实验三 牛顿迭代法及其修正

# 实验目的

1. 掌握牛顿迭代法
2. 掌握两种修正的牛顿迭代法：简化牛队迭代法和牛顿下山法
3. 通过练习熟悉牛顿迭代法求方程近似根的过程

# 实验环境

1. 计算机
2. MATLAB集成环境

# 实验内容与代码

## 用简单迭代法和牛顿迭代法求出方程在附近的根，讨论两种迭代法的收敛速度。

简单迭代法与牛顿迭代法的编程算法如此实时脚本末尾所示。

则可由如下编程代码实现：

clear;clc;

f\_x = inline('x^3-x-1','x');

err = 1e-5;

[simpleRes, simpleCount] = simpleIteration(f\_x, 1.5, err)

count=1, res=1.3572

count=2, res=1.3309

count=3, res=1.3259

count=4, res=1.3249

count=5, res=1.3248

count=6, res=1.3247

count=7, res=1.3247

simpleRes = 1.3247

simpleCount = 7

coef = [1 0 -1 -1];

[newtonRes, newtonCount] = newtonIteration(coef, 1.5)

newtonRes = 1.3247

newtonCount = 4

对比迭代次数，牛顿迭代法的收敛速度更快。

## 构造立方根表，要求：

1. 构造 101-111 的立方表，以矩阵的形式输出
2. 保留 6 位有效数字
3. 至少使用两种方法。例如：牛顿迭代法，简化牛顿迭代法

clear;clc;

coef=[1,0,0,0];

for i=1:11

coef(4)=-(i+100);

g(1,i)=newtonIteration(coef, fix((i+100)^(1/3)));

g(2,i)=simplifiedNewtonIteration(coef, fix((i+100)^(1/3)));

end

disp(num2str(g));

4.657 4.6723 4.6875 4.7027 4.7177 4.7326 4.7475 4.7622 4.7769 4.7914 4.8059

4.657 4.6723 4.6875 4.7027 4.7177 4.7326 4.7475 4.7622 4.7769 4.7914 4.8059

## 用牛顿下山法求解方程在附近的根，要求初值选取.

牛顿下山法算法见脚本末尾。

可由如下编程代码实现：

clear;clc;

coef = [1 0 -1 -1];

[res, count] = newtonDownHill(coef, 0.4, 0.1)

res = 1.3247

count = 10

## 此实时脚本中使用的函数：

function [res, count] = simpleIteration(f\_x, x, errRequest)

err = 1e4;

count = 0;

while err >= errRequest

res = (x+1)^(1/3);

err = abs(f\_x(res));

x = res;

count = count + 1;

disp("count="+count+", res="+res)

end

end

function [res, count]=newtonIteration(coef, x)

coeff = sym2poly( diff( poly2sym(coef) ) );

count = 0;

res = x - (polyval(coef, x))/(polyval(coeff, x));

while true

fxx= res - (polyval(coef, res))/(polyval(coeff, res));

if res == fxx

break;

else

res = fxx;

count = count+1;

end

end

end

function [res, count] = simplifiedNewtonIteration(coef,x)

coeff=sym2poly(diff(poly2sym(coef)));

c=polyval(coeff,x);

res=x-polyval(coef,x)/c;

count = 0;

while true

c=polyval(coeff,res);

fxx=res-polyval(coef,res)/c;

count = count + 1;

if res==fxx

break;

else

res=fxx;

end

end

end

function [res,count] = newtonDownHill(coef,x,e\_l)

coeff=sym2poly(diff(poly2sym(coef)));

fx=x-(polyval(coef,x))/(polyval(coeff,x));

lambda=1;

while true

fxx=fx-lambda\*(polyval(coef,fx))/(polyval(coeff,fx));

if abs(fx)>abs(fxx)

[res,count]=newtonIteration(coef,fxx);

break;

elseif lambda>e\_l

lambda=lambda/2;

else

disp('请重新选根');

break;

end

end

end

# 实验总结：

通过此次实验，熟悉掌握了牛顿迭代法及两种修正的牛顿迭代法，并通过联系熟悉了牛顿迭代法求方程近似根的过程。