# 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

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

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

#### 牛顿迭代法代码:Newton Iteration.m

```
function [] = Newton_Iteration(f,x0)
            %%
2
            syms x
            f(x)=f;
            m=20;TOL=1e-6;
5
            a = -2; b = 2;
            dy(x)=diff(f(x));
            ddy=dy(a:0.05:b);
            x1=x0;
            t = 0;
10
            %%
11
            if (double(f(a))*double(f(b))<0) & (double(ddy)\neq0) & ...
12
```

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

#### 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:

```
850394070132341 x
  35184372088832
                     562949953421312
                                      562949953421312
                                                      2251799813685248
   2391321451474669 x
                   8950344653540533 x
                                     2998407605677521 x
     4503599627370496
                     2251799813685248
                                     1125899906842624
   4095318864527835 | | 1610266392994343 x
                                    9211275273760137 x
                                                    10458436020989165 x
   ----- |/| -----
   4503599627370496 / \ 35184372088832
                                     281474976710656
   850394070132341 x 7173964354424007 x 8950344653540533 x 2998407605677521 |
    562949953421312 4503599627370496 1125899906842624 1125899906842624 /
牛顿迭代结果:
迭代收敛唯一解,收敛阶数为2
迭代次数为:
   9
解为:
 -0. 8082
```

图 1: result1

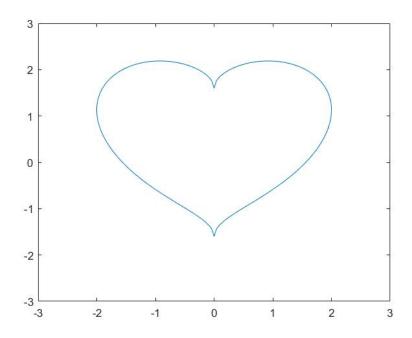


图 2: result2