数值实验报告2

实验名称	实验2.2及2.3	实验时间	2025.3.8

姓名	秦浩政 郭凯平 刘佳鑫 刘桂凡	班级	数据2301	学号	2306030214 2306020510 2309050116 2309050117	成绩	

一、实验目的，内容

本次数值实验通过具体的数值实验来研究不动点迭代法, 以及用$Steffensen$迭代加速方法的可行性，同时考虑算法的收敛性、
初值的选择、收敛速度以及算法的复杂性与稳定性等。

二、算法描述

1. **迭代法**

　　我们经过一定的变形, 使原本需要求零点的函数$f(x)$变成$x=\phi(x)$的形式并以此作为迭代的格式, 即$x_{k+1}=\phi(x_k)$,由此得到的序列$\{x_k\}$会逐渐收敛于$\phi(x)$的不动点x^*, 即求的原函数的零点. 具体思路:

- 定义函数`iteration_fixedpoint()`, 其中的参数有:
  - `function1`: 对应输入函数
  - `x0`: 初值
  - `TOL`: 误差要求
  - `Nmax`: 最大迭代次数
  - `n`: 对应的算法序号

  每次迭代先使用公式计算出新的近似值，再检查收敛性，若满足误差精度条件, $|x_n-x_{n+1}|<TOL$则跳出循环，将结果赋值给变量$res1$。否
  则继续迭代，若k 达到最大迭代次数 $Nmax$时仍未收敛，即循环结束时$res1$的值为$None$, 程序输出迭代失败的提示信息。
- 给出三个测试用例, 分别为对$x^2-x-1=0$的三种变形, 用`iteration_fixedpoint()`进行迭代观察三种变形的敛散性, 收敛速度, 以及结果的精度

2. $Steffensen$ **方法**

   $Steffensen$ 方法是一种迭代法的改良，使用两次迭代得到新的迭代值，构造一个新的迭代公式，使得收敛性大大增强，收敛速度大大增加。具体思路:

- 定义函数`steffensen()`, 其中的参数有:
  - `function`: 对应输入函数
  - `x0`: 初值
  - `TOL`: 误差要求

  - `Nmax`: 最大迭代次数

每次迭代中先计算出伪新值x_{k+1}和x_{k+2}, 记为$\overline{x}_{k+1}$和$\overline{x}_{k+2}$, 然后使用迭代公式$x_{k+1} = \frac{x_0\overline{x}_{k+2} - \overline{x}_{k+1}^2}{x_k - 2x_{k+1} + x_{k+2}}$, 由此进行迭代. 直到$|x_n - x_{n+1}| < TOL$跳出循环. 返回数值结果.

- 通过估算零点大小, 结合此处导数大小, 大致判断出两个方程的初始点分别为$x_0 = 0.5$以及$x = 1.5$, 使用$Steffensen$迭代法进行计算并输出结果.

三、程序代码

1. 迭代法

```python
from math import sqrt

3 usages
def iteration_fixedpoint(function1, x0, TOL, Nmax, n) -> None:
    """使用不动点算法迭代计算方程的解
        function1: 对应输入函数
        x0: 初值
        TOL: 误差要求
        Nmax: 最大迭代次数
        n: 对应的算法序号"""
    k = 1 #用于统计迭代次数
    res1 = None
    xk = x0
    while k <= Nmax: # 如果迭代失败k应变为Nmax + 1
        y = function1(xk)
        if abs(y - xk) < TOL:
            res1 = y
            break
        xk = y
        k += 1

    if res1 is None:
        print("算法1失败")
    else:
        print(f"算法{n}的结果为{res1}\n迭代了{k}次")

```

```python
1 usage
def function1(x) -> int:
    return x**2 - 1
1 usage
def function2(x) -> int:
    return 1 + 1/x
1 usage
def function3(x) -> int:
    return sqrt(x + 1)

iteration_fixedpoint(function1, x0: 1.5, TOL: 1e-8, Nmax: 20, n: 1)
iteration_fixedpoint(function2, x0: 1.5, TOL: 1e-8, Nmax: 20, n: 2)
iteration_fixedpoint(function3, x0: 1.5, TOL: 1e-8, Nmax: 20, n: 3)
```

2. $Steffensen$迭代法

```python
from math import exp

5 usages
def steffensen(function, x0, TOL, Nmax) -> float:

    for _ in range(Nmax):
        y = function(x0)
        z = function(y)
        res = (x0*z - y**2)/(x0-2*y+z)
        if abs(res-x0) <= TOL:
            return res
        x0 = res

    return None

3 usages
def e(x) -> float:
    return exp(-x)
steffensen(e, x0: 0.5, TOL: 1e-5, Nmax: 20)
2 usages
def phi(x) -> float:
    return x**3 -1
if steffensen(e, x0: 0.5, TOL: 1e-5, Nmax: 20) is None:
    print("算法失败")
else:
    print(steffensen(e, x0: 0.5, TOL: 1e-5, Nmax: 20))

if steffensen(phi, x0: 1.5, TOL: 1e-5, Nmax: 20) is None:
    print("算法失败")
else:
    print(steffensen(phi, x0: 1.5, TOL: 1e-5, Nmax: 20))

```

四、数值结果

1. 迭代法

```
算法1失败
算法2的结果为1.6180339901755971
迭代了19次
算法3的结果为1.6180339860648685
迭代了15次
```

2. $Steffensen$迭代法

```
0.5671432909625258
1.3247179574287269
```

五、计算结果分析

通过本次数值实验，我们分别使用 **不动点迭代法** 和 **Steffensen 迭代法** 对非线性方程的根进行求解，并分析了不同方法的收敛性、收敛速度、初值选择对结果的影响等。实验结果表明，两种方法各有优劣，具体分析如下：

1. 迭代法分析

- 从实验结果来看，对于给定方程 $x^2 - x - 1 = 0$ 的不同变形，**不动点迭代法的收敛性依赖于迭代函数的选择**。在三种变形形式中，二三两种变形收敛较快，而第一种并不收敛。这与迭代函数 $\phi(x)$ 的导数 $\phi'(x)$ 有关。根据不动点迭代法的收敛性理论，若 $|\phi'(x^*)| < 1$，则迭代过程是收敛的，否则可能会发散。如变形二迭代19次后结果为1.6180339901755971, 而变形三迭代十五次即可得到1.6180339860648685; 同时将参数中的误差变为10^{-12}时我们发现结果为1.6180339887496715, 也就是说变形三迭代十五次就在10^{-8}量级上获得了比变形二更高的精确度, 尽管变形二迭代次数更多
- 迭代法的**初值选择对收敛性有较大影响**，若初值较好（即靠近实际根且 $\phi(x)$ 满足收敛条件），则能够较快收敛，否则可能会进入振荡或发散状态。

2. Steffensen 迭代法分析

- 通过实验结果可以看出，相比普通的不动点迭代法，Steffensen 迭代法的**收敛速度大大提升**，通常仅需较少的迭代步数即可达到预期的精度要求。这与 Steffensen 迭代法的二阶收敛性（相比于普通不动点迭代法的一阶收敛性）相吻合。

- 该方法对初值的依赖性较低，相较于普通不动点迭代法，在更大的范围内都能保持较好的收敛性。
- 但 Steffensen 迭代法的计算复杂度稍高，每一步迭代需要额外计算一次 $\phi(x)$，因此在某些计算量较大的问题上，可能会增加额外的计算成本。

3. 方法比较与总结

- **收敛性**：Steffensen 方法比不动点迭代法收敛性更好，能更快收敛到精确解，而普通不动点迭代法的收敛性取决于 $\phi(x)$ 的选择。
- **收敛速度**：Steffensen 方法的收敛速度显著高于普通不动点迭代法，实验结果表明 Steffensen 方法能够在较少的迭代次数内达到较高精度。
- **初值选择的敏感性**：普通不动点迭代法对初值较为敏感，选择不当可能导致收敛缓慢甚至发散，而 Steffensen 迭代法对初值的要求较低，鲁棒性更强。
- **计算复杂度**：Steffensen 方法的单步计算量较普通迭代法略高，但由于其收敛速度更快，因此整体计算效率较高。

综上，本次数值实验验证了 **Steffensen 迭代法** 的有效性，并说明了普通不动点迭代法的适用条件及其局限性。在实际应用中，如果迭代函数 $\phi(x)$ 适合不动点迭代法且满足收敛条件，可以使用普通不动点迭代法进行求解；而若希望提高收敛速度或减少迭代次数，则 Steffensen 迭代法是一个更优的选择。

六、计算中出现的问题,解决方法及体会

1. 问题及解决方法

- 在实验2.2.1中, 我们很早就得到了算法失败的结论, 然而算法失败的原因有很多, 如可能是因为收敛速度较慢, 导致在Nmax限制内并不能得到精度要求, 也有可能是算法并不收敛, 或者是区间选择有问题. 后来我们选择了打印过程的方法, 确定了算法失败的原因是其在此区间震荡并不收敛.

2. 体会

通过此次实验，我们更深一步理解了迭代法的实现过程、收敛性及收敛速度，熟悉了两种不同的迭代法的推导过程以及使用优缺点, 对 Steffensen 方法有了新的认识与了解. 也发现实验过程中面对未知的算法失败有多样化的可能性, 需要我们去注意排查.

教师评语	**指导教师: 年 月 日**