

数值分析第二次上机练习报告

——非线性方程组求根

周懿

力 1-2021013053

一、问题的描述

试用以下方法求方程:

$$x^3 + 2x^2 + 10x - 20 = 0$$

在 $x_0 = 1$ 附近的根.(准确解为 $x^* = 1.368808107\dots$) 希望精度达到 10^{-7} (要注意避免死循环):

(1) $x_{k+1} = \frac{20 - 2x_k^2 - x_k^3}{10};$

(2) $x_{k+1} = \sqrt[3]{20 - 10x_k - 2x_k^2};$

(3) 方法 (1) 的 Steffensen 加速方法;

(4) 方法 (2) 的 Steffensen 加速方法;

(5) Newton 法;

二、方法描述

(a) 直接构造迭代函数

这里我们尝试三个迭代函数:

$$x_{k+1} = \varphi_1(x_k) = \frac{20 - 2x_k^2 - x_k^3}{10} \quad (1)$$

$$x_{k+1} = \varphi_2(x_k) = \sqrt[3]{20 - 10x_k - 2x_k^2} \quad (2)$$

$$x_{k+1} = \varphi_3(x_k) = \frac{20}{x^2 + 2x + 10} \quad (3)$$

我们根据上述三个迭代式进行程序的编写, 经过迭代可以获知这个迭代函数是否收敛以及收敛者收敛所需的迭代次数。

(b) Steffensen 加速法

理论上可以证明, 对于任意一个迭代函数

$$x_{k+1} = \varphi(x),$$

我们对其进行 Steffensen 加速, 得到新的迭代函数:

$$\psi(x) = x - \frac{(\varphi(x) - x)^2}{\varphi(\varphi(x)) - 2\varphi(x) + x} \quad (4)$$

则这个迭代函数是二阶收敛的。我们根据式(4)编写 MATLAB 程序，分别将式(1)(2)(3)代入，即可对上述三个迭代方法实现加速。

(c) Newton 迭代法

Newton 迭代法的原理如右面图 1所示，其原理是从初始值开始，不断地对曲线做切线，将切点投影至 x 轴作为新的迭代值继续做切线。

理论上可以证明，在曲线在局部凹凸性保持一致的情况下，经过多轮迭代之后解序列可以收敛到精确解 x^* 。迭代函数可以写成下式：

$$\varphi(x) = x - \frac{f(x)}{f'(x)} \quad (5)$$

进而得到迭代式：

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (6)$$

我们根据式(6)编写 MATLAB 程序，即可实现牛顿迭代法求解非线性方程组。

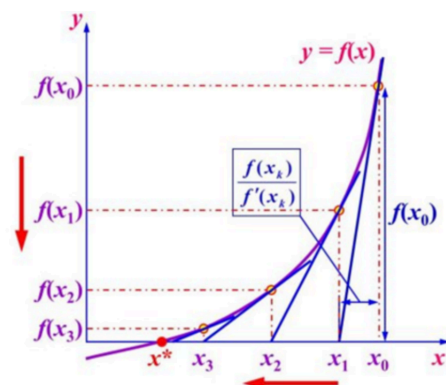


图 1: 牛顿迭代法示意图

三、 方案设计

我们通过编写 MATLAB 程序来进行非线性方程的求解。在程序文件中：

1. main.m 是主程序文件;
2. directIter.m 是直接迭代法的程序。给它任意一个目标函数、迭代函数、初始值、精度和最大迭代次数（最后两者可选。如果未设置，则默认精度为 $e = 10^{-7}$ ，最大迭代次数为 100），它将在预设的迭代次数以内调用输入的迭代函数对初始值进行迭代。若收敛到指定精度，则停止迭代，并输出迭代结果和迭代总次数；如果迭代次数已用完，程序将会提示不收敛。
3. steffen_acc.m 是使用 Steffensen 加速法迭代的程序。给它任意一个目标函数、预设的迭代函数、初始值、精度和最大迭代次数（最后两者可选。如果未设置，则默认精度为 $e = 10^{-7}$ ，最大迭代次数为 100），它将在预设的迭代次数以内对输入的迭代函数进行 Steffensen 加速并生成新的迭代函数，用新的迭代函数对初始值进行迭代。若收敛到指定精度，则停止迭代，并输出迭代结果和迭代总次数；如果迭代次数已用完，程序将会提示不收敛。
4. newton.m 是牛顿迭代法的程序。给它任意一个目标函数、初始值、精度和最大迭代次数（最后两者可选。如果未设置，则默认精度为 $e = 10^{-7}$ ，最大迭代次数为 100），它对初始值进行牛顿迭代。若收敛到指定精度，则停止迭代，并输出迭代结果和迭代总次数；如果迭代次数已用完，程序将会提示不收敛。

5. further_exp.m 是基于主程序实验结果的一些异常现象进行深入探索的程序，主要研究了两个迭代函数不收敛的原因。

四、 计算结果及其分析

图 5是我们根据程序计算结果得到的数据。此处为了记录方便，我们只记录了前 40 次迭代的结果。可以看出前两个直接构造的函数都不收敛，而第三个直接构造的函数是收敛的。我们对这三个函数都使用 Steffensen 加速，发现两个不收敛的函数加速后都收敛了，而收敛的函数加速后收敛得更快，均在 5 步之内。可见 Steffensen 加速可以使迭代函数获得更高的收敛阶。我们从数据中可以发现，前两个函数的解出现了震荡现象，我们把迭代的过程绘成折线图，如图 2所示。

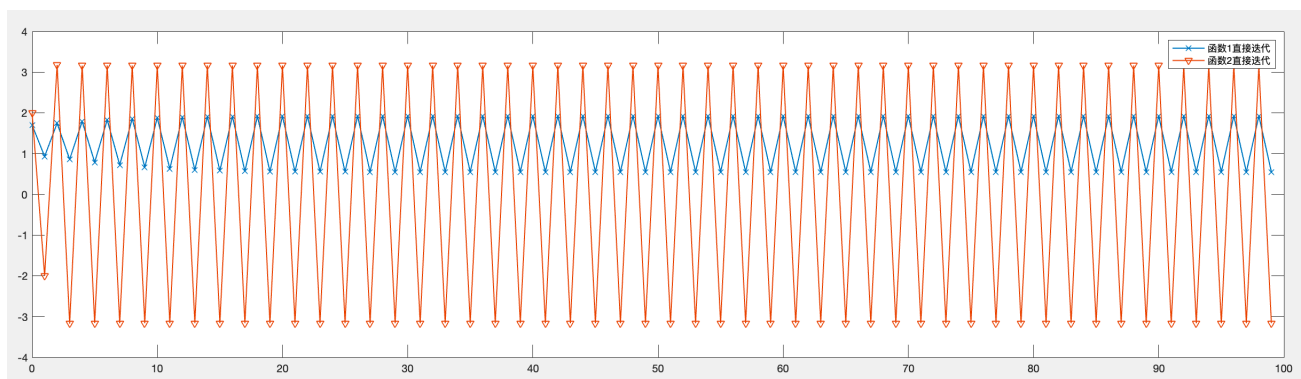


图 2: 函数 1 和函数 2 的迭代过程示意图

关于出现这种震荡现象的原因，我们进行了进一步的探究。我们使用函数 φ_1 和函数 φ_2 的二次迭代函数：

$$\varphi'_1 = \varphi_1(\varphi_1(x))$$

$$\varphi'_2 = \varphi_2(\varphi_2(x))$$

并绘出它们和曲线 $y = x$ 的交点，结果如图 34所示。

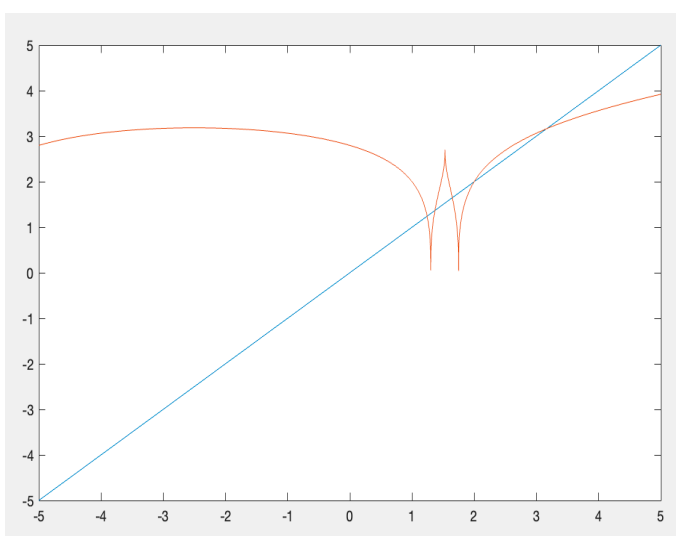
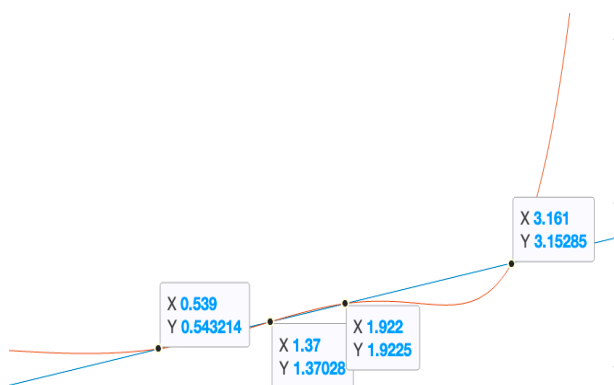


图 3: 函数 $\varphi'_1(x)$ 与曲线 $y = x$ 交点示意图 图 4: 函数 $\varphi'_2(x)$ 与曲线 $y = x$ 交点示意图

从图 3可以看出，函数 φ'_1 与 $y = x$ 有四个交点，这些都是该函数的稳定点（二阶不动点），观察它们的值，其中一个值恰为我们需要的解（也就是一阶不动点），另外三个稳定点均非不动点，这些稳定点的值恰好与我们数据表中出现震荡的两个值吻合，证明这两个稳定点正是迭代出现震荡的原因。同时通过观察斜率可以发现，最右端的稳定点实际上并不收敛，一旦我们设置的初始值在该值附近或超过该值，迭代结果可以接近正无穷，而实验证明，当我们在主程序中将初值设置为 4 时，迭代 100 次的结果已经为 inf。

从图 4可以看出，函数 φ'_2 与 $y = x$ 有五个交点，这些都是该函数的稳定点（二阶不动点），观察它们的值，其中一个值恰为我们需要的解（也就是一阶不动点），另外四个稳定点均非不动点，这些稳定点的值恰好与我们数据表中出现震荡的两个值吻合，证明这两个稳定点正是迭代出现震荡的原因。同时通过观察斜率可以发现，只有最右端（约为 3.16）是收敛的，其他的稳定点均不收敛。然而我们进一步的实验发现，无论初值取为何值，迭代之后都会收敛到最右端的稳定点处（但并不是我们要的解，所以迭代器仍然会提示不收敛），这与函数的因变量值分布有关，导致无论初值如何取，因为函数值始终分布在该稳定点附近或高于该稳定点，导致迭代值最终必然进入该稳定点的收敛范围，最终表现为迭代是收敛的。

五、 结论

许多非线性方程无法直接求得精确解，因而我们不得不使用迭代法求解。然而并不是依据方程随意构造一个迭代函数都可以收敛。从实验中我们可以看出，有些迭代函数是不收敛的。但是使用 Steffensen 加速后，无论一开始选用什么迭代函数，最后都必然收敛，而且会以很快的速度收敛（至少二阶收敛）。而 Newton 迭代法在可以收敛的情况下是二阶收敛，因而使用 Steffensen 加速和 Newton 迭代法都可以很快速的获得一个较为精确的解。非常有趣的是，我们在实验中发现直接构造的函数 1 和函数 2 都拥有稳定点（二阶不动点），这使得直接迭代出现了震荡现象，最终导致不收敛，关于这一现象我们在本次实验也进行了进一步探究，最终发现了稳定点和收敛过程的一些联系。

Iteration	Function 1	Function 2	Function 3	Func 1+ Steffensen	Func 2+ Steffensen	Func 3+ Steffensen	Newton
1	1.7000E+00	2.0000E+00	1.5385E+00	1.3335E+00	1.2000E+00	1.3708E+00	1.4118E+00
2	9.3070E-01	-2.0000E+00	1.2950E+00	1.3684E+00	1.2737E+00	1.3688E+00	1.3693E+00
3	1.7461E+00	3.1748E+00	1.4018E+00	1.3688E+00	1.3083E+00	1.368808	1.3688E+00
4	8.5780E-01	-3.1717E+00	1.3542E+00	1.368808 Iter: 4	1.3473E+00	1.368808 Iter: 3	1.368808 Iter: 4
5	1.7897E+00	3.1614E+00	1.3753E+00		1.3667E+00		
6	7.8612E-01	-3.1616E+00	1.3659E+00		1.3688E+00		
7	1.8278E+00	3.1623E+00	1.3701E+00		1.368808 Iter: 7		
8	7.2115E-01	-3.1623E+00	1.3682E+00				
9	1.8585E+00	3.1623E+00	1.3691E+00				
10	6.6729E-01	-3.1623E+00	1.3687E+00				
11	1.8812E+00	3.1623E+00	1.3689E+00				
12	6.2642E-01	-3.1623E+00	1.3688E+00				
13	1.8969E+00	3.1623E+00	1.3688E+00				
14	5.9773E-01	-3.1623E+00	1.3688E+00				
15	1.9072E+00	3.1623E+00	1.3688E+00				
16	5.7882E-01	-3.1623E+00	1.3688E+00				
17	1.9136E+00	3.1623E+00	1.3688E+00				
18	5.6689E-01	-3.1623E+00	1.3688E+00				
19	1.9175E+00	3.1623E+00	1.3688E+00				
20	5.5959E-01	-3.1623E+00	1.3688E+00				
21	1.9198E+00	3.1623E+00	1.3688E+00				
22	5.5522E-01	-3.1623E+00	1.3688E+00				
23	1.9212E+00	3.1623E+00	1.368808 Iter: 23				
24	5.5262E-01	-3.1623E+00					
25	1.9220E+00	3.1623E+00					
26	5.5110E-01	-3.1623E+00					
27	1.9225E+00	3.1623E+00					
28	5.5020E-01	-3.1623E+00					
29	1.9228E+00	3.1623E+00					
30	5.4968E-01	-3.1623E+00					
31	1.9230E+00	3.1623E+00					
32	5.4937E-01	-3.1623E+00					
33	1.9231E+00	3.1623E+00					
34	5.4919E-01	-3.1623E+00					
35	1.9231E+00	3.1623E+00					
36	5.4909E-01	-3.1623E+00					
37	1.9231E+00	3.1623E+00					
38	5.4903E-01	-3.1623E+00					
39	1.9232E+00	3.1623E+00					
40	5.4900E-01	-3.1623E+00					
Converged	×	×	✓	✓	✓	✓	✓

图 5: 计算结果

六、 特别鸣谢

特别感谢力 1 的黄天成同学，我在发现直接迭代出现震荡现象时尚未将这一现象与二阶不动点联系在一起，收到他关于二阶不动点的发现启发之后，我们合作进一步探究了对于函数 φ_1 稳定点与迭代时出现震荡现象的联系，在他的启发之下我又对 φ_2 也进行了类似的探究，均发现了有趣的现象。