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

Due October 6, 1159PM PST, to be uploaded as a single PDF document to blackboard (under the Assignments tab).

Homework 6

Exercise 1. Write a computer program on your own that finds the QR factorization of an $n \times n$ matrix for arbitrary n, and apply this program to the following matrix.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 2 \\ 2 & 3 & 0 & 0 & 0 \\ 4 & 5 & 1 & 0 & 2 \\ -6 & 0 & 1 & 2 & 0 \\ 3 & 0 & 4 & 0 & -1 \end{pmatrix}$$

(You should not use any built in linear algebra functions such as `qr`.)

Exercise 2. In this exercise, we will compare the speed and error of solving the linear system of equations

$$Ax = b$$

using either the LU or QR decomposition. Here A is a known $n \times n$ real matrix, $b \in \mathbf{R}^n$ is known, and we want to solve for $x \in \mathbf{R}^n$.

To do this, let's first construct an $n \times n$ integer matrix A with $n = 1000$. Then, let's construct the LU decomposition using built-in commands, and then time how long it takes to solve $Ax = b$ with a randomly chosen b (and let's also use built-in solver commands for the upper and lower triangular systems):

```
n=10^3;
A=round(n*(rand(n,n) - 1/2));
tic;   % start the timer
[L U P]=lu(A);
b=rand(n,1);
%% Solve Ax=b, i.e. LUx=PAx=Pb.
%% first solve Ly=Pb, then solve Ux=y
y=L\(P*b);
x=U\y;
rtime=toc;   % run time of the linear solver
aerr=norm(A*x -b);   % approximate error of linear solver
```

In the last line, we approximated the error of $Ax - b$. (Since the computation of Ax itself has numerical errors, we only obtain an approximation of the actual value of $Ax - b$, where

x is the output of the program.) Plot the run time a function of n, and also plot the error as a function of n, where n takes the values $2^2, 2^3, 2^4, \ldots, 2^{12}, 2^{13}$.

Modify the above program to solve $Ax = b$ using instead the QR decomposition.

Finally, repeat the above exercise (both for LU and QR decompositions) by using the specific matrix A that we previously found to be troublesome for the LU decomposition. An $n \times n$ version of this matrix can be created in Matlab using the following commands.

```
A=-((ones(n,1))*(1:n)<((1:n)')*ones(1,n)) + eye(n);
A(1:n,n)=1;
```

In each above case, does the LU or QR decomposition do better? (Answer in terms of run time and in terms of errors.)

Exercise 3. Write a computer program on your own that finds a Cholesky Decomposition of an $n \times n$ symmetric real matrix for arbitrary $n > 1$, and then use your program for the matrix

$$A = \begin{pmatrix} 6 & 0 & -4 & 0 \\ 0 & 7 & 0 & -1 \\ -4 & 0 & 6 & 0 \\ 0 & -1 & 0 & 7 \end{pmatrix}.$$

(In a previous homework, we noted that A is symmetric and positive definite.)

Exercise 4 (The Power Method)**.** This exercise gives an algorithm for finding the eigenvectors and eigenvalues of a symmetric matrix. In modern statistics, this is often a useful thing to do. The Power Method described below is not the best algorithm for this task, but it is perhaps the easiest to describe and analyze.

Let A be an $n \times n$ real symmetric matrix. Let $\lambda_1 \geq \cdots \geq \lambda_n$ be the (unknown) eigenvalues of A, and let $v_1, \ldots, v_n \in \mathbf{R}^n$ be the corresponding (unknown) eigenvectors of A such that $||v_i|| = 1$ and such that $Av_i = \lambda_i v_i$ for all $1 \leq i \leq n$.

Given A, our first goal is to find v_1 and λ_1. For simplicity, assume that $1/2 < \lambda_1 < 1$, and $0 \leq \lambda_n \leq \cdots \leq \lambda_2 < 1/4$. Suppose we have found a vector $v \in \mathbf{R}^n$ such that $||v|| = 1$ and $|\langle v, v_1\rangle| > 1/n$. (An exercise more suitable for a probability class shows that a randomly chosen v satisfies this property, with probability at least $1/2$.) Let k be a positive integer. Show that

$$A^k v$$

approximates v_1 well as k becomes large. More specifically, show that for all $k \geq 1$,

$$\left|\left|A^k v - \langle v, v_1\rangle \lambda_1^k v_1\right|\right|^2 \leq \frac{n-1}{16^k}.$$

(Hint: use the spectral theorem for symmetric matrices.)

Since $|\langle v, v_1\rangle|\,\lambda_1^k > 2^{-k}/n$, this inequality implies that $A^k v$ is approximately an eigenvector of A with eigenvalue λ_1. That is, by the triangle inequality,

$$\left|\left|A(A^k v) - \lambda_1(A^k v)\right|\right| \leq \left|\left|A^{k+1}v - \langle v, v_1\rangle\lambda_1^{k+1}v_1\right|\right| + \lambda_1\left|\left|\langle v, v_1\rangle\lambda_1^k v_1 - A^k v\right|\right| \leq 2\frac{\sqrt{n-1}}{4^k}.$$

Moreover, by the reverse triangle inequality,

$$\left|\left|A^k v\right|\right| = \left|\left|A^k v - \langle v, v_1\rangle\lambda_1^k v_1 + \langle v, v_1\rangle\lambda_1^k v_1\right|\right| \geq \frac{1}{n}2^{-k} - \frac{\sqrt{n-1}}{4^k}.$$

In conclusion, if we take k to be large (say $k > 10\log n$), and if we define $z := A^k v$, then z is approximately an eigenvector of A, that is

$$\left|\left|A\frac{A^k v}{||A^k v||} - \lambda_1\frac{A^k v}{||A^k v||}\right|\right| \leq 4n^{3/2}2^{-k} \leq 4n^{-4}.$$

And to approximately find the first eigenvalue λ_1, we simply compute

$$\frac{z^T A z}{z^T z}.$$

That is, we have approximately found the first eigenvector and eigenvalue of A.

Remarks. To find the second eigenvector and eigenvalue, we can repeat the above procedure, where we start by choosing v such that $\langle v, v_1\rangle = 0$, $||v|| = 1$ and $|\langle v, v_2\rangle| > 1/(10\sqrt{n})$. To find the third eigenvector and eigenvalue, we can repeat the above procedure, where we start by choosing v such that $\langle v, v_1\rangle = \langle v, v_2\rangle = 0$, $||v|| = 1$ and $|\langle v, v_3\rangle| > 1/(10\sqrt{n})$. And so on.

Google's PageRank algorithm uses the power method to rank websites very rapidly. In particular, they let n be the number of websites on the internet (so that n is roughly 10^9). They then define an $n \times n$ matrix C where $C_{ij} = 1$ if there is a hyperlink between websites i and j, and $C_{ij} = 0$ otherwise. Then, they let B be an $n \times n$ matrix such that B_{ij} is 1 divided by the number of 1's in the i^{th} row of C, if $C_{ij} = 1$, and $B_{ij} = 0$ otherwise. Finally, they define

$$A = (.85)B + (.15)D/n$$

where D is an $n \times n$ matrix all of whose entries are 1.

The power method finds the eigenvector v_1 of A, and the size of the i^{th} entry of v_1 is proportional to the "rank" of website i.

Exercise 5. 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 power method (i.e. by examining large powers of A in Matlab), find the largest eigenvalue $\lambda \in \mathbf{R}$ of A and a corresponding eigenvector $v \in \mathbf{R}^5$ with $||v||_2 = 1$.

Note that $(A - \lambda vv^T)v = Av - \lambda v = 0$, and if w is any other eigenvector of A, then $(A - \lambda vv^T)w = Aw$. Using this observation, apply the power method to $A - \lambda vv^T$ to find the second largest eigenvalue of A.

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

Exercise 6. Let A be an $m \times n$ complex matrix. Show that $||A||_{2\to 2}^2$ is equal to the largest eigenvalue of AA^* (or of A^*A). That is, $||A||_{2\to 2}$ is the largest singular value of A.