Tutorial 4

https://github.com/Blink29/Numerical-Techniques → Assignment 4

Name: Paurush Kumar   Roll Number: NA22B002

Problem Statement

To solve the Laplace equation in a two-dimensional potential flow with a positive source and a negative sink

Governing Equations

1. **Laplace's Equation**

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = -S(x, y)$$

where $f(x, y)$ is the potential and $S(x, y)$ is the source term.

2. **Velocity Components**

Velocity is derived from the potential field using:

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

3. **Boundary Conditions**

- **Dirichlet Conditions**:

$$f = 10 \quad \text{at the entrance}$$

$$f = 0 \quad \text{at the exit opening}$$

- **Neumann Conditions**:

$$\frac{\partial f}{\partial y} = 0 \quad \text{on upper and lower walls}$$

Code

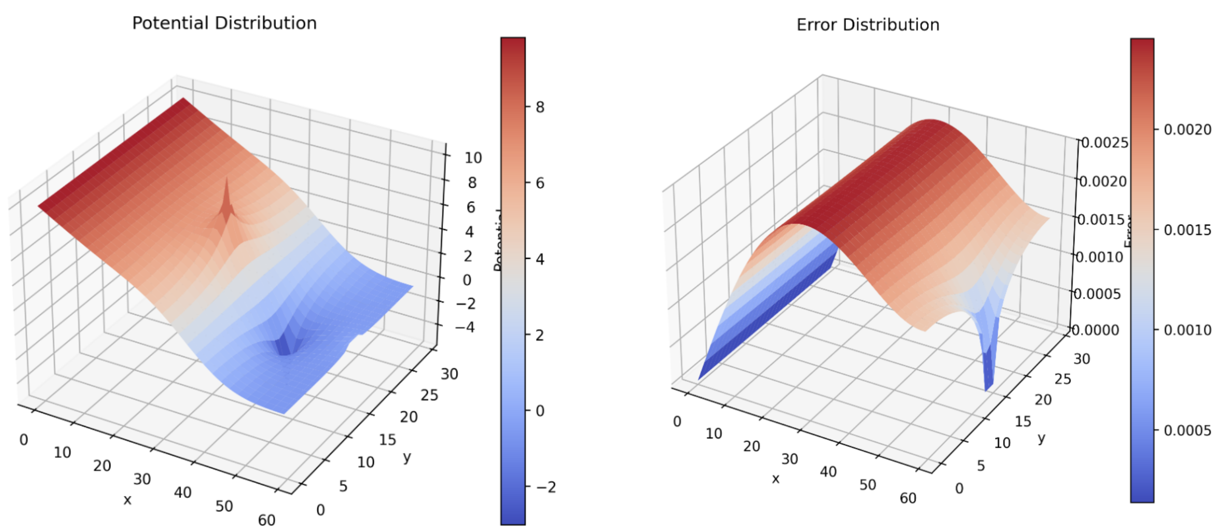
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

# Input parameters
im = 30  # vertical nodes
jm = 60  # horizontal nodes
SS = 10  # positive source strength
SN = -10 # negative source strength
iter = 1000

# Initialize grid
f = np.zeros((im, jm))
S = np.zeros((im, jm))

# Set sources
S[14, 29] = SS  # Positive source (15,30)
S[14, 44] = SN  # Negative source (15,45)

# Dirichlet boundary conditions
f[:, 0] = 10    # Entrance
is_ = im // 2   # Center index
f[is_ - 1:is_ + 2, -1] = 0  # Exit opening

# Main iteration loop
for k in range(iter):
    fold = f.copy()

    # Interior points
    for i in range(1, im - 1):
        for j in range(1, jm - 1):
            f[i, j] = (f[i + 1, j] + f[i - 1, j] + f[i, j -
1] + f[i, j + 1] + S[i, j]) / 4
```

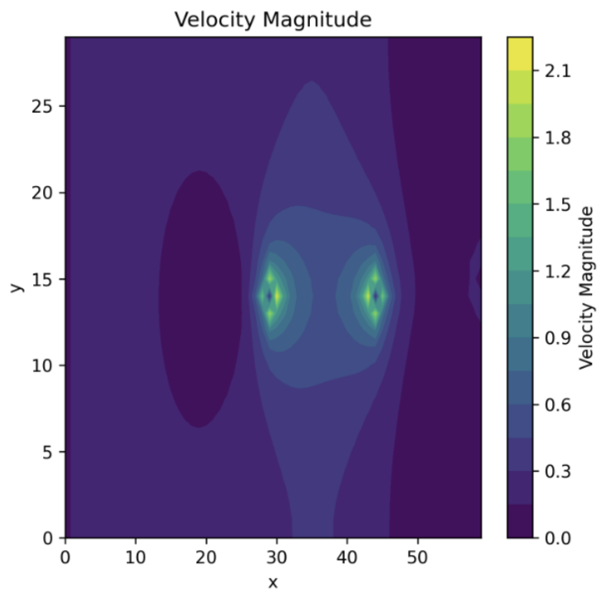
```
    # Von Neumann boundary conditions
    # Upper and lower walls
    f[0, :] = f[1, :]
    f[-1, :] = f[-2, :]

    # End wall (except opening)
    f[:is_ - 1, -1] = f[:is_ - 1, -2]
    f[is_ + 2:, -1] = f[is_ + 2:, -2]

    # Calculate error
    fer = f - fold

# Calculate velocities using central differences
u = np.zeros_like(f)
v = np.zeros_like(f)

# Interior points
u[:, 1:-1] = -(f[:, 2:] - f[:, :-2]) / 2  # df/dx
v[1:-1, :] = -(f[2:, :] - f[:-2, :]) / 2  # df/dy

vel_mag = np.sqrt(u**2 + v**2)

X, Y = np.meshgrid(np.arange(jm), np.arange(im))

# Potential distribution
fig1 = plt.figure(figsize=(8, 6))
ax1 = fig1.add_subplot(111, projection='3d')
surf1 = ax1.plot_surface(X, Y, f, cmap=cm.coolwarm)
plt.colorbar(surf1, ax=ax1)
ax1.set_title('Potential Distribution')
ax1.set_xlabel('x')
ax1.set_ylabel('y')
ax1.set_zlabel('Potential')
plt.savefig('potential_distribution.png', dpi=300, bbox_inche
s='tight')
plt.close(fig1)
```

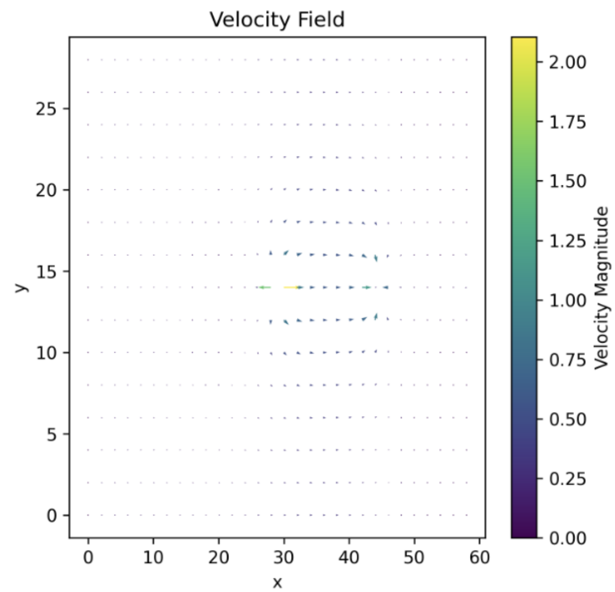
```
# Error distribution
fig2 = plt.figure(figsize=(8, 6))
ax2 = fig2.add_subplot(111, projection='3d')
surf2 = ax2.plot_surface(X, Y, fer, cmap=cm.coolwarm)
plt.colorbar(surf2, ax=ax2)
ax2.set_title('Error Distribution')
ax2.set_xlabel('x')
ax2.set_ylabel('y')
ax2.set_zlabel('Error')
plt.savefig('error_distribution.png', dpi=300, bbox_inches='t
ight')
plt.close(fig2)

# Velocity magnitude
fig3 = plt.figure(figsize=(8, 6))
ax3 = fig3.add_subplot(111)
contour = ax3.contourf(X, Y, vel_mag, levels=20, cmap='viridi
s')
plt.colorbar(contour, ax=ax3, label='Velocity Magnitude')
ax3.set_title('Velocity Magnitude')
ax3.set_xlabel('x')
ax3.set_ylabel('y')
plt.savefig('velocity_magnitude.png', dpi=300, bbox_inches='t
ight')
plt.close(fig3)

# Velocity vectors
fig4 = plt.figure(figsize=(8, 6))
ax4 = fig4.add_subplot(111)
skip = 2
quiver = ax4.quiver(X[::skip, ::skip], Y[::skip, ::skip],
                   u[::skip, ::skip], v[::skip, ::skip],
                   vel_mag[::skip, ::skip],
                   scale=50, cmap='viridis')
plt.colorbar(quiver, ax=ax4, label='Velocity Magnitude')
```

```
ax4.set_title('Velocity Field')
ax4.set_xlabel('x')
ax4.set_ylabel('y')
plt.savefig('velocity_field.png', dpi=300, bbox_inches='tight')
plt.close(fig4)

print(f"Maximum velocity magnitude: {np.max(vel_mag):.4f}")
```

Results







Conclusion

- The velocity is highest near the source and sink due to steep gradients in the potential.
- Maximum velocity magnitude: 2.1044