数值分析作业*

CY.SP

2023 年 11 月 28 日

目录

*Github 仓库

1 第一周数值分析实验

1.1 第一节: 级数求和与二元函数绘图

Q 1.1.1. 编程求 $\sum_{n=1}^{12} n!$ 的值.

```
clc
clear
s=zeros(1,12);
for i=1:12
    %s(i)=factorial(i);%使用阶乘函数
    s(i)=prod(1:i);
end
result=sum(s);
disp(result);
```

Q 1.1.2. 绘制二元函数图像.

$$f(x,y)=x^4-2x^2y+x^2-2xy+2y^2+\frac{9}{2}x-4y+4 \quad (-2\le x\le 3,-1\le y\le 7)$$

```
clc
clear
x=linspace(-2,3,100);
y=linspace(-1,7,100);
[X,Y]=meshgrid(x,y);
Z=X.^4-2.*X.^2.*Y+X.^2-2.*X.*Y+2.*Y.^2+9.*X./2-4.*Y+4;
fig=mesh(X,Y,Z);
```

1.2 第二节: 积分递推公式

Q 1.2.1. 计算 $I_n=e^{-1}\int_0^1 x^ne^xdx,(n=0,1,2,\cdots)$ 并估计误差

方法 1: 利用递推公式 (A)

$$(A)\quad \begin{cases} I_n=1-nI_{n-1}, & n=1,2,\cdots,20 \\ I_0=1-e^{-1}\approx 0.6321. \end{cases}$$

方法 2: 利用递推公式 (B)

$$(B)\begin{cases} \widetilde{I_{20}}=?; \\ \widetilde{I}_{n-1}=\frac{1}{n}\left(1-\widetilde{I}_n\right), & n=19,\cdots,9,8,\cdots 0. \end{cases}$$

```
%% 第一种递推方式
clc
clear
N=20;
IA=zeros(1,N);%积分值
EA=zeros(1,N);%误差估计
I0=0.6321;%by Mathematica I_0=0.632121...
EA(1)=5E-5;
IA(1)=1-I0;
for n=1:N-1
    IA(n+1)=1-(n+1)*IA(n);% k from 1 to 20
    EA(n+1)=n*EA(n);
end
disp(IA);
disp(EA);
%% Equation 第二种递推方式
clc
clear
N=20;
EB=zeros(1,N);%误差估计
IB=zeros(1,N);%积分值
IB(20)=0.0684;%by Mathematica I_{20}=0.0455449...
EB(20)=5E-2;
for n=N:-1:2
    IB(n-1)=1/n*(1-IB(n));% k from 1 to 20
    EB(n-1)=1/n*EB(n);
end
disp(IB);
disp(EB);
```

Q 1.2.2. 计算 $I_n = \displaystyle\int_0^1 \frac{x^n}{x+5}dx, (n = 0, 1, 2, \cdots 20)$ 并估计误差.

```
%% 第一种递推方式
clc
clear
N=20;
```

```
IA=zeros(1,N);%积分值
EA=zeros(1,N);%误差值
I0=0.1823;%by Mathematica I_0=0.182322...
EA(1)=5E-6;
IA(1)=1-5*I0;
for n=1:N-1
    IA(n+1)=1/(n+1)-5*IA(n);% k from 1 to 20
    EA(n+1)=5*EA(n);
end
disp(IA);
disp(EA);
%% 第二种递推方式
clc
clear
N=20;
EB=zeros(1,N);%误差值
IB=zeros(1,N);%积分值
IB(20)=0.00799;%by Mathematica I_{20}=0.00799752...
EB(20)=5E-5;
for n=N:-1:2
    IB(n-1)=1/(5*n)-1/5*IB(n);% k from 1 to 20
    EB(n-1)=1/5*EB(n);
end
disp(IB);
disp(EB);
```

2 第二周数值分析实验

2.1 第一节: 插值 function

2.1.1 直接法

```
function y=Vslove(a,B) %slove Matrix AX=B
%a=matrix of 1\times n
%B=matrix of n\times 1
    syms x
    y=0;
    A=zeros(length(a),length(a));
    A=vander(a);
    X=zeros(length(B'),1);%定义X矩阵n\times 1 即为多项式系数
    X=inv(A)*B;
    for i=1:length(X')
        y=y+X(length(X')+1-i)*x^(i-1);
    end
    y=collect(y,x);%合并x项
    t=min(a):0.01:max(a);
    y1=matlabFunction(y);
    plot(t,y1(t));
    hold on
    plot(a,B,'ro');
end
```

2.1.2 Lagrange 法

```
function y=lagrangeslove(X,Y)
%X=matrix of 1\times n 各X坐标
%Y=matrix of 1\times n 相应的Y
    syms x
    s1=0;
    for i=1:length(X)
        t=1;
        for j=1:length(Y)
            if j~=i
```

```
                t=t*(x-X(j))/(X(i)-X(j));
            end
        end
        s1=s1+Y(i)*t;
    end
    y=collect(s1,x);%合并x项
    z=min(X):0.01:max(X);
    s2=matlabFunction(s1);
    plot(z,s2(z));
    hold on
    plot(X,Y,'ro');
end
```

2.2 第二节: 计算实例

Q 2.2.1. 给出插值点

$$x = (0.25, 0.3, 0.39, 0.45, 0.53, 0.66, 0.72)^T,$$

$$y = (0.5, 0.5477, 0.6245, 0.6708, 0.7280, 1.0254, 0.9521)^T.$$

时的两种方法插值多项式表达式及对应的函数图像, 并与 *MATLAB* 拟合工具箱 *(cftool)* 进行比较.

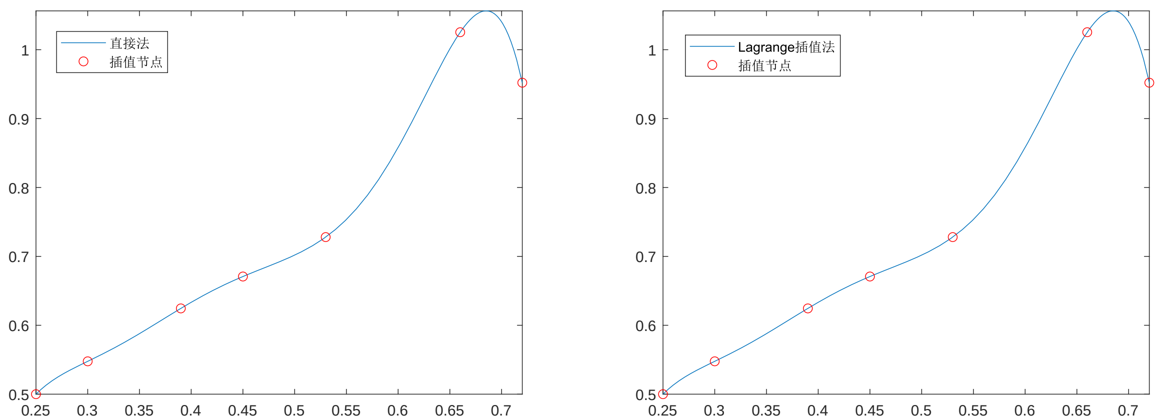
2.2.1 直接法计算

```
clc
clear
a=[0.25, 0.3, 0.39, 0.45, 0.53, 0.66, 0.72];
B=[0.5, 0.5477, 0.6245, 0.6708, 0.7280, 1.0254, 0.9521]';
y=Vslove(a,B);
```

2.2.2 Lagrange 法计算

```
clc
clear
a=[0.25, 0.3, 0.39, 0.45, 0.53, 0.66, 0.72];
B=[0.5, 0.5477, 0.6245, 0.6708, 0.7280, 1.0254, 0.9521];
y=lagrangeslove(a,B)
```

2.2.3 结果



(a) 直接法图像

(b) Lagrange 法图像

图 1: 运行结果

3 第三周数值分析实验

3.1 Newton 法均差表 function

```
function A=newtonmatrix(X,Y)
%均差表
    %X为所有x_i所构成的行向量
    %Y为所有y_i所构成的行向量
    n=length(X);
    A=[X',Y',zeros(n,n-1)];%l=n+1,h=n
    for l=3:n+1 %列
        for h=l-1:n %行
            A(h,l)=(A(h,l-1)-A(h-1,l-1))/(X(h)-X(h-l+2));
        end
    end
end
```

3.2 page32ex4 实例

```
clc
clear
syms x
X=[0.40 0.55 0.65 0.80 0.90 1.05];
Y=[0.41075 0.57815 0.69675 0.88811 1.02652 1.25382];
y=Y(1);
A=newtonmatrix(X,Y);%均差表
for h=2:length(X)
    y=y+A(h,h+1)*prod(x-X(1:h-1));
end
y=collect(y,x);%合并化简
T=0.35:0.01:1.10;
f=matlabFunction(y);
plot(T,f(T));
hold on
plot(X,Y,'ro');
```

3.2.1 结果

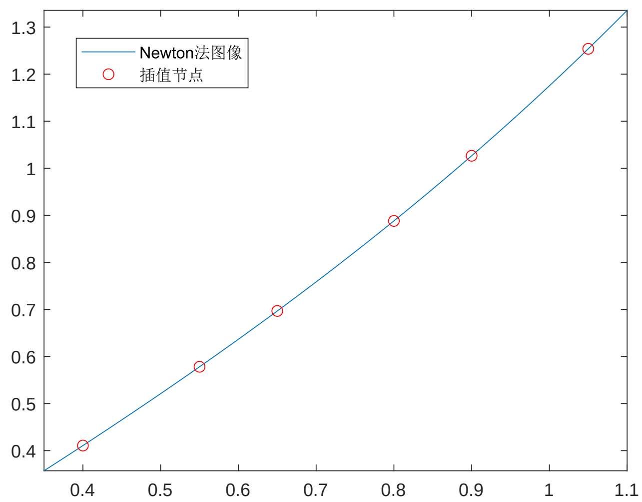


图 2: Newton 法结果

4 第四周数值分析实验

Q 4.0.1. 自行选择节点, 构造插值多项式, 并估计 $\sqrt{2}, \sqrt{2.2}, \sqrt{2.5}$ 的值.

- 不得使用牛顿切线法;
- 不得涉及无理数运算;
- 有效数字位数不少于 5 位.

4.1 三点三次 Hermite 插值

```
function y=hermite33(X,Y,i,DY_i)
%适用于n点n次Hermite插值
%X为前n点的横坐标(1\times n)
%Y为前n点的纵坐标(1\times n)
%DY_i为第i个点处的一阶导数值
    syms x a
    n=length(X);
    y=Y(1);
    A=[X',Y',zeros(n,n-1)];%均差表(l=n+1,h=n)
    for l=3:n+1 %列
        for h=l-1:n %行
            A(h,l)=(A(h,l-1)-A(h-1,l-1))/(X(h)-X(h-l+2));
        end
    end
    for h=2:length(X)
        y=y+A(h,h+1)*prod(x-X(1:h-1));%前n-1项
    end
    y1=y+a*prod(x-X(1:n));
    a1=solve(diff(y1,x)==DY_i,a);%符号运算求解
    a2=subs(a1,x,X(i));%代值已知f'(x_i)求a
    y2=y+a2*prod(x-X(1:n));
    y=collect(y2,x);%化简
end
```

4.1.1 三点三次估计

```
%% 三点三次Hermite插值方式
clc
clear
syms x
X=[1.9321 2.2201 2.5281];
Y=sqrt(X);%Y=[1.39 1.49 1.59]
i=2;
DY_i=subs(diff(sqrt(x),1),x,X(i));
y=hermite33(X,Y,i,DY_i)
X2=[2 2.2 2.5];%估计点值
y2=vpa(subs(y,x,X2),5)
% plot(X,Y,'ro');
% hold on
% t=min(X):0.01:max(X);
% f=matlabFunction(y);
% plot(t,f(t));
```

4.2 两点三次 Hermite 插值

```
function y=hermite23(X,Y,DY_all)
%2点3次Hermite插值
%X为2点的横坐标(1\times 2)
%Y为2点的纵坐标(1\times 2)
%DY_all为2点处的一阶导数值(1\times 2)
    syms x
    y=(1+2*(x-X(1))/(X(2)-X(1)))*((x-X(2))/(X(1)-X(2)))^2*Y(1)+...
        (1+2*(x-X(2))/(X(1)-X(2)))*((x-X(1))/(X(2)-X(1)))^2*Y(2)+...
        (x-X(1))*((x-X(2))/(X(1)-X(2)))^2*DY_all(1)+...
        (x-X(2))*((x-X(1))/(X(2)-X(1)))^2*DY_all(2)
    y=collect(y);
end
```

4.2.1 两点三次估计

```
%% 两点三次Hermite插值方式
```

```
clc
clear
syms x
X=[1.9321 2.5281];
Y=sqrt(X);%Y=[1.39 1.59]
DY_all=subs(diff(sqrt(x),1),x,X);
%y=hermite23(X,Y,DY_all);
syms x
y=(1+2*(x-X(1))/(X(2)-X(1)))*((x-X(2))/(X(1)-X(2)))^2*Y(1)+...
  (1+2*(x-X(2))/(X(1)-X(2)))*((x-X(1))/(X(2)-X(1)))^2*Y(2)+...
  (x-X(1))*((x-X(2))/(X(1)-X(2)))^2*DY_all(1)+...
  (x-X(2))*((x-X(1))/(X(2)-X(1)))^2*DY_all(2);
y=collect(y);
X2=[2 2.2 2.5];%估计点值
y2=vpa(subs(y,x,X2),5);
disp(y2);
% plot(X,Y,'ro');
% hold on
% t=min(X):0.01:max(X);
% f=matlabFunction(y);
% plot(t,f(t));
```

5 第六周数值分析实验

5.1 Bernstein 多项式

Definition 5.1.1. 设 $f:[0,1]\to\mathbb{R}$, 称

$$B_n(f;x)=\sum_{i=0}^{n} f\left(\frac{i}{n}\right)B_i^n(x), x\in[0,1]$$

为 f 的 n 次 *Bernstein* 多项式, 其中 $B_i^n(x)=\binom{n}{i}x^i(1-x)^{n-i}$.

```
function y=bernstein_n(f,n)
%f为[0,1]区间上的连续函数
%n为Bernstein多项式的次数
syms x
y=0;
f1=matlabFunction(f);
for k=0:1:n
    B_in=nchoosek(n,k)*x^k*(1-x)^(n-k);%Bernstein基底
    y=y+f1(k/n)*B_in;
end
y=collect(y);
end
```

Q 5.1.1. 当 $f(x)=\cos 2\pi x$ 时, 分别给出 $n=3,5,7,9,10$ 时的 $B_n(f,x)$ 表达式, 并绘制 $f(x)$ 与 $B_n(f,x)$ 的图像.

```
syms x
f=cos(2*pi*x);
N=[3 5 7 9 10];
fplot(f,[0 1]);
hold on
for n=1:5
    y=bernstein_n(f,N(n));%此处调用自定义函数
    disp(y);
    fplot(y,[0 1]);
    hold on
end
```

```
legend('$f$','$B_3(f,x)$','$B_5(f,x)$','$B_7(f,x)$','$B_9(f,x)$','$B_{10}(f,x)$','
    Interpreter','latex');
```

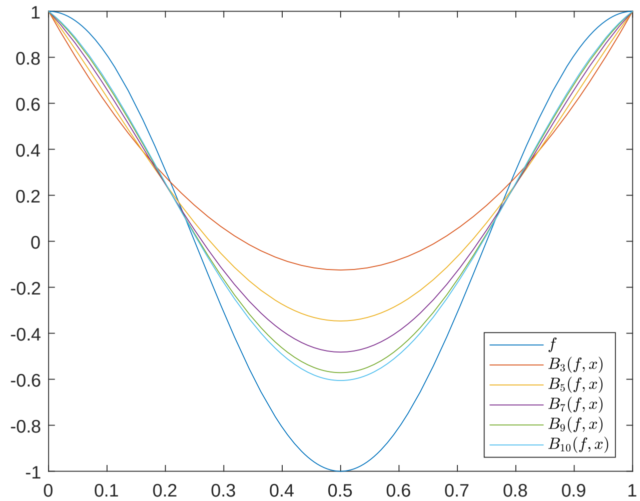


图 3: Bernstein 多项式图像

6 第七周数值分析实验

6.1 Gram-Schmidt 正交化

Q 6.1.1. 编程实现幂函数使用施密特正交化方法生成正交多项式

施密特正交化: 只要给定 $[a,b]$ 上的权函数 $\rho(x)$, 由 $\{1,x,\cdots x^n,\cdots\}$ 利用逐个正交化立得正交多项式序列.

$$\begin{aligned}
&\varphi_0(x)=1,\\
&\varphi_1(x)=x,\\
&\varphi_2(x)=x^2-\frac{(x^2,\varphi_0)}{(\varphi_0,\varphi_0)}\varphi_0(x)-\frac{(x^2,\varphi_1)}{(\varphi_1,\varphi_1)}\varphi_1(x),\\
&\varphi_n(x)=x^n-\sum_{j=0}^{n-1}\frac{(x^n,\varphi_j)}{(\varphi_j,\varphi_j)}\varphi_j(x),\quad n=3,4,\cdots.
\end{aligned}$$

或利用递推公式:

$$\varphi_{n+1}(x)=(x-\alpha_n)\,\varphi_n(x)-\beta_n\varphi_{n-1}(x),n=0,1,\cdots$$

其中

$$\begin{aligned}
&\varphi_0(x)=1,\varphi_{-1}(x)=0,\\
&\alpha_n=\frac{(x\varphi_n(x),\varphi_n(x))}{(\varphi_n(x),\varphi_n(x))},\beta_n=\frac{(\varphi_n(x),\varphi_n(x))}{(\varphi_{n-1}(x),\varphi_{n-1}(x))},\\
&\text{这里}\,(x\varphi_n(x),\varphi_n(x))=\int_a^b x\varphi_n^2(x)\rho(x)dx.
\end{aligned}$$

```
function s=Schmidt1(a,b,rho,n)
%Schmidt正交化方法
syms x
P=[1,x];
for i=2:n
    P=[P,x^i];
end
for i=2:n
    sum=0;
    for k=1:i
        sum=sum+P(k)*int(rho*x^i*P(k),a,b)/int(rho*P(k)*P(k),a,b);
    end
    P(i+1)=x^i-sum;
end
s=P;
%示例Legendre多项式
```

```
%Schmidt1(-1,1,1,6)
```

6.2 Schmidt 正交化-Legendre 多项式

```
(* ::Package:: *)

a=-1;
b=1;
rho=1;
n=7;
P=Table[x^k,{k,0,n}];
For[i=3,i<=n+1,i++,
  P[[i]]=x^(i-1)-Sum[(Integrate[rho x^(i-1) P[[j]],{x,a,b}])/(Integrate[rho P[[j
      ]]^2,{x,a,b}])P[[j]],{j,1,i-1}]
  ];
P
```

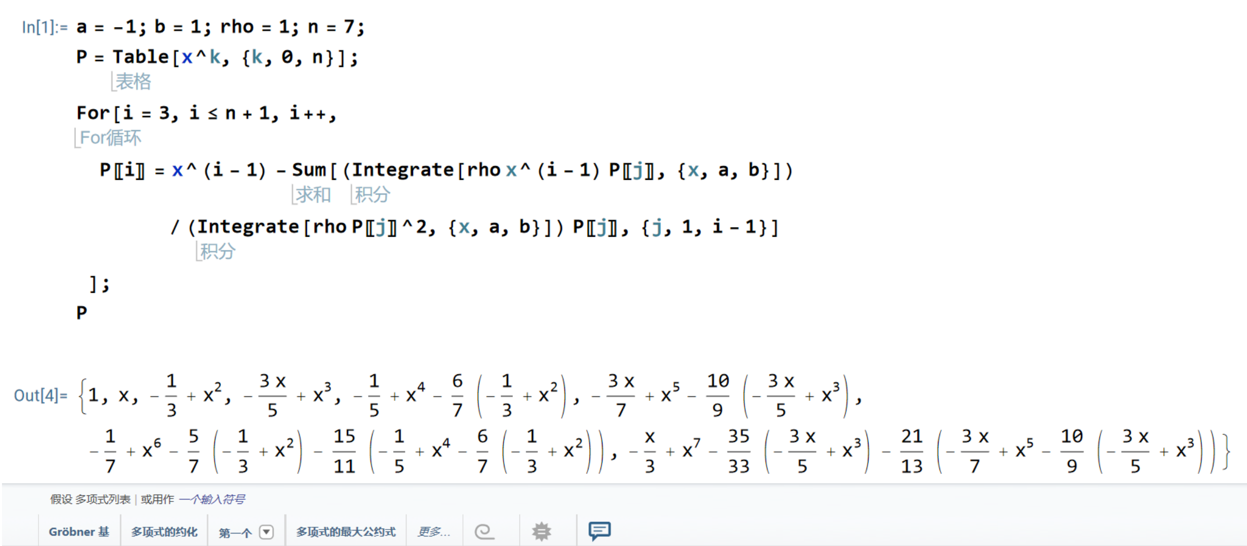


图 4: Legendre

6.3 Schmidt 正交化-Chebyshev 多项式

```
(* ::Package:: *)

a=-1;
```

```
b=1;
rho=1/Sqrt[1-x^2];
n=7;
P=Table[x^k,{k,0,n}];
For[i=3,i<=n+1,i++,
  P[[i]]=x^(i-1)-Sum[(Integrate[rho x^(i-1) P[[j]],{x,a,b}])/(Integrate[rho P[[j
     ]]^2,{x,a,b}])P[[j]],{j,1,i-1}]
  ];
P
```

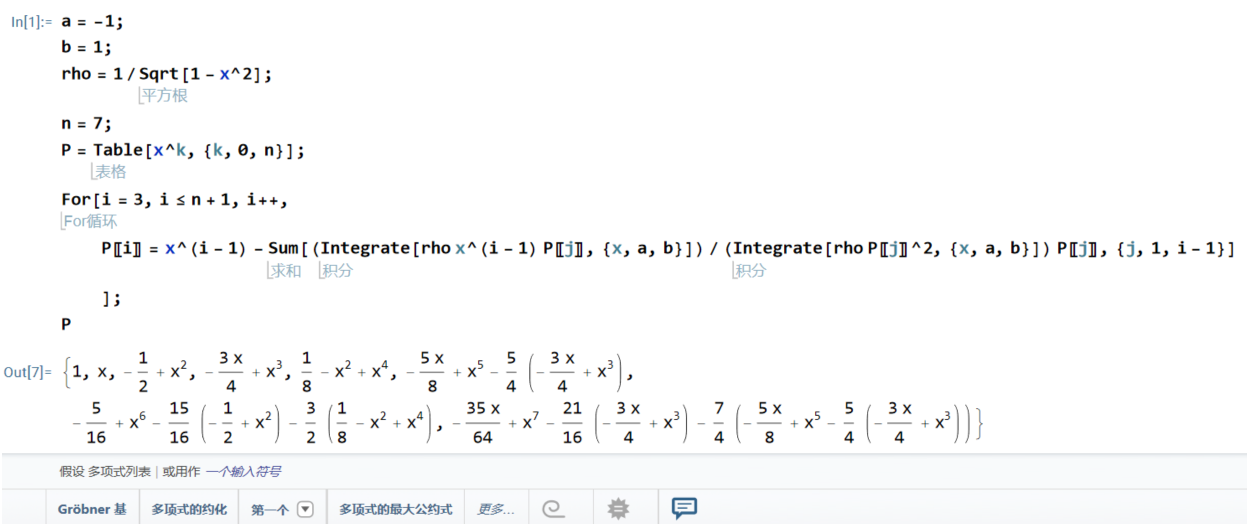


图 5: Chebyshev

6.4 递推法-Legendre 多项式

```
function y=legendremap(n)
%递推公式法求Legendre多项式
syms x;
P0=1;
P1=x;
for i=1:n-1
    P_idelete1=P0;
    P_i=P1;
    P_iadd1=(2*i+1)/(i+1).*x*P_i-i/(i+1)*P_idelete1;
    P0=P_i;
    P1=P_iadd1;
```

```
end
if n==0
    y=1;
elseif n==1
    y=x;
else
    y=collect(P_iadd1);
end
end
%示例
%legendremap(6)
```

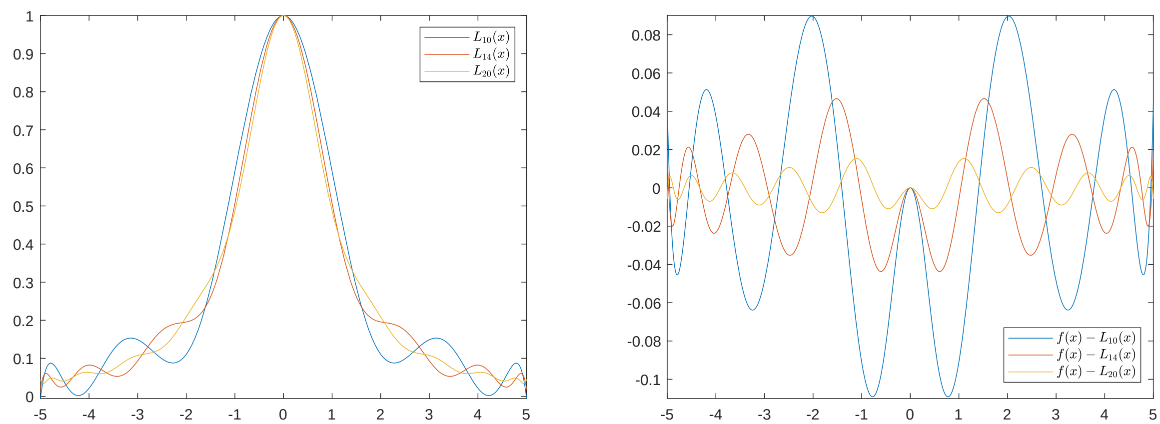
7 第八周数值分析实验

7.1 Chebyshev 多项式零点插值

Q 7.1.1. 设 $f(x)=\dfrac{1}{1+x^2}$，在 $[-5,5]$ 上分别利用 $T_{11}(x),T_{15}(x),T_{21}(x)$ 的零点作为插值点，构造 10 次，14 次，20 次插值多项式 $L_{10}(x),L_{14}(x),L_{20}(x)$，并作图表示；此外，作出误差曲线 $f(x)-L_{10}(x),f(x)-L_{14}(x),f(x)-L_{20}(x)$.

```
clc
clear
syms x
f=1/(1+x^2);
f1=matlabFunction(f);
a=-5;
b=5;
N=[11 15 21];
X=zeros(length(N),max(N));
for i=1:length(N)-1
    Y=[x,x^(i)];
end
for i=1:length(N)
    for j=1:N(i)
        X(i,j)=(b-a)/2*cos((2*(j-1)+1)/(2*N(i))*pi)+(a+b)/2;
    end
end
for i=1:length(N)
    P=X(i,1:N(i));
    R=f1(P);
    %Y(i)=lagrangeslove(P,R);%第二周作业
    s1=0;
    for l=1:length(P)
        t=1;
        for j=1:length(R)
            if j~=l
                t=t*(x-P(j))/(P(l)-P(j));
            end
        end
```

```
        s1=s1+R(l)*t;
    end
    Y(i)=collect(s1,x);
    % 绘图
    fplot(Y(i));%Lagrange插值多项式图像
    % fplot(f-Y(i));%误差曲线
    hold on
end
legend({'$L_{10}(x)$','$L_{14}(x)$','$L_{20}(x)$'},'Interpreter','latex');
% legend({'$f(x)-L_{10}(x)$','$f(x)-L_{14}(x)$','$f(x)-L_{20}(x)$'},'Interpreter','latex');
```



(a) Lagrange 插值多项式图像

(b) 误差曲线 $f(x) - L_n(x)$

图 6: 结果图示

7.2 最佳一致逼近与最佳平方逼近

Q 7.2.1. 设 $f(x) = 2x^4 - 3x^3 + 2x - 1$, 绘制 $[0, 1]$ 上的三次最佳一致逼近多项式 $P_3(x)$, 二次最佳平方逼近多项式 $S_2(x)$, 伯恩斯坦多项式 $B_1(f, x), B_2(f, x), B_3(f, x)$.

7.2.1 最佳一致逼近

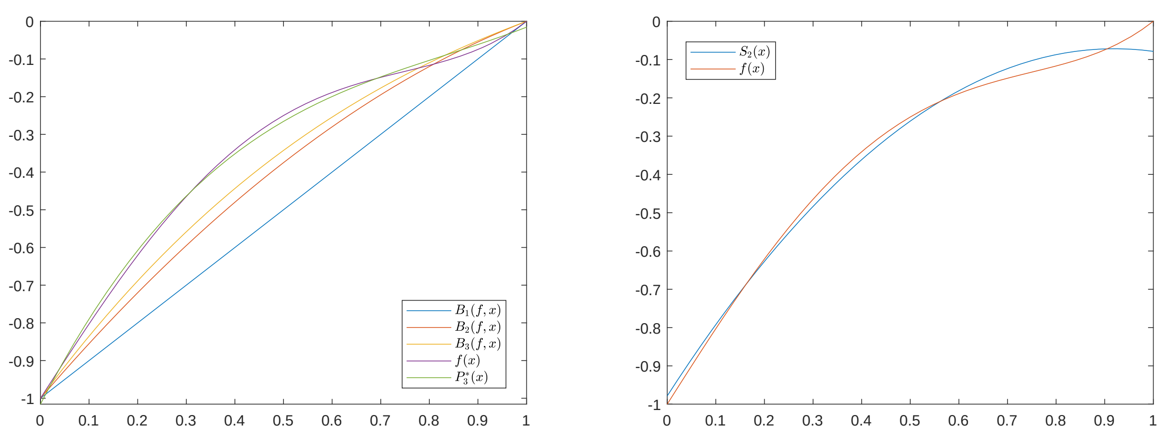
```
clc
clear
syms x t
a=0;
```

```
b=1;
f=2*x^4-3*x^3+2*x-1;
f1=matlabFunction(f);
%求解Berstein多项式
Y=[x x x];%初始化定义Berstein多项式
for i=1:3
    y=0;
    for k=0:1:i
        B_in=nchoosek(i,k)*x^k*(1-x)^(i-k);%Bernstein基底(第六周作业)
        y=y+f1(k/i)*B_in;
        Y(i)=collect(y);
    end
    fplot(Y,[a b]);
end
hold on
fplot(f,[a b]);
%求解三次最佳一致逼近多项式P_3(x)=f(x)-a_n*T'_4(x)
T_4=(8*t^4-8*t^2+1)/(2^3);%首一
x1=((b-a)*t+a+b)/2;
f2=collect(subs(f,x1),t);%t\in [-1,1]
c=coeffs(f2);
P_3=f2-c(end)*T_4;
t=(2*x-a-b)/(b-a);
P=subs(P_3,t);
hold on
fplot(P,[a b]);
legend({'$B_1(f,x)$','$B_2(f,x)$','$B_3(f,x)$','$f(x)$','$P_3^*(x)$'},'Interpreter
    ','latex');
```

7.2.2 最佳平方逼近

```
clc
clear
syms x t;
a=0;
b=1;
```

```
f=2*x^4-3*x^3+2*x-1;
x1=(b-a)/2*t+(b+a)/2;
f1=subs(f,x1);
S=0;
%直接定义Legendre多项式
L=[1 t (3*t^2-1)/2 (5*t^3-3*t)/2];
for n=0:2
    S=(2*n+1)/2*int(f1*L(n+1),-1,1)*L(n+1)+S;
end
% 递推函数法求Legendre多项式
% for n=0:2
%     S=(2*n+1)/2*int(f1*legendremap(n),-1,1)*legendremap(n)+S;
% end
t=(2*x-a-b)/(b-a);
S=subs(S,t);
S=collect(S,x);
fplot(S,[a b]);
hold on
fplot(f,[a b]);
legend({'$S_2(x)$','$f(x)$'},'Interpreter','latex');
```



(a) Berstein 多项式及 $P_3(x)$ 图像　　(b) 二次最佳平方逼近 $S_2(x)$

图 7: 结果图示

8 第九周数值分析实验

8.1 正交函数族最佳平方逼近

Q 8.1.1. 设 $f(x)=\dfrac{1}{1+25x^2}, x\in[-1,1]$, 利用勒让德正交多项式族, 分别构造 3 次, 6 次, 10 次最佳平方逼近多项式, 并绘图与 $f(x)$ 比较.

```
function y=legendremap(n)
%递推公式法
syms x;
P0=1;
P1=x;
for i=1:n-1
    P_idelete1=P0;
    P_i=P1;
    P_iadd1=(2*i+1)/(i+1).*x*P_i-i/(i+1)*P_idelete1;
    P0=P_i;
    P1=P_iadd1;
end
if n==0
    y=1;
elseif n==1
    y=x;
else
    y=collect(P_iadd1);
end
end
```

```
clc
clear
syms x
f=1/(1+25*x^2);
N=[3 6 10];
S=[x x x];
for i=1:3
    s=0;
    for j=0:N(i)
```

```
        s=s+int(legendremap(j)*f,-1,1)*legendremap(j)*(2*j+1)/2;%自定义函数
            legendremap第七周作业
    end
    S(i)=collect(s,x);
    fplot(S(i),[-1,1]);
    hold on
end
hold on
fplot(f,[-1 1]);
legend({'$S_3^*$','$S_6^*$','$S_{10}^*$','$f(x)$'},'Interpreter','latex');
```

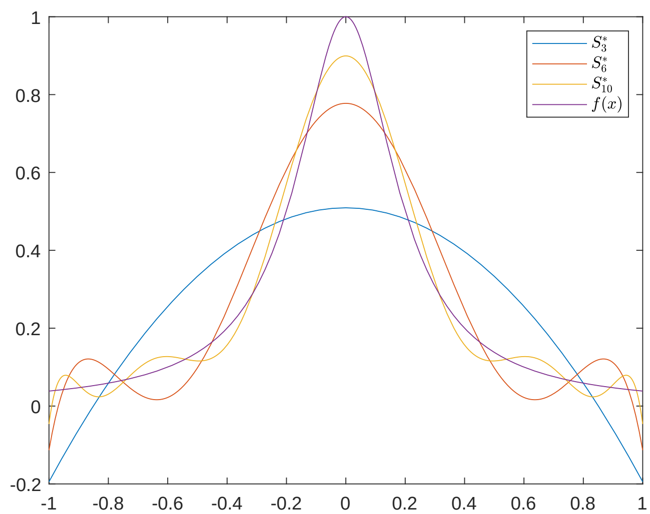


图 8: 最佳平方逼近

8.2 最小二乘拟合

Q 8.2.1. 实验数据如下

x_i	-1	-0.8	-0.6	-0.4	-0.3	-0.1	0	0.2	0.4	0.6	0.8	1
y_i	0	1.27	2.16	2.86	3.44	3.87	4.15	4.37	4.51	4.58	4.62	4.64

(1) 确定法方程组, 并求 2 次及 3 次拟合多项式函数 $s = s(x)$*;*

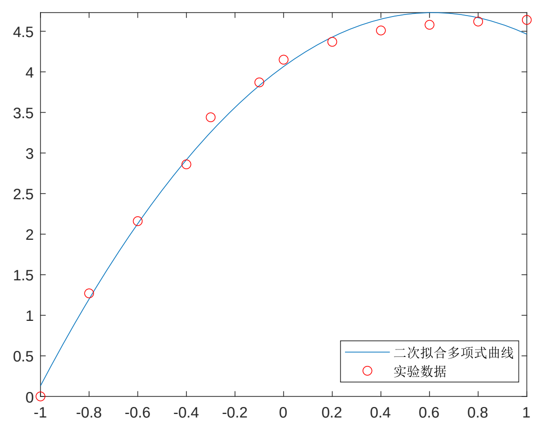
(2) 与 MATLAB 拟合工具箱 cftool 对比.

```
%% base {1,x,x^2}
clc
clear
```

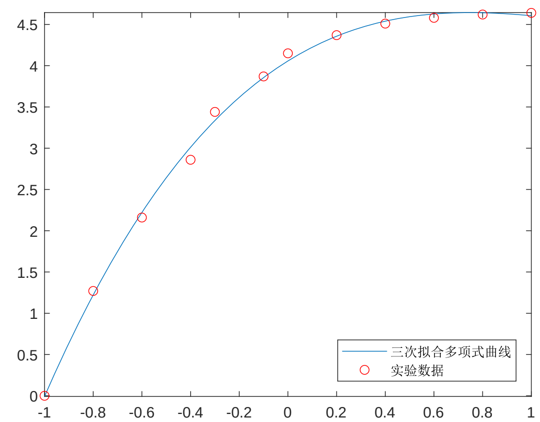
```
G=zeros(3,3);
syms x
B=[1 x x^2];
X=[-1 -0.8 -0.6 -0.4 -0.3 -0.1 0 0.2 0.4 0.6 0.8 1];
Y=[0 1.27 2.16 2.86 3.44 3.87 4.15 4.37 4.51 4.58 4.62 4.64];
%Gram Matrix
for i=2:3
    f1=matlabFunction(B(i));
    for j=2:3
        f2=matlabFunction(B(j));
        G(i,j)=f1(X)*f2(X)';
    end
end
for j=2:3
    f3=matlabFunction(B(j));
    G(1,j)=sum(f3(X));
end
G(1,1)=length(X);
for j=2:3
    G(j,1)=G(1,j);
end
D=zeros(3,1);
for i=1:3
    if i==1
        D(i,1)=sum(Y);
    else
        f4=matlabFunction(B(i));
        D(i,1)=f4(X)*Y';
    end
end
A=G\D;
y=0;
for i=1:length(A)
    y=y+A(i)*x^(i-1);
end
y=collect(y,x);
fplot(y,[min(X) max(X)]);
```

```
hold on
plot(X,Y,'ro');
legend({'二次拟合多项式曲线','实验数据'});
%% base {1,x,x^2,x^3}
clc
clear
G=zeros(4,4);
syms x
B=[1 x x^2 x^3];
X=[-1 -0.8 -0.6 -0.4 -0.3 -0.1 0 0.2 0.4 0.6 0.8 1];
Y=[0 1.27 2.16 2.86 3.44 3.87 4.15 4.37 4.51 4.58 4.62 4.64];
%Gram Matrix
for i=2:4
    f1=matlabFunction(B(i));
    for j=2:4
        f2=matlabFunction(B(j));
        G(i,j)=f1(X)*f2(X)';
    end
end
for j=2:4
    f3=matlabFunction(B(j));
    G(1,j)=sum(f3(X));
end
G(1,1)=length(X);
for j=2:4
    G(j,1)=G(1,j);
end
D=zeros(4,1);
for i=1:4
    if i==1
        D(i,1)=sum(Y);
    else
        f4=matlabFunction(B(i));
        D(i,1)=f4(X)*Y';
    end
end
A=G\D;
```

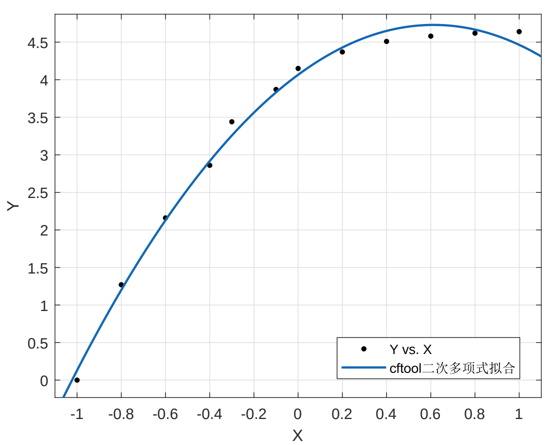
```
y=0;
for i=1:length(A)
    y=y+A(i)*x^(i-1);
end
y=collect(y,x);
fplot(y,[min(X) max(X)]);
hold on
plot(X,Y,'ro');
legend({'三次拟合多项式曲线','实验数据'});
```



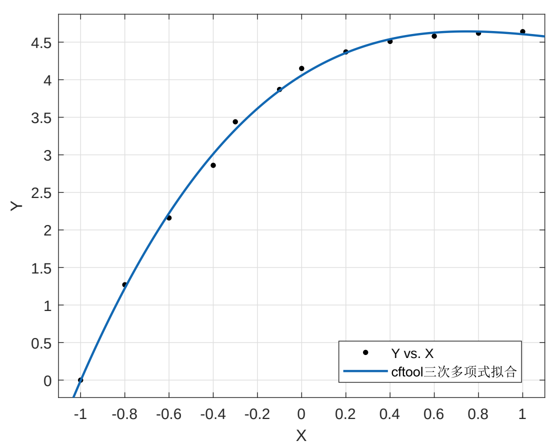
(a) 二次拟合多项式



(b) 三次拟合多项式



(c) 二次拟合多项式



(d) 三次拟合多项式

图 9: 最小二乘拟合

9 第十周数值分析中期练习

9.1 三次插值

```
%% Lagrange方法
clc
clear
X=[-2 0 1 2];
Y=[-7 1 2 9];
syms x
s1=0;
for i=1:length(X)
    t=1;
    for j=1:length(Y)
        if j~=i
           t=t*(x-X(j))/(X(i)-X(j));
        end
    end
    s1=s1+Y(i)*t;
end
y=collect(s1,x);%合并x项
disp(y);
fplot(y,[min(X) max(X)]);
hold on
plot(X,Y,'ro');
legend({'Lagrange插值法','插值节点'});
f=matlabFunction(y);
Output=f(1.3);
disp(Output);
```

9.2 最小二乘法

```
%% base {1,x}
clc
clear
G=zeros(2,2);
```

```
syms x
B=[1 x];
X=[10 11 12 13 14 15];
Y=[20 23 25 27 26 28];
%Gram Matrix
for i=2:2
    f1=matlabFunction(B(i));
    for j=2:2
        f2=matlabFunction(B(j));
        G(i,j)=f1(X)*f2(X)';
    end
end
for j=2:2
    f3=matlabFunction(B(j));
    G(1,j)=sum(f3(X));
end
G(1,1)=length(X);
for j=2:2
    G(j,1)=G(1,j);
end
D=zeros(2,1);
for i=1:2
    if i==1
        D(i,1)=sum(Y);
    else
        f4=matlabFunction(B(i));
        D(i,1)=f4(X)*Y';
    end
end
A=G\D;
y=0;
for i=1:length(A)
    y=y+A(i)*x^(i-1);
end
y=collect(y,x);
fplot(y,[min(X) max(X)]);
hold on
```

```
plot(X,Y,'ro');
legend({'一次拟合多项式曲线','实验数据'});
%% base {1,x,x^2}
clc
clear
G=zeros(3,3);
syms x
B=[1 x x^2];
X=[10 11 12 13 14 15];
Y=[20 23 25 27 26 28];
%Gram Matrix
for i=2:3
    f1=matlabFunction(B(i));
    for j=2:3
        f2=matlabFunction(B(j));
        G(i,j)=f1(X)*f2(X)';
    end
end
for j=2:3
    f3=matlabFunction(B(j));
    G(1,j)=sum(f3(X));
end
G(1,1)=length(X);
for j=2:3
    G(j,1)=G(1,j);
end
D=zeros(3,1);
for i=1:3
    if i==1
        D(i,1)=sum(Y);
    else
        f4=matlabFunction(B(i));
        D(i,1)=f4(X)*Y';
    end
end
A=G\D;
y=0;
```

```
for i=1:length(A)
    y=y+A(i)*x^(i-1);
end
y=collect(y,x);
fplot(y,[min(X) max(X)]);
hold on
plot(X,Y,'ro');
legend({'二次拟合多项式曲线','实验数据'});
```

《数值分析》中期练习

一、计算题

1. 已知$y=f(x)$的在部分节点上的函数值如下表所示:

x_i	-2	0	1	2
$f(x_i)$	-7	1	2	9

试构造三次插值多项式，并估计在$x=1.3$处的值。

2. 已知实验数据如下:

x_i	10	11	12	13	14	15
$f(x_i)$	20	23	25	27	26	28

请列出用最小二乘法求线性及二次拟合函数的方程组。

二、证明题

1. 设$f^{(n)}(x)$在区间$[a,b]$上连续，$f^{(n+1)}(x)$在(a,b)内存在，节点$a\le x_0<x_1<\cdots<x_n\le b$，$L_n(x)$是对应的插值多项式，证明：对任何$x\in[a,b]$，插值余项为

$$R_n(x)=f(x)-L_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\,\omega_{n+1}(x),$$

其中$\xi\in(a,b)$依赖于x，且$\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$.

2. 设X是实数域上的内积空间，证明对$\forall u,v\in X$，都有

$$(u,v)^2\le(u,u)(v,v).$$

3. 证明切比雪夫多项式$T_n(x)=\cos(n\arccos x),|x|\le 1\ (n=0,1,2,\cdots)$是关于权函数$\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$在区间$[-1,1]$上的正交多项式.

10 第十一周数值分析实验

10.1 牛顿—科特斯积分

Q 10.1.1. 编写牛顿—科特斯积分公式程序，思路如下：

(1) 对区间 $[a,b]$ 作 n 等份，确定数值点 x_k，求解被积函数的函数值 $y_k = f(x_k)$;

(2) 计算第 k 个科特斯系数 $C_k^{(n)} = \dfrac{(-1)^{n-k}}{nk!(n-k)!}\displaystyle\int_0^n \prod_{\substack{j=0\\ j\neq k}}^{n}(t-j)\mathrm{d}t$

(3) 构造牛顿—科特斯积分公式 $\displaystyle\int_a^b f(x)dx = (b-a)\sum_{k=0}^{n} C_k^{(n)} f(x_k)$.

(4) 利用 *MATLAB* 编写算法实现，参考函数格式为：

$$I_{out} = NewtonCotesIntegration\,(f,a,b,n)$$

```matlab
function result=NewtonCotesIntegration(f,a,b,n)
syms x t
h=(b-a)/n;
X=a:h:b;
f1=matlabFunction(f);
Y=f1(X);
S=zeros(1,n+1);
for k=0:n
    prod_c=1;
    for j=0:n
        if j~=k
            %prod_c=prod_c*(t-j)/(k-j);
            prod_c=prod_c*(t-j);
        end
    end
    int_prod=int(prod_c,0,n);
    %S(k+1)=h/(b-a)*int_prod;
    S(k+1)=(-1)^(n-k)/(n*factorial(k)*factorial(n-k))*int_prod;
end
result=vpa(sum(Y.*S)*(b-a));
end
```

例子

Q 10.1.2. 利用程序计算下列两个积分

$$I_1 = \int_1^9 \sqrt{x} dx \quad \text{及} \quad I_2 = \int_0^2 \sqrt{4-x^2}\,\mathrm{d}x$$

并比较结果 *1.* 梯形公式; *2.* 辛普森公式; *3.* 科特斯公式.

```
%% I1=\int_1^9 \sqrt(x) dx
clc
clear
syms x t
f=sqrt(x);
a=1;b=9;
f1=matlabFunction(f);
%梯形公式
I1=(f1(a)+f1(b))*(b-a)/2;
I11=NewtonCotesIntegration(f,a,b,1);
%Simpson公式
I2=(f1(a)+4*f1((a+b)/2)+f1(b))*(b-a)/6;
I21=NewtonCotesIntegration(f,a,b,2);
%柯特斯公式
I31=NewtonCotesIntegration(f,a,b,4);
%% I2=\int_0^2 \sqrt(4-x^2) dx
clc
clear
syms x t
f=sqrt(4-x^2);
a=0;b=2;
f1=matlabFunction(f);
%梯形公式
I1=(f1(a)+f1(b))*(b-a)/2;
I11=NewtonCotesIntegration(f,a,b,1);
%Simpson公式
I2=(f1(a)+4*f1((a+b)/2)+f1(b))*(b-a)/6;
I21=NewtonCotesIntegration(f,a,b,2);
%柯特斯公式
I31=NewtonCotesIntegration(f,a,b,4);
```

Q 10.1.3. 选择尽可能精确的方法计算下列积分

$$S = a\int_0^{\pi/2}\sqrt{1-\left(\frac{c}{a}\right)^2\sin^2\theta}d\theta$$

其中

$$a = (2R+H+h)/2, c = (H-h)/2,$$
$$R = 6371, \quad H = 2384, \quad h = 439$$

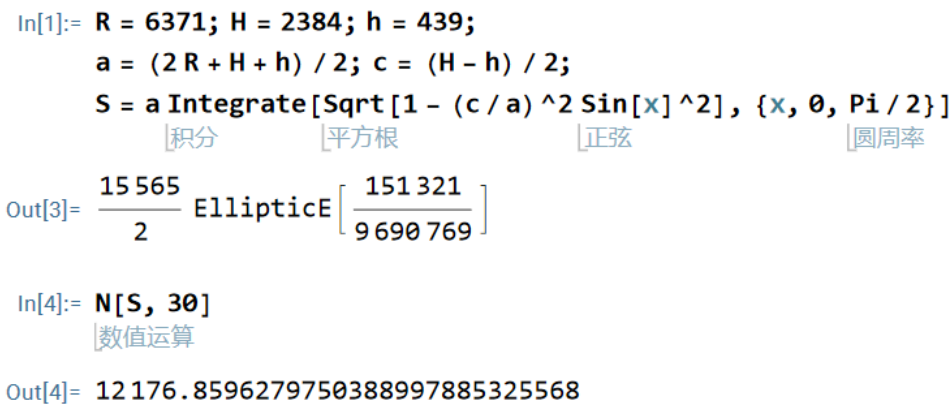


图 10: Mathematica 结果

```
clc
clear
syms x t
R=6371;H=2384;h=439;
a=(2*R+H+h)/2;c=(H-h)/2;
f=sqrt(1-(c/a)^2*sin(x)^2);
xmin=0;xmax=pi/2;
f1=matlabFunction(f);
S=12176.8596279750388997885325568;%from Mathematica N[S,30]
X=1:7;
Y=1:7;
for i=1:7
    Y(i)=S-a*NewtonCotesIntegration(f,xmin,xmax,X(i));
end
plot(X,Y,'r-o');
legend("Mathematica计算结果减NC公式计算结果");
xlabel("牛顿科特斯公式划分数目");
%% 结果
```

```
%由上则n=2时(Simpson)计算结果与Mathematica结果更接近
I=a*NewtonCotesIntegration(f,xmin,xmax,2);
disp(I);
```

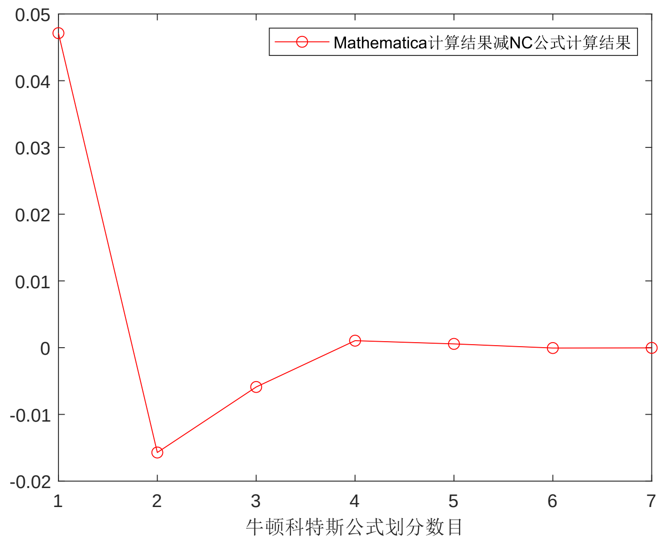


图 11: 结果比较

11 第十二周数值分析实验

11.1 复合梯形公式与复合 Simpson 公式

Q 11.1.1. 试用复合梯形公式、复合辛普森公式 ($n = 4, 8, 16, 32, 64$) 求解积分

$$I = \int_0^5 x^5 e^{-x} \sin x dx$$

已知精确值 $-e^{-5}(3660\cos 5 + 487.5\sin 5) - 15 \approx -18.8455351153$.

```
%% 复合梯形求积公式
clc
clear
syms x
y=x^5*exp(-x)*sin(x);
f=matlabFunction(y);
N=[4 8 16 32 64];
I=zeros(1,length(N));%积分结果
a=0;b=5;
for n=1:length(N)
    h=(b-a)/N(n);
    X=a:h:b;
    Y=f(X);
    I(n)=(2*sum(Y)-Y(1)-Y(length(X)))*h/2;
end
%% 复合Simpson公式
clc
clear
syms x
y=x^5*exp(-x)*sin(x);
f=matlabFunction(y);
N=[4 8 16 32 64];
I=zeros(1,length(N));%积分结果
a=0;b=5;
for n=1:length(N)
    h=(b-a)/N(n);
    X=a:h:b;
    Y=f(X);
```

```
    Z=zeros(1,length(X)-1);
    for i=1:length(Z)
        Z(i)=X(i)+h/2;
    end
    D=f(Z);
    I(n)=(4*sum(D)+2*sum(Y)-Y(1)-Y(length(X)))*h/6;
end
```

11.2 高斯一勒让德求积公式

Q 11.2.1. 试用 5 个节点的高斯一勒让德求积公式计算上例.

```
%% 高斯-勒让德求积公式
clc
clear
syms x t
y=x^5*exp(-x)*sin(x);
a=0;b=5;
x=(b-a)*t/2+(a+b)/2;
y1=subs(y,x);%区间变换
f=matlabFunction(y1);
%求解t_k
P_6=(231*t^6-315*t^4+105*t^2-5)/16;
T=vpa(solve(P_6==0));
Y=f(T);
A=[0.4679139 0.3607616 0.1713245 0.4679139 0.3607616 0.1713245];
I=A*Y*(b-a)/2;
```

11.3 复合型的高斯一勒让德求积公式

Q 11.3.1. 试用复合型的高斯一勒让德求积公式计算上例, 其中区间做 $n(n=5,10)$ 等份, 每个小区间上使用 2 个节点的 $G-L$ 公式.

```
%% 复合高斯-勒让德求积公式
clc
clear
syms x t
```

```
y=x^5*exp(-x)*sin(x);
a=0;b=5;
XK=[0.7745967 -0.7745967 0];
A=[0.5555556 0.5555556 0.8888889];
N=[5 10];
Result=zeros(1,length(N));
for n=1:length(N)
    h=(b-a)/N(n);
    X=a:h:b;
    I=zeros(1,length(X)-1);%小区间上的积分值
    for i=1:length(I)%小区间G-L
        x=h*t/2+(X(i+1)+X(i))/2;
        y1=subs(y,x);
        f=matlabFunction(y1);
        I(i)=A*f(XK)'*h/2;
    end
    Result(n)=sum(I);
end
```

12 第十三周数值分析实验

12.1 矩阵的 LU 分解

Q 12.1.1. 实现 n 阶非奇异矩阵 A 的 LU 分解程序, 函数格式为: $[L, U] = lu\_decomposition(A)$ 并计算 $P153$, 例 5.

```
function [L,U]=lu_decomposition(A)
% n=length(A);
% L=eye(n)+zeros(n,n);
% U=[A(1,:);zeros(n-1,n)];
% %L(1,:)
% for i=1:n
%     L(i,1)=A(i,1)/A(1,1);
% end
% %U(:,2)
% for j=2:n
%     U(2,j)=A(2,j)-L(2,1)*A(1,j);
% end
% %L(2,:)
% for i=1:n
%     L(i,2)=
% end
[m,n]=size(A);
L=eye(m);
L(:,1)=A(:,1)/A(1,1);%L第一列赋值
U=zeros(m,n);
U(1,:)=A(1,:);%U第一行赋值
for i=2:m
    for j=2:n
        if i<=j
            U(i,j)=A(i,j)-sum(L(i,1:i-1).*U(1:i-1,j)');
        else
            if U(j,j)==0
                L(i,j)=0;
            else
                L(i,j)=(A(i,j)-sum(L(i,1:j-1).*U(1:j-1,j)'))/U(j,j);
            end
```

```
        end
    end
end
end
```

```
%page153ex5
clc
clear
A=[1 2 3;2 5 2;3 1 5];
b=[14;18;20];
%使用内置函数lu
[L,U]=lu(A);
y=L\b;
x=U\y;
%使用lu_decomposition
[L1,U1]=lu_decomposition(A);
y1=L1\b;
x1=U1\y1;
%直接计算
x2=A\b;
```

12.2 矩阵的 Cholesky 分解

Q 12.2.1. *实现 n 阶对称正定矩阵 A 的 Cholesky 分解程序，函数格式为：$L = cholesky\_Factorization(A)$ 并计算 P177 第 9 题. (MATLAB 自带程序为 chol(A)).*

```
function L=cholesky_Factorization(A)
N=size(A);%记录矩阵的大小
m=N(1);%矩阵的行数
n=N(2);%矩阵的列数
L=zeros(m,n);%初始化L矩阵
for i=1:n
    L(i:n,i)=A(i:n,i);%将A的下半矩阵赋值给L
end
L(1,1)=sqrt(A(1,1));%步骤一：先算矩阵的第一个元素
L(2:n,1)=L(2:n,1)/L(1,1);%步骤二：利用第一个元素计算第一列元素
for k=2:n
```

```
    L(k,k)=sqrt(A(k,k)-L(k,k-1)*(L(k,k-1)));%步骤三：计算所有的对角线元素
    for j=k+1:n
        L(j,k)=(A(j,k)-L(j,k-1)*L(k,k-1))/L(k,k);%步骤四：计算对角线以下元素的值
    end
end
end
```

```
%page177ex9
clc
clear
A=[2 -1 0 0 0;-1 2 -1 0 0;0 -1 2 -1 0;0 0 -1 2 -1;0 0 0 -1 2];
b=[1;0;0;0;0];
%使用内置函数chol
L=chol(A);%L'L=A
y=L'\b;
x=L\y;
%使用cholesky_Factorization
L1=cholesky_Factorization(A);%L1L1'=A
y1=L1\b;
x1=L1'\y1;
%直接计算
x2=A\b;
```