第二次作业

1929401206 丁誉洋

2022.5.8

题目 1: 分别用高斯消去法和高斯-若当消去法求解线性方程组

$$\begin{pmatrix} 1 & 4 & 2 \\ 1 & 5 & 2 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$

解:

(1) 高斯消去法:

$$(A\vdots b) = \begin{pmatrix} 1 & 4 & 2 & \vdots & 2 \\ 1 & 5 & 2 & \vdots & 3 \\ 0 & 1 & 1 & \vdots & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 4 & 2 & \vdots & 2 \\ 0 & 1 & 0 & \vdots & 1 \\ 0 & 1 & 1 & \vdots & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 4 & 2 & \vdots & 2 \\ 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 1 & \vdots & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 4 & 2 & \vdots & 2 \\ 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 1 & \vdots & 0 \end{pmatrix}$$

把原方程等价约化为:

$$\begin{cases} x_1 + 4x_2 + 2x_3 = 2 \\ x_2 + 0x_3 = 1 \\ x_3 = 0 \end{cases}$$

据之回代解得:

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

(2) 高斯-若当消去法

$$(A\vdots b) = \begin{pmatrix} 1 & 4 & 2 & \vdots & 2 \\ 1 & 5 & 2 & \vdots & 3 \\ 0 & 1 & 1 & \vdots & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 4 & 2 & \vdots & 2 \\ 0 & 1 & 0 & \vdots & 1 \\ 0 & 1 & 1 & \vdots & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 2 & \vdots & -2 \\ 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 1 & \vdots & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 & \vdots & -2 \\ 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 1 & \vdots & 0 \end{pmatrix}$$

所以，解为 $x = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}$

题目 2: 用选列主元的高斯消去法求解线性方程组

$$\begin{pmatrix} 1 & 2 & 1 & -2 \\ 2 & 5 & 3 & -2 \\ -2 & -2 & 3 & 5 \\ 1 & 3 & 2 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 7 \\ -1 \\ 0 \end{pmatrix}$$

解:

$$(A \vdots b) = \begin{pmatrix} 1 & 2 & 1 & -2 & \vdots & 4 \\ \mathbf{2} & 5 & 3 & -2 & \vdots & 7 \\ -2 & -2 & 3 & 5 & \vdots & -1 \\ 1 & 3 & 2 & -3 & \vdots & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 2.5 & 1.5 & -1 & \vdots & 3.5 \\ 0 & -0.5 & -0.5 & -1 & \vdots & 0.5 \\ 0 & \mathbf{3} & 6 & 3 & \vdots & 6 \\ 0 & 0.5 & 0.5 & -2 & \vdots & -3.5 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 2.5 & 1.5 & -1 & \vdots & 3.5 \\ 0 & 1 & 2 & 1 & \vdots & 2 \\ 0 & 0 & \mathbf{0.5} & -0.5 & \vdots & 1.5 \\ 0 & 0 & -0.5 & -2.5 & \vdots & -4.5 \end{pmatrix}$$

$$\longrightarrow \begin{pmatrix} 1 & 2.5 & 1.5 & -1 & \vdots & 3.5 \\ 0 & 1 & 2 & 1 & \vdots & 2 \\ 0 & 0 & 1 & -1 & \vdots & 3 \\ 0 & 0 & 0 & \mathbf{-3} & \vdots & -3 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 2.5 & 1.5 & -1 & \vdots & 3.5 \\ 0 & 1 & 2 & 1 & \vdots & 2 \\ 0 & 0 & 1 & -1 & \vdots & 3 \\ 0 & 0 & 0 & 1 & \vdots & 1 \end{pmatrix}$$

回代解得:

$$\begin{cases} x_1 = 16 \\ x_2 = -7 \\ x_3 = 4 \\ x_4 = 1 \end{cases}$$

题目 3: 用选全主元的高斯-若当消去法求解如下线性方程组

$$\begin{pmatrix} 2 & 1 & -2 \\ 3 & 1 & -4 \\ 1 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 6 \\ 0 \end{pmatrix}$$

解:

$$(A\vdots b)=\begin{pmatrix}2 & 1 & -2 & \vdots & 6\\ 3 & 1 & \textbf{-4} & \vdots & 6\\ 1 & -1 & 2 & \vdots & 0\end{pmatrix}\longrightarrow\begin{pmatrix}1 & -0.25 & -0.75 & \vdots & -1.5\\ 0 & 0.5 & 0.5 & \vdots & 3\\ 0 & -0.5 & \mathbf{2.5} & \vdots & 3\end{pmatrix}$$

$$\longrightarrow\begin{pmatrix}1 & 0 & -0.4 & \vdots & -0.6\\ 0 & 1 & -0.2 & \vdots & 1.2\\ 0 & 0 & \mathbf{0.6} & \vdots & 2.4\end{pmatrix}\longrightarrow\begin{pmatrix}1 & 0 & 0 & \vdots & 1\\ 0 & 1 & 0 & \vdots & 2\\ 0 & 0 & 1 & \vdots & 4\end{pmatrix}$$

所以解得

$$\begin{cases}x_1=\widetilde{x_2}=2\\ x_2=\widetilde{x_3}=4\\ x_3=\widetilde{x_1}=1\end{cases}$$

题目 4: 用克洛特分解法分解以下矩阵

$$(1)\ \mathbf{A}=\begin{pmatrix}1 & 2 & 1\\ 0 & 2 & 2\\ 2 & 4 & 5\end{pmatrix}\qquad(2)\ \begin{pmatrix}1 & -1 & 1\\ 5 & -4 & 3\\ 2 & 1 & 1\end{pmatrix}$$

解:

(1)

$$\begin{pmatrix}1 & 2 & 1\\ 0 & 2 & 2\\ 2 & 4 & 5\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1}\\ \mathbf{0} & 2 & 2\\ \mathbf{2} & 4 & 5\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1}\\ \mathbf{0} & \mathbf{2} & 2\\ \mathbf{2} & \mathbf{0} & 5\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1}\\ \mathbf{0} & \mathbf{2} & \mathbf{1}\\ \mathbf{2} & \mathbf{0} & 5\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1}\\ \mathbf{0} & \mathbf{2} & \mathbf{1}\\ \mathbf{2} & \mathbf{0} & \mathbf{3}\end{pmatrix}$$

解得:

$$L=\begin{pmatrix}1 & & \\ 0 & 2 & \\ 2 & 0 & 3\end{pmatrix},R=\begin{pmatrix}1 & 2 & 1\\ & 1 & 1\\ & & 1\end{pmatrix}$$

(2)

$$\begin{pmatrix}1 & -1 & 1\\ 5 & -4 & 3\\ 2 & 1 & 1\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \textbf{-1} & \mathbf{1}\\ \mathbf{5} & -4 & 3\\ \mathbf{2} & 1 & 1\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \textbf{-1} & \mathbf{1}\\ \mathbf{5} & \mathbf{1} & 3\\ \mathbf{2} & \mathbf{3} & 1\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \textbf{-1} & \mathbf{1}\\ \mathbf{5} & \mathbf{1} & \textbf{-2}\\ \mathbf{2} & \mathbf{3} & 1\end{pmatrix}\longrightarrow\begin{pmatrix}\mathbf{1} & \textbf{-1} & \mathbf{1}\\ \mathbf{5} & \mathbf{1} & \textbf{-2}\\ \mathbf{2} & \mathbf{3} & \mathbf{5}\end{pmatrix}$$

解得:

$$L=\begin{pmatrix}1 & & \\ 5 & 1 & \\ 2 & 3 & 5\end{pmatrix}, R=\begin{pmatrix}1 & -1 & 1 \\ & 1 & -2 \\ & & 1\end{pmatrix}$$

题目 5: 用克洛特分解法求解线性方程组

$$\begin{pmatrix}1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0\end{pmatrix}\begin{pmatrix}x_1 \\ x_2 \\ x_3\end{pmatrix}=\begin{pmatrix}0 \\ 3 \\ 2\end{pmatrix}$$

解:

对 A 进行克洛特分解

$$\begin{pmatrix}1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0\end{pmatrix} \longrightarrow \begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{2} & 2 & 3 \\ \mathbf{-1} & -3 & 0\end{pmatrix} \longrightarrow \begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{2} & \mathbf{-2} & 3 \\ \mathbf{-1} & \mathbf{-1} & 0\end{pmatrix} \longrightarrow \begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{2} & \mathbf{-2} & \mathbf{-0.5} \\ \mathbf{-1} & \mathbf{-1} & 0\end{pmatrix} \longrightarrow \begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{2} & \mathbf{-2} & \mathbf{-0.5} \\ \mathbf{-1} & \mathbf{-1} & \mathbf{0.5}\end{pmatrix}$$

所以

$$L=\begin{pmatrix}1 & & \\ 2 & -2 & \\ -1 & -1 & 0.5\end{pmatrix}, R=\begin{pmatrix}1 & 2 & 1 \\ & 1 & -0.5 \\ & & 1\end{pmatrix}$$

由:

$$\begin{pmatrix}1 & & \\ 2 & -2 & \\ -1 & -1 & 0.5\end{pmatrix}\begin{pmatrix}y_1 \\ y_2 \\ y_3\end{pmatrix}=\begin{pmatrix}0 \\ 3 \\ 2\end{pmatrix}$$

解得:

$$\begin{pmatrix}y_1 \\ y_2 \\ y_3\end{pmatrix}=\begin{pmatrix}0 \\ -1.5 \\ 1\end{pmatrix}$$

由:

$$\begin{pmatrix}1 & 2 & 1 \\ & 1 & -0.5 \\ & & 1\end{pmatrix}\begin{pmatrix}x_1 \\ x_2 \\ x_3\end{pmatrix}=\begin{pmatrix}0 \\ -1.5 \\ 1\end{pmatrix}$$

解得:

$$\begin{pmatrix}x_1 \\ x_2 \\ x_3\end{pmatrix}=\begin{pmatrix}1 \\ -1 \\ 1\end{pmatrix}$$

题目 6: 取初值 $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0.0$，分别用雅可比迭代法与高斯-塞德尔迭代法解线性方程组（精度要求为 $\epsilon = 10^{-3}$）

$$\begin{pmatrix} 3.0 & 0.15 & -0.09 \\ 0.08 & 4.0 & -0.16 \\ 0.05 & -0.3 & 5.0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6.09 \\ 11.52 \\ 19.20 \end{pmatrix}$$

解:

相应的迭代公式为

1. 雅可比迭代:

$$\begin{cases} x_1^{k+1} = 2.03 - 0.05x_2^k + 0.03x_3^k \\ x_2^{k+1} = 2.88 - 0.02x_1^k + 0.04x_3^k \\ x_3^{k+1} = 3.84 - 0.01x_1^k + 0.06x_2^k \end{cases}$$

迭代过程为

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.03 \\ 2.88 \\ 3.84 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0012 \\ 2.9930 \\ 3.9925 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0001 \\ 2.9997 \\ 3.9996 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0000 \\ 3.0000 \\ 4.0000 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0000 \\ 3.0000 \\ 4.0000 \end{pmatrix}$$

所以

$$\begin{pmatrix} x_1^4 \\ x_2^4 \\ x_3^4 \end{pmatrix} = \begin{pmatrix} 2.000 \\ 3.000 \\ 4.000 \end{pmatrix}$$

2. 高斯-塞德尔迭代:

$$\begin{cases} x_1^{k+1} = 2.03 - 0.05x_2^k + 0.03x_3^k \\ x_2^{k+1} = 2.88 - 0.02x_1^{k+1} + 0.04x_3^k \\ x_3^{k+1} = 3.84 - 0.01x_1^{k+1} + 0.06x_2^{k+1} \end{cases}$$

迭代过程为

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0300 \\ 2.8394 \\ 2.9901 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0077 \\ 2.9994 \\ 3.9999 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0000 \\ 3.0000 \\ 4.0000 \end{pmatrix} \longrightarrow \begin{pmatrix} 2.0000 \\ 3.0000 \\ 4.0000 \end{pmatrix}$$

所以

$$\begin{pmatrix} x_1^3 \\ x_2^3 \\ x_3^3 \end{pmatrix} = \begin{pmatrix} 2.000 \\ 3.000 \\ 4.000 \end{pmatrix}$$

题目 7: 利用定理 3.8 对以下线性方程组讨论雅可比迭代法与高斯-塞德尔迭代法的收敛性。

$$\begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

解:

$$M_J = D^{-1}(L+U) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{pmatrix}$$

特征值 $\lambda_1 = \lambda_2 = \lambda_3 = 0$，所以 $\rho(M) = \max\limits_{1 \le i \le n} |\lambda_i| = 0 < 1$，所以雅可比迭代法收敛

$$M_{G-S} = (D-L)^{-1}U = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 0 & -2 & 2 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{pmatrix}$$

特征值 $\lambda_1 = 0, \lambda_2 = \lambda_3 = 2$，所以 $\rho(M) = \max\limits_{1 \le i \le n} |\lambda_i| = 2 > 1$，所以高斯-塞德尔迭代法无法收敛

具体代码实现见下一页

此外，在作业的过程中，我将上述方法封装成了一个 `Matrix` 类，方便计算，以下是对高斯消元法，高斯-若当消元法（包括主元的选择方法），克洛特分解法，雅可比迭代法和高斯-塞德尔迭代法的 **C++** 实现，复杂度均为课件中所提到的复杂度

```cpp
#include<bits/stdc++.h>
using namespace std;

const int N = 5e2 + 5;
double mat[N][N];

struct Matrix {
    double mat[N][N];
    int n;

    void read(void) {
        scanf("%d", &n);
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n + 1; j++) {
                scanf("%lf", &mat[i][j]);
            }
        }
    }
    void printMat(double mat[N][N], int n, int m) {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                printf("%f ", mat[i][j]);
            }
            printf("\n");
        }
    }
    void printVec(double vec[], int n) {
        for (int i = 0; i < n; i++) {
            printf("%f ", vec[i]);
        }
        printf("\n");
    }

    // pivot_choice_method must be in ["none", "column", "all"]
    void GaussianElimination(bool GaussionJordan=true, string
        pivot_choice_method="none") {
```

```
    double ans[N];
    int row, col, pos[N];
    for (int i = 0; i < n; i++) {
        pos[i] = i;
    }
    for (row = 0, col = 0; row < n && col < n; row++, col++) {
        printf("Stage %d\n", row);
        if (pivot_choice_method == "column") {
            int maxrow = row;
            for (int i = row; i < n; i++) {
                if (fabs(mat[i][col]) > fabs(mat[maxrow][col])) {
                    maxrow = i;
                }
            }
            for (int j = 0; j < n + 1; j++){
                swap(mat[row][j], mat[maxrow][j]);
            }
        }
        else if (pivot_choice_method == "all") {
            int maxrow = row, maxcol = col;
            for (int i = row; i < n; i++) {
                for (int j = col; j < n; j++) {
                    if (fabs(mat[i][j]) > fabs(mat[maxrow][maxcol])) {
                        maxrow = i; maxcol = j;
                    }
                }
            }
            for (int j = 0; j < n + 1; j++) {
                swap(mat[row][j], mat[maxrow][j]);
            }
            for (int i = 0; i < n; i++) {
                swap(mat[i][col], mat[i][maxcol]);
            }
            swap(pos[col], pos[maxcol]);
        }
        if (mat[row][col] == 0) {
            printf("Fail!\n");
            return;
        }
```

```c
        double div = mat[row][col];
        for (int j = col; j < n + 1; j++) {
            mat[row][j] /= div;
        }
        for (int i = GaussionJordan? 0: row + 1; i < n; i++) {
            if (i == row) continue;
            double temp = mat[i][col];
            for (int j = col; j < n + 1; j++) {
                mat[i][j] -= mat[row][j] * temp;
            }
            mat[i][col] = 0;
        }
        printMat(mat, n, n + 1);
    }
    for (int i = n - 1; i >= 0; i--) {
        ans[i] = mat[i][n];
        for (int j = i + 1; j < n; j++) {
            ans[i] -= ans[j] * mat[i][j];
        }
    }
    printf("Result: \n");
    for (int i = 0; i < n; i++) {
        printf("ans%d = ans'%d = %f\n", pos[i], i, ans[i]);
    }
}
void CroutSplit(void) {
    double l[N][N], u[N][N], x[N], y[N];
    for (int i = 0; i < n; i++) {
        double sum;
        for (int j = 0; j <= i; j++) {
            sum = 0;
            for (int k = 0; k < j; k++) {
                sum += l[i][k] * u[k][j];
            }
            l[i][j] = mat[i][j] - sum;
        }
        for (int j = i + 1; j < n; j++) {
            sum = 0;
            for (int k = 0; k < i; k++) {
```

```
                    sum += l[i][k] * u[k][j];
                }
                u[i][j] = (mat[i][j] - sum) / l[i][i];
            }
            u[i][i] = 1;
        }
        printf("L = \n"); printMat(l, n, n);
        printf("U = \n"); printMat(u, n, n);
        for (int k = 0; k < n; k++) {
            double sum = 0;
            for (int i = 0; i < k; i++) {
                sum += l[k][i] * y[i];
            }
            y[k] = (mat[k][n] - sum) / l[k][k];
        }
        printf("y = \n"); printVec(y, n);
        for (int k = n - 1; k >= 0; k--) {
            double sum = 0;
            for (int i = k + 1; i < n; i++) {
                sum += u[k][i] * x[i];
            }
            x[k] = (y[k] - sum) / u[k][k];
        }
        printf("x = \n"); printVec(x, n);
    }

    void JacobiMethod(double x[], int iter) {
        double newx[N];
        for (int iter_num = 1; iter_num <= iter; iter_num++) {
            for (int i = 0; i < n; i++) {
                double sum = 0;
                for (int j = 0; j < n; j++) {
                    if (j == i) continue;
                    sum += mat[i][j] * x[j];
                }
                newx[i] = (mat[i][n] - sum) / mat[i][i];
            }
            printf("Iter %d: \n", iter_num);
            for (int i = 0; i < n; i++) {
```

```
                x[i] = newx[i];
            }
            printVec(x, n);
        }
    }

    void GaussSeidelMethod(double x[], int iter) {
        double newx[N];
        for (int iter_num = 1; iter_num <= iter; iter_num++) {
            for (int i = 0; i < n; i++) {
                double sum = 0;
                for (int j = 0; j < i; j++) {
                    sum += mat[i][j] * newx[j];
                }
                for (int j = i + 1; j < n; j++) {
                    sum += mat[i][j] * x[j];
                }
                newx[i] = (mat[i][n] - sum) / mat[i][i];
            }
            printf("Iter %d: \n", iter_num);
            for (int i = 0; i < n; i++) {
                x[i] = newx[i];
            }
            printVec(x, n);
        }
    }
}now;

int main(void) {
    now.read();
    now.GaussianElimination(true, "none");
    // now.CroutSplit();
    // double x[N] = {0, 0, 0};
    // now.JacobiMethod(x, 20)
    // now.GaussSeidelMethod(x, 20);
    return 0;
}
```