实验报告四 牛顿迭代法

题目（摘要）

使用牛顿迭代法计算公式：

$$x_0 = \alpha$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$n = 0,1,\ldots$$

求非线性方程$f(x) = 0$的根x^*。

其中：

问题	$f(x)$	ε_1	ε_2	N	x_0
1	$cosx - x$	10^{-6}	10^{-4}	10	$\pi/4$
2	$e^{-x} - sinx$				0.6
3	$x - e^{-x}$				0.5
4	$x^2 - 2xe^{-x} + e^{-2x}$				0.5

前言：（目的和意义）

目的：

1. 用牛顿迭代法求解方程的根
2. 了解迭代法的原理
3. 代码实现牛顿迭代法
4. 考虑特殊情况

意义：

1. 学习使用 matlab 语言
2. 提升对牛顿迭代法的认识
3. 增加对上机作业的经验

数学原理

一般地，牛顿迭代法具有局部收敛性，为了保证迭代收敛，要求，对充分小的$\delta > 0, \alpha \in O(x^*, \delta)$。

如果$f(x) \in C^2[a,b], f(x^*) = 0, f(x) \neq 0$，那么，对充分小的$\delta > 0$，当$\alpha \in O(x^*, \delta)$时，由牛顿迭代法计算出的$\{x_n\}$收敛于$x^*$,且收敛速度为 2 阶的；

如果

$$\begin{cases} f(x) \in C^m[a,b] \\ f(x^*) = f'(x^*) = \cdots = f^{(m-1)}(x^*) = 0 \\ f^{(m)}(x^*) \neq 0 \end{cases}$$

那么，对于充分小的$\delta > 0$，当$\alpha \in O(x^*, \delta)$时，由牛顿迭代法计算出的$\{x_n\}$收敛于$x^*$,且收敛速度为 1 阶的。

程序设计流程

输入：初值α，精度$\varepsilon_1, \varepsilon_2$,最大迭代次数 N

输出：方程$f(x) = 0$根x^*的近似值或者计算失败的标志

流程：

1 置$n = 1$

2 当$n \leq N$时：

2.1 置$F = f(x_{n-1}), DF = f'(x_{n-1})$

如果$|F| < \varepsilon_1$,输出x_0；停机

如果$|DF| < \varepsilon_2$,输出失败标志；停机

2.2 置$x_n = x_{n-1} - \frac{F}{DF}$

2.3 置$Tol = |x_n - x_{n-1}|$

如果$Tol < \varepsilon_1$,输出x_n；停机

2.4 置$n = n + 1$

3 输出失败标志

4 停机

实验结果、结论与讨论

问题	$f(x)$	ε_1	ε_2	N	x_0	x_n
1	$cosx-x$	10^{-6}	10^{-4}	10	$\pi/4$	0.739085178106010
2	$e^{-x}-sinx$				0.6	0.588532742847979
3	$x-e^{-x}$				0.5	0.567143165034862
4	$x^2-2xe^{-x}+e^{-2x}$				0.5	0.566605704128158

显然，四个问题都能成功完成迭代，没有失败输出。

而且在 N 足够大的情况下，根的近似值能足够精确。

思考题：

1. 选择一个x_0，使得$f(x)$接近于精确解，实际运算中可以考虑二分法大致确定一个不太精确的数开始迭代；
2. 数学上来看显然两个解是完全相等的，但是在程序顺利完成的情况下得出的结果有所不同。

   是因为前一问是单根，二阶收敛；后一问是重根，是局部线性收敛，的迭代次数就会更多，要求更准确的解才能满足题目给出的误差条件。所以两者的结果有所不同。