第 3 章　线性代数方程组的求解

3.1　求解实系数方程组的全选主元高斯消去法

【功能】

用全选主元高斯(Gauss)消去法求解 n 阶线性代数方程组 $\boldsymbol{AX}=\boldsymbol{B}$。其中

$$\boldsymbol{A}=\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{bmatrix},\quad \boldsymbol{X}=\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

【方法说明】

全选主元高斯消去法求解线性代数方程组的步骤如下。

(1) 对于 k 从 0 到 $n-2$ 做以下运算。

全选主元：$\max\limits_{k\leqslant i,j\leqslant n-1}\{|a_{ij}|\}$

通过行交换和列交换将绝对值最大的元素交换到主元素位置上。

系数矩阵归一化：$a_{kj}/a_{kk}\Rightarrow a_{kj},j=k+1,\cdots,n-1$

常数向量归一化：$b_k/a_{kk}\Rightarrow b_k$

系数矩阵消元：$a_{ij}-a_{ik}a_{kj}\Rightarrow a_{ij},i=k+1,\cdots,n-1;j=k+1,\cdots,n-1$

常数向量消元：$b_i-a_{ik}b_k\Rightarrow b_i,i=k+1,\cdots,n-1$

(2) 进行回代。

解出 x_{n-1}：$b_{n-1}/a_{n-1,n-1}\Rightarrow x_{n-1}$

回代逐个解出：$x_{n-2},\cdots,x_1,x_0$。即

$$b_k-\sum_{j=k+1}^{n-1}a_{kj}x_j\Rightarrow x_k,\quad k=n-2,\cdots,1,0$$

(3) 恢复解向量。

即对解向量中的元素顺序进行调整。

【数据成员与函数成员】

类名：gaus

数据成员	说　明
double a[n][n]	存放方程组的系数矩阵
double b[n]	存放方程组右端的常数向量。最后存放方程组的解向量
int n	方程组的阶数

主要函数成员	说　　明
void input()	从文件读入系数矩阵 ***A*** 以及常数向量 ***B***
void gauss()	执行全选主元 Gauss 消去法
void output()	输出解向量到文件并显示

【程序】(文件名为 3GAUS.CPP)

```
//3GAUS.CPP
//全选主元 Gauss 消去法求解实系数方程组
#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;
class  gaus
{
private:
        int n;
        double ** a, * b;
public:
        gaus (int nn)
        {
            int i;
            n=nn;
            a=new double * [n];        //动态分配内存空间
            for (i=0; i<n; i++) a[i]=new double[n];
            b=new double[n];
        }
        void input();                  //从文件读入系数矩阵 A 以及常数向量 B
        void gauss();                  //执行全选主元 Gauss 消去法
        void output();                 //输出结果到文件并显示
        ~gaus()
        {
            int i;
            for (i=0; i<n; i++) {delete[] a[i];}
            delete[] a;
            delete[] b;
        }
};

void gaus::input()                     //从文件读入系数矩阵 A 以及常数向量 B
{
    int i, j;
    char str1[20];
    cout <<"\n 输入文件名:";
    cin >>str1;
    ifstream  fin (str1);
```

```
    if (!fin)
    {cout <<"\n 不能打开这个文件 " <<str1 <<endl; exit(1);}
    for (i=0; i<n; i++)                          //读入矩阵 A
        for (j=0; j<n; j++)   fin >>a[i][j];
    for (i=0; i<n; i++)   fin >>b[i];            //读入常数向量 B
    fin.close();
}

void gaus::gauss()          //执行全选主元 Gauss 消去法
{
    int *js,l,k,i,j,is;
    double d,t;
    js=new int[n];
    l=1;
    for (k=0; k<=n-2; k++)
    {
        d=0.0;
        for (i=k;i<=n-1;i++)
        for (j=k;j<=n-1;j++)
        {
            t=fabs(a[i][j]);
            if (t>d) {d=t; js[k]=j; is=i;}
        }
        if (d+1.0==1.0) l=0;
        else
        {if (js[k]!=k)
            for (i=0;i<=n-1;i++)
            {
                t=a[i][k];
                a[i][k]=a[i][js[k]];
                a[i][js[k]]=t;
            }
            if (is!=k)
            {
                for (j=k;j<=n-1;j++)
                {
                    t=a[k][j];
                    a[k][j]=a[is][j];
                    a[is][j]=t;
                }
                t=b[k]; b[k]=b[is]; b[is]=t;
            }
        }
        if (l==0)
        {
            delete[] js;
```

```
            cout <<"\n 系数矩阵奇异!无解." <<endl;
            return;
        }
        d=a[k][k];
        for (j=k+1;j<=n-1;j++)
            a[k][j]=a[k][j]/d;
        b[k]=b[k]/d;
        for (i=k+1;i<=n-1;i++)
        {
            for (j=k+1;j<=n-1;j++)
                a[i][j]=a[i][j]-a[i][k] * a[k][j];
            b[i]=b[i]-a[i][k] * b[k];
        }
    }
    d=a[n-1][n-1];
    if (fabs(d)+1.0==1.0)
    {
        delete[] js;
        cout <<"\n 系数矩阵奇异!无解." <<endl;
        return;
    }
    b[n-1]=b[n-1]/d;
    for (i=n-2;i>=0;i--)
    {
        t=0.0;
        for (j=i+1;j<=n-1;j++)
            t=t+a[i][j] * b[j];
        b[i]=b[i]-t;
    }
    js[n-1]=n-1;
    for (k=n-1;k>=0;k--)
      if (js[k]!=k)
      {
          t=b[k]; b[k]=b[js[k]]; b[js[k]]=t;
      }
  delete[] js;
}

void gaus::output()                               //输出结果到文件并显示
{
    int i;
    char str2[20];
    cout <<"\n 输出文件名:";
    cin >>str2;
    ofstream fout (str2);
    if (!fout)
    {cout <<"\n 不能打开这个文件 " <<str2 <<endl; exit(1);}
    fout <<endl;  cout <<endl;
```

```
    for (i=0; i<n; i++)
    {
        fout <<b[i] <<"   ";
        cout <<b[i] <<"   ";
    }
    fout <<endl;  cout <<endl;
    fout.close();
}

void main()             //主函数
{
    gaus  c(4);
    c.input();          //从文件读系数矩阵 A 以及常数向量 B
    c.gauss();          //执行全选主元 Gauss 消去法
    c.output();         //输出结果到文件并显示
}
```

【例】

求解下列4阶方程组：

$$\begin{cases}0.2368x_0+0.2471x_1+0.2568x_2+1.2671x_3=1.8471\\0.1968x_0+0.2071x_1+1.2168x_2+0.2271x_3=1.7471\\0.1581x_0+1.1675x_1+0.1768x_2+0.1871x_3=1.6471\\1.1161x_0+0.1254x_1+0.1397x_2+0.1490x_3=1.5471\end{cases}$$

主函数见上述程序。

输入文件内容为(文件名为 gaus.txt)

```
0.2368  0.2471  0.2568  1.2671
0.1968  0.2071  1.2168  0.2271
0.1581  1.1675  0.1768  0.1871
1.1161  0.1254  0.1397  0.1490

1.8471  1.7471  1.6471  1.5471
```

输出文件内容为(文件名为 gaus.out)

```
1.04058   0.987051   0.93504   0.881282
```

3.2　求解实系数方程组的全选主元高斯-约当消去法

【功能】

用全选主元高斯-约当(Gauss-Jordan)消去法同时求解系数矩阵相同而右端具有 m 组常数向量的 n 阶线性代数方程组 $\boldsymbol{AX}=\boldsymbol{B}$。其中

$$\boldsymbol{A}=\begin{bmatrix}a_{00}&a_{01}&\cdots&a_{0,n-1}\\a_{10}&a_{11}&\cdots&a_{1,n-1}\\\vdots&\vdots&\ddots&\vdots\\a_{n-1,0}&a_{n-1,1}&\cdots&a_{n-1,n-1}\end{bmatrix},\quad\boldsymbol{X}=\begin{bmatrix}x_{00}&x_{01}&\cdots&x_{0,m-1}\\x_{10}&x_{11}&\cdots&x_{1,m-1}\\\vdots&\vdots&\ddots&\vdots\\x_{n-1,0}&x_{n-1,1}&\cdots&x_{n-1,m-1}\end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0,m-1} \\ b_{10} & b_{11} & \cdots & b_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1,0} & b_{n-1,1} & \cdots & b_{n-1,m-1} \end{bmatrix}$$

【方法说明】

全选主元高斯-约当消去法求解线性代数方程组的步骤如下。

(1) 对于 k 从 0 到 $n-1$ 做以下运算。

全选主元：$\max\limits_{k\leqslant i,j\leqslant n-1}\{|a_{ij}|\}$

通过行交换和列交换将绝对值最大的元素交换到主元素位置上。

系数矩阵归一化：$a_{kj}/a_{kk}\Rightarrow a_{kj},j=k+1,\cdots,n-1$

常数向量归一化：$b_{kj}/a_{kk}\Rightarrow b_{kj},j=0,1,\cdots,m-1$

系数矩阵消元：$a_{ij}-a_{ik}a_{kj}\Rightarrow a_{ij},i=0,\cdots,k-1,k+1,\cdots,n-1;j=k+1,\cdots,n-1$

常数向量消元：$b_i-a_{ik}b_k\Rightarrow b_i,i=0,\cdots,k-1,k+1,\cdots,n-1;j=0,1,\cdots,m-1$

(2) 恢复解向量。

即对解向量中的元素顺序进行调整。

【数据成员与函数成员】

类名：gjdn

数据成员	说明
double a[n][n]	存放方程组的系数矩阵
double b[n][m]	存放方程组右端的 m 组常数向量。最后存放方程组的 m 组解向量
int n	方程组的阶数
int m	方程组右端常数向量的组数

主要函数成员	说明
void input()	从文件读入系数矩阵 $\boldsymbol{A}$ 以及 m 组常数向量 $\boldsymbol{B}$
void gauss_jordan()	执行 Gauss-Jordan 消去法
void output()	输出解向量到文件并显示

【程序】(文件名为 3GJDN.CPP)

```
//3GJDN.CPP
//全选主元 Gauss-Jordan 消去法求解实系数方程组
#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;
class  gjdn
{
private:
         int n, m;
         double ** a, ** b;
public:
```

```
        gjdn (int nn, int mm)
        {
            int i;
            n=nn; m=mm;
            a=new double * [n];        //动态分配内存空间
            for (i=0; i<n; i++) a[i]=new double[n];
            b=new double * [n];
            for (i=0; i<n; i++) b[i]=new double[m];
        }
        void input();                  //从文件读入系数矩阵 A 以及 m 组常数向量 B
        void gauss_jordan();           //执行 Gauss-Jordan 消去法
        void output();                 //输出结果到文件并显示
        ~gjdn()
        {
            int i;
            for (i=0; i<m; i++) {delete[] a[i];}
            delete[] a;
            delete[] b;
        }
};

void gjdn::input()                     //从文件读入系数矩阵 A 以及 m 组常数向量 B
{
    int i, j;
    char str1[20];
    cout <<"\n 输入文件名:";
    cin >>str1;
    ifstream  fin (str1);
    if (!fin)
    {cout <<"\n 不能打开这个文件 " <<str1 <<endl; exit(1);}
    for (i=0; i<n; i++)                         //读入系数矩阵 A
        for (j=0; j<n; j++)  fin >>a[i][j];
    for (i=0; i<m; i++)                         //读入 m 组常数向量 B
    for (j=0; j<n; j++)  fin >>b[j][i];
    fin.close();
}

void gjdn::gauss_jordan()              //执行 Gauss-Jordan 消去法
{
    int *js,l,k,i,j,is;
    double d,t;
    js=new int[n];
    l=1;
    for (k=0;k<=n-1;k++)
    {
        d=0.0;
```

```
        for (i=k;i<=n-1;i++)
        for (j=k;j<=n-1;j++)
        {
            t=fabs(a[i][j]);
            if (t>d) {d=t; js[k]=j; is=i;}
        }
        if (d+1.0==1.0) l=0;
        else
        {
            if (js[k]!=k)
              for (i=0;i<=n-1;i++)
              {
                  t=a[i][k];
                  a[i][k]=a[i][js[k]];
                  a[i][js[k]]=t;
              }
            if (is!=k)
            {
                for (j=k;j<=n-1;j++)
                {
                    t=a[k][j];
                    a[k][j]=a[is][j];
                    a[is][j]=t;
                }
                for (j=0;j<=m-1;j++)
                {
                    t=b[k][j];
                    b[k][j]=b[is][j];
                    b[is][j]=t;
                }
            }
        }
        if (l==0)
        {
            delete[]js;
            cout <<"\n 系数矩阵奇异!无解. " <<endl;
            return;
        }
        d=a[k][k];
        for (j=k+1;j<=n-1;j++)
           a[k][j]=a[k][j]/d;
        for (j=0;j<=m-1;j++)
           b[k][j]=b[k][j]/d;
        for (j=k+ 1;j<=n-1;j++)
        for (i=0;i<=n-1;i++)
        {
```

```
            if (i!=k)
              a[i][j]=a[i][j]-a[i][k] * a[k][j];
        }
        for (j=0;j<=m-1;j++)
        for (i=0;i<=n-1;i++)
        {
            if (i!=k)
            b[i][j]=b[i][j]-a[i][k] * b[k][j];
        }
    }
    for (k=n-1;k>=0;k--)
      if (js[k]!=k)
        for (j=0;j<=m-1;j++)
        {
            t=b[k][j]; b[k][j]=b[js[k]][j]; b[js[k]][j]=t;
        }
    delete[] js;
}

void gjdn::output()                    //输出结果到文件并显示
{
    int i, j;
    char str2[20];
    cout <<"\n输出文件名:";
    cin >>str2;
    ofstream fout (str2);
    if (!fout)
    {cout <<"\n不能打开这个文件 " <<str2 <<endl; exit(1);}
    fout <<endl;  cout <<endl;
    for (i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
        {

            fout <<b[j][i] <<"   ";
            cout <<b[j][i] <<"   ";
        }
        fout <<endl;  cout <<endl;
    }
    fout <<endl;  cout <<endl;
    fout.close();
}

void main()                  //主函数
{
    gjdn  c(4, 2);
```

```
    c.input();            //从文件读入系数矩阵A以及m组常数向量B
    c.gauss_jordan();     //执行Gauss-Jordan消去法
    c.output();           //输出结果到文件并显示
}
```

【例】

求解下列 4 阶方程组 $\boldsymbol{AX}=\boldsymbol{B}$。其中

$$\boldsymbol{A}=\begin{bmatrix}1 & 3 & 2 & 13\\ 7 & 2 & 1 & -2\\ 9 & 15 & 3 & -2\\ -2 & -2 & 11 & 5\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}9 & 0\\ 6 & 4\\ 11 & 7\\ -2 & -1\end{bmatrix}$$

主函数见上述程序。

输入文件内容为(文件名为 gjdn.txt)

```
 1      3      2     13
 7      2      1     -2
 9     15      3     -2
-2     -2     11      5

 9      6     11     -2
 0      4      7     -1
```

输出文件内容为(文件名为 gjdn.out)

```
0.980745     0.267982     -0.222629     0.589274
0.497931     0.144494     0.0628581     -0.0813176
```

3.3 求解复系数方程组的全选主元高斯消去法

【功能】

用全选主元高斯(Gauss)消去法求解 n 阶复系数线性代数方程组 $\boldsymbol{AX}=\boldsymbol{B}$。其中

$$\boldsymbol{A}=\boldsymbol{AR}+\mathrm{j}\boldsymbol{AI}$$
$$\boldsymbol{B}=\boldsymbol{BR}+\mathrm{j}\boldsymbol{BI}$$
$$\boldsymbol{X}=\boldsymbol{XR}+\mathrm{j}\boldsymbol{XI}$$

【方法说明】

高斯消去法同 3.1 节的方法说明。

在本函数中要用到复数乘法和复数除法。

复数乘法采用如下算法：

设

$$e+\mathrm{j}f=(a+\mathrm{j}b)(c+\mathrm{j}d)$$

令

$$p=ac,\quad q=bd,\quad s=(a+b)(c+d)$$

则有

$$e=p-q,\quad f=s-p-q$$