数值分析作业1

一. 设计方案与源代码

本题**第一问**要求求出最小的的特征值λ_1，最大的特征值λ_{501}与绝对值最小的特征值$|\lambda_s|$。不需要求解全部特征值，故无需采用QR方法求解全部特征值。采用幂法求解。

幂法用于求解矩阵按照模最大的特征值$|\lambda_{max}^{'}|$，反幂法求解模最小的特征值$|\lambda_s|$,由于$\lambda_1 < \lambda_2 < \cdots < \lambda_{501}$，且$|\lambda_{max}^{'}| = max|\lambda_i|$。

$$\begin{cases} \lambda_1 > 0, \lambda_{501} > 0 \to \lambda_1 = \lambda_{min} \\ \lambda_1 < 0, \lambda_{501} < 0 \to \lambda_{501} = \lambda_{min} \\ \lambda_{min} \times \lambda_{max} < 0 \to \lambda_{max} = \lambda_1 or \lambda_{501} \end{cases} \tag{1}$$

由以上讨论可知，在最大特征值与最小特征值符号不同时，通过幂法仅能确定其中的一个值，为求出另一个值，类似于求解矩阵A的每个特征值使用的带原点平移反幂法，使用**带原点平移的幂法。**

$$\begin{aligned} Ax &= \lambda x \\ (A - pI)x &= (\lambda - p)x \end{aligned} \tag{2}$$

在程序设计中，为节约存储空间，矩阵A的非0元素不予存储，同时考虑到该带状矩阵除对角线以外元素的值为定值b 或c, 所以**仅使用一个一维数组**$a[i]$存储对角线元素。

为降低舍入误差，提高精确度，在幂法迭代中使用无穷范数$||_\infty$对迭代向量进行归一化,其迭代格式为:

$$\begin{cases} u_0 = (h_1^0, \cdots, h_n^0)^T (\text{非零向量}) \\ |h_r^{(k-1)}| = \max_{1 \leqslant j \leqslant n} |h_j^{k-1}| \\ y_{k-1} = u_{k-1}/|h_r^{k-1}| \\ u_k = Ay_k = (h_1^k, \cdots, h_n^k)^T \\ \beta_k = sgn(h_r^{k-1})h_r^k \end{cases} \tag{3}$$
$$k = 1, 2, \cdots$$

迭代终止格式为$|\beta_k - \beta_{k-1}|/|\beta_k| \leqslant \varepsilon$, $\lambda_{max} \approx \beta_k$。

```
double power(double a[]){
    int N = 5000;
    double b = 0.16, c = -0.064;
    double m = 1, beta;
    double u[501], y[501];
    for (int i = 0; i < 501; i++) {
        u[i] = 1;
    }
    j = 0;
    while (j<N){
        for (int i = 0; i < 501; i++) {
            y[i] = u[i]/fabs(m);
        }
        for (int i = 2;i<499;i++){
            u[i] = c*y[i-2]+b*y[i-1]+a[i]*y[i]+b*y[i+1]+c*y[i+2];
        }
        u[0]=a[0]*y[0]+b*y[1]+c*y[2];
        u[1]=b*y[0]+a[1]*y[1]+b*y[2]+c*y[3];
        u[499]=c*y[497]+b*y[498]+a[499]*y[499]+b*y[500];//边界处理
        u[500]=c*y[498]+b*y[499]+a[500]*y[500];
        beta = 0;
```

```
        for (int i = 0;i<501;i++){
            if (fabs(u[i])>=fabs(beta))
                beta = u[i];
        }
        if(fabs(beta-m)/fabs(beta)<ep)  //收敛判定
            break;
        //if(beta<0)
        else if(fabs(fabs(beta)-fabs(m))/fabs(beta)<ep)
                break;
        m = beta;
        j++;
    }
    return beta;
}
```

取位移量$p=\lambda_{max}$, 令$B=A-pI$, 对B应用幂法，得到λ_b, $\lambda_a=p+\lambda_b$。λ_a即为λ_1与λ_{501}之间的未确定值。模最小的特征值$|\lambda_s|$则通过反幂法求解。

```
double inv_power(double a[]) //反幂法
{
    double u[501],y[501]; //LU
    double beta,m=1;
    int i,j,N=1000;
    LUde(a);
    for(i=0;i<501;i++)
        u[i]=1;
    j=0;
    while(j<N)
    {
        for(i=0;i<501;i++)
        {
            y[i]=u[i]/fabs(m);
        }
        u[0]=y[0]/p[0];
        u[1]=(y[1]-l1[0]*u[0])/p[1];
        for(i=2;i<501;i++)
            u[i]=(y[i]-c*u[i-2]-l1[i-1]*u[i-1])/p[i];// 回代求解
        u[499]=u[499]-up1[499]*u[500];
        for(i=498;i>=0;i--)
            u[i]=u[i]-up1[i]*u[i+1]-up2[i]*u[i+2];

        beta=0;
        for(i=0;i<501;i++)
        {
            if(fabs(u[i])>=fabs(beta))
                beta=u[i];
        }
        if(beta<0)
            if(fabs(fabs(beta)-fabs(m))/fabs(beta)<ep)
                break;
        else if(fabs(beta-m)/fabs(beta)<ep)
            break;
        m=beta;
        j++;
    }
    return 1/beta;
}
```

第二问使用带原点平移的反幂法，取位移量$\mu_k=\lambda_1+k\frac{\lambda_1-\lambda_{501}}{40},(k=1,2,3,\cdots,39)$。即可求出矩阵$A$与$\mu_k$最接近的特征值$\lambda_{ik}$。反幂法迭代格式为:

$$\begin{cases} u_0=(h_1^0,\cdots,h_n^0)^T \\ |h_r^{k-1}|=\max_{1\leqslant j\leqslant n}|h_j^{k-1}| \\ y_{k-1}=u_{k-1}/|h_r^{k-1}| \\ Au_k=y_{k-1},u_k=(h_1^k,\cdots,h_n^k)^T \\ \beta_k=sgn(h_r^{k-1})h_r^k \end{cases} \\ k=(1,2,\cdots) \tag{4}$$

迭代终止格式为$|\beta_k-\beta_{k-1}|/|\beta_k|\leqslant\varepsilon$, $\lambda_{min}\approx1/\beta_k$。

因$Au_k=y_{k-1}$, u_k需要求解方程组获得，**将A进行$crout$分解**，相当于求解两个三角形方程组

$$\begin{matrix} Lx_k=y_k-1 \\ Uu_k=x_k \end{matrix} \tag{5}$$

在LU分解中，首先全局定义存储分解后对角线元素的数组p，$l1$为存储LU分解后$l[i+1][i]$元素，$l[i+2][i]$在LU分解后仍然全部等于c，$up1$为存储LU分解后$u[i][i+1]$元素，$up2$继续存储$u[i][i+2]$元素，$i=0,1,\cdots,501$。

```
double p[501],l1[500],up1[500],up2[499];
void LUde(double a[]){
    p[0]=a[0];
    l1[0]=b;
    up1[0]=b/p[0];
    up2[0]=c/p[0];
    p[1]=a[1]-l1[0]*up1[0];
    up2[1]=c/p[1];
    up1[1]=(b-l1[0]*up2[0])/p[1];
    l1[1] = b - c*up1[0];
    for(int i=2;i<501;i++)
    {
        l1[i]=b-c*up1[i-1];
        p[i]=a[i]-c*up2[i-2]-l1[i-1]*up1[i-1];
        up2[i]=c/p[i];
        up1[i]=(b-l1[i-1]*up2[i-1])/p[i];
    }
}
```

第三问需求解行列式$det A$与条件数$cond(A)_2$,

同样通过对矩阵A进行LU分解，乘以分解后对角线上元素获得矩阵行列式值，同时A为非奇异实对称矩阵，有

$$cond(A)_2=|\frac{\lambda_1}{\lambda_n}| \tag{6}$$

λ_1与λ_n为矩阵A最大模的特征值与最小模的特征值，于本题中，则为$\frac{\lambda_1}{\lambda_s}$。

```
double det(double a[]) //求det
{
    double det_A=1;
    LUde(a);
    for(int i=0;i<501;i++)
        det_A=det_A*p[i];
    return det_A;
}
```

二. 计算结果

1. $\lambda_1 = -1.070011361488e+01$
   $\lambda_{501} = 9.724634101479e+00$
   $\lambda_s = -5.557910794214e-03$
2. λi1 = -1.018293403315e+01 λi2 = -9.585707425068e+00 λi3 = -9.172672423928e+00
   λi4 = -8.652284007898e+00 λi5 = -8.093483808675e+00 λi6 = -7.659405407692e+00
   λi7 = -7.119684648691e+00 λi8 = -6.611764339397e+00 λi9 = -6.066103226595e+00
   λi10 = -5.585101052628e+00 λi11 = -5.114083529812e+00 λi12 = -4.578872176865e+00
   λi13 = -4.097829307903e+00 λi14 = -3.554211215751e+00 λi15 = -3.041090018133e+00
   λi16 = -2.526430459138e+00 λi17 = -2.003230769564e+00 λi18 = -1.503557611227e+00
   λi19 = -9.935586060075e-01 λi20 = -4.870426738850e-01 λi21 = 2.231736249575e-02
   λi22 = 5.324174742069e-01 λi23 = 1.052898962693e+00 λi24 = 1.589445881881e+00
   λi25 = 2.060330460274e+00 λi26 = 2.558075597073e+00 λi27 = 3.080240509307e+00
   λi28 = 3.580798705450e+00 λi29 = 4.091378510451e+00 λi30 = 4.603035378279e+00
   λi31 = 5.132924283898e+00 λi32 = 5.594906348083e+00 λi33 = 6.080933857027e+00
   λi34 = 6.680354092112e+00 λi35 = 7.293877448126e+00 λi36 = 7.717111714236e+00
   λi37 = 8.225220014050e+00 λi38 = 8.648666065194e+00 λi39 = 9.254200344575e+00
3. 条件数 $cond(A)_2 = 1.925204273883e+03$
   行列式$detA = 2.772786141752e+118$

三. 讨论分析

将计算结果与课后计算参考答案进行对比，发现结果之间的匹配度较高，误差小于千分之一，模最小的特征值$|\lambda_s|$与参考答案之间的误差相比λ_1与λ_{501}较大，推测原因为真实值较小的情况下，计算机产生的**舍入误差影响更为显著**。可以认为编写的计算程序比较可靠，完成了以数值方式该实对称矩阵特征值，条件数与行列式的任务。

在求解行列式函数的编写过程中，尝试过求解全部的特征值，将全部特征值相乘来获取行列式的方法来获得行列式，但是由于事先不能确定每一个特征值的具体范围，应用带原点平移的反幂法求特征值时特征值的重数很容易有误，使得求$detA$的准确性难以保证。

同时注意到编写的幂法，反幂法计算函数通用性较差，仅适合于题目中501×501大小，带内元素已经确定的带状矩阵，可以通过**面向对象编程**(OOP)的方式，引入C++中的$Vector$容器，通过初始化一个带状矩阵对象，对其元素进行赋值，从而提高幂法，反幂法计算函数的通用化程度，使其可以适用于任意尺寸大小，任意矩阵元素数值的带状矩阵特征值，条件数，行列式求解。将其适用于更为多样和实际的任务。