COMPUTER SCIENCE 349A, FALL 2018
ASSIGNMENT #3 - 20 MARKS

DUE FRIDAY NOVEMBER 16, 2018 (11:55 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. Consider the function $f(x) = \sin^2(x)$ in the interval $[0, 2\pi]$. You are given the following 4 points of this function:

x_i	$f(x_i)$
0	0
$\frac{2\pi}{3}$	0.75
$\frac{4\pi}{3}$	0.75
2π	0

   (a) **(4 points)** Calculate the cubic Lagrange interpolating polynomial as the sum of the $L_0(x)f(x_0), L_1(x)f(x_1), L_2(x)f(x_2), L_3(x)f(x_3)$ polynomials we defined in class. The final answer should be in the form $P(x) = ax^3 + bx^2 + cx + d$, but with a, b, c, d known. [Note: if any of the coefficients are 0 you do not need to include the term in the final polynomial.]

   DELIVERABLES: All your work in constructing the polynomial. This is to be done by hand not MATLAB.

   (b) **(2 points)** Plot $f(x) = \sin^2(x)$ in the interval $[0, 2\pi]$ by creating vectors

```
x = [0:0.1:2*pi];
y =sin(x).^2;
```

   On the same graph plot the interpolating polynomial $P(x)$ from part (a).

   DELIVERABLES: The commands and the resulting plot from MATLAB.

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

$$S(x) = \begin{cases} S_0(x), & \text{if } 0 \le x \le \frac{2\pi}{3} \\ S_1(x), & \text{if } \frac{2\pi}{3} \le x \le \frac{4\pi}{3} \\ S_2(x), & \text{if } \frac{4\pi}{3} \le x \le 2\pi \end{cases}$$

where

$$\begin{aligned} S_0(x) &= a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3 \\ S_1(x) &= a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3 \\ S_2(x) &= a_2 + b_2(x - x_2) + c_2(x - x_2)^2 + d_2(x - x_2)^3 \end{aligned}$$

Using the data points given in question 1., specify all twelve conditions that $S(x)$ must satisfy in order for it to be a *clamped*, cubic spline interpolant for $f(x) = \sin^2(x)$. Do not simplify the equations, just move all terms with unknown coefficients to the left of the equation and the known constants to the right. Also, do not evaluate the π's, leave them as π in your equations. Set up the augmented matrix that needs to be solved to get the spline. **DO NOT SOLVE THE SYSTEM**.

DELIVERABLES: Show all your work ending with the 12 unsolved equations in the augmented matrix form.

3. Cubic spline interpolating functions can be computed in MATLAB. Many types of boundary conditions are possible, including the 'clamped' boundary conditions and the 'second' 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 3 cubic polynomials of the cubic spline interpolant with clamped boundary conditions for the data points given in questions 1. and 2.:

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)** In its simplest form, `PLOT(X,Y,'-')` plots a vector of values `Y` versus a vector of values `X` using a solid line. If `'-'` is replaced by `':'` then the graph is drawn with a dotted line. See **`help plot`** for other options. The graph is drawn by connecting the points in X by a straight line, so if the points in X are quite dense, then the graph will appear as a smooth continuous curve. For example, the statements

```
X=linspace(0,pi,100);
Y=sin(X);
plot(X,Y,'-')
```

will open a graphics window with the graph of $\sin x$ plotted on the interval $[0, \pi]$ using 100 equally spaced data points. The graph of $\cos x$ could be drawn with a dotted line on the same graph as $\sin x$ on $[0, \pi]$, by executing the following statements:

```
X1=linspace(0,pi,100);
Y1=sin(X1);
Y2=cos(X1);
plot(X1,Y1,'-',X1,Y2,':')
```

The functions $\sin x$ and $\cos x$ could be drawn on the same graph but on different intervals by specifying a different set of X values for each.

Use the MATLAB function plot to draw a graph of the spline constructed in (a) on the interval $[0, 2\pi]$ along with the function $f(x) = \sin^2(x)$. So, $f(x)$ and the 3 cubic polynomials should be drawn on one graph in the same graphics window. The first cubic polynomial should be drawn with a dotted line on $[0, \frac{2\pi}{3}]$, the second with a solid line on $[\frac{2\pi}{3}, \frac{4\pi}{3}]$, and the third with a dotted line on $[\frac{4\pi}{3}, 2\pi]$. As $x_0 = 0$ in this problem, the cubic polynomial in (*) above could be plotted on $[0, \frac{2\pi}{3}]$ with the following statements:

```
X1=linspace(0, 2*pi/3,50);
Y1=c(1,1)*(X1-0).^3+c(1,2)*(X1-0).^2+c(1,3)*(X1-0)+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; for example, if `X = [1 2 3 4]`, then `X.^2` is equal to `[1 4 9 16]`, whereas `X^2` is undefined.

DELIVERABLES: The statements you use to create the data to plot the original function and the spline function along with the plot itself.