COMPUTER SCIENCE 349A, SPRING 2018
ASSIGNMENT #5 - 20 MARKS

DUE MONDAY MARCH 19, 2018 (11:30 p.m. PST)

This is a really large class and the logistics of grading assignments are challenging. Me and the markers require your help in making this process go smoothly. Please ensure that your assignments conform to the following requirements - any violation will result in getting a zero for the particular assignment.

- All assignments should be submitted electronically through the ConneX course website and shoud be **SINGLE PDF FILES**. No other formats will be accepted. Handwritten answers are ok but they will need to be scanned and merged into a single pdf file together with any code examples and associated plots.
- The assignment number, student name and student number should be clearly visible on the top of every page of your assignment submission.
- **PLEASE DO NOT COPY THE ASSIGNMENT DESCRIPTION IN YOUR SUBMISSION**
- The asnwers to the questions should be in the same order as in the assignment specification.
- Some of the questions of the assignments are recycled from previous years but typically with small changes in either the description or the numbers. Any submission that contains numbers from previous years in any questions will be immediately graded with zero.
- Any assignment related email questions should have a subject line of the form CSC349A Assignment X, where X is the number of the corresponding assignment.
- The total number of points for this assignment is 20.

1. (a) **(2 points)** Specify a system of linear equations whose solution solves the following problem. Determine the coefficients of the polynomial $p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$ that interpolates $f(x) = \cos(\pi x)$ at the points $x = \left[\frac{\pi}{4}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}\right]$.

   Note: either write out this system by hand, using entries like $\left(\frac{\pi}{4}\right)^3$, or determine the entries of this linear system in MATLAB.

   (b) **(2 points)** Use the MATLAB \ operator to solve the above linear system for the coefficients of $p(x)$.

   (c) **(2 points)** Using *fplot* in MATLAB, plot (on one graph) the graphs of $f(x)$ and $p(x)$ on the interval $\left[\frac{\pi}{4}, \frac{5\pi}{6}\right]$.

   Hint: You will need to use the *hold on* command to get them both on the same graph with *fplot*.

2. **(8 points)** Consider the piecewise cubic polynomial

$$S(x) = \begin{cases} S_0(x) = a_0 + b_0 x + d_0 x^3, & \text{if } 0 \le x \le 1 \\ S_1(x) = a_1 + b_1(x-1) + c_1(x-1)^2 + d_1(x-1)^3, & \text{if } 1 \le x \le 3 \end{cases}$$

   Determine values for the 7 unknown coefficients above so that $S(x)$ is a cubic spline interpolant with natural boundary conditions for the following data:

x_i	$f(x_i)$
0	1
1	2
3	-20

   Specify all eight conditions that $S(x)$ must satisfy (these are conditions (b), (c), (d), (e), and (f) in Handout #24). Then solve these equations for the unknown coefficients. (Note: 2 of the variables should be easily determined. The values of the remaining 5 variables must be computed from a system of 5 linear equations in 5 unknowns. Specify this linear system, and then solve it by any means. For example, do it by hand, or use your calculator or MATLAB.)

3. Cubic spline interpolating functions can be computed in MATLAB. Many types of boundary conditions are possible, including the 'clamped' boundary conditions and the 'not-a-knot' boundary conditions. We will consider only the clamped boundary conditions. For the cubic spline with clamped boundary conditions, the data to be interpolated should be stored in vectors, say X (the x_i's) and Y (the $f(x_i)$'s), where Y has 2 more entries than X, the first and last entries of Y are the two clamped boundary conditions ($f'(x_0)$ and $f'(x_n)$), respectively. If $S(x)$ denotes the cubic spline interpolant, and z is a given number, then the value of $S(z)$ can be computed by entering

```
spline(X, Y, z)
```

Note that z can also be a vector of values at which you want to evaluate the spline (as in part (b) below).

If you want to actually determine the coefficients of the spline, you first must determine the `pp` (piecewise polynomial) form of the spline by entering

```
pp = spline(X, Y)
```

Some information (which you can ignore) about the `pp` form of the spline is given. Then enter

```
[b, c] = unmkpp( pp )
```

The values returned are:
`b` – a vector of the knots (or nodes) of the spline,
`c` – an array, the i-th row of which contains the coefficients of the i-th spline

Note: if the entries of X are denoted by $\{x_0, x_1, \ldots, x_n\}$ and the entries in the first row of c are $c(1,1)$, $c(1,2)$, $c(1,3)$, $c(1,4)$, then the first cubic polynomial of the spline is

$$(*) \qquad S_0(x) = c(1,1)(x - x_0)^3 + c(1,2)(x - x_0)^2 + c(1,3)(x - x_0) + c(1,4),$$

and similarly for the other cubic polynomials $S_1(x), \ldots, S_{n-1}(x)$.

(a) **(3 points)** Use MATLAB to determine the coefficients of the 8 cubic polynomials of the cubic spline interpolant with clamped boundary conditions for the following data:

i	x_i	$f(x_i)$	$f'(x_i)$
0	1	3	1
1	2	3.7	
2	5	3.9	
3	6	4.2	
4	7	5.7	
5	8	6.6	
6	10	7.1	
7	13	6.7	
8	17	4.5	-0.67

Use `format short` to display your output.

DELIVERABLES: The commands and the results from MATLAB plus the final piecewise polynomial with coefficients.

(b) **(3 points)** Use the MATLAB function *plot* to draw a graph of the spline constructed in (a) on the interval [1, 17]. The 8 cubic polynomials should be drawn on one graph in the same graphics window. The first cubic polynomial should be drawn with a solid line on [1, 2], the second with a dotted line on [2, 5], the third with a solid line on [5, 6], and so on. As $x_0 = 1$ in this problem, the cubic polynomial above can be plotted on [1, 2] as a solid line with the following statements:

```
X1=linspace(1,2,50);
Y1=c(1,1)*(X1-1).^3+c(1,2)*(X1-1).^2+c(1,3)*(X1-1)+c(1,4);
plot(X1,Y1,'-')
```

NOTE that ^ must be replaced by the MATLAB operator `.^` for the cubed and squared powers in this expression because the argument X1 is a vector (rather than a scalar). This means that the exponentiation is done component wise to the entries in the vector X1. Also, the third input value, `'-'`, specifies that the function is plotted as a solid line. If `'-'` is replaced by `':'` then the graph is drawn with a dotted line.

DELIVERABLES: The commands used to create the plot and the plot itself.