

Finite Difference Approximation of Derivatives

Mathematical Modeling Week 4

Kurt Bryan and Yosi Shibberu

Sometimes you have a differentiable function $f(x)$ for which there is no method to compute $f'(x)$. Or perhaps the formula for $f'(x)$ is simply unwieldy or difficult to compute. In this case one can approximate $f'(x)$ by using *finite differences*. The idea is simple: the derivative of a function $f(x)$ is defined by

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (1)$$

provided the limit exists. If f is all we know how to compute, then we can estimate f' by using the right hand side of equation (1). Pick a small number h and estimate $f'(x)$ by computing

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \quad (2)$$

There's only one problem: how small should h be?

There are two sources of error in using equation (2) to estimate derivatives. The first we will call the *differencing* error, which is due to the simple fact that the derivative is defined by taking the limit as h goes to zero but we take small, non-zero value for h . The differencing error can be estimated by using Taylor's Theorem, which says that if f is twice differentiable then

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2f''(a) \quad (3)$$

for some number a such that $x \leq a \leq x+h$. Of course, Taylor's Theorem only says that such a number a exists; it does not say what the number is, and we typically can't compute it. Also, the number a depends on h . Nonetheless, if you take equation (3) and put it into the finite difference approximation for $f'(x)$ you find that the difference error satisfies

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \frac{1}{2}hf''(a). \quad (4)$$

The last term represents the error due to the fact that h isn't zero.

The above equation seems to make it obvious: the differencing error gets smaller as h gets smaller. Unfortunately there is another source of error

in using equation (2) to approximate the derivative: Computers have only finite precision arithmetic. On most machines, if you compute a function in single precision arithmetic you get about 7 significant figures; in double precision, about 15 or 16. To see what effect this has on the finite difference, suppose that the computation of $f(x + h)$ is in error by an amount ϵ_1 , and the computation of $f(x)$ is in error by an amount ϵ_2 . The finite difference approximation is then really

$$\frac{f(x + h) + \epsilon_1 - f(x) - \epsilon_2}{h} = f'(x) + \frac{1}{2}hf''(a) + \frac{\epsilon_1 - \epsilon_2}{h}. \quad (5)$$

Note that the left hand side of the above equation is what *really* gets computed. The term $\frac{1}{2}hf''(a)$ is the differencing error; the term $(\epsilon_1 - \epsilon_2)/h$ we'll call the *numerical error*, due to the fact that the function isn't computed exactly. While the differencing error gets smaller as h goes to zero, the numerical error gets BIGGER. You might hope that $\epsilon_1 - \epsilon_2$ cancels out to zero, but that is highly unlikely; it's just as likely that ϵ_1 is positive and ϵ_2 is negative so that they reinforce each other.

Let's suppose that both $|\epsilon_1|$ and $|\epsilon_2|$ are less than some value ϵ . For double precision arithmetic we might have $\epsilon \approx 10^{-16}$. Then the worst case error in equation (5) is

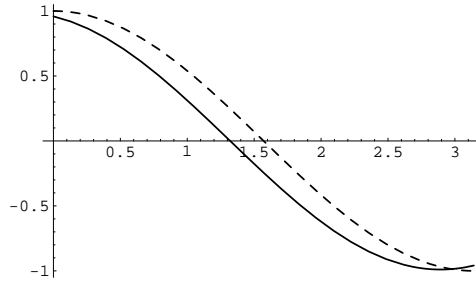
$$\left| f'(x) - \frac{f(x + h) + \epsilon_1 - f(x) - \epsilon_2}{h} \right| \leq \frac{1}{2}h \max |f''| + \frac{2\epsilon}{h}. \quad (6)$$

We should choose h to minimize the right hand side of equation (6). It's easy to check that the correct value of h is then

$$h = \sqrt{\frac{\epsilon}{\max |f''|}}. \quad (7)$$

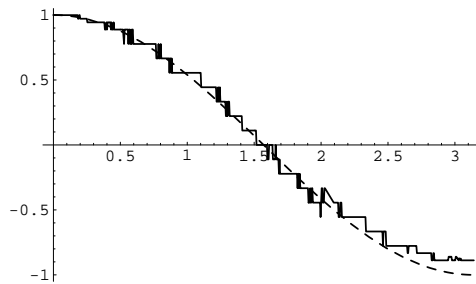
You might not know the maximum value of f'' on the interval of interest, but you can usually estimate it. The number ϵ is often called the *machine epsilon*.

Examples: Consider the function $f(x) = \sin x$. We know that $f''(x) = -\sin x$ and so the maximum value of f'' is one. If we use double precision arithmetic then we can estimate $\epsilon \approx 10^{-16}$. This indicates that the "best" value of h for estimating $f'(x)$ from equation (2) is $h = 10^{-8}$. First, look at what happens if we take h absurdly large, $h = 0.5$:



The dotted line is the true derivative and the solid line is the finite difference approximation.

Now consider taking h to be very small, say $h = 10^{-15}$. This yields



You can clearly see the effect of too small an increment in the finite difference formula.

The table below shows the maximum error in the finite difference approximation to $f'(x)$ for a variety of h from 1 to 10^{-16} :

h	max error
1.0	0.5
10^{-2}	5×10^{-3}
10^{-4}	5×10^{-5}
10^{-6}	5×10^{-7}
10^{-8}	1.5×10^{-8}
10^{-10}	1.0×10^{-6}
10^{-12}	1.5×10^{-4}
10^{-14}	3.0×10^{-2}
10^{-16}	1.0

These results are in quite good agreement with the analysis above. The best h really is around 10^{-8} .

One last warning—it's tempting to use a finite difference approximation to replace the derivative of a function $f(x)$ even when the function isn't differentiable. For example, replacing the derivative by the finite difference turns any optimization method that needs derivatives into an algorithm that doesn't need derivatives. But that doesn't mean the algorithm will work well on non-differentiable functions. Optimization algorithms that need derivatives almost always require the function to be smooth in order to get acceptable performance. If you shoehorn a non-differentiable function into an optimization algorithm using the finite difference trick, it will run, but poorly.