# Homework 2, Due: Thursday, 11/8

This assignment is due on **Thursday, November 8**, by 11:59 PM. Your assignment should be well-organized, typed (or neatly written and scanned) and saved as a .pdf for submission on Canvas. You must show all of your work to receive full credit. For problems requiring the use of MATLAB code, remember to also submit your .m-files on Canvas as a part of your completed assignment. Your code should be appropriately commented to receive full credit.

## Problems

$\boxed{1}$ Consider the following data:

$$f(0) = 1, \quad f(0.25) = 1.64872, \quad f(0.5) = 2.71828, \quad f(0.75) = 4.48169.$$

(a) (6 points) Use appropriate Lagrange interpolating polynomials of degrees 1, 2, and 3 to approximate $f(0.43)$.

(b) (6 points) The data were generated using $f(x) = e^{2x}$. Use the error formula to find a bound for the truncation error, and compare the error bound to the absolute error for the cases $n = 1$ and $n = 2$ (i.e., for the Lagrange polynomials of degrees 1 and 2).

(c) (6 points) Compute the entries of Newton's divided-difference table relating to the given data. Use the appropriate table entries to write the Lagrange interpolating polynomial of degree 3 in Newton divided-difference form, and show that it is equivalent to the degree 3 Lagrange interpolating polynomial found in part (a).

$\boxed{2}$ (4 points) Suppose $x_j = j$ for $j = 0, 1, 2, 3$, and it is known that

$$P_{0,1}(x) = x + 1, \quad P_{1,2}(x) = 3x - 1, \quad \text{and} \quad P_{1,2,3}(1.5) = 4.$$

Find $P_{0,1,2,3}(1.5)$.

$\boxed{3}$ Consider the MATLAB function `nevilles.m` posted on Canvas, which implements Neville's method for iterated interpolation (Algorithm 3.1, pg. 120).

(a) (4 points) Describe what is happening in each of the following lines of code (lines 17-27 in the corresponding .m-file):

```
N = length(xpts);
Q = zeros(N,N);
Q(:,1) = ypts;
```

```
% Recursively generate Lagrange interpolating polynomials
for i = 1:N-1
    for j = 1:i
        Q(i+1,j+1) = ((xeval-xpts(i-j+1))*Q(i+1,j)-(xeval-xpts(i+1))*Q(i,j))...
            /(xpts(i+1)-xpts(i-j+1));
    end
end
```

In particular, describe the indexing used in computing `Q` within the two `for` loops. How is this equivalent to the indexing used in the statement of the algorithm on pg. 120?

(b) (6 points) Modify `nevilles.m` to stop running when successive polynomials `Q(i,i)` and `Q(i-1,i-1)` differ in absolute value by less than $10^{-3}$ (instead of generating the full Neville table). Save your modified function as a new function called `nevilles_mod.m`, and provide a copy of your .m-file, describing your changes to the code. Compare the results of the Neville table that you get using your `nevilles_mod.m` to the full Neville table that you get using `nevilles.m` in approximating $f(1.5)$ with the following data:

```
  i   x_i   f(x_i)
-------------------------------
  0   1.0   0.765197686557967
  1   1.3   0.620085989561509
  2   1.6   0.455402167639381
  3   1.9   0.281818559374385
  4   2.2   0.110362266922174
```

$\boxed{4}$ (4 points) Show that the cubic polynomials

$$P(x) = 3 - 2(x+1) + 0(x+1)(x) + (x+1)(x)(x-1)$$

and

$$Q(x) = -1 + 4(x+2) - 3(x+2)(x+1) + (x+2)(x+1)(x)$$

both interpolate the data

| $x$ | $-2$ | $-1$ | $0$ | $1$ | $2$ |
|---|---|---|---|---|---|
| $f(x)$ | $-1$ | $3$ | $1$ | $-1$ | $3$ |

Why does this not violate the uniqueness property of interpolating polynomials? Explain.

**Note:** For any of the above problems for which you use MATLAB to help you solve, you must submit your code/.m-files as part of your work. Your code must run in order to receive full credit.