2022秋计算方法–实验报告#3

姓名：刘家骥

学号：PB20071417

2022 年 10 月 2 日

运行环境：　Windows 11， Red Panda Dev C++, C语言

实验内容

给定非线性方程

$$f(x) \triangleq \arctan(x) + \sin(x) - 1 = 0$$

分别编写Newton 迭代 (通常也称 Newton-Raphson 迭代)

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

和 Damped-Newton (DN) 迭代

$$x_{k+1} = x_k - \tau\frac{f(x_k)}{f'(x_k)} \quad (\text{其中阻尼参数}\tau : 0 < \tau < 1)$$

的程序. 取阻尼参数$\tau = 0.5$, 两种迭代方法的初始点x_0 依次取值为 −4, −3, −2, −1, 0, 1, 2, 3, 5, 7, 9; 停止条件为 $|f(x_k)| < 10^{-8}$ 或 迭代步数 $k > 10^4$ (此时, 可认为迭代失败).

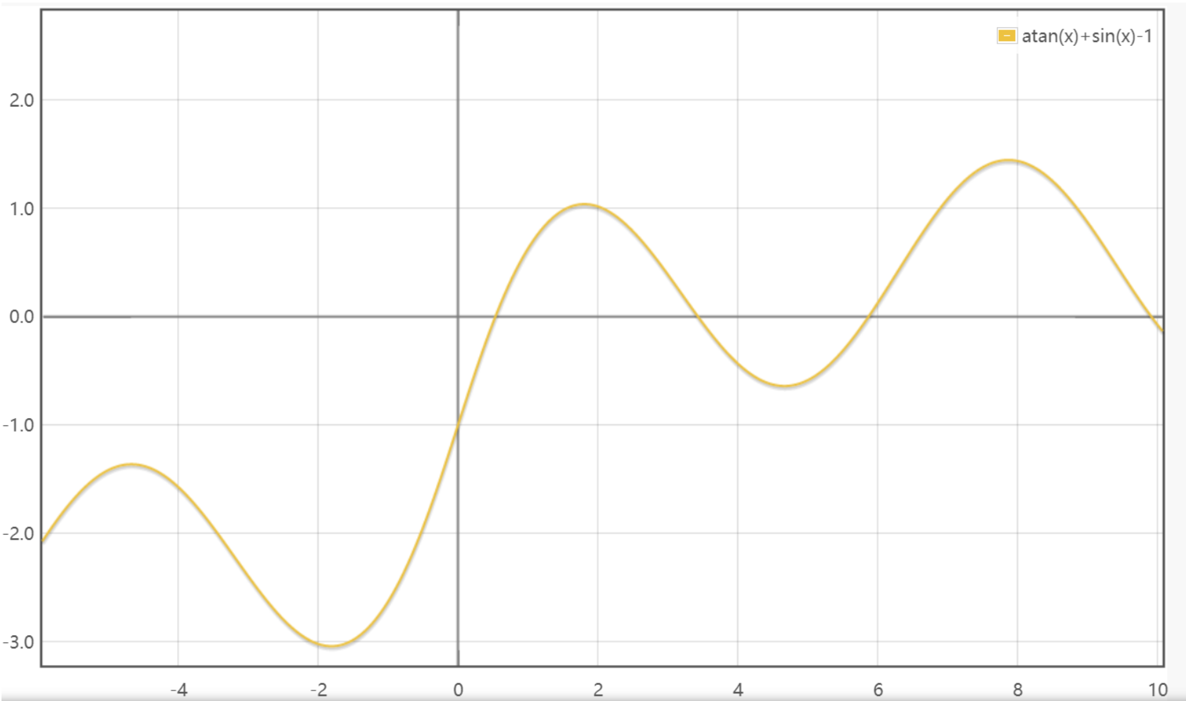
- 列表给出两种迭代方法在初始点 x0 依次取值为 −4, −3, −2, −1, 0, 1, 2, 3, 5, 7, 9时的迭代 步数（如果迭代步数超过1万步，视为迭代失败）以及相应的数值解 x_k (保留小数点 后6位);

- 比较并分析两种方法的优劣，给出合理的算法分析并作实验小结。

1 数值结果 (作表或图)

	第一种方法		第二种方法	
x	步数k	x_k的值	步数k	x_k的值
-4	10001	-6.389409	37	22.545593
-3	10001	-6.389409	34	0.534332
-2	10001	-16.287301	50	0.534332
-1	6	3.433055	27	0.534332
0	4	0.534332	27	0.534332
1	4	0.534332	26	0.534332
2	5	5.869439	25	3.433055
3	3	3.433055	26	3.433055
5	5	5.869439	23	5.869439
7	4	5.869439	27	5.869439
9	4	9.914378	27	9.914378

为作参考, 作出$f(x)$函数图象如下:



2 算法分析

1)第一种方法也就是我们上课提到的正常的Newton迭代法, 第二种方法是我们上课时没有接触到的一种改进方法. 在运算中可以看到除了$x=-4,-3,-2$的三种情况下, 第一种方法迭代失败以外, 取其他x值, 两种方法都能迭代成功, 但是两者的迭代结果有时又会有明显不同.

2)对于第一种方法, 除去那几个迭代失败的情况, 迭代次数k一般都是个位数级的, 即迭代次数一般很少. 但是结果有问题: 得到的x_k确实都是近似的解, 但是并不是给定x_0附近的解. 例如$x_0=-1$时离x_0最近的精确解应该是在0.534332附近, 但是迭代出来的$x_k=3.433055$, 不是我们所期待的那个解.

3)对于第二种方法, 除去$x = -4$时迭代产生的解距离最近的精确解较远外, 其他的迭代结果x_k都能很好的满足要求, 即: 距离x_0较近, 且二者中间不会跳过解. 但是略显不足的是, 迭代次数k明显多于第一种方法, 少则25次, 多则50次.

4)课上提到过, 第一种方法, 即Newton迭代法的一个优势就是收敛速度快, 这点在实验中有体现(除开迭代失败的几次以外, 方法1的k值明显很小). 但是它有一个明显的缺陷: 与初始值x_0密切相关, 初值不好可能导致迭代失败或者是结果x_k距离x_0较远, 导致其不是一个好的结果. 因此这种方法通常是要求在取x_0时, $f(x_0)$就已经接近0, 或者我们已经知道了$f(x)$的图象大致长什么样子, 这样才有利于之后的迭代计算.

5)但是第二种迭代法(Damped-Newton, 阻尼牛顿迭代)可以基本上消除第一种方法的缺陷. 因为在其中加入了一个阻尼系数τ, 使得迭代效果变好, 这个是可以在图象中体现的. 第一种方法的核心就是在$(x_i, f(x_i))$点作切线, 切线和x轴的交点横坐标就是x_{i+1}, 而第二种方法过那个点做的直线斜率就是把切线斜率除以了$\tau\ \ (\tau < 1)$, 导致斜率绝对值增大, 使得x_{i+1}更加靠近原先的x_i, 这个就直接导致了迭代速度变慢, 迭代次数k变多, 但是这也避免了x_{i+1}取到一个很不合理的, 距离x_i比较远的值, 从而使得第二种方法的迭代结果x_k更加符合要求.

3 实验小结

综上所述, 通过以上的实验和分析, 我认为第二种方法相对而言更加符合要求, 但是其迭代次数偏多.