PART 1

Task 1

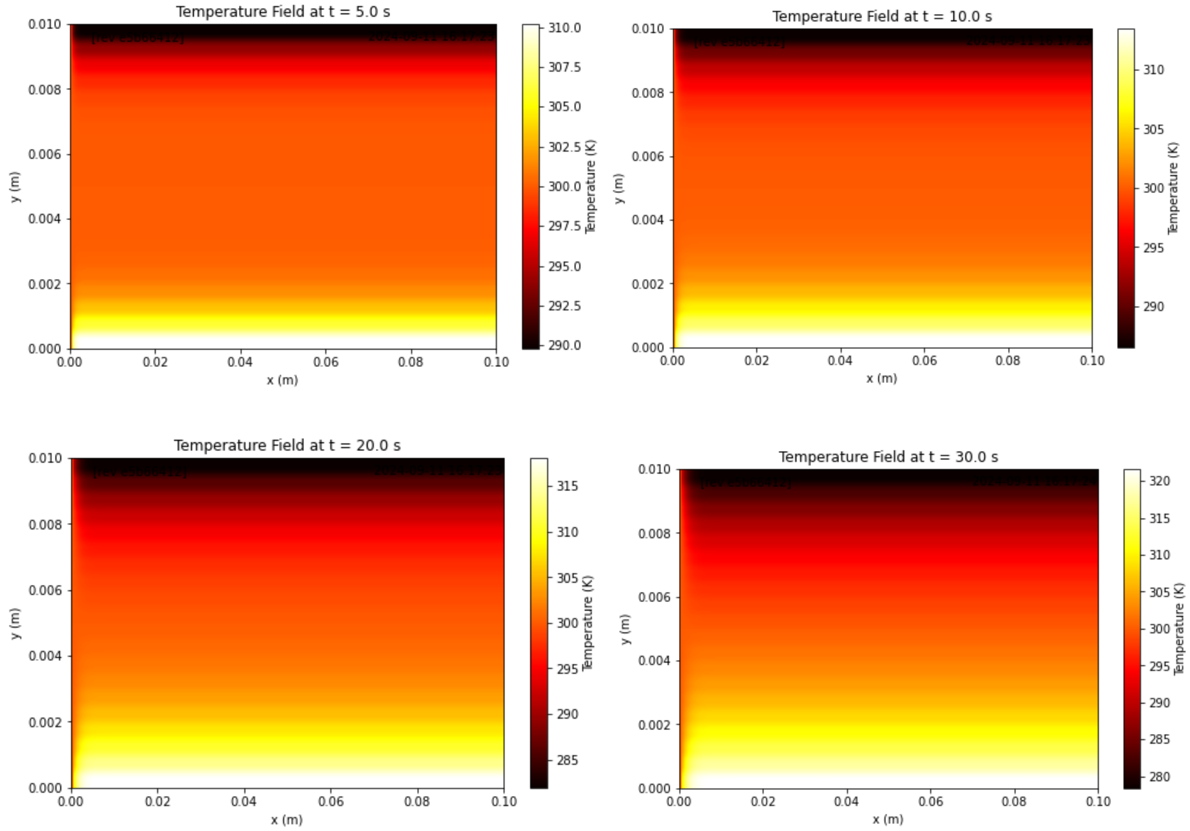
Q1: the discretization for the fluxes through the interior faces for both the convective and diffusive terms is based on finite differences, which can be summarized into two 3 steps

1. discretise the 2D domain into a grid with M ×N points. Which has been defined by Nx = 200; Ny = 20
2. Diffusion term (∇ · (k∇T)), This is a centered difference approximation for the temperature gradient in the x-direction. It uses the temperature values at the grid points i and i−1, where i is the current point, and i−1 is the previous point in the x-direction. This approach assumes that the convective flux is symmetrically distributed between the adjacent control volumes.
3. Convective term (∇·(uT)), second-order central difference approximations for the diffusion term, capturing the temperature gradient between neighboring grid points in both x and y directions.
4. Boundary at y=0 (bottom) and y=h (top): uses Neumann boundary condition for the temperature gradient, which corresponds to a constant heat flux at the boundaries based on equation: $Ti,0 = Ti,1 + \frac{qflux \cdot dy}{k}$ and $Ti,NY-1 = Ti,NY - \frac{qflux \cdot dy}{k}$ where q = 5000 W/m^2
5. Boundary at outlet also uses a Neumann condition (zero gradient) to represent an insulating boundary $TNX-1,j = TNX-2,j$
6. Time step which satisfies the stability criteria, considering both the convection and diffusion terms. The Courant-Friedrichs-Lewy (CFL) condition will help ensure stability.
7. Governing equation is the convection-diffusion equation becomes, after discretization:

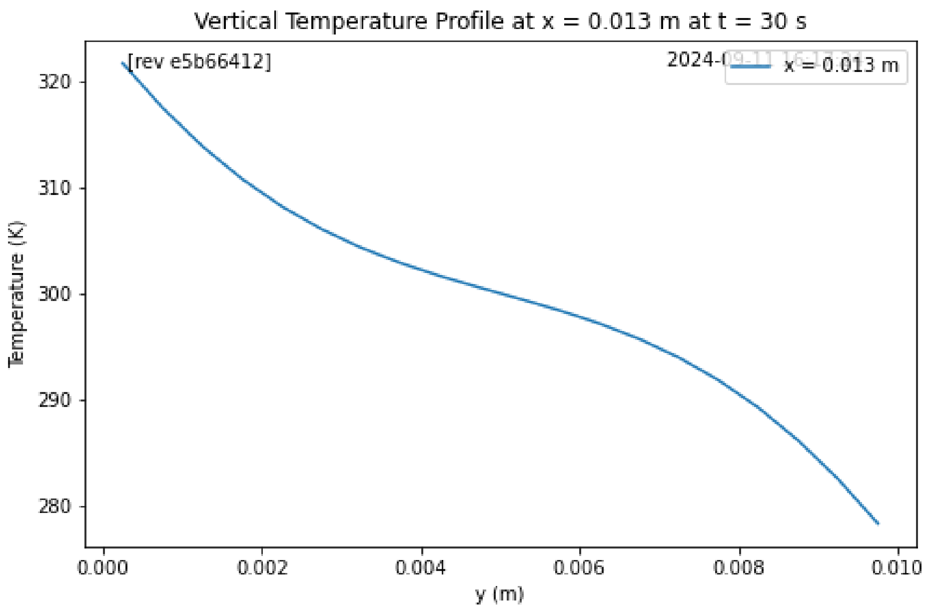
$$\rho cp \frac{(Ti,j^{n+1} - Ti,j^{n})}{\Delta t} + \frac{u(y)}{2\Delta x}(Ti+1,j^{n} - Ti-1,j^{n}) = k\left(\frac{Ti,j+1^{n} - 2Ti,j^{n} + Ti,j-1^{n}}{\Delta y^2} + \frac{Ti+1,j^{n} - 2Ti,j^{n} + Ti-1,j^{n}}{\Delta x^2}\right)$$

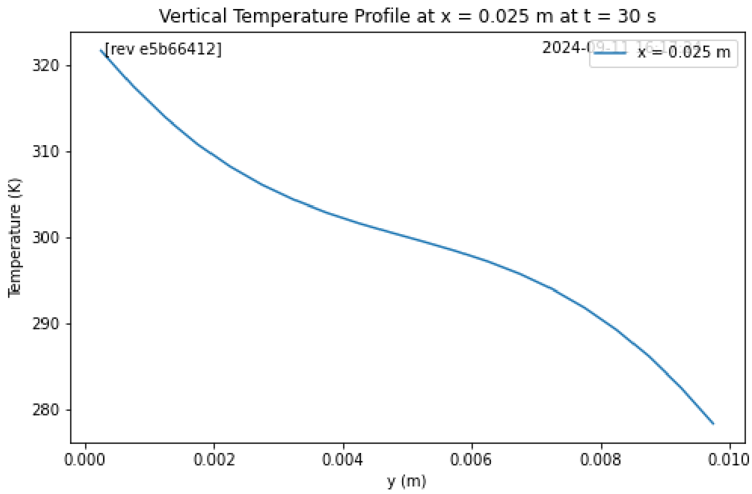
For a low Reynolds number (Re = 208), the flow remains laminar, where convective transport is smooth, and numerical diffusion is less problematic. In such flows, the centered difference scheme is effective, as the flow behavior is well-resolved without strong advection or sharp gradients that could cause instability. The Poiseuille flow profile, which is parabolic and smooth, further supports the use of centered differencing, as it captures the velocity variation without significant errors. Upwind schemes, typically necessary in turbulent flows to prevent oscillations, introduce unnecessary numerical diffusion in this case. The centered difference method, being second-order accurate, is preferred for accurately capturing the temperature gradients in such smooth flows. Additionally, since the problem assumes one-way coupling, where the velocity field remains unaffected by the thermal field, the flow behavior is predictable, further eliminating the need for upwind schemes and their artificial diffusion.

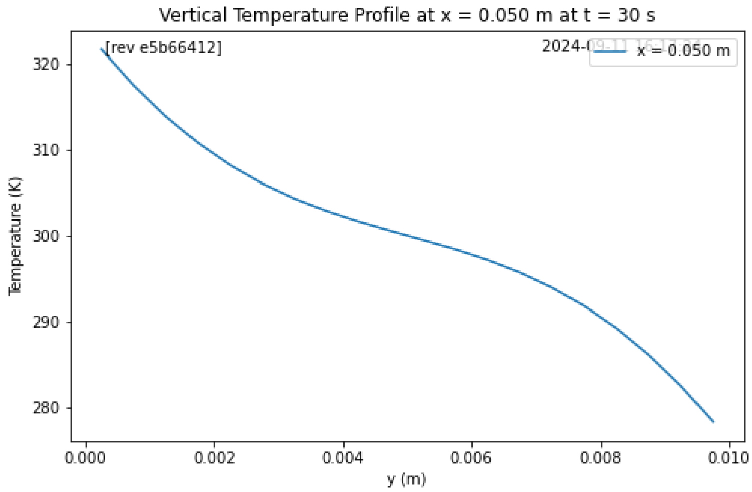
Q2: The time step has been set to 0.05 seconds to reduce number of iterations.



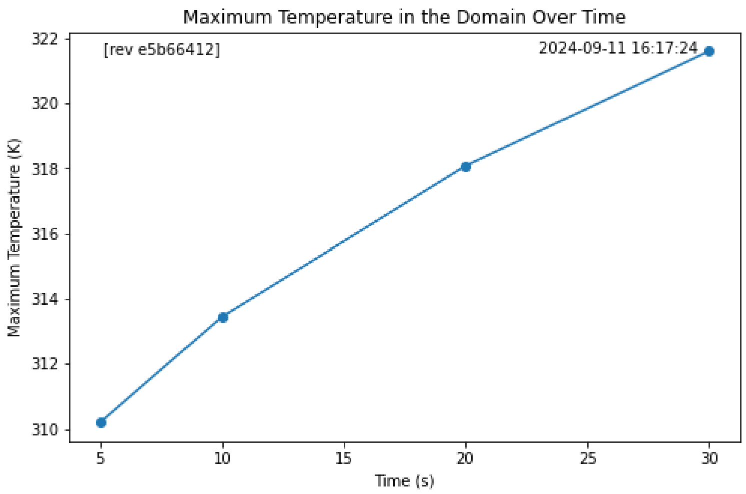
Q3:







Q4: Based on the Maximum Temperature profile and code, steady state has not been reached




```
In [15]: runfile('C:/Users/13747/OneDrive/文档/GitHub/mech6480-2024/contact-n-tutorial-w-07/untitled5.py', wdir='C:/Users/13747/OneDrive/文档/GitHub/mech6480-2024/contact-n-tutorial-w-07')
Total computation time: 6.41 seconds
Based on the plot of the maximum temperature over time, steady-state is not reached.
```

Part 2
Task 2

In y-direction:

$$\frac{\partial}{\partial t} v + \nabla \cdot v u = \frac{-\nabla p}{\rho} + \nu \nabla^2 u$$

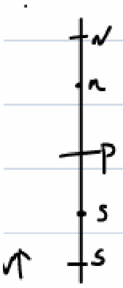
apply FVM:

$$\underbrace{\frac{\partial}{\partial t} \int_{cv} v \, dV}_{\text{change in y-momentum}} = -[\underbrace{\int_{cs} v u \cdot n \, dS}_{\text{convection (A)}} - \underbrace{\nu \int_{cs} \nabla v \cdot n}_{\text{diffusion (D)}}] - \underbrace{\frac{1}{\rho} \int_{cs} p \cdot n_y \, dS}_{\text{pressure}}$$

Time integration:

$$\frac{v^{n+1} - v^n}{\Delta t} = -A_v^n + D_v^n - \frac{1}{\rho}\frac{\partial p}{\partial y}$$

$\nabla \cdot u = 0$



$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \qquad \frac{\partial u^{n+1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y} = 0$$

$$\frac{\partial u^{n+1}}{\partial x} + \frac{\partial}{\partial y}\left(v^n + \Delta t\left(-A_v^n + D_v^n - \frac{1}{\rho}\frac{\partial p}{\partial y}\right)\right) = 0$$

$$\frac{\Delta t}{\rho} \nabla^2 p = \nabla \cdot (A_v^n - D_v^n)$$

let $v^* = v^n + \Delta t(-A_v^n + D_v^n)$

$$\nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot u^* \quad [u^* \text{ been given in the task}].$$

$$v^{n+1} = v^* - \frac{\Delta t}{\rho}\frac{\partial p}{\partial y}.$$

convection term $- A_y$

$$A_y = \int_{cs} v^n (u^n \cdot n) \, dS$$

$$A_y = (uv)_e^n \Delta y - (uv)_w^n \Delta y + (vv)_n^n \Delta x - (vv)_s^n \Delta x$$

A x-direction:

$$\Delta y\left(\frac{u^n_{i+1,j} + u^n_{i+1,j-1}}{2}\right)\left(\frac{v^n_{i+1,j} + v^n_{i,j}}{2}\right) - \Delta y\left(\frac{u^n_{i,j} + u^n_{i,j-1}}{2}\right)\left(\frac{v^n_{i,j} + v^n_{i-1,j}}{2}\right)$$

A y-direction:

$$\Delta x\left(\frac{v^n_{i,j+1} + v^n_{i,j}}{2}\right)^2 - \Delta x\left(\frac{v^n_{i,j} + v^n_{i,j-1}}{2}\right)^2$$

$$\boxed{A_y = A\ x\text{-direction} + A\ y\text{-direction}}$$

Diffusion term: D_y.

$$D_y = \nu \int_{cs} \nabla v^n \cdot n \, dS$$

$$D_y = \nu \left(\frac{\partial v}{\partial x}\right)_e \Delta y - \nu \left(\frac{\partial v}{\partial x}\right)_w \Delta y + \nu \left(\frac{\partial v}{\partial y}\right)_n \Delta x - \nu \left(\frac{\partial v}{\partial y}\right)_s \Delta x$$

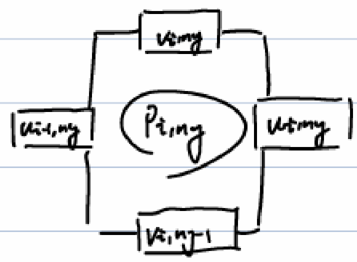
same central difference approach:

$$\boxed{D_y = \nu \left[\Delta y \left(\frac{v^n_{i+1,j} - v^n_{i,j}}{\Delta x} - \frac{v^n_{i,j} - v^n_{i-1,j}}{\Delta x} \right) + \Delta x \left(\frac{v^n_{i,j+1} - v^n_{i,j}}{\Delta y} - \frac{v^n_{i,j} - v^n_{i,j-1}}{\Delta y} \right) \right]}$$

now v^* has been derived, $v^{n+1}_{i,j} = v^*_{i,j} - \frac{\Delta t}{\rho} \frac{\partial p}{\partial y}$

$$= v^*_{i,j} - \frac{\Delta t}{\rho \Delta y} (p_{i,j+1} - p_{i,j})$$

every term in this discretization for y momentum has been analyzed.

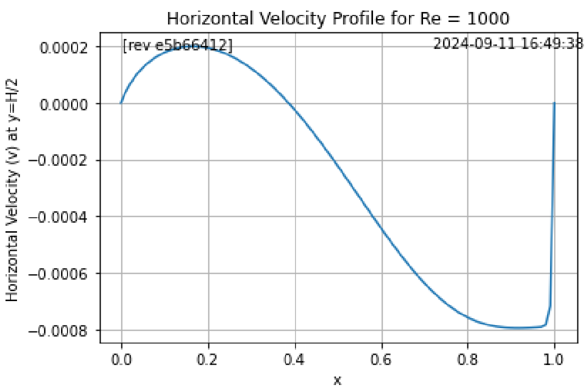


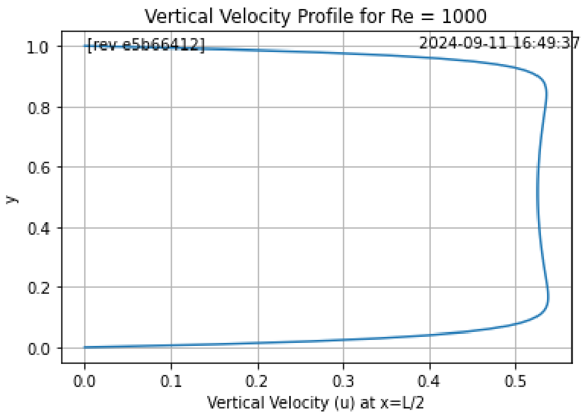
Task 3

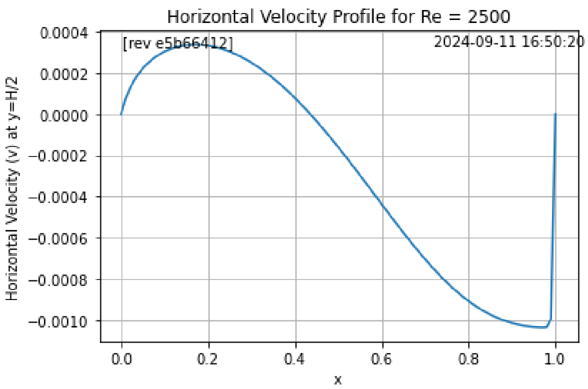
Q1: While using finite difference methods to discretise the Navier-Stokes equations on this structured grid.

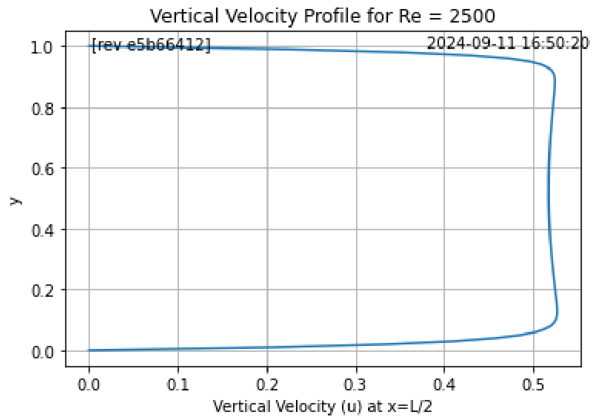
1. Grid Size (Δx, Δy): The grid resolution has been selected such that it is possible to capture sufficient detail for both cases of Re=1000 and Re=2500. Typically, a grid size of 100×100 or higher is reasonable for this kind of problem. Smaller the grid size, corresponds to more accurate solution, but it increases computational cost and time. Thus, $\Delta x = \Delta y = 1/N$ has been set, where N is the number of grid points in each direction (N = 100).
2. Time Step (Δt): The time step should satisfy the CFL condition for stability:
$$\Delta t < \frac{Re \cdot (\Delta x)^2}{2}$$
This ensures stability in solving the Navier-Stokes equations using an explicit time-stepping method. Alternatively, an implicit or semi-implicit method can allow for larger time steps.
3. Steady-State: Steady-state can be determined when the residuals or changes in velocity fields between iterations are below a certain tolerance (10^-6). Simulating until changes are negligible in velocity fields will provide a steady-state solution.

Q2:
The following are the simulation results generated from the code

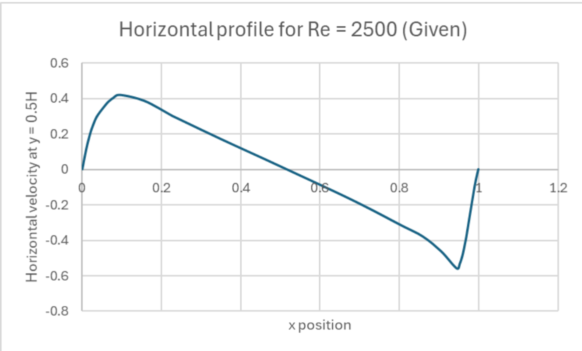


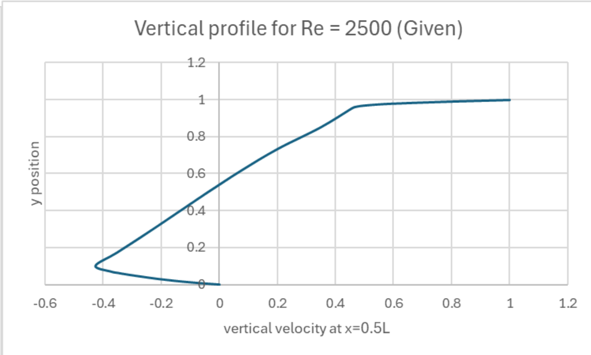


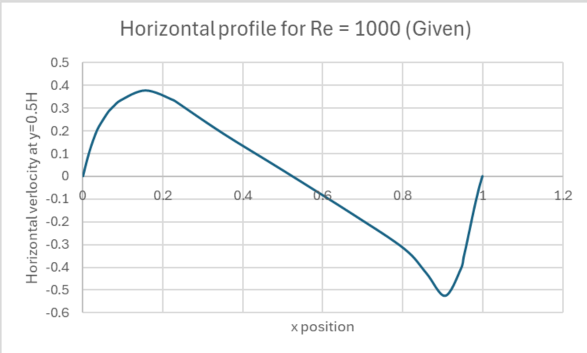


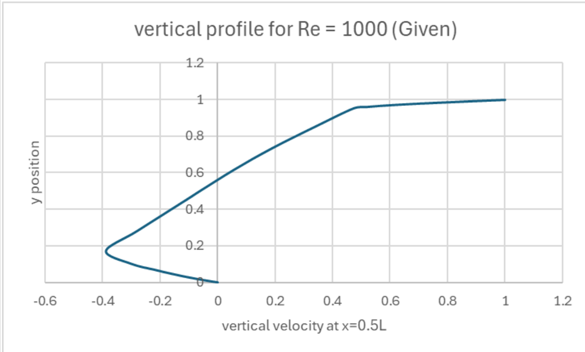


The following result is generated from the given data:

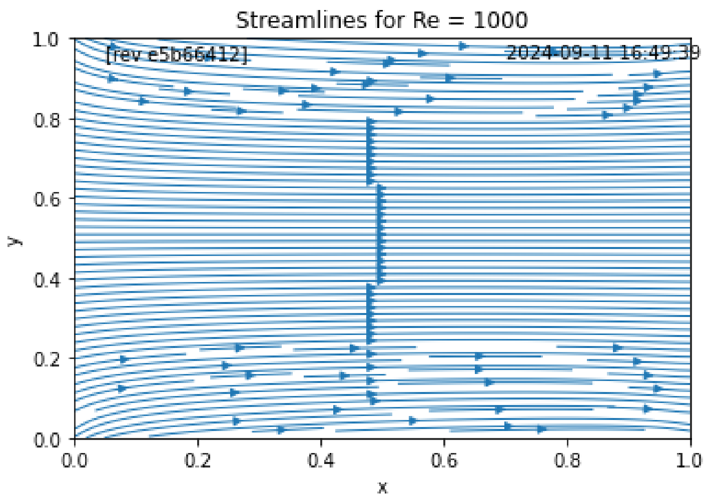


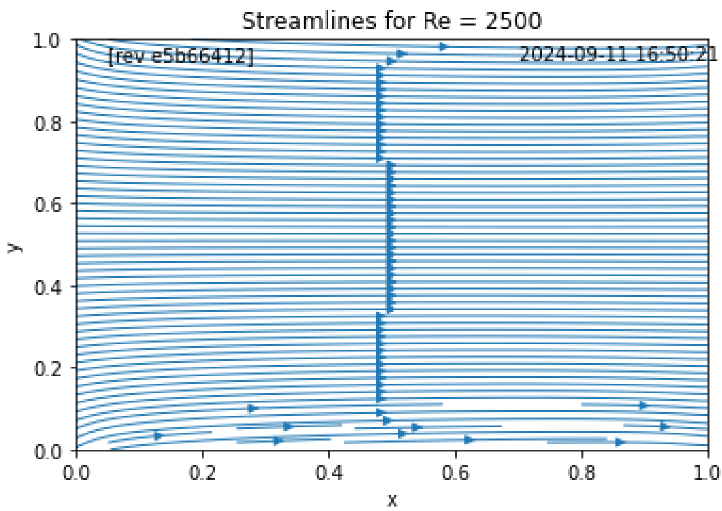






Q3:





Q4 the delta t value has been set to 0.001 and plot every 100 iterations, hwoever during the forming of animation the software limitation do not allow more than 30 flames of picture where steady states reached around 5000 iterations. Thus, the animation contains picture with a 500 iteration step.

```
In [18]: runfile('C:/Users/13747/OneDrive/文档/GitHub/mech6480-2024/tutorial-w-01/untitled8.py', wdir='C:/Users/13747/OneDrive/文档/GitHub/mech6480-2024/tutorial-w-01')
Steady-state reached after 4874 iterations
Steady-state reached after 4874 iterations
Steady-state reached after 5369 iterations
```

Task 4

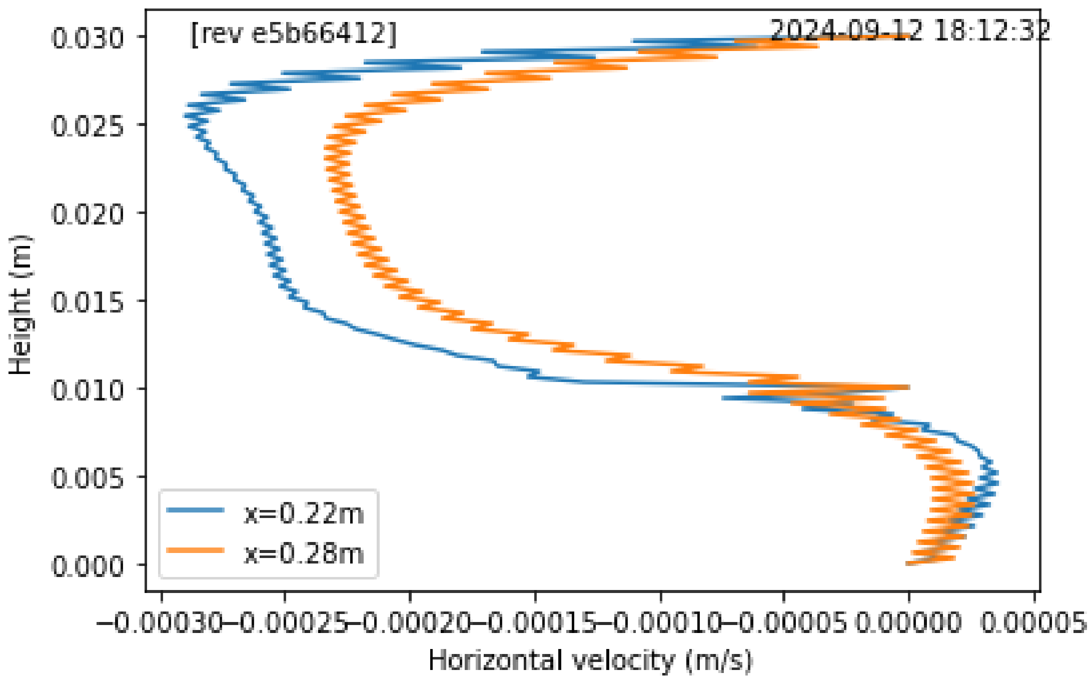
Q1. Boundary conditions: for the backward-facing step, a uniform inlet velocity is applied at the left boundary above the step, with a no-slip condition on the top, bottom walls and step. At the outlet (right boundary), a zero-gradient condition is applied for velocity, allowing flow to exit without reflections. The vertical velocity is set to zero at the inlet, while horizontal velocity below the step is also set to zero.

Grid: compared to the previous task, the horizontal direction has 300 grid points where there are 100 points vertically.

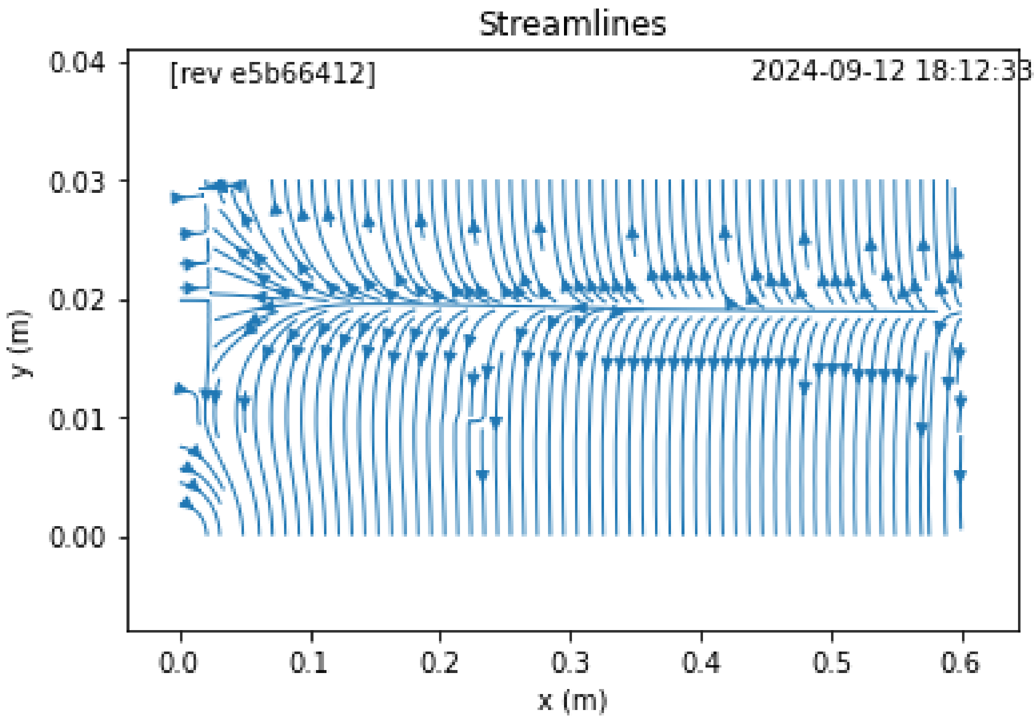
A time step Δt of 0.0001 seconds was chosen to satisfy the CFL condition for numerical stability, ensuring that the simulation captures the flow evolution accurately without causing instabilities.

Steady state: same as before, the steady state was determined by monitoring the change in the velocity fields between iterations. Once the norm of the difference between successive velocity fields dropped below a tolerance of 10^−6, the flow was assumed to have reached steady-state.

Q2.



Q3.



Streamlines
[rev e5b66412]
2024-09-12 18:12:33
0.04
0.03
0.02
0.01
0.00
y (m)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
x (m)