实验一 利用 Newton 迭代法和割线法求方程的根

孙寒石 06219109 2021 年 3 月 24 日

一、实验目的及原理

Newton 迭代公式：$x_{k+1} = x_k - f(x_k)/f'(x_k)$，Newton 迭代法是在根 x^* 附近以 x_k 作为第k次迭代值,然后带进迭代公式，得到第 $k+1$ 次迭代近似值，看是否满足用户要求精度，不满足的话，继续迭代，得到满足用户要求的精度的近似值。实验要求利用 Newton 迭代法求以下方程的根：

- $x^2 - e^x = 0$
- $xe^x - 1 = 0$
- $\lg x + x - 2 = 0$

割线法：割线法，又称弦割法、弦法，是基于牛顿法的一种改进，基本思想是用弦的斜率近似代替目标函数的切线斜率，并用割线与横轴交点的横坐标作为方程式的根的近似。它是求解非线性方程的根的一种方法，属于逐点线性化方法。迭代公式：$x_{k+1} = x_k - f(x_k)/[f(x_k) - f(x_{k-1})] * (x_k - x_{k-1})$，实验要求利用割线法求以下方程的根：

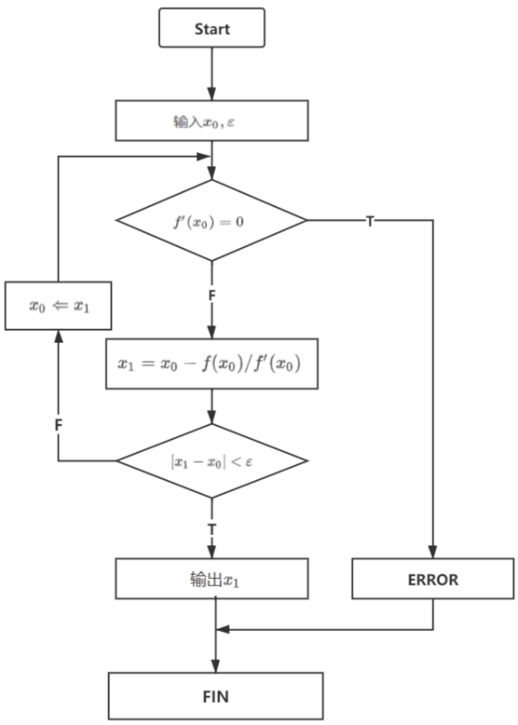
- $x^2 - e^x = 0$
- $xe^x - 1 = 0$
- $\lg x + x - 2 = 0$

二、实验环境

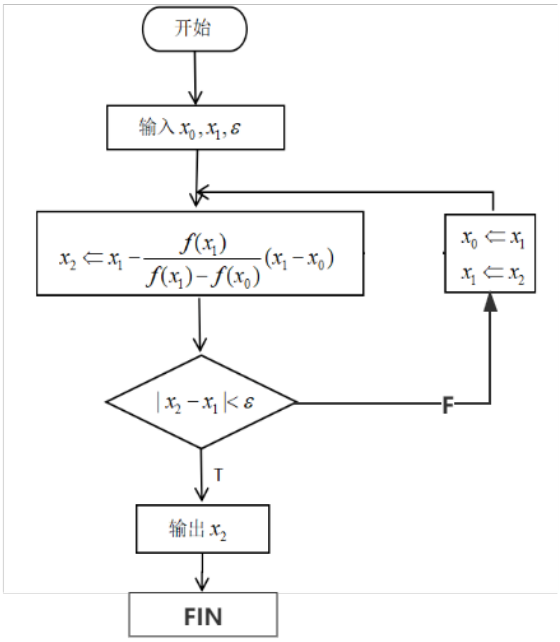
- 编程语言：Python
- 编程环境：Jupyter Notebook

三、实验步骤

Newton 迭代法算法框图：



割线法算法框图：



四、实验代码及结果

Newton 迭代法：

• 函数定义（三个方程分别对应的函数）

```
import math
def f1(x):
    if (2*x-math.exp(x) == 0):
        return "error"
    else:
        return x-(x*x-math.exp(x))/(2*x-math.exp(x))

def f2(x):
    if ((x+1)*math.exp(x) == 0):
        return "error"
    else:
        return x-(x*math.exp(x)-1)/((x+1)*math.exp(x))

def f3(x):
    if (x <= 0):
        return "error"
    if (1/(x*math.log(10))+1 == 0):
        return "error"
    else:
        return x-(math.log(x)/math.log(10)+x-2)/(1/(x*math.log(10))+1)
```

• 对第一个方程进行求解

```
x = 1
temp =0
while (abs(x-temp)>0.00000000000000001):
    temp = x
    x = f1(x)
    print(x)
```

输出结果：

```
-1.3922111911773332
-0.8350875293671394
-0.7098340945745987
-0.7034834042362847
-0.703467422599462
-0.7034674224983917
-0.7034674224983917
```

所以根为：-0.7034674224983917

• 对第二个方程进行求解

```
x = 1
temp =0
while (abs(x-temp)>0.00000000000000001):
    temp = x
    x = f2(x)
    print(x)
```

输出结果：

```
0.6839397205857212
0.5774544771544498
0.5672297377301171
0.5671432965302959
0.567143290409784
0.5671432904097838
0.5671432904097838
```

所以根为：0.5671432904097838

• 对第三个方程进行求解

```
x = 1
temp =0
while (abs(x-temp)>0.00000000000000001):
    temp = x
    x = f3(x)
    print(x)
```

输出结果：

1.6972068934358862

1.7553795434500208

1.75557949700258

1.755579499261178

1.755579499261178

所以根为：1.755579499261178

结果：

- $x^2 - e^x = 0$ 根为 -0.7034674224983917
- $xe^x - 1 = 0$ 根为 0.5671432904097838
- $\lg x + x - 2 = 0$ 根为 1.755579499261178

割线法：

• 函数定义(三个方程分别对应的函数)

```
import math
def f4(x1,x2):
    if ((x1*x1-math.exp(x1))-(x2*x2-math.exp(x2)) == 0):
        return x1
    else:
        return x1-(x1*x1-math.exp(x1))/((x1*x1-math.exp(x1))-(x2*x2-math.exp(x2)))*(x1-x2)

def f5(x1,x2):
    if ((x1*math.exp(x1)-1)-(x2*math.exp(x2)-1) == 0):
        return x1
    else:
        return x1-(x1*math.exp(x1)-1)/((x1*math.exp(x1)-1)-(x2*math.exp(x2)-1))*(x1-x2)

def f6(x1,x2):
    if (x1 <= 0 or x2 <= 0):
        return "error"
```

```
    if ((math.log(x1)/math.log(10)+x1-2)-(math.log(x2)/math.log(10)+x2-2) == 0):
        return x1
    else:
        return x1-(math.log(x1)/math.log(10)+x1-2)/((math.log(x1)/math.log(10)+x1-2)-(math.log(x2)/math.log(10)+x2-2))*(x1-x2)
```

- 对第一个方程进行求解

```
x1 = 1
x2 = 0
while (abs(x2-x1)>0.000000000000000001):
    temp = x1
    x1 = f4(x1,x2)
    x2 = temp
    print(x1)
```

输出结果：

```
-1.3922111911773332
-0.206127512714066O4
-0.5778333714634147
-0.7330842972939767
-0.7019544393346077
-0.7034498327184769
-0.7034674330341212
-0.7034674224983183
-0.7034674224983917
-0.7034674224983917
```

所以根为：-0.7034674224983917

- 对第二个方程进行求解

```
x1 = 1
x2 = 0
while (abs(x2-x1)>0.000000000000000001):
    temp = x1
    x1 = f5(x1,x2)
    x2 = temp
    print(x1)
```

输出结果：

0.36787944117144233
0.5033143321329856
0.5786158630519874
0.5665323438586994
0.5671375717285394
0.567143293272022 4
0.5671432904097705
0.5671432904097838
0.5671432904097838

所以根为：0.5671432904097838

- 对第三个方程进行求解

```
x1 = 1
x2 = 0.5
while (abs(x2-x1)>0.0000000000000001):
    temp = x1
    x1 = f6(x1,x2)
    x2 = temp
    print(x1)
```

输出结果：

1.624196350581785
1.7476883930575944
1.7555181658929515
1.755579471853784
1.755579499261083
1.7555794992611777
1.755579499261178
1.755579499261178

所以根为：1.755579499261178

结果：

- $x^2 - e^x = 0$ 根为 -0.7034674224983917
- $xe^x - 1 = 0$ 根为 0.5671432904097838
- $\lg x + x - 2 = 0$ 根为 1.755579499261178

五、分析和讨论

通过程序输出，我们可以得到如下结果：

- $x^2 - e^x = 0$ 根为 -0.7034674224983917
- $xe^x - 1 = 0$ 根为 0.5671432904097838
- $\lg x + x - 2 = 0$ 根为 1.755579499261178

通过观察发现，Newton 迭代法的收敛速度要比割线法快，需要迭代的次数更少。在牛顿法中，每一步都需要计算当前点的导数值，需要手动求导。如果不想人工求导，可以使用两点割线来代替切线，从而可以用割线法来进行计算。也就是说，通过牺牲迭代速度来避免复杂的求导运算和计算工作量。

六、附件

- 计算方法 exp1.ipynb
- 计算方法 exp1.pdf