PHY 325 Notes

Computational Physics

Jaeden Bardati

Last modified January 25, 2022

1 Numerical Evaluation of Derivatives

Let $y = f(x)$, then the **derivative** is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}x} \equiv \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \tag{1}$$

Now consider $y = f(x_1, ..., x_n)$, then the **partial derivative** is defined as

$$\frac{\partial f}{\partial x_i} \equiv \lim_{h\to 0} \frac{f(x_1, ..., x_i + h, ..., x_n) - f(x_1, ..., x_n)}{h} \tag{2}$$

We can numerically compute derivatives in three main ways: forwards, backwards and with an average of the two.

The **forward** derivative is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+h) - f(x)}{h} \tag{3}$$

The **backward** derivative is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x-h) + f(x)}{h} \tag{4}$$

The **central** derivative is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+h) - f(x-h)}{2h} \tag{5}$$

The smaller n is, the more accurate this approximation is.

Note that, while the forward and backward derivatives take the same amount of time to compute, the central derivative has twice the number of calculations to do and so will take longer to run.

Furthermore, the central derivative can have divergent behaviour. To illustrate this, we can take the central derivative at the point (0, 0) on the absolute value function (see figure 1). Notice that the value of the derivative is 0 here, which could lead to undesired effects in the simulations of such systems.

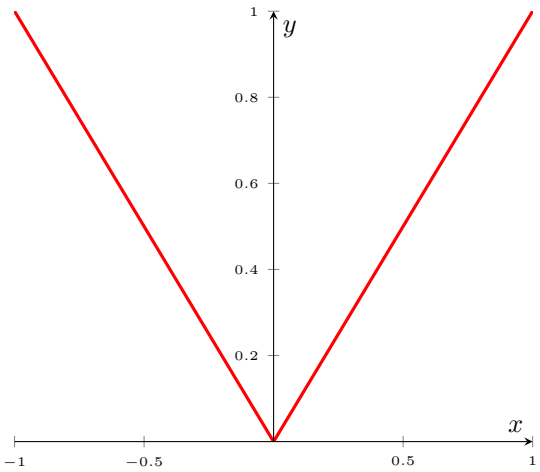

Figure 1: Absolute value function

2 Error in Derivative Calculation

When implementing a numerical derivative calculation, we would like to use a small value of h. But how small? One might think that we should try to make h as small as it can be. Of course, zero will not work since it would result in an error due to division by 0 (see equations 3, 4, and 5).

If we plot h by the error in the derivative, we will find that the derivative is inaccurate at both high and low values for h. There is a "sweet spot" in the middle (roughly around 10^{-8}) that minimizes the error. The error at low values of h is called **round-off error** and at high values of h is called **truncation error**.

2.1 Round-off Error

To understand this error, we must first ask: How are numbers represented by a computer? Specifically, we are interested in floats. The form is:

$$s \times M \times B^{e-E} \tag{6}$$

where s is the sign (0 if number is positive, 1 if negative), M is the Mantissa, B is the base, E is the bias, and e is the exponent.

This form is similar to scientific notation in principle. Namely, it is much more space-efficient to write 1.2×10^{-8} rather than 0.000000012. Note that here, 1.2 is the Mantissa, 10 is the base, the sign is 0, and $e - E = -8$.

For example, we will look at how 10.75 is stored. In binary, $(10)_{10} = (1010)_2$ and $(0.75)_{10} = (11)_2$, where the subscript indicates the base). So,

$$(10.75)_{10} = (1010.11)_2 = (1.01011)_2 \times 2^3$$

The bias term E is highly dependent on the particular machine you use. However, for a typical 32-bit computer, it is common to have a bias of $E = 2^{n-1} + 1$ with $n = 8$. This gives that $E = 129$. The base is, of course, $B = 2$.

Calculating e now, we know

$$e - E = 3 \implies e = 3 + E = 3 + 129 = 132$$

In binary, $(132)_{10} = (10000100)_2$.

Therefore, we would store the float value of 10.75 as

0	10000100	10101100000...000
s	$e - E$	M (23-bits)

In a 32-bit system, 1 bit is used to store the sign s, 8 bits for the exponent $e - E$, and 23 bits for the mantissa M. In a 64-bit system (double precision), 1 bit is used to store the sign s, 11 bits for the exponent $e - E$, and 52 bits for the mantissa M.

Since the mantissa is only a certain size, once a decimal value becomes lower than what can be stored in the mantissa, the value is lost. This is round-off error. The machine accuracy for a 32-bit system is around 10^{-8} and for a 64-bit system is 10^{-16}.

It is important when analyzing numerical systems to pick normalized units (close to unity). That is to say, choose units such that the numbers that are being used do not have this round-off error effect.

2.2 Truncation Error