POLYTECHNIC UNIVERSITY
Department of Computer and Information Science

One-Sided Difference Formula for the First Derivative

K. Ming Leung

Abstract: Derive a one-sided formula for the first derive of a continuous function, and to study the interplay between roundoff error and truncation error in practical applications.

Directory

Last Revision Date: April 17, 2004

We want to derive a formula that can be used to compute the first derivative of a given function $f(x)$ at any given point x assuming that we can compute the function at any arbitrary point to the right of x. Our interest here is to obtain the so-called one-sided difference formula for the first derivative.

We start with the Taylor expansion of the function about the point of interest, x,

$$f(x+h) \approx f(x) + f'(x)h + \frac{1}{2!}f''(x)h^2 + \frac{1}{3!}f'''(x)h^3 + \dots, \qquad (1)$$

where h, referred to as the step size, is supposed to be small. We are interested in the first derivative but clearly we don't want the formula for the first derivative to involve the second or higher derivatives since we don't know how to compute those.

If we drop all those derivatives and solve for $f'(x)$, we have a first order forward (one-sided) difference formula for the first derivative. This formula has first order accuracy since the first term neglected is proportional to h. (Note that one power of h is cancelled out in the derivation.) That kind of accuracy is sometimes not sufficient and therefore we want to derive a higher order formula.

In order to obtain a higher order formula, we need to somehow account for the term quadratic in h in the Taylor expansion. Clearly we need to consider the Taylor expansion at a new point. In the present case we can only consider points to the right of x. So we consider

$$f(x+2h) \approx f(x) + 2f'(x)h + \frac{4}{2!}f''(x)h^2 + \frac{8}{3!}f'''(x)h^3 + \dots \quad (2)$$

Notice that we can evaluate all the function values in Eqs. (1) and(2), except for any of the derivatives. In order to account for the h^2 term and at the same time avoid having to compute $f''(x)$, we look for linear combinations of Eqs. (1) and(2) so that the h^2 terms cancel one another. It is clear that we want to take 4 times Eq. (1) and subtract of Eq. (2). The result is

$$4f(x+h) - f(x+2h) = 3f(x) + 2f'(x)h - \frac{2}{3}f'''(x)h^3 + \dots$$

Solving for the first derivative gives the one-sided difference formula for the first derivative if we neglect all terms in h^2 and higher:

$$f'(x) = \frac{4f(x+h) - f(x+2h) - 3f(x)}{2h} - \frac{1}{3}f'''(x)h^2 + \ldots. \quad (3)$$

The truncation error, $E_{\text{truncation}}$, which is given by the magnitude of the term that has been neglected, is clearly bounded by $Mh^2/3$, where M is a bound for $f'''(x)$. Thus we have

$$E_{\text{truncation}} = \frac{Mh^2}{3}. \quad (4)$$

The one-sided difference formula is a second order scheme since the truncation error, goes as the second power of h. Thus the formula is more and more accurate with smaller and smaller values of h. For the same value of h, a second order scheme has less truncation error than a first order scheme.

However we want to consider the effect of rounding error when we apply this formula for numerical work. Assuming that the magnitudes of the rounding errors in computing the function values in Eq. (3) are given by by the machine ϵ. (Note that the situation in general is expected to be worst.) That is the computation has $f(x)$ any where between $f(x) + \epsilon$ and $f(x) - \epsilon$. Similarly the same is true for the values of $f(x+h)$ and $f(x+2h)$. The rounding error in computing the numerator in Eq. (3) therefore is given by 8ϵ. Notice that in error analysis, we must take the worst case scenario since we are looking for an upper bound of the error. (So that at the end we can claim that the error is no larger than a certain quantity.) Consequently the rounding error in evaluating the above formula is

$$E_{\text{rounding}} = 4\epsilon/h. \tag{5}$$

Thus rounding error increases with decreasing h.

The total computational error, E, is therefore bounded by the sum of these two errors

$$E = E_{\text{truncation}} + E_{\text{rounding}} = \frac{Mh^2}{3} + \frac{4\epsilon}{h}. \tag{6}$$

Since the first term coming from truncation decreases with decreasing h and the second term coming from rounding increases with decreasing h, there must be an optimal value for h that represents the best tradeoffs between these two sources of errors. This optimal value of h then yields the smallest total error.

To find this optimal value we differentiate E and set it to zero:

$$\frac{dE}{dh} = \frac{2M}{3}h - \frac{4\epsilon}{h^2} = 0.$$

Solving for h gives the optimal value

$$h_{\rm min} = \left(\frac{6\epsilon}{M}\right)^{1/3}.$$

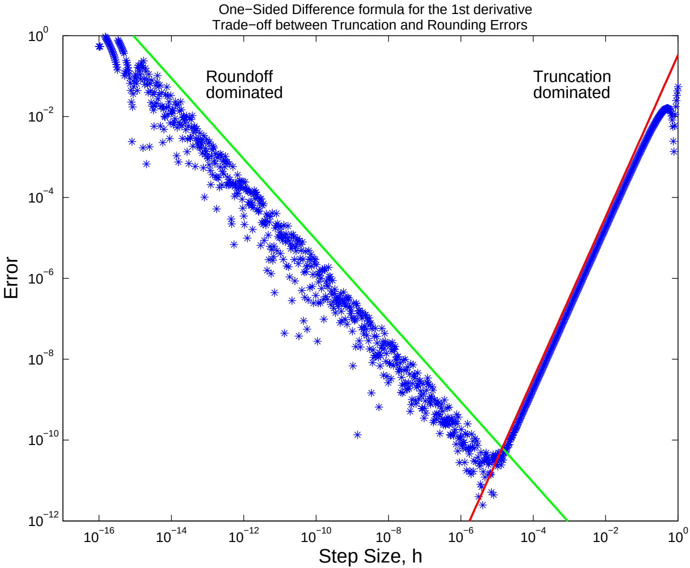
Inserting this optimal value for h into the expression for E gives the minimum error that can be achieved using this optimal h:

$$\begin{aligned} E_{\rm min} &= \frac{M}{3}\left(\frac{6\epsilon}{M}\right)^{2/3} + 4\epsilon\left(\frac{M}{3\epsilon}\right)^{1/3} = \frac{M}{3}\left(\frac{36\epsilon}{M^2}\right)^{1/3} + 4\epsilon\left(\frac{M}{3\epsilon}\right)^{1/3} \\ &= \left(\frac{4M\epsilon^2}{3}\right)^{1/3} + \left(\frac{8\times 4M\epsilon^2}{3}\right)^{1/3} = 3\left(\frac{4M\epsilon^2}{3}\right)^{1/3} \\ &= \left(36M\epsilon^2\right)^{1/3}. \end{aligned}$$

Notice that in this example, rounding errors contribute 2/3 and truncation errors contribute 1/3 to $E_{\rm min}$, the total minimum computational error. This is to be expected. As the order of a scheme goes up, the more and more is the limitation imposed by rounding on the ultimate highest accuracy that can be achieved.

One can easily check to see that the above optimal h clearly corresponds to a minimum of the total error.

As an example we compute using the forward difference formula the first derivative of the sine function $f(x) = \sin(x)$ at $x = 1$. The exact answer is clearly given by $\cos(1)(= 0.5403023058681\,4$, to machine accuracy). The second derivative is $-sin(x)$ and its magnitude is therefore bounded by $M = 1$. We use the formula for values of h varying from 10^{-16} to 10^{0}. For each value of h we computed the first derivative, from which we can compute the absolute value of the error. We then generate a log-log plot of the absolute value of the errors versus h. The result is shown in the following figure. Plotted in the figure in green is the truncation error as predicted by Eq.(4), and in red the rounding error as predicted by Eq.(5). We see that for small h, the error is dominated by the rounding error and indeed follows (and bounded by) the green line. The errors appear rather noisy as indicative of the rounding process. On the other hand, for large h, the error is dominated by the truncation error and indeed follows the red line. In MATLAB, the machine epsilon is about 2×10^{-16}. From Eq. () we find that $h_{\rm min}$ is bout 1×10^{-5} and from Eq. (7) we find that $E_{\rm min}$ is about 1×10^{-10}. These results are in accord with what we see in the figure.



One−Sided Difference formula for the 1st derivative
Trade−off between Truncation and Rounding Errors
Roundoff dominated
Truncation dominated
Error
Step Size, h
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-16}
10^{-14}
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^{0}