MA 780
Homework #1
Due January 27

1. Let A be the Vandermonde matrix generated by $\{x_i\}$, $i = 0, \cdots, n$. Show that

$$det(A) = \prod_{0 \le j < i \le n} (x_i - x_j),$$

and the solution of the polynomial interpolation exists and is unique if x_i's are distinct. **Hint:** Use mathematical induction and consider the determinant of the following:

$$det \begin{vmatrix} 1 & x_0 & \cdots & x_0^n & x_0^{n+1} \\ 1 & x_1 & \cdots & x_1^n & x_1^{n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n & x_n^{n+1} \\ 1 & x & \cdots & x^n & x^{n+1} \end{vmatrix} = \phi(x).$$

It is easy to spot n roots of $\phi(x)$ (if two rows are the same). Write down the factorized form of $\phi(x)$ and then use the induction to find the coefficient of x^{n+1}. Finally, you can plug x_{n+1} in to $\phi(x)$ to get the desired result.

2. Let the nodal points be $x_i = 0, \pi/6, \pi/4$, and the function values be $y_i = \sin(x_i)$.

   (a) Find the least squares approximation using $y = a_0 + a_1 x$. How is the solution defined?

   (b) Suppose that we use $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$. How to find the coefficients? How is the solution defined?

   (c) Find *both* the Lagrange polynomial and Newton polynomial interpolations. Show the two results are the same. Approximate $y(x)$ at $x = \pi/8$ and find the error $|\sin(\pi/8) - p_2(\pi/8)|$.

   (d) Give a least upper bound of the error $\max_{0 \le x \le \pi/4} |\sin x - p_2(x)|$.

   Hint: Use the error estimate and notice that $|\cos x| \le 1$, find the maximum/minimum of $\omega(x)$ between 0 and $\pi/4$.

3. Let $l_i(x)$ be the Lagrange polynomials, show the following:

$$\sum_{i=0}^{n} x_i^k l_i(x) = x^k, \qquad k = 0, 1, \cdots, n;$$

$$\sum_{i=0}^{n} (x_i - x)^k l_i(x) = 0, \qquad k = 1, 2, \cdots, n.$$

   Hint: For the second equality, use the binomial expansion of $(x_i - x)^k$ and make use of the first equality. Note that $(a - b)^m = \sum_{k=0}^{m} (-1)^k C_m^k a^{m-k} b^k$.

4. Use simple method to find the following divided differences:

   (a) $f[2^0, 2^1, \cdots, 2^7]$ if $f(x) = x^7 + x^3 + 1$.

(b) $f[2^0, 2^1, \cdots, 2^7, 2^8]$ if $f(x) = x^7 + x^3 + 1$.

(c) $f[x_0, x_1, x_2, \cdots, x_p]$ if $f(x) = \omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$ assuming $p \leq n$.

Hint: Use

$$f[x_0, x_1, \cdots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!} \qquad f[x_0, x_1, \cdots, x_n] = \sum_{i=0}^{n} \frac{f(x_i)}{\omega'_{n+1}(x_i)}.$$

5. Programming Part: Implement the Newton interpolation formula (see the code on the class web-page, and Program 66 on page 342), debug/test your code, and analyze your results.

- Use $\{0, 0.1, 0.2, \cdots 0.9, 1\}$ as nodal points.
- Use $\{\frac{k}{1000}\}, k = 0, 1, \cdots, 1000$ as output points.
- Plot the exact solution and the approximation from the polynomial interpolation on the same plot.
- If you use Matlab, run Matlab function *interp1* and compare the CPU time (*Matlab function cputime*). **Hint:** In Matlab, type *help interp1* and *help cputime* for the usage. If the CPU numbers are too small, *use format short e*.
- Plot the error plot. You should label, title all the plots. You can use Matlab command: *subplot* to put multiple plots into a single paper.
- Do the test for the following functions:
  (a) f(x) $= e^x$.
  (b) $f(x) = \cos(10\pi x)$.
  (c) $f(x) = \dfrac{1}{1 + 25x^2}$

You should keep you code for late use. Note that: you need to submit your results and analysis, selected plots, along with your homework. Your programming code should be submitted to Moodle.