

-
1. 用有限差分法发展一个程序，采取均匀分割， x 和 y 方向各分 20 份，用 Gauss-Seidel 迭代法数值求解正方形场域($0 \leq x \leq 1, 0 \leq y \leq 1$)的拉普拉斯方程（要求给出代码，并画出区域里的等高线）：

$$\begin{cases} \nabla^2 \varphi(x, y) = 0 \\ \varphi(x, 0) = \varphi(x, 1) = 0, \varphi(0, y) = \varphi(1, y) = 1 \end{cases}$$

Python 代码：

```
import numpy as np
import matplotlib.pyplot as plt

K = 10000
N = 20
elta = 1e-8

#给定初始值
pha = np.zeros((21, 21))
pha[:, 0] = 1
pha[:, N] = 1
pha[0, :] = 0
pha[N, :] = 0

for i in range(K):
    err = np.zeros((21, 21))
    for m in range(1, N):
        for n in range(1, N):
            a = pha[m, n]
            pha[m, n] = 1/4 * (pha[m + 1, n] + pha[m, n + 1] + pha[m - 1, n] + pha[m, n -
1])
            err[m, n] = abs(pha[m, n] - a)
    flag = np.max(err)
    if flag < elta:
        print("迭代已收敛")
        break

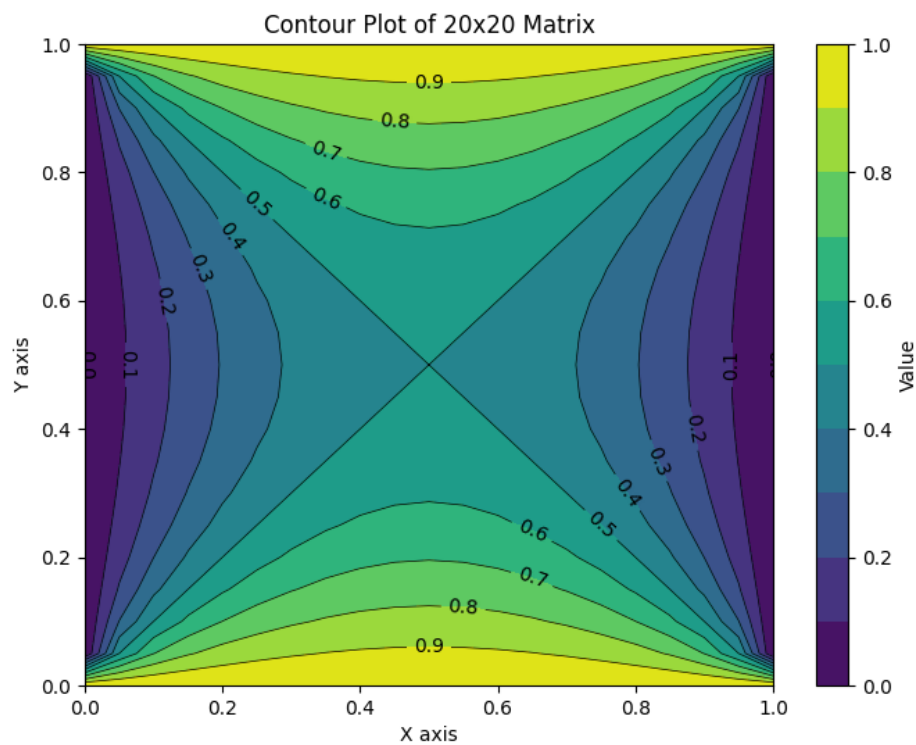
Pha = np.array(pha)
x = y = np.linspace(0, 1, 21)
X, Y = np.meshgrid(x, y)

plt.figure(figsize=(8, 6))
```

```
contour = plt.contourf(X, Y, Pha, levels=10, cmap='viridis') # levels 控制等高线层数
plt.colorbar(label='Value')
contour_lines = plt.contour(X, Y, Pha, levels=10, colors='black', linewidths=0.5) # 绘制等高线
plt.clabel(contour_lines, inline=True, fontsize=10, fmt="%.1f") # 添加标签, 设置字体大小和格式
```

```
plt.title("Contour Plot of 20x20 Matrix")
plt.xlabel("X axis")
plt.ylabel("Y axis")
```

```
plt.show()
```



MATLAB 代码:

```
clear;

% 网格划分
N = 20;

% 设置误差限
E = 1e-16;

% 设置初始值
phi = zeros(N + 1);
phi(:, 1) = ones(1, N + 1);
```

```
phi(:, N + 1) = ones(1, N + 1);
phi_k = ones(N + 1);

% 未收敛到误差限内则持续迭代
while norm(phi - phi_k) > E
    for i = 2 : N
        for j = 2 : N
            phi_k = phi;
            phi(i, j) = 1/4 * (phi_k(i + 1, j) + phi_k(i, j + 1) + phi(i
- 1, j) + phi(i, j - 1));
        end
    end
end

% 设置 x,y 坐标
X = zeros(N + 1);
Y = zeros(N + 1);
for i = 1 : 21
    X(i, :) = linspace(0, 1, N + 1);
    Y(:, i) = linspace(0, 1, N + 1);
end

% 绘制等高线图
contour(X, Y, phi, 20);
```

