# CFD-HW5

李张鑫 2200011085 工学院

## 一、 数理算法原理

### 1. 控制方程与边界条件

#### （1） 控制方程

不可压缩流动由 Navier-Stokes 方程控制：

$$\begin{cases} \dfrac{\partial\omega}{\partial t} + (v\cdot\nabla)\omega = \nu\Delta\omega \\ \Delta\psi = -\omega \\ u = \dfrac{\partial\psi}{\partial y}, v = -\dfrac{\partial\psi}{\partial x} \end{cases}$$

#### （2） 边界条件：

左边界($x = 0, 0 \le y \le 1$)：

$$u(0,y) = 0, v(0,y) = 0$$

右边界($x = 1, 0 \le y \le 1$)：

$$u(1,y) = 0, v(1,y) = 0$$

下边界($y = 0, 0 \le x \le 1$)：

$$u(x,0) = 0, v(x,0) = 0$$

上边界($y = 1, 0 \le x \le 1$)：

$$u(x,1) = \sin^2(\pi x), v(x,1) = 0$$

### 2. 数值求解方法

#### （1） 差分格式

$$\frac{\omega_{i,j}^{n+1} - \omega_{i,j}^{n}}{\Delta t} + \frac{\psi_{i,j+1}^{n} - \psi_{i,j-1}^{n}}{2h}\frac{\omega_{i+1,j}^{n} - \omega_{i-1,j}^{n}}{2h} - \frac{\psi_{i+1,j}^{n} - \psi_{i-1,j}^{n}}{2h}\frac{\omega_{i,j+1}^{n} - \omega_{i,j-1}^{n}}{2h} = \nu\frac{\omega_{i,j+1}^{n} + \omega_{i,j-1}^{n} + \omega_{i+1,j}^{n} + \omega_{i-1,j}^{n} - 4\omega_{i,j}^{n}}{h^2}$$

（2） SOR 迭代求解

$$\omega_{i,j}^{n+1} = \Delta t * \left( -\left(\frac{\psi_{i,j+1}^{n}-\psi_{i,j-1}^{n}}{2h}\right)\left(\frac{\omega_{i+1,j}^{n}-\omega_{i-1,j}^{n}}{2h}\right) + \left(\frac{\psi_{i+1,j}^{n}-\psi_{i-1,j}^{n}}{2h}\right)\left(\frac{\omega_{i,j+1}^{n}-\omega_{i,j-1}^{n}}{2h}\right) + \nu \frac{\omega_{i,j+1}^{n}+\omega_{i,j-1}^{n}+\omega_{i+1,j}^{n}+\omega_{i-1,j}^{n}-4\omega_{i,j}^{n}}{h^2} + \omega_{i,j}^{n} \right)$$

$$\nabla^2 \psi_{i,j} = -\omega_{i,j} \Longrightarrow \psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} - 4\psi_{i,j} = -h^2\omega_{i,j}$$

$$\psi_{i,j}^{(k+1)} = (1-\omega_{\text{SOR}})\psi_{i,j}^{(k)} + \frac{\omega_{\text{SOR}}}{4}\left[\psi_{i+1,j}^{(*)} + \psi_{i-1,j}^{(*)} + \psi_{i,j+1}^{(*)} + \psi_{i,j-1}^{(*)} + h^2\omega_{i,j}\right]$$

收敛条件： $\max_{1\le i\le n_x-2, 1\le j\le n_y-2} \left|\psi_{i,j}^{(k+1)} - \psi_{i,j}^{(k)}\right| < \varepsilon_{\text{SOR}}$

# 二、 程序实现

**文件结构 具体可见** https://github.com/NngLee/CFD_HW/HW5

main.py — 主程序与数值核心

StreamlinePlot.png — 自动生成的流场箭头图

StreamFunction.png —流函数可视化图

MiddleVelocityU.png —中心线剖面速度 x 方向

MiddleVelocityV.png —中心线剖面速度 y 方向

Voracity.png —涡量可视化图

**核心代码片段**

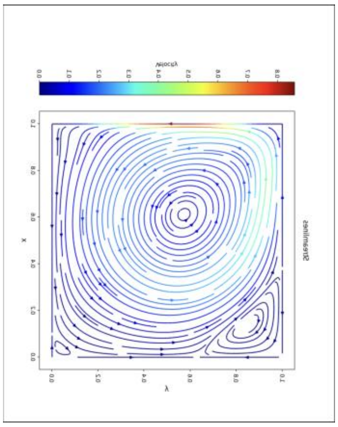
```
def SOR(f, omega, xsize, ysize, h, relax_factor):
    max_diff = 0.0
    iter_count = 0

    while iter_count < MAX_ITER:
        max_diff = 0.0
        for i in range(1, xsize-1):
            for j in range(1, ysize-1):
                old_value = f[i][j]
                new_value = (1 - relax_factor) * old_value + relax_factor * 0.25 * \
                            (f[i+1][j] + f[i][j+1] + f[i-1][j] + f[i][j-1] + h*h * omega[i][j])
                f[i][j] = new_value
                diff = abs(new_value - old_value)
                if diff > max_diff:
                    max_diff = diff

        iter_count += 1
        print(iter_count)
        if max_diff <= EPS:
            break

    return iter_count < MAX_ITER and iter_count > 1
```

# 三、 结果分析与讨论

本数值模拟旨在复现并对比文献《High-Re Solutions for Incompressible Navier–Stokes Equations Using the Multigrid Method》中 Re = 1000、129×129 网格条件下的实验结果。计算参数设定如下：网格间距 h = 0.01，SOR 迭代收敛判据 EPS = 10⁻5，时间推进步长 Δt = 0.01。对比分析将聚焦于流线图、涡量图、中心线速度剖面等核心流场特征。

**计算结果与基准解对比**

**A. 流线图**



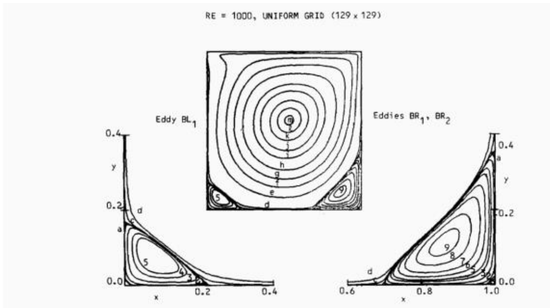
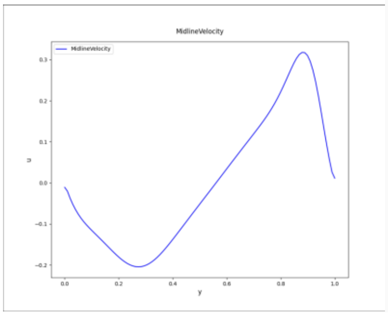
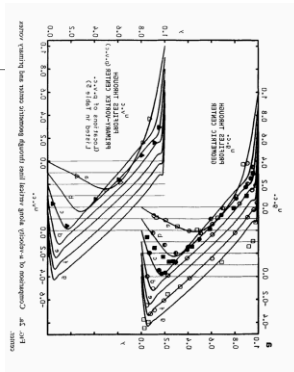


FIG. 3. Streamline pattern for primary, secondary, and additional corner vortices.

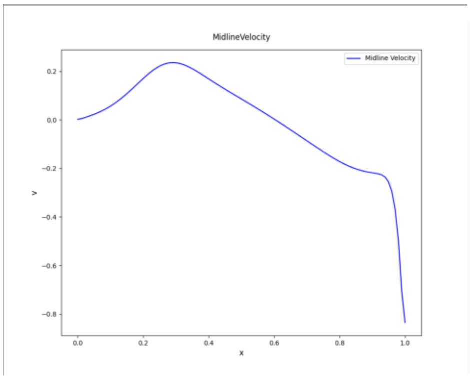
左图为数值模拟计算得到的流线图，右图为基准图，可以发现形状类似，与基准解比较吻合

**B. 中心线剖面速度**



可以发现数值模拟的图与基准图大致形状相同，在临近 y=1 时略微有些差异，可能是 eps 设置得不够充分小的原因



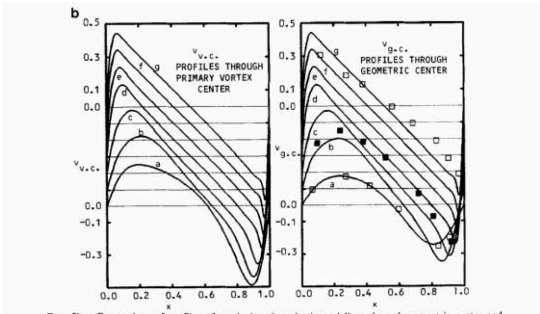
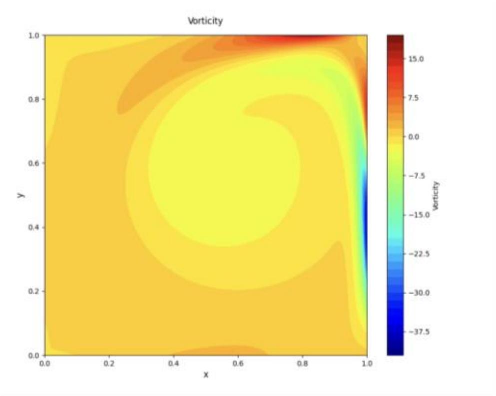
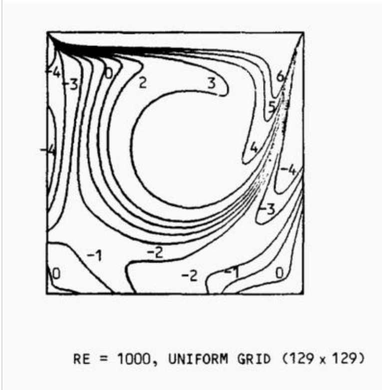




FIG. 2b. Comparison of profiles of v-velocity along horizontal lines through geometric center and primary vortex center.

可以发现数值模拟的图与基准图大致形状相同，在临近 x=1 时略微有些差异，可能是 eps 设置得不够充分小的原因

### c. 涡量图




涡量图形状与基准解比较类似,计算得到主涡位置 (0.61, 0.57)，对应流函数-0.074，文献中主涡位置位于（0.5313，0.5625），对应流函数最小值为-0.117929。存在一定程度的误差，可能是因为 EPS 选取不够小，导致收敛的精度较低

# 四、 AI 工具使用说明与 commit 记录

使用的 AI 工具名称：deepseek

核心算法部分比例:100%