Problem 1. (10) Let $A$ be a square nonsingular matrix of order $n$. Gaussian elimination with partial pivoting (i.e., where the pivot at step $k$ is chosen from the $k$th column) transforms $A$ into an upper triangular matrix using a series of permutations and row manipulations that can be represented by the mathematical expression

$$L_{n-1}P_{n-1} \cdots L_2P_2L_1P_1A = U.$$ 

Here, $P_k$ is a permutation matrix and $L_k$ is a lower triangular matrix all of whose diagonal entries are 1 and with nonzero entries in the $k$th column below the diagonal. Show that this implies that there is an LU decomposition of a permuted version of $A$, that is

$$PA = LU$$

where

$$P = P_{n-1} \cdots P_2P_1$$

and

$$L = \hat{L}_1\hat{L}_2\cdots\hat{L}_{n-1}$$

in which each component $\hat{L}_k$ has the same nonzero structure as $L_k$.

Problem 2. (20) Let $A \in \mathbb{R}^{m \times n}$ denote a real matrix. We saw in class how to construct a QR-factorization of $A$ using Gram-Schmidt orthogonalization. Here is another way, based on so-called “Householder transformations.”

a. Given a nonzero vector $u \in \mathbb{R}^n$, let $Q = I - \frac{2}{u^Tu}uu^T$. Show that $Q$ is an orthogonal matrix, i.e., $Q^TQ = QQ^T = I$.

b. Show that $Qu = -u$ and that $Qv = v$ for any $v \in \mathbb{R}^n$ such that $v$ is orthogonal to $u$.

c. Give a geometric interpretation of this transformation. That is, suppose $x \in \mathbb{R}^n$, and let $u_2, \ldots, u_n$ be an orthonormal basis of the $(n-1)$-dimensional subspace that is orthogonal to $u$. Using the basis $\{u, u_2, \ldots, u_n\}$ for $\mathbb{R}^n$, give a geometric interpretation of the result of computing $Qx$. Part of your solution should include a picture for the case of $\mathbb{R}^3$.

d. Show that if $x$ and $y$ are any two vectors in $\mathbb{R}^n$ with $\|x\| = \|y\|$, then for the choice $u = x - y$, $Qx = y$. Hint: Use the result of part (b).

e. Finally, show that $n-1$ suitable choices of $Q$ leads to a QR factorization of $A$. Hint: For the first step, let $x$ denote the first column $a_1$ of $A$ and $y = \|a_1\| e_1$ where $e_1$ is the “unit vector” with one in the first entry and zero in all other entries.

The next two problems concern the Lanczos method, an algorithm for estimating eigenvalues of symmetric matrices. Problem 3 concerns the derivation of the method and Problem 4 concerns an experimental study of it.
Problem 3. (15) Let $A$ be a symmetric matrix in $\mathbb{R}^{N \times N}$. Given a vector $v^{(1)} \in \mathbb{R}^N$ such that $\|v^{(1)}\| = 1$, consider the construction of vectors $v^{(2)}, v^{(3)}, \ldots$

$$
\gamma_{j+1}v^{(j+1)} = Av^{(j)} - \delta_j v^{(j)} - \gamma_j v^{(j-1)}, \quad 1 \leq j \leq k - 1,
$$

where $\delta_j = (Av^{(j)}, v^{(j)})$, $\gamma_{j+1}$ is chosen so that $\|v^{(j+1)}\| = 1$, and $v^{(0)} = 0$.

a. Show that the vectors $\{v^{(j)}\}$ constructed this way are orthonormal, that is,

$$
(v^{(i)}, v^{(j)}) = \begin{cases} 
0 & \text{for } i \neq j \\
1 & \text{for } i = j
\end{cases}
$$

b. Let $T_k$ denote the symmetric tridiagonal matrix of order $k$ given by

$$
\text{tridiag} \begin{bmatrix} \gamma_j, \delta_j, \gamma_{j+1} \end{bmatrix}, \quad 1 \leq j \leq k,
$$

that is, with $\delta_j$ in entry $(j, j)$, $\gamma_j$ in entry $(j, j - 1)$ and $\gamma_{j+1}$ in entry $(j, j + 1)$. Let

$$
V_k = \begin{bmatrix} v^{(1)}, \ldots, v^{(k)} \end{bmatrix}
$$

denote the $N \times k$ matrix whose columns are the vectors generated above. Show that

$$
V_k^T AV_k = T_k.
$$

(1)

c. $T_k$ has an eigenvector/eigenvalue decomposition $T_k = U_k \Lambda_k U_k^T$, that is, $\Lambda_k$ is a diagonal matrix containing the eigenvalues of $T_k$, and the columns of $U_k$ are the eigenvectors. Because $T_k$ is small and tridiagonal, its eigenvalues can be computed inexpensively using the QR iteration described in class. The Lanczos algorithm uses these eigenvalues as estimates for those of $A$, and it uses the columns of $W_k \equiv V_k U_k$ as estimates for the associated eigenvectors. Show that the residual of the eigenvalue problem,

$$
AW_k - W_k \Lambda_k,
$$

is orthogonal to the space spanned by the columns of $W_k$. This is referred to as a Galerkin condition.

Problem 4. (20) For this problem, you will need a sample matrix $A$. You can get this from a repository of benchmark problems available on the Matrix Market web page

http://math.nist.gov/MatrixMarket/

Go to this web page, find the problem “662_BUS” and download it. In addition, download the MATLAB function mmread.m, which can be used to put the matrix $A$ in MATLAB format.

a. Write a program that implements the Lanczos algorithm and explore its performance for estimating the eigenvalues of the matrix obtained from Matrix Market. Specifically:
• Identify how many steps are needed to find estimates for each of the five largest eigenvalues of $A$ so that Euclidian norms of each of the residuals $Aw^{(j)} - \lambda_j w^{(j)}$ associated with the five eigenpair estimates $(w^{(j)}, \lambda_j)$ is less than $10^{-6}$. That is, you want

$$\|Aw^{(j)} - \lambda_j w^{(j)}\|_2/\lambda_j < 10^{-6}$$

for each of the five largest eigenvalues obtained from the Lanczos method.

• Compare the eigenvalues found with those obtained with the `eig` function, which can be used to compute all the eigenvalues of the matrix.

• Also keep track of the residuals of the estimates of the five smallest eigenvalues obtained during the computation. Are these larger or smaller than those for the largest eigenvalues?

b. Identify a way to improve on the performance of the algorithm for computing the smallest eigenvalues and implement this alternative strategy. As in part (a), identify how many steps are needed to find estimates for each of the five smallest eigenvalues and associated eigenvectors so that the residual norms are less than $10^{-6}$, and compare those found to those found by `eig`.

c. You should have found that the Lanczos method is finding multiple copies of some eigenvalues that aren’t found by `eig`. This is caused by loss of orthgonality of the vectors $\{v^{(j)}\}$. One way to fix this is to force each new $v^{(j+1)}$ to be orthogonal to $v^{(1)}, \ldots, v^{(j)}$ by “reorthogonalizing” it using the (modified) Gram-Schmidt process. Incorporate this into your Lanczos algorithm (this should entail just a few extra lines of code) and redo the computations from parts (a) and (b). How do the results compare to those obtained previously?

The Lanczos program should be a cleanly written `MATLAB` function that takes as arguments $A$ and $k$ (the size of the matrix $T_k$). For part (a), it should return as output the $k$ eigenvalue estimates for $A$. For part (b), it should be modified in an appropriate way to return what you are looking for. In fact, there really needs to be just one program, with one input parameter specifying whether you are working in the mode of part (a) or part (b), and another one specifying whether you reorthogonalize or not.