数值分析作业

CY.SP

2023 年 10 月 23 日

目录

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);%阶乘函数
end
result=sum(s)
```

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 $I_n=e^{-1}\int_0^1 x^n e^x dx, (n=0,1,2,\cdots)$ 并估计误差

方法 1: 利用递推公式 (A)

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

方法 2: 利用递推公式 (B)

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

```
%% 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
%% 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
```

Q 1.2.2. 计算 $I_n = \int_0^1 \frac{x^n}{x+5}dx, (n = 0, 1, 2, \cdots 20)$ 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
%% 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
```

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.

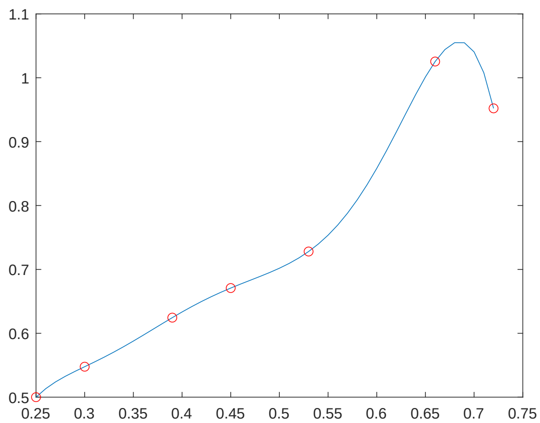
2.2.1 直接法计算

```
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)
```

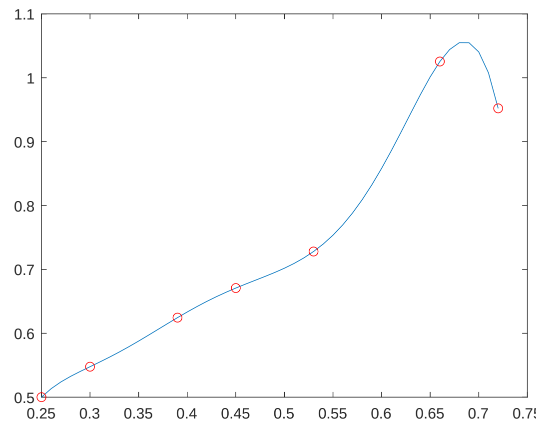
2.2.2 Lagrange 法计算

```
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

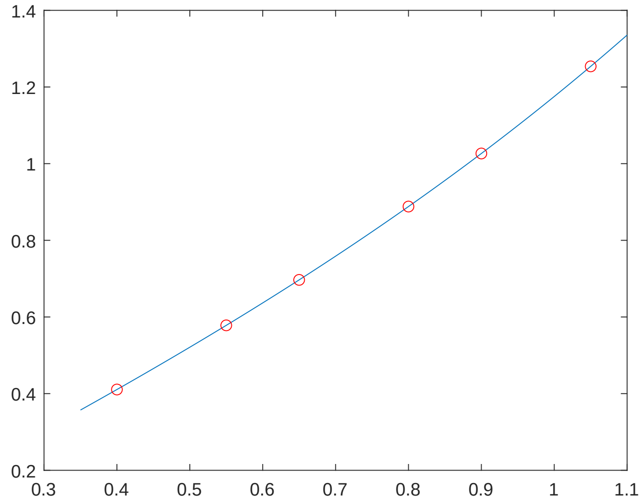


图 2: Newton 法结果

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)
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));
```

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];
X=0:0.01:1;
f1=matlabFunction(f);
plot(X,f1(X));
hold on
for n=1:5
    y=bernstein_n(f,N(n));
    disp(y);
    fy=matlabFunction(y);
```

```
    plot(X,fy(X));
    hold on
end
legend('f','n=3','n=5','n=7','n=9','n=10');
```

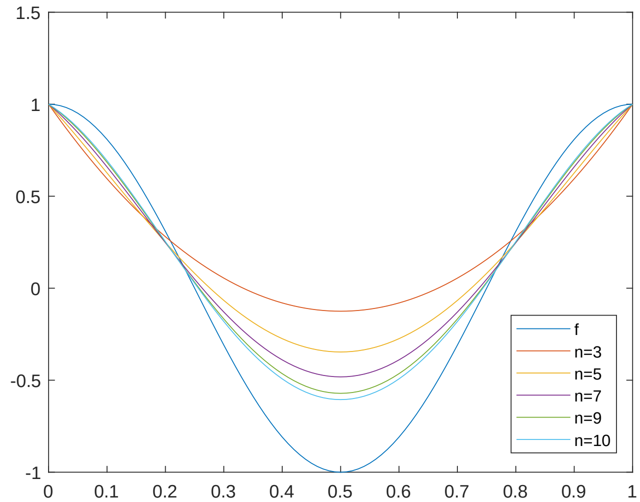


图 3: Bernstein 多项式图像

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

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

6.2 直接法-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
```

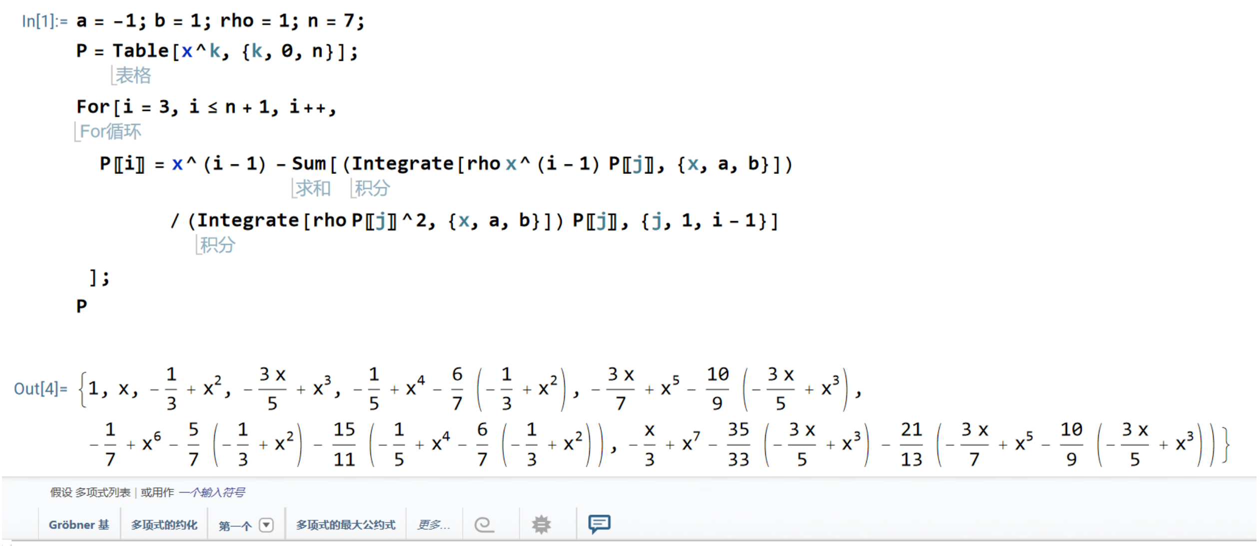


图 4: Legendre

6.3 直接法-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
```

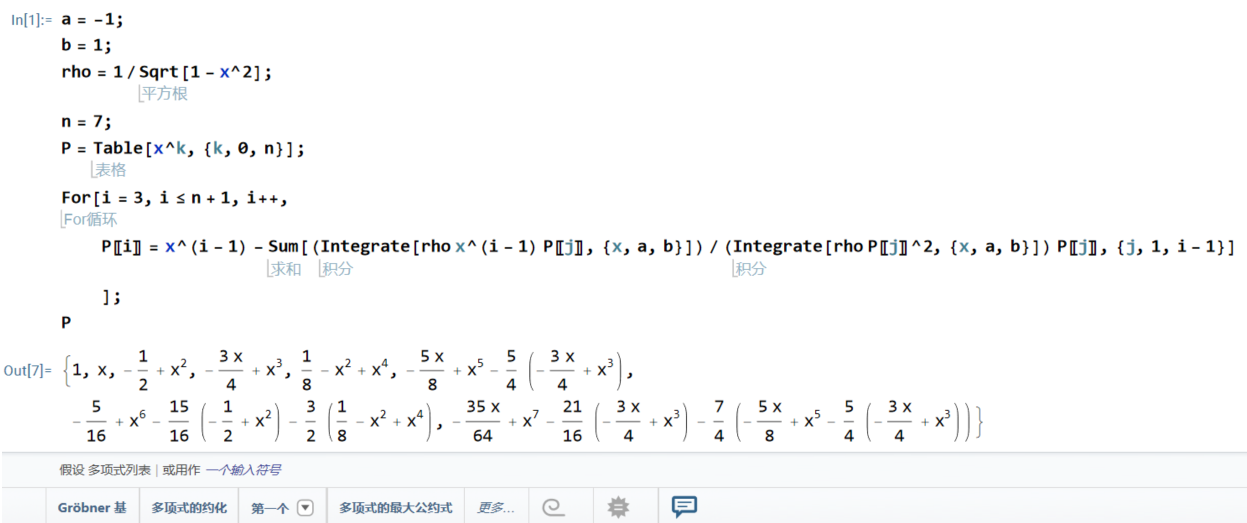


图 5: Chebyshev

6.4 递推法-Legendre 多项式

```
function y=legendremap(n)
%递推公式法
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
```

```
%% P71ex7
syms x;
f=exp(x);
S=0;
for n=0:3
    S=(2*n+1)/2*int(f*legendremap(n),-1,1)*legendremap(n)+S;
end
collect(S,x)
fplot(S,[-1,1]);
hold on
fplot(f,[-1 1]);
```

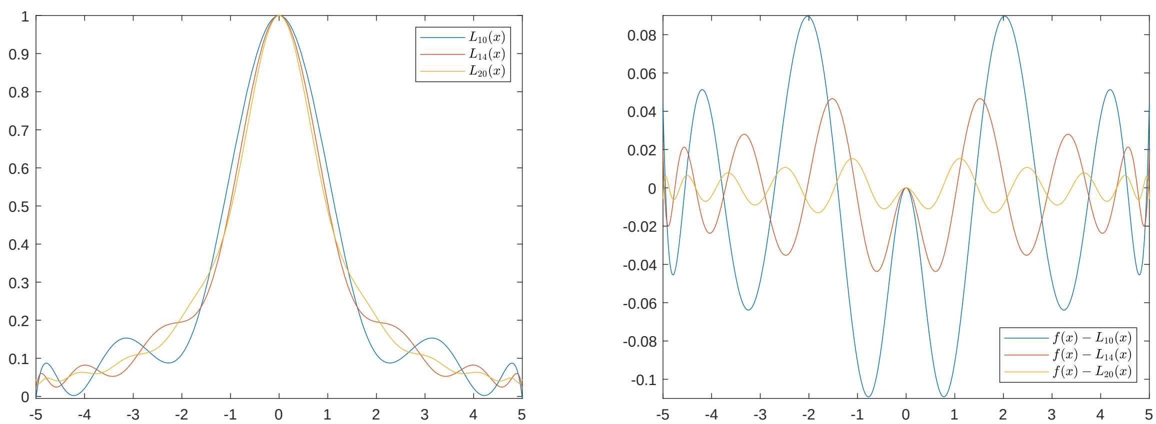
7 第八周数值分析实验

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;
```

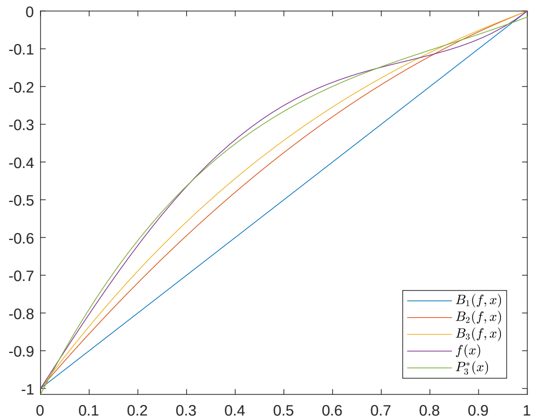
```
b=1;
f=2*x^4-3*x^3+2*x-1;
f1=matlabFunction(f);
%求解Berstein多项式
Y=[x x x];%初始化定义Berstein多项式
for i=1:3
    y=0;
    for k=0:1:i
        B_in=nchoosek(i,k)*x^k*(1-x)^(i-k);%Bernstein基底(第六周作业)
        y=y+f1(k/i)*B_in;
        Y(i)=collect(y);
    end
    fplot(Y,[a b]);
end
hold on
fplot(f,[a b]);
%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

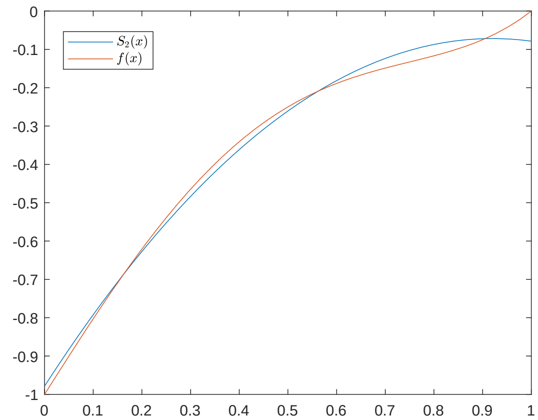
```
function y=legendremap(n)
%递推公式法
syms t;
P0=1;
P1=t;
```

```
for i=1:n-1
    P_idelete1=P0;
    P_i=P1;
    P_iadd1=(2*i+1)/(i+1).*t*P_i-i/(i+1)*P_idelete1;
    P0=P_i;
    P1=P_iadd1;
end
if n==0
    y=1;
elseif n==1
    y=t;
else
    y=collect(P_iadd1);
end
end
```

```
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;
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);
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