

# 数值分析作业\*

CY.SP

2023 年 11 月 28 日

## 目录

<b>1 第一周数值分析实验</b>	<b>3</b>	<b>5 第六周数值分析实验</b>	<b>14</b>
1.1 第一节: 级数求和与二元函数绘图 . . . . .	3	5.1 Bernstein 多项式 . . . . .	14
1.2 第二节: 积分递推公式 . . . . .	3	<b>6 第七周数值分析实验</b>	<b>16</b>
<b>2 第二周数值分析实验</b>	<b>6</b>	6.1 Gram-Schmidt 正交化 . . . . .	16
2.1 第一节: 插值 function . . . . .	6	6.2 Schmidt 正交化-Legendre 多项式 . . . . .	17
2.1.1 直接法 . . . . .	6	6.3 Schmidt 正交化-Chebyshev 多项式 . . . . .	17
2.1.2 Lagrange 法 . . . . .	6	6.4 递推法-Legendre 多项式 . . . . .	18
2.2 第二节: 计算实例 . . . . .	7	<b>7 第八周数值分析实验</b>	<b>20</b>
2.2.1 直接法计算 . . . . .	7	7.1 Chebyshev 多项式零点插值 . . . . .	20
2.2.2 Lagrange 法计算 . . . . .	7	7.2 最佳一致逼近与最佳平方逼近 . . . . .	21
2.2.3 结果 . . . . .	8	7.2.1 最佳一致逼近 . . . . .	21
<b>3 第三周数值分析实验</b>	<b>9</b>	7.2.2 最佳平方逼近 . . . . .	22
3.1 Newton 法均差表 function . . . . .	9	<b>8 第九周数值分析实验</b>	<b>24</b>
3.2 page32ex4 实例 . . . . .	9	8.1 正交函数族最佳平方逼近 . . . . .	24
3.2.1 结果 . . . . .	10	8.2 最小二乘拟合 . . . . .	25
<b>4 第四周数值分析实验</b>	<b>11</b>	<b>9 第十周数值分析中期练习</b>	<b>29</b>
4.1 三点三次 Hermite 插值 . . . . .	11	9.1 三次插值 . . . . .	29
4.1.1 三点三次估计 . . . . .	11	9.2 最小二乘法 . . . . .	29
4.2 两点三次 Hermite 插值 . . . . .	12	<b>10 第十一周数值分析实验</b>	<b>34</b>
4.2.1 两点三次估计 . . . . .	12	10.1 牛顿-科特斯积分 . . . . .	34

\*Github 仓库

<b>11 第十二周数值分析实验</b>	<b>38</b>	<b>12 第十三周数值分析实验</b>	<b>41</b>
11.1 复合梯形公式与复合 Simpson 公式	38	12.1 矩阵的 LU 分解 . . . . .	41
11.2 高斯-勒让德求积公式 . . . . .	39	12.2 矩阵的 Cholesky 分解 . . . . .	42
11.3 复合型的高斯-勒让德求积公式 . .	39		

## 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.

```
1 clc
2 clear
3 s=zeros(1,12);
4 for i=1:12
5     %s(i)=factorial(i);%使用阶乘函数
6     s(i)=prod(1:i);
7 end
8 result=sum(s);
9 disp(result);
```

Q 1.1.2. 绘制二元函数图像.

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

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

### 1.2 第二节: 积分递推公式

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

方法 1: 利用递推公式 (A)

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

方法 2: 利用递推公式 (B)

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

```

1 %% 第一种递推方式
2 clc
3 clear
4 N=20;
5 IA=zeros(1,N);%积分值
6 EA=zeros(1,N);%误差估计
7 IO=0.6321;%by Mathematica I_0=0.632121...
8 EA(1)=5E-5;
9 IA(1)=1-IO;
10 for n=1:N-1
11     IA(n+1)=1-(n+1)*IA(n);% k from 1 to 20
12     EA(n+1)=n*EA(n);
13 end
14 disp(IA);
15 disp(EA);
16 %% Equation 第二种递推方式
17 clc
18 clear
19 N=20;
20 EB=zeros(1,N);%误差估计
21 IB=zeros(1,N);%积分值
22 IB(20)=0.0684;%by Mathematica I_{20}=0.0455449...
23 EB(20)=5E-2;
24 for n=N:-1:2
25     IB(n-1)=1/n*(1-IB(n));% k from 1 to 20
26     EB(n-1)=1/n*EB(n);
27 end
28 disp(IB);
29 disp(EB);

```

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

```

1 %% 第一种递推方式
2 clc
3 clear
4 N=20;

```

```
5 IA=zeros(1,N);%积分值
6 EA=zeros(1,N);%误差值
7 I0=0.1823;%by Mathematica I_0=0.182322...
8 EA(1)=5E-6;
9 IA(1)=1-5*I0;
10 for n=1:N-1
11     IA(n+1)=1/(n+1)-5*IA(n);% k from 1 to 20
12     EA(n+1)=5*EA(n);
13 end
14 disp(IA);
15 disp(EA);
16 %% 第二种递推方式
17 clc
18 clear
19 N=20;
20 EB=zeros(1,N);%误差值
21 IB=zeros(1,N);%积分值
22 IB(20)=0.00799;%by Mathematica I_{20}=0.00799752...
23 EB(20)=5E-5;
24 for n=N:-1:2
25     IB(n-1)=1/(5*n)-1/5*IB(n);% k from 1 to 20
26     EB(n-1)=1/5*EB(n);
27 end
28 disp(IB);
29 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

```
1 function y=Vslove(a,B) %slove Matrix AX=B
2 %a=matrix of 1\times n
3 %B=matrix of n\times 1
4     syms x
5     y=0;
6     A=zeros(length(a),length(a));
7     A=vander(a);
8     X=zeros(length(B'),1);%定义X矩阵n\times 1 即为多项式系数
9     X=inv(A)*B;
10    for i=1:length(X')
11        y=y+X(length(X')+1-i)*x^(i-1);
12    end
13    y=collect(y,x);%合并x项
14    t=min(a):0.01:max(a);
15    y1=matlabFunction(y);
16    plot(t,y1(t));
17    hold on
18    plot(a,B,'ro');
19 end
```

#### 2.1.2 Lagrange 法

```
1 function y=lagrangeslove(X,Y)
2 %X=matrix of 1\times n 各X坐标
3 %Y=matrix of 1\times n 相应的Y
4     syms x
5     s1=0;
6     for i=1:length(X)
7         t=1;
8         for j=1:length(Y)
9             if j~=i
```

```

10         t=t*(x-X(j))/(X(i)-X(j));
11     end
12 end
13     s1=s1+Y(i)*t;
14 end
15     y=collect(s1,x);%合并x项
16     z=min(X):0.01:max(X);
17     s2=matlabFunction(s1);
18     plot(z,s2(z));
19     hold on
20     plot(X,Y,'ro');
21 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

```

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

```

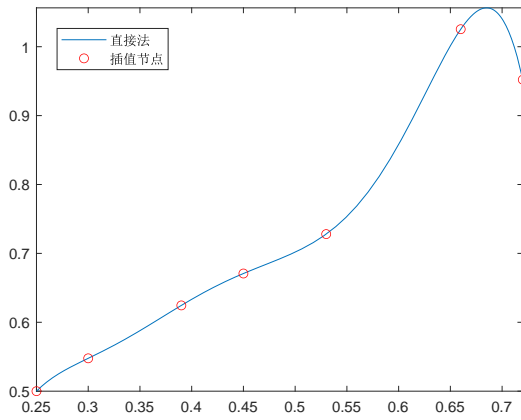
### 2.2.2 Lagrange 法计算

```

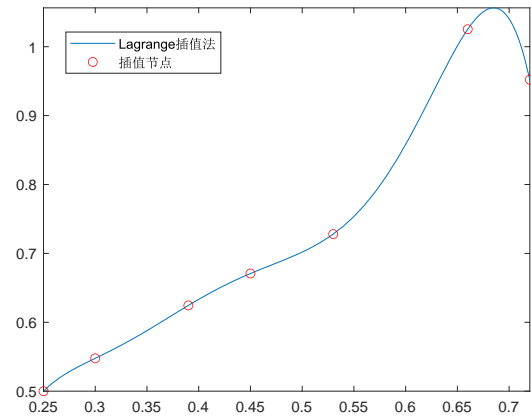
1 clc
2 clear
3 a=[0.25, 0.3, 0.39, 0.45, 0.53, 0.66, 0.72];
4 B=[0.5, 0.5477, 0.6245, 0.6708, 0.7280, 1.0254, 0.9521];
5 y=lagrangeslove(a,B)

```

## 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

```
1 function A=newtonmatrix(X,Y)
2 %均差表
3 %X为所有x_i所构成的行向量
4 %Y为所有y_i所构成的行向量
5 n=length(X);
6 A=[X',Y',zeros(n,n-1)];%l=n+1,h=n
7 for l=3:n+1 %列
8     for h=1-1:n %行
9         A(h,l)=(A(h,l-1)-A(h-1,l-1))/(X(h)-X(h-1+2));
10    end
11 end
12 end
```

#### 3.2 page32ex4 实例

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

## 3.2.1 结果

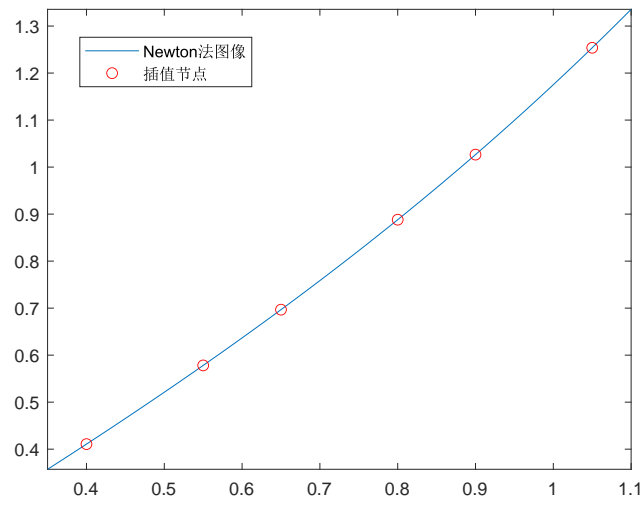


图 2: Newton 法结果

## 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

```

1 function y=hermite33(X,Y,i,DY_i)
2 %适用于n点n次Hermite插值
3 %X为前n点的横坐标(1\times n)
4 %Y为前n点的纵坐标(1\times n)
5 %DY_i为第i个点处的一阶导数值
6     syms x a
7     n=length(X);
8     y=Y(1);
9     A=[X',Y',zeros(n,n-1)];%均差表(l=n+1,h=n)
10    for l=3:n+1 %列
11        for h=1-1:n %行
12            A(h,l)=(A(h,l-1)-A(h-1,l-1))/(X(h)-X(h-1+2));
13        end
14    end
15    for h=2:length(X)
16        y=y+A(h,h+1)*prod(x-X(1:h-1));%前n-1项
17    end
18    y1=y+a*prod(x-X(1:n));
19    a1=solve(diff(y1,x)==DY_i,a);%符号运算求解
20    a2=subs(a1,x,X(i));%代值已知f'(x_i)求a
21    y2=y+a2*prod(x-X(1:n));
22    y=collect(y2,x);%化简
23 end

```

#### 4.1.1 三点三次估计

```

1 %% 三点三次Hermite插值方式
2 clc
3 clear
4 syms x
5 X=[1.9321 2.2201 2.5281];
6 Y=sqrt(X);%Y=[1.39 1.49 1.59]
7 i=2;
8 DY_i=subs(diff(sqrt(x),1),x,X(i));
9 y=hermite33(X,Y,i,DY_i)
10 X2=[2 2.2 2.5];%估计点值
11 y2=vpa(subs(y,x,X2),5)
12 % plot(X,Y,'ro');
13 % hold on
14 % t=min(X):0.01:max(X);
15 % f=matlabFunction(y);
16 % plot(t,f(t));

```

## 4.2 两点三次 Hermite 插值

```

1 function y=hermite23(X,Y,DY_all)
2 %2点3次Hermite插值
3 %X为2点的横坐标(1\times 2)
4 %Y为2点的纵坐标(1\times 2)
5 %DY_all为2点处的一阶导数值(1\times 2)
6     syms x
7     y=(1+2*(x-X(1))/(X(2)-X(1)))*((x-X(2))/(X(1)-X(2)))^2*Y(1)+...
8         (1+2*(x-X(2))/(X(1)-X(2)))*((x-X(1))/(X(2)-X(1)))^2*Y(2)+...
9         (x-X(1))*((x-X(2))/(X(1)-X(2)))^2*DY_all(1)+...
10        (x-X(2))*((x-X(1))/(X(2)-X(1)))^2*DY_all(2)
11     y=collect(y);
12 end

```

### 4.2.1 两点三次估计

```

1 %% 两点三次Hermite插值方式

```

```
2  clc
3  clear
4  syms x
5  X=[1.9321 2.5281];
6  Y=sqrt(X);%Y=[1.39 1.59]
7  DY_all=subs(diff(sqrt(x),1),x,X);
8  %y=hermite23(X,Y,DY_all);
9  syms x
10 y=(1+2*(x-X(1))/(X(2)-X(1)))*((x-X(2))/(X(1)-X(2)))^2*Y(1)+...
11    (1+2*(x-X(2))/(X(1)-X(2)))*((x-X(1))/(X(2)-X(1)))^2*Y(2)+...
12    (x-X(1))*((x-X(2))/(X(1)-X(2)))^2*DY_all(1)+...
13    (x-X(2))*((x-X(1))/(X(2)-X(1)))^2*DY_all(2);
14 y=collect(y);
15 X2=[2 2.2 2.5];%估计点值
16 y2=vpa(subs(y,x,X2),5);
17 disp(y2);
18 % plot(X,Y,'ro');
19 % hold on
20 % t=min(X):0.01:max(X);
21 % f=matlabFunction(y);
22 % 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  $f : [0, 1] \rightarrow \mathbb{R}$ , 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  $B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}$ .

```

1 function y=bernstein_n(f,n)
2 %f为[0,1]区间上的连续函数
3 %n为Bernstein多项式的次数
4 syms x
5 y=0;
6 f1=matlabFunction(f);
7 for k=0:1:n
8     B_in=nchoosek(n,k)*x^k*(1-x)^(n-k);%Bernstein基底
9     y=y+f1(k/n)*B_in;
10 end
11 y=collect(y);
12 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.

```

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

```

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

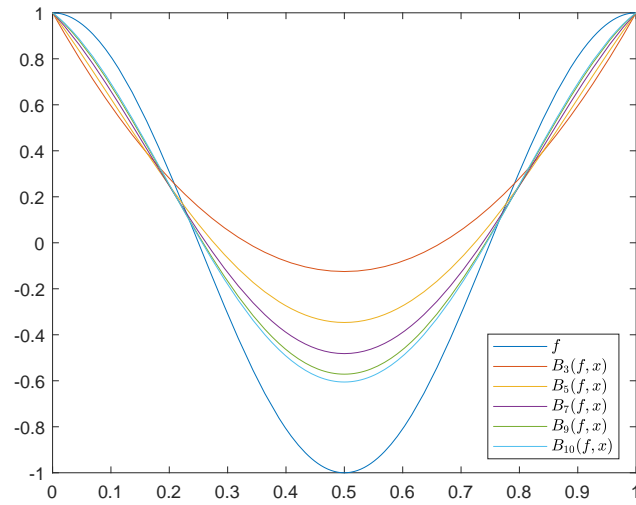


图 3: Bernstein 多项式图像

## 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  $\{1, x, \dots, x^n, \dots\}$  利用逐个正交化立得正交多项式序列.

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

或利用递推公式:

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

其中

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

```
1 function s=Schmidt1(a,b,rho,n)
2 %Schmidt正交化方法
3 syms x
4 P=[1,x];
5 for i=2:n
6     P=[P,x^i];
7 end
8 for i=2:n
9     sum=0;
10    for k=1:i
11        sum=sum+P(k)*int(rho*x^i*P(k),a,b)/int(rho*P(k)*P(k),a,b);
12    end
13    P(i+1)=x^i-sum;
14 end
15 s=P;
16 %示例Legendre多项式
```



```
17 %Schmidt1(-1,1,1,6)
```

## 6.2 Schmidt 正交化-Legendre 多项式

```
1 (* ::Package:: *)
2
3 a=-1;
4 b=1;
5 rho=1;
6 n=7;
7 P=Table[x^k,{k,0,n}];
8 For[i=3,i<=n+1,i++,
9   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}]
10 ];
11 P
```

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

```
Out[4]= {1, x, -1/3 + x^2, -3x/5 + x^3, -1/5 + x^4 - 6/7 (-1/3 + x^2), -3x/7 + x^5 - 10/9 (-3x/5 + x^3),
-1/7 + x^6 - 5/7 (-1/3 + x^2) - 15/11 (-1/5 + x^4 - 6/7 (-1/3 + x^2)), -x/3 + x^7 - 35/33 (-3x/5 + x^3) - 21/13 (-3x/7 + x^5 - 10/9 (-3x/5 + x^3))}
```

假设 多项式列表 | 或用作 一个输入符号

Gröbner 基 多项式的约化 第一个 多项式的最大公约式 更多...

图 4: Legendre

## 6.3 Schmidt 正交化-Chebyshev 多项式

```
1 (* ::Package:: *)
2
3 a=-1;
```

```

4 b=1;
5 rho=1/Sqrt[1-x^2];
6 n=7;
7 P=Table[x^k,{k,0,n}];
8 For[i=3,i<=n+1,i++,
9   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}]
10 ];
11 P

```

```

In[1]:= 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

```

Out[7]=  $\left\{1, x, -\frac{1}{2} + x^2, -\frac{3x}{4} + x^3, \frac{1}{8} - x^2 + x^4, -\frac{5x}{8} + x^5 - \frac{5}{4} \left(-\frac{3x}{4} + x^3\right), -\frac{5}{16} + x^6 - \frac{15}{16} \left(-\frac{1}{2} + x^2\right) - \frac{3}{2} \left(\frac{1}{8} - x^2 + x^4\right), -\frac{35x}{64} + x^7 - \frac{21}{16} \left(-\frac{3x}{4} + x^3\right) - \frac{7}{4} \left(-\frac{5x}{8} + x^5 - \frac{5}{4} \left(-\frac{3x}{4} + x^3\right)\right)\right\}$

假设 多项式列表 | 或用作 一个输入符号




Gröbner 基 多项式的约化 第一个 ▾ 多项式的最大公约式 更多...   

图 5: Chebyshev

## 6.4 递推法-Legendre 多项式

```

1 function y=legendremap(n)
2 %递推公式法求Legendre多项式
3 syms x;
4 P0=1;
5 P1=x;
6 for i=1:n-1
7   P_idete1=P0;
8   P_i=P1;
9   P_iadd1=(2*i+1)/(i+1).*x*P_i-i/(i+1)*P_idete1;
10  P0=P_i;
11  P1=P_iadd1;

```

```
12 end
13 if n==0
14     y=1;
15 elseif n==1
16     y=x;
17 else
18     y=collect(P_iadd1);
19 end
20 end
21 %示例
22 %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  $f(x) = \frac{1}{1+x^2}$ , 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  $f(x) - L_{10}(x), f(x) - L_{14}(x), f(x) - L_{20}(x)$ .

```

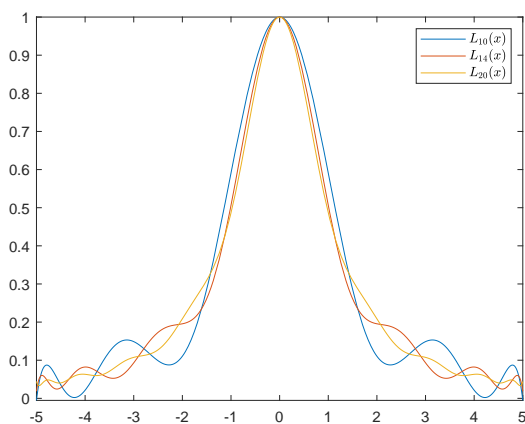
1  clc
2  clear
3  syms x
4  f=1/(1+x^2);
5  f1=matlabFunction(f);
6  a=-5;
7  b=5;
8  N=[11 15 21];
9  X=zeros(length(N),max(N));
10 for i=1:length(N)-1
11     Y=[x,x^(i)];
12 end
13 for i=1:length(N)
14     for j=1:N(i)
15         X(i,j)=(b-a)/2*cos((2*(j-1)+1)/(2*N(i))*pi)+(a+b)/2;
16     end
17 end
18 for i=1:length(N)
19     P=X(i,1:N(i));
20     R=f1(P);
21     %Y(i)=lagrangeslove(P,R);%第二周作业
22     s1=0;
23     for l=1:length(P)
24         t=1;
25         for j=1:length(R)
26             if j~=l
27                 t=t*(x-P(j))/(P(l)-P(j));
28             end
29         end

```

```

30     s1=s1+R(1)*t;
31     end
32     Y(i)=collect(s1,x);
33     % 绘图
34     fplot(Y(i));%Lagrange插值多项式图像
35     % fplot(f-Y(i));%误差曲线
36     hold on
37 end
38 legend({'$L_{10}(x)$','$L_{14}(x)$','$L_{20}(x)$'}, 'Interpreter', 'latex');
39 % 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

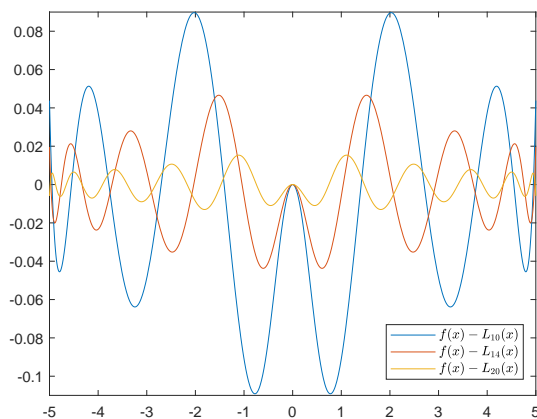
(b) 误差曲线  $f(x) - L_n(x)$ 

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

```

1  clc
2  clear
3  syms x t
4  a=0;

```

```

5 b=1;
6 f=2*x^4-3*x^3+2*x-1;
7 f1=matlabFunction(f);
8 %求解Bernstein多项式
9 Y=[x x x];%初始化定义Bernstein多项式
10 for i=1:3
11     y=0;
12     for k=0:1:i
13         B_in=nchoosek(i,k)*x^k*(1-x)^(i-k);%Bernstein基底(第六周作业)
14         y=y+f1(k/i)*B_in;
15         Y(i)=collect(y);
16     end
17     fplot(Y,[a b]);
18 end
19 hold on
20 fplot(f,[a b]);
21 %求解三次最佳一致逼近多项式 $P_3(x)=f(x)-a_n*T'_4(x)$ 
22 T_4=(8*t^4-8*t^2+1)/(2^3);%首一
23 x1=((b-a)*t+a+b)/2;
24 f2=collect(subs(f,x1),t);%t\in [-1,1]
25 c=coeffs(f2);
26 P_3=f2-c(end)*T_4;
27 t=(2*x-a-b)/(b-a);
28 P=subs(P_3,t);
29 hold on
30 fplot(P,[a b]);
31 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

```

1 clc
2 clear
3 syms x t;
4 a=0;
5 b=1;

```

```

6 f=2*x^4-3*x^3+2*x-1;
7 x1=(b-a)/2*t+(b+a)/2;
8 f1=subs(f,x1);
9 S=0;
10 %直接定义Legendre多项式
11 L=[1 t (3*t^2-1)/2 (5*t^3-3*t)/2];
12 for n=0:2
13     S=(2*n+1)/2*int(f1*L(n+1),-1,1)*L(n+1)+S;
14 end
15 % 递推函数法求Legendre多项式
16 % for n=0:2
17 %     S=(2*n+1)/2*int(f1*legendre(n),-1,1)*legendre(n)+S;
18 % end
19 t=(2*x-a-b)/(b-a);
20 S=subs(S,t);
21 S=collect(S,x);
22 fplot(S,[a b]);
23 hold on
24 fplot(f,[a b]);
25 legend({'$S_2(x)$','$f(x)$'},'Interpreter','latex');

```

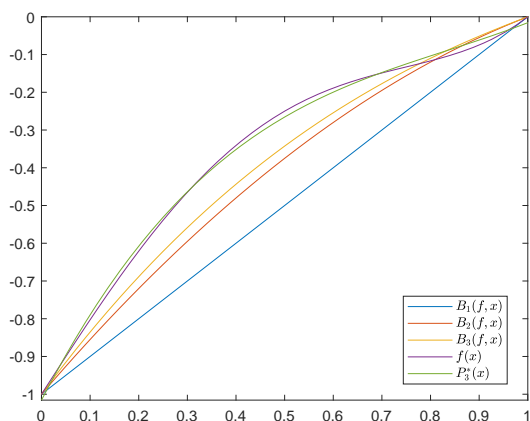
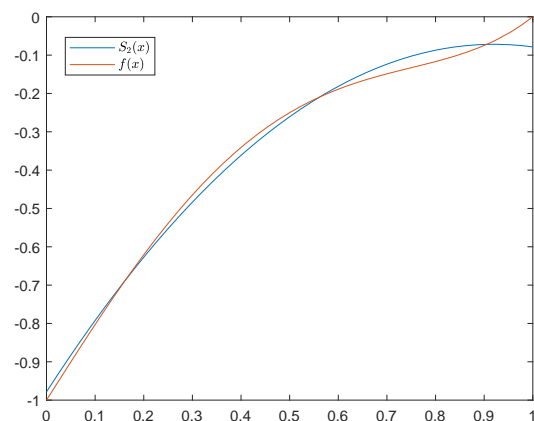
(a) Bernstein 多项式及  $P_3(x)$  图像(b) 二次最佳平方逼近  $S_2(x)$ 

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  $f(x) = \frac{1}{1+25x^2}$ ,  $x \in [-1, 1]$ , 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.

```
1 function y=legendremap(n)
2 %递推公式法
3 syms x;
4 P0=1;
5 P1=x;
6 for i=1:n-1
7     P_idelete1=P0;
8     P_i=P1;
9     P_iadd1=(2*i+1)/(i+1).*x*P_i-i/(i+1)*P_idelete1;
10    P0=P_i;
11    P1=P_iadd1;
12 end
13 if n==0
14     y=1;
15 elseif n==1
16     y=x;
17 else
18     y=collect(P_iadd1);
19 end
20 end
```

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



```

10     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
11     end
12     S(i)=collect(s,x);
13     fplot(S(i),[-1,1]);
14     hold on
15 end
16 hold on
17 fplot(f,[-1 1]);
18 legend({'$S_3^*$','$S_6^*$','$S_{10}^*$','$f(x)$'},'Interpreter','latex');

```

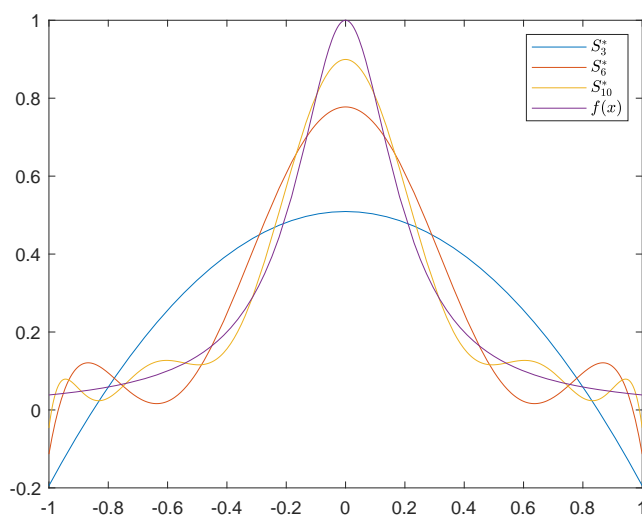


图 8: 最佳平方逼近

## 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.

```

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

```

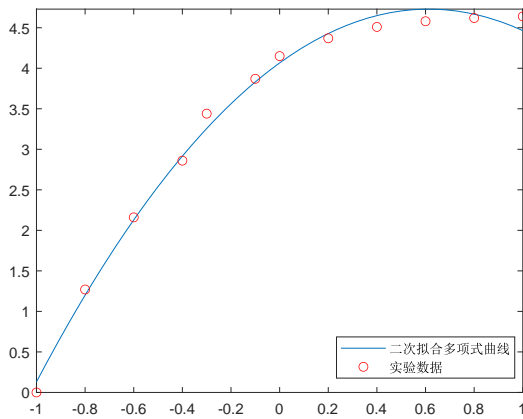
```
4 G=zeros(3,3);
5 syms x
6 B=[1 x x^2];
7 X=[-1 -0.8 -0.6 -0.4 -0.3 -0.1 0 0.2 0.4 0.6 0.8 1];
8 Y=[0 1.27 2.16 2.86 3.44 3.87 4.15 4.37 4.51 4.58 4.62 4.64];
9 %Gram Matrix
10 for i=2:3
11     f1=matlabFunction(B(i));
12     for j=2:3
13         f2=matlabFunction(B(j));
14         G(i,j)=f1(X)*f2(X)';
15     end
16 end
17 for j=2:3
18     f3=matlabFunction(B(j));
19     G(1,j)=sum(f3(X));
20 end
21 G(1,1)=length(X);
22 for j=2:3
23     G(j,1)=G(1,j);
24 end
25 D=zeros(3,1);
26 for i=1:3
27     if i==1
28         D(i,1)=sum(Y);
29     else
30         f4=matlabFunction(B(i));
31         D(i,1)=f4(X)*Y';
32     end
33 end
34 A=G\D;
35 y=0;
36 for i=1:length(A)
37     y=y+A(i)*x^(i-1);
38 end
39 y=collect(y,x);
40 fplot(y,[min(X) max(X)]);
```

```
41 hold on
42 plot(X,Y,'ro');
43 legend({'二次拟合多项式曲线','实验数据'});
44 %% base {1,x,x^2,x^3}
45 clc
46 clear
47 G=zeros(4,4);
48 syms x
49 B=[1 x x^2 x^3];
50 X=[-1 -0.8 -0.6 -0.4 -0.3 -0.1 0 0.2 0.4 0.6 0.8 1];
51 Y=[0 1.27 2.16 2.86 3.44 3.87 4.15 4.37 4.51 4.58 4.62 4.64];
52 %Gram Matrix
53 for i=2:4
54     f1=matlabFunction(B(i));
55     for j=2:4
56         f2=matlabFunction(B(j));
57         G(i,j)=f1(X)*f2(X)';
58     end
59 end
60 for j=2:4
61     f3=matlabFunction(B(j));
62     G(1,j)=sum(f3(X));
63 end
64 G(1,1)=length(X);
65 for j=2:4
66     G(j,1)=G(1,j);
67 end
68 D=zeros(4,1);
69 for i=1:4
70     if i==1
71         D(i,1)=sum(Y);
72     else
73         f4=matlabFunction(B(i));
74         D(i,1)=f4(X)*Y';
75     end
76 end
77 A=G\D;
```

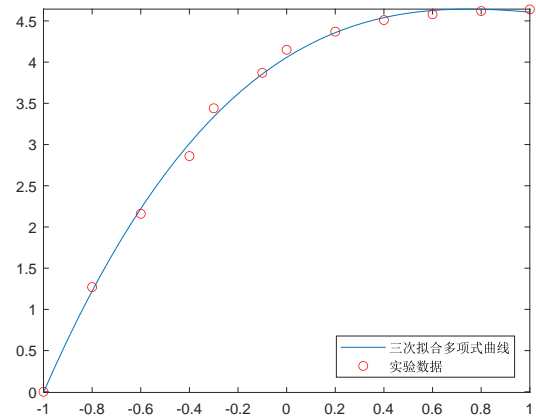
```

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

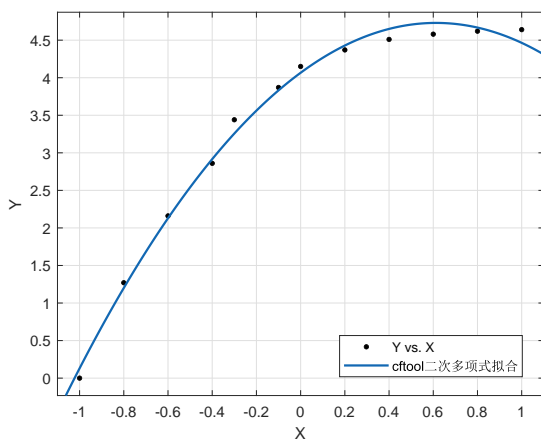
```



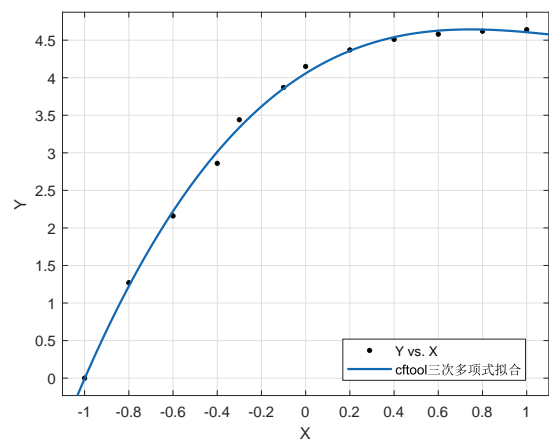
(a) 二次拟合多项式



(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

```
1 %% Lagrange方法
2 clc
3 clear
4 X=[-2 0 1 2];
5 Y=[-7 1 2 9];
6 syms x
7 s1=0;
8 for i=1:length(X)
9     t=1;
10    for j=1:length(Y)
11        if j~=i
12            t=t*(x-X(j))/(X(i)-X(j));
13        end
14    end
15    s1=s1+Y(i)*t;
16 end
17 y=collect(s1,x);%合并x项
18 disp(y);
19 fplot(y,[min(X) max(X)]);
20 hold on
21 plot(X,Y,'ro');
22 legend({'Lagrange插值法','插值节点'});
23 f=matlabFunction(y);
24 Output=f(1.3);
25 disp(Output);
```

### 9.2 最小二乘法

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

```
5 syms x
6 B=[1 x];
7 X=[10 11 12 13 14 15];
8 Y=[20 23 25 27 26 28];
9 %Gram Matrix
10 for i=2:2
11     f1=matlabFunction(B(i));
12     for j=2:2
13         f2=matlabFunction(B(j));
14         G(i,j)=f1(X)*f2(X)';
15     end
16 end
17 for j=2:2
18     f3=matlabFunction(B(j));
19     G(1,j)=sum(f3(X));
20 end
21 G(1,1)=length(X);
22 for j=2:2
23     G(j,1)=G(1,j);
24 end
25 D=zeros(2,1);
26 for i=1:2
27     if i==1
28         D(i,1)=sum(Y);
29     else
30         f4=matlabFunction(B(i));
31         D(i,1)=f4(X)*Y';
32     end
33 end
34 A=G\D;
35 y=0;
36 for i=1:length(A)
37     y=y+A(i)*x^(i-1);
38 end
39 y=collect(y,x);
40 fplot(y,[min(X) max(X)]);
41 hold on
```

```
42 plot(X,Y,'ro');
43 legend({'一次拟合多项式曲线','实验数据'});
44 %% base {1,x,x^2}
45 clc
46 clear
47 G=zeros(3,3);
48 syms x
49 B=[1 x x^2];
50 X=[10 11 12 13 14 15];
51 Y=[20 23 25 27 26 28];
52 %Gram Matrix
53 for i=2:3
54     f1=matlabFunction(B(i));
55     for j=2:3
56         f2=matlabFunction(B(j));
57         G(i,j)=f1(X)*f2(X)';
58     end
59 end
60 for j=2:3
61     f3=matlabFunction(B(j));
62     G(1,j)=sum(f3(X));
63 end
64 G(1,1)=length(X);
65 for j=2:3
66     G(j,1)=G(1,j);
67 end
68 D=zeros(3,1);
69 for i=1:3
70     if i==1
71         D(i,1)=sum(Y);
72     else
73         f4=matlabFunction(B(i));
74         D(i,1)=f4(X)*Y';
75     end
76 end
77 A=G\D;
78 y=0;
```

```
79 for i=1:length(A)
80     y=y+A(i)*x^(i-1);
81 end
82 y=collect(y,x);
83 fplot(y,[min(X) max(X)]);
84 hold on
85 plot(X,Y,'ro');
86 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 $a \leq x_0 < x_1 < \dots < x_n \leq b$ ,  $L_n(x)$ 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

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

3. 证明切比雪夫多项式 $T_n(x) = \cos(n \arccos x)$ ,  $|x| \leq 1$  ( $n = 0, 1, 2, \dots$ )是关于

权函数 $\rho(x) = \frac{1}{\sqrt{1-x^2}}$ 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  $C_k^{(n)} = \frac{(-1)^{n-k}}{nk!(n-k)!} \int_0^n \prod_{\substack{j=0 \\ j \neq k}}^n (t-j) dt$

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

(4) 利用 *MATLAB* 编写算法实现, 参考函数格式为:

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

```

1 function result=NewtonCotesIntegration(f,a,b,n)
2 syms x t
3 h=(b-a)/n;
4 X=a:h:b;
5 f1=matlabFunction(f);
6 Y=f1(X);
7 S=zeros(1,n+1);
8 for k=0:n
9     prod_c=1;
10    for j=0:n
11        if j~=k
12            %prod_c=prod_c*(t-j)/(k-j);
13            prod_c=prod_c*(t-j);
14        end
15    end
16    int_prod=int(prod_c,0,n);
17    %S(k+1)=h/(b-a)*int_prod;
18    S(k+1)=(-1)^(n-k)/(n*factorial(k)*factorial(n-k))*int_prod;
19 end
20 result=vpa(sum(Y.*S)*(b-a));
21 end

```

## 例子

**Q 10.1.2.** 利用程序计算下列两个积分

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

并比较结果 1. 梯形公式; 2. 辛普森公式; 3. 科特斯公式.

```

1 %% I1=\int_1^9 \sqrt{x} dx
2 clc
3 clear
4 syms x t
5 f=sqrt(x);
6 a=1;b=9;
7 f1=matlabFunction(f);
8 %梯形公式
9 I1=(f1(a)+f1(b))*(b-a)/2;
10 I11=NewtonCotesIntegration(f,a,b,1);
11 %Simpson公式
12 I2=(f1(a)+4*f1((a+b)/2)+f1(b))*(b-a)/6;
13 I21=NewtonCotesIntegration(f,a,b,2);
14 %柯特斯公式
15 I31=NewtonCotesIntegration(f,a,b,4);
16 %% I2=\int_0^2 \sqrt{4-x^2} dx
17 clc
18 clear
19 syms x t
20 f=sqrt(4-x^2);
21 a=0;b=2;
22 f1=matlabFunction(f);
23 %梯形公式
24 I1=(f1(a)+f1(b))*(b-a)/2;
25 I11=NewtonCotesIntegration(f,a,b,1);
26 %Simpson公式
27 I2=(f1(a)+4*f1((a+b)/2)+f1(b))*(b-a)/6;
28 I21=NewtonCotesIntegration(f,a,b,2);
29 %柯特斯公式
30 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

$$a = (2R + H + h)/2, c = (H - h)/2,$$

$$R = 6371, \quad H = 2384, \quad h = 439$$

```

In[1]:= R = 6371; H = 2384; h = 439;
a = (2 R + H + h) / 2; c = (H - h) / 2;
S = a Integrate[Sqrt[1 - (c / a) ^2 Sin[x] ^2], {x, 0, Pi / 2}]
      [积分]      [平方根]      [正弦]      [圆周率]

Out[3]=  $\frac{15565}{2} \text{EllipticE}\left[\frac{151321}{9690769}\right]$ 

In[4]:= N[S, 30]
      [数值运算]

Out[4]= 12176.8596279750388997885325568

```

图 10: Mathematica 结果

```

1  clc
2  clear
3  syms x t
4  R=6371;H=2384;h=439;
5  a=(2*R+H+h)/2;c=(H-h)/2;
6  f=sqrt(1-(c/a)^2*sin(x)^2);
7  xmin=0;xmax=pi/2;
8  f1=matlabFunction(f);
9  S=12176.8596279750388997885325568;%from Mathematica N[S,30]
10 X=1:7;
11 Y=1:7;
12 for i=1:7
13     Y(i)=S-a*NewtonCotesIntegration(f,xmin,xmax,X(i));
14 end
15 plot(X,Y,'r-o');
16 legend("Mathematica计算结果减NC公式计算结果");
17 xlabel("牛顿科特斯公式划分数目");
18 %% 结果

```

```
19 %由上则n=2时(Simpson)计算结果与Mathematica结果更接近
20 I=a*NewtonCotesIntegration(f,xmin,xmax,2);
21 disp(I);
```

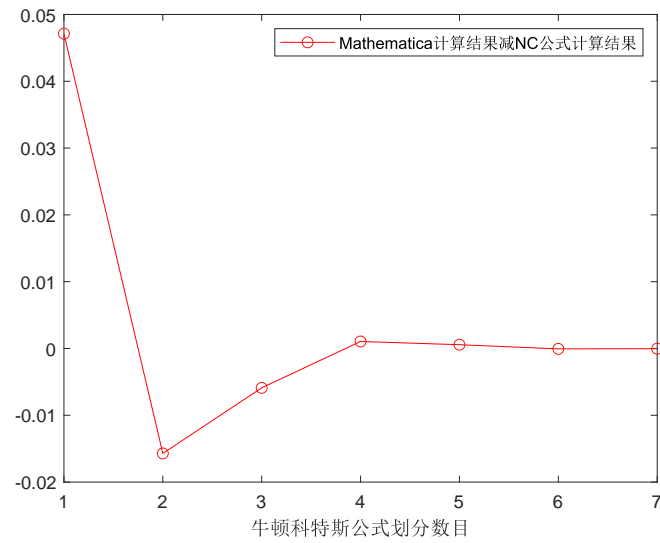


图 11: 结果比较

## 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  $-e^{-5}(3660 \cos 5 + 487.5 \sin 5) - 15 \approx -18.8455351153$ .

```
1 %% 复合梯形求积公式
2 clc
3 clear
4 syms x
5 y=x^5*exp(-x)*sin(x);
6 f=matlabFunction(y);
7 N=[4 8 16 32 64];
8 I=zeros(1,length(N));%积分结果
9 a=0;b=5;
10 for n=1:length(N)
11     h=(b-a)/N(n);
12     X=a:h:b;
13     Y=f(X);
14     I(n)=(2*sum(Y)-Y(1)-Y(length(X)))*h/2;
15 end
16 %% 复合Simpson公式
17 clc
18 clear
19 syms x
20 y=x^5*exp(-x)*sin(x);
21 f=matlabFunction(y);
22 N=[4 8 16 32 64];
23 I=zeros(1,length(N));%积分结果
24 a=0;b=5;
25 for n=1:length(N)
26     h=(b-a)/N(n);
27     X=a:h:b;
28     Y=f(X);
```

```

29     Z=zeros(1,length(X)-1);
30     for i=1:length(Z)
31         Z(i)=X(i)+h/2;
32     end
33     D=f(Z);
34     I(n)=(4*sum(D)+2*sum(Y)-Y(1)-Y(length(X)))*h/6;
35 end

```

## 11.2 高斯—勒让德求积公式

**Q 11.2.1.** 试用 5 个节点的高斯—勒让德求积公式计算上例.

```

1 %% 高斯-勒让德求积公式
2 clc
3 clear
4 syms x t
5 y=x^5*exp(-x)*sin(x);
6 a=0;b=5;
7 x=(b-a)*t/2+(a+b)/2;
8 y1=subs(y,x);%区间变换
9 f=matlabFunction(y1);
10 %求解t_k
11 P_6=(231*t^6-315*t^4+105*t^2-5)/16;
12 T=vpa(solve(P_6==0));
13 Y=f(T);
14 A=[0.4679139 0.3607616 0.1713245 0.4679139 0.3607616 0.1713245];
15 I=A*Y*(b-a)/2;

```

## 11.3 复合型的高斯—勒让德求积公式

**Q 11.3.1.** 试用复合型的高斯—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.

```

1 %% 复合高斯-勒让德求积公式
2 clc
3 clear
4 syms x t

```

```
5 y=x^5*exp(-x)*sin(x);
6 a=0;b=5;
7 XK=[0.7745967 -0.7745967 0];
8 A=[0.5555556 0.5555556 0.8888889];
9 N=[5 10];
10 Result=zeros(1,length(N));
11 for n=1:length(N)
12     h=(b-a)/N(n);
13     X=a:h:b;
14     I=zeros(1,length(X)-1);%小区间上的积分值
15     for i=1:length(I)%小区间G-L
16         x=h*t/2+(X(i+1)+X(i))/2;
17         y1=subs(y,x);
18         f=matlabFunction(y1);
19         I(i)=A*f(XK) '*h/2;
20     end
21     Result(n)=sum(I);
22 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.

```

1 function [L,U]=lu_decomposition(A)
2 % n=length(A);
3 % L=eye(n)+zeros(n,n);
4 % U=[A(1,:);zeros(n-1,n)];
5 % %L(1,:)
6 % for i=1:n
7 %     L(i,1)=A(i,1)/A(1,1);
8 % end
9 % %U(:,2)
10 % for j=2:n
11 %     U(2,j)=A(2,j)-L(2,1)*A(1,j);
12 % end
13 % %L(2,:)
14 % for i=1:n
15 %     L(i,2)=
16 % end
17 [m,n]=size(A);
18 L=eye(m);
19 L(:,1)=A(:,1)/A(1,1);%L第一列赋值
20 U=zeros(m,n);
21 U(1,:)=A(1,:);%U第一行赋值
22 for i=2:m
23     for j=2:n
24         if i<=j
25             U(i,j)=A(i,j)-sum(L(i,1:i-1).*U(1:i-1,j)');
26         else
27             if U(j,j)==0
28                 L(i,j)=0;
29             else
30                 L(i,j)=(A(i,j)-sum(L(i,1:j-1).*U(1:j-1,j)'))/U(j,j);
31             end

```

```

32     end
33 end
34 end
35 end

```

```

1  %page153ex5
2  clc
3  clear
4  A=[1 2 3;2 5 2;3 1 5];
5  b=[14;18;20];
6  %使用内置函数lu
7  [L,U]=lu(A);
8  y=L\b;
9  x=U\y;
10 %使用lu_decomposition
11 [L1,U1]=lu_decomposition(A);
12 y1=L1\b;
13 x1=U1\y1;
14 %直接计算
15 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:

$L = \text{cholesky\_Factorization}(A)$  并计算 P177 第 9 题. (MATLAB 自带程序为  $\text{chol}(A)$ ).

```

1  function L=cholesky_Factorization(A)
2  N=size(A);%记录矩阵的大小
3  m=N(1);%矩阵的行数
4  n=N(2);%矩阵的列数
5  L=zeros(m,n);%初始化L矩阵
6  for i=1:n
7      L(i:n,i)=A(i:n,i);%将A的下半矩阵赋值给L
8  end
9  L(1,1)=sqrt(A(1,1));%步骤一: 先算矩阵的第一个元素
10 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
11 for k=2:n

```

```
12     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
13     for j=k+1:n
14         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
15     end
16 end
17 end
```

```
1 %page177ex9
2 clc
3 clear
4 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];
5 b=[1;0;0;0;0];
6 %使用内置函数chol
7 L=chol(A);%L'L=A
8 y=L'\b;
9 x=L\y;
10 %使用cholesky_Factorization
11 L1=cholesky_Factorization(A);%L1L1'=A
12 y1=L1\b;
13 x1=L1'\y1;
14 %直接计算
15 x2=A\b;
```