Math 541: Written Homework 5
Due: 03/13

Theory : Suppose for a collection of data points $f(x_n)$ I define the forward difference to be

$$\Delta f(x_n) = f(x_{n+1}) - f(x_n).$$

To define higher forward differences, I define them recursively via the formula

$$\Delta^k f(x_n) = \Delta\left(\Delta^{k-1} f(x_n)\right).$$

Why is this formula described as recursive? Using the definition above, find $\Delta^2 f(x_n)$.

Now suppose that I give you a mesh $(x_0, x_1, \cdots, x_n)$ that is evenly spaced, *i.e.*

$$x_{j+1} = x_j + h.$$

First, show $x_j = x_0 + jh$. Then show that

$$f[x_0, x_1] = \frac{1}{h}\Delta f(x_0), \;\; f[x_0, x_1, x_2] = \frac{1}{2h^2}\Delta^2 f(x_0).$$

Show that Δ is linear, *i.e.*

$$\Delta(f+g)(x_n) = \Delta f(x_n) + \Delta g(x_n).$$

Coding : As we talked about in class, Hermite interpolation is an extension of Lagrange interpolation where we use information about $f'(x)$ to improve our interpolatory approximation. So, suppose you are given mesh $(x_0, x_1, \cdots, x_n)$, data points $(f(x_0), f(x_1), \cdots, f(x_n))$, and derivatives $(f'(x_0), f'(x_1), \cdots, f'(x_n))$. We now introduce a modified divided difference scheme in the following way. Define a new mesh, $(z_0, z_1, \cdots, z_{2n+1})$ in the following way

$$z_{2i} = z_{2i+1} = x_i, \;\; i = 0, \cdots, n$$

Thus $z_0 = z_1 = x_0$, $z_2 = z_3 = x_1$, and so forth. As input to the divided difference scheme, we put in $(f(z_0), f(z_1), \cdots, f(z_{2n}), f(z_{2n+1}))$.

However, we see there is an issue with this. If we followed the usual divided difference approach we would have

$$f[z_0, z_1] = \frac{f(z_1) - f(z_0)}{(z_1 - z_0)} = \frac{0}{0},$$

which is undefined. Therefore, where $f[z_0, z_1]$ would appear in the divided difference tree, we set

$$f[z_0, z_1] = f'(x_0).$$

Likewise, we set

$$f[z_{2i}, z_{2i+1}] = f'(x_i).$$

We then proceed as we usually would in a divided difference tree. The coefficients we obtain then give us the Hermite interpolating polynomial in the form

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, \cdots, z_k](x - z_0)(x - z_1)\cdots(x - z_{k-1}).$$

See pg. 139 for reference.

Modify your existing divided difference code to generate the coefficients for Hermite interpolation. You can check your program using the values listed in Table 3.15 on page 138 against the results listed in Table 3.17 on page 141. Now write a program that evaluates $H_{2n+1}(x)$ at an arbitrary point x using the formula given above. Using the function

$$f(x) = \sin(\lambda x) + \cos(\lambda^2 x),$$

on the mesh $(0, .1, .2, \cdots, 1)$, generate plots showing the actual function, the Lagrange interpolating polynomial, and the Hermite polynomial for $\lambda = 1$, $\lambda = 2$, and $\lambda = 5$. Note, you need to use a finer mesh to generate your plots; perhaps cut the step size of the original mesh in half. Plot the absolute error in using the Lagrange interpolating polynomial and the Hermite interpolating polynomial. Describe your results.