课程名称：数值代数 **指导教师**：刘兰冬

姓名：袁天乐 王哲 贾琪

第十组 林佳怡 李飞雨 刘京 **签字**：

实验项目名称：

线性方程组古典迭代法——Jacobi 迭代法

实验目的及要求：

了解使用 Jacobi 迭代法求解方程组的原理和相关步骤,根据算法编写程序去求解一个具体的方程组的解，再对计算解和准确解进行比较,观察准确程度。

实验原理：

①Jacobi 迭代法定义：

考虑非奇异线性代数方程组 Ax=b。
令 A=D-L-U,

其中 $A=[a_{ij}], D=diag(a_{11},a_{22},\cdots a_{nn})$；

$$L=\begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & -a_{n-1,n} & 0 \end{bmatrix};$$

$$U=\begin{bmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & 0 & -a_{n-1,n} \\ & & & & 0 \end{bmatrix};$$

那么 Ax=b 可以写成 x=Bx+g，其中 $B=D^{-1}(L+U)$，$g=D^{-1}b$，若给定初始向量 $x_0=(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)})^T$，并带入 x=Bx+g 式子的右边，就可以计算出一个新的向量 x_1；

即 $x_1=Bx_0+g$；

再把 x_1 带入 x=Bx+g 式子的右边，又可以得到一个向量 x_2；依次类推，有：

$x_k = Bx_{k-1} + g, k = 1,2,\cdots$。

这就是所谓的 Jacobi 迭代法，其中 B 叫做 Jacobi 迭代法的迭代矩阵，g 叫做 Jacobi 迭代法的常数项。

②Jacobi 迭代法编程算法：

设 N 为最大迭代次数，Tol 为允许误差。

- 步骤 1：置 $k \Leftarrow 1$，
- 步骤 2：$k \le N$，做如下：

  （2.1）对于 i=1,2，...，n 做：

  $$y_i = (b_i - \sum_{\substack{j=1 \\ j \ne i}}^{n} a_{ij} x_j) / a_{ii}$$

  （2.2）如果 $\left\| \vec{y} - \vec{x} \right\| = \max_{1 \le i \le n} \left\| y_i - x_i \right\| < Tol$

  则输出 $\vec{y} = (y_1, y_2, \cdots, y_n)$，停机。

  (2.3) $k \Leftarrow k + 1$

  （2.4）更新：对 i=1,2，...，n 做：

  $$x_i \Leftarrow y_i$$

- 步骤 3：输出最大迭代次数 N。

实验内容（方法和步骤）：

问题：

1. 利用 Jacobi 迭代法求解下列方程组：

$$\begin{pmatrix} 15 & 4 & 7 \\ 2 & 15 & 8 \\ 3 & 6 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 26 \\ 25 \\ 19 \end{pmatrix}$$，精确解为 $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

2. 用 Jacobi 迭代法求解下列方程组.对计算解和准确解比较,观察准确程度。第一方程组和第二个方程组的解 差别大吗？（用向量范数刻画）

(1) $$\begin{pmatrix} 4 & 1 & -1 & 0 \\ 1 & 3 & -1 & 0 \\ -1 & -1 & 5 & 2 \\ 0 & 0 & 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 7 \\ 8 \\ -4 \\ 6 \end{pmatrix}$$

(2) $\begin{pmatrix} 4 & 1.002 & -0.998 & 0 \\ 1.002 & 3 & -1 & 0 \\ -1 & -0.998 & 5.001 & 1.998 \\ 0 & 0 & 1.998 & 3.998 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 7 \\ 8.002 \\ -4.001 \\ 6.003 \end{pmatrix}$

解：根据上述原理和算法可编写如下 Jacobi 迭代法程序：
(以下程序是基于给定误差作为 while 循环判定条件)

①主程序：

```
function [y,k]=jff(a,b,e)
    digits(6);                  % 规定有效数字位数
    k=1;
    n=length(b);
    for i=1:n
        x(i)=0;
    end
    y=diedai(x,a,b);            % 调用函数 diedai
    while yifanshu(y-x)>e    % 调用函数 yifanshu(可以换成 erfanshu,wuqiongfan)
        y=diedai(x,a,b);
        x=y;
        y=diedai(x,a,b);
        k=k+1;
        y=vpa(y);
End
wucha=yifanshu(y-x)
                        % 误差 yifanshu(y-x)(根据所调用函数换成 erfanshu(y-x),wuqiongfan(y-x))
```

②主函数中所调用的函数：

迭代函数：

```
function y=diedai(x,a,b)
n=length(x);
for i=1:n
        sum=0;
        for j=1:n
            if j~=i
                sum=sum+a(i,j)*x(j);
            end
```

```
        end
        y(i)=(b(i)-sum)/a(i,i);
    end
```

1−范数：

```
function y=yifanshu(x)
n=length(x);
sum=0;
for i=1:n
    sum=sum+abs(x(i));
end
y=sum;
```

2−范数：

```
function y=erfanshu(x)
n=length(x);
sum=0;
for i=1:n
    sum=sum+(abs(x(i)))^2;
end
y=sqrt(sum);
```

∞−范数：

```
function y=wuqiongfan(x)
n=length(x);
for i=1:n
    x(i)=abs(x(i));
end
y=max(x);
```

实验结果与分析：

问题 1：

（1）在命令行窗口输入：

a=[15,4,7;2,15,8;3,6,10]

b=[26;25;19]

e=10^(-5)

并调用主程序：

[y,k]=jff(a,b,e)

得到结果如下表：

	1-范数	2-范数	∞-范
误差	0.00000862526	0.00000833587	0.00000878644
最大迭代次数	53	51	49
方程组的解	[1.0;1.0;1.0]	[1.0;1.0;1.0]	[1.0;1.0;1.0]

问题 2:

（1）在命令行窗口输入：

a=[4,1,-1,0;1,3,-1,0;-1,-1,5,2;0,0,2,4]

b=[7;8;-4;6]

e=10^(-5)

并调用主程序：

[y,k]=jff(a,b,e)

得到结果如下表：

	1-范数	2-范数	∞-范
误差	0.00000852798	0.00000669146	0.00000944752
最大迭代次数	30	29	27
方程组的解	[0.999999;2.0; -0.999999;2.0]	[1.0;2.0;-1.0;2.0]	[1.0;2.0;-1.0;2.0]

（2）在命令行窗口输入：

a=[4,1.002,-0.998,0;1.002,3,-1,0;-1,-0.998,5.001,1.998;0,0,1.998,3.998]

b=[7;8.002;-4.001;6.003]

e=10^(-5)

并调用主程序：

[y,k]=jff(a,b,e)

得到结果如下表：

	1-范数	2-范数	∞-范
误差	0.00000842389	0.00000661277	0.0000093379
最大迭代次数	30	29	27
方程组的计算解	[0.999308;1.99996;-1.00081;2.00165]	[0.999309;1.99996;-1.00081;2.00165]	[0.999311;1.99997;-1.00081;2.00165]

对比（1）与（2）方程组的解：

	1-范数	2-范数	∞-范
方程组(1)的解 y(1)	[0.999999;2.0;-0.999999;2.0]	[1.0;2.0;-1.0;2.0]	[1.0;2.0;-1.0;2.0]
方程组(2)的解 y(2)	[0.999308;1.99996;-1.00081;2.00165]	[0.999309;1.99996;-1.00081;2.00165]	[0.999311;1.99997;-1.00081;2.00165]
两个解的 2-范数之差	0.001	0.0001	0.0001

结论：两个解差别不大。

实验感想：

这次编程,我们小组在代码的编写和安排上,采用的是模块化编程的思想,其中 diedai()代表的是 Jacobi 迭代函数程序；对于误差的处理上，我们采用保留 6 位小数；精确解和数值解的误差大小我们给了 yifanshu()、erfanshu()、wuqiongfan()三个处理误差的函数分别代表求两者的 1-范数、2-范数和∞-范数处理，得出误差结果。

我们组的方程不是病态的方程，因此可以求出近似解。我们对比了一下 while 循环的判定条件（1-范数，2-范数，及∞-范）对于结果和迭代次数的影响，发现 1-范数在时间复杂度上比其他两种好但是结果的误差和迭代次数大，2-范数和∞-结果准确程度相比，2-范数更准确，但是迭代次数上无穷范少。由于数据不是很大，很难判定这三种方法的优劣。

注：三种范数算出的 11 位小数误差均是高位小数的精确结果。原因是程序里最后两次迭代结果 x 和 y，在高位小数位上产生了震荡：

[0.999999, 2.0, -0.999999, 2.0]（y 取 6 位精度的时候）；

[0.99999927908, 1.9999991216, -0.99999902422, 2.000000754]（y 取 11 位精度的时候）；

这是正常现象，由此证明了高位小数不是由电脑产生的随机数码。

成绩：　　　　批阅教师签名：　　　　2017 年 10 月 31 日