

一点超前数值差分公式的提出、研究与实践*

张雨浓, 陈宇曦, 陈锦浩, 殷勇华
(中山大学信息科学与技术学院, 广东 广州 510006)

摘要: 根据数值微分理论, 若给定未知目标函数在指定区间上的离散采样点数据, 可使用数值差分公式求目标点处的一阶导数近似值。但对于靠近边界的目标点而言, 多点中心差分公式可能因单边数据点不足而无法使用。另外, 目标函数的一阶导数在目标点处可能发生加速变化, 而前(后)向差分公式只考虑了单边数据点, 可能无法适应该变化, 使导数值误差较大。实际上, 针对靠近右边界的目标点, 可将后向差分公式在形式上“前移”一点来计算一阶导数, 因此, 一点超前数值差分公式被提出与研究。计算机数值实验表明: 一点超前数值差分公式可使所求目标点一阶导数值具有较高的计算精度。

关键词: 未知目标函数; 一阶导数; 一点超前; 数值差分公式; 计算精度

中图分类号: O241.3 **文献标志码:** A **文章编号:** 0529-6579 (2012) 02-0001-05

Proposing, Investigation and Practice on One-Node-Ahead Numerical Differentiation Formulas

ZHANG Yunong, CHEN Yuxi, CHEN Jinhao, YIN Yonghua

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China)

Abstract: Based on the numerical differential theory, it is available to calculate the approximate first derivative of the target-node by using numerical differentiation formulas when the discrete sampling points of the unknown target function on specified interval are given. But for the target-nodes close to the boundary, it may be unable to use the center differentiation formulas involving multiple nodes because of the lack of sampling points on one side of the target-node. Besides, an accelerating change of the first derivative of the target-node may occur in some target functions. However, the use of forward/backward differentiation formulas simply takes the nodes on one side of the target-node into consideration, which probably makes the formulas difficult to adapt to such a change, and thus leads to less accuracy in estimating the first derivative of the target-node. Actually, for the target-nodes close to the right boundary, it is available to move the backward differentiation formulas one node ahead to calculate the first derivatives. Therefore, one-node-ahead numerical differentiation formulas are proposed and investigated. Experimental results verify and show that the first derivatives of the target-nodes with high computational precision can be obtained by using the one-node-ahead numerical differentiation formulas.

Key words: unknown target function; first derivatives; one node ahead; numerical differentiation formulas; computational precision

数值微分是数值计算应用中的一个重要概念, 是求解常微分方程和偏微分方程的重要工具^[1~4]。

* 收稿日期: 2012-01-09

基金项目: 国家自然科学基金资助项目 (61075121); 教育部高等学校博士学科点专项科研基金博导类资助项目 (20100171110045)

作者简介: 张雨浓 (1973年生), 男, 教授、博士生导师; E-mail: zhyong@mail.sysu.edu.cn

它根据给定的离散采样点数据,使用多项式插值法、有限差分法、正则化方法或待定系数法等方法逼近目标函数在目标点处的一阶导数值^[5-6]。对于有着广泛应用背景和实用性的有限差分类型的方法,人们对其研究兴趣与日俱增^[5,7-10]。一般来说,若给定未知目标函数在指定区间上的离散采样点数据,可使用多点中心差分公式求远离边界的目标点的一阶导数值,以及使用多点前向差分公式和多点后向差分公式求区间边界处目标点的一阶导数值。然而,多点中心差分公式的使用需要保证目标点两端有相同数量的计算数据点,但对于靠近边界的目标点来说,它靠近边界的一边会出现数据点不足的情况,因此可能无法使用多点中心差分公式。另外,未知目标函数的一阶导数在这类目标点处可能会发生加速变化,而前(后)向差分公式只使用目标点远离边界一端的计算数据点,因此可能无法适应该变化,使得用该公式得到的一阶导数近似值误差较大。考虑到这类目标点的特殊性,需研究出一种综合三种传统有限差分公式特性的新差分公式对其进行一阶导数的近似计算。本文将针对靠近右边界目标点一阶导数的计算,将后向差分公式在形式上“前移”一点,即可得到适应该类目标点的一点超前数值差分公式。该公式参考了前(后)向差分公式在数据点利用上的不平衡性,解决了目标点单边数据点不足的问题,同时它又利用所“前移”的一点,适当调节了目标点的一阶导数估算,适应了函数在该处可能发生的一阶导数加速变化。

1 一点超前数值差分公式

基于多项式插值理论,利用采样数据点以及相应的函数值,可以通过构造插值多项式 $P(x)$ 来近似求得一个未知目标函数 $f(x)$ 的逼近表达式,即 $f(x) \approx P(x)$; 进而,对 $P(x)$ 求一阶导数便可近似得到未知目标函数 $f(x)$ 的一阶数值微分公式,即 $f'(x) \approx P'(x)$ ^[11-12]。另外,在实际应用中,数字离散系统往往是等间距地采集数据^[13],也即 $x_{i+1} - x_i = h$ ($i = 0, 1, 2, \dots, n$),其中 $h > 0$ 为采样间隔常数,为此,本文将主要研究和探讨等间距一点超前数值差分公式。

定义 1^[5,11,14] 设未知目标函数 $f(x)$ 在离散采样数据点 x_i 处的函数值为 $f(x_i)$, $i = 0, 1, 2, \dots, n$ 。若存在 n 次多项式 $P_n(x) = a_0 + a_1x + \dots + a_nx^n$ 满足条件 $P_n(x_i) = f(x_i)$, 则称 $P_n(x)$ 为未知目标函数 $f(x)$ 的 n 次插值多项式。

引理 1^[5,11,14] 如果 $x_0, x_1, x_2, \dots, x_n$ 是采样区间 $[a, b]$ 上 $(n+1)$ 个互不相同的数据点,且未知目标函数 $f(x)$ 的 $(n+1)$ 阶导数连续(记 $f \in C^{(n+1)}[a, b]$), 则对任一 $x \in [a, b]$, 存在 $c \in (a, b)$ 使得 $f(x) = P_n(x) + R_{n+1}(x)$, 其中 $P_n(x)$ 为未知目标函数 $f(x)$ 的插值多项式, $R_{n+1}(x)$ 为余项且 $R_{n+1}(x) := f^{(n+1)}(c)(x-x_0)(x-x_1)\dots(x-x_n)/(n+1)!$ 。

根据定义 1 和引理 1, 可以构造出一个 n 次插值多项式 $P_n(x)$ 来逼近未知目标函数 $f(x)$ 。再对多项式 $P_n(x)$ 求一阶导数, 便可以近似求得未知目标函数 $f(x)$ 的一阶数值微分公式, 即 $f'(x) = P'_n(x) + R'_{n+1}(x)$ 其中 $R'_{n+1}(x)$ 为相应余项的一阶导数项。当各数据点为等间距分布时, 有如下 $f(x)$ 的一阶数值微分公式^[5-6]:

$$f'(x_i) = \frac{1}{h} \sum_{\substack{j=0 \\ j \neq i}}^n \frac{(-1)^{i-j} \frac{i!(n-i)!}{j!(n-j)!} f(x_j) + f(x_i)}{(i-j)} + \frac{(-1)^{n-i} h^n f^{(n+1)}(c)}{(n+1)C_n^i} \quad (1)$$

其中 $C_n^i := n! / ((n-i)!i!)$ 且 $c \in (a, b)$ 。值得指出的是, 对于 $f(x)$ 的一阶数值微分公式 (1), 当 $i = n-1$ 时, 忽略余项 [即公式 (1) 最右端项], 可近似得到未知目标函数 $f(x)$ 一阶导数的等间距一点超前数值差分公式:

$$f'(x_{n-1}) \approx \frac{1}{h} \sum_{\substack{j=0 \\ j \neq n-1}}^n \frac{(-1)^{n-j-1} C_n^j f(x_j) / n + f(x_{n-1})}{n-j-1} \quad (2)$$

对于公式 (2), 分别令 $n=1, 2, \dots, 15$, 经过计算, 即可推导出二至十六个数据点的等间距一点超前数值差分公式。其中, 表 1 列出了二至八个数据点的等间距一点超前数值差分公式, 而九至十六个数据点的等间距一点超前数值差分公式在附录中列出。

表 1 二至八个数据点的等间距一点超前数值差分公式

Table 1 Equidistant one-node-ahead numerical differentiation formulas with 2 to 8 data points

点数	等间距一点超前数值差分公式	误差项
二	$f'(x_0) \approx (f(x_1) - f(x_0))/h$	$O(h)$
三	$f'(x_1) \approx (f(x_2) + 0f(x_1) - f(x_0))/(2h)$	$O(h^2)$
四	$f'(x_2) \approx (2f(x_3) + 3f(x_2) - 6f(x_1) + f(x_0))/(6h)$	$O(h^3)$
五	$f'(x_3) \approx (3f(x_4) + 10f(x_3) - 18f(x_2) + 6f(x_1) - f(x_0))/(12h)$	$O(h^4)$
六	$f'(x_4) \approx (12f(x_5) + 65f(x_4) - 120f(x_3) + 60f(x_2) - 20f(x_1) + 3f(x_0))/(60h)$	$O(h^5)$
七	$f'(x_5) \approx (10f(x_6) + 77f(x_5) - 150f(x_4) + 100f(x_3) - 50f(x_2) + 15f(x_1) - 2f(x_0))/(60h)$	$O(h^6)$
八	$f'(x_6) \approx (60f(x_7) + 609f(x_6) - 1260f(x_5) + 1050f(x_4) - 700f(x_3) + 315f(x_2) - 84f(x_1) + 10f(x_0))/(420h)$	$O(h^7)$

观察表 1 可以发现，二数据点等间距一点超前数值差分公式即为传统的二数据点前向差分公式，而三数据点等间距一点超前数值差分公式即为传统的二数据点中心差分公式。

2 计算机验证与实践

由于一点超前数值差分公式在数据点的利用上具有了“前瞻性”，有效调节目标点的导数值计算，因此能很好地解决因目标函数的一阶导数加速变化而影响微分计算精度的问题。为验证一点超前数值差分公式的这种特性，这里选取两个目标函数 $f_1(x) = \sin(x)$ 和 $f_2(x) = \arctan(x)$ 来进行计算机数值实验，用一点超前数值差分公式和后向差分公式分别计算这两个目标函数在一阶导数变化最快的地方的一阶导数近似值，再通过比较两个公式的求导效果研究一点超前数值差分公式的性能优势。

通过分析可以发现，目标函数 $f_1(x) = \sin(x)$ 的一阶导数在点 $x_1 = \pi/2 \approx 1.570796327$ 处变化最快，实取 $x_1 = 1.571$ ；而目标函数 $f_2(x) = \arctan(x)$ 的一阶导数则在点 $x_2 = 1/\sqrt{3} \approx 0.577350269$ 处变化最快，实取 $x_2 = 0.577$ 。现取这两点分别作为这两个目标函数的测试节点。

如前所述，二数据点等间距一点超前数值差分公式即为二数据点的前向差分公式，而三数据点等间距一点超前数值差分公式即为二数据点的中心差分公式，而前人已对它们进行了有效性研究^[5,11]，因此本文不再对它们展开更深入的探讨。这里将使用四至八个数据点的等间距一点超前数值差分公式进行实验。

一般来说，数值微分计算的结果只有计算机表示能力的一半精度^[11]，因此，我们在实验中使用

优化步长。另外，考虑到在实际工程应用中所得到的样本数据精度有限，现对实验中所使用的目标函数样本数据作出精度限制，统一精确到小数点后 9 位，并以四数据点等间距一点超前数值差分公式为例求其优化步长。

用 e_i ($i=0, 1, 2, 3$) 表示表 1 中四数据点等间距一点超前数值差分公式中对应样本函数 $f(x_i)$ ($i=0, 1, 2, 3$) 的舍入误差，再根据式 (1) 中微分公式的余项，可得到 $f'(x_i)$ 的误差项表达式^[11]：

$$E(f, h) = \frac{2e_3 + 3e_2 - 6e_1 + e_0}{6h} - \frac{h^3 f^{(4)}(c)}{12}$$

设每个误差 e_i 的量级为 ε ，同时误差是积累的，且 $|f^{(4)}(x)| \leq M$ ，则可得到如下误差界：

$$|E(f, h)| \leq \frac{2\varepsilon}{h} + \frac{Mh^3}{12}$$

通过求函数 $g(h) = 2\varepsilon/h + Mh^3/12$ 的最小值可得到优化步长，从而使误差界最小。设 $g'(h) = 0$ ，可得 $-2\varepsilon/h^2 + Mh^2/4 = 0$ ，即 $h^4 = 8\varepsilon/M$ ，这样可得到优化步长：

$$h = \left(\frac{8\varepsilon}{M}\right)^{1/4} \quad (3)$$

对于目标函数 $f_1(x) = \sin(x)$ ，使用边界 $|f_1^{(4)}(x)| \leq |\sin(x)| \leq 1 = M$ 和值 $\varepsilon = 0.5 \times 10^{-9}$ ，代入式 (3) 计算可得优化步长 $h = 0.007952707$ ，可以看到 $h = 0.01$ 时最接近优化值。用同样的方法可得到关于这两个目标函数其它多点等间距一点超前数值差分公式的优化步长。为了方便比较，这里同时算出后向差分公式在同样情况下的优化步长，并根据两个差分公式的优化步长差异，统一选择用于计算机实验的优化步长，如表 2 所示。

表 2 四至八个数据点的等间距数值差分公式的优化步长

Table 2 Optimal step sizes for one-node-ahead numerical differentiation formulas with 4 to 8 data points

点数	等间距后向差分公式优化步长		等间距一点超前差分公式优化步长		实验统一选用优化步长	
	$\sin(x)$	$\arctan(x)$	$\sin(x)$	$\arctan(x)$	$\sin(x)$	$\arctan(x)$
四	0.008 164 966	0.005 554 675	0.007 952 707	0.005 410 274	0.01	0.005
五	0.023 162 304	0.012 267 032	0.023 972 232	0.012 695 980	0.01	0.01
六	0.046 599 722	0.021 613 173	0.049 093 200	0.022 769 660	0.05	0.01
七	0.077 088 290	0.030 116 053	0.081 344 521	0.031 778 834	0.10	0.05
八	0.112 846 332	0.039 551 525	0.118 519 606	0.041 539 952	0.10	0.05

综合以上分析,我们将使用表 2 中实验统一选用的优化步长,分别利用后向差分公式和一点超前数值差分公式,求目标函数 $f_1(x) = \sin(x)$ 和

$f_2(x) = \arctan(x)$ 分别在 $x_1 = 1.571$ 和 $x_2 = 0.577$ 处一阶导数近似值,并以近似值与真值之差的绝对值作为实验计算误差进行比较分析,如表 3 所示。

表 3 目标函数 $f_1(x) = \sin(x)$ 和 $f_2(x) = \arctan(x)$ 一阶导数的求解误差Table 3 Solution errors of first derivatives of target functions $f_1(x) = \sin(x)$ and $f_2(x) = \arctan(x)$

点数	求 $f_1(x) = \sin(x)$ 一阶导数的计算误差		求 $f_2(x) = \arctan(x)$ 一阶导数的计算误差	
	后向差分公式	一点超前数值差分公式	后向差分公式	一点超前数值差分公式
四	$1.934\ 629\ 641 \times 10^{-7}$	$1.065\ 370\ 237 \times 10^{-7}$	$1.601\ 753\ 001 \times 10^{-7}$	$9.350\ 864\ 283 \times 10^{-8}$
五	$1.065\ 370\ 015 \times 10^{-7}$	$3.153\ 701\ 011 \times 10^{-8}$	$6.017\ 535\ 115 \times 10^{-8}$	$3.517\ 530\ 467 \times 10^{-8}$
六	$4.187\ 038\ 870 \times 10^{-8}$	$1.412\ 963\ 962 \times 10^{-8}$	$5.982\ 480\ 089 \times 10^{-8}$	$4.017\ 532\ 584 \times 10^{-8}$
七	$1.120\ 368\ 634 \times 10^{-8}$	$1.046\ 297\ 383 \times 10^{-8}$	$7.728\ 246\ 693 \times 10^{-7}$	$1.271\ 753\ 198 \times 10^{-7}$
八	$5.593\ 915\ 098 \times 10^{-8}$	$7.367\ 727\ 189 \times 10^{-9}$	$7.574\ 678\ 285 \times 10^{-8}$	$1.396\ 108\ 451 \times 10^{-9}$

分析表 3 可以发现,使用一点超前数值差分公式计算目标函数导数时计算误差均小于使用后向差分公式计算目标函数导数时计算误差。这说明了,一点超前数值差分公式的求导性能优于后向差分公式,而且,当目标函数的一阶导数在目标点处加速变化时,一点超前数值差分公式能很好地适应该变化,使计算得到的一阶导数值具有更高的精度。

3 结 论

一阶数值微分在科学研究中有着重要的应用。而作为常用的数值微分方法,有限差分法包含了传统的前向差分公式、后向差分公式以及中间差分公式,它们在计算目标函数在指定区间边界和中部目标点的一阶导数时体现出高度的方便性和有效性。然而对于靠近区间边界的部分目标点,因其两端的计算数据点个数具有明显的不平衡性,点数较多的多点中心差分公式可能无法使用。而且目标函数的

一阶导数在目标点处可能加速变化,对于只考虑单边数据点的前(后)向差分公式来说,它们可能无法调节由该变化带来的误差影响。而一点超前数值差分公式正是针对上述方法的缺点而研究出来的新的有限差分公式,它具有对数据点利用的不平衡性和对目标函数一阶导数加速变化的适应性,使实际的求导性能优于后向差分公式,因此成为一种更为方便和高效的差分算法。计算机数值实验进一步验证表明:一点超前数值差分公式可以取得较好的一阶导数估计精度。

附录 九至十六个数据点的等间距一点超前数值差分公式。

前文已列出了二至八个数据点的等间距一点超前数值差分公式,考虑到方便其他研究者和应用者对公式的深入研究和使用的,此处我们再列出九至十六个数据点的等间距一点超前数值差分公式,如表 4 所示。

表 4 九至十六个数据点的等间距一点超前数值差分公式
Table 4 Equidistant one-node-ahead numerical differentiation formulas with 9 to 16 data points

点数	等间距一点超前数值差分公式
九	$f'(x_7) \approx (105f(x_8) + 1338f(x_7) - 2940f(x_6) + 2940f(x_5) - 2450f(x_4) + 1470f(x_3) - 588f(x_2) + 140f(x_1) - 15f(x_0)) / (840h)$
十	$f'(x_8) \approx (280f(x_9) + 4329f(x_8) - 10080f(x_7) + 11760f(x_6) - 11760f(x_5) + 8820f(x_4) - 4704f(x_3) + 1680f(x_2) - 360f(x_1) + 35f(x_0)) / (2520h)$
十一	$f'(x_9) \approx (252f(x_{10}) + 4609f(x_9) - 11340f(x_8) + 15120f(x_7) - 17640f(x_6) + 15876f(x_5) - 10584f(x_4) + 5040f(x_3) - 1620f(x_2) + 315f(x_1) - 28f(x_0)) / (2520h)$
十二	$f'(x_{10}) \approx (2520f(x_{11}) + 53471f(x_{10}) - 138600f(x_9) + 207900f(x_8) - 277200f(x_7) + 291060f(x_6) - 232848f(x_5) + 138600f(x_4) - 59400f(x_3) + 17325f(x_2) - 3080f(x_1) + 252f(x_0)) / (27720h)$
十三	$f'(x_{11}) \approx (2310f(x_{12}) + 55991f(x_{11}) - 152460f(x_{10}) + 254100f(x_9) - 381150f(x_8) + 457380f(x_7) - 426888f(x_6) + 304920f(x_5) - 163350f(x_4) + 63525f(x_3) - 16940f(x_2) + 2772f(x_1) - 210f(x_0)) / (27720h)$
十四	$f'(x_{12}) \approx (27720f(x_{13}) + 757913f(x_{12}) - 2162160f(x_{11}) + 3963960f(x_{10}) - 6606600f(x_9) + 8918910f(x_8) - 9513504f(x_7) + 7927920f(x_6) - 5096520f(x_5) + 2477475f(x_4) - 880880f(x_3) + 216216f(x_2) - 32760f(x_1) + 2310f(x_0)) / (360360h)$
十五	$f'(x_{13}) \approx (25740f(x_{14}) + 785633f(x_{13}) - 2342340f(x_{12}) + 4684680f(x_{11}) - 8588580f(x_{10}) + 12882870f(x_9) - 15459444f(x_8) + 14723280f(x_7) - 11042460f(x_6) + 6441435f(x_5) - 2862860f(x_4) + 936936f(x_3) - 212940f(x_2) + 30030f(x_1) - 1980f(x_0)) / (360360h)$
十六	$f'(x_{14}) \approx (24024f(x_{15}) + 811373f(x_{14}) - 2522520f(x_{13}) + 5465460f(x_{12}) - 10930920f(x_{11}) + 18036018f(x_{10}) - 24048024f(x_9) + 25765740f(x_8) - 22084920f(x_7) + 15030015f(x_6) - 8016008f(x_5) + 3279276f(x_4) - 993720f(x_3) + 210210f(x_2) - 27720f(x_1) + 1716f(x_0)) / (360360h)$

参考文献：

[1] ZHANG Y N, JIANG D C, WANG J. A recurrent neural network for solving Sylvester equation with time-varying coefficients [J]. IEEE Transactions on Neural Networks, 2002, 13(5): 1053 - 1063.

[2] ZHANG Y N, GE S S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion [J]. IEEE Transactions on Neural Networks, 2005, 16(6): 1477 - 1490.

[3] ZHANG Y N, CHEN K. Comparison on Zhang neural network and gradient neural network for time-varying linear matrix equation $AXB = C$ Solving [C] // Proceedings of International Conference on Industrial Technology Chengdu, China, 2008.

[4] MA W M, ZHANG Y N, WANG J. MATLAB simulink modeling and simulation of Zhang neural networks for on-line Time-varying Sylvester Equation Solving [C] // Proceedings of International Joint Conference on Neural Networks. Hong Kong, China, 2008: 286 - 290.

[5] 张雨浓, 郭东生, 徐思洪, 等. 未知目标函数之一阶数值微分公式验证与实践 [J]. 甘肃科学学报, 2009, 21(1): 13 - 18.

[6] LI J P. General explicit difference formulas for numerical

differentiation [J]. Journal of Computational and Applied Mathematics, 2005, 183: 29 - 52.

[7] 郭莹莹, 余向阳. 光学 Bloch 方程的数值解法 [J]. 中山大学学报: 自然科学版, 2005, 44(5): 108 - 110.

[8] 孙建平, 赵亚红. 二阶差分系统正解的存在性 [J]. 甘肃科学学报, 2001, 14(3): 7 - 10.

[9] KHAN I R, OHBA R, HOZUMI N. Mathematical proof of closed form expressions for finite difference approximations based on Taylor series [J]. Journal of Computational and Applied Mathematics, 2003, 150: 303 - 309.

[10] 田瑞, 李明, 滑斌. 有限差分法求解方同轴线的特性 [J]. 甘肃科学学报, 2007, 19(1): 1112113.

[11] JOHN H M, KURTIS D F. 数值方法 (MATLAB 版) [M]. 周璐, 陈渝, 钱方, 等译. 4 版. 北京: 电子工业出版社, 2009.

[12] ZHANG Y N, MA W M, YI C F. The Link between Newton Iteration for Matrix Inversion and Zhang Neural Network (ZNN) [C]. Proceedings of IEEE Conference on Industrial Technology, 2008: 1 - 6.

[13] FRANKLIN G F, POWELL J D, NAEINI A E. 自动控制原理与设计 [M]. 李中华, 张雨浓, 译. 5 版. 北京: 人民邮电出版社, 2007.

[14] 李建良, 蒋勇, 汪光先. 计算机数值方法 [M]. 南京: 东南大学出版社, 2000.