Tutorial 5

https://github.com/Blink29/Numerical-Techniques → Assignment 5

Name: Paurush Kumar   Roll Number: NA22B002

Problem Statement

Solving the Lid-Driven Cavity Flow problem using Vorticity-Stream function method and visualizing the velocity distribution, streamlines, and centerline velocity profiles.

Governing Equations

The governing equations for lid-driven cavity flow are the Navier-Stokes equations for incompressible flow. In the stream function-vorticity formulation, these equations are transformed to decouple the pressure from velocity fields.

- **Vorticity Transport Equation**:

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right)$$

  where ω is the vorticity, u and v are the velocity components, and Re is the Reynolds number defined as:

$$Re = \frac{\rho U L}{\mu}$$

- **Stream Function-Vorticity Relation**:

$$\nabla^2 \psi = -\omega$$

  where ψ is the stream function.

The velocities are derived from the stream function:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}$$

Code

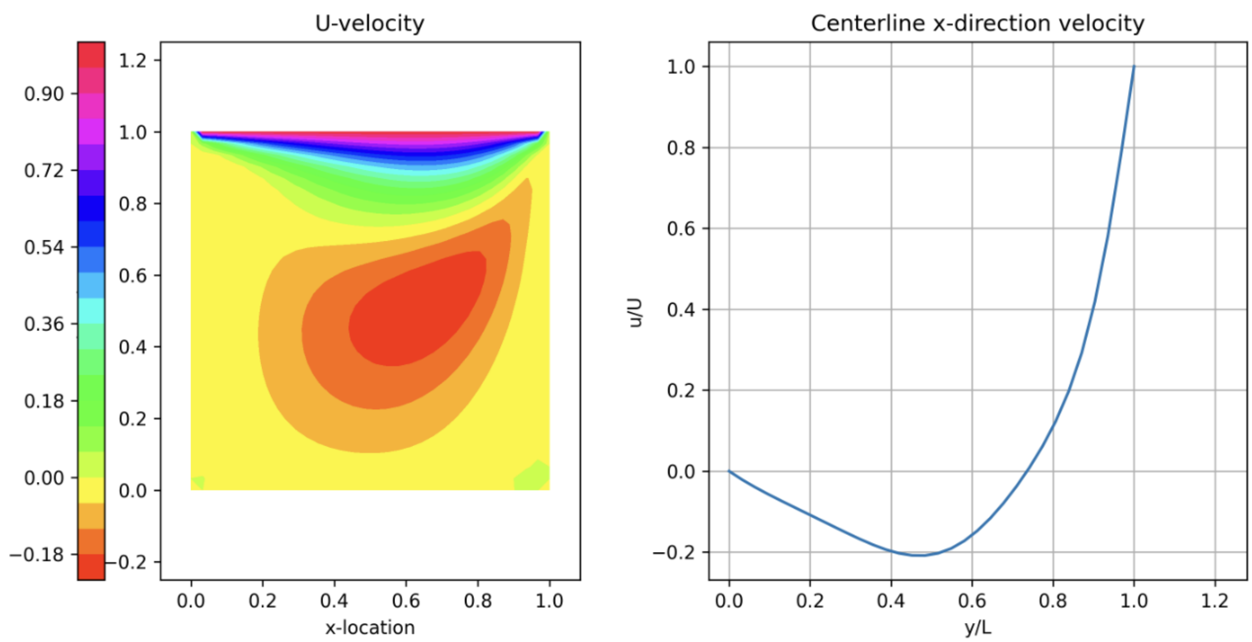
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle

# Parameters
Nx = 32 # Number of grid points in x-direction
L = 1 # Length of the domain
Wall_Velocity = 1 # Velocity of the moving wall
rho = 1 # Density
mu = 0.01 # Viscosity
dt = 0.001 # Time step
maxIt = 50000 # Maximum number of iterations
maxe = 1e-7 # Maximum error for convergence
Re = rho * Wall_Velocity * L / mu # Reynolds number

# Grid setup
Ny = Nx
h = L/(Nx-1)
x = np.linspace(0, L, Nx)
y = np.linspace(0, L, Ny)
X, Y = np.meshgrid(x, y)

# Initialize arrays
Vo = np.zeros((Nx, Ny)) # Vorticity
St = np.zeros((Nx, Ny)) # Stream function
u = np.zeros((Nx, Ny)) # x-velocity
v = np.zeros((Nx, Ny)) # y-velocity

# Main solver loop
for iter in range(maxIt):
    # Boundary conditions
    Vo[0:Nx, Ny-1] = -2*St[0:Nx, Ny-2]/(h**2) - Wall_Velocity
*2/h # Top
    Vo[0:Nx, 0] = -2*St[0:Nx, 1]/(h**2) # Bottom
```

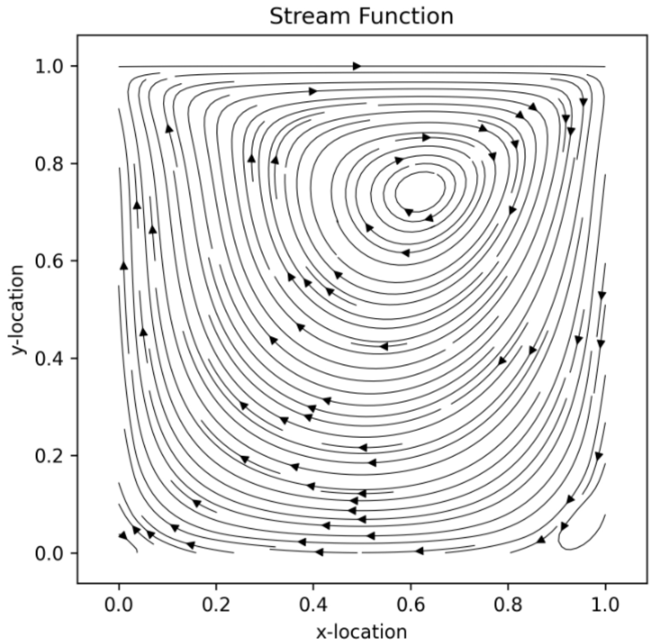
```
    Vo[0, 0:Ny] = -2*St[1, 0:Ny]/(h**2) # Left
    Vo[Nx-1, 0:Ny] = -2*St[Nx-2, 0:Ny]/(h**2) # Right

    # Store old vorticity
    Vop = Vo.copy()

    # Update vorticity (interior points)
    i = slice(1, Nx-1)
    j = slice(1, Ny-1)
    ip = slice(2, Nx)
    im = slice(0, Nx-2)
    jp = slice(2, Ny)
    jm = slice(0, Ny-2)

    Vo[i,j] = Vop[i,j] + dt * (
        -1*(St[i,jp]-St[i,jm])/(2*h) * (Vop[ip,i]-Vop[im,j])/
(2*h) +
        (St[ip,j]-St[im,j])/(2*h) * (Vop[i,jp]-Vop[i,jm])/(2*
h) +
        (1/Re)*(Vop[ip,j]+Vop[im,j]-4*Vop[i,j]+Vop[i,jp]+Vop
[i,jm])/(h**2)
    )

    # Update stream function
    St[i,j] = (Vo[i,j]*h**2 + St[ip,j] + St[i,jp] + St[i,jm]
+ St[im,j])/4

    # Check convergence
    if iter > 10:
        error = np.max(np.abs(Vo - Vop))
        if error < maxe:
            print(f"Converged after {iter} iterations")
            break

# Calculate velocities
u[1:Nx-1, Ny-1] = Wall_Velocity
```

```
u[i,j] = (St[i,jp]-St[i,jm])/(2*h)
v[i,j] = (-St[ip,j]+St[im,j])/(2*h)


plt.figure(figsize=(15, 5))

# U-velocity Plot
plt.subplot(131)
plt.contourf(X, Y, u.T, levels=23, cmap='hsv')
plt.colorbar(location='left')
plt.title('U-velocity')
plt.xlabel('x-location')
plt.ylabel('y-location')
plt.axis('equal')

# Centerline velocity Plot
plt.subplot(132)
plt.plot(y, u[Nx//2,:])
plt.title('Centerline x-direction velocity')
plt.xlabel('y/L')
plt.ylabel('u/U')
plt.grid(True)
plt.axis('square')

# Streamlines Plot
plt.subplot(133)
N = 1000
xstart = L * np.random.rand(N)
ystart = L * np.random.rand(N)
plt.streamplot(X, Y, u.T, v.T, start_points=np.column_stack
((xstart, ystart)),
              density=2, linewidth=0.5, color='k')
plt.title('Stream Function')
plt.xlabel('x-location')
plt.ylabel('y-location')
plt.axis('equal')
```

```
plt.tight_layout()

plt.savefig('streamlines_plot.png', dpi=300, bbox_inches='tig
ht')
plt.show()
```

Results and Visualization





Conclusion

- The velocity profile along the cavity centerline (vertical line at x=L/2) demonstrates the formation of primary vortices. The profile is symmetric for this setup, reflecting the equilibrium state achieved in the flow.
- Streamlines visualize the flow structure and capture vortices within the cavity. A dominant primary vortex forms in the center of the cavity, accompanied by secondary vortices near the corners.
- The results converge after 17449 iterations, demonstrating the stability and accuracy of the numerical scheme.