

学号：20215034001

信阳师范学院

《数值分析》课程设计

学 院：数学与统计学院

专 业：信息与计算科学

年 级：2021 级

姓 名：李浩斌

论文题目：《数值分析》课程设计

指导教师：郑重 职称：副教授

2023 年 12 月 21 日

目录

1 第一周数值分析实验	3	7.2 最佳一致逼近与最佳平方逼近	24
1.1 第一节: 级数求和与二元函数绘图	3	7.2.1 最佳一致逼近	24
1.2 第二节: 积分递推公式 . . .	4	7.2.2 最佳平方逼近	25
2 第二周数值分析实验	7	8 第九周数值分析实验	27
2.1 第一节: 插值 function . . .	7	8.1 正交函数族最佳平方逼近 .	27
2.1.1 直接法	7	8.2 最小二乘拟合	29
2.1.2 Lagrange 法	7	9 第十周数值分析中期练习	33
2.2 第二节: 计算实例	8	9.1 三次插值	33
2.2.1 直接法计算	8	9.2 最小二乘法	33
2.2.2 Lagrange 法计算 . .	8	10 第十一周数值分析实验	37
3 第三周数值分析实验	10	10.1 牛顿-科特斯积分	37
3.1 Newton 法均差表 function .	10	11 第十二周数值分析实验	41
3.2 page32ex4 实例	10	11.1 复合梯形公式与复合 Simpson 公式	41
4 第四周数值分析实验	12	11.2 高斯-勒让德求积公式 . . .	42
4.1 三点三次 Hermite 插值 . . .	12	11.3 复合型的高斯-勒让德求积公式	42
4.2 两点三次 Hermite 插值 . . .	13	12 第十三周数值分析实验	44
5 第六周数值分析实验	15	12.1 矩阵的 LU 分解	44
5.1 Bernstein 多项式	15	12.2 矩阵的 Cholesky 分解 . . .	45
6 第七周数值分析实验	18	13 第十五周数值分析实验	47
6.1 Gram-Schmidt 正交化 . . .	18	13.1 迭代法	47
6.2 Schmidt 正交化-Legendre 多项式	19	13.2 高斯塞德尔迭代法	48
6.3 Schmidt 正交化-Chebyshev 多项式	19	14 第十六周数值分析实验	49
6.4 递推法-Legendre 多项式 . .	20	14.1 Jacobi 和 G-S 迭代法 . . .	49
7 第八周数值分析实验	22	14.2 SOR 迭代法	50
7.1 Chebyshev 多项式零点插值	22	14.3 选做题	51

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

题 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);
```

结果: 522956313

题 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);
```

结果:

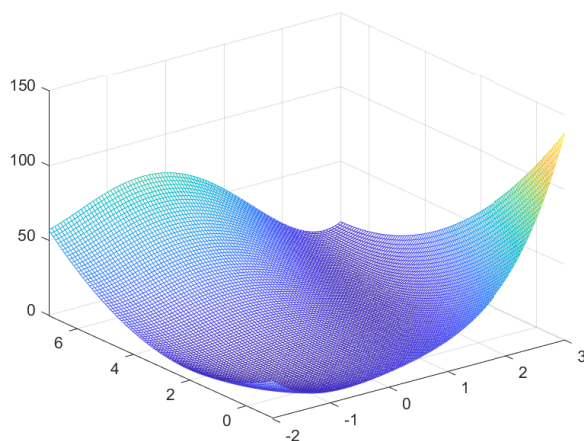


图 1: 运行结果

1.2 第二节: 积分递推公式

题 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 I0=0.6321;%by Mathematica I_0=0.632121...
8 EA(1)=5E-5;
9 IA(1)=1-I0;
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);

```

结果:

表 1: 运行结果

n	1	2	3	4	5	6	7	8	9	10
IA	0.3679	0.2642	0.2074	0.1704	0.148	0.112	0.216	-0.728	7.552	-74.52
EA	5.00E-05	5.00E-05	0.0001	0.0003	0.0012	0.006	0.036	0.252	2.016	18.144
n	11	12	13	14	15	16	17	18	19	20
IA	820.72	-9847.64	128020.3	-1792283	26884253	-4.3E+08	7.31E+09	-1.3E+11	2.5E+12	-5E+13
EA	181.44	1995.84	23950.08	311351	4358915	65383718	1.05E+09	1.78E+10	3.2E+11	6.08E+12
n	1	2	3	4	5	6	7	8	9	10
IB	0.367879	0.264241	0.207277	0.170893	0.145533	0.126802	0.112384	0.100932	0.091612	0.083877
EB	2.06E-20	4.11E-20	1.23E-19	4.93E-19	2.47E-18	1.48E-17	1.04E-16	8.29E-16	7.46E-15	7.46E-14
n	11	12	13	14	15	16	17	18	19	20
IB	0.077352	0.071773	0.066948	0.062732	0.059018	0.05572	0.052768	0.05018	0.04658	0.0684
EB	8.20E-13	9.84E-12	1.28E-10	1.79E-09	2.69E-08	4.30E-07	7.31E-06	0.000132	0.0025	0.05

题 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);

```

结果:

表 2: 运行结果

n	1	2	3	4	5	6	7	8	9	10
IA	0.0885	0.0575	0.045833	0.020833	0.095833	-0.3125	1.705357	-8.40179	42.12004	-210.5
EA	5.00E-06	2.50E-05	0.000125	0.000625	0.003125	0.015625	0.078125	0.390625	1.953125	9.765625
n	11	12	13	14	15	16	17	18	19	20
IA	1052.592	-5262.88	26314.46	-131572	657861.2	-3289306	16446529	-8.2E+07	4.11E+08	-2.1E+09
EA	48.82813	244.1406	1220.703	6103.516	30517.58	152587.9	762939.5	3814697	19073486	95367432
n	1	2	3	4	5	6	7	8	9	10
IB	0.088392	0.058039	0.043139	0.034306	0.028468	0.024325	0.021233	0.018837	0.016926	0.015368
EB	2.62E-18	1.31E-17	6.55E-17	3.28E-16	1.64E-15	8.19E-15	4.10E-14	2.05E-13	1.02E-12	5.12E-12
n	11	12	13	14	15	16	17	18	19	20
IB	0.014071	0.012977	0.01204	0.011229	0.010521	0.009896	0.009342	0.008846	0.008402	0.00799
EB	2.56E-11	1.28E-10	6.40E-10	3.20E-09	1.60E-08	8.00E-08	4.00E-07	2.00E-06	1.00E-05	5.00E-05

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

题 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);
    
```

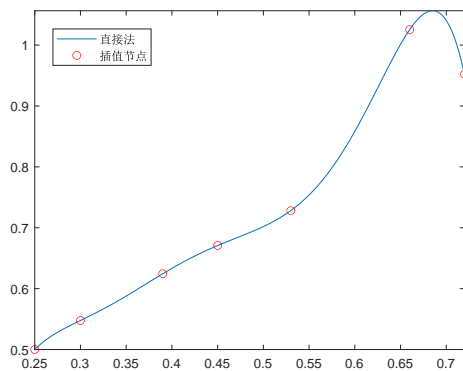
结果:
$$y = -\frac{6036797614447613 x^6}{2199023255552} + \frac{2000819460633247 x^5}{274877906944} - \frac{4307287503018749 x^4}{549755813888} + \frac{4823507311313907 x^3}{1099511627776} - \frac{1483367792093851 x^2}{1099511627776} + \frac{7639215406409353 x}{35184372088832} - \frac{1947716212680825}{140737488355328}.$$

2.2.2 Lagrange 法计算

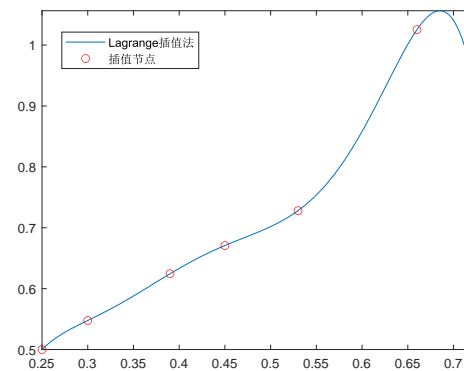
```

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)
    
```


结果:
$$y = -\frac{5060842318750000 x^6}{1843512183513} + \frac{4472937450287500 x^5}{614504061171} - \frac{2063393234673125 x^4}{263358883359} + \frac{10783205041239325 x^3}{2458016244684} - \frac{2210764632642529 x^2}{1638677496456} + \frac{4941513026732767 x}{22759409673000} - \frac{163157510611}{11789386000}.$$



(a) 直接法图像



(b) Lagrange 法图像

图 2: 运行结果

装

订

线

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=l-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 %非调用自定义函数法
8 n=length(X);
9 A=[X',Y',zeros(n,n-1)];%l=n+1,h=n
10 for l=3:n+1 %列
11     for h=l-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 disp(A);%均差表
16 %调用自定义函数法
17 %A=newtonmatrix(X,Y);%均差表
18 %代入求函数表达式
19 for h=2:length(X)
20     y=y+A(h,h+1)*prod(x-X(1:h-1));
```

```

21 end
22 y=collect(y,x);%合并化简
23 disp(y);
24 %绘图
25 fplot(y,[0.35 1.10]);
26 hold on
27 plot(X,Y,'ro');
28 legend({'Newton法图像','插值节点'});
    
```

结果:
$$y = \frac{2702819642032443 x^5}{9223372036854775808} + \frac{2792012692852253541 x^4}{9223372036854775808} + \frac{456060272710172342613 x^3}{3689348814741910323200} + \frac{218532157850334883771 x^2}{7378697629483820646400} + \frac{91322270122367737446143 x}{92233720368547758080000} + \frac{146974433410736655459}{115292150460684697600000}.$$

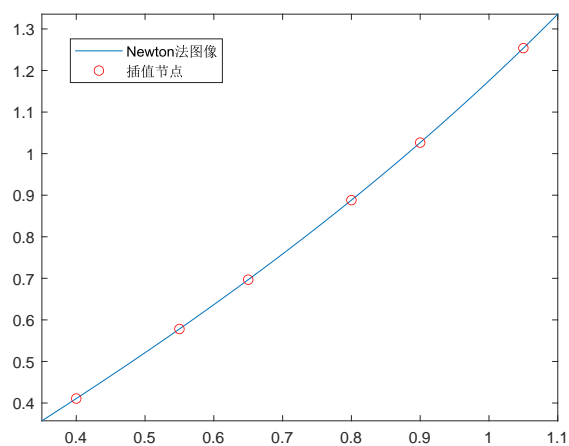


图 3: Newton 法结果

4 第四周数值分析实验

题 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 %% 三点三次Hermite插值方式
2 clc
3 clear
4 syms x a
5 X=[1.9321 2.2201 2.5281];%选择的插值节点
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 n=length(X);
10 y=Y(1);
11 A=[X',Y',zeros(n,n-1)];%均差表(l=n+1,h=n)
12 for l=3:n+1 %列
13     for h=1-1:n %行
14         A(h,l)=(A(h,l-1)-A(h-1,l-1))/(X(h)-X(h-1+2));
15     end
16 end
17 for h=2:length(X)
18     y=y+A(h,h+1)*prod(x-X(1:h-1));%前n-1项
19 end
20 y1=y+a*prod(x-X(1:n));
21 a1=solve(diff(y1,x)==DY_i,a);%符号运算求解
22 a2=subs(a1,x,X(i));%代值已知f'(x_i)求a
23 y2=y+a2*prod(x-X(1:n));
24 y=collect(y2,x);%化简
25 X2=[2 2.2 2.5];%估计点值
26 y3=vpa(subs(y,x,X2),5);
27 disp(y3);%估计点函数值
28 %绘图
29 plot(X,Y,'ro');
30 hold on
    
```

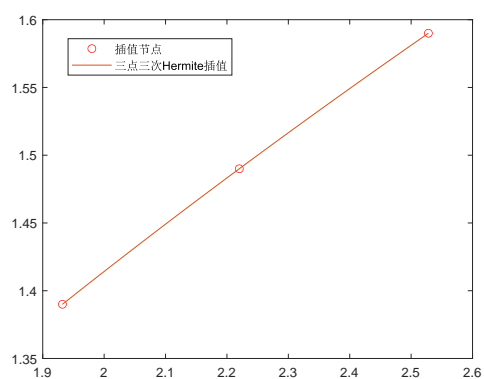
```
31 fplot(y,[min(X) max(X)]);
32 legend({'插值节点','三点三次Hermite插值'});
```

结果: [1.4142, 1.4832, 1.5811]

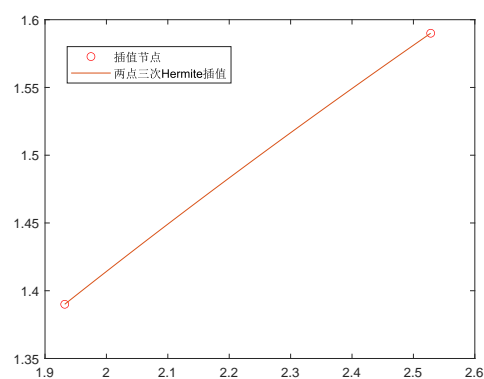
4.2 两点三次 Hermite 插值

```
1 %% 两点三次Hermite插值方式
2 clc
3 clear
4 syms x
5 X=[1.9321 2.5281];%选择的插值节点
6 Y=sqrt(X);%Y=[1.39 1.59]
7 DY_all=subs(diff(sqrt(x),1),x,X);
8 syms x
9 y=(1+2*(x-X(1))/(X(2)-X(1)))*((x-X(2))/(X(1)-X(2)))^2*Y(1)+...
10 (1+2*(x-X(2))/(X(1)-X(2)))*((x-X(1))/(X(2)-X(1)))^2*Y(2)+...
11 (x-X(1))*((x-X(2))/(X(1)-X(2)))^2*DY_all(1)+...
12 (x-X(2))*((x-X(1))/(X(2)-X(1)))^2*DY_all(2);
13 y=collect(y);
14 X2=[2 2.2 2.5];%估计点值
15 y2=vpa(subs(y,x,X2),5);
16 disp(y2);%估计函数值
17 %绘图
18 plot(X,Y,'ro');
19 hold on
20 fplot(y,[min(X) max(X)]);
21 legend({'插值节点','两点三次Hermite插值'});
```

结果: [1.4142, 1.4833, 1.5811]



(a) 三点三次 Hermite 插值



(b) 两点三次 Hermite 插值

图 4: 运行结果

装
订
线

5 第六周数值分析实验

5.1 Bernstein 多项式

定义 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
    
```

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

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

结果:

$$B_3(f, x) = \frac{9x^2}{2} - \frac{9x}{2} + 1.$$

$$B_5(f, x) = -\frac{17396110258977045x^4}{2251799813685248} + \frac{69584441035908185x^3}{4503599627370496} - \frac{19232666484692435x^2}{4503599627370496} - \frac{3889888508315415x}{1125899906842624} + 1.$$

$$B_7(f, x) = \frac{7x^7}{36028797018963968} + \frac{48511846945173955x^6}{18014398509481984} - \frac{291071081671043793x^5}{36028797018963968} - \frac{39777928968693195x^4}{9007199254740992} + \frac{100417737650160665x^3}{4503599627370496} - \frac{22201645688437845x^2}{2251799813685248} - \frac{11869558316351393x}{4503599627370496} + 1.$$

$$B_9(f, x) = -\frac{63x^9}{2251799813685248} - \frac{3651501963845295x^8}{9007199254740992} + \frac{912875490961461x^7}{562949953421312} + \frac{4844390630064489x^6}{1125899906842624} - \frac{41846600653847829x^5}{2251799813685248} + \frac{21573830928373101x^4}{4503599627370496} + \frac{26215294559596821x^3}{1125899906842624} - \frac{29056921947987975x^2}{2251799813685248} - \frac{18965558858231295x}{9007199254740992} + 1.$$

$$B_{10}(f, x) = \frac{36617051598889x^{10}}{4503599627370496} - \frac{183085257994425x^9}{4503599627370496} - \frac{1745013596718105x^8}{2251799813685248} + \frac{3764655080427825x^7}{1125899906842624} + \frac{16286793569370555x^6}{4503599627370496} - \frac{51167254958838837x^5}{2251799813685248} + \frac{42639379132365645x^4}{4503599627370496} + \frac{25803329789012535x^3}{1125899906842624} - \frac{15656499233079345x^2}{1125899906842624} - \frac{1075138741208855x}{562949953421312} + 1.$$

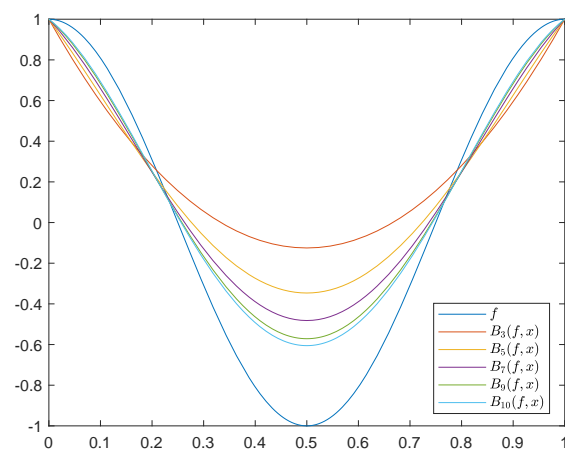


图 5: Bernstein 多项式图像

装
订
线

6 第七周数值分析实验

6.1 Gram-Schmidt 正交化

题 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.

$$\varphi_0(x) = 1,$$

$$\varphi_1(x) = x,$$

$$\varphi_2(x) = x^2 - \frac{(x^2, \varphi_0)}{(\varphi_0, \varphi_0)} \varphi_0(x) - \frac{(x^2, \varphi_1)}{(\varphi_1, \varphi_1)} \varphi_1(x),$$

$$\varphi_n(x) = x^n - \sum_{j=0}^{n-1} \frac{(x^n, \varphi_j)}{(\varphi_j, \varphi_j)} \varphi_j(x), \quad n = 3, 4, \dots$$

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

$$\varphi_0(x) = 1, \varphi_{-1}(x) = 0,$$

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

$$\text{这里 } (x\varphi_n(x), \varphi_n(x)) = \int_a^b x\varphi_n^2(x)\rho(x)dx.$$

```

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
10     [rho P[[j]]^2,{x,a,b}])P[[j]],{j,1,i-1}]
11 ];
P

```

结果: (Mathematica)

```

In[1]:= 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

```

$$\text{Out[4]} = \left\{ 1, x, -\frac{1}{3} + x^2, -\frac{3x}{5} + x^3, -\frac{1}{5} + x^4 - \frac{6}{7} \left(-\frac{1}{3} + x^2 \right), -\frac{3x}{7} + x^5 - \frac{10}{9} \left(-\frac{3x}{5} + x^3 \right), \right.$$

$$\left. -\frac{1}{7} + x^6 - \frac{5}{7} \left(-\frac{1}{3} + x^2 \right) - \frac{15}{11} \left(-\frac{1}{5} + x^4 - \frac{6}{7} \left(-\frac{1}{3} + x^2 \right) \right), -\frac{x}{3} + x^7 - \frac{35}{33} \left(-\frac{3x}{5} + x^3 \right) - \frac{21}{13} \left(-\frac{3x}{7} + x^5 - \frac{10}{9} \left(-\frac{3x}{5} + x^3 \right) \right) \right\}$$

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 多项式的最大公约数 更多...

图 6: 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
10      [rho P[[j]]^2,{x,a,b}])P[[j]],{j,1,i-1}]
11  P
    
```

结果: (Mathematica)

```

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 7: 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)
```

装
订
线

7 第八周数值分析实验

7.1 Chebyshev 多项式零点插值

题 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(l)*t;
31     end
32     Y(i)=collect(s1,x);

```

```

33 latex(vpa(Y(i),5))
34 % 绘图
35 fplot(Y(i));%Lagrange插值多项式图像
36 % fplot(f-Y(i));%误差曲线
37 hold on
38 end
39 legend({'$L_{10}(x)$','$L_{14}(x)$','$L_{20}(x)$'},'Interpreter','latex'
40 );
% legend({'$f(x)-L_{10}(x)$','$f(x)-L_{14}(x)$','$f(x)-L_{20}(x)$'},'
Interpreter','latex');

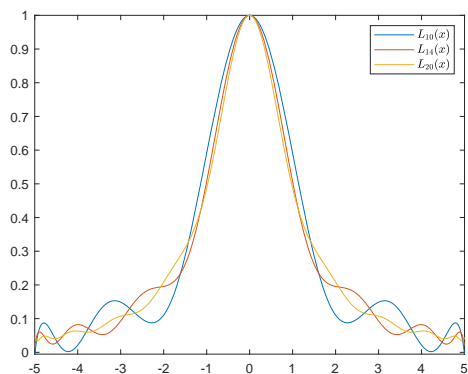
```

结果:

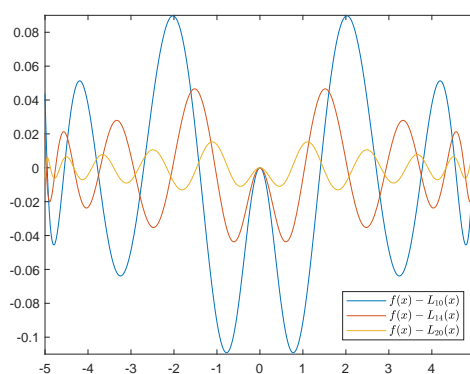
$$\begin{aligned}
 L_{10}(x) \approx & -4.7752e - 6x^{10} + 1.0795e - 20x^9 + 0.00033307x^8 - 6.2457e - 19x^7 \\
 & - 0.0085405x^6 + 1.1751e - 17x^5 + 0.098309x^4 - 7.3764e - 17x^3 - 0.49906x^2 \\
 & + 4.8306e - 17x + 1.0.
 \end{aligned}$$

$$\begin{aligned}
 L_{14}(x) \approx & -5.466e - 8x^{14} + 2.5422e - 23x^{13} + 5.179e - 6x^{12} - 0.00019734x^{10} \\
 & + 2.2204e - 16x^9 + 0.0038672x^8 - 0.041399x^6 + 0.23844x^4 + 5.6843e - 14x^3 \\
 & - 0.69456x^2 - 1.1369e - 13x + 1.0.
 \end{aligned}$$

$$\begin{aligned}
 L_{20}(x) \approx & 6.7807e - 11x^{20} + 4.8908e - 26x^{19} - 8.9674e - 9x^{18} - 1.3553e - 20x^{17} \\
 & + 5.0957e - 7x^{16} - 1.301e - 18x^{15} - 0.000016269x^{14} + 2.7756e - 17x^{13} \\
 & + 0.00032046x^{12} + 1.1102e - 15x^{11} - 0.0040277x^{10} + 0.032347x^8 \\
 & - 1.4211e - 14x^7 - 0.16239x^6 + 1.5632e - 13x^5 + 0.49061x^4 - 0.87049x^2 + 1.0.
 \end{aligned}$$



(a) Lagrange 插值多项式图像



(b) 误差曲线 $f(x) - L_n(x)$

图 8: 结果图示

7.2 最佳一致逼近与最佳平方逼近

题 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     latex(Y(i))
18     fplot(Y,[a b]);
19 end
20 hold on
21 fplot(f,[a b]);
22 %求解三次最佳一致逼近多项式P_3(x)=f(x)-a_n*T'_4(x)
23 T_4=(8*t^4-8*t^2+1)/(2^3);%首一
24 x1=((b-a)*t+a+b)/2;
25 f2=collect(subs(f,x1),t);%t\in [-1,1]
26 c=coeffs(f2);
27 P_3=f2-c(end)*T_4;
28 t=(2*x-a-b)/(b-a);
29 P=subs(P_3,t);
30 latex(P)
31 hold on
32 fplot(P,[a b]);
33 legend({'$B_1(f,x)$','$B_2(f,x)$','$B_3(f,x)$','$f(x)$','$P_3^*(x)$'},'
```



```
Interpreter','latex');
```

结果:

$$B_1(f, x) = x - 1.$$

$$B_2(f, x) = -\frac{x^2}{2} + \frac{3x}{2} - 1.$$

$$B_3(f, x) = \frac{2x^3}{9} - \frac{26x^2}{27} + \frac{47x}{27} - 1.$$

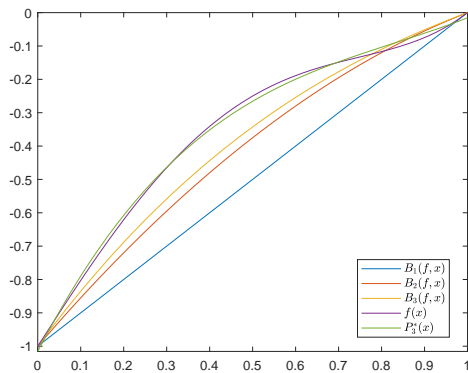
$$P_3(x) = \frac{3x}{4} - \frac{(2x-1)^2}{4} + \frac{(2x-1)^3}{8} - \frac{41}{64}.$$

7.2.2 最佳平方逼近

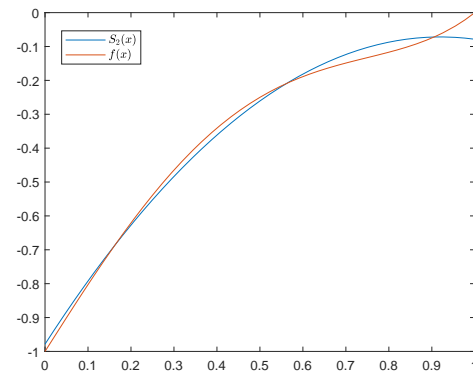
```
1 clc
2 clear
3 syms x t;
4 a=0;
5 b=1;
6 f=2*x^4-3*x^3+2*x-1;
7 x1=(b-a)/2*t+(b+a)/2;
8 f1=subs(f,x1);
9 S=0;
10 %直接定义Legendre多项式
11 L=[1 t (3*t^2-1)/2 (5*t^3-3*t)/2];
12 for n=0:2
13     S=(2*n+1)/2*int(f1*L(n+1),-1,1)*L(n+1)+S;
14 end
15 % 递推函数法求Legendre多项式
16 % for n=0:2
17 %     S=(2*n+1)/2*int(f1*legendremap(n),-1,1)*legendremap(n)+S;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

结果:

$$S_2(x) = -\frac{15x^2}{14} + \frac{69x}{35} - \frac{137}{140}.$$



(a) Berstein 多项式及 $P_3(x)$ 图像



(b) 二次最佳平方逼近 $S_2(x)$

图 9: 结果图示

装
订
线

8 第九周数值分析实验

8.1 正交函数族最佳平方逼近

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

```

结果:

$$S_3(x) = \left(\frac{9}{20} - \frac{21 \operatorname{atan}(5)}{25} \right) x^2 + \frac{12 \operatorname{atan}(5)}{25} - \frac{3}{20}.$$

$$S_6(x) = \left(\frac{6579573}{250000} - \frac{28501473 \operatorname{atan}(5)}{1250000} \right) x^6 + \left(\frac{1102563 \operatorname{atan}(5)}{31250} - \frac{1988679}{50000} \right) x^4 \\ + \left(\frac{787143}{50000} - \frac{3698793 \operatorname{atan}(5)}{250000} \right) x^2 + \frac{166579 \operatorname{atan}(5)}{125000} - \frac{52633}{50000}.$$

$$S_{10}(x) = \left(\frac{15053862021341}{150000000000} - \frac{37844796458469 \operatorname{atan}(5)}{500000000000} \right) x^{10} + \left(\frac{4806364476771 \operatorname{atan}(5)}{25000000000} - \frac{4448386331961}{17500000000} \right) x^8 \\ + \left(\frac{1145902042857}{5000000000} - \frac{8707875715539 \operatorname{atan}(5)}{50000000000} \right) x^6 \\ + \left(\frac{1668177980469 \operatorname{atan}(5)}{25000000000} - \frac{217491363947}{2500000000} \right) x^4 + \left(\frac{123339641139}{10000000000} - \frac{970103687553 \operatorname{atan}(5)}{100000000000} \right) x^2 \\ + \frac{19197012681 \operatorname{atan}(5)}{62500000000} - \frac{1037189439}{312500000}.$$

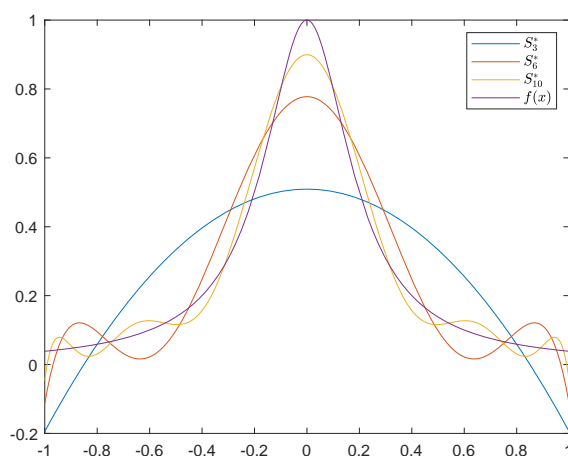


图 10: 最佳平方逼近

8.2 最小二乘拟合

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

```

```

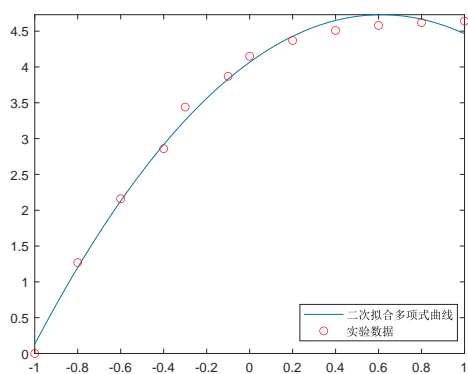
73     else
74         f4=matlabFunction(B(i));
75         D(i,1)=f4(X)*Y';
76     end
77 end
78 A=G\D;
79 y=0;
80 for i=1:length(A)
81     y=y+A(i)*x^(i-1);
82 end
83 y=collect(y,x);
84 latex(y)
85 fplot(y,[min(X) max(X)]);
86 hold on
87 plot(X,Y,'ro');
88 legend({'三次拟合多项式曲线','实验数据'});

```

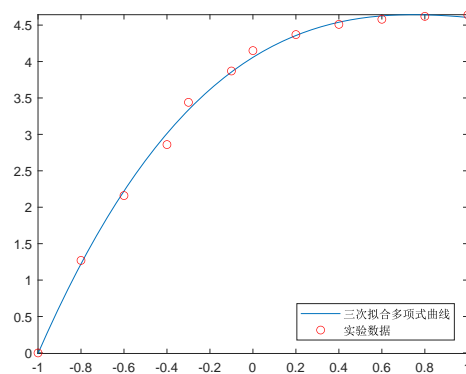
结果:

$$y = -\frac{3982742614460357 x^2}{2251799813685248} + \frac{2439825684435733 x}{1125899906842624} + \frac{2288943766543947}{562949953421312}.$$

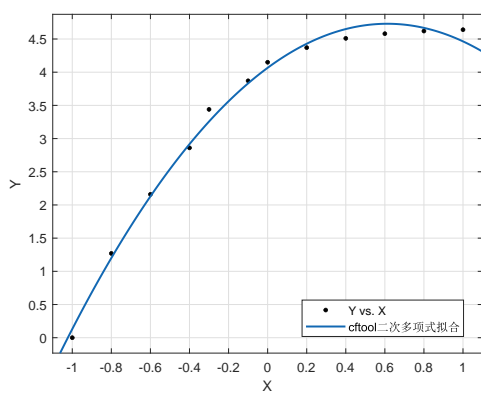
$$y = \frac{8521429937993895 x^3}{18014398509481984} - \frac{7918081977677879 x^2}{4503599627370496} + \frac{4129100537421519 x}{2251799813685248} + \frac{4568116039683039}{1125899906842624}.$$



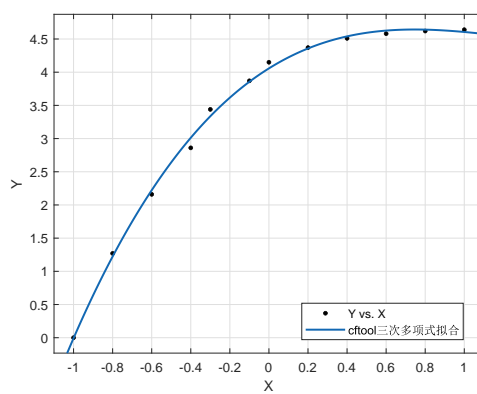
(a) 二次拟合多项式



(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 11: 最小二乘拟合

装
订
线

9 第十周数值分析中期练习

9.1 三次插值

题 9.1.1. 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。

```

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);
    
```

结果: $x^3 + 1$; 3.1970e+000

9.2 最小二乘法

题 9.2.1. 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。

```

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';

```

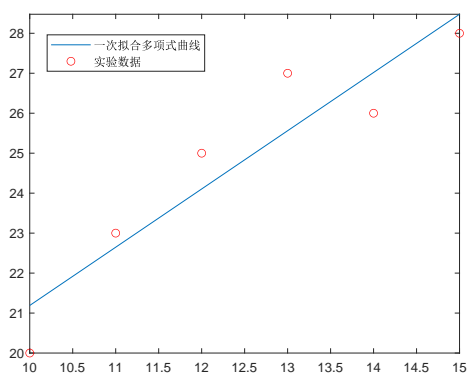
```

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'});
    
```

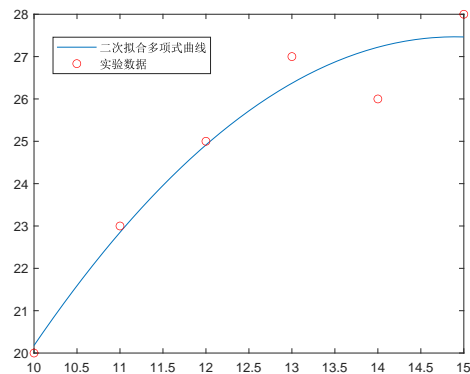
结果:

$$y = \frac{51x}{35} + \frac{1863096274418193}{281474976710656}.$$

$$y = -\frac{683582086296581}{2251799813685248}x^2 + \frac{5092686542910377}{562949953421312}x - \frac{5619446856466419}{140737488355328}.$$



(a) 一次拟合多项式



(b) 二次拟合多项式

图 12: 最小二乘法

10 第十一周数值分析实验

10.1 牛顿-科特斯积分

题 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

题 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);

```

结果:

表 3: 运行结果

	梯形公式	Simpson 公式	柯斯特公式
I_1	16	17.25903	17.32644
I_2	2	2.976068	3.090764

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

In[4]:= N[S, 30]
Out[4]= 12176.8596279750388997885325568
```

图 13: 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);
```

结果: 12176.875336227736163485779741222

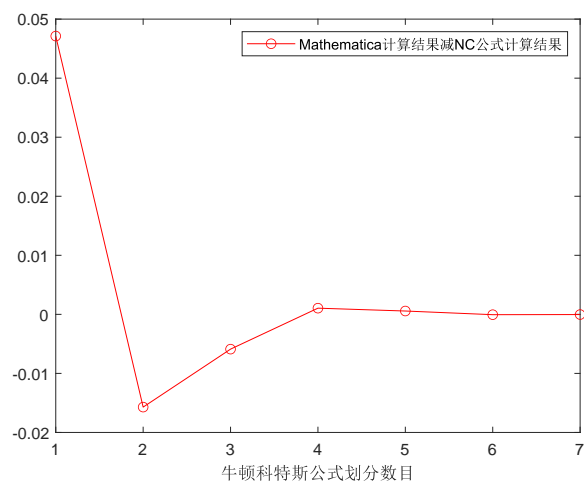


图 14: 结果比较

装
订
线

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

题 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
    
```

结果:

表 4: 运行结果

n	4	8	16	32	64
复合梯形公式	-18.0457	-18.6488	-18.7968	-18.8334	-18.8425
复合 Simpson 公式	-18.8498	-18.8461	-18.8456	-18.8455	-18.8455

11.2 高斯-勒让德求积公式

题 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;
    
```

结果: -18.850292536760721693352062553335

11.3 复合型的高斯-勒让德求积公式

题 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=[0.55555556 0.55555556 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

```

结果: [-18.8454781687866,-18.8455360298194]

12 第十三周数值分析实验

12.1 矩阵的 LU 分解

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

```

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

结果:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 3 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & -24 \end{bmatrix} \Rightarrow \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

12.2 矩阵的 Cholesky 分解

题 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;

```

结果:

$$\begin{bmatrix} 1.41421356237310 & 0 & 0 & 0 & 0 \\ -0.707106781186548 & 1.22474487139159 & 0 & 0 & 0 \\ 0 & -0.816496580927726 & 1.15470053837925 & 0 & 0 \\ 0 & 0 & -0.866025403784439 & 1.11803398874990 & 0 \\ 0 & 0 & 0 & -0.894427190999916 & 1.09544511501033 \end{bmatrix}$$

$x = [0.83333333333333, 0.66666666666667, 0.50000000000000, 0.33333333333333, 0.16666666666667]^T$

装
订
线

13 第十五周数值分析实验

13.1 迭代法

题 13.1.1. 设线性方程组
$$\begin{cases} 8x_1 - 3x_2 + 2x_3 = 20, \\ 4x_1 + 11x_2 - x_3 = 33, \\ 6x_1 + 3x_2 + 12x_3 = 36. \end{cases}$$
 精确解是 $x^* = (3, 2, 1)^T$. 按

如下格式求解

$$\begin{cases} x_1^{(k+1)} = (3x_2^{(k)} - 2x_3^{(k)} + 20) / 8 \\ x_2^{(k+1)} = (-4x_1^{(k)} + x_3^{(k)} + 33) / 11 \\ x_3^{(k+1)} = (-6x_1^{(k)} - 3x_2^{(k)} + 36) / 12 \end{cases}$$

其中 $x^{(0)} = (0, 0, 0)^T$. 试给出第 10, 20, 100, 200, 1000 次的迭代结果及误差向量的无穷范数.

```
1 clc
2 clear
3 B=[0 3/8 -2/8;-4/11 0 1/11;-6/12 -3/12 0];
4 f=[20/8 33/11 36/12]';
5 x=[3,2,1]';%方程组的解
6 N=[10,20,100,200,1000];%迭代次数
7 for i=1:length(N)
8     j=0;
9     x0=[0,0,0]';%迭代初始值
10    while j<N(i)
11        x0=B*x0+f;
12        j=j+1;
13    end
14    disp(x0);
15    disp(max(abs(x0-x)));%误差值
16 end
```

结果:

表 5: 运行结果

	10	20	100	200	1000
x_1	3.0000	3.0000	3	3	3
x_2	1.9999	2.0000	2	2	2
x_3	0.9999	1.0000	1	1	1
误差向量的无穷范数	1.26E-04	3.84E-09	0	0	0

13.2 高斯塞德尔迭代法

题 13.2.1. 用高斯一赛德尔迭代法求解练习 1 , 试给出第 10,20,100,200,1000 次的迭代结果及误差向量的无穷范数.

```

1  clc
2  clear
3  A=[8 -3 2;4 11 -1;6 3 12];
4  D=diag(diag(A));%对角阵
5  L=D-tril(A);%下三角阵
6  U=D-triu(A);%上三角阵
7  B=(D-L)\U;
8  b=[20 33 36]';
9  f=(D-L)\b;
10 x=[3,2,1]';%方程组的解
11 N=[10,20,100,200,1000];%迭代次数
12 for i=1:length(N)
13     j=0;
14     x0=[0,0,0]';%迭代初始值
15     while j<N(i)
16         x0=B*x0+f;
17         j=j+1;
18     end
19     disp(x0);
20     disp(max(abs(x0-x)));%误差值
21 end
    
```

结果:

表 6: 运行结果

	10	20	100	200	1000
x_1	3.0000	3	3	3	3
x_2	2.0000	2	2	2	2
x_3	1.0000	1	1	1	1
误差向量的无穷范数	6.32E-09	0	0	0	0

14 第十六周数值分析实验

14.1 Jacobi 和 G-S 迭代法

题 14.1.1. 设线性方程组

$$\begin{cases} x_1 + 2x_2 - 2x_3 = 1, \\ x_1 + x_2 + x_3 = 1, \\ 2x_1 + 2x_2 + x_3 = 1. \end{cases}$$

试编制程序, 列表给出前 10 个迭代向量, 并考察 Jacobi 和 G-S 迭代法的收敛性.

```

1 %% Jacobi
2 clc;clear;
3 A=[1 2 -2;1 1 1;2 2 1];
4 D=diag(diag(A));%对角阵
5 L=D-tril(A);%下三角阵
6 U=D-triu(A);%上三角阵
7 B=D\(L+U);
8 b=[1 1 1]';
9 f=D\b;
10 N=10;%迭代次数
11 x0=[0,0,0]';%迭代初始值
12 j=0;
13 while j<N
14     x0=B*x0+f;
15     j=j+1;
16     disp(x0);
17 end
18 %% G-S
19 clc;clear;
20 A=[1 2 -2;1 1 1;2 2 1];
21 D=diag(diag(A));%对角阵
22 L=D-tril(A);%下三角阵
23 U=D-triu(A);%上三角阵
24 B=(D-L)\U;
25 b=[1 1 1]';
26 f=(D-L)\b;
27 N=10;%迭代次数
28 x0=[0,0,0]';%迭代初始值
29 j=0;
30 while j<N

```

```
31 x0=B*x0+f;
32 j=j+1;
33 disp(x0);
34 end
```

结果:

表 7: 运行结果

(Jacobi)n	0	1	2	3	4	5	6	7	8	9	10
x_1	0	1	1	-3	-3	-3	-3	-3	-3	-3	-3
x_2	0	1	-1	3	3	3	3	3	3	3	3
x_3	0	1	-3	1	1	1	1	1	1	1	1
(G-S)n	0	1	2	3	4	5	6	7	8	9	10
x_1	0	1	-1	-11	-43	-131	-355	-899	-2179	-5123	-11779
x_2	0	0	3	15	51	147	387	963	2307	5379	12291
x_3	0	-1	-3	-7	-15	-31	-63	-127	-255	-511	-1023

14.2 SOR 迭代法

题 14.2.1. 用 SOR 方法解线性方程组

$$\begin{pmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

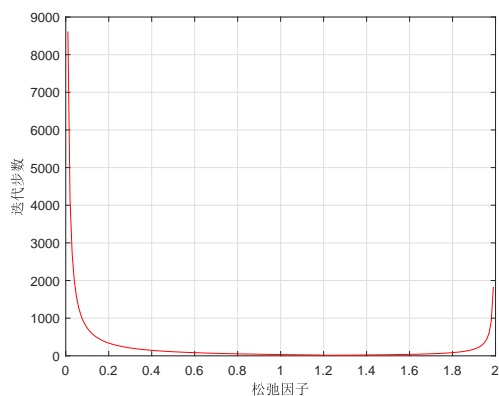
初值取 $x^{(0)} = (0, 0, 0, 0)^T$, 计算到 $\|x^{(k)} - x^{(k-1)}\|_{\infty} < 10^{-8}$ 停止, 并绘制迭代步数 IT 关于松弛因子 ω (步长取 0.01) 的图像.

```
1 clc;clear;
2 A=[4 -1 -1 -1;-1 4 -1 -1;-1 -1 4 -1;-1 -1 -1 4];
3 b=ones(4,1);
4 x0=zeros(4,1);%初值
5 omgea=0.01:0.01:1.99;
6 I=length(omgea);
7 N=zeros(1,length(omgea));%迭代步数
8 for i=1:length(N)
9     N(i)=SOR(A,b,x0,omgea(i));
10 end
11 plot(omgea,N,'r-');
12 xlabel('松弛因子');
```

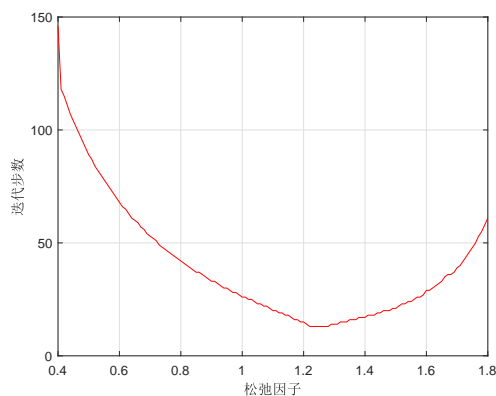
```

13 ylabel('迭代步数');
14 grid on;
15
16 function n=SOR(A,b,x0,omega)
17 D=diag(diag(A));%对角阵
18 L=D-tril(A);%下三角阵
19 U=D-triu(A);%上三角阵
20 B=(D-omega*L)\((1-omega)*D+omega*U);
21 f=omega*(D-omega*L)\b;
22 n=0;
23 epsilon=1;%误差初始值
24 while epsilon>=10^(-8)
25     d=x0;
26     x0=B*x0+f;
27     c=x0;
28     epsilon=max(abs(c-d));
29     n=n+1;
30 end
31 end
    
```

结果: $x = [0.308641974897860, 0.308641979217244, 0.308641971803977, 0.308641977711201]^T$.



(a) $0.01 \leq \omega \leq 1.99$



(b) $0.40 \leq \omega \leq 1.80$

图 15: 迭代步数

14.3 选做题

题 14.3.1. 已知 50×50 带状方程组, 设初始解向量为 $x^{(0)} = 0.01 \text{ ones}(50, 1)$, 精度 $\varepsilon = 10^{-14}$, 试采用 *Jacobi* 迭代法和 *G-S* 迭代法求解带状线性方程组满足精度要求

的近似解.

$$\begin{pmatrix} 12 & -2 & 1 & & & \\ -2 & 12 & -2 & 1 & & \\ 1 & -2 & 12 & -2 & 1 & \\ & 1 & -2 & 12 & -2 & 1 \\ & & \ddots & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 12 & -2 & 1 \\ & & & & 1 & -2 & 12 & -2 \\ & & & & & 1 & -2 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{48} \\ x_{49} \\ x_{50} \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 5 \\ 5 \\ \vdots \\ 5 \\ 5 \\ 5 \end{pmatrix}$$

```

1 %% Jacobi
2 clc;clear;
3 A=zeros(50,50)+diag(12*ones(1,50));
4 for i=1:length(A(:,1))-1
5     A(i,i+1)=-2;
6 end
7 for i=1:length(A(:,1))-2
8     A(i,i+2)=1;
9 end
10 for i=1:length(A(:,1))%行
11     for j=1:i%列
12         A(i,j)=A(j,i);
13     end
14 end
15 D=diag(diag(A));%对角阵
16 L=D-tril(A);%下三角阵
17 U=D-triu(A);%上三角阵
18 B=D\(L+U);
19 b=5*ones(50,1);
20 f=D\b;
21 x=A\b;%精确值
22 x0=0.01*ones(50,1);%迭代初始值
23 epsilon=1;%误差初始值
24 while epsilon>10^(-14)
25     x0=B*x0+f;
26     epsilon=norm(x0-x,"inf");
27 end
28 disp(x0);
29 %% G-S
30 clc;clear;

```

```

31 clc;clear;
32 A=zeros(50,50)+diag(12*ones(1,50));
33 for i=1:length(A(:,1))-1
34     A(i,i+1)=-2;
35 end
36 for i=1:length(A(:,1))-2
37     A(i,i+2)=1;
38 end
39 for i=1:length(A(:,1))%行
40     for j=1:i%列
41         A(i,j)=A(j,i);
42     end
43 end
44 D=diag(diag(A));%对角阵
45 L=D-tril(A);%下三角阵
46 U=D-triu(A);%上三角阵
47 B=(D-L)\U;
48 b=5*ones(50,1);
49 f=(D-L)\b;
50 x=A\b;%精确值
51 x0=0.01*ones(50,1);%迭代初始值
52 epsilon=1;%误差初始值
53 while epsilon>10^(-14)
54     x0=B*x0+f;
55     epsilon=norm(x0-x,"inf");
56 end
57 disp(x0);
    
```

结果:

表 8: 运行结果

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0.46379552381655	0.537284605199966	0.509022924601331	0.498221634436174	0.498941860239762	0.499985351248131	0.500088723890136	0.500015318846052
x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
0.499994793266975	0.499997856913468	0.500000108425199	0.500000201576687	0.500000022610945	0.499999986238572	0.499999995873979	0.500000000530294
x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
0.500000000439039	0.50000000023939	0.499999999966017	0.49999999992557	0.500000000001741	0.500000000000922	0.499999999999990	0.499999999999923
x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}	x_{31}	x_{32}
0.499999999999998	0.499999999999987	0.499999999999921	0.500000000000001	0.500000000000015	0.5000000000001741	0.499999999992558	0.499999999996019
x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}	x_{39}	x_{40}
0.500000000023937	0.500000000439038	0.500000000530295	0.499999995873980	0.499999986238572	0.500000022610945	0.500000201576688	0.500000108425199
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
0.499997856913467	0.499994793266975	0.500015318846052	0.500088723890136	0.499985351248131	0.498941860239762	0.498221634436174	0.509022924601331
x_{49}	x_{50}						
0.537284605199966	0.463795523816550						