EECS 639: Homework 4
Due: Friday, October 21, 2016 (at 2pm)

No late homework will be accepted so that solutions can be handed out the same
day in preparation for your exam.

1. (20 points) In class, we learned that the SVD can be used as a helpful tool in image processing. Here
we will explore the use of the SVD in image compression.

Recall that if $A = U \Sigma V^T$ is the SVD of $A$, then $\sigma_1 E_1 + \ldots + \sigma_k E_k$ is the best rank-$k$
approximation to $A$, where $E_i = u_i v_i^T$. We can use this idea to represent an image as a compressed image by using a
lower-rank approximation to the matrix which represents the original image.

For this problem, you should first download beach.gif from the class website. Then, use the Matlab
function “imread” to read-in the image and store it in a matrix $A$. Next, use the grayscale map given
below, and plot the image using the “image” command.

**Note:** To get a properly scaled grayscale colormap, use the following lines of code:

```matlab
maxrange=255; % Grey levels are 0..255 for 8 bit
g=[0:maxrange]; % Vector containing grey levels
g=(1/maxrange)*g; % Scale to 0..1
graymap=[g' g' g'];
colormap(graymap);
```

Precede the image command by this line: “subplot(2,2,1)” so that the plot appears in the upper
left-hand corner of the figure. This shows you what the original image looks like.

Now you are ready to begin the image compression phase. Start by computing the SVD of $A$. [In order
to use the built-in SVD command, you first need to convert $A$ to a matrix of doubles. To do this, type
$A = double(A)$. Then, type $[U,S,V] = svd(A)$; This gives you the matrices $U$, $\Sigma$, and $V$.]

For our first compressed image, create a matrix $A_2$ which is the best rank-$\dim/2$ approximation to
$A$, where $\dim = \min\{m,n\}$. Use the above formula to do this. Using the image command, plot the
compressed image corresponding to $A_2$ in the upper right-hand corner of the figure window by preceding
the plot command with subplot(2,2,2). (Be sure to set the colormap to gray before plotting.)

For the second and third compressed images, create matrices $A_4$ and $A_{10}$ which are the best rank-$\dim/4$
and rank-$\dim/10$ approximations to $A$. Plot the corresponding compressed images in the bottom half
of the figure using subplot(2,2,3) and subplot(2,2,4).

You should see the quality of the images decrease as the rank of the approximation to $A$ decreases.
This is to be expected as less information is being used to generate each image. We can quantify the
quality of each image using the **Peak Signal to Noise Ratio** (PSNR) which is defined as follows:

$$PSNR = 10 \log_{10} \left( \frac{(\text{max range})^2}{RMSE} \right),$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - \hat{A}_{ij})^2}{mn}}.$$

Here $A$ and $\hat{A}$ denote the matrices corresponding to the original and approximate images, and max range
denotes the maximum allowed range in the pixel values.

Compute the quality of each of the three approximate images above. Briefly (in 2-3 sentences) sum-
marize what you’ve learned from this problem.
2. (15 points) Let $A$ be an $m \times n$ matrix, $b$ be an $m \times 1$ vector, and consider solving $Ax \approx b$ for $x$.

(a) (5 points) Suppose $m = 10n$. Would it be better, in terms of computational complexity, to use the Method of Normal Equations or the Householder QR method to solve the problem?

(b) (5 points) Suppose $\text{cond}(A) = 10,000$. Would it be better, in terms of conditioning/sensitivity, to use the Method of Normal Equations or the Householder QR method to solve the problem?

(c) (5 points) Give an example of an $m \times n$ matrix $A$ for which you would not want to use either the Method of Normal Equations or the Householder QR method to solve the corresponding linear least squares problem. Explain why.

3. (10 points) Prove that $x^2 - 4 \sin(x) = 0$ has a solution in $[-1, 1]$.

4. (10 points) Determine the multiplicity of each root of the following equation:

$$x^4 + 3x^3 - 25x^2 - 39x + 180 = 0.$$  

**Hint:** To gain intuition about where the roots lie, make a plot of the function using Matlab. Then factor the expression on the left-hand side of the equation into a more user-friendly form.

5. (20 points)

From which rootfinding code (i.e., bisection or Newton’s method) is the following output? (Note that the bisection method has a linear rate of convergence, whereas Newton’s method typically has a quadratic rate of convergence.) Use the formula on p. 223 in Heath to determine this. Show your work in determining $r$ and $C$ and explain your choice of method.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.900000000000000e+00</td>
</tr>
<tr>
<td>2</td>
<td>1.421775761503564e+00</td>
</tr>
<tr>
<td>3</td>
<td>1.230515018797e+00</td>
</tr>
<tr>
<td>4</td>
<td>1.1997210140281e+00</td>
</tr>
<tr>
<td>5</td>
<td>1.198692214383051e+00</td>
</tr>
<tr>
<td>6</td>
<td>1.198691243516861e+00</td>
</tr>
<tr>
<td>7</td>
<td>1.198691243515997e+00</td>
</tr>
</tbody>
</table>

Table 1: Output of Rootfinding Code