
数值计算方法实验报告

不动点迭代法与牛顿迭代法

米科润 19 信计二班

201905755824

June 2, 2021

目录

1	实验题目	1
2	实现算法	1
3	程序代码	1
4	实验结果	4

1 实验题目

1、生成一个系数由 (0, 7) 之间的随机数组成的七次多项式, 并使用不多点迭代法和牛顿迭代法找出解、收敛阶、收敛速度。

2、将 $x^2 + (\frac{5}{4} \times y - \sqrt{|x|})^2 - 4 = 0$ 的图像画出来。

2 实现算法

书本 217 至 227 的算法

3 程序代码

不动点迭代法代码:Fixed Point Iteration.m

```
1      function []=F_P_Iteration(f,x0)
2      %%
3      %初始设置
4      syms x
5      f(x)=f;
6      m=20;TOL=1e-6;
7      df(x)=diff(f(x));
8      a=-100;b=100;
9      phi=df(a:0.05:b);
10     x1=x0;
11     t=0;
12     %%
13     %判断是否收敛
14     if abs(double(phi)) <= 1
15         disp('具有全局收敛性');
16     end
17     %%
18     %迭代
19     for i=1:m
20         xx=f(x1);
21         if abs(double(xx)-double(x1))<=TOL
22             disp('迭代次数为: ');
23             disp(i);
```

```

24         t=1;
25         break;
26     end
27     x1=double(xx);
28 end
29 if t==1
30     disp('不动点为:');
31     disp(double(xx));
32 else
33     warning('未迭代到解');
34 end
35 %%
36 %判断收敛阶
37 if t==1
38     c=zeros(1,7);
39     for j = 1:7
40         Δ(x)=diff(f,j);
41         c(j)=double(Δ(xx));
42     end
43     I1=find(c,1,'first');
44     disp('收敛阶为:');
45     disp(I1);
46 end
47 end

```

牛顿迭代法代码:Newton Iteration.m

```

1     function []=Newton_Iteration(f,x0)
2     %%
3     syms x
4     f(x)=f;
5     m=20;TOL=1e-6;
6     a=-2;b=2;
7     dy(x)=diff(f(x));
8     ddy=dy(a:0.05:b);
9     x1=x0;
10    t=0;
11    %%
12    if (double(f(a))*double(f(b))<0) & (double(ddy)≠0) & ...

```

```

        (abs(double(f(a))/double(dy(a)))<(b-a)) ...
13    & (abs(double(f(b))/double(dy(b)))<(b-a))
14        disp('迭代收敛唯一解, 收敛阶数为2');
15    end
16    %%
17    for i=1:m
18        xx=f(x1);
19        if abs(double(xx)-double(x1))≤TOL
20            disp('迭代次数为: ');
21            disp(i);
22            t=1;
23            break;
24        end
25        x1=double(xx);
26    end
27    if t==1
28        disp('解为: ');
29        disp(double(xx));
30    else
31        warning('未迭代到解');
32    end
33    end

```

运行代码:test iteration.m

```

1  clc , clear
2  A=7*rand(8,1);
3  syms x
4  f=A(7)*x^7+A(6)*x^6+A(5)*x^5+A(4)*x^4+A(3)*x^3+A(2)*x^2+A(1)*x+A(8);
5  x0=0.5;
6  disp('不动点迭代结果: ');
7  for k = 1:7
8      g=(f-x^k)^(1/k);
9      F_P_Iteration(g,x0);
10 end
11 %F_P_Iteration(f,x0);
12 f=x-f/diff(f);
13 disp('牛顿迭代法迭代序列为: ');
14 pretty(f);

```

```
15 disp('牛顿迭代结果:');  
16 Newton_Iteration(f,x0);  
17 syms x y  
18 F=x.^2+(5/4.*y-sqrt(abs(x))).^2-4;  
19 fimplicit(F,[-3,3 -3,3])
```

4 实验结果

运行 test Iteration.m 可得运算结果:

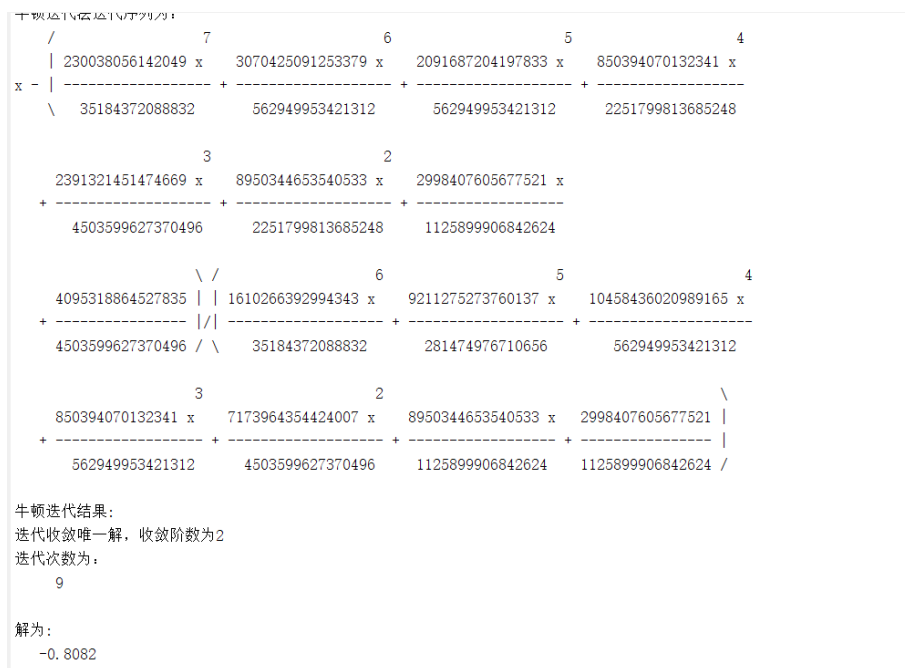


图 1: result1

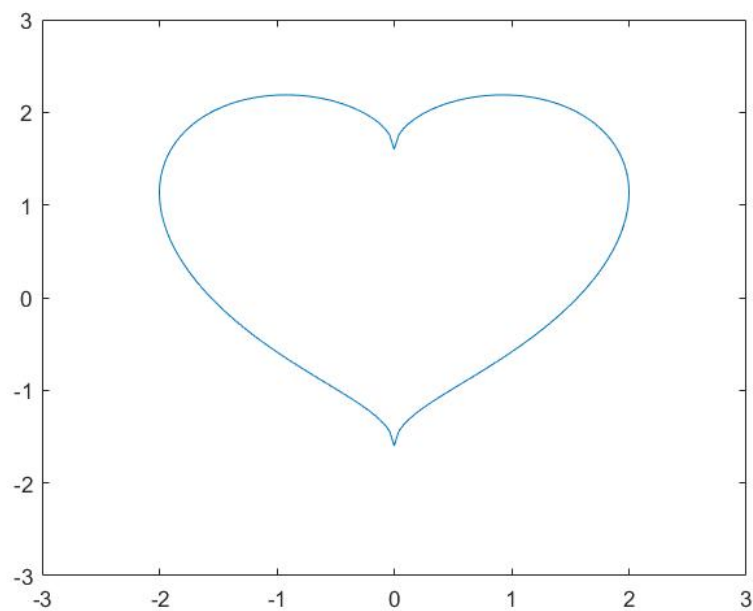


图 2: result2