Math 4513, Homework 4, Due on 11/9/2012

1. (10 points) A car traveling along a straight road is clocked at a number of points. The data from the observations are given in the following table, where the time is in seconds, the distance is in feet, and the speed is in feet per second.

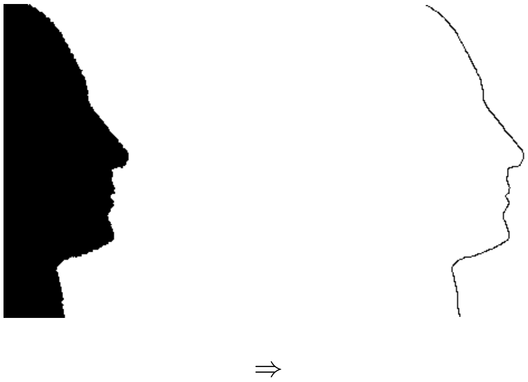
Time	0	3	5	8	13
Distance	0	225	383	623	993
Speed	75	77	80	74	72

   (a) Use a Hermite polynomial to predict the position of the car and its speed when $t = 10$.

   (b) Use the derivative of the Hermite polynomial to determing whether the car ever exceeds a 55 mi/h speed limit on the road. If so, what is the first time the car exceeds this speed?

   (c) What is the predicted maximum speed for the car?

2. (10 points) We can use *edge* function in Matlab to extract the edge of a given picture, as shown below.



What you need to do the regenerate the above graphs is to download the file "silhouette.gif" (the picture on the left) from our class webpage, and then run the following commands:

```
A = imread('silhouette.gif');
A(A>0) = 255;
figure(1), imshow(A);
BW = 1-edge(A,'canny');
BW = BW(2:(end-1),:);
figure(2), imshow(BW)
```

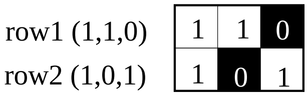
The "edge" (the picture on the right) is now stored in a Matlab matrix "BW". Each entry of "BW" has value either equal to 0 or 1, where 0 corresponds to a black pixel and 1 correspond to a white pixel. We need to briefly discuss how can such a matrix represent a picture. Let us first look at a small 2×3 example:

$$B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Run the following matlab command to see the graph (it might be very tiny on your screen):

```
  B = [1, 1, 0; 1, 0, 1];
  imshow(B)
```

Matlab draw the given matrix B in the following way:



row1 (1,1,0)	1	1	0
row2 (1,0,1)	1	0	1

The following commands and outputs

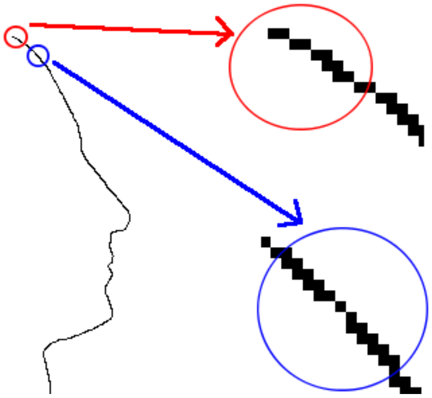
```
>> find(B(1,:)==0)
ans =
     3
>> find(B(2,:)==0)
ans =
     2
```

basically says that row 1 of matrix B has one entry equal to 0 (black pixel), which is the 3rd entry. Row 2 of matrix B has one entry equal to 0 (black pixel), which is the 2nd entry.

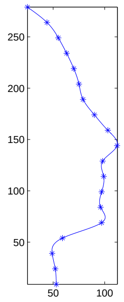
Matrix BW follows the same rule, except that its size is much larger, 279×120. To find how many black pixel does a row contain, you can similarly do

```
>> find(BW(5,:)==0)
ans =
    31    32
>> find(BW(30,:)==0)
ans =
    54
```

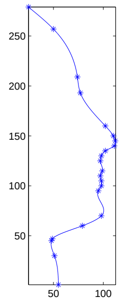
This tells us row 5 contains two black pixels, with column indices 31 and 32, while row 30 contains only 1 black pixel at column 54. If you magnify the picture, you will see exactly what this means.



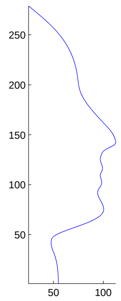
We are interested in reproduce the human face silhouette using a cubic spline. By taking sampling points on the "edge", one can create a natural cubic spline like the following (* stands for sampling points):



Or if you want to show more details on certain parts, try taking unevenly-spaced sampling points, as shown in the following:



Erase these sampling points, you will see a nice, smooth silhouette:



Use the provided image file “silhouette.gif”, generate the “edge” and take a few sampling points (≤ 30), using the coordinates of some black pixels in the “edge” graph. Then compute a natural cubic spline using these sampling points and draw a graph.

Remark, you might find the following two commands useful when drawing the graph:

```
>>axis equal
>>axis tight
```