Assigned: Friday, April 3, 2015
Due: Friday, April 10, 2015

MTH 371: Homework 9
Newton Interpolation

GENERAL HOMEWORK GUIDELINES:

- On the very first page of your homework, provide your name, date, and homework number.
- Homework will be graded in part on neatness, organization, and completeness of solutions. Multiple pages MUST BE STAPLED.
- Attach all Scilab code, output, and plots to the page immediately following each problem. Also, clearly indicate the problem they correspond to.

1. Consider the following table of $(x, f(x))$ point values.

x	8	9	10	11	12
$f(x)$	2	9	8	8	2

   (a) Create a Newton divided difference table for these points in the order given. List the x-values in the first column, and the 0th, 1st, 2nd, 3rd, and 4th divided differences in the next columns. Write down the Newton form interpolating polynomial and plot it in Scilab. Also plot these points to verify your work.

   (b) Using the table from part (a), write down the Newton form obtained by interpolating nodes in the order 8, 9, 10, 11, 12. Compare the final polynomal to that of (a).

   (c) Using the table from part (a), write down the Newton form obtained by interpolating nodes in the order 10, 11, 9, 12, 8. Compare the final polynomal to that of (a).

   (d) Why can't the table from (a) be used for the Newton form with order 8, 10, 9, 11, 12?

2. Write a Scilab function (`.sci` file) for polynomial interpolation using Newton form. The input for this function should be an array of x-values, `x`, with $(n+1)$ distinct points $x_0, x_1, x_2, \ldots, x_n$ and also an array of function values, `f`, with $(n+1)$ function values $f_0, f_1, f_2, \ldots, f_n$. Your function should return a Scilab polynomial `P` such that $P(x_i) = f(x_i)$ for $i = 0, 1, \ldots, n$.

```
P = NewtonInt(x,f)
```

   To test your fantastic new function, write a script (`.sce` file) which plots the Newton degree $n = 4$ interpolation polynomial using the `subplot` command for each of the following. Include the original function in each plot and use 200 x-values. You should use the `horner` command in Scilab to evaluate a polynomial at x-values. Note, this function should produce the exact same end result as the previous homework `LagrangeInt` function.

   (a) $f(x) = \sin(x),\ 0 \le x \le 2\pi$

   (b) $g(x) = \cos(x),\ 0 \le x \le 2\pi$

   (c) $h(x) = \ln(x),\ 0.5 \le x \le 2$

   (d) $i(x) = e^x,\ 0 \le x \le 2$

3. Prove the recursion formula for computing Newton divided differences.

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0}$$

   To do this, let P be the interpolating polynomial for $\{(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_{k-1}, f(x_{k-1}))\}$ and Q the interpolating polynomial for $\{(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_k, f(x_k))\}$ and consider the polynomial

$$R(x) = \frac{x_k - x}{x_k - x_0} P(x) + \frac{x - x_0}{x_k - x_0} Q(x).$$

   (a) Prove R is the unique polynomial of at most degree k which interpolates points $\{(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_k, f(x_k))\}$.

   (b) Determine the coefficient of x^k on each side of the equation.

4. The polynomial $p(x) = x^4 - x^3 + x^2 - x + 1$ has the values shown below.

x	-2	-1	0	1	2	3
$p(x)$	31	5	1	1	11	61

   Using as little work as possible find a polynomial q which has values

x	-2	-1	0	1	2	3
$q(x)$	31	5	1	1	11	30

   Hint: This can be done with little work! Note, the degree of p is 4.

5. Computer libraries often use tables of function values together with piecewise linear interpolation to evaluate elementary functions such as $\sin(x)$, because table lookup and interpolation can be faster than using a Taylor series expansion. Write one Scilab `.sci` file to accomplish the following.

   (a) Create a vector `x` of 1000 uniformly-spaced values between 0 and π. Then, create a vector `y` with the values of the sine function at each of these points. This will serve as your lookup table.

   (b) Next create a vector `r` of 100 randomly distributed values between 0 and π. (This can be done is Scilab by typing `r = %pi*rand(100,1)`). Estimate `sin(r)` as follows:

   For each value `r(j)`, find the two consecutive `x` entries, `x(i)` and `x(i+1)` which satisfy `x(i)` $\leq$ `r(j)` $\leq$ `x(i+1)`. Having identified the subinterval containing `r(j)`, use linear interpolation to estimate `sin(r(j))`.

   Compare your results with those returned by Scilab when you type `sin(r)`. Find the maximum absolute error and maximum relative error in your results to Scilabs results.