Statistics, Uncertainties and Linear Fitting

Jorge A. Garcia

February 6th, 2018

1 Machine Numbers

A great resource that helped with this problem is the "IEEE-754 Floating Point Converter" by Schmidt, which can be found in the web. It allows to see the binary conversion of a number and even the error associated due to conversion (although this error probably should be taken with a grain of salt, since it itself is a product from a numerical computation).

Entering 0.0078125, the number can be represented as 1×2^{-7}, so it can be represented exactly on a computer. It requires only a value for the exponent, since the mantissa has always as given the value of $1 * 2^0$. This is what is called a machine number [Hjorth-Jensen, 2012]. Its binary representation in a 32-bit machine is then:

$$|0|01111000|00000000000000000000000|$$

Which is divided into $|sign|exponent|mantissa|$.

The number 0.2 on the other hand is not a linear combination of powers of 2, so it can not be represented exactly. It's then approximated with an error of roughly 3^{-9}. It's binary representation is then:

$$|0|01111100|10011001100110011001101|$$

2 Hermite Polynomials

In order to recreate the mathematical expressions for the first 5 Hermite polynomials, a Python module named SymPy was used. It functions similarly to the symbolic manipulation in MATLAB. Using a loop to determine the order of the derivative, Figure 1 shows the generated functions plotted from $[-3, 3]$.

$$H_0 = 1$$
$$H_1 = 2x$$
$$H_2 = 4x^2 - 2$$
$$H_3 = 8x^3 - 12x$$
$$H_4 = 16x^4 - 48x^2 + 12$$

For the indicated range, the numbers are sufficiently small that even a fourth power doesn't seem to alter the results that much.

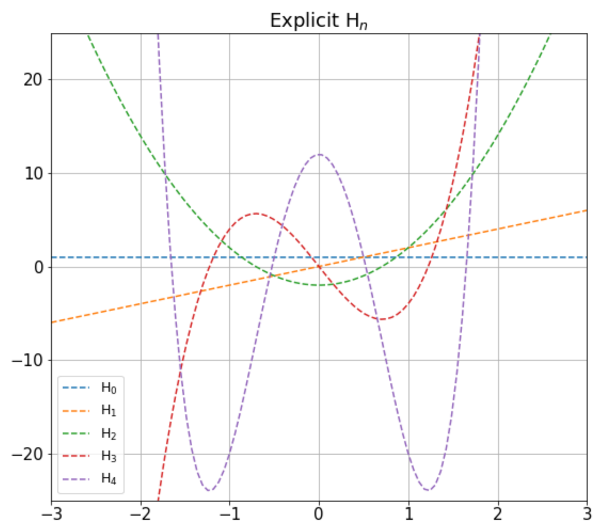

Figure 1: Hermite Polynomials: Explicit Method

A recursion function was then used as another approach to generate the Hermite polynomials. Figure 2 shows the corresponding graph of this method.

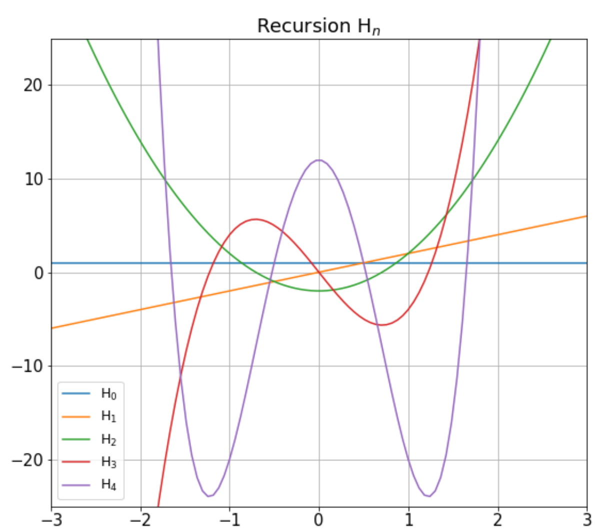

Figure 2: Hermite Polynomials: Recursion Method

To ensure that the mathematical expressions of the symbolic manipulation and the graphs were correct, Weissten's article on MathWorld was referenced. Figure 3 shows the plot he provides.

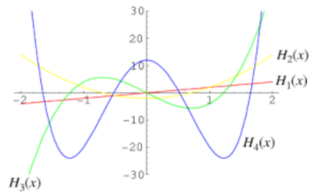

Figure 3: Hermite Polynomials [Weisstein, 2002a]

Comparisons for H_3 and H_4 can be seen in Figures 4 and 5 accordingly. They are similar in value, and there does not seem to be any noticeable difference between them.

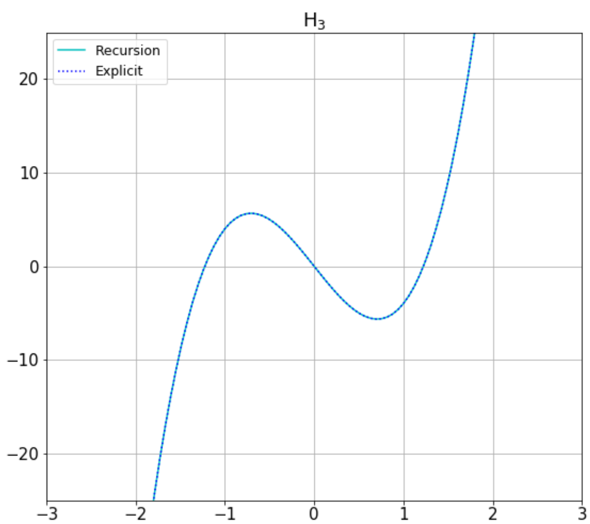

Figure 4: Explicit and Recursion for H_3

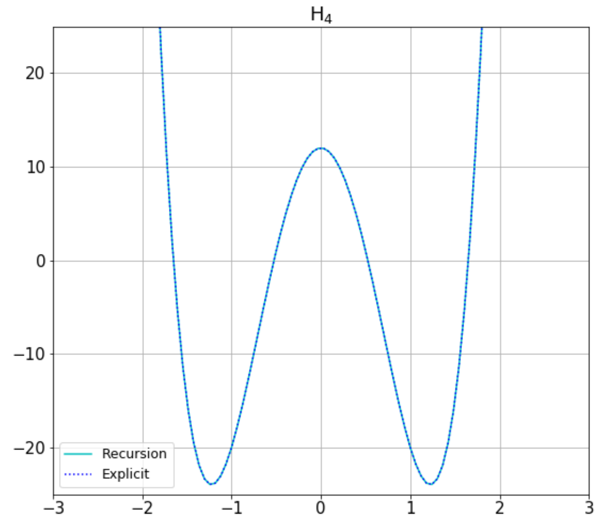

Figure 5: Explicit and Recursion for H_4

3 Legendre Polynomials

Using the provided recursion method for generating the Legendre Polynomials, plots were made for polynomials P_0 through P_5 and can be seen in Figure 6.

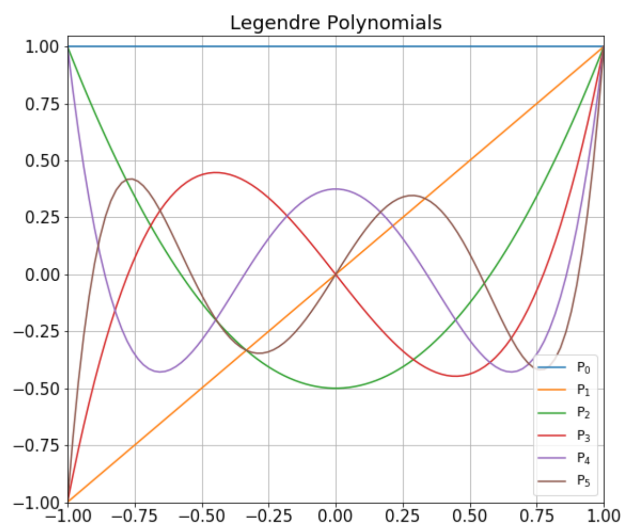

Figure 6: Legendre Polynomials

For good measure, this was compared to Weissten's article about Legendre Polynomials, his plot shown in Figure 7.

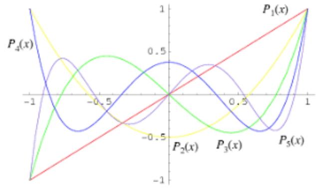

Figure 7: Legendre Polynomials [Weisstein, 2002b]

4 Numerical Derivatives

Three methods were used to approximated derivatives for the function $f(x) = x^4$ at $x = 10$. For the first derivative, 2-point and 4-point central finite differences were used, and for the second derivative a 2-point difference was used.

To see the effects of different step-sizes h, 5 linearly spaced vectors were created with 2 orders of magnitude of difference between the boundries (ie 100 and 1). These vectors where then stacked to define the interval for h, $h \in [10^2, 10^{-8}]$.

Figures 8 through 10 accordingly represent the plot of the unsigned absolute error against the step-size for approximations of the derivatives.

The 2-point approximation for the first derivative can be seen in Figure 8, andd seems to have best results when using step-sizes roughly between 10^{-5} and 10^{-4} orders of magnitude.

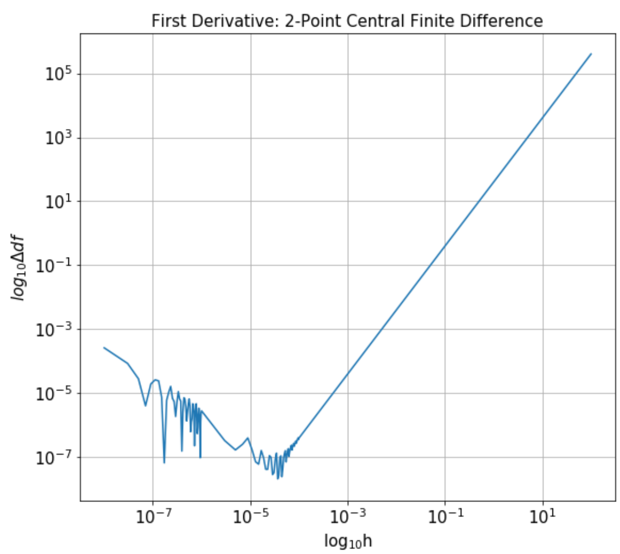

Figure 8: First Derivative: 2-Point Central Finite Difference

The 4-point approximation (Figure 9) was much more volatile with its error, but its optimal values are seen in step-sizes between 10^{-1} and 10^1.

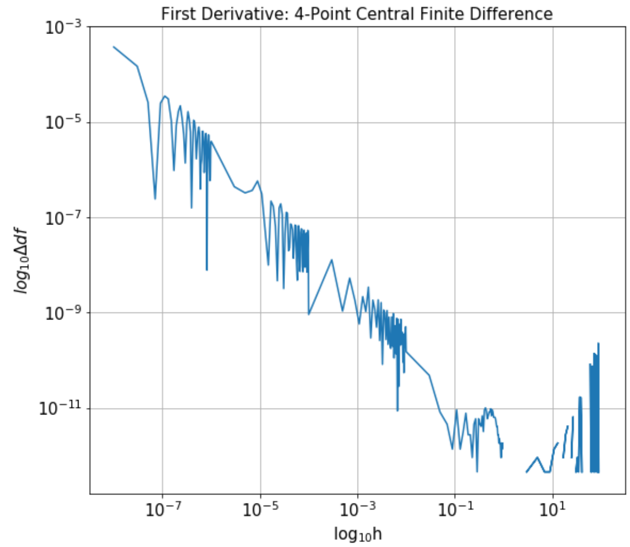

Figure 9: First Derivative: 4-Point Central Finite Difference

Figure 10 then shows the 2-point approximation for the second derivative of the function. Its minimal error can be seen when using a step-size of 10^{-3}.

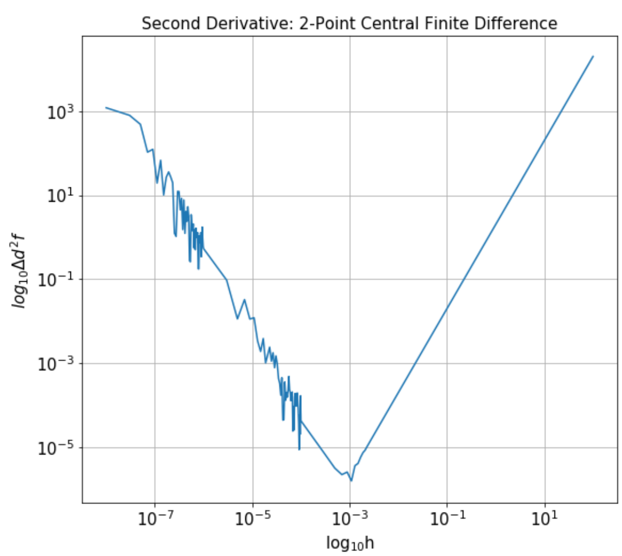

Figure 10: Second Derivative: 2-Point Central Finite Difference

References

[Hjorth-Jensen, 2012] Hjorth-Jensen, M. (2012). *Computational Physics*.

[Schmidt, 2015] Schmidt, H. (2015). Ieee-754 floating point converter. [Online; accessed Feb 07, 2018].

[Weisstein, 2002a] Weisstein, E. W. (2002a). Hermite polynomial. [Online; accessed Feb 07, 2018].

[Weisstein, 2002b] Weisstein, E. W. (2002b). Legendre polynomial. [Online; accessed Feb 07, 2018].