Numerical Methods for Derivatives

Leandro Borges

January 4, 2025

1 Introduction

The first derivative of a function $f(x)$ is given by the expression

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

Given a function $f(x)$ we can compute its derivative analytically by hand using the methods we learn in Calculus. However, when modeling physical systems, handling large data sets or dealing with simulations, it is unpractical or even impossible to solve everything by hand. At some point we can't avoid using a computer. So here I show how to solve derivatives using some numerical methods.

2 Errors

The good new is that computers can approximate the value of a derivative at some point x quite well. The bad new is that computers can only approximate it, they cannot give the exact answer unless it is under the range of decimal places allowed. Thus all methods come with some unavoidable errors. We have two kinds of errors: approximation and roundoff errors. The latter kind is due the finite memory of computers, which limits the precision. Bellow we show how approximation errors happen.

3 Method 1 - Forward Difference

Consider the Taylor expansion of the function $f(x+h)$:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(x) + ...$$

We can manipulate it easily, first isolate $hf'(x)$ then divide both sides by h to get:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) - \frac{h^2}{6}f'''(x) - \frac{h^3}{24}f^{(4)}(x) ...$$

Now, for simplicity, let

$$\mathcal{O}(h) \approx -\frac{h}{2}f''(x) - \frac{h^2}{6}f'''(x) - \frac{h^3}{24}f^{(4)}(x) - ... \approx -\frac{h}{2}f''(x)$$

The result is simply:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h)$$

This expression is quite similar to the definition of the derivative, it misses the $\lim_{h\to 0}$ part and adds the error term $\mathcal{O}(h)$, which results from approximation: $\mathcal{O}(h) \approx -\frac{h}{2}f''(x)$. We can do so because all the other terms in the Taylor expansion depend on some power of h, and we want h very small, so $h > h^n$ for $n \in \mathbb{N}$ and $0 < h < 1$. Besides, notice how the denominator keeps growing as a factorial. Thus, the most meaningful term to describe the error is the one we assigned to $\mathcal{O}(h)$. If we have the value of the analytical derivative at a point x, an estimate of the error can be obtained with:

$$\epsilon_{app} = \frac{|f_a'(x) - f_n'(x)|}{|f_a'(x)|}$$

Here f_a' is the analytical derivative, f_n' is the numerical one.

To implement the forward difference method, all we have to do is to express the definition of the derivative as it is. Here I do it in Python. The numerical derivative function takes three arguments: a function f to derive, a number x with respect to which we want to know the derivative and a point h to the right of x. This is where the name **forward difference** comes from (we are going from x to the right towards h). In the file "main.py" I will use this method to find $f'(x)$ for the function

$$f(x) = \frac{sin(x^2)e^{x/3}}{\sqrt{x^2+4}}$$

for $x = 3$. This function can be quite challenging to derive analytically. The implemented methods are in the file "methods.py". A graph of the function looks like this:

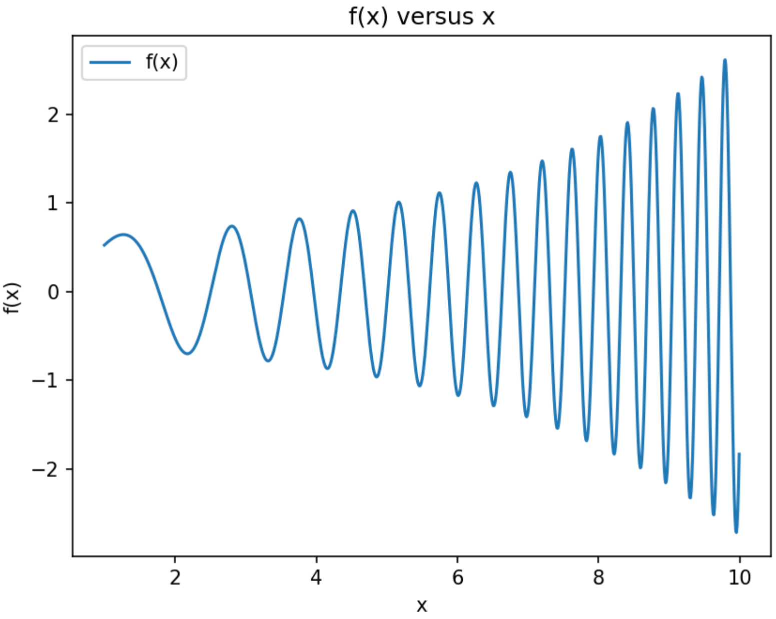

Figure 1: A graph of the function we are going to derive.

In order to see what happens for decreasing values of h, we use an array of values of h from 1 to 10^{-16}. The value of the derivative of f at point $x = 3$ is -4.08963. As the graph bellow shows us, values of h too close to 1 give a very unsatisfactory result, as do values smaller than 10^{-13}. So, we want values of h that are neither too large nor too small.

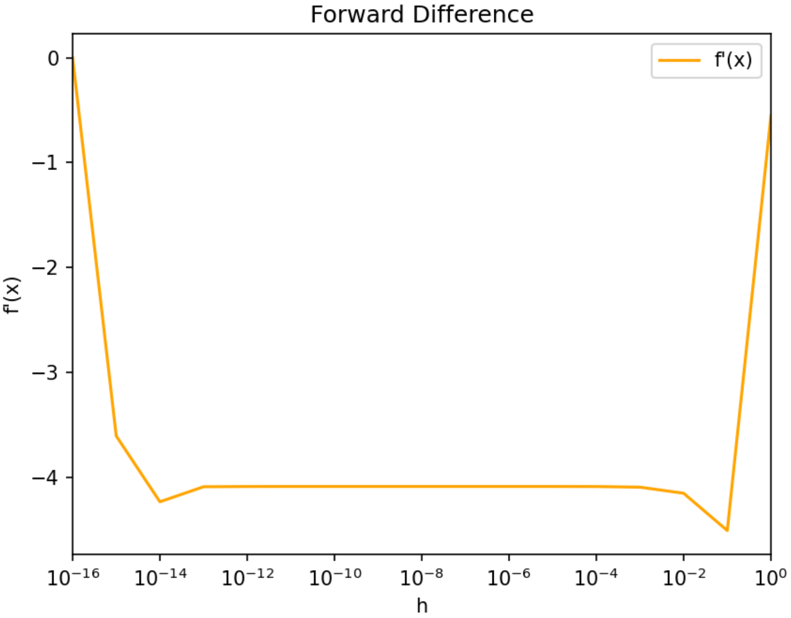

Figure 2: A graph showing how well the forward difference method can approximate the value of the derivative of a function about a point x, depending on the magnitude of h.