Homework 3

1. Write your own functions to perform polynomial interpolation. Use the following structure:

   a) Write a function \texttt{A/ddiff(x,y)} to create a divided difference array \texttt{A} from the points \((x(i),y(i))\) in the one-dimensional arrays \texttt{x} and \texttt{y}. The divided difference table is to be created in the two-dimensional array \texttt{A}. The input arrays \texttt{x} and \texttt{y} should be columns, but if they are rows, your program should transpose them if necessary. The array \texttt{A} may or may not contain \texttt{x} as its first column - your choice. Otherwise, it should have the form

   \[
   \begin{align*}
   f(x_0) & \rightarrow f[x_0, x_1] \rightarrow f[x_0, x_1, x_2] \rightarrow f[x_0, x_1, x_2, x_3] \rightarrow \cdots \\
   f(x_1) & \rightarrow f[x_1, x_2] \rightarrow f[x_1, x_2, x_3] \rightarrow \cdots \\
   f(x_2) & \rightarrow f[x_2, x_3] \rightarrow \cdots \\
   f(x_3) & \rightarrow \cdots \\
   \cdots
   \end{align*}
   \]

   where the arrows show how each successive column is to be calculated.

   b) Write a function \texttt{y/newteval(a,t,x)} that will evaluate the polynomial in Newton form given by

   \[
   a_0 + a_1 (x - t_0) + a_2 (x - t_0)(x - t_1) + \ldots + a_n (x - t_0)(x - t_1) \cdots (x - t_{n-1})
   \]

   where the \(a_i\) are in the array \texttt{a} and the \(t_i\) are in the array \texttt{t}, and \texttt{x} is an array of points at which the polynomial is to be evaluated. Use nested multiplication to evaluate the polynomial:

   \[
   p(x) = a_0 + [a_1 + [a_2 + \ldots + [a_{n-1} + [a_n (x - t_{n-1})] \ldots] (x - t_2)] (x - t_1)] (x - t_0)
   \]

   c) Finally, incorporate the functions of a) and b) into function \texttt{a/interp(x,y)} which will take interpolation points \((x(i),y(i))\) contained in the arrays \texttt{x} and \texttt{y}, calculate in the array \texttt{a} the coefficients of the interpolating polynomial in Newton form, and plot the interpolating polynomial and the interpolation points. (Plot the interpolation points as circles using \texttt{plot(x,y,'o')}).

   You will need to use the MATLAB functions \texttt{size( )} or \texttt{length( )} to determine the size of the arrays that are passed to the functions.

2. Use your interpolation program on Runge’s example: \(f(x) = \frac{1}{1 + x^2}\) on the interval \([-5, 5]\) with equally spaced points. Plot \(f(x)\) as well as \(p_n(x)\). Spacing of \(h = 1\) and \(h = .5\) should be adequate to see what is happening - what is happening?

3. Apply your interpolation program to interpolate \(f(x) = \cos(\pi x)\) on the interval \([0, 1]\). (Use spacing of \(.2, .1, .05\) ). Plot the interpolation error and see if it is consistent with the estimate that we developed in class. Add noise of size \(10^{-3}\) to the function and see the effect on the interpolant - to add noise of magnitude \(k\), add \(k \ast (2 \ast \text{rand(size(y))} - 1)\) to \texttt{y}. (Why is this the appropriate term to add?)
4. Recall that the Lagrange fundamental polynomials \( L_j \) interpolate the data \( y_i = 0, \ i \neq j \) and \( y_j = 1 \). For any other interpolation scheme, we will refer to the functions that interpolate the data \( y_i = 0, \ i \neq j \) and \( y_j = 1 \) as Lagrange fundamental functions. Find and plot the Lagrange fundamental functions for the cubic tabular interpolation scheme discussed in class; more specifically, we are given function values 
\[ f(a - h), f(a), f(a + h), f(a + 2h), \ldots, f(b - h), f(b), f(b + h) \] 
at the \( x \) values 
\[ x_{-1} = a - h, x_0 = a, \ldots, x_n = b, x_{n+1} = b + h, \] 
where \( nh = b - a \). Then for any \( x \) on the interval \([a, b]\) we approximate \( f(x) \) with the value \( p(x) \) of the cubic polynomial interpolating the function values at the four datapoints nearest to \( x \). You are to find and plot the approximating function obtained by applying this interpolation scheme to data for which \( f(x_j) = 1 \) and \( f(x_i) = 0 \) for \( i \neq j \), where \( 1 < j < n - 1 \) may be assumed.