Journal of Computational and Applied Mathematics 239 (2013) 406–414

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Presentation, error analysis and numerical experiments on a group of 1-step-ahead numerical differentiation formulas

Yunong Zhang[*], Yao Chou, Jinhao Chen, Zhijun Zhang, Lin Xiao

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

A R T I C L E I N F O

Article history:
Received 20 May 2012

Keywords:
The first-order derivative
Computational precision
Numerical differentiation
1-step-ahead
Error analysis
Optimal step length

A B S T R A C T

In order to achieve higher computational precision in approximating the first-order derivative of the target point, the 1-step-ahead numerical differentiation formulas are presented. These formulas greatly remedy some intrinsic weaknesses of the backward numerical differentiation formulas, and overcome the limitation of the central numerical differentiation formulas. In addition, a group of formulas are proposed to obtain the optimal step length. Moreover, the error analysis of the 1-step-ahead numerical differentiation formulas and the backward numerical differentiation formulas is further investigated. Numerical studies show that the proposed optimal step-length formulas are effective, and the performance of the 1-step-ahead numerical differentiation formulas is much better than that of the backward numerical differentiation formulas in the first-order derivative approximation.

1. Introduction

Numerical differentiation, which is usually used to solve ordinary differential equations (ODEs) and partial differential equations (PDEs) [1–5], has been widely applied in numerical analysis [5,6] and engineering applications [7–9]. With discrete sampling points of the given target function, different methods can be used to obtain the approximation of the first-order derivative at the target point, such as the polynomial interpolation method, the finite difference method, the regularization method and the method of undetermined coefficients [10]. However, considering the following two situations: (1) that the backward differentiation formulas may not adapt to the fast variational rate of the first-order derivative of target point, and (2) that the central differentiation formulas cannot approximate the first-order derivative of the target points without enough number of data points on either side, a new kind of 1-step-ahead numerical differentiation formulas is needed.

In addition, it is an important task in the numerical differentiation to analyze the total errors of such 1-step-ahead numerical differentiation formulas. Here, the total errors contain two parts, i.e., the round-off errors and the truncation errors. Besides, from the previous work [5], we know that the step length brings about a great influence on the total errors. Thus, in this paper, we will mainly investigate the relationship between the step length and the total error, and, finally, analyze and find out the optimal step length to minimize the total error. Lots of numerical experimental results indicate that the optimal step-length formulas are effective.

The remainders of this paper are organized into four sections and two appendices. In Section 2, the 1-step-ahead numerical differentiation formulas using two to seven data points are presented. In Section 3, the total errors are analyzed and the related numerical experimental results are illustrated. The comparison of the errors between the presented 1-step-ahead numerical differentiation formulas and the backward numerical differentiation formulas is further discussed in Section 4, and conclusions are in Section 5. In Appendix A, we present a novel method as the proof of the related formula,

[*] Corresponding author.
E-mail addresses: ynzhang@ieee.org, zhynong@mail.sysu.edu.cn, jallonzyn@sina.com (Y. Zhang).

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.09.011

which is different from Refs. [10–13]. In Appendix B, the 1-step-ahead numerical differentiation formulas using eight to sixteen data points are presented for related researchers and employers.

To the best of our knowledge, no one else has been published such papers dealing with the 1-step-ahead numerical differentiation formulas in the literature at present stage. The main contributions in this paper are as follows.

(1) This paper presents and further investigates a new kind of the 1-step-ahead numerical differentiation formulas to approximate the first-order derivative of the target point, and thus overcome the limitations of the backward differentiation formulas and the central differentiation formulas.
(2) This paper proposes a group of formulas to obtain the optimal step lengths of the 1-step-ahead numerical differentiation formulas. Besides, the complete numerical experiments with different step lengths are provided to substantiate the efficacy of these optimal step-length formulas.
(3) The total errors of such 1-step-ahead numerical differentiation formulas and the backward numerical differentiation formulas are further compared to substantiate the superiority of the presented 1-step-ahead numerical differentiation formulas.

2. Formula derivation with alternative proof

Based on polynomial interpolation theory, a interpolation polynomial $\phi(x)$ can be constructed to approximate the unknown target function $f(x), f(x) \approx \phi(x)$. Therefore, we can obtain the first-order derivative of unknown target function $f(x)$ by computing the first-order derivative of $\phi(x)$, that is, $f'(x) \approx \phi'(x)$ [5]. Besides, in practical applications, sampling data points are collected with equal space in the digital discrete system, i.e., $x_{i+1} - x_i = h$, $(i = 0, 1, 2, \ldots, n)$, where $h > 0$ is the sampling interval constant. Hence, we will present and investigate the equally spaced 1-step-ahead differentiation formulas in the following. Firstly, the following definition and lemma are presented for further discussion.

Definition 1. Assuming that the value of unknown target function $f(x)$ at the discretely-sampling data point x_i is $f(x_i)$, $(i = 0, 1, 2, \ldots, n)$. If there is a n-degree polynomial $\phi_n(x_i) = \alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n$ satisfying $\phi_n(x_i) = f(x_i)$, then we name $\phi_n(x)$ as the n-degree interpolation polynomial of the unknown target function $f(x)$.

Lemma 1. *If $x_0, x_1, x_2, \ldots, x_n$ are $(n+1)$ mutually different data points in the sampling interval $[a, b]$, and the $(n+1)$th-order derivative of unknown target function $f(x)$ is continuous (marked as $f \in C^{(n+1)}[a, b]$), then for any $x \in [a, b]$, there exists a constant $c \in (a, b)$ that satisfies*

$$f(x) = \phi_n(x) + \sigma_{n+1}(x), \tag{1}$$

where $\phi_n(x)$ is the interpolation polynomial of the unknown target function $f(x)$, and the remainder term $\sigma_{n+1}(x)$ can be expressed as

$$\sigma_{n+1}(x) = \frac{f^{n+1}(c) \prod_{i=0}^{n} (x - x_i)}{(n+1)!}. \tag{2}$$

According to Definition 1 and Lemma 1, we can construct an n-order interpolation polynomial $\phi_n(x)$ to approximate the unknown target function $f(x)$. Therefore, we have the following theorem.

Theorem 1. *The approximation of the first-order derivative of unknown target function $f(x)$ can be determined by calculating that of the interpolation polynomial $\phi_n(x)$, that is,*

$$f'(x) = \phi_n'(x) + \sigma_{n+1}'(x), \tag{3}$$

where $\sigma_{n+1}'(x)$ is the first-order derivative of the remainder term $\sigma_{n+1}(x)$. Considering that the data points are sampled with equal space h, the first-order derivative of $f(x)$ can be formulated as

$$f'(x_i) = \frac{1}{h} \sum_{j=0, j\neq i}^{n} \frac{(-1)^{i-j} C_n^j / C_n^i f(x_j) + f(x_i)}{i-j} + \frac{(-1)^{n-i} h^n f^{n+1}(c)}{(n+1) C_n^i}, \tag{4}$$

where $C_n^i = n!/((n-i)!i!)$, $C_n^j = n!/((n-j)!j!)$, and $c \in [a, b]$.

Proof. See Appendix A. □

It is worth pointing out that, by letting $i = n-1$ and ignoring the first-order derivative of the remainder item in the numerical differentiation formula (4), we can approximate the first-order derivative of the unknown target function $f(x)$ as follows:

$$f'(x_{n-1}) \approx \frac{1}{h} \sum_{j=0, j\neq n-1}^{n} \frac{(-1)^{n-j-1} C_n^j f(x_j)/n + f(x_{n-1})}{n-j-1}. \tag{5}$$

Table 1
The 1-step-ahead differentiation formulas using two to seven data points.

Data points	1-step-ahead formulas	Truncation errors
2	$f'(x_0) \approx (f(x_1) - f(x_0))/h$	$o(h)$
3	$f'(x_1) \approx (f(x_2) - f(x_0))/(2h)$	$o(h^2)$
4	$f'(x_2) \approx (2f(x_3) + 3f(x_2) - 6f(x_1) + f(x_0))/(6h)$	$o(h^3)$
5	$f'(x_3) \approx (3f(x_4) + 10f(x_3) - 18f(x_2) + 6f(x_1) - f(x_0))/(12h)$	$o(h^4)$
6	$f'(x_4) \approx (12f(x_5) + 65f(x_4) - 120f(x_3) + 60f(x_2) - 20f(x_1) + 3f(x_0))/(60h)$	$o(h^5)$
7	$f'(x_5) \approx (10f(x_6)+77f(x_5)-150f(x_4)+100f(x_3)-50f(x_2)+15f(x_1)-2f(x_0))/(60h)$	$o(h^6)$

For formula (5), let $n = 1, 2, \ldots, 15$, and then we can derive the equally spaced 1-step-ahead numerical difference formulas using two to sixteen data points, respectively. The equally spaced 1-step-ahead numerical difference formulas using two to seven data points are shown in Table 1, while those using eight to sixteen data points are shown in Appendix B.

As we can see from Table 1, the 1-step-ahead numerical differentiation formula using two data points is in the same form with the forward differentiation formula using two data points. Moreover, the one using three data points is the same with the central differentiation formula using two data points. Hence, we will mainly discuss the 1-step-ahead numerical differentiation formulas using four to seven data points.

3. Total error analysis and numerical verification

In this section, a special kind of formulas will be proposed to determine the optimal step length of the 1-step-ahead numerical differentiation formulas, and then related simulations will be presented to substantiate the correctness of these optimal step-length formulas.

3.1. Optimal step-length formulas

As we mentioned above, the step length brings about a great influence on the total errors. Thus, in practical engineering applications, the optimal step length is generally expected for obtaining a relatively higher accuracy. In this subsection, we propose a special kind of formulas to determine the optimal step length of the 1-step-ahead numerical differentiation formulas.

For better understanding, we firstly give the derivation of the optimal step-length formulas for the 1-step-ahead numerical differentiation formulas using four and five data points in detail. The following derivation is about the optimal step-length formula of the 1-step-ahead numerical differentiation formulas using four data points.

In digital computers, there always exist round-off errors in the numerical computations [5], which can be simplified as the following equations:

$$\begin{aligned} f(x_3) &= y_3 + \varepsilon_3, \\ f(x_2) &= y_2 + \varepsilon_2, \\ f(x_1) &= y_1 + \varepsilon_1, \\ f(x_0) &= y_0 + \varepsilon_0, \end{aligned}$$

where $f(x_3), f(x_2), f(x_1)$ and $f(x_0)$ are approximated by numerical values y_3, y_2, y_1 and y_0 respectively, while ε_3, ε_2, ε_1 and ε_0 are the corresponding round-off errors.

According to the 1-step-ahead numerical differentiation formula using four data points, we can obtain

$$f'(x_2) \approx \frac{2y_3 + 3y_2 - 6y_1 + y_0}{6h}.$$

Then the error analysis can be explained by the following equation:

$$f'(x_2) = \frac{2y_3 + 3y_2 - 6y_1 + y_0}{6h} + E(f, h),$$

in which

$$E(f, h) = \frac{2\varepsilon_3 + 3\varepsilon_2 - 6\varepsilon_1 + \varepsilon_0}{6h} - \frac{h^3 f^{(4)}(c)}{12},$$

where the total error term $E(f, h)$ contains two parts, i.e., a part due to round-off errors, and a part due to truncation errors.

Assuming that $|\varepsilon_m| \le \varepsilon, (m = 0, 1, 2, 3,$ and ε denotes a positive constant), which is accumulative, and that $|f^{(4)}(c)| \le M$ with $f \in C^{(n+1)}[a, b]$ and M denoting a positive constant, then

$$|E(f, h)| \le \frac{2\varepsilon}{h} + \frac{Mh^3}{12}. \tag{6}$$

Table 2
The optimal step length for the 1-step-ahead numerical differentiation formulas using four to seven data points.

Data points	1-step-ahead formula (h)
4	$(8\varepsilon/M)^{1/4}$
5	$(95\varepsilon/6M)^{1/5}$
6	$(28\varepsilon/M)^{1/6}$
7	$(707\varepsilon/15M)^{1/7}$

Thus, the value of h that minimizes the right-hand side of formula (6) is

$$h = \left(\frac{8\varepsilon}{M}\right)^{1/4}. \tag{7}$$

In addition, the following derivation is given for the optimal step-length formula of the 1-step-ahead numerical differentiation formulas using five data points.

Similarly, $f(x_4), f(x_3), f(x_2), f(x_1)$, and $f(x_0)$ are respectively approximated by numerical values y_4, y_3, y_2, y_1, and y_0 with the corresponding round-off errors ε_4, ε_3, ε_2, ε_1 and ε_0. That is

$$f(x_4) = y_4 + \varepsilon_4,$$
$$f(x_3) = y_3 + \varepsilon_3,$$
$$f(x_2) = y_2 + \varepsilon_2,$$
$$f(x_1) = y_1 + \varepsilon_1,$$
$$f(x_0) = y_0 + \varepsilon_0.$$

Based on the 1-step-ahead numerical differentiation formula using five data points, we can derive

$$f'(x_3) \approx \frac{3y_4 + 10y_3 - 18y_2 + 6y_1 - y_0}{12h}.$$

Then the error analysis can be explained by the following equation

$$f'(x_3) = \frac{3y_4 + 10y_3 - 18y_2 + 6y_1 - y_0}{12h} + E(f, h),$$

of which

$$E(f, h) = \frac{3\varepsilon_4 + 10\varepsilon_3 - 18\varepsilon_2 + 6\varepsilon_1 - \varepsilon_0}{12h} - \frac{h^4 f^{(5)}(c)}{20}.$$

Assuming that $|\varepsilon_i| \le \varepsilon$ $(i = 0, 1, 2, 3, 4)$, which is accumulative, and $|f^{(5)}(c)| \le M$ with $f \in C^{(n+1)}[a, b]$, then

$$|E(f, h)| \le \frac{38\varepsilon}{12h} + \frac{Mh^4}{20}, \tag{8}$$

and the value of h that minimizes the right-hand side of formula (8) is

$$h = \left(\frac{95\varepsilon}{6M}\right)^{1/5}. \tag{9}$$

Following the above two derivations, we can similarly obtain the optimal step-length formulas of the 1-step-ahead numerical differentiation formulas using different numbers of data points. Specifically, the optimal step-length formulas of the 1-step-ahead numerical differentiation formulas using four to seven data points are shown in Table 2.

3.2. Numerical verification

As we mentioned above, different from the backward numerical differentiation formula, the 1-step-ahead numerical differentiation formula can adapt to the fast variational rate of the first-order derivative of target point for the reason that the latter one takes the data points on both sides into consideration. It is worth noting that we carry sixteen decimal places in all the calculations, thus the round-off error ε is set as 0.5×10^{-16}. To analyze the total errors of the 1-step-ahead numerical differentiation formula, we use the following function as the target function in this subsection:

$$f_1(x) = \cos(x).$$

Since the first-order derivative of the target function $f_1'(x) = -\sin(x)$ has the fast variational rate at $x = \pm k\pi$, $(k = 0, 1, 2, 3, \ldots)$, we use $x = 0$ (i.e., $k = 0$) as the target point in the following simulations. Additionally, eighteen different

Table 3
Errors in approximation to $f_1'(x) = -\sin(x)$ using the 1-step-ahead formulas with different step lengths h.

h	4 data points	5 data points	6 data points	7 data points
0.5	$9.9906861022 \times 10^{-3}$	$1.2230341975 \times 10^{-3}$	$7.9876250224 \times 10^{-4}$	$2.7011594455 \times 10^{-4}$
0.2	$6.6223553159 \times 10^{-4}$	$1.3200620420 \times 10^{-5}$	$1.0244736087 \times 10^{-5}$	$5.1787483101 \times 10^{-7}$
0.1	$8.3194548564 \times 10^{-5}$	$4.1562621196 \times 10^{-7}$	$3.3000929278 \times 10^{-7}$	$4.1361971625 \times 10^{-9}$
0.05	$1.0412327201 \times 10^{-5}$	$1.3012699066 \times 10^{-5}$	$1.0390640644 \times 10^{-8}$	$3.2486013878 \times 10^{-11}$
0.02	$6.6662223148 \times 10^{-7}$	$1.3333038377 \times 10^{-10}$	$1.0662877987 \times 10^{-10}$	$4.7369515717 \times 10^{-14}$
0.01	$8.3331930245 \times 10^{-8}$	$4.1892415462 \times 10^{-12}$	$3.2921813423 \times 10^{-12}$	$3.2566542055 \times 10^{-14}$
0.005	$1.0416652725 \times 10^{-8}$	$1.5543122344 \times 10^{-13}$	$1.1842378929 \times 10^{-13}$	$3.1086244689 \times 10^{-14}$
0.002	$6.6671668186 \times 10^{-10}$	$1.3877787807 \times 10^{-14}$	$5.1810407815 \times 10^{-14}$	$2.5905203907 \times 10^{-14}$
0.001	$8.3488771451 \times 10^{-11}$	$1.5728159515 \times 10^{-13}$	$7.4014868308 \times 10^{-15}$	$1.9984014443 \times 10^{-14}$
5×10^{-4}	$1.0214051826 \times 10^{-11}$	$1.8503717077 \times 10^{-13}$	$2.8125649957 \times 10^{-13}$	$1.9984014443 \times 10^{-13}$
2×10^{-4}	$4.6259292692 \times 10^{-13}$	$3.2381504884 \times 10^{-13}$	$2.2204460492 \times 10^{-13}$	$2.2204460492 \times 10^{-13}$
1×10^{-4}	$0.0000000000 \times 10^{-0}$	$1.0177044392 \times 10^{-12}$	$1.4802973661 \times 10^{-12}$	$4.8109664400 \times 10^{-13}$
5×10^{-5}	$1.4802973661 \times 10^{-12}$	$1.2952601953 \times 10^{-12}$	$2.9605947323 \times 10^{-12}$	$4.0708177569 \times 10^{-12}$
2×10^{-5}	$9.2518585385 \times 10^{-13}$	$3.2381504884 \times 10^{-12}$	$3.7007434154 \times 10^{-12}$	$3.7007434154 \times 10^{-12}$
1×10^{-5}	$0.0000000000 \times 10^{-0}$	$2.2204460492 \times 10^{-11}$	$5.1810407815 \times 10^{-12}$	$2.3314683517 \times 10^{-11}$
5×10^{-6}	$2.9605947323 \times 10^{-11}$	$2.5905203907 \times 10^{-11}$	$2.3684757858 \times 10^{-11}$	$1.0362081563 \times 10^{-11}$
2×10^{-6}	$9.2518585385 \times 10^{-11}$	$6.4763009769 \times 10^{-11}$	$7.4014868308 \times 10^{-11}$	$1.4802973661 \times 10^{-11}$
1×10^{-6}	$3.7007434154 \times 10^{-11}$	$3.3306690738 \times 10^{-10}$	$2.9605947323 \times 10^{-11}$	$2.2944609175 \times 10^{-10}$

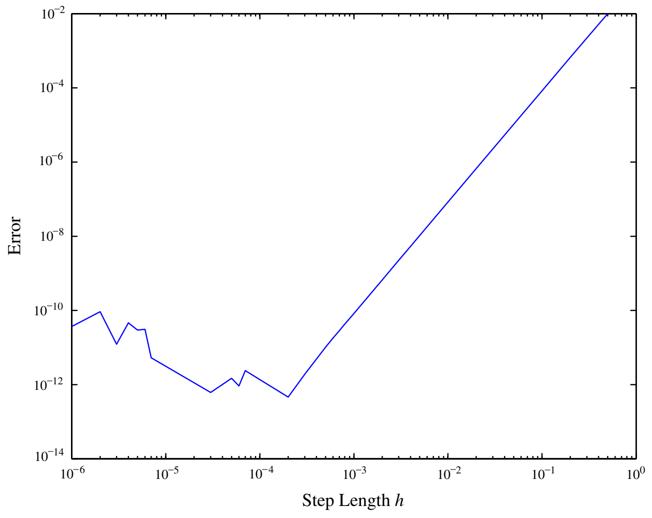

Fig. 1. Errors with the step length h changing from 10^{-6} to 0.5 using 4 data points.

step lengths h are investigated to verify the efficacy of the 1-step-ahead numerical differentiation formulas. Besides, the errors between the first-order derivative of target function values $f_1'(x)$ and the first-order derivative of the interpolation polynomial values $\phi_{f1}'(x)$ at the target point are presented in Table 3 (note that we only show ten decimal places of errors due to space limitation).

In addition, for the target function $f_1(x) = \cos(x)$, $|f_1^{(4)}(x)| = |\cos(x)| \leq 1 = M$. From Table 2, we can get the optimal step length $h = 1.4142 \times 10^{-4}$ for the 1-step-ahead numerical differentiation formula using 4 data points. To be more intuitive, we can present the relationship between the errors and the step length h (varying from 10^{-6} to 0.5) in Fig. 1 for the 1-step-ahead numerical differentiation formula using 4 data points.

From Fig. 1, we can see the minimal error occurs approximately at the step length $h = 10^{-4}$, which substantiates that the above optimal step-length formulas for the 1-step-ahead numerical differentiation formulas are effective and accurate.

In order to further verify the 1-step-ahead numerical differentiation formulas, we present another target function $f_2(x) = \arctan(x)$ with the target point being $x = 1/\sqrt{3}$. With the step length h changing from 10^{-6} to 0.5. Table 4 presents the errors between the first-order derivative of target function values $f_2'(x)$ and the first-order derivative of the interpolation polynomial values $\phi_{f2}'(x)$ at the target point.

4. Comparison with backward numerical differentiation formulas

The backward numerical differentiation formulas can be used to approximate the first-order derivative of the target function. In order to illustrate the superiority of the 1-step-ahead numerical differentiation formulas, we present the

Table 4
Errors in approximation to $f_2'(x) = \arctan'(x)$ using the 1-step-ahead formulas by different step lengths h.

h	4 data points	5 data points	6 data points	7 data points
0.5	$3.4322597261 \times 10^{-2}$	$2.6735325849 \times 10^{-2}$	$5.9998178013 \times 10^{-3}$	$1.5076288427 \times 10^{-2}$
0.2	$2.4164707297 \times 10^{-3}$	$3.3921025627 \times 10^{-4}$	$7.5974285402 \times 10^{-4}$	$9.6439958186 \times 10^{-5}$
0.1	$2.7655834630 \times 10^{-4}$	$4.3829933915 \times 10^{-5}$	$1.3503825063 \times 10^{-5}$	$8.0717829157 \times 10^{-6}$
0.05	$3.2576901872 \times 10^{-5}$	$3.0670502828 \times 10^{-6}$	$1.8806123514 \times 10^{-7}$	$1.2431557017 \times 10^{-7}$
0.02	$2.0048580308 \times 10^{-6}$	$8.0622992282 \times 10^{-8}$	$6.9323313756 \times 10^{-10}$	$4.2022285651 \times 10^{-10}$
0.01	$2.4723914515 \times 10^{-7}$	$5.0565872600 \times 10^{-9}$	$1.0508149905 \times 10^{-11}$	$6.0000893142 \times 10^{-12}$
0.005	$3.0694074393 \times 10^{-8}$	$3.1631985919 \times 10^{-10}$	$1.8152146452 \times 10^{-13}$	$1.0269562977 \times 10^{-13}$
0.002	$1.9563585196 \times 10^{-8}$	$8.0330186946 \times 10^{-12}$	$7.8936857050 \times 10^{-14}$	$7.8936857050 \times 10^{-14}$
0.001	$2.4413560062 \times 10^{-10}$	$7.7959860789 \times 10^{-13}$	$2.7633451082 \times 10^{-13}$	$5.1314508198 \times 10^{-14}$
5×10^{-4}	$3.0750735291 \times 10^{-11}$	$3.9412917374 \times 10^{-14}$	$4.3420822493 \times 10^{-13}$	$4.3420822493 \times 10^{-13}$
2×10^{-4}	$1.6628920462 \times 10^{-12}$	$9.2770235937 \times 10^{-13}$	$3.9523939676 \times 10^{-14}$	$1.2236878177 \times 10^{-12}$
1×10^{-4}	$9.2770235937 \times 10^{-13}$	$5.5266902165 \times 10^{-13}$	$3.9523939676 \times 10^{-14}$	$2.3289148387 \times 10^{-12}$
5×10^{-5}	$5.5266902165 \times 10^{-13}$	$5.5266902165 \times 10^{-13}$	$3.9523939676 \times 10^{-14}$	$3.9523939676 \times 10^{-14}$
2×10^{-5}	$1.2927436898 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$2.4079627181 \times 10^{-11}$
1×10^{-5}	$2.4079627181 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$3.3119174069 \times 10^{-11}$	$9.4344532186 \times 10^{-12}$
5×10^{-6}	$1.7211010394 \times 10^{-11}$	$3.2013947048 \times 10^{-11}$	$3.7935099506 \times 10^{-11}$	$3.7935099506 \times 10^{-11}$
2×10^{-6}	$1.2927436898 \times 10^{-12}$	$7.1606942597 \times 10^{-11}$	$5.6804005943 \times 10^{-11}$	$2.9365154752 \times 10^{-10}$
1×10^{-6}	$2.4079627181 \times 10^{-12}$	$1.5043766232 \times 10^{-10}$	$5.6804005943 \times 10^{-11}$	$2.9365154752 \times 10^{-10}$

Table 5
Backward numerical differentiation formulas using 4–7 data points.

Data points	Backward formulas	Truncation errors
4	$f'(x_3) \approx (11f(x_3) - 18f(x_2) + 9f(x_1) - 2f(x_0))/6h$	$o(h^3)$
5	$f'(x_4) \approx (25f(x_4) - 48f(x_3) + 36f(x_2) - 16f(x_1) + 3f(x_0))/12h$	$o(h^4)$
6	$f'(x_5) \approx (137f(x_5) - 300f(x_4) + 300f(x_3) - 200f(x_2) + 75f(x_1) - 12f(x_0))/60h$	$o(h^5)$
7	$f'(x_6) \approx (147f(x_6) - 360f(x_5) + 450f(x_4) - 400f(x_3) + 225f(x_2) - 72f(x_1) + 10f(x_0))/60h$	$o(h^6)$

Table 6
Errors in approximation to $f_1'(x) = -\sin(x)$ using backward formulas by different step lengths h.

h	4 data points	5 data points	6 data points	7 data points
0.5	$2.5079921516 \times 10^{-2}$	$8.8859493015 \times 10^{-3}$	$2.3731168439 \times 10^{-3}$	$2.0270607771 \times 10^{-3}$
0.2	$1.9339041131 \times 10^{-3}$	$1.0402616212 \times 10^{-4}$	$4.8116431434 \times 10^{-5}$	$4.5366627337 \times 10^{-6}$
0.1	$2.4792114083 \times 10^{-4}$	$3.3125513110 \times 10^{-6}$	$1.6252292311 \times 10^{-6}$	$3.6977686098 \times 10^{-8}$
0.05	$3.1184930826 \times 10^{-5}$	$1.0400401132 \times 10^{-7}$	$5.1758282400 \times 10^{-8}$	$2.9194779926 \times 10^{-10}$
0.02	$1.9993333710 \times 10^{-6}$	$1.0663710655 \times 10^{-9}$	$5.3281231278 \times 10^{-10}$	$4.8405723873 \times 10^{-13}$
0.01	$2.4997920030 \times 10^{-7}$	$3.3388107093 \times 10^{-11}$	$1.6620778827 \times 10^{-11}$	$7.9936057773 \times 10^{-14}$
0.005	$3.1249284641 \times 10^{-8}$	$9.4739031434 \times 10^{-13}$	$6.2764608325 \times 10^{-13}$	$1.7763568394 \times 10^{-13}$
0.002	$1.9999372528 \times 10^{-9}$	$1.8503717077 \times 10^{-13}$	$2.3684757858 \times 10^{-13}$	$3.7007434154 \times 10^{-13}$
0.001	$2.5020726231 \times 10^{-10}$	$1.1102230246 \times 10^{-13}$	$1.1842378929 \times 10^{-13}$	$9.1778436702 \times 10^{-13}$
5×10^{-4}	$3.1530333899 \times 10^{-11}$	$2.2204460492 \times 10^{-13}$	$1.1842378929 \times 10^{-13}$	$4.7369515717 \times 10^{-13}$
2×10^{-4}	$2.7755575615 \times 10^{-12}$	$1.1102230246 \times 10^{-12}$	$2.5165055224 \times 10^{-12}$	$2.9605947323 \times 10^{-13}$
1×10^{-4}	$1.1102230246 \times 10^{-12}$	$0.0000000000 \times 10^{-0}$	$4.7369515717 \times 10^{-12}$	$5.3290705182 \times 10^{-12}$
5×10^{-5}	$7.4014868308 \times 10^{-13}$	$5.9211894646 \times 10^{-12}$	$4.1448326252 \times 10^{-12}$	$2.0724163126 \times 10^{-11}$
2×10^{-5}	$1.2952601953 \times 10^{-11}$	$2.5905203907 \times 10^{-11}$	$1.9243865760 \times 10^{-11}$	$2.8125649957 \times 10^{-11}$
1×10^{-5}	$2.9605947323 \times 10^{-11}$	$3.7007434154 \times 10^{-12}$	$7.4014868308 \times 10^{-11}$	$8.8817841970 \times 10^{-12}$
5×10^{-6}	$1.4802973661 \times 10^{-11}$	$1.1842378929 \times 10^{-10}$	$1.1842378929 \times 10^{-11}$	$1.7763568394 \times 10^{-11}$
2×10^{-6}	$1.1102230246 \times 10^{-10}$	$3.7007434154 \times 10^{-10}$	$2.9605947323 \times 10^{-11}$	$5.9211894646 \times 10^{-11}$
1×10^{-6}	$4.4408920985 \times 10^{-10}$	$1.4802973661 \times 10^{-10}$	$5.3290705182 \times 10^{-10}$	$3.8487731520 \times 10^{-10}$

backward numerical differentiation formulas using four to seven data points in Table 5. With the step length h changing from 10^{-6} to 0.5, approximation errors of the backward numerical differentiation formulas using four to seven data points are shown in Tables 6 and 7 for the target functions $f_1(x) = \cos(x)$ and $f_2(x) = \arctan(x)$, respectively.

By comparing Tables 3 and 4 with Tables 6 and 7 at the same step length, the approximation errors of the first-order derivative of the target function using the 1-step-ahead numerical differentiation formulas are much smaller than those using the backward differentiation formulas. This can be explained as that the 1-step-ahead numerical differentiation formulas make use of a future data point in approximating the first-order derivative of the target function, and these formulas can thus adapt to the fast variational rates when the target point is close to the boundary.

In summary, from the above verification and comparison about the 1-step-ahead numerical differentiation formulas and backward numerical differentiation formulas, we can conclude that the 1-step-ahead numerical differentiation formulas are not only effective in computing the approximation of the first-order derivative of the target function, but also superior to the backward numerical differentiation formulas in numerical computations and applications.

Table 7
Errors in approximation to $f_2'(x) = \arctan'(x)$ using backward formulas by different step lengths h.

h	4 data points	5 data points	6 data points	7 data points
0.5	$3.9735116141 \times 10^{-3}$	$7.6942214392 \times 10^{-2}$	$1.2045681957 \times 10^{-1}$	$1.3317102817 \times 10^{-1}$
0.2	$8.6062532141 \times 10^{-3}$	$2.4418732451 \times 10^{-3}$	$3.2200745211 \times 10^{-3}$	$3.7563832623 \times 10^{-3}$
0.1	$1.0049947745 \times 10^{-3}$	$1.0780061034 \times 10^{-4}$	$1.1594982281 \times 10^{-4}$	$3.1613369791 \times 10^{-5}$
0.05	$1.0999890675 \times 10^{-4}$	$1.1327894944 \times 10^{-5}$	$1.6861995943 \times 10^{-6}$	$8.1402131191 \times 10^{-7}$
0.02	$6.3370660817 \times 10^{-6}$	$3.1902580333 \times 10^{-7}$	$5.9874910585 \times 10^{-9}$	$2.7270551372 \times 10^{-9}$
0.01	$7.6194379206 \times 10^{-7}$	$2.0173808290 \times 10^{-8}$	$8.8731022529 \times 10^{-11}$	$3.7650882411 \times 10^{-11}$
0.005	$9.3347413798 \times 10^{-8}$	$1.2644897351 \times 10^{-9}$	$1.4763745781 \times 10^{-12}$	$4.1844305798 \times 10^{-13}$
0.002	$5.9010968334 \times 10^{-9}$	$3.2527092130 \times 10^{-11}$	$5.1314508198 \times 10^{-13}$	$3.9412917374 \times 10^{-14}$
0.001	$7.3485906249 \times 10^{-10}$	$1.1447509606 \times 10^{-12}$	$9.8676622428 \times 10^{-13}$	$1.9343415758 \times 10^{-14}$
5×10^{-4}	$9.0633611726 \times 10^{-11}$	$1.1447509606 \times 10^{-12}$	$1.8552936964 \times 10^{-12}$	$1.8552936964 \times 10^{-12}$
2×10^{-4}	$6.8488548166 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$3.9523939676 \times 10^{-14}$	$4.7765125188 \times 10^{-12}$
1×10^{-4}	$2.4079627181 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$3.9523939676 \times 10^{-14}$	$9.5133900757 \times 10^{-15}$
5×10^{-5}	$3.5131897391 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$9.4344532186 \times 10^{-12}$	$2.8382185490 \times 10^{-11}$
2×10^{-5}	$2.4079627181 \times 10^{-12}$	$2.7197910590 \times 10^{-11}$	$9.4344532186 \times 10^{-12}$	$3.7935099506 \times 10^{-11}$
1×10^{-5}	$2.4079627181 \times 10^{-12}$	$2.4079627181 \times 10^{-12}$	$1.0417344764 \times 10^{-10}$	$9.4344532186 \times 10^{-12}$
5×10^{-6}	$6.1619931379 \times 10^{-11}$	$6.1619931379 \times 10^{-11}$	$9.4344532186 \times 10^{-12}$	$1.8004364665 \times 10^{-10}$
2×10^{-6}	$1.4562184791 \times 10^{-10}$	$2.9365154752 \times 10^{-10}$	$2.9365154752 \times 10^{-10}$	$7.6734674170 \times 10^{-10}$
1×10^{-6}	$2.4079627181 \times 10^{-12}$	$2.9846736193 \times 10^{-10}$	$7.6734674170 \times 10^{-10}$	$1.1274339239 \times 10^{-9}$

5. Conclusion

In this paper, we present a special kind of numerical differentiation formulas called 1-step-ahead numerical differentiation formulas to approximate the first-order derivative of the target function. Specifically, different from the traditional forward/backward numerical differentiation formulas, the 1-step-ahead numerical differentiation formulas make use of a future data point in the approximation of the first-order derivative at the target point. In addition, we present the derivation for a group of formulas which can obtain the optimal step length for the 1-step-ahead numerical differentiation formulas. Finally, we have verified the efficiency and efficacy of the optimal step-length formulas by computing the first-order derivative at the target point with different step lengths. Moreover, the comparison between the approximation errors of the 1-step-ahead numerical differentiation formulas and the backward numerical differentiation formulas further substantiate the superiority of the 1-step-ahead numerical differentiation formulas.

Appendix A. Proof of formula (4)

Proof. According to Ref. [5], the Lagrange interpolation polynomial $\phi_n(x)$ of degree n can be formulated as follows:

$$\phi_n(x) = \sum_{i=0}^{n} f(x_i)\mathcal{L}_i(x), \tag{10}$$

where $\mathcal{L}_i(x)$ is the Lagrange basis polynomial using given data points:

$$\mathcal{L}_i(x) = \prod_{j=0,j\neq i}^{n} \frac{x-x_j}{x_i-x_j}. \tag{11}$$

Then let

$$\omega(x) = (x-x_0)(x-x_1)\cdots(x-x_n) = \prod_{i=0}^{n}(x-x_i),$$

and the first-order derivative of $\omega(x)$ can thus be obtained as

$$\omega'(x) = \sum_{i=0}^{n}((x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)).$$

Hence,

$$\mathcal{L}_i(x) = \prod_{j=0,j\neq i}^{n} \frac{x-x_j}{x_i-x_j} = \frac{\omega(x)}{(x-x_i)\omega'(x_i)}.$$

Formula (10) can be rewritten as

$$\phi_n(x) = \sum_{i=0}^{n} f(x_i)\mathcal{L}_i(x) = \sum_{i=0}^{n} f(x_i)\frac{\omega(x)}{(x-x_i)\omega'(x_i)}. \tag{12}$$

With $i \neq j$, let

$$\omega_i(x) = \frac{\omega(x)}{(x - x_i)}, \qquad \omega_{i,j}(x) = \frac{\omega(x)}{(x - x_i)(x - x_j)},$$

then

$$\frac{\omega_{i,j}(x)}{\omega_i(x)} = \frac{1}{(x - x_j)}.$$

So we can get

$$\omega_i(x_i) = \omega'(x_i).$$

In order to get the first-order derivative of $\phi_n(x)$, we rewrite $\phi_n(x)$ as

$$\phi_n(x) = \sum_{j=0, j\neq i}^{n} f(x_j)\frac{\omega(x)}{(x - x_j)\omega'(x_j)} + f(x_i)\frac{\omega_i(x)}{\omega'(x_i)}$$

and thus,

$$\phi_n'(x) = \sum_{j=0, j\neq i}^{n} f(x_j)\frac{\omega'(x)(x - x_j) - \omega(x)}{(x - x_j)^2\omega'(x_j)} + f(x_i)\frac{\omega_i'(x)}{\omega'(x_i)}.$$

It follows that

$$\phi_n'(x) = \sum_{j=0, j\neq i}^{n} f(x_j)\left(\frac{\omega'(x)}{(x - x_j)\omega'(x_j)} - \frac{\omega(x)}{(x - x_j)^2\omega'(x_j)}\right) + f(x_i)\frac{\omega_i'(x)}{\omega'(x_i)}.$$

Noting that, for $i = 0, 1, 2, \ldots, n$, $\omega(x_i) = 0$ and $\omega_i'(x) = \sum_{j=0, j\neq i}^{n} \omega_{i,j}(x)$, we can further get $\phi_n'(x_i)$ as

$$\phi_n'(x_i) = \sum_{j=0, j\neq i}^{n} f(x_j)\frac{\omega'(x_i)}{(x_i - x_j)\omega'(x_j)} + f(x_i)\frac{\sum_{j=0, j\neq i}^{n} \omega_{i,j}(x_i)}{\omega'(x_i)},$$

i.e.,

$$\phi_n'(x_i) = \sum_{j=0, j\neq i}^{n} f(x_j)\frac{\omega'(x_i)}{(x_i - x_j)\omega'(x_j)} + f(x_i)\sum_{j=0, j\neq i}^{n} \frac{1}{x_i - x_j}.$$

When $\{x_i, i = 0, 1, 2, 3, \ldots, n\}$ are equally spaced, i.e., $x_{i+1} - x_i = h$, we can obtain the equally spaced differentiation formula as follows:

$$\phi_n'(x_i) = \frac{1}{h}\sum_{j=0, j\neq i}^{n} \frac{(-1)^{i-j}C_n^j/C_n^i f(x_j) + f(x_i)}{i - j}.$$

According to Lemma 1, the first-order derivative of the remainder term of the unknown target function $f(x)$ can be written as

$$\sigma_{n+1}'(x_i) = \frac{(-1)^{n-i}h^n f^{n+1}(c)}{(n+1)C_n^i}.$$

Therefore,

$$f'(x) = \phi_n'(x) + \sigma_{n+1}'(x).$$

The proof is thus complete. □

Appendix B

The equally spaced 1-step-ahead numerical differentiation formulas using eight to sixteen data points are shown in Table 8.

Table 8
The 1-step-ahead numerical differentiation formulas using 8–16 data points.

Data points	The 1-step-ahead numerical difference formulas	Truncation errors
8	$f'(x_6) \approx (60f(x_7) + 609f(x_6) - 1260f(x_5) + 1050f(x_4) - 700f(x_3) + 315f(x_2) - 84f(x_1) + 10f(x_0))/420h$	$o(h^7)$
9	$f'(x_7) \approx (105f(x_8) + 1338f(x_7) - 2940f(x_6) + 2940f(x_5) - 2450f(x_4) + 1470f(x_3) - 588f(x_2) + 140f(x_1) - 15f(x_0))/(840h)$	$o(h^8)$
10	$f'(x_8) \approx (280f(x_9) + 4329f(x_8) - 10080f(x_7) + 11760f(x_6) - 11760f(x_5) + 8820f(x_4) - 4704f(x_3) + 1680f(x_2) - 360f(x_1) + 35f(x_0))/(2520h)$	$o(h^9)$
11	$f'(x_9) \approx (252f(x_{10}) + 4609f(x_9) - 11340f(x_8) + 15120f(x_7) - 17640f(x_6) + 15876f(x_5) - 10584f(x_4) + 5040f(x_3) - 1620f(x_2) + 315f(x_1) - 28f(x_0))/(2520h)$	$o(h^{10})$
12	$f'(x_{10}) \approx (2520f(x_{11}) + 53471f(x_{10}) - 138600f(x_9) + 207900f(x_8) - 277200f(x_7) + 291060f(x_6) - 232848f(x_5) + 138600f(x_4) - 59400f(x_3) + 17325f(x_2) - 3080f(x_1) + 252f(x_0))/(27720h)$	$o(h^{11})$
13	$f'(x_{11}) \approx (2310f(x_{12}) + 55991f(x_{11}) - 152460f(x_{10}) + 254100f(x_9) - 381150f(x_8) + 457380f(x_7) - 426888f(x_6) + 304920f(x_5) - 163350f(x_4) + 63525f(x_3) - 16940f(x_2) + 2772f(x_1) - 210f(x_0))/(27720h)$	$o(h^{12})$
14	$f'(x_{12}) \approx (27720f(x_{13}) + 757913f(x_{12}) - 2162160f(x_{11}) + 3963960f(x_{10}) - 6606600f(x_9) + 8918910f(x_8) - 9513504f(x_7) + 7927920f(x_6) - 5096520f(x_5) + 2477475f(x_4) - 880880f(x_3) + 216216f(x_2) - 32760f(x_1) + 2310f(x_0))/(360360h)$	$o(h^{13})$
15	$f'(x_{13}) \approx (25740f(x_{14}) + 785633f(x_{13}) - 2342340f(x_{12}) + 4684680f(x_{11}) - 8588580f(x_{10}) + 12882870f(x_9) - 15459444f(x_8) + 14723280f(x_7) - 11042460f(x_6) + 6441435f(x_5) - 2862860f(x_4) + 936936f(x_3) - 212940f(x_2) + 30030f(x_1) - 1980f(x_0))/(360360h)$	$o(h^{14})$
16	$f'(x_{14}) \approx (24024f(x_{15}) + 811373f(x_{14}) - 2522520f(x_{13}) + 5465460f(x_{12}) - 10930920f(x_{11}) + 18036018f(x_{10}) - 24048024f(x_9) + 25765740f(x_8) - 22084920f(x_7) + 15030015f(x_6) - 8016008f(x_5) + 3279276f(x_4) - 993720f(x_3) + 210210f(x_2) - 27720f(x_1) + 1716f(x_0))/(360360h)$	$o(h^{15})$

References

[1] Y.N. Zhang, D.C. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, IEEE Trans. Neural Netw. 13 (5) (2002) 1053–1063.

[2] Y.N. Zhang, S.S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, IEEE Trans. Neural Netw. 16 (6) (2005) 1477–1490.

[3] Y.N. Zhang, K. Chen, Comparison on Zhang neural network and gradient neural network for time-varying linear matrix equation $AXB = C$ solving, in: Proceedings of International Conference on Industrial Technology, 2008, pp. 1–6.

[4] W.M. Ma, Y.N. Zhang, J. Wang, MATLAB simulink modeling and simulation of Zhang neural networks for online time-varying Sylvester equation solving, in: Proceedings of International Joint Conference on Neural Networks, Hong Kong, China, 2008, pp. 286–290.

[5] J.H. Mathews, K.D. Fink, Numerical Methods Using MATLAB, third ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1999.

[6] R.L. Burden, J.D. Faires, Numerical Analysis, seventh ed., Brooks/Cole, Pacific Grove, CA, 2001.

[7] D. Baleanu, O. Defterli, O.P. Agrawl, A central difference numerical scheme for fractional optimal control problems, J. Vib. Control 15 (4) (2009) 583–597.

[8] S.C. Chapra, R.P. Canale, Numerical Methods for Engineers, third ed., McGraw-Hill, New York, 1998.

[9] T.N. Krishnamurti, L. Bounoua, An Introduction to Numerical Weather Prediction Technique Techniques, CRC Press, Boca Raton, FL, 1995.

[10] J.P. Li, General explicit difference formulas for numerical differentiation, J. Comput. Appl. Math. 183 (2005) 29–52.

[11] I.R. Khan, R. Ohba, Closed-form expressions for the finite difference approximations of first and higher derivatives based on Taylor series, J. Comput. Appl. Math. 107 (1999) 179–193.

[12] I.R. Khan, R. Ohba, New finite difference formulas for numerical differentiation, J. Comput. Appl. Math. 126 (2000) 269–276.

[13] I.R. Khan, R. Ohba, N. Hozumi, Mathematical proof of closed form expressions for finite difference approximations based on Taylor series, J. Comput. Appl. Math. 150 (2003) 303–309.