Matlab Exercises with the SVD

1. In $\mathbb{R}^3$, project 300 points in $\mathbb{R}^3$ (stored in the 300×3 array named B) to the subspace given as the the plane $x - y - z = 0$. Plot the coordinates of the projection in $\mathbb{R}^2$ and then plot both the original data and the projected data on the plane in $\mathbb{R}^3$. Here are some notes:

   - On the class website, download the Matlab data file `SphereData.mat` and put it in the directory you are currently working in.

     To get the data into Matlab, in the command window type: `load SphereData`

     You should see the variable X in the Workspace window, where X is 300×3.
   - To plot the data you've loaded, type

     ```
     plot3(X(:,1),X(:,2),X(:,3),'k-');
     ```

     into the command window.
   - Once you have your projected data in a 300×3 array Y, you can plot it as well, keeping the original data:

     ```
     hold on;  % This keeps all the old data in the figure
     plot3(Y(:,1),Y(:,2),Y(:,3),'r-');
     hold off;
     ```

2. As we saw in class, polynomial interpolation (using a degree n polynomial with $n+1$ points) can lead to a high degree of oscillations between data points. If there is error in the data, then perhaps we can make a trade-off between a smoother, well-behaved function, but one that does not exactly hit the data points (like the best fitting line we have constructed in the past).

   We can do this using the SVD. Recall that, if we're solving the matrix equation:

   $$A\mathbf{c} = \mathbf{y}$$

   for the parameters in $\mathbf{c}$, we can use the pseudoinverse with smaller rank to provide an approximate solution. Recall that, if `[U,S,V]=svd(A,'econ')` is the SVD of A, then we can look at the singular values: `diag(S)` to see how many large values we have. Once we determine the rank r, then we can compute the pseudoinverse using that rank:

   ```
   [U,S,V]=svd(A);
   plot(diag(S))
   c=V(:,1:k)*S(1:k,1:k)*U(:,1:k)'*y;
   ```

   In the Matlab code provided, we will also use `polyval` to help us evaluate polynomials. In Matlab, if you have some data in a vector x and a polynomial defined by a coefficient vector $\mathbf{c}$ so that

   $$c_1 t^n + c_2 t^{n-1} + \cdots + c_n t + c_{n+1}$$

In Matlab, you can type: `polyval(c,x)`

On the class website, download the file `VanderExample.m`. Read it over to see if you understand it, then run it in Matlab.

3. Be sure you read over and understand the Matlab commands in the `SampleAuthor.m` code. Especially the following:

   - We compute the mean and store it in the vector `Ymean`. Is this a vector or a matrix? What is its size?
   - What does the command `repmat` do? (We've seen it a couple times before).
   - When we project data to a plane, that plane is isomorphic to $\mathbb{R}^2$- What is the isormorphism? Do we compute these coordinates somewhere in the code? Are they plotted somewhere?
   - When we represent the data in the original plane, that space is a subspace of $\mathbb{R}^{19200}$. This is also called the "reconstruction" of the data back to its original subspace. Where is this done in the code?

4. Here is the code that solves Exercise 1. There is an error in one of the lines (not a Matlab error, but a linear algebra error). See if you can find it.

```
load SphereData;
[U,S,V]=svd([1 -1 -1])
%Check that V has orthonormal columns.
V'*V
Coords=V(:,2:3)'*X';
figure(1)
plot(Coords(1,:),Coords(2,:),'.');

Y=V(:,1:2)*V(:,1:2)'*X';
Y=Y';  %This puts Y as 300 x 3 (instead of 3 x 300)
figure(2)
plot3(X(:,1),X(:,2),X(:,3),'k-');
hold on;  % This keeps all the old data in the figure
plot3(Y(:,1),Y(:,2),Y(:,3),'r-');
hold off
```