POLYTECHNIC UNIVERSITY
Department of Computer and Information Science

Forward Difference Formula for the First Derivative

K. Ming Leung

Abstract: Study the interplay between roundoff error and truncation error in using the forward difference formula for the first derivative.

Directory

Last Revision Date: February 2, 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. Our interest here is to obtain the so-called forward difference formula for the first derivative.

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

$$f(x+h) \approx f(x) + f'(x)h + \frac{f''(x)h^2}{2} + \ldots,$$

where h, referred to as the step size, is supposed to be small. Since we are interested in the first derivative, we solve for $f'(x)$ to give

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(x)h}{2} + \ldots.$$

Suppose the absolute value of the second derivative of the function at x is bounded by a positive number M, then assuming that h is a small quantity, we can drop the last term in the above equation to obtain the so-called forward difference formula for the first derivative:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \tag{1}$$

The truncation error, $E_{\text{truncation}}$, which is given by the magnitude of the term that has been neglected, is clearly bounded by $Mh/2$, thus we have

$$E_{\text{truncation}} = Mh/2. \tag{2}$$

The forward difference formula is referred to as a first order scheme since the truncation error, goes as the first power of h. Thus the formula is more and more accurate with smaller and smaller values of h.

However in order to achieve very high accuracy in computing the first derivative using the above formula, one must also consider the effect of rounding error. Assuming that the magnitudes of the rounding errors in computing the function values in Eq. (1) are given by by the machine ϵ. That is the computation has $f(x+h)$ any where between $f(x+h)+\epsilon$ and $f(x+h)-\epsilon$. Similarly for the value of $f(x)$. The difference of these two function values therefore has an error of

2ϵ. 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

$$E_{\text{rounding}} = 2\epsilon/h. \tag{3}$$

Thus rounding error increases with decreasing h. That is why we cannot simply use an extremely small h to compute the derivative.

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

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

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:

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

Solving for h gives the optimal value

$$h_{\min} = 2\sqrt{\frac{\epsilon}{M}}.$$

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

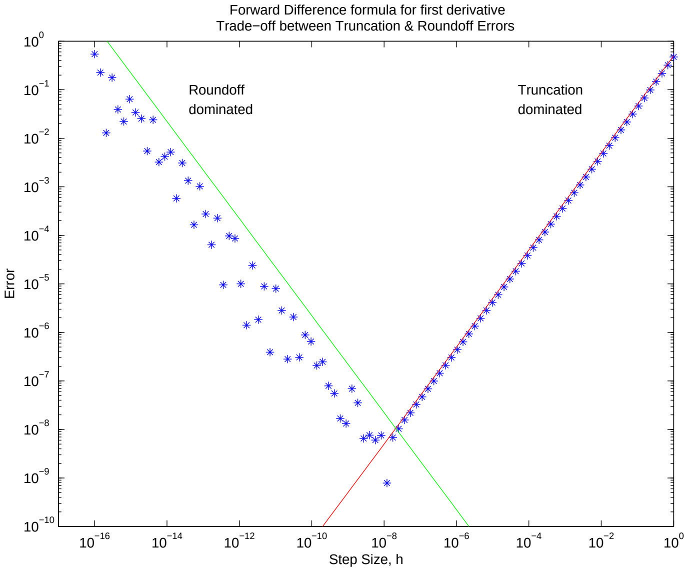
$$\begin{aligned} E_{\min} &= \frac{M}{2} 2\sqrt{\frac{\epsilon}{M}} + 2\epsilon\frac{1}{2}\sqrt{\frac{M}{\epsilon}} \\ &= \sqrt{M\epsilon} + \sqrt{M\epsilon} = 2\sqrt{M\epsilon}. \end{aligned} \tag{5}$$

Notice that in this example, truncation and rounding errors contribute equally to $E_{\min}$, the total minimum computational error.

We are so confident that the above optimal h clearly corresponds to a minimum that we don't even bother to compute the second derivative of E at the optimal h to check its sign.

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 given by $\cos(1) (= 0.54030230586814$, to machine accuracy). The second derivative is $-sin(x)$ and its magnitude is bounded by $M = 1$. We use the above formula for step size 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.(2), and in red is the rounding error as predicted by Eq.(3). 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}. Thus h_{min} is bout 3×10^{-8} and E_{min} is also about 3×10^{-8}. These results are in accord with what we see in the figure.


Forward Difference formula for first derivative
Trade−off between Truncation & Roundoff Errors
Roundoff dominated
Truncation dominated
Error
Step Size, h
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{-16}
10^{-14}
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^{0}