

数值分析作业

CY.SP

2023 年 10 月 17 日

目录

1 第一周数值分析实验	2	3.2 page32ex4 实例	8
1.1 第一节: 级数求和与二元函数绘图	2	3.2.1 结果	9
1.2 第二节: 积分递推公式	2		
2 第二周数值分析实验	5	4 第四周数值分析实验	10
2.1 第一节: 插值 function	5	4.1 三点三次 Hermite 插值	10
2.1.1 直接法	5	4.1.1 三点三次估计	10
2.1.2 Lagrange 法	5	4.2 两点三次 Hermite 插值	11
2.2 第二节: 计算实例	6	4.2.1 两点三次估计	11
2.2.1 直接法计算	6		
2.2.2 Lagrange 法计算	6	5 第五周数值分析实验	13
2.2.3 结果	7	5.1 Bernstein 多项式	13
3 第三周数值分析实验	8	6 第六周数值分析实验	15
3.1 Newton 法均差表 function	8	6.1 Gram-Schmidt 正交化	15
		6.2 直接法-Legendre 多项式	15
		6.3 递推法-Legendre 多项式	16

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);%阶乘函数
6 end
7 result=sum(s)
```

Q 1.1.2. 绘制二元函数图像.

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

```
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 %% Equation 第二种递推方式
15 clc
16 clear
17 N=20;
18 EB=zeros(1,N);%误差估计
19 IB=zeros(1,N);%积分值
20 IB(20)=0.0684;%by Mathematica I_{20}=0.0455449...
21 EB(20)=5E-2;
22 for n=N:-1:2
23     IB(n-1)=1/n*(1-IB(n));% k from 1 to 20
24     EB(n-1)=1/n*EB(n);
25 end

```

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 IO=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 %% 第二种递推方式
15 clc
16 clear
17 N=20;
18 EB=zeros(1,N);%误差值
19 IB=zeros(1,N);%积分值
20 IB(20)=0.00799;%by Mathematica I_{20}=0.00799752...
21 EB(20)=5E-5;
22 for n=N:-1:2
23     IB(n-1)=1/(5*n)-1/5*IB(n);% k from 1 to 20
24     EB(n-1)=1/5*EB(n);
25 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

```
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)

```

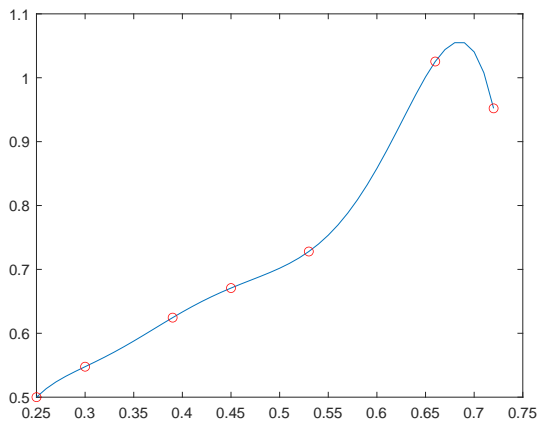
2.2.2 Lagrange 法计算

```

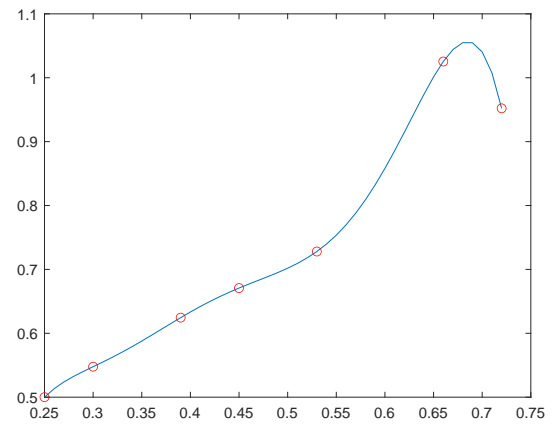
1 clc
2 clear
3 a=[0.25, 0.3, 0.39, 0.45, 0.53, 0.66, 0.72];
4 B=[0.5, 0.5477, 0.6245, 0.6708, 0.7280, 1.0254, 0.9521];
5 y=lagrangeslove(a,B)

```

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

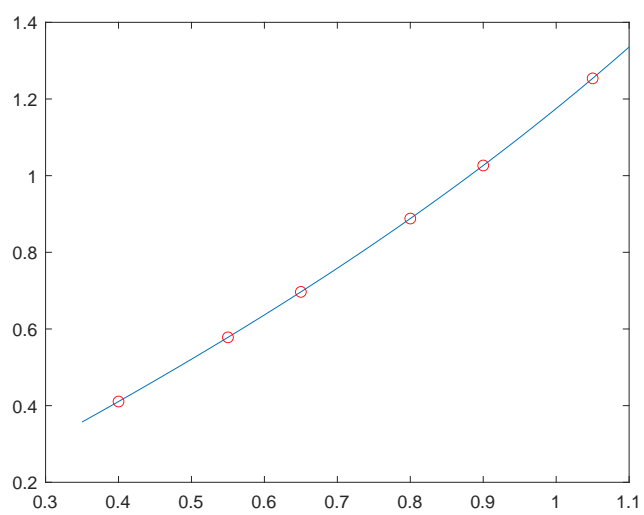


图 2: Newton 法结果

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  X2=[2 2.2 2.5];%估计点值
10 y2=vpa(subs(y,x,X2),5)
11 % plot(X,Y,'ro');
12 % hold on
13 % t=min(X):0.01:max(X);
14 % f=matlabFunction(y);
15 % plot(t,f(t));
```

5 第五周数值分析实验

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 X=0:0.01:1;
5 f1=matlabFunction(f);
6 plot(X,f1(X));
7 hold on
8 for n=1:5
9     y=bernstein_n(f,N(n));
10    disp(y);
11    fy=matlabFunction(y);

```

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

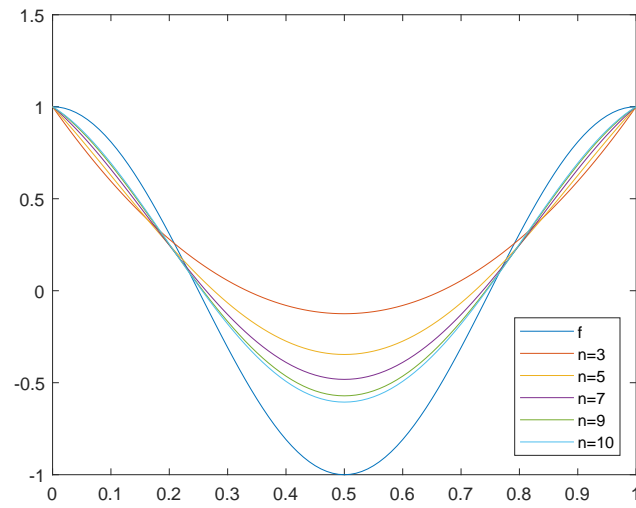


图 3: Bernstein 多项式图像

6 第六周数值分析实验

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:

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

其中

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

6.2 直接法-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, -3/5 x + x^3, -1/5 + x^4 - 6/7 (-1/3 + x^2), -3/7 x + x^5 - 10/9 (-3/5 x + 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 (-3/5 x + x^3) - 21/13 (-3/7 x + x^5 - 10/9 (-3/5 x + 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: 直接法结果

6.3 递推法-Legendre 多项式

```

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_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
21 end

```

```
1 %% P71ex7
2 syms x;
3 f=exp(x);
4 S=0;
5 for n=0:3
6     S=(2*n+1)/2*int(f*legendre(n),-1,1)*legendre(n)+S;
7 end
8 collect(S,x)
9 fplot(S);
10 hold on
11 fplot(f);
```