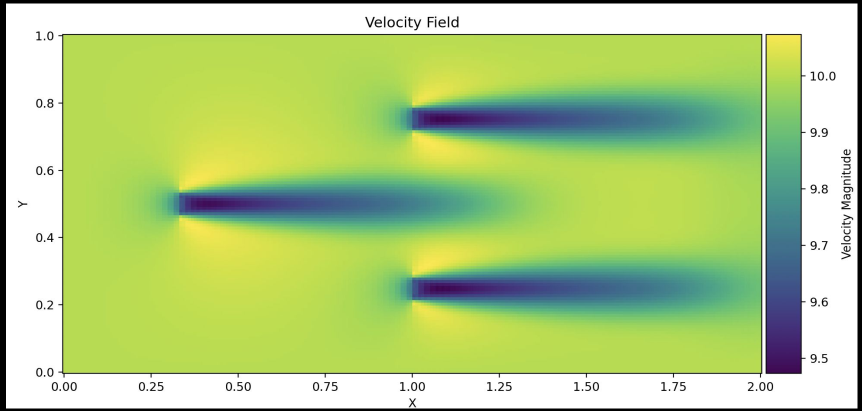
- **Velocity.**
  - **Inlet: Dirichlet**
  - **Outlet, Top, Bottom: Neumann**
- **Pressure.**
  - **Inlet, Top, Bottom: Neuman**
  - **Outlet: Dirichlet**

nu=0.05
steps= 500000
P iterations= 5
F= -1000
inVel= 10
120x240
dt=0.00001 (fix)







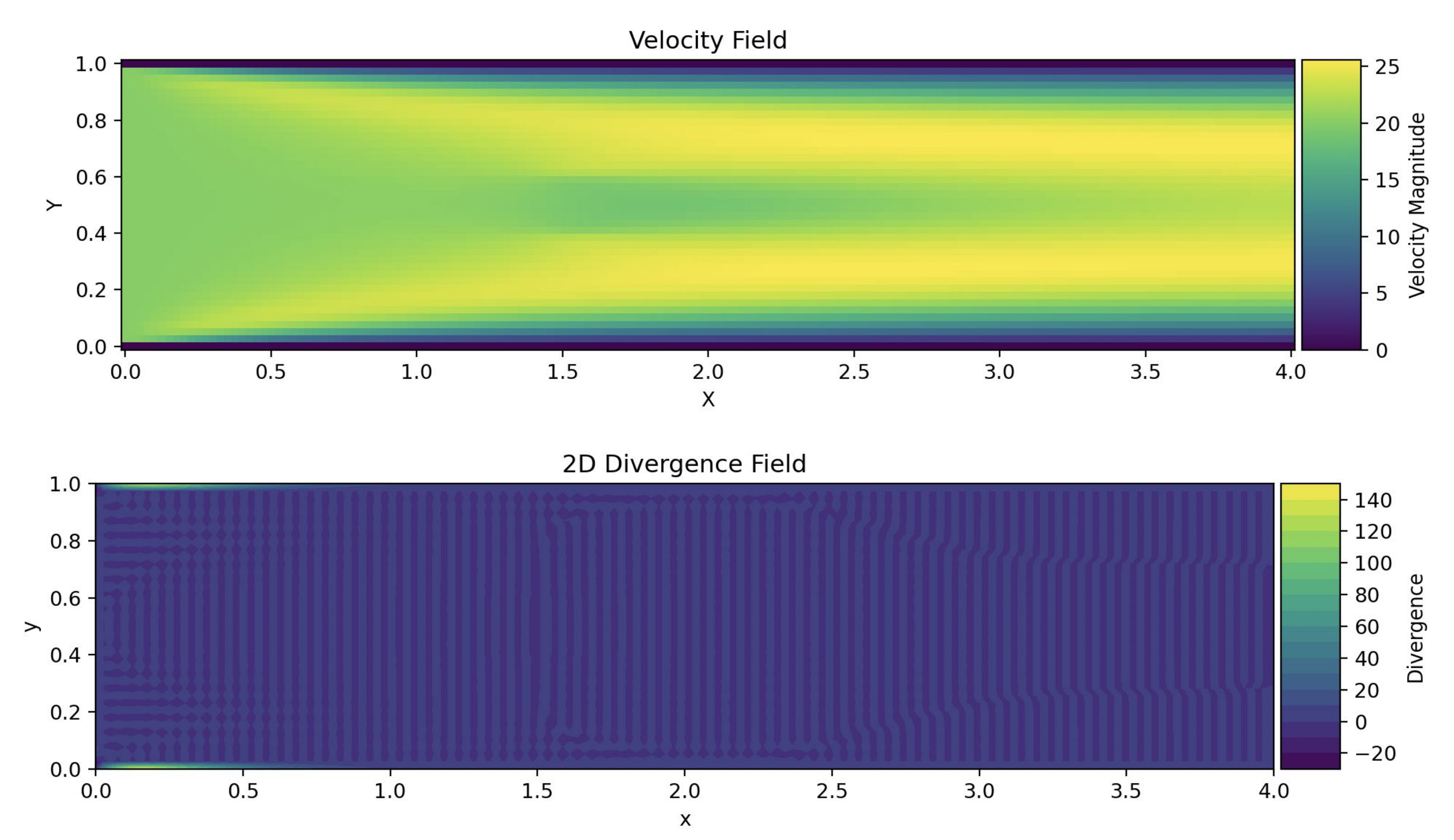


Divergence field barely changes.

Try more Poisson iterations?

Channel Flow (walls) with body force.

Applied the axial induction body force to the momentum eq itself.
No changes on the Poisson equation (because the divergence of a constant force is zero).



Does it make sense to add the turbine to a channel flow? It looks very much affected by the walls.. and it is not even fully developed..
Let's try with no walls and Neumann BC's?

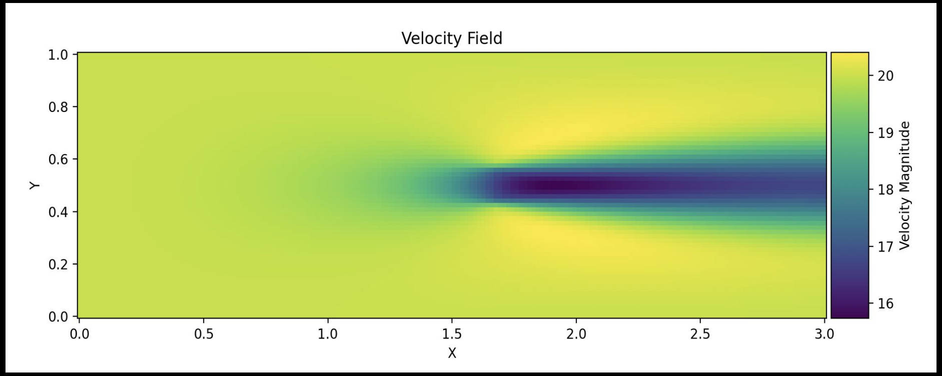
Divergence looks alright!

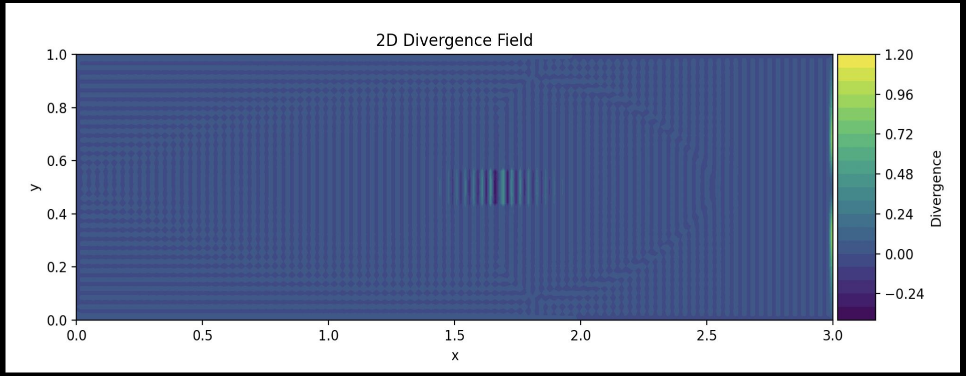
Re=50

Channel Flow with body force F and no walls. OLD

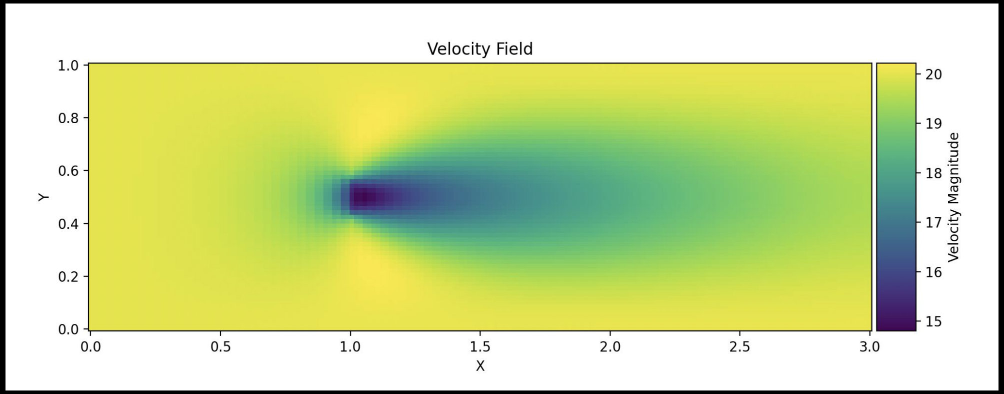
Applied the axial induction body force (F=-12000) to the momentum eq itself.
No changes on the Poisson equation (because the divergence of a constant force is zero).
Replace no-slip BCs by Neumann dv/dy=0 and du/dy=0 at top and bottom.

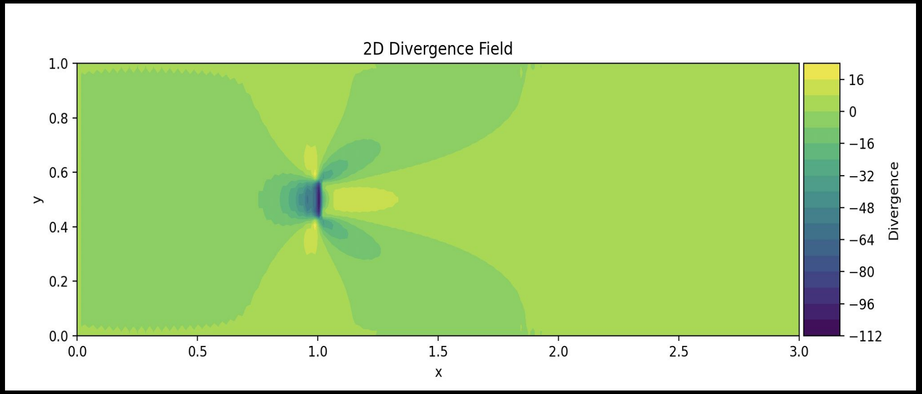
Domain: 60x180 Re: 40 nu = 0.5





How was this run? Why not divergence?



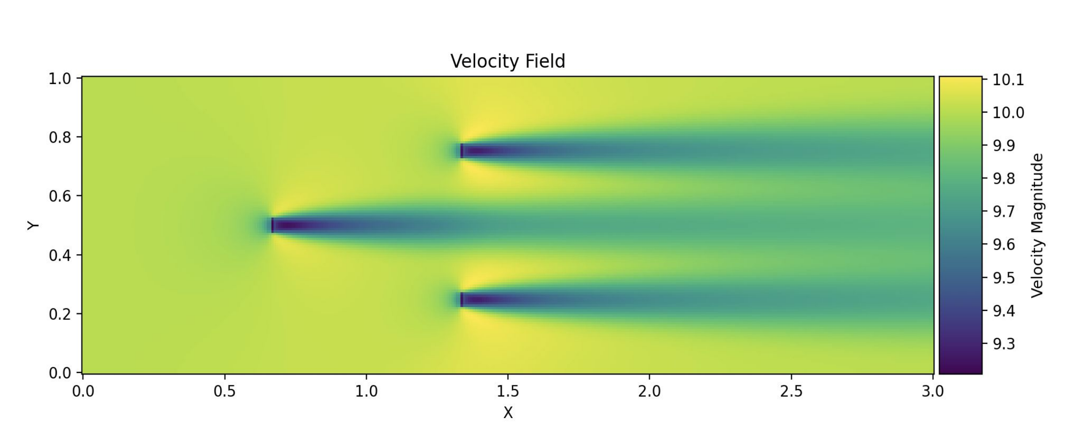


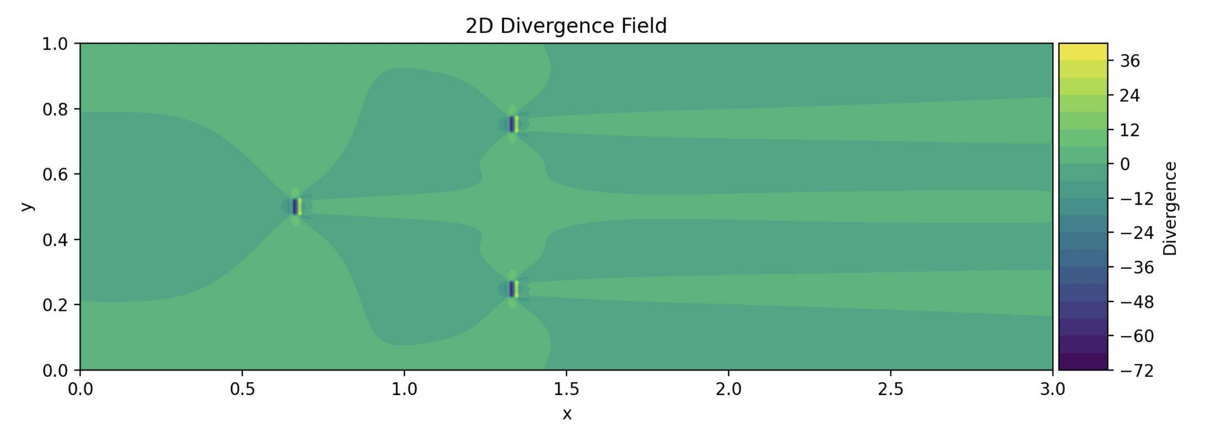
Three turbine wind farm

BC's:
- Velocity.
  - Inlet: Dirichlet
  - Outlet, Top, Bottom: Neumann
- Pressure.
  - Inlet, Top, Bottom: Neuman
  - Outlet: Dirichlet

nu=0.05
steps= 100000
P iterations= 8
F= -1200
inVel= 10
120x240
dt= 0.003 (auto)

1-8 iterations





I get the same diveregence no matter the number of iterations (tried 1 to 8).