1. Construct in two ways the polynomial of degree 2 or less whose graph passes through the points $(0, 1.1), (1, 2)$, and $(2, 4.2)$. Verify that they are the same.

2. Simple polynomial interpolation in two dimensions is not always possible. For example, suppose that the following data are to be represented by a polynomial of first degree in x and in y, i.e. $p(x, y) = a + bx + cy$:

(x, y)	(1,1)	(3,2)	(5,3)
$f(x, y)$	3	2	6

   Show that it is not possible. Discuss why this may not work in general.

3. Reading MATLAB code:
   Consider the MATLAB M-file `polyinterp.m` found on the class website. Expose as clearly and carefully as possible what is going on in this routine. Explain precisely what is being done in each line of the code? (Be as verbose as you need to be.) What is being computed?

4. Write a MATLAB function `Newtinterp(x,y,t)` which returns the value of the Lagrange interpolating polynomial using the Newton formulation at the values in the vector `t`. The vector `x` contains the interpolation nodes, and `y` contains the corresponding function values.
   This routine should make a call to `InterpN(x,y)`, located on the course website, in order to compute the Newton divided difference coefficients. Having these coefficients, the routine should then compute the polynomial interpolant values at `t`.
   Note: we discussed in class how to write the polynomial in nested notation, thus reducing the number of computations. See the pdf file `div_diff.pdf`.

5. Consider polynomial interpolation of the Runge function $R(x) = 1/(1 + 20x^2)$ on $[-2, 2]$.

   (a) Plot the Runge function along with the polynomial interpolant (using `polyinterp.m` or `Newtinterp.m`) through n equally spaced nodes for varying values of n.

   (b) Use next the *Chebyshev interpolating nodes* which are given by

   $$x_i = \frac{1}{2}[(b-a)\cos\left(\frac{\pi(2i-1)}{2n}\right) + a + b], \quad i = 1, 2, \cdots, n,$$

   for interpolation on the interval $[a, b]$. Again use varying values of n

   Discuss and compare your results.
   Provide your commented code. Use *publish*.

6. (a) Use `polyinterp` or `Newtinterp.m` to do #5b,c in Section 3.1. This is the question that begins
   Use appropriate Lagrange interpolating polynomials of degrees one, two, $\cdots$
   b. $f(-1/3)$ *if* $f(-0.75) = \cdots$,
   c. $f(0.25)$ *if* $f(0.1) = \cdots$

(b) Do #9b,c in Section 3.1. This is the question that begins
The data for Exercise 5 were generated $\cdots$

Note 1: In class, our theorems on polynomial interpolation use nodes $x_1, \cdots, x_n$ yielding interpolating polynomials of degree $\leq n-1$, and an error formula involving $f^{(n)}$ and $n!$.
In the text, the authors use nodes $x_0, x_1, \cdots, x_n$ (i.e. $n+1$ points) yielding interpolating polynomials of degree $\leq n$, and an error formula involving $f^{(n+1)}$ and $(n+1)!$. Thus, when the book says $n = 1$, they really want you to use 2 nodes, etc.

Note 2: The answers in the back of the book have rounded their answers to about 5 decimal places. You do not need to do that; you may use the precision that Matlab allows you.

7. In class, I meantioned *Weierstrass Approximation Theorem*. What is that theorem? What does that have to do with what we have been doing?