

关于对牛顿迭代法的优化

谢志超¹, 王玲^{1*}, 王晨², 龚佃选¹

¹华北理工大学理学院, 河北 唐山

²华北理工大学人工智能学院, 河北 唐山

收稿日期: 2022年3月20日; 录用日期: 2022年4月14日; 发布日期: 2022年4月24日

摘要

牛顿迭代法是数值计算中最重要的常用方法之一, 但是其明显缺点是每次迭代都需要函数的导数。本文利用中心差商结合弦截法对牛顿迭代法进行了改进, 提出了几种改良的迭代格式。在相同函数的及初始值的前提下, 数值实验显示, 改进后的牛顿迭代法的迭代速度与牛顿法相比有了明显提升。

关键词

牛顿法, 差商, 弦截法, 迭代法

On the Optimization of Newton's Iterative Method

Zhichao Xie¹, Ling Wang^{1*}, Chen Wang², Dianxuan Gong¹

¹College of Science, North China University of Science and Technology, Tangshan Hebei

²School of Intelligence, North China University of Science and Technology, Tangshan Hebei

Received: Mar. 20th, 2022; accepted: Apr. 14th, 2022; published: Apr. 24th, 2022

Abstract

Newton's iterative method is one of the most important common methods in numerical calculation, but its obvious disadvantage is that each iteration needs the derivative of the function. In this paper, Newton's iterative method is improved by using central difference quotient and chord section method, and several improved iterative schemes are proposed. Under the premise of the same function and initial value, numerical experiments show that the iteration speed of the improved Newton's iteration method is significantly improved compared with Newton's method.

*通讯作者。

Keywords

Newton's Method, Difference Quotient, Chord Section Method, Iteration Method

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在我们现实生活中,有些问题可以转换为求非线性方程组的数值解的问题,迭代法是求解非线性方程组的常用方法,不同的迭代方法可以构造出不同的迭代公式。牛顿迭代法是基于泰勒展式构造的经典迭代公式,由于其形式简单而且是二阶收敛的,是求解线性方程组 $f(x)=0$ 近似解的典型方法。牛顿迭代法又称为牛顿-拉佛森,也可以称为牛顿切线法。在17世纪由牛顿提出,解决的是一种在实数域和复数域上近似求解方程根的方法,它是数值分析中重要的方法之一,不仅可以解决方程或方程组的求解问题,也可以用于微分方程和积分方程的求解。在计算机领域有着重要的地位,因为对于一个方程的求解来说,不存在或者很难找到求根公式进行求解,因此对于大多数的方程来说,得到一个精确的解是非常困难的,或者可能得不到方程的精确解,所以求方程的近似解是解决方程问题的关键。牛顿迭代法使用函数 $f(x)$ 的泰勒级数的前几项来寻找方程 $f(x)$ 的根,它的最大优点是在方程 $f(x)=0$ 的单根附近具有平方收敛,而且该法还可以用来求方程的重根、复根,此时线性收敛,但是可通过一些方法达到超线性收敛,这些方法在计算机算法领域得到广泛应用。

经过长时间的发展,众多学者通过不同的方法对原始的牛顿法提出了一些改进,例如王乐成[1]提出的简化牛顿法,算数平均牛顿法,中点牛顿迭代法,牛顿下山法,修正的算数平均牛顿法通过实验数据表明在提升函数的迭代速度上有一定的进步;吴江[2]主要思想是将其与牛顿迭代相结合,然后利用插值的方法来估计代牛顿步中的导数值;李惠敏[3]在原始的牛顿法基础上也总结出了五种不同的改进方法,分别为简化牛顿迭代法,推广的简化牛顿迭代法,牛顿下山法,弦位法,Newton-like法;陈玉骥[4]在牛顿迭代法的基础上,通过调整非线性方程对应曲线切线的斜率,从而保证在取任意初值时,迭代都是收敛的,有效改善了牛顿迭代法对初值的苛刻要求。在上述文献中提到的各种改进的牛顿法常用的改进思想是:用函数的均值来代替函数的一阶导数,因此可以得到更高的收敛阶数。本文与常用的改进思想不同的是利用中心差商结合弦截法对牛顿迭代法进行改进,利用两种方法相结合的方式,经过数值实验验证后,在函数的迭代速度上对比其他学者改进后的算法有一定的速度优势。

2. 牛顿法的原理

牛顿法是科研及工程技术中常用的数值方法。它不仅有良好的几何直观性,而且有二阶收敛性。牛顿法用迭代求解一个方程或方程组的解,原理如下:

对于一个函数 $f(x)$,它的泰勒级数展开式:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2}f''(x_0)(x-x_0)^2 + \cdots + \frac{I}{n!}f^n(x_0)(x-x_0)^n \quad (2.1)$$

当使用牛顿迭代法来求解方程时,使用泰勒级数的前两项来代替这个函数,即用 $\Phi(x)$ 代替 $f(x)$,其中

$$\Phi(x) = f(x_0) + f'(x_0)(x - x_0) \quad (2.2)$$

令 $\Phi(x) = 0$ ，则

$$x = x_0 - \frac{f(x)}{f'(x_0)} \quad (2.3)$$

所以牛顿法的迭代公式是

$$x_{n+1} = x_n - \frac{f(x_{n+1})}{f'(x_n)} \quad (2.4)$$

从几何意义上简单地来说就是一个不断地取切线的过程。对于形如 $f(x) = 0$ 的方程，先对其解估算一个近似值令其为 x_0 ，再把 x_0 代入原方程，一般来说这样是不可能直接得到方程的正确解，此时令 $f(x) = a$ ，然后计算方程在 x_0 处的斜率 $f'(x_0)$ ，得到 $f(x)$ 在 x_0 处的切线，这条切线与 x 轴的交点为 x_1 。方程 $f(x) = 0$ 精确解的意义是当取得的根是方程的正确根时，函数值为零。因此，可以得出 x_1 比 x_0 更加接近精确解，然后用迭代的方法，不断的更新 x ，一直迭代到可以取得一个无限接近方程根的近似精确解。但是在这个过程中需要计算斜率 $f'(x_0)$ ，也就是说要用迭代法求精确解必须要对某一个点进行求导计算，但是在计算很多方程时没有办法直接计算导数或者不能方便的得到导数，因此，这篇文章针对求导问题的不便提出了一些改进方案。

3. 牛顿法的改进

在牛顿法的基础上，本文是利用中心差商结合弦截法对牛顿迭代法进行改进，提出了五种方法。五种改进的方法均是在避免直接求解函数导数的同时对函数迭代速度进行一个提升，也有了更好的收敛阶数，使得在相同的函数求解过程中使用更少的迭代次数得到更高精度的解，为牛顿法在求解线性方程组中提供更优的数值结果，同时也让牛顿法在计算机算法领域得到更好的应用。一下是五种改进方法的具体内容：

方法 1:

在用牛顿法时，需要每次都对函数进行求导，也就要求能用牛顿法的函数必须处处可导，所以导致在用牛顿法时就有很大的局限性。本文针对于这种局限性，对牛顿法尝试一点改进。在得到初始点后仅对函数进行一次求导，得到第二个迭代点，在后面的求导过程用中心差商来代替当前点的导数，这样就可以避免在每个点都对函数进行求导，在一定程度上规避了函数的求导问题，具体迭代过程如下：

Step 1: 给定误差 δ ，选取一个初始值 x_0 ，对函数进行一次求导得到 $f'(x_0)$ ，利用牛顿法得到第二个点

$$x_1 = x_0 - \frac{f(x_1)}{f'(x_0)} \quad (3.1)$$

Step 2: 用差商代替第二个点的导数

$$f'(x_1) = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \quad (3.2)$$

Step 3: 得到第三个点

$$x_2 = \frac{x_0 + x_1}{2} - \frac{f(x_0)(x_0 - x_1)}{f(x_0) - f(x_1)} \quad (3.3)$$

Step 4: 从 $n=1$ 开始迭代

$$x_{n+2} = \frac{x_n + x_{n+1}}{2} - \frac{f(x_n)(x_n - x_{n+1})}{f(x_n) - f(x_{n+1})} \quad (3.4)$$

Step 5: 若 $|f(x)| < \delta$, 停止迭代, 否则继续 Step 4。

方法 2:

在得到初始点后仅对函数进行一次求导, 得到第二个迭代点, 从迭代的第二个点开始, 由差商代替当前点的导数值, 从而也可以避免无法求导的情况, 此方法的可行性由均差的性质可以得到, 与方法 1 的数值试验对比显示, 此方法在迭代速度上更快, 具体迭代过程如下:

Step 1: 给定误差 δ , 选取一个初始值 x_0 , 对函数进行一次求导得到 $f'(x_0)$, 利用牛顿法得到第二个点

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3.5)$$

Step 2: 用差商代替第二个点的导数

$$f'(x_1) = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \quad (3.6)$$

Step 3: 求得第三个点

$$x_2 = x_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)} \quad (3.7)$$

Step 4: 从 $n=1$ 开始迭代

$$x_{n+2} = x_{n+1} - \frac{f(x_{n+1})(x_n - x_{n+1})}{f(x_n) - f(x_{n+1})} \quad (3.8)$$

Step 5: 若 $|f(x)| < \delta$, 停止迭代, 否则继续 Step 4。

方法 3:

方法 2 中利用两个点建立迭代格式, 现在方法 3 上使用三个点进行迭代, 迭代速度要比使用两个点速度更快, 也就是使得算法逼近真实值的时间更短, 效率得到了一定的提升。具体迭代过程如下:

Step 1: 给定误差 δ , 选取一个初始值 x_0 , 对函数进行一次求导得到 $f'(x_0)$, 利用牛顿法得到第二个点

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3.9)$$

Step 2: 用差商代替第二个点的导数

$$f'(x_1) = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \quad (3.10)$$

Step 3: 求得第三个点

$$x_2 = x_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)} \quad (3.11)$$

Step 4: 用第一个点与第三个点的差商代替第二个点的导数, 求得第四个点的值

$$x_3 = x_1 - \frac{f(x_1)(x_0 - x_2)}{f(x_0) - f(x_2)} \quad (3.12)$$

Step 4: 从 $n=1$ 开始迭代

$$x_{n+3} = x_{n+1} - \frac{f(x_{n+1})(x_n - x_{n+2})}{f(x_n) - f(x_{n+2})} \quad (3.13)$$

Step 5: 若 $|f(x)| < \delta$, 停止迭代, 否则继续 Step 4。

方法 4:

在使用方法 3 时, 利用前后两个点的中心差商代替当前点的导数是不够准确的, 一般情况下, 我们认为当前点越靠近哪个点, 应该受哪个点的影响更大。在此基础上, 方法 4 对中心差商的计算进行加权修正, 在赋予权重后就能避免越过真实值, 使得在加速的基础上适当的提高精确度, 具体迭代过程如下:

Step 1: 给定误差 δ , 选取一个初始值 x_0 , 对函数进行一次求导得到 $f'(x_0)$, 利用牛顿法得到第二个点

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3.14)$$

Step 2: 用差商代替第二个点的导数

$$f'(x_1) = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \quad (3.15)$$

Step 3: 求得第三个点

$$x_2 = x_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)} \quad (3.16)$$

Step 4: 在方法 3 的基础上赋予权重, 利用三个点之间横坐标的距离来计算权重。

$$A = \frac{f(x_1) - f(x_2)}{x_1 - x_2}, \quad B = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \quad (3.17)$$

得到第四个点

$$x_3 = x_1 - \frac{f(x_1)}{2(A+B)} \quad (3.18)$$

Step 5: 从 $n=1$ 开始迭代

$$x_{n+3} = x_{n+1} - \frac{f(x_{n+1})}{2(A+B)} \quad (3.19)$$

其中:

$$A = \frac{f(x_{n+1}) - f(x_{n+2})}{x_{n+1} - x_{n+2}}, \quad B = \frac{f(x_n) - f(x_{n+1})}{x_n - x_{n+1}} \quad (3.20)$$

Step 6: 若 $|f(x)| < \delta$, 停止迭代, 否则继续 Step 4。

方法 5:

在用方法 2 对求导进行替换时, 当选取的初始值或者函数不同时, 有时会出现迭代速度不够快或精度不够, 在前面四种优化方法的基础上, 当把方法 2 和方法 3 结合在一起时, 不论是在迭代的速度上还

是精度上都有了很大程度上的提升，在每次进行迭代时，都需要用到前面三个点，看似比改进前的牛顿法复杂，实则迭代速度比原始牛顿法更快，精度更高，具体迭代过程如下：

Step 1: 给定误差 δ ，选取一个初始值 x_0 ，对函数进行一次求导得到 $f'(x_0)$ ，利用牛顿法得到第二个点

$$x_1 = x_0 - \frac{f(x_1)}{f'(x_0)} \quad (3.21)$$

Step 2: 用差商代替第二个点的导数

$$f'(x_1) = \frac{f(x_0) - f(x_1)}{x_0 - x_1} \quad (3.22)$$

Step 3: 得到伪点 X

$$X = x_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)} \quad (3.23)$$

Step 4: 第三个点为：

$$x_2 = \frac{x_1 + X}{2} \quad (3.24)$$

Step 4: 第四个点为：

$$x_3 = x_2 - \frac{f(x_2)(x_0 - X)}{f(x_0) - f(X)} \quad (3.25)$$

Step 5: 从 $n=1$ 开始迭代

$$x_{n+3} = x_{n+2} - \frac{f(x_{n+2})(x_n - X)}{f(x_n) - f(X)} \quad (3.26)$$

其中：

$$X = x_{n+1} - \frac{f(x_{n+1})(x_n - x_{n+1})}{f(x_n) - f(x_{n+1})} \quad (3.27)$$

Step 6: 若 $|f(x)| < \delta$ ，停止迭代，否则继续 Step 4。

4. 数值实验

为了验证经过优化后的算法的计算能效，选取了一下四种不同类型的函数，进行数值实验： $f(x) = \sin(x) + x$ ， $f(x) = (x-1)^3$ ， $f(x) = e^x - \cos(x)$ ， $f(x) = e^x + 3x^3$ 。下面将由表格的形式对比算法改进后的迭代速度，此数值实验设置的迭代终止条件为 $|f(x)| < \delta$ (x 为迭代点，取 $\delta = 10^{-9}$)，因考虑到部分函数中有三角函数，为保证算法的有效性提升，故迭代的初始值统一选取为 10。本数值实验均由 Python-3.80 版本得出数据结果。

如表 1 所示， $f(x) = (x-1)^3$ ， $f(x) = e^x - \cos(x)$ ， $f(x) = e^x + 3x^3$ 也由 python 输出的数据整理后得出表 2。

5. 结论

根据表 2 得出的数值试验可以看出，方法 5 在迭代速度和精度上均有更明显的优势，在选择相同的函数与初始值的前提下，方法 5 用了更少的迭代次数逼近函数值。与王乐成[1]提出的简化牛顿法、算数

Table 1. $f(x) = \sin(x) + x$ iteration times of statistical experiment**表 1.** $f(x) = \sin(x) + x$ 实验迭代次数统计

迭代次数	方法 1	方法 2	方法 3	方法 4	方法 5
1	10	10	10	10	10
2	-48.7589	-48.7589	-48.7589	-48.7589	-48.7589
3	-3.87767	1.041324	0.289188	0.289188	-0.43617
4	8.507069	-0.89407	-0.29364	-0.0302	-0.00028
5	-15.199	-0.00599	-3.1E-05	-0.00076	6.56E-13
6	108.5511	0.022957	2.27E-07	1.59E-05	6.56E-13
7	-22.6994	-2.5E-07	2.27E-07	1.51E-09	6.56E-13
8	68.98003	-2.5E-07	2.27E-07	1.51E-09	6.56E-13

Table 2. Summary of iteration times of four functions and five methods**表 2.** 四种函数五种方法迭代次数汇总

次数	$f(x) = \sin(x) + x$	$f(x) = (x-1)^3$	$f(x) = e^x - \cos(x)$	$f(x) = e^x + 3x^3$
方法 1	无穷大	18	15	16
方法 2	7	27	22	19
方法 3	6	25	21	19
方法 4	7	35	28	26
方法 5	5	15	12	13
牛顿法	8	33	25	32

平均牛顿法、中点牛顿迭代法、牛顿下山法、修正的算数平均牛顿法在收敛步数上相比，方法 5 收敛速度更快，迭代次数更少。对比肖光强[5]等对解非线性方程的收敛条件的改进，文中提出的改进方法具有更好的广泛性，上表在不同的函数中使用改进后的算法表明，在多数情况下都能迭代逼近函数值，使得能用牛顿迭代法的方程不仅仅局限于非线性方程。总的来说，在笔者改进后的牛顿迭代法中，不论是在收敛速度上还是在算法的广泛性上对比其他做法都有一定的优势。

基金项目

本工作受到国家自然科学基金(No. 11601151)和河北省青年拔尖人才支持计划项目支持。

参考文献

- [1] 王乐成, 赫亚兰, 韩新丽, 李小花, 卢凤兰, 马秋菊, 杨录峰. 对牛顿迭代法的改进[J]. 高师理科学刊, 2020, 40(03): 23-26.
- [2] 吴江. 求解非线性方程高阶迭代法研究[D]: [硕士学位论文]. 杭州: 杭州师范大学, 2019.
- [3] 李慧敏, 王晓燕. 对牛顿迭代法及改进的总结[J]. 科技信息, 2013(4): 275-277.
- [4] 陈玉骥. 牛顿迭代法的一种改进方法[J]. 佛山科学技术学院学报(自然科学版), 2012, 30(5): 1-3.
- [5] 肖光强, 方壮, 余显志. 对牛顿迭代法条件的一个改进[J]. 湖北民族学院学报(自然科学版), 2008, 26(4): 395-397.