重 庆 交 通 大 学
学 生 实 验 报 告

实验课程名称 数值计算方法 B(II)

开课实验中心 数学与统计学实验教学中心

开课学院 数学与统计学院

专业年级 信息与计算科学 2018 级

姓　　名 曾 晶 晶

学　　号 631810040303

任课老师 杨 绪 君

开课时间 2020—2021 学年第 2 学期

实验一：非线性方程求根

一、实验目的

掌握二分法、迭代法和牛顿迭代法的基本原理，能够通过上机实验运用二分法、迭代法和牛顿迭代法求解非线性方程的近似根。

二、实验要求

1. 能够用 $Matlab$ 编程实现求解过程，并将问题分析、程序代码、运算结果等内容写入实验报告。

2. 美化文字排版 (数字符号使用 $MathType$ 输入、语言通顺、逻辑正确、图片清晰等等)。

三、实验内容

1.1 二分法的算法步骤

(a) 取初始有根区间 $[a,b]$(满足 $f(a)\cdot f(b)\leq 0$), 以及精度要求 ε;

(b) 若 $\dfrac{b-a}{2}<\varepsilon$, 则停止计算;

(c) 取 $x=\dfrac{a+b}{2}$, 若 $f(a)\cdot f(b)\leq 0$，则置 $b=x$; 否则置 $a=x$; 转 (b).

1.2 实例分析

利用二分法求方程 $f(x)=x^5-4x-16=0$ 在区间 $[3,5]$ 内的根。

1.3 程序代码及运行结果

```
fun=inline('x^5-4*x-16');
[x_star,index,it]=bisect(fun,3,5)
```

```
x_star = 215 3089
index = 0
it = 0
```

2.1 迭代法的算法步骤

(a) 取初始点 x_0, 最大迭代次数 N 和精度要求 ε, 置 $k=0$;

(b) 计算 $x_{k+1}=\phi(x_k)$;

(c) 若 $|x_{k+1}-x_k|<\varepsilon$, 则停止计算;

(d) 若 $k=N$, 则停止计算; 否则, 置 $k=k+1$, 转 (b).

2.2 实例分析

用迭代法求方程 $f(x)=x^4-4x+1=0$ 在区间 $[0,1]$ 上的根.

2.3 程序代码及运行结果

```
phi=inline('x^4/4-1/4');
[x_star,index,it]=iterate(phi,0.5)
```

```
[x_star,index,it]=iterate(phi,0.5)
x_star =-0.2490
index = 1
it =4
```

3.1 牛顿法的算法步骤

(a) 取初始点 x_0, 最大迭代次数 N 和精度要求 ε, 置 $k=0$;

(b) 如果 $f'(x_k)=0$, 则停止计算；否则计算：$x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$;

(c) 若 $|x_{k+1}-x_k|<\varepsilon$，则停止计算；

(d) 若 $k=N$，则停止计算；否则，置 $k=k+1$，转 (b).

3.2 实例分析

求方程 $f(x)=x^3-x-1=0$ 在区间 $1,2$ 内的根。

3.3 程序代码及运行结果

```
fun=inline('[x^5-3*x^3-2*x,5*x^4-9*x^2-2]');
[x_star,index,it]=Newton(fun,1.5)
```

```
%结果
x_star = 1.8872
index = 1
it = 8
```

四、实验心得

通过这个实验，我更好的理解了牛顿法的思想，同时我也思考了双点法和单点法同牛顿法的区别。我认为数值分析主要是通过迭代的方式求出解，所以我们一定要把握好误差。

附录

(1) 二分法程序

```
function[x_star,index,it]=bisect(fun,a,b)
%求解非线性计算方程的二分法，其中，fun(x)为需要求根的函数;
%a,b为初始区间的端点;
%ep为精度，当(b-a)/2<ep时，算法能终止计算，
%缺省值为1e-5;
%当x_star迭代成功时，输出方程的根
%当x_start迭代失败时，输出两端点的值；
%index为指标变量，当index=1时，表明迭代成功，
%当index=0时，表明初始区间不是有根区间；
%it为迭代次数
if nargin<4
ep=1e-5;
end
fa=feval(fun,a);fb=feval(fun,b);
if fa*fb>0
x_star=[fa,fb];index=0;it=0;
return;
end
k=1;
while abs(b-a)/2≥ep
x=(a+b)/2;fx=feval(fun,x);
if fx*fa<0
b=x;fb=fx;
else
a=x;fa=fx;
end
k=k+1;
end
x_star=(a+b)/2;index=1;it=k;
```

(2) 迭代法程序

```
function [x_star,index,it]=iterate(phi,x,ep,it_max)
%求解非线性方程的一般迭代法，其中phi(x)为迭代函数，x为初始点；
%ep为精度，当|x(k)-x(k-1)|<ep时，终止计算，缺省值为le-5；
%it_max为最大迭代次数，缺省值为100；
%x_star为当迭代成功是，输出方程的根，
%      为当迭代失败时，输出最后的迭代值；
%inde为指标变量，当index=1时，表明迭代成功，
%it为迭代次数.
if nargin<4 it_max=100;end
if nargin<3 ep=1e-5;end
index=0;k=1;
while k<it_max
x1=x;x=feval(phi,x);
if abs(x-x1)<ep
index=1;break;
end
k=k+1;
end
x_star=x;it=k;
```

(3) 牛顿法程序

```
function [x_star,index,it]=Newton(fun,x,ep,it_max)
%求解非线性方程的牛顿法
```

```
%第一个分量是函数值，第二个分量是导数值
% x为初始点
% ep为精度,当 | x(k)-x(k-1) |<ep时，终止计算,缺省值为1e-5
% it_max为最大迭代次数,缺省值为100
% x_star为当迭代成功时，输出方程的根
%  当迭代失败，输出最后的迭代值
% index为指标变量，当index=1时，表明迭代成功
% 当index=0时，表明迭代失败（迭代次数≥it_max）
% it为迭代次数
if nargin<4 it_max=100;end
if nargin<3 ep=1e-5;end
index=0;k=1;
while k<it_max
x1=x;f=feval(fun,x);
x=x-f(1)/f(2);
if abs(x-x1)<ep
index=1;break;
end
k=k+1;
end
x_star=x;it=k;
```

实验二：解线性方程组直接法—高斯消元法和矩阵分解法

一、实验目的

1. 掌握高斯消元法和高斯列主元消去法的基本原理，能够通过上机实验运用高斯消元法和高斯列主元消去法求解线性方程组；

2. 掌握杜立特分解法的基本原理，能够通过上机实验运用杜立特分解法求解线性方程组；

3. 掌握追赶法的基本原理，能够通过上机实验运用追赶法求解三对角线性方程组；

4. Matlab 调用不同的方法求解四阶、五阶、六阶等线性方程组（方程组自拟）。

二、实验要求

1. 能够用 *Matlab* 编程实现求解过程，并将问题分析、程序代码、运算结果等内容写入实验报告。

2. 美化文字排版（数学符号使用 *MathType* 输入、语言通顺、逻辑正确、图片清晰等等）。

三、实验内容

1.1 高斯消元法的算法步骤

(a) 找到第 i 列中不为零的那一行 j，要求 $j >= i$;

(b) 交换第 i 行和第 j 行;

(c) 将第 i 行以下的每一行 k 的第 l 个数都减去 $f[i,l] \times \dfrac{a[i,j]}{a[k,j]}$;

(d) 此时已经形成下三角，可以用 hyc 的比较差的方法迭代求解，或者如果没有自由元的话，可以用行列比求解，$\frac{d}{d[i]}$ 即为 $x[i]$ 的解。

1.2 实例分析:

用高斯消元法解方程组

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

1.3 程序代码及运行结果

```
A=[1 2 3; 4 5 6; 7 8 0];b=[1 1 1]';
 [x,index]=Gauss(A,b)
 x = -1      1      0
 index = 1
```

2.1 杜立特分解法的算法步骤

(a) 实现矩阵三角分解 $A = LU$；

(b) 利用分解因子 L，U 解方程组 $AX = b$ 即先求解 $LY = b$ 再求解 $UX = Y$ 的子程序。

2.2 实例分析

用杜立特分解法解方程组

$$\begin{bmatrix} -1 & 2 & 2 \\ 0 & 4 & 2 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 12 \\ 10 \end{bmatrix}$$

2.3 程序代码及运行结果

```
A=[-1 1 2;0 4 2;1 3 1];b=[4 12 10]';
 [L,U,x,y,index]=Doolittle(A,b)
 x =2      2       2
 L = 1 0 0
 0 1 0
 -1 1 1
 U = -1 1 2
 0 4 2
 0 0 1
 x =2 2 2
 y =4 12 2
 index =1
```

2. 追赶法的算法步骤

(a) 设直接三角分解为:$A = LR$;

$$L=\begin{bmatrix}1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & 1 \\ & & l_n & 1\end{bmatrix}, R=\begin{bmatrix}r_1 & p_1 & & \\ & r_2 & \ddots & \\ & & \ddots & p_{n-1} \\ & & & r_n\end{bmatrix}$$

(b) 其中 $p_i=c_i(i=1,2,\ldots,n-1)$, 计算 L 和 R 的其余元素的公式为

$$r_i=b_i$$

$$l_i=\frac{a_i}{r_{i-1}}$$

$$r_i=b_i-l_ic_{i-1}$$

2.2 实例分析

用追赶法解方程组

$$\begin{bmatrix}-4 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & -4\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 1\end{bmatrix}$$

2.3 程序代码及运行结果

```
D=[-4 1 0;1 -4 1;0 1 -4];d=[1 1 1]';
x=zhuiganfa(D,d)
x = -0.357142857143117 -0.428571428571558 -0.357142857142854
```

四、实验心得

最开始没有分清楚高斯消元和列主元素消元法的区别，这次的实验让我明白了什么是列主元素消元法，直接法求解低阶稠密矩阵很有效，但是如果矩阵复杂，那么效率就很低。运行时间特别长。通过将系数矩阵分解成 LU 矩阵的方法很多，但是要满足的条件就是顺序主子式不为零。

(1) 高斯消元法程序

```
function[x_star,index,it]=bisect(fun,a,b)
%求解非线性计算方程的二分法，其中，fun(x)为需要求根的函数；
%a,b为初始区间的端点；
%ep为精度，当(b-a)/2<ep时，算法能终止计算，
%缺省值为1e-5;
%当x_star迭代成功时，输出方程的根
%当x_start迭代失败时，输出两端点的值；
%index为指标变量，当index=1时，表明迭代成功，
%当index=0时，表明初始区间不是有根区间；
%it为迭代次数
if nargin<4
```

```
ep=1e-5;
end
fa=feval(fun,a);fb=feval(fun,b);
if fa*fb>0
x_star=[fa,fb];index=0;it=0;
return;
end
k=1;
while abs(b-a)/2≥ep
x=(a+b)/2;fx=feval(fun,x);
if fx*fa<0
b=x;fb=fx;
else
a=x;fa=fx;
end
k=k+1;
end
x_star=(a+b)/2;index=1;it=k;
```

(2) 杜立特分解法程序

```
function [x,R,index]=DoolittleR(A,b)
%A为要分解的矩阵；
%R下三角阵为L的下三角阵（不包括对角线）；
%R上三角阵为U的上三角阵；
%b为方程组的右端项；
%x为方程组Ax=b的解；
%index以为指标变量，index=0，表示计算失败，index=1，表示计算成功
[m,n]=size(A);
if m≠n
disp('Argument matrix A must be square!');
index=0;
return;
end
%开始计算，赋初值
index=1;
R(1,1:n)=A(1,1:n);
R(2:n,1)=A(2:n,1)/R(1,1);
for k=2:n
for i=k:n
R(k,i)=A(k,i)-R(k,1:(k-1))*R(1:(k-1),i);
end
if R(k,k)==0
disp('The Linear System is singular!');
index=0;
return;
end
for j=(k+1):n
R(j,k)=(A(j,k)-R(j,1:(k-1))*R(1:(k-1),k))/R(k,k);
end
end
y(1)=b(1);
for k=1:n
y(k)=b(k)-R(k,1:k-1)*y(1:k-1)';
end
x(n)=y(n)/R(n,n);
for k=n:-1:1
x(k)=(y(k)-R(k,k+1:n)*x(k+1:n)')/R(k,k);
end
```

(2) 追赶法程序

```
function x=zhuiganfa(A,d)
%首先说明：追赶法适用于三对角矩阵的线性方程组求解的方法，并不适用于其他类型矩阵。
%定义三对角矩阵A的各组成单元，方程为Ax=d.
%b为A的对角线元素(1~n),a为-1对角线元素(2~n),c为+1对角线元素(1~n-1)
a=[0 diag(A,-1)'];
b=diag(A)';
c=diag(A,1)';
n=length(b);
u0=0;y0=0;a(1)=0;
%“追”的过程
L(1)=b(1)-a(1)*u0;
y(1)=(d(1)-y0*a(1))/L(1);
u(1)=c(1)/L(1);
for i=2:(n-1)
L(i)=b(i)-a(i)*u(i-1);
y(i)=(d(i)-y(i-1)*a(i))/L(i);
u(i)=c(i)/L(i);
end
L(n)=b(n)-a(n)*u(n-1);
y(n)=(d(n)-y(n-1)*a(n))/L(n);
%“赶”的过程
x(n)=y(n);
for i=(n-1):-1:1
x(i)=y(i)-u(i)*x(i+1);
end
```

实验三：解线性方程组直接法—分析 Hilbert 矩阵的病态特征

一、实验目的

1. 掌握矩阵范数和矩阵条件数的概念；

2. 运用 *Matlab* 还原课本第 170 页例 9 的求解过程，分析 *Hilbert* 矩阵的病态现象；

3. 自选四阶或更高阶的 *Hilbert* 矩阵，运用 *Matlab* 分析其病态特性。

二、实验要求

1. 能够用 *Matlab* 编程实现求解过程，并将问题分析、程序代码、运算结果等内容写入实验报告。

2. 美化文字排版（数学符号使用 *MathType* 输入、语言通顺、逻辑正确、图片清晰等等）。

三、实验内容

已知 $Hilbert$ 矩阵

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \dfrac{1}{2} & \frac{1}{3} & \cdots \frac{1}{n+1} & \\ \vdots & \vdots & \ddots & \cdots \\ \frac{1}{n} & \frac{1}{1+n} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

分析不同阶 $Hilbert$ 矩阵的病态特性。

1 三阶 Hilbert 矩阵 (a) 计算 3 的条件数 $(3)_\infty$:

(b) 输入：$H = [11/21/3; 1/21/31/4; 1/31/41/5];$

(c) 计算 $_3$ 的逆：$h = inv(H);$

(d) 当 n 愈大， 矩阵病态愈严重；

```
input m:=3
1.0000      0.5000      0.3333
0.5000      0.3333      0.2500
0.3333      0.2500      0.2000
Hilbert 矩阵阶数
3
实际误差 er_actual
9.4019e-16
近似的最大可能差 er_approx
1.1636e-13
最大可能误差 er_max
2.6873e-13
```

2 四阶或更高阶的 Hilbert 矩阵

```
input m:=4
1.0000e+00    5.0000e-01    3.3333e-01    2.5000e-01
5.0000e-01    3.3333e-01    2.5000e-01    2.0000e-01
3.3333e-01    2.5000e-01    2.0000e-01    1.6667e-01
2.5000e-01    2.0000e-01    1.6667e-01    1.4286e-01
Hilbert 矩阵阶数
4
实际误差 er_actual
8.9514e-14
近似的最大可能差 er_approx
3.4447e-12
最大可能误差 er_max
5.5116e-11
```

四、实验心得

在上课的时候曾听到老师说到实验会分析 $Hilbert$ 矩阵的病态特性，所以留意一下。希尔伯特矩阵是一个典型的病态矩阵。这个实验最主要的目的就是对矩阵进行误差分析，判断矩阵是否病态。$Hilbert$ 在阶数越大的时候，误差越大。**附录**

分析不同阶 *Hilbert* 矩阵的病态特性代码

```
m=input('input m:=');
N=[m];
for k=1:length(N)
n=N(k);
H=hilb(n);
disp(H)
Hi=invhilb(n);
b=ones(n,1);
x_approx=H\b;
x_exact=Hi*b;
ndb=norm(H*x_approx-b);nb=norm(b);
ndx=norm(x_approx-x_exact);nx=norm(x_approx);
er_actual(k)=ndx/nx;
K=cond(H);
er_approx(k)=K*eps;
er_max(k)=K*ndb/nb;
end
disp('Hilbert 矩阵阶数'),disp(N)
format short e
disp('实际误差 er_actual'),disp(er_actual),disp('')
disp('近似的最大可能差 er_approx'),disp(er_approx),disp('')
disp('最大可能误差 er_max'),disp(er_max),disp('')
```

实验四：解线性方程组迭代法

一、实验目的

1. 掌握雅可比迭代法的基本原理，能够通过上机实验运用雅可比迭代法求解线性方程组；

1. 掌握高斯-赛德尔迭代法的基本原理，能够通过上机实验运用高斯-赛德尔迭代法求解线性方程组；

4. *Matlab* 调用雅可比迭代法和高斯-赛德尔迭代法程序求解四阶、五阶线性方程组 (方程组自拟)。

二、实验要求

1. 能够用 *Matlab* 编程实现求解过程，并将问题分析、程序代码、运算结果等内容写入实验报告。

2. 美化文字排版（数学符号使用 *MathType* 输入、语言通顺、逻辑正确、图片清晰等等）。

三、实验内容

1. 用雅可比迭代法求解线性方程组

1.1 算法步骤

(a) 置初始矩阵 R=I. 置精度要求 ϵ.

(b) 选取 A 中非对角元素绝对值的最大元素，制定指标 i, j, 即令

$$|a_{ij}| = max|a_{lk}||1 \leq l < k \leq n|.$$

(c) 若 $|a_{ij}| < \epsilon$, 则停止计算。

1.2 雅可比迭代法求四阶或五阶方程组

计算方程组如下：

$$\begin{bmatrix} 1 & 2 & 3 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 3 & 4 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 10 \\ 9 \end{bmatrix}$$

要求误差小于等于 10^{-5}.

1.3 实例代码

```
A=[1 2 3 0;1 1 1 1;2 1 3 4;0 1 2 3];b=[6 4 10 9]';
ep=1e-5;it_max=100;
[x,k,index]=Jacobi(A,b,ep,it_max)
x =
1.0e+41 *
9.5676
6.0515
4.2101
1.8660
k =101
index =1
```

2 用高斯-赛德尔迭代法求解线性方程组

2.1 算法步骤

(a) 将方阵 A 分裂为 $A = M - N$. 其中 M 可逆，则线性方程组 $Ax = b$ 可写成等价形式

$$Mx = Nx + b$$

(b) 称 $x^{(k+1)} = Bx^k + g$ 为简单迭代矩阵。

(c) 在计算新分量 x_i^k 时, 将前面已经算出的分量替代式中分量，即为高斯赛德尔迭代法。

2.2 高斯-赛德尔迭代法求四阶或五阶方程组

计算方程组如下：

$$\begin{bmatrix} 1 & 2 & 3 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 3 & 4 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 10 \\ 9 \end{bmatrix}$$

要求误差小于等于 10^{-6}

2.3 实例代码

```
A=[1 2 3 0;1 1 1 1;2 1 3 4;0 1 2 3];b=[6 4 10 9]';
ep=1e-5;it_max=100;
[x,k,index]=Gau_Seid(A,b,ep,it_max)
x =
1.0e+58 *
3.3405
-3.4765
-1.9227
2.4406
k =101
index =1
```

四、实验心得

这是最后一个实验啦，听说杨老师以后都不教我们了，好舍不得杨老师呀。谢谢老师的悉心培养。这个实验完成于结课考试之后，因为认真复习过了，所以这次的实验相对于以前来说，更简单了。高斯赛德尔求解时，比雅克比更快。说明高斯赛德尔的收敛速度比雅克比的收敛速度更快。

附录

(1) 雅可比迭代法程序

```
function [x,k,index]=Jacobi(A,b,ep,it_max)
%求线性方程组的雅可比迭代法;
%A为方程组的系数矩阵;
%b为方程组的右端项;
%x为方程组的解;
%ep为精度要求,缺省值为1e-5;
%it_max为最大迭代次数，缺省值为100;
%k为迭代次数;
%index为指标变量，index=0表示计算失败，index=1表示计算成功;
if nargin<4
it_max=100;
end
if nargin<3
ep=1e-5;
end
n=length(A);k=0;x=zero(n,1);y=zeros(n,1);index=1;
while k≤it_max
for i=1:n
if abs(A(i,i))<1e-10
index=0;
return;
```

```
end
y(i)=(b(i)-A(i,1:n)*x(i:n)+A(i,i)*x(i)/A(i,i);
end
if norm(y-x,inf)<ep
break;
end
k=k+1;
x=y;
end
```

(2) 高斯-赛德尔迭代法程序

```
function [x,k,index]=Gau_Seid(A,b,ep,it_max)
%求线性方程组的高斯—赛德尔迭代法;
%A为方程组的系数矩阵;
%b为方程组的右端项;
%x为方程组的解;
%ep为精度要求，缺省值为1e-5;
%it_max为最大迭代次数，缺省值为100;
%k为迭代次数;
%index为指标变量，index=0表示计算失败，index=1表示计算成功;
if nargin<4 it_max=100;end
if nargin<3 ep=le-5;end
n=length(A);k=0;
x=zeros(n,1);y=zeros(n,1);index=1;
while k<it_max
for i=1:n
if abs(A(i,i))<1e-10
index=0;
return;
end
if i==1
y(i)=(b(i)-A(i,i+1:n)*x(i+1:n))/A(i,i):
elseif i==n
y(i)=(b(i)-A(i,1:i-1)*y(1:i-1))/A(i,i);
else
y(i)=(b(i)-A(i,i:i-1)*y(1:i-1)-A(i,i+1:n))/A(i,i);
end
end
if norm(y-x,inf)<ep
break;
end
k=k+1;
x=y;
end
```