**东南大学**

**数值分析上机作业报告**

[](https://www.baidu.com/link?url=9juKVhVsVxmpBMo776GI7-_C3TOuWfwo8Ixl7RfKvRuoU1DJth3iXiZccCS-F9T6Wj7c_aann7vQlkpb-qfaedKBKGoVsu0zA5F7GI6isVWqlwqOQ43JpwI0xHkOILhGAYIaHiYSLdWdAo-wK-Va5ZYylsYfAqNsSOfohOgU3fPgnkyLrZICYKMm7ldHI2XEo2x3Dsq7M4pL645gg7K0W8afZ53oasap6ceBT9wyAPe-Y-q6oXKdPxyDenxyYpaexilbuxQd-e2FItd7ysTcXtuteVUjoWVbxBWv2rZVnJnmEO_UDewF3P1ePG4xjK3YD4TVBuM95tlHRfK7bdNJ4eVz35DYfwZZrGKu1Z7CEA8hVStecFcUWh33EOTtRLD9FgGUk-3CNiu2xOtKif76AwAwr1P9LSyf8dbSve4lkUjEIxIRzVsLUpcDVuXX3xvOwIzV1_j33dXktCJXtWyawap_N-mstJFYJ81QtdDRWejwlf9FC0L41ZTA2RH4NtJt4yRJ7rSTIuGySzVjHiGxGhUvrXi-pM2mkgHm6pUOk1ZZiLEkpVY1Y4BLwjCd8NeXCLM9xKiIqwPphy3C40rIH8BiMEJXW8MABtpi4N-OOsrD6f5TohKN9Nzx2dT0tlyZ6EyoPrQXJDHi41uCLObcgPciXnIyWaalL3v63_tbhGGQuRHPa-ab5T9J0T6G0XQIqQGX1E0WV-_IOqGoEWZXOcF0Lmo_8h2zqmtrnPNUTX5yl-7rUGkvI5JvcRtPA17v&timg=&click_t=1540285255174&s_info=1350_661&wd=&eqid=bd1672330003099c000000035bcee333)

**课程名称： 数值分析**

**姓 名： 曾昕卢**

**学 号： 192774（A20）**

**指导老师： 杜睿**

## 一、第一章上机题

记



则有



给定函数 记



称为函数的阶Taylor多项式。

计算的近似值，可以用如下三种方法：

(Ⅰ)将在处进行Taylor展开，取前10项的和 估计误差



计算出，并回答作为的近似值具有几位有效数字。

程序如下：

clc

clear

syms x

y1=log(1+x);

N=input('请输入N:');

S=taylor(y1,x,0,'Order',N);

SS=abs(y1-S);

data=zeros(1,10000);

n=1;

for z=0:0.001:1 %精度可调整

answer=subs(SS,x,z);

answer=single(answer);

data(n)=answer;

n=n+1;

end

result=max(data);

fprintf('result=%f\n',result);

S1=subs(S,x,1);

fprintf('S(1)=%f\n',S1);

e1=subs(SS,x,1);

fprintf('e1=%f\n',e1);

运行结果：



作为的近似值没有有效数字。

(Ⅱ)将在处进行Taylor展开，取前10项的和 估计误差



计算出，并回答作为的近似值具有几位有效数字。

程序如下：

clc

clear

syms x

y2=log(1/(1+x));

N=input('请输入N:');

S=taylor(y2,x,0,'Order',N);

SS=abs(y2-S);

data=zeros(1,10000);

n=1;

for z=-0.5:0.001:0.5 %精度可调整

answer=subs(SS,x,z);

answer=single(answer);

data(n)=answer;

n=n+1;

end

result=max(data);

fprintf('result=%f\n',result);

S2=subs(S,x,-0.5);

fprintf('S(-0.5)=%f\n',S2);

e2=subs(SS,x,-0.5);

fprintf('e2=%f\n',e2);

运行结果：



作为的近似值具有三位有效数字。

(Ⅲ)将在处进行Taylor展开，取前10项的和 估计误差



计算出，并回答作为的近似值具有几位有效数字。

程序如下：

clc

clear

syms x

y3=log((1+x)/(1-x));

N=input('请输入N:');

S=taylor(y3,x,0,'Order',N);

SS=abs(y3-S);

data=zeros(1,10000);

n=1;

for z=-1/3:0.001:1/3 %精度可调整

answer=subs(SS,x,z);

answer=single(answer);

data(n)=answer;

n=n+1;

end

result=max(data);

fprintf('result=%f\n',result);

S3=subs(S,x,1/3);

fprintf('S(1/3)=%f\n',S3);

e3=subs(SS,x,1/3);

fprintf('e3=%f\n',e3);

运行结果：



作为的近似值具有五位有效数字。

(Ⅳ)比较上述三种算法的计算精度，体会编程中要注意的“尽量避免大数吃小数”、“尽量避免相近数相减”的准则。

答：对于，

对于，

对于，

在中，当时，同时有“大数吃小数”和“相似数相减”；在中，当时，各项均为正数，仅有“大数吃小数”；在中，当时，“大数吃小数”的程度相比又小了一些。由运算结果可以看出，的误差最小，的误差最大，因此在数值分析编程中要注意“尽量避免大数吃小数”和“尽量避免相近数相减”的准则。

## 二、第二章上机题

记



(Ⅰ)分析方程存在几个实根。给出每个根所在的区间。

程序如下：

clc

clear

syms x;

f=3\*x^2-exp(x);

g=diff(f);

h=0.1; %区间宽度可调整

x0=0;

n=0;

data=zeros(1,100);

for i=-100:100

a=subs(g,x,x0+i\*h);

b=subs(g,x,x0+(i+1)\*h);

if(a\*b<0)

n=n+1;

data(n)=i;

else

n=n;

end

end

fprintf('实根个数为%d\n',n);

disp('根所在区间为:');

for j=1:n

fprintf('[%.1f,%.1f]\n',x0+data(j)\*h,x0+(data(j)+1)\*h);

end

运行结果：

实根个数为2

根所在区间为：

(Ⅱ)研究用Newton法求方程的根，分析迭代初值在什么范围内选取时，迭代序列收敛到相应的根。用Newton法求出具有8位有效数字的近似根。

程序如下：

求根函数程序为：

function [answer]=qiu\_gen(x0)

syms x

f=3\*x^2-exp(x);

g=diff(f);

g1=diff(g);

y=x-g/g1;

t1=x0;

t2=double(subs(y,x,x0));

e=0.5e-8;

while(abs(t2-t1)>e)

t1=t2;

t2=double(subs(y,x,t1));

end

answer=t2;

end

主程序为：

clc

clear

x0=input('迭代初值x0=');

t0=qiu\_gen(x0);

h=0.001;%精度可修改

i=1;j=1;

tmax=t0;tmin=t0;

if(t0<1.5)

while(qiu\_gen(tmax)==t0)

tmax=t0+h\*i;

i=i+1;

end

fprintf('g(x)=0的根为%.8f\n',t0);

fprintf('收敛区间为(-inf %.3f)\n',tmax);

else

while(qiu\_gen(tmin)==t0)

tmin=t0-h\*j;

j=j+1;

end

fprintf('g(x)=0的根为%.7f\n',t0);

fprintf('收敛区间为(%.3f inf)\n',tmin);

end

运行结果：

|  |  |
| --- | --- |
| 的根 | 收敛区间 |
| 0.20448145 |  |
| 2.8331479 |  |

(Ⅲ)分析方程存在几个实根。给出每个根所在的区间。

程序如下：

clc

clear

syms x;

f=@(x)expression;

f=3\*x^2-exp(x);

h=0.1; %区间宽度可调整

x0=0;

n=0;

data=zeros(1,100);

for i=-500:500

a=subs(f,x,x0+i\*h);

b=subs(f,x,x0+(i+1)\*h);

if(a\*b<0)

n=n+1;

data(n)=i;

else

n=n;

end

end

fprintf('实根个数为%d\n',n);

disp('根所在区间为:');

for j=1:n

fprintf('[%.1f,%.1f]\n',x0+data(j)\*h,x0+(data(j)+1)\*h);

end

运行结果：

实根个数为3

根所在区间为：

(Ⅳ)研究用Newton法求方程的根，分析迭代初值在什么范围内选取时，迭代序列收敛到相应的根。用Newton法求出具有8位有效数字的近似根。

程序如下：

求根函数程序为：

function [answer]=qiuf\_genf(x0)

syms x

f=3\*x^2-exp(x);

g=diff(f);

y=x-f/g;

t1=x0;

t2=double(subs(y,x,x0));

e=0.5e-8;

while(abs(t2-t1)>e)

t1=t2;

t2=double(subs(y,x,t1));

end

answer=t2;

end

主程序为：

clc

clear

x0=input('迭代初值x0=');

t0=qiuf\_genf(x0);

h=0.001;%精度可修改

i=1;j=1;

p=1;q=1;

tmax=t0;tmin=t0;

if(t0<0)

while(qiuf\_genf(tmax)==t0)

tmax=t0+h\*i;

i=i+1;

end

fprintf('f(x)=0的根为%.8f\n',t0);

fprintf('收敛区间为(-inf,%.3f)\n',tmax);

else

if(t0>1)

while(qiuf\_genf(tmin)==t0)

tmin=t0-h\*j;

j=j+1;

end

fprintf('f(x)=0的根为%.7f\n',t0);

fprintf('收敛区间为(%.3f,inf)\n',tmin);

else

while(qiuf\_genf(tmin)==t0)

tmin=t0-p\*h;

p=p+1;

end

while(qiuf\_genf(tmax)==t0)

tmax=t0+q\*h;

q=q+1;

end

fprintf('f(x)=0的根为%.8f\n',t0);

fprintf('收敛区间为(%.3f,%.3f)\n',tmin,tmax);

end

end

运行结果：

|  |  |
| --- | --- |
| 的根 | 收敛区间 |
| -0.45896227 |  |
| 0.91000757 |  |
| 3.7330790 |  |

## 三、第三章上机题

已知Hilbert矩阵的元素为

(Ⅰ)编程计算的无穷范数的程序。

程序如下：

clc

clear

n=input('矩阵阶数n=');

H=zeros(n,n);

for i=1:n

for j=1:n

H(i,j)=1/(i+j-1);

end

end

S=zeros(n,1);

for m=1:n

S(m,1)=0;

end

for i=1:n

for j=1:n

S(i,1)=S(i,1)+abs(H(i,j));

end

end

A=max(S);

fprintf('%d阶Hilbert矩阵的无穷范数为%.8f\n',n,A);

运行结果：

矩阵阶数n=5

5阶Hilbert矩阵的无穷范数为2.28333333

(Ⅱ)编写计算的无穷范数条件数的程序（可调用求逆函数，在Mathematica中为，在Matlab中为，在其他语言中请自行查找。）

程序如下：

clc

clear

n=input('矩阵阶数n=');

H=zeros(n,n);

for i=1:n

for j=1:n

H(i,j)=1/(i+j-1);

end

end

H1=inv(H);

% H1=invhilb(n);

S=zeros(n,1);

S1=zeros(n,1);

for m=1:n

S(m,1)=0;

S1(m,1)=0;

end

for i=1:n

for j=1:n

S(i,1)=S(i,1)+abs(H(i,j));

S1(i,1)=S1(i,1)+abs(H1(i,j));

end

end

A=max(S)\*max(S1);

fprintf('%d阶Hilbert矩阵的无穷范数条件数为%f\n',n,A);

运行结果：

矩阵阶数n=5

5阶Hilbert矩阵的无穷范数条件数为943655.999999

(Ⅲ)对，计算的无穷范数条件数；画出的关系图。

程序如下：

计算无穷范数条件数函数程序如下：

function answer=inf\_cond(n)

H=zeros(n,n);

for i=1:n

for j=1:n

H(i,j)=1/(i+j-1);

end

end

H1=inv(H);

% H1=invhilb(n);

S=zeros(n,1);

S1=zeros(n,1);

for m=1:n

S(m,1)=0;

S1(m,1)=0;

end

for i=1:n

for j=1:n

S(i,1)=S(i,1)+abs(H(i,j));

S1(i,1)=S1(i,1)+abs(H1(i,j));

end

end

answer=max(S)\*max(S1);

end

主程序如下：

clc

clear

n=zeros(10,1);

x=zeros(10,1);

y=zeros(10,1);

for i=1:10

n(i,1)=10\*i;

x(i,1)=inf\_cond(n(i,1));

y(i,1)=log(x(i,1));

end

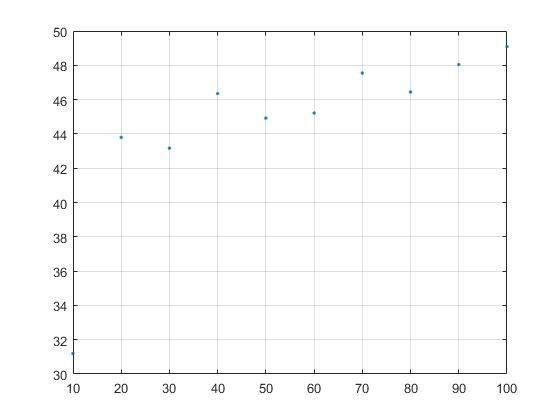
plot(n,y,'.');

grid on;

运行结果：

|  |  |  |  |
| --- | --- | --- | --- |
| n |  | n |  |
| 10 | 31.1964 | 60 | 45.2136 |
| 20 | 43.7960 | 70 | 47.5446 |
| 30 | 43.1681 | 80 | 46.4422 |
| 40 | 46.3531 | 90 | 48.0468 |
| 50 | 44.9171 | 100 | 49.0898 |

的关系图如下：



(Ⅳ)令，，对，求解，并计算和以及它们的无穷范数。

程序如下：

clc

clear

n=input('矩阵阶数n=');

H=zeros(n,n);

for i=1:n

for j=1:n

H(i,j)=1/(i+j-1);

end

end

x=zeros(n,1);

for m=1:n

x(m,1)=1;

end

b=H\*x;

x0=H\b;

c1=x-x0;

c2=b-H\*x0;

c11=max(abs(c1));

c22=max(abs(c2));

运行结果：

和的具体值见工作区的c11和c22，由于是的向量，这里就不予列出，仅列出其无穷范数。

|  |  |  |
| --- | --- | --- |
| n |  |  |
| 10 |  |  |
| 50 |  |  |
| 100 |  |  |

(Ⅴ)通过以上的数值实验，你有哪些体会？

答：对于高阶Hilbert矩阵而言，其无穷范数条件数远大于1，是病态方程组。由第(Ⅳ)问可以看出，无论多少阶数的Hilbert矩阵，尽管都很小，但是随着Hilbert矩阵阶数的增加，其无穷范数条件数越来越大，解向量的相对误差仍会变得非常大。由此说明用余量的大小来检验近似解的准确程度的方法仅对良态方程组适用，对于病态方程组是不可靠的。