Numerical Computing Homework 9
Due: Tuesday April 25 at the beginning of class or by email to rjs454@nyu.edu.

1. Write down the interpolating quadratic for the following set of points in three different forms: the monomial basis, the Lagrange basis, and the Newton basis. Show they are all equal.

<table>
<thead>
<tr>
<th>x</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

2. For the quadratic above, suppose we add one more data point (3, 6). Show the updated divided difference table and the new cubic interpolating polynomial.

3. Horner’s rule for polynomial evaluation is an algorithm that halves the number of multiplications. For a polynomial using the monomial basis
\[ p_n(x) = \sum_{i=0}^{n} a_i x^i \], Horner’s method is:
```
result = a_n;
for i = n-1 downto 0 by -1
    result = result*x + a_i
end
```
Write down (in pseudocode) a nested method to evaluate a polynomial in the Newton basis.

4. In this problem we want to fit a cubic, but instead of having 4 distinct interpolating points, one of the points has data from both a function value \( f(x) \) and a derivative value \( f'(x) \). The other two data points are distinct with function values associated with them.

Write down a linear system of equations to solve for the coefficients of the polynomial that interpolates these 4 pieces of information, \((x_0, f(x_0)), (x_1, f(x_1)),(x_2, f(x_2)), \text{ and } (x_2, f'(x_2))\). You can use the monomial basis for this.

5. Suppose you want to guarantee that the relative error in linear interpolation for the function \( y = \sqrt{x} \) is no larger than 1%. If your use two interpolating points, at \( x = 1 \) and \( x = 1 + \delta \), how large can \( \delta \) be?

6. This problem will look at the results of a Matlab script that you should write that compares a single high-order Lagrange polynomial of degree \( n \) with a cubic spline through the same data. Run your script for different values of \( n \) and different Lagrange polynomials - i.e., use a polynomial where the first data point is 1 and all the rest are zero, and another one where the non-zero data point is near the middle. Note that the Matlab commands to compute and plot a spline are in the text on p. 204. Comment on the results of your plots.