Computing Method - Chapter8-1

Zhixin Zhang, 3210106357

Problems：1,2,4,5

1

用 Jacobi 迭代法，Gauss-Seidel 迭代法和取 $\omega = 1.46$ 的超松弛迭代法解方程组（误差为 0.5×10^{-4}）

$$\begin{cases} 2x_1 - x_2 = 1 \\ -x_1 + 2x_2 - x_3 = 0 \\ -x_2 + 2x_3 - x_4 = 1 \\ -x_3 + 2x_4 = 0 \end{cases}$$

并写出相应的迭代矩阵．若取最优松弛因子计算，结果如何？

解:

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

(1) Jacobi 迭代法：

$$T_j = -\begin{bmatrix} 0 & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & 0 \end{bmatrix}, \boldsymbol{c} = \left[\frac{1}{2}, 0, \frac{1}{2}, 0\right]^T, \boldsymbol{x}^{(0)} = [0,0,0,0]^T, \boldsymbol{x}^{(k+1)} = T_j \boldsymbol{x}^{(k)} + \boldsymbol{c}, k \in \mathbb{N}^+$$

```python
import numpy as np
eps = 5e-5

def iter(x0, T, c):
    x1 = T @ x0 + c
    k = 1
    while np.linalg.norm(x0 - x1, ord=np.inf) > eps:
        x0 = x1
        x1 = T @ x0 + c
        k = k + 1
    print("k = ", k)
    return x1

Tj = - np.array([[0,-1/2,0,0],[-1/2,0,-1/2,0],[0,-1/2,0,-1/2],[0,0,-1/2,0]])
c = np.array([1/2,0,1/2,0])
x0 = np.array([0,0,0,0])
c = c.transpose()
x0 = x0.transpose()

print(iter(x0, Tj, c))
```

求解结果为：$\boldsymbol{x}^{(46)} = [1.199938891.399920011.599901130.79995056]^T$.

(2) Gauss-Seidel 迭代法：

```
D_L = np.array([[2,0,0,0],[-1,2,0,0],[0,-1,2,0],[0,0,-1,2]])
U = -np.array([[0,-1,0,0],[0,0,-1,0],[0,0,0,-1],[0,0,0,0]])
T = np.linalg.inv(D_L) @ U
print ("T = ", T)
c = np.linalg.inv(D_L) @ np.array([1,0,1,0])
print("x = ", iter(x0, T, c))
```

$$T_{GS} = -\begin{bmatrix} 0 & 0.5 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0 \\ 0 & 0.125 & 0.25 & 0.5 \\ 0 & 0.0625 & 0.125 & 0.25 \end{bmatrix}, \boldsymbol{c} = [0.5, 0.25, 0.625, 0.3125]^T, \boldsymbol{x}^{(0)} = [0,0,0,0]^T, \boldsymbol{x}^{(k+1)} = T_j\boldsymbol{x}^{(k)} + \boldsymbol{c}, k \in \mathbb{N}^+$$

求解结果为：$\boldsymbol{x}^{(24)} = [1.199946961.399930571.599943830.79997191]^T$.

(3) SOR 迭代法：

```
U = -np.array([[0,-1,0,0],[0,0,-1,0],[0,0,0,-1],[0,0,0,0]])
L = -np.array([[0,0,0,0],[-1,0,0,0],[0,-1,0,0],[0,0,-1,0]])
D = np.array([[2,0,0,0],[0,2,0,0],[0,0,2,0],[0,0,0,2]])
b = np.array([1,0,1,0])
x0 = np.array([0,0,0,0])
w = 1.46

T = np.linalg.inv(D - w*L) @ (w * U + (1-w) * D)
c = w * np.linalg.inv(D - w * L) @ b

print ("T = ", T)
print ("c = ", c)
print("x = ", iter(x0, T, c))
```

$$T_{SOR} = \begin{bmatrix} -0.46 & 0.73 & 0 & 0 \\ -0.3358 & 0.0729 & 0.73 & 0 \\ -0.245134 & 0.053217 & 0.0729 & 0.73 \\ -0.17894782 & 0.03884841 & 0.053217 & 0.0729 \end{bmatrix}, c = [0.73, 0.5329, 1.119017, 0.81688241]^T$$

最优松弛因子：$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho(T_j)}^2} \approx 1.25961618$.

取 $\omega = 1.46$ 时，求解结果为：$\boldsymbol{x}^{15} = [1.19999113, 1.40000495, 1.60000927, 0.80000289]^T$;

取 $\omega = 1.26$ （最优松弛因子）时，求解结果为：$\boldsymbol{x}^{11} = [1.19998703, 1.39998917, 1.59999415, 0.79999811]^T$

$\square$

2

设有系数矩阵

$$A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

证明：(1) 对系数矩阵 A，Jacobi 迭代法收敛，而 Gauss-Seidel 迭代法不收敛.

(2) 对系数矩阵 B，Jacobi 迭代法不熟练，而 Gauss-Seidel 迭代法收敛.

证明: 通过以下代码计算 A, B 两个系数矩阵的谱半径：

```python
def getLDU(A, n):
    L = np.zeros(np.shape(A))
    D = np.zeros(np.shape(A))
    U = np.zeros(np.shape(A))

    for i in range(0, n):
        for j in range(0, n):
            if(i < j):
                U[i][j] = -A[i][j]
            if(i > j):
                L[i][j] = -A[i][j]
            if(i == j):
                D[i][j] = A[i][j]
    return [L,D,U]

def rho(A):
    return np.max(np.abs(np.linalg.eigvals(A)))

def Tj(A, n):
    [L, D, U] = getLDU(A, n)
    return np.linalg.inv(D) @ (L + U)

def TGS(A, n):
    [L, D, U] = getLDU(A, n)
    return np.linalg.inv(D - L) @ U

A = np.array([[1,2,-2],[1,1,1],[2,2,1]])
B = np.array([[2,-1,1],[1,1,1],[1,1,-2]])

print("rho(Tj(A)) = ", rho(Tj(A, 3)), ", rho(TGS(A)) = ", rho(TGS(A, 3)))
print("rho(Tj(B)) = ", rho(Tj(B, 3)), ", rho(TGS(B)) = ", rho(TGS(B, 3)))
```

输出：

```
rho(Tj(A)) =  3.759933925918774e-06 , rho(TGS(A)) =  2.0
rho(Tj(B)) =  1.1180339887498942 , rho(TGS(B)) =  0.5
```

(1) 对于 A，Jacobi 迭代矩阵的谱半径小于 1，而 GS 迭代矩阵的谱半径大于 1；
(2) 对于 B，Jacobi 迭代矩阵的谱半径大于 1，而 GS 迭代矩阵的谱半径小于 1；

根据定理 8.1，可以得到结论.

□

4

设有系数矩阵

$$A = \begin{bmatrix} a & 1 & 3 \\ 1 & a & 2 \\ -3 & 2 & a \end{bmatrix}$$

问：a 取什么值时，Jacobi 迭代法收敛？ a 取什么值时，Gauss-Seidel 迭代法收敛？$a = 3$ 时又怎样？并比较谱半径.

解:

$$T_j = \frac{1}{a}\begin{bmatrix} 0 & 1 & 3 \\ 1 & 0 & 2 \\ -3 & 2 & 0 \end{bmatrix} \Rightarrow \lambda_{1,2,3} = 0, \pm\frac{2}{a}$$

所以 $|a| > 2$ 时，Jacobi 迭代法收敛.

$$T_{GS} = \begin{bmatrix} 0 & -\frac{1}{a} & -\frac{3}{a} \\ 0 & \frac{1}{a^2} & \frac{3}{a^2} - \frac{2}{a} \\ 0 & -\frac{3}{a^2} - \frac{2}{a^3} & -\frac{5}{a^2} - \frac{6}{a^3} \end{bmatrix}$$

当 $a \in (-1.817, 2.862)$ 时，GS 迭代法收敛． $a = 3$ 时，使用 **2** 的代码计算得到两种迭代方法的谱半径分别为 0.6667 和 0.9107.

□

5

设 $A \in R^{n\times n}$，对称正定，其最小特征值和最大特征值分别是 λ_n, λ_1．试证迭代法

$$x^{k+1} = x^k + \alpha\left(b - Ax^k\right)$$

收敛的充分必要条件是 $0 < \alpha < 2/\lambda_1$． 又问参数取何值时迭代矩阵的谱半径最小？

证明:

迭代矩阵 a 为 $I - \alpha A$，所以迭代矩阵的谱半径为 $\max(|1-\alpha\lambda_1|, |1-\alpha\lambda_n|)$．令其小于 1，得到 $0 < \alpha < 2/\lambda_1$．由图像得， 当 $1 - \alpha\lambda_n = \alpha\lambda_1 - 1$，即 $\alpha = \frac{2}{\lambda_1+\lambda_n}$ 时，迭代矩阵的谱半径最小.

□