1. 对给定的线性方程组进行迭代求解。

（1）给出Jacobi迭代的通用程序。

（2）给出Gauss-Seidel迭代的通用程序。

调用条件：系数矩阵，右端项，初值，精度要求。

输出结果：方程组的近似解。

给定线性方程组，和，取初值为0，

分别利用Jacobi迭代和G-S迭代进行求解，观察并解释其中的数学现象。

解：

%雅可比迭代法

%Jacobi迭代的通用程序

%迭代公式x(k+1)=Gx(k)+d,(k=0,1,…).

function[x]=jacobi(A,b,x0,cc)

%A系数矩阵

%右端项b(ax=b)

%初值x0

%精度要求cc

%x的近似解

D=diag(diag(A)); %提取Ax=b中A的分解D L U中的D

L=tril(A,-1); %L low

U=triu(A,1); %U up

G=(-inv(D))\*(L+U); %雅可比迭代法的G

d=inv(D)\*b ; %雅可比迭代法的d

[v,s]=eig(G); %判断矩阵是否收敛

eigenvalue=diag(s); %提取G的特征值

max\_d=max(eigenvalue); %提取特征值的最大值

if(max\_d>=1)

fprintf('矩阵不收敛');

else

%开始迭代

x1=x0;

for i=0:1000

fprintf('第%d次迭代：',i);

x2=G\*x1+d;

if norm(x2-x1,'inf')<cc

break;

end

fprintf('\n与上次计算结果的距离(2范数):%f \n',norm(x2-x1,inf));

x1=x2;

end

x=x2

C=A\*x

end

end

函数调用：

clear

clc

%%%%%%%%%%%%%%% 线性方程组A %%%%%%%%%%%%%%%%%%%%%%%%

A=[2 -1 1; 2 2 2;-1 -1 2];

b=[-1; 4; -5];

x0=[0;0;0];

cc=0.00005; %精度要求

%%%%%%%%%% 性方程组B %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B=[1 2 -2;1 1 1;2 2 1];

b1=[7; 2 ;5];

x1=[0;0;0];

cc=0.00005; %精度要求

%调用雅可比迭代函数

jacobi(A,b,x0,cc)

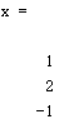
jacobi(B,b1,x1,cc)

结果：

矩阵A：



矩阵B：



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%Gauss-Seidel迭代的通用程序

function[x]=Gauss\_Seidel(A,b,x0,cc)

%A系数矩阵

%右端项b(ax=b)

%初值x0

%精度要求cc

%x的近似解

D=diag(diag(A)); %提取Ax=b中A的分解D L U中的D

L=tril(A,-1); %L low

U=triu(A,1); %U up

G=(-inv(D+L))\*(U); %迭代法的G

d=inv(D+L)\*b; %迭代法的d

[v,s]=eig(G);

eigenvalue=diag(s);

max\_d=max(eigenvalue);

% if (norm(G,inf)>1||norm(G,2)>1||norm(G,1)>1) %判断矩阵是否收敛

if(max\_d>=1)

fprintf('矩阵不收敛');

else

%开始迭代

x1=x0;

for i=0:1000

fprintf('第%d次迭代：',i);

x2=G\*x1+d;

if norm(x2-x1,'inf')<cc

break;

end

fprintf('\n与上次计算结果的距离(2范数):%f \n',norm(x2-x1,inf));

x1=x2;

end

x=x1

A\*x

end

end

函数调用

clear

clc

%%%%%%%%%%%%%%%%%%%%% 线性方程组A %%%%%%%%%%%%%%%%%%%%%%%%

A=[2 -1 1; 2 2 2;-1 -1 2];

b=[-1; 4; -5];

x0=[0;0;0];

cc=0.00005; %精度要求

%%%%%%%%%%%% 性方程组B %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B=[1 2 -2;1 1 1;2 2 1];

b1=[7; 2 ;5];

x1=[0;0;0];

cc=0.00005; %精度要求

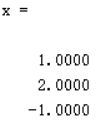
%调用G-S迭代法

Gauss\_Seidel(A,b,x0,cc)

Gauss\_Seidel(B,b1,x1,cc)

结果：

A矩阵：



B矩阵：



2. 利用紧凑格式（即直接分解法或逐框运算法）对给定的矩阵进行Doolittle分解，并用其求线性方程组的解。

调用条件：矩阵。

输出结果：单位下三角矩阵和上三角矩阵。

给定矩阵，利用以下算法：

1）将A作Doolittle分解，

2）令，并对作Doolittle分解，

3）重复2）的过程令，并对作Doolittle分解，，

观察，，的变化趋势，思考其中的数学现象。

解：

clear

clc

A=[1 1;1 2];

n=20; %迭代次数

for i=1:n

[L,U]=lu(A); %将A进行LU分解

A=U\*L;

fprintf('\n第%d次输出',i);

if(i==20)

A=L\*U

L %下三角矩阵

U %上三角矩阵

break;

end

A

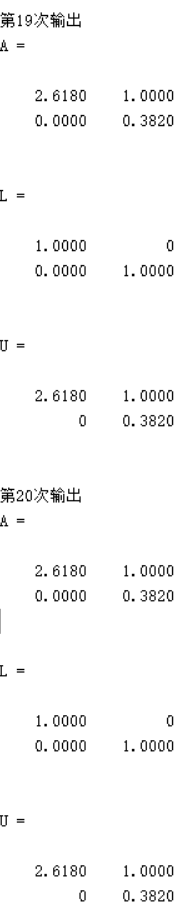
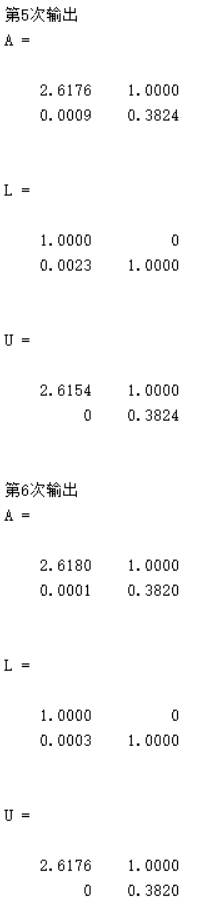
L

U

End

结果：

从图中可以看出L逐渐趋近与单位矩阵，U与A逐渐成为上三角矩阵。



3. 给定函数，取，用等距节点 对原函数进行多项式插值和五次多项式拟合，试画出插值和拟合曲线，并给出数--学解释。

解：

function [a,A] = newton\_CZF(n,x0,xi,yi)

% newton\_CZF牛顿法求插值多项式。

% n 插值节点的个数；

% xi 节点值；

%yi 节点函数值；

%a 插值多项式系数；

%A 差商表；

%x0 待计算的点；

%result 多项式在x0处的取值；

A=zeros(n);

for k=1:n

A(k,1)=yi(k);

end

for j = 2 : n

for i = j : n

A(i,j) = (A(i,j-1) - A(i-1,j-1))/(xi(i)-xi(i-j+1));

end

end

B=diag(A); %得到系数向量；

stemp = ['系数向量为 = ', num2str(B')];

disp(stemp)

Z=ones(1,n);

for i=2:n

Z(1,i)=x0-xi(i);

end

W=zeros(1,n);

for i=1:n

temp=Z(1:i);

W(i)=prod(temp); %x0-xi

end

result=W\*B; %输出函数在x0出结果；

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

end

调用：

%插值多项式

n=4;

yi=zeros(1,n+1);

xi=zeros(1,n+1);

xi(1)=0;

for i=1:n+1

x=-1+2\*i/n;

x=x-2/n; %为确保x在[-1，1]内

y=1/(1+25\*x^2);

if(xi(i)>1)

break;

end

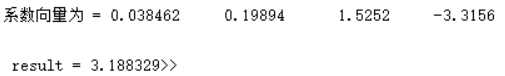
yi(i)=y;

xi(i)=x;

end

newton\_CZF(n,-1,xi,yi)

结果：



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clc

clear

%五次多项式拟合

n=4;

yi=zeros(1,n+1);

xi=zeros(1,n+1);

for i=1:n+1

x=-1+2\*i/n;

x=x-2/n; %为确保x在[-1，1]内

y=1/(1+25\*x^2);

if(xi(i)>1||xi(i)<-1)

break;

end

yi(i)=y;

xi(i)=x;

end

figure(1)

plot(xi,yi,'r');

结果：

N=4



N=8



N=16



观察可得：

多项式插值效果随n的增加变好，但拟合效果随n的增加而下降。

4. 给出迭代法求非线性方程的根的程序。

调用条件：迭代函数，初值

输出结果：根的近似值和迭代次数

给定方程，用迭代格式计算附近的根，要使计算结果具有四位有效数字，利用估计式，或估计式来判断需要的迭代次数，分别需要迭代多少次？两者是否有冲突？

解：

%非线性方程组迭代法

function[x\_k,i]=nonlineDD(x\_k,e)

%精度e

%x\_k1 迭代函数

%x\_k迭代初值

%x\_kk迭代近似值

%i迭代次数

for i=1:10000

x\_k1=(x\_k^2+1)^(1/3);

if abs(x\_k1-x\_k<e)

fprintf('第%d次迭代\n',i);

fprintf('x的值%f\n',x\_k1);

break;

end

x\_k=x\_k1;

end‘

调用函数：

clear

clc

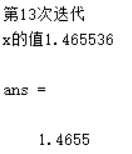
%迭代法

x\_k=0; %初值

e=0.00005; %精度

nonlineDD(x\_k,e) %调用函数

结果：



5. 利用数值求积算法计算。

（1）利用Romberg算法计算积分。

调用条件：被积函数，精度要求。

输出结果：定积分的近似值。

（2）将区间进行等分，每个小区间上利用点高斯型求积公式计算。

调用条件：被积函数，区间等分数，每个子区间的高斯点数。

输出结果：定积分的近似值。

给定，利用上述方法进行计算，并与准确值进行比较。

解：

（1）：

%romberg求积函数

function[R]=Romberg5(f,a,b,e)

%被积函数f

%精度e

%a x的左区间

%b x的右区间

%romberg表

n=1; %区间二分次数

while 1 %在此谨代表多次二分，在后面的判断中止循环

R=zeros([n+1,n+1]); %生成（n+1)(n+1)的0矩阵

R(1,1)=(b-a)/2\*(feval(f,a)+feval(f,b)); %初始值2点梯形公式

for i=1:n %按照公式计算romberg表

h=(b-a)/2^i;

s=0;

for k=1:2^(i-1)

s=s+feval(f,a+(2\*k-1)\*h); %复化体形递推公式-7-21的求和项

end

R(i+1,1)=R(i,1)/2+h\*s; %Tn复化体形递推公式-7-21

end

for j=1:n %计算romberg表的其他列

fac=1/(4^j-1);

for m=j:n

R(m+1,j+1)=R(m+1,j)+fac\*(R(m+1,j)-R(m,j)); %Tm（k) romberg求积算法 4.11

end

end

if(abs(R(n,n)-R(n+1,n+1)))<e %当精度满足要求退出循环

break;

end

n=n+1; %未达到指定精度，继续二分

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

（2）

%高斯求积函数

function[]=Gauss\_5(f,a,b,n,m)

%被积函数f

%精度e

%a x的左区间

%b x的右区间

%n n等分

%m m个高斯点

K1=[0];

K2=[0.5773503 -0.5773503]; %组成高斯求积节点数和系数表

K3=[-0.7745967 0 0.7745967];

K4=[-0.8611363 -0.3399810 0.3399810 0.8611363];

W1=[2];

W2=[1 1];

W3=[0.5555556 0.8888889 0.5555556];

W4=[0.3478548 0.6521452 0.6521452 0.3478548 ];

switch (m)

case 1

GaussP=K1; %高斯点

GaussA=W1; %g高斯系数

case 2

GaussP=K2; %高斯点

GaussA=W2; %g高斯系数

case 3

GaussP=K3; %高斯点

GaussA=W3; %g高斯系数

case 4

GaussP=K4; %高斯点

GaussA=W4; %g高斯系数

otherwise

fprintf('超出高斯节点系数表范围，请重新选点（点数在1-4范围内）\n' );

end

h=abs((b-a)/n); %部分步长

x=a:h:b;

for i=1:length(x)-1

points = h/2\*GaussP+(x(i+1)+x(i))/2; %区间变换

ff(i)=0;

for k=1:m

ff(i)=ff(i)+h/2\*feval(f,points(k))\*GaussA(k); %高斯求积公式

end

end

result=sum(ff)

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数调用：

clc

clear

%利用Romberg算法计算积分

f=@(x)4/(1+x^2); %被积函数f

a=0; %a x的左区间

b=1; %b x的右区间

e=0.00001; %精度e

n=10; %10的步长

m=3; %3个高斯点

%调用romberg函数

fprintf('romberg求值表')

Romberg5(f,a,b,e)

%调用高斯勒让德积分函数

fprintf('高斯勒让德积分函数结果')

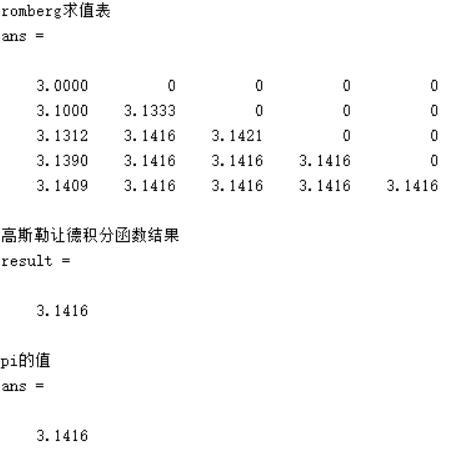
Gauss\_5(f,a,b,n,m)

%pi的值

fprintf('pi的值')

pi

结果：



6. 给定常微分方程，

（1）给出Runge-Kutta 4阶算法的通用程序。

（2）给出Adams显式4阶公式的通用程序（初值由Runge-Kutta 4阶算法提供）。

调用条件：，区间，初值，步长。

输出结果：节点处函数的近似值。

针对初值问题，分别取步长计算各点（）的值，并与准确值比较，解释其中的数学现象。

解：

（1）

%4阶龙格库塔函数

function[x,y]=rungekutta6(f,y0,x0,xn,hh)

%f 函数句柄

%y0初始值

%计算范围x0-xn

%hh 步长

x=x0:hh:xn; %计算节点

y(1)=y0;

N=length(x); %节点数

for k=1:N-1 %龙格库塔函数公式

kk(1)=f(x(k),y(k));

kk(2)=f(x(k)+hh/2,y(k)+hh/2\*kk(1));

kk(3)=f(x(k)+hh/2,y(k)+hh/2\*kk(2));

kk(4)=f(x(k)+hh,y(k)+hh\*kk(3));

y(k+1)=y(k)+hh/6\*(kk(1)+2\*kk(2)+2\*kk(3)+kk(4)); %四级四阶Runge-Kutta公式

end

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

（2）

%adams4阶函数

function[yy]=adams6(f,yy,x0,xn,hh)

%f 函数句柄

%y0初始值

%计算范围x0-xn

%hh 步长

x=x0:hh:xn;

n=length(x);

for i=4:n-2

yy(i+1)=yy(i)+(hh/24)\*(55\*f(x(i),yy(i))-59\*f(x(i-1),yy(i-1))+37\*f(x(i-2),yy(i-2))-9\*f(x(i-3),yy(i-3)));

end

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

调用：

clc

clear

% 题目

f=@(x,y) -30\*y+30\*(x^2)+2\*x; %f 函数句柄

y0=1/3; %y0初始值

x0=0; %计算范围x0-xn

xn=1;

hh=0.005; %hh 步长

%调用4阶龙格库塔函数

figure(1)

[x,y]=rungekutta6(f,y0,x0,xn,hh);

plot(x,y,'.',x,1/3\*(exp(-30\*x))+x.^2,'r-.d') %绘制数值解与解析解

legend('数值解','解析解','Location','northwest')

title('4阶龙格库塔函数计算结果与解析解比较')

yy=zeros(1,length(x0:hh:xn)); %adams初值调用龙格库塔函数

yy(1)=y(1);

yy(2)=y(2);

yy(3)=y(3);

yy(4)=y(4);

%调用adams显示4阶函数

[yy]=adams6(f,yy,x0,xn,hh);

figure(2)

plot(x,yy,'.',x,1/3\*(exp(-30\*x))+x.^2,'r-.d') %绘制数值解与解析解

legend('数值解','解析解','Location','northwest')

title('adams显式4阶函数计算结果与解析解比较')

结果：

H=0.1





H=0.05





H=0.005



