Remember to adequately label all plots and include any MATLAB scripts and functions with your solutions. A clear and complete presentation of your solutions is required for full credit.

1. *Polynomial Interpolation.* By hand, find the unique polynomial $P(x)$ having degree at most two that passes through the three points $(-1, 1)$, $(2, 3)$, $(3, 0)$ using each of the following three approaches.

   (a) Vandermonde matrix.

   (b) Lagrange's interpolating polynomials.

   (c) Newton's divided differences.

   Verify that the polynomial you obtain is the same in (a)–(c).

2. *Lagrange's Interpolating Polynomials.*

   (a) Recall that the unique polynomial of degree $m \le n-1$ interpolating n points $x_1 < x_2 < \cdots < x_n$ can be written as

   $$P_m(x) = \sum_{i=1}^{n} y_i \, \ell_i(x) \quad \text{where} \quad \ell_i(x) = \prod_{\substack{k=1 \\ k \ne i}}^{n} \frac{x - x_k}{x_i - x_k}.$$

   Write a function with the declaration `function yy = lagrange(x,y,xx)` which evaluates the interpolating polynomial using the Lagrange polynomials. Its inputs are `x` and `y`, vectors containing the x- and y-coordinates of the data to be interpolated, and a vector `xx` of points at which to evaluate the polynomial. Its output `yy` is a vector containing the values of the polynomial.

   (b) Demonstrate empirically that the computational cost of this method of evaluating the interpolating polynomial is $\mathcal{O}(n^2)$ for large values of n. To do this, you might interpolate $f(x) = \sin(x)$ at n equally spaced points on the interval $-\pi \le x \le \pi$, where you take $n = \{10, 20, 50, 100, 200, 500, 1000, \dots\}$. Measure the time that it takes to evaluate the polynomial using the functions `tic` and `toc`, averaging over many evaluations. Then plot your results on log-log axes. Explain in a few sentences how your data verify that the cost is $\mathcal{O}(n^2)$.

3. *Newton's Divided Differences.*

   (a) Recall that the unique polynomial of degree $m \le n-1$ interpolating n points $x_1 < x_2 < \cdots < x_n$ can be written as

   $$P_m(x) = \sum_{i=1}^{n} c_i \, p_i(x) \quad \text{where} \quad p_i(x) = \prod_{k=1}^{i-1} (x - x_k)$$

   and the coefficients are given by the divided differences $c_i = f[x_1, x_2, \dots, x_i]$. Write a function with the declaration `function c = newton_coef(x,y)` which computes the coefficients of the Newton polynomials. Its inputs `x` and `y` are vectors containing the x- and y-coordinates of the data to be interpolated and its output `c` is a vector containing the coefficients $(c_1, c_2, \dots, c_n)$.

(b) Write a function with the declaration `function yy = newton_eval(c,x,xx)` which uses the nested formula

$$P(x) = c_1 + (x - x_1)\big[c_2 + (x - x_2)\big[c_3 + (x - x_3)\big[c_4 + \cdots + (x - x_{n-1})\, c_n\big]\big]\big]$$

to evaluate the interpolating polynomial. Its inputs are the vector `c` of coefficients, a vector `x` containing the x-coordinates of the data points, and a vector `xx` of points at which to evaluate the polynomial. Its output `yy` is a vector containing the values of the polynomial.

(c) Use the functions you have written to evaluate the polynomial that interpolates $f(x) = \sin(x)$ at 5 equally spaced points on the interval $-\pi \leq x \leq \pi$. Plot the function and its interpolating polynomial on this interval. Both of these should be smooth functions. Also, plot the interpolation error on this interval.

(d) Analytically determine an upper bound for the interpolation error at $x = \pi/4$. Does this match your calculations?