

数值分析方法

作业2

Problem 1

解: 由题意: $f(x) = -x^3 - \cos x$, $f'(x) = -3x^2 + \sin x$
代码如下:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int i=0;
7      double p,p0,delta=1;
8      scanf("%lf",&p0);
9      while(i<=1){
10         p=p0-(-pow(p0,3)-cos(p0))/\
11         (-3*pow(p0,2)+sin(p0));
12         delta=p-p0;
13         if (delta<0){
14             delta=-delta;
15         }
16         printf("n = %2d, p = %12.8lf, \
17         |p-p0| = %.8lf\n",++i,p,delta);
18         p0=p;
19     }
20 }
```

可得:

n	p_n	$ p_n - p_{n-1} $
1	-0.88033290	0.11966710
2	-0.86568416	0.01464874

$$\therefore p_2 = -0.86568416$$

$$\therefore f'(0) = 0$$

\therefore 不能使用 $p_0 = 0$

Problem 2

解: (i) 由题意:

$$f(x) = b - \frac{1}{x}$$

则:

$$f'(x) = \frac{1}{x^2}$$

由牛顿迭代法:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{b - \frac{1}{x_k}}{\frac{1}{x_k^2}} = 2x_k - bx_k^2$$

所以

$$|\epsilon_{k+1}| = \frac{|x - x_{k+1}|}{|x|} = \frac{|\frac{1}{b} - x_{k+1}|}{\frac{1}{b}} = |1 - bx_{k+1}| = |b^2x_k^2 - 2bx_k + 1| = (bx_k - 1)^2$$

又因为

$$\epsilon_k = \frac{\frac{1}{b} - x_k}{\frac{1}{b}} = 1 - bx_k$$

所以

$$|\epsilon_{k+1}| = \epsilon_k^2$$

证毕.

(ii) 由(i)得:

$$\begin{aligned} \left|x_{k+1} - \frac{1}{b}\right| &= \left|2x_k - bx_k^2 - \frac{1}{b}\right| = \frac{|2bx_k - b^2x_k^2 - 1|}{b} = \frac{|bx_k - 1|^2}{b} \\ \left|x_k - \frac{1}{b}\right| &= \frac{|bx_k - 1|}{b}, \quad k = 0, 1, 2, \dots \end{aligned}$$

所以

$$\left|x_{k+1} - \frac{1}{b}\right| = \frac{|bx_k - 1|^2}{b}$$

同理可得

$$\left|x_k - \frac{1}{b}\right| = \frac{|bx_k - 1|}{b} = \frac{|bx_{k-1} - 1|^2}{b}$$

所以

$$\begin{aligned} |bx_k - 1| &= |bx_{k-1} - 1|^2 \\ \therefore \left|x_{k+1} - \frac{1}{b}\right| &= \frac{|bx_k - 1|^2}{b} = \frac{|bx_{k-1} - 1|^4}{b} = \dots = \frac{|bx_0 - 1|^{2^{k+1}}}{b} \end{aligned}$$

由题意:

$$0 < x_0 < \frac{2}{b}$$

则: $|bx_0 - 1| \in [0, 1)$

$$\therefore \lim_{k \rightarrow \infty} \left|x_{k+1} - \frac{1}{b}\right| = \lim_{k \rightarrow \infty} \frac{|bx_0 - 1|^{2^{k+1}}}{b} = 0$$

即数列 $\{x_k\}_{k=0}^{\infty}$ 收敛到 $\frac{1}{b}$, 证毕.

Problem 3

解: a. 由题意, 令

$$\begin{aligned}f_1 &= 3x_1 - \cos(x_2x_3) - \frac{1}{2} \\f_2 &= 4x_1^2 - 625x_2^2 + 2x_2 - 1 \\f_3 &= e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3}\end{aligned}$$

则雅可比矩阵

$$\mathbf{J} = \begin{pmatrix} 3 & x_3 \sin(x_2x_3) & x_2 \sin(x_2x_3) \\ 8x_1 & -1250x_2 + 2 & 0 \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{pmatrix}$$

代码如下:

```
1  #include<stdio.h>
2  #include<math.h>
3
4  #define e 2.718281828459
5  #define pi 3.1415926535898
6
7  int main()
8  {
9      int i=0;
10     double f1,f2,f3;
11     double x01,x02,x03;
12     double x1,x2,x3;
13     double y1,y2,y3;
14     double j1[3],j2[3],j3[3];
15     double jn1[3],jn2[3],jn3[3];
16     double det;
17     scanf("%lf,%lf,%lf",&x01,&x02,&x03);
18     while (i<=1){
19         f1=3*x01-cos(x02*x03)-0.5;
20         f2=4*pow(x01,2)-625*pow(x02,2)+2*x02-1;
21         f3=pow(e,-x01*x02)+20*x03+(10*pi-3)/3;
22         j1[0]=3; //Jacobi Matrix
23         j1[1]=x03*sin(x02*x03);
24         j1[2]=x02*sin(x02*x03);
25         j2[0]=8*x01;
26         j2[1]=-1250*x02+2;
27         j2[2]=0;
28         j3[0]=-x02*pow(e,-x01*x02);
29         j3[1]=-x01*pow(e,-x01*x02);
```

```

30      j3 [2]=20;
31      det=j1 [0]*(j2 [1]*j3 [2]-j2 [2]*j3 [1])\
32      -j2 [0]*(j1 [1]*j3 [2]-j1 [2]*j3 [1])+
33      j3 [0]*(j1 [1]*j2 [2]-j1 [2]*j2 [1]);
34      //the Determinant
35
36      jn1 [0]=(j2 [1]*j3 [2]-j2 [2]*j3 [1])/det;
37      jn1 [1]=(j1 [2]*j3 [1]-j1 [1]*j3 [2])/det;
38      jn1 [2]=(j1 [1]*j2 [2]-j1 [2]*j2 [1])/det;
39      jn2 [0]=(j2 [2]*j3 [0]-j2 [0]*j3 [2])/det;
40      jn2 [1]=(j1 [0]*j3 [2]-j1 [2]*j3 [0])/det;
41      jn2 [2]=(j2 [0]*j1 [2]-j1 [0]*j2 [2])/det;
42      jn3 [0]=(j2 [0]*j3 [1]-j2 [1]*j3 [0])/det;
43      jn3 [1]=(j1 [1]*j3 [0]-j1 [0]*j3 [1])/det;
44      jn3 [2]=(j1 [0]*j2 [1]-j2 [0]*j1 [1])/det;
45      //inverse matrix of the Jacobi Matrix
46
47      y1=jn1 [0]*f1+jn1 [1]*f2+jn1 [2]*f3;
48      y2=jn2 [0]*f1+jn2 [1]*f2+jn2 [2]*f3;
49      y3=jn3 [0]*f1+jn3 [1]*f2+jn3 [2]*f3;
50      x1=x01-y1;
51      x2=x02-y2;
52      x3=x03-y3;
53      printf("n = %d, x1 = %.8f, x2 = %.8f\
54 , x3 = %.8f\n",++i,x1,x2,x3);
55      x01=x1;
56      x02=x2;
57      x03=x3;
58   }
59 }

```

可得:

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$
1	0.50000000	0.50000000	-0.52359878
2	0.50016669	0.25080364	-0.51738743

$\therefore \mathbf{x}^{(2)} = (0.50016669, 0.25080364, -0.51738743)^T$.

b. 由题意, 令

$$f_1 = x_1^2 + x_2 - 37$$

$$f_2 = x_1 - x_2^2 - 5$$

$$f_3 = x_1 + x_2 + x_3 - 3$$

则雅可比矩阵

$$\mathbf{J} = \begin{pmatrix} 2x_1 & 1 & 0 \\ 1 & -2x_2 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

代码如下:

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int i=0;
7      double f1, f2, f3;
8      double x01, x02, x03;
9      double x1, x2, x3;
10     double y1, y2, y3;
11     double j1[3], j2[3], j3[3];
12     double jn1[3], jn2[3], jn3[3];
13     double det;
14     scanf("%lf, %lf, %lf", &x01, &x02, &x03);
15     while (i <= 1) {
16         f1 = pow(x01, 2) + x02 - 37;
17         f2 = x01 - pow(x02, 2) - 5;
18         f3 = x01 + x02 + x03 - 3;
19         j1[0] = 2 * x01;
20         j1[1] = 1;
21         j1[2] = 0;
22         j2[0] = 1;
23         j2[1] = -2 * x02;
24         j2[2] = 0;
25         j3[0] = 1;
26         j3[1] = 1;
27         j3[2] = 1;
28         det = j1[0] * (j2[1] * j3[2] - j2[2] * j3[1]) \
29             - j2[0] * (j1[1] * j3[2] - j1[2] * j3[1]) + \
30             j3[0] * (j1[1] * j2[2] - j1[2] * j2[1]);
31         //the Determinant
32
33         jn1[0] = (j2[1] * j3[2] - j2[2] * j3[1]) / det;

```

```

34      jn1 [1]=(j1 [2]*j3 [1]-j1 [1]*j3 [2])/det;
35      jn1 [2]=(j1 [1]*j2 [2]-j1 [2]*j2 [1])/det;
36      jn2 [0]=(j2 [2]*j3 [0]-j2 [0]*j3 [2])/det;
37      jn2 [1]=(j1 [0]*j3 [2]-j1 [2]*j3 [0])/det;
38      jn2 [2]=(j2 [0]*j1 [2]-j1 [0]*j2 [2])/det;
39      jn3 [0]=(j2 [0]*j3 [1]-j2 [1]*j3 [0])/det;
40      jn3 [1]=(j1 [1]*j3 [0]-j1 [0]*j3 [1])/det;
41      jn3 [2]=(j1 [0]*j2 [1]-j2 [0]*j1 [1])/det;
42      //inverse matrix of the Jacobi Matrix
43
44      y1=jn1 [0]*f1+jn1 [1]*f2+jn1 [2]*f3;
45      y2=jn2 [0]*f1+jn2 [1]*f2+jn2 [2]*f3;
46      y3=jn3 [0]*f1+jn3 [1]*f2+jn3 [2]*f3;
47      x1=x01-y1;
48      x2=x02-y2;
49      x3=x03-y3;
50      printf("n = %d, x1 = %.8f, x2 = %.8f\
51 , x3 = %.8f\n",++i,x1,x2,x3);
52      x01=x1;
53      x02=x2;
54      x03=x3;
55  }
56 }

```

可得:

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$
1	0.50000000	37.00000000	-39.00000000
2	4.35087719	18.49122807	-19.84210526

$\therefore \mathbf{x}^{(2)} = (4.35087719, 18.49122807, -19.84210526)^T$.

Problem 4

解: a. 由题意:

$$f_1 = 15x_1 + x_2^2 - 4x_3 - 13$$

$$f_2 = x_1^2 + 10x_2 - x_3 - 11$$

$$f_3 = x_2^3 - 25x_3 + 22$$

令

$$g(\mathbf{x}) = f_1^2 + f_2^2 + f_3^2$$

$$\therefore \nabla g(\mathbf{x}) = \left(\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \frac{\partial g}{\partial x_3} \right) = 2 \begin{pmatrix} 2x_1^3 + 15x_2^2 - 2x_1x_3 + 20x_1x_2 + 203x_1 - 60x_3 - 195 \\ 10x_1^2 + 2x_2^3 - 3x_2^5 - 66x_2^2 + 74x_2 + 30x_1x_2 - 10x_3 - 110 + 75x_2^2x_3 \\ -x_1^2 - 4x_2^2 - 25x_2^3 - 60x_1 - 10x_2 + 643x_3 - 487 \end{pmatrix}$$

代码如下:

```

1  #include<stdio.h>
2  #include<math.h>
3  #include<stdlib.h>
4
5  #define e 2.718281828459
6  #define pi 3.1415926535898
7
8  int main()
9  {
10     int i=0;
11     double f1,f2,f3;
12     double a,a0,a1,a2,a3;
13     double h1,h2,h3;
14     double g,g0,g1=0,g2,g3;
15     double x01,x02,x03;
16     double z1,z2,z3,z0;
17     double tol,min=10;
18     scanf("%lf,%lf,%lf,%lf",&x01,&x02,&x03,&tol);
19     while (fabs(min-g1)>=tol){
20         f1=15*x01+pow(x02,2)-4*x03-13;
21         f2=pow(x01,2)+10*x02-x03-11;
22         f3=pow(x02,3)-25*x03+22;
23         g1=f1*f1+f2*f2+f3*f3;
24         z1=2*(2*x01*x01*x01+15*x02*x02-2*x01*\
25             x02+203*x01-60*x03-195);
26         z2=2*(10*x01*x01+2*x02*x02*x02-3*\
27             pow(x02,5)-66*pow(x02,2)+74*x02\
28             +30*x01*x02-10*x03-110+\
29             75*pow(x02,2)*x03);
30         z3=2*(-x01*x01-4*x02*x02-25*pow(x02,3)\
31             -60*x01-10*x02+643*x03-487);
32         z0=sqrt(pow(z1,2)+pow(z2,2)+pow(z3,2));
33         //z0 is the L2-norm of the vector z
34         if (z0==0){
35             printf("failed , z0 = 0, n = \

```

```

36 %d, x1 = %.8f, x2 = %.8f, x3 = %.8f\n", i, x01, x02, x03);
37     exit(0);
38 }
39 z1/=z0;
40 z2/=z0;
41 z3/=z0;
42 a1=0;
43 a3=1;
44 g3=pow((15*(x01-a3*z1)+pow(x02-a3*z2,\
45     2)-4*(x03-a3*z3)-13),2)+pow((pow\
46     (x01-a3*z1,2)+10*(x02-a3*z2)-\
47     (x03-a3*z3)-11),2)+pow((pow\
48     (x02-a3*z2,3)-25*(x03-a3*z3)+22),2);
49 while (g3>=g1){
50     a3/=2;
51     g3=pow((15*(x01-a3*z1)+\
52     pow(x02-a3*z2,2)-\
53     4*(x03-a3*z3)-13),2)+\
54     pow((pow(x01-a3*z1,2)+\
55     10*(x02-a3*z2)-(x03-a3*z3)\
56     -11),2)+pow((pow(x02-a3*z2\
57     ,3)-25*(x03-a3*z3)+22),2);
58     if (a3<0.01*tol){
59         printf("failed , a3 = \
60 %.8lf , n = %d, x1 = %.8lf, x2 = %.8lf, x3 = %.8lf\n", \
61 a3, i, x01, x02, x03);
62         exit(0);
63     }
64 }
65 a2=a3/2;
66 g2=pow((15*(x01-a2*z1)+pow(x02-a2*z2\
67     ,2)-4*(x03-a2*z3)-13),2)+pow((pow\
68     (x01-a2*z1,2)+10*(x02-a2*z2)-\
69     (x03-a2*z3)-11),2)+pow((pow\
70     (x02-a2*z2,3)-25*(x03-a2*z3)+22),2);
71 h1=(g2-g1)/a2;
72 h2=(g3-g2)/(a3-a2);
73 h3=(h2-h1)/a3;
74 a0=(a2-h1/h3)/2;
75 g0=pow((15*(x01-a0*z1)+pow(x02-a0*z2\
76     ,2)-4*(x03-a0*z3)-13),2)+pow((pow\

```



```

77         (x01-a0*z1,2)+10*(x02-a0*z2)-\
78         (x03-a0*z3)-11),2)+pow((pow\
79         (x02-a0*z2,3)-25*(x03-a0*z3)+22),2);
80     min=g0;
81     if(g3<min){
82         min=g3;
83     }
84     if(min==g0){
85         a=a0;
86     }
87     else {
88         a=a3;
89     }
90     x01-=a*z1;
91     x02-=a*z2;
92     x03-=a*z3;
93     printf("n = %d, x1 = %.8lf, x2 = \
94     %.8lf, x3 = %.8lf, g = %.8lf\n",++i,x01,x02,x03,min);
95     }
96 }

```

令 $\mathbf{x} = (1, 1, 1)^T$, 可得:

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$g^{(n)}$
1	1.05321268	0.99712364	0.91946190	0.71702448
2	1.05481094	0.99718700	0.91937309	0.71640813

\therefore 一个近似解为 $\mathbf{x} = (1.05481094, 0.99718700, 0.91937309)^T$.

b. 由题意:

$$f_1 = 10x_1 - 2x_2^2 + x_2 - 2x_3 - 5$$

$$f_2 = 8x_2^2 + 4x_3^2 - 9$$

$$f_3 = 8x_2x_3 + 4$$

令

$$g(\mathbf{x}) = f_1^2 + f_2^2 + f_3^2$$

$$\therefore \nabla g(\mathbf{x}) = \left(\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \frac{\partial g}{\partial x_3} \right) = 2 \begin{pmatrix} 100x_1 - 20x_2^2 + 10x_2 - 20x_3 - 50 \\ 136x_2^3 - 6x_2^2 + 128x_2x_3^2 - 40x_1x_2 + 8x_2x_3 + 10x_1 - 123x_2 + 30x_3 - 5 \\ 32x_3^3 + 128x_2^2x_3 + 4x_2^2 - 20x_1 + 30x_2 - 68x_3 + 10 \end{pmatrix}$$

代码如下:

```

1  #include<stdio.h>
2  #include<math.h>
3  #include<stdlib.h>
4
5  #define e 2.718281828459
6  #define pi 3.1415926535898
7
8  int main()
9  {
10     int i=0;
11     double f1,f2,f3;
12     double a,a0,a1,a2,a3;
13     double h1,h2,h3;
14     double g,g0,g1=0,g2,g3;
15     double x01,x02,x03;
16     double z1,z2,z3,z0;
17     double tol,min=10;
18     scanf("%lf,%lf,%lf,%lf",&x01,&x02,&x03,&tol);
19     while (fabs(min-g1)>=tol){
20         f1=10*x01-2*pow(x02,2)+x02-2*x03-5;
21         f2=8*pow(x02,2)+4*pow(x03,2)-9;
22         f3=8*x02*x03+4;
23         g1=f1*f1+f2*f2+f3*f3;
24         z1=2*(100*x01-20*x02*x02+10*x02\
25             -20*x03-50);
26         z2=2*(136*x02*x02*x02-6*x02*x02+128*\
27             x02*x03*x03-40*x01*x02+8*x02*x03\
28             +10*x01-123*x02+30*x03-5);
29         z3=2*(32*x03*x03*x03+128*x02*x02*x03+4\
30             *pow(x02,2)-20*x01+30*x02-68*x03+10);
31         z0=sqrt(pow(z1,2)+pow(z2,2)+pow(z3,2));
32         //z0 is the L2-norm of the vector z
33         if (z0==0){
34             printf("failed , z0 = 0, n = \
35 %d, x1 = %.8f, x2 = %.8f, x3 = %.8f\n",i,x01,x02,x03);
36             exit(0);
37         }
38         z1/=z0;
39         z2/=z0;
40         z3/=z0;
41         a1=0;

```

```

42         a3=1;
43         g3=pow((10*(x01-a3*z1)-2*pow((x02-\
44             a3*z2),2)+(x02-a3*z2)-2*(x03-a3*z3)\
45             -5),2)+pow((8*pow((x02-a3*z2),2)+\
46             4*pow((x03-a3*z3),2)-9),2)+pow\
47             ((8*(x02-a3*z2)*(x03-a3*z3)+4),2);
48         while (g3>=g1){
49             a3/=2;
50             g3=pow((10*(x01-a3*z1)-2*pow(\
51                 (x02-a3*z2),2)+(x02-a3*z2)\
52                 -2*(x03-a3*z3)-5),2)+pow(\
53                 (8*pow((x02-a3*z2),2)+4*\
54                 pow((x03-a3*z3),2)-9),2)+\
55                 pow((8*(x02-a3*z2)*(x03-\
56                 a3*z3)+4),2);
57             if (a3<0.01*tol){
58                 printf("failed , a3 = \
59 %8lf , n = %d, x1 = %8lf , x2 = %8lf , x3 = %8lf\n",\
60 a3,i,x01,x02,x03);
61                 exit(0);
62             }
63         }
64         a2=a3/2;
65         g2=pow((10*(x01-a2*z1)-2*pow((x02-\
66             a2*z2),2)+(x02-a2*z2)-2*(x03-a2*z3)\
67             -5),2)+pow((8*pow((x02-a2*z2),2)+\
68             4*pow((x03-a2*z3),2)-9),2)+pow(\
69             (8*(x02-a2*z2)*(x03-a2*z3)+4),2);
70         h1=(g2-g1)/a2;
71         h2=(g3-g2)/(a3-a2);
72         h3=(h2-h1)/a3;
73         a0=(a2-h1/h3)/2;
74         g0=pow((10*(x01-a0*z1)-2*pow((x02\
75             -a0*z2),2)+(x02-a0*z2)-2*(x03-a0*z3)\
76             -5),2)+pow((8*pow((x02-a0*z2),2)+\
77             4*pow((x03-a0*z3),2)-9),2)+pow(\
78             (8*(x02-a0*z2)*(x03-a0*z3)+4),2);
79         min=g0;
80         if (g3<min){
81             min=g3;
82         }

```

```

83         if (min==g0){
84             a=a0;
85         }
86         else {
87             a=a3;
88         }
89         x01-=a*z1;
90         x02-=a*z2;
91         x03-=a*z3;
92         printf("n = %d, x1 = %.8lf, x2 = \
93 %.8lf, x3 = %.8lf, g = %.8lf\n",++i,x01,x02,x03,min);
94     }
95 }

```

令 $\mathbf{x} = (1, 1, 1)^T$, 可得:

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$g^{(n)}$
1	0.92753552	0.49999507	0.57970600	81.62449064
2	0.67589563	0.65390087	0.63774376	69.52080786
3	0.52253096	0.89189684	-0.32133767	7.88536591
4	0.51592829	1.03474947	-0.38996809	0.65404290
5	0.53651718	1.02248306	-0.46597604	0.14174062
6	0.52198314	1.00728128	-0.46735906	0.07238155
7	0.51584441	1.01388912	-0.47501960	0.04200303

∴ 一个近似解为 $\mathbf{x} = (0.51584441, 1.01388912, -0.47501960)^T$.