1 Math 406: Ass. 1: Due Friday 21 Sept.

1. **Least squares fitting - $m-$th degree polynomial through N points.**

   (a) Find a system of equations for the coefficients of an $m-$th degree polynomial that fits a function f at given data points $(x_k, f(x_k)), k = 1, \ldots, N$ in the least squares sense. Write the equations in matrix form. What is the size of the matrix ?

   (b) Solve the system of equations using Matlab for $x_k = \{0, 0.2, 0.3, 0.4, 0.6, 0.7\}$, $f(x) = \cos x$, and $m = 4$ and 5. Evaluate the resulting polynomial at $x = 0.5$. Compare the answers to the "*exact*" value of $\cos x$. Use the MATLAB function "polyfit" to check the results.

2. **Lagrange interpolating polynomial.**

   (a) Plot the Lagrange interpolating basis functions $l_1^{(5)}(x)$ and $l_3^{(5)}(x)$ $(0 < x < 0.7)$, generated using $x_k = \{0, 0.2, 0.3, 0.4, 0.6, 0.7\}$. Use markers to highlight points x_k on the plots.

   (b) Plot the Lagrange interpolating polynomial $p_5(x)(0 < x < 0.7)$ that passes through the points $(x_k, f(x_k))$, $f(x) = \cos x$. Evaluate the resultant polynomial at $x = 0.5$. Compare the answer to the result in problem 1.

3. **Finite Difference Tables:** Let S_N^k denote the sum of the k th powers of the first N integers i.e.:

   $$S_N^k = \sum_{i=1}^{N} i^k$$

   Write a simple MATLAB program to evaluate these sums for a specified value of k for values of N from $1 \ldots k+3$. Now write MATLAB code to form the forward difference table (since the sample points are uniform). Notice that for each value of k the difference table terminates - why does this happen? For the special case $k = 2$ extract the differences from your table and use the Gregory-Newton divided difference formula to derive the formula:

   $$S_N^2 = \sum_{i=1}^{N} i^2 = \frac{1}{6} N \left(2N+1\right) \left(N+1\right)$$

4. **The pchip function in MATLAB:** Piecewise Cubic Hermite interpolation assumes the following representation of a given function $f(x)$ in terms of the cubic basis functions $h_i^{(0)}$ and $h_i^{(1)}$:

   $$f(x) \approx h(x) = \sum_{i=0}^{N} f(x_i) h_i^{(0)}(x) + \sum_{i=0}^{N} f^{'}(x_i) h_i^{(1)}(x)$$

where $h_i^{(0)}$ and $h_i^{(1)}$ are assumed to have the properties

$$\left.\begin{array}{ll} h_i^{(0)}(x_j) = \delta_{ij} & \frac{d}{dx} h_i^{(1)}(x_j) = \delta_{ij} \\ \frac{d}{dx} h_i^{(0)}(x_j) = 0 & h_i^{(1)}(x_j) = 0 \end{array}\right\} \tag{1}$$

(a) Show that the expressions for these basis functions are given by

$$\begin{aligned} h_i^{(0)}(x) &= \frac{[\Delta x_i + 2\,(x - x_i)]\,(x_{i+1} - x)^2}{(\Delta x_i)^3} \\ h_{i+1}^{(0)}(x) &= \frac{[\Delta x_i + 2\,(x_{i+1} - x)]\,(x - x_i)^2}{(\Delta x_i)^3} \\ h_i^{(1)}(x) &= \frac{(x - x_i)(x_{i+1} - x)^2}{(\Delta x_i)^2} \\ h_{i+1}^{(1)}(x) &= -\frac{(x_{i+1} - x)(x - x_i)^2}{(\Delta x_i)^2} \end{aligned}$$

(b) In order to specify the interpolant it is necessary to supply both the function values $f(x_i)$ and the derivatives $f'(x_i)$ at the interpolation points. Write a matlab function `hermite`(x, y, yp, xi) that will evaluate the Piecewise Cubic Hermite interpolant of a function sampled at the points given by the vector x and whose values and derivatives are given by y and yp respectively. Let xi be the set of desired sample points. For the function $f(x) = xe^{-x}$ on the interval $[0, 2]$ plot the Piecewise Cubic Hermite interpolant and the function f. Compare the value of the interpolant with the function at $x = 1/2$ by completing the table

$f(1/2)$	
$h(1/2)$	

(c) It is often not possible to provide the derivatives $f'(x_i)$ at the interpolation points. The pchip routine in MATLAB uses the following scheme to approximate the derivatives at the interpolation points. The slope at each interior node is taken to be a weighted harmonic mean of the slopes of the piecewise linear interpolant either side of the interior point. One-sided derivative conditions are imposed at the two end points. Now construct a pchip routine of your own to interpolate a function given only nodal values, constructing the weighted derivative approximations at the nodes and then calling your `hermite` routine.