CS 322: Prelim 1 Solution Guide

95-100	
90-94	xxxxxxxxxx
85-89	xxxxxxxxxxxx
80-84	xxxxxxxxxxxxxx
75-79	xxxxxxxxxxxxxxxxxxxxxxxxxxx
70-74	xxxxxxxxxxxxxxxxxxxxxxx
65-69	xxxxxxxxxxxxxxxxxxxxxxxxxxx
60-64	xxxxxxxxxxxxxxxxxxxxxxxxxxx
55-59	xxxxxxxxxxxxxxxx
50-54	xxxxxxxxxx
45-49	xxx
40-44	xxxxxxxx
< 40	xxx

Median = 69. Rough Grade Guidelines: A = [80,100], B = [65,75], C = [50,60]

---

1. Assume that `x` and `y` are given column vectors and that the command `plot(x,y)` results in the display of the first-quadrant portion of the ellipse

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1.$$

Without any sine, cosine, or, square root computations, show how to plot the entire ellipse

$$\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-k}{b}\right)^2 = 1$$

assuming that `h`, `k`, `x`, and `y` are given. Do not worry about axis scaling. You may use `hold`. Express your solution in the form of a MATLAB script.

```
plot(x+h,y+k)

hold on
plot(-x+h,y+k)
plot(-x+h,-y+k)
plot(x+h,-y+k)
hold off
```

Or:

```
x = [x;-x;-x;x];
y = [y;y;-y;-y];
%plot(x,y) now would do the ellipse centered at (0,0)
plot(x+h,y+k)
```

-5 if plot with -h and -k
-3 if plot with some +h (or k) and some -h (or k)
-2 if hold missing
-10 if plot correctly at (0,0)
-3 if fail to plot one quadrant

2. **(a)** (10 points) Assume that $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n\times n}$ are stored in n-by-n arrays `A` and `C` and that $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ are stored in n-by-1 arrays `x` and `y`. Write a MATLAB 1-liner that efficiently evaluates $z = xy^T AC$ and assigns the result to `z`.

```
% There are three operations and 5 possible algorithms:

z = ((x*y')*A)*C      % This is O(n^3) and the same as z = x*y'*A*C      4 points for this
z = (x*y')*(A*C)      % This is O(n^3)                                    4 points for this
z = (x*(y'*A))*C      % This is O(n^3)                                    7 points for this
z = x*((y'*A)*C)      % This is O(n^2)                                   10 points for this
z = x*(y'*(A*C))      % This is O(n^3)                                    4 points for this
```

If the wrong answer was calculated (i.e. `(A*x)'(C*y)` calculation), then 2 further points were deducted. No points were deducted for failing to use correct Matlab syntax.

(b) (10 points) Suppose we are given the points $(x_1, y_1), \ldots, (x_n, y_n)$ with $n \geq 3$ and $x_1, \ldots, x_n$ distinct. Assume that q and r are degree $n-2$ polynomials with the property that

$$\begin{array}{ll} q(x_i) = y_i & \quad i = 1{:}n-1 \\ r(x_i) = y_i & \quad i = 2{:}n \end{array}$$

Determine real numbers a and b so that $p(x) = a(x - x_n)q(x) + b(x - x_1)r(x)$ interpolates $(x_1, y_1), \ldots, (x_n, y_n)$. It must be clear from your answer that p interpolates all n points.

Since $y_1 = p(x_1) = a(x_1 - x_n)q(x_1) = a(x_1 - x_n)y_1$ we must have $a = 1/(x_1 - x_n)$.

Since $y_n = p(x_n) = b(x_n - x_1)r(x_1) = b(x_n - x_1)y_n$ we must have $b = 1/(x_n - x_1)$. Thus,

$$p(x) = \frac{(x - x_n)q(x) - (x - x_1)r(x)}{x_1 - x_n}$$

For $i = 2{:}n-1$ we have

$$p(x_i) = \frac{(x_i - x_n)q(x_i) - (x_i - x_1)r(x_i)}{x_1 - x_n} = y_i \frac{(x_i - x_n) - (x_i - x_1)}{x_1 - x_n} = y_i$$

3 points for getting the correct answer for $a (= 1/(x_1 - x_n))$
3 points for getting the correct answer for $b (= 1/(x_n - x_1))$
3 points for showing $p(x_i) = y_i$
1 point for correctly stating the limitations on i when showing $p(x_i) = y_i$ (i.e. the algebra only holds for $i = 2{:}n-1$)
9 points maximum were given for a solution where a system of linear equations were presented. To get all 9 points, the student had to present the correct system of linear equations, and actually give some of the matrix equations

---

3. **(a)** (10 points) Assume that x, y, and z are consecutive positive floating point numbers with $x < y < z$. If $(x + z)/2 = y + 1/64$ in exact arithmetic then what is the value of $y - x$ in exact arithmetic? Hint: the spacing between x and y is not the same as the spacing between y and z.

Assume that $y = x + d$, i.e., d is the spacing between x and y. Since the average of x and z is bigger than y, we must have $z = y + 2d = x + 3d$. Thus, $y + 1/64 = (x + z)/2 = x + (3/2)d = y + d/2$. Thus, $d = 1/32 = y - x$.

3 points for recognizing that spacing between y and z is twice the spacing between x and y.

(b) (10 points) $S(x)$ is the not-a-knot cubic spline interpolant of the points $(x_1, y_1), \ldots, (x_4, y_4)$ with $x_1 < x_2 < x_3 < x_4$. Assume that

$$S(x) = \begin{cases} p_1(x) = c_{11}(x - x_1)^3 + c_{12}(x - x_1)^2 + c_{13}(x - x_1) + c_{14} & \text{if } x_1 \le x \le x_2 \\ p_2(x) = c_{21}(x - x_2)^3 + c_{22}(x - x_2)^2 + c_{23}(x - x_2) + c_{24} & \text{if } x_2 \le x \le x_3 \\ p_3(x) = c_{31}(x - x_3)^3 + c_{32}(x - x_3)^2 + c_{33}(x - x_3) + c_{34} & \text{if } x_3 \le x \le x_4 \end{cases}$$

Explain why $c_{11} = c_{21} = c_{31}$.

For the not-a-knot spline, $p_1'''(x_2) = p_2'''(x_2)$ and so $6c_1 = 6c_2$. Likewise, $p_2'''(x_3) = p_3'''(x_3)$ and so $6c_2 = 6c_3$.

-2 if you don't correctly differentiate the cubics three times
-1 confusing f"' and s"'
3 pts for understanding that s, s' and s" are all continuous.

---

4. Suppose $f(x)$ is always positive with the property that $f(-x) = f(x)$ for all x. How would you compute

$$I = \int_{-a}^{b} f(x)dx \qquad 0 < a < b$$

with relative error 10^{-6} or less? Make effective use of `QUAD`. Assume that the function f is implemented in `f.m` and recall that if `numI = QUAD('f',L,R,tol)`, then we (usually) have

$$\left| \texttt{numI} - \int_L^R f(x)dx \right| \le \texttt{tol} \cdot \int_L^R f(x)dx.$$

express your answer in the form of a MATLAB script and explain why your choice of the tolerances works.

Since

$$I = \int_{-a}^{b} f(x)dx = \int_{-a}^{0} f(x)dx + \int_{0}^{a} f(x)dx + \int_{a}^{b} f(x)dx = 2\int_{0}^{a} f(x)dx + \int_{a}^{b} f(x)dx$$

```
numI = 2*QUAD('f',0,a,tol1) + QUAD('f',a,b,tol2)
```

$$\begin{aligned} |I - \texttt{numI}| &= \left| \left(2\int_0^a f(x)dx - 2\texttt{QUAD('F',0,a,tol1)} \right) + \left(\int_a^b f(x)dx - \texttt{QUAD('F',a,b,tol2)} \right) \right| \\ &\le 2\texttt{tol1} \int_0^a f(x)dx + \texttt{tol2} \int_a^b f(x)dx \end{aligned}$$

Note that by setting `tol1 = tol2 = tol` $= 10^{-8}$ we obtain the required result:

$$|I - \texttt{numI}| \le \texttt{tol} \int_{-a}^{b} f(x)dx$$

matlab script:
+14 for numI = 2*quad('f',0,a,tol1) + quad('f',a,b,tol2)
+12 if use min/max to determine if a¡b or b¡a (given in problem statement) and everything else with above line is correct

explanation:
+3 correct argument
+3 correct reasoning

---

5. The 2-point Gauss-Legendre rule is given by

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\left(f(x_1) + f(x_2)\right)$$

where

$$\begin{aligned} x_1 &= \frac{a+b}{2} - \frac{b-a}{2\sqrt{3}} \\ x_2 &= \frac{a+b}{2} + \frac{b-a}{2\sqrt{3}} \end{aligned}$$

Complete the following function so that it performs as specified

```
   function alfa = SectorArea(A,P,theta)
% A and P are scalars with A >= P > 0 and theta is a column n-vector
% with n > 2 and theta(1) < theta(2) < ... < theta(n).
%
% alfa is a column (n-1)-vector with the property that for k=1:n-1
% alfa(k) is the 2-point Gauss-Legendre approximation of the integral
% of the function f(x) from x = theta(k) to x = theta(k+1) where
% f(x)= .5( A*P/( P*(1 - cos(x)) + A*(1 + cos(x)) ) )^2.
```

Vectorize and avoid redundant computation.

```
u = (theta(1:n-1) + theta(2:n))/2;                     % 4pts
v = (theta(2:n) - theta(1:n-1))/(2*sqrt(3));           % 4 pts

x1 = u - v; f1 = .5*(A*P./(P+A+(A-P)*cos(x1))).^2;     % 3 pts
x2 = u + v; f2 = .5*(A*P./(P+A+(A-P)*cos(x2))).^2;     % 3 pts

alfa = (theta(2:n) - theta(1:n-1)).*(f1 + f2)/2        % 6pts
```

It is possible to combine the u and v steps and the f1 and f2 steps, but not required for full credit