Please provide complete and well-written solutions to the following exercises.

Due October 20, 1159PM PST, to be uploaded as a single PDF document to blackboard (under the Assignments tab).

Homework 7

Exercise 1. Consider the following symmetric real matrix

$$A = \begin{pmatrix} 5 & 1 & -2 & 3 & 1 \\ 1 & 3 & 6 & 0 & 0 \\ -2 & 6 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 & 3 \end{pmatrix}.$$

Using the QR algorithm, find all eigenvalues and eigenvectors of A.

Finally, compare your results with the built-in Matlab function `eigs`.

Exercise 2. Let A be an $m \times n$ real matrix with $m \geq n$. Show that A has rank n if and only if $A^T A$ is positive definite.

(Hint: $A^T A$ is always positive semidefinite.)

Exercise 3. Suppose we have data points $(0,1), (1,3), (2,3), (3,5), (4,2) \in \mathbf{R}^2$ denoted as $\{(a_i, b_i)\}_{i=1}^5$. Find the line that best fits the data. That is, find the line $f: \mathbf{R} \to \mathbf{R}$ that minimizes the sum of squared differences $\sum_{j=1}^5 |f(a_i) - b_i|^2$. When you solve the equation $A^T A x = A^T b$, you can use Matlab's built-in matrix decomposition functions, such as the QR decomposition.

Then, find the best fit degree two polynomial, and the best fit degree three polynomial to these data points.

Exercise 4. Sometimes it is more convenient to use randomness to minimize a sum of squares. Suppose we have an $m \times n$ real matrix A, and $b \in \mathbf{R}^m$ are given. We want to find $x \in \mathbf{R}^n$ minimizing $||Ax - b||^2$. We try to do this in following program:

```
[m n]=size(A);
x=zeros(n,1);
for i=1:1000
    xnew=x+(1/100)*(rand(n,1) -1/2);  %randomly perturb x
    % if an improvement occurs, update x, otherwise don't
    if norm(A*xnew -b) < norm(A*x -b)
        x=xnew;
    end
end
```

Use the above program when

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Estimate the absolute error of the output x compared with the explicit formula $(A^TA)^{-1}A^Tb$.

Finally, use the above program with the A, b from the previous exercise (when we tried to fit a line to the data, i.e. the degree one case of the previous exercise). How far are the different outputs from each other?

Exercise 5.

- Give an example of a real 2×2 matrix A with a non-unique singular value decomposition. That is, find a diagonal 2×2 matrix D, unitary 2×2 matrices $U_1 \neq U_2$, unitary 2×2 matrices $V_1 \neq V_2$ such that
$$A = U_1DV_1 = U_2DV_2.$$
- Give an example of a real 2×3 matrix A with a non-unique singular value decomposition. That is, find a diagonal 2×2 matrix D, unitary 2×2 matrices $U_1 \neq U_2$, unitary 3×3 matrices $V_1 \neq V_2$ such that
$$A = U_1(D,\, 0)V_1 = U_2(D,\, 0)V_2.$$

Exercise 6. This exercise investigates the condition number of a matrix and its relation to solving linear systems. In a previous homework, we considered solving the linear system of equations

$$17x_1 + 5x_2 = 22,$$
$$1.7x_1 + .5x_2 = 2.2.$$

We then rewrite this equation as $Ax = b$ with $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $A = \begin{pmatrix} 17 & 5 \\ 1.7 & .5 \end{pmatrix}$. Even though A has determinant zero, Matlab computes the determinant of A to be quite large and nonzero. Also, Matlab struggles to solve the equation $Ax = b$ with the built-in solver `x=A\b`. However, the determinant itself is not a good way to measure the numerical errors with solving $Ax = b$, since multiplying A by a constant changes the determinant. A more standard way to measure how "badly behaved" a matrix is for solving linear systems is the **condition number** of an invertible matrix A, defined to be

$$\kappa(A) := ||A||_{2\to 2} \left|\left|A^{-1}\right|\right|_{2\to 2}.$$

Or, if A is not invertible, we define $\kappa(A)$ to be infinity.

- Show that if $t \neq 0, t \in \mathbf{R}$ and if A is a real $n \times n$ matrix, then
$$\kappa(tA) = \kappa(A).$$
That is, multiplying A by a nonzero number does not change the condition number of A (unlike the determinant).
- Show that if A, B are real $n \times n$ matrices, then
$$||AB||_{2\to 2} \leq ||A||_{2\to 2} ||B||_{2\to 2}.$$

- Show that if A is a real $n \times n$ matrix, then
$$\kappa(A) \geq 1.$$
- When $A = \begin{pmatrix} 17 & 5 \\ 1.7 & .5 \end{pmatrix}$, what does Matlab compute the condition number of A to be? (You can use the built-in command `cond` to answer this question.) A large condition number corresponds to more difficulty in solving $Ax = b$.
- Suppose we would like to solve $Ax = b$ where $b \in \mathbf{R}^n$ is given and $x \in \mathbf{R}^n$ is unknown. Suppose we compute $x_0 \in \mathbf{R}^n$ as an approximate solution to this equation. Define
$$r := b - Ax_0 = A(x - x_0).$$
Show that $||x - x_0|| \leq ||A^{-1}||_{2\to 2}\, ||r||$. Conclude that the relative error satisfies
$$\frac{||x - x_0||}{||x||} \leq \kappa(A) \frac{||r||}{||b||}.$$
(Assume that $x, b \neq 0$.) That is, the relative error for x is bounded by the condition number, multiplied by the relative error for b.

Exercise 7. Write a Matlab program that computes the singular value decomposition of an arbitrary $m \times n$ real matrix, without using Matlab's built-in program for the same task. (You are allowed to use Matlab's built in QR decomposition.) Then, apply your program to the following matrix
$$A = \begin{pmatrix} 5 & 1 & -2 \\ 1 & 3 & 6 \end{pmatrix}.$$
Finally, compare your result with the built-in Matlab function `svd`.

Exercise 8. In a previous exercise, we used nonnegative matrix factorization (NMF) to better understand the matrix from the table below, and we found that we could group the different foods into different categories, based on the NMF.

	apple	banana	bell pepper	crab	broccoli	carrot	pear	shrimp
calories	130	110	25	100	45	30	100	100
sodium	0	0	40	330	80	60	0	240
potassium	260	450	220	300	460	250	190	220
carbohydrates	34	30	6	0	8	7	26	0
vitamin A	2	2	4	0	6	110	0	4
vitamin C	8	15	190	4	220	10	10	4

It turns out that we can use the singular value decomposition of the matrix A to do something similar. The built-in Matlab function `[U S V]=svd(A)` produces a singular value decomposition of A of the form $A = USV^T$. The first 3 columns of V should correspond to the 3 largest singular values of A. The first row of V corresponds to apple, the second row of V corresponds to banana, and so on. We can plot all of the coordinates of V in $\mathbf{R}^3$ with the command

```
A=[130 110 25 100 45 30 100 100; 0 0 40 330 80 60 0 240;
260 450 220 300 460 250 190 220; 34 30 6 0 8 7 26 0;
2 2 4 0 6 110 0 4; 8 15 190 4 220 10 10 4];
[U S V]=svd(A);
```

```
plot3(V(:,1),V(:,2),V(:,3),'ro')
labels={'apple', 'banana', 'bell pepper', 'crab',...
 'broccoli', 'carrot', 'pear', 'shrimp'};
text(V(:,1),V(:,2),V(:,3),labels,'VerticalAlignment','bottom', ...
'HorizontalAlignment','right')
xlabel('V column 1');
ylabel('V column 2');
zlabel('V column 3');
```

In this plot, do the food categories look like they are grouped together in a similar way to what we found with NMF?

This process of examining a small number of the largest singular values is called **Principal Component Analysis** (PCA).

Exercise 9. The nodes $\{\cos((j+1/2)\pi/(n+1))\}_{j=0}^{n}$ were shown to be optimal for interpolation error on $[-1,1]$. One might guess that choosing equally spaced points for interpolation might lead to comparable errors, but this is not the case. Consider the function

$$f(x) := \frac{1}{1+25x^2}, \qquad \forall\, x \in [-1,1].$$

Using Matlab, plot various interpolating polynomials p_n of this function on $[-1,1]$ using equally spaced nodes. (When you do your plot, it might be best to plot both $f(x)$ and $p_n(x)$ at a lot of points, e.g. you could plot f using the commands `x=linspace(-1,1,1000)` and `plot(x,1./(1+25*x.^2))`.) Verify experimentally that $||f-p_n||_\infty := \max_{x\in[-1,1]} |f(x)-p_n(x)|$ does not go to zero as $n \to \infty$. (In fact, these numbers go to infinity!)

Then, plot the interpolating polynomials of f on $[-1,1]$ using the Chebyshev nodes $\{\cos(j\pi/n)\}_{j=0}^{n}$. Verify experimentally that $\lim_{n\to\infty} ||f-p_n||_\infty = 0$.