Instructor: Emre Mengi

Math 304 (Spring 2010)

Study Guide for Week 4
Homework 3 concerns the topics listed below.

- Orthonormal sets and bases
- Orthogonal complement of a subspace
- Orthogonal projection of a vector onto a subspace
- Orthogonal matrices (Watkins 3.2)
- QR factorization (Watkins 3.2)
- Householder reflectors (Watkins 3.2, Fausett 4.2.1)
- Computation of the QR factorization by Householder reflectors (Watkins 3.2, Fausett 4.3)

Homework 3 (due on March 19th, Friday by 4pm)
In Matlab question (questions **5**) attach the Matlab outputs and print-outs of the Matlab routines that you implemented.

1. Consider the orthonormal basis for $\mathbb{R}^3$ given below.

$$\mathcal{B} = \left\{ \underbrace{\frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}}_{b_1}, \underbrace{\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{b_2}, \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}}_{b_3} \right\}$$

(a) Find the orthogonal complement of the subspace $\mathcal{W} = \text{span}\{b_1\}$.

(b) Find the orthogonal projections of $u = \begin{bmatrix} 7 \\ 1 \\ 2 \end{bmatrix}$ onto the subspaces **(i)** $\mathcal{W}_1 = \text{span}\{b_1\}$ and **(ii)** $\mathcal{W}_2 = \text{span}\{b_2, b_3\}$.

(c) Find the orthogonal projector $P \in \mathbb{R}^{3\times 3}$ onto the subspace $\mathcal{W}_1 = \text{span}\{b_1\}$.

2. This question concerns the Householder reflectors. Recall that the Householder reflectors are used to compute a QR factorization of a matrix.

(a) Express the Householder reflector $Q \in \mathbf{R}^{n\times n}$ such that

$$Qb = \|b\|_2 e_1$$

in terms of $b \in \mathbf{R}^n$ where e_1 denotes the first column of the $n \times n$ identity matrix.

(b) Verify that the Householder reflector Q in part **(a)** is an orthogonal and symmetric matrix.

(c) Find an orthogonal matrix $Q \in \mathbf{R}^{3\times 3}$ such that

$$Q \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}.$$

(d) Find an orthogonal matrix $Q \in \mathbf{R}^{5\times 5}$ such that

$$Q \begin{bmatrix} 3 \\ 4 \\ 1 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 3 \\ 0 \\ 0 \end{bmatrix}.$$

3. Compute the QR factorization of the matrix A given below by hand.

$$A = \begin{bmatrix} 5 & 2 \\ -1 & 7 \end{bmatrix}$$

4. Consider the pseudocode given below to compute the upper triangular factor R of a QR factorization of A by Householder reflectors. Note that the majority of the computational work is due to the operation

$$A_{k:n,k:n} \leftarrow A_{k:n,k:n} - 2u_k u_k^T A_{k:n,k:n}.$$

---

for $k = 1, n-1$ **do**
  $v \leftarrow A_{k:n,k}$
  $u_k \leftarrow v - \|v\| e_1$
  $u_k \leftarrow u_k / \|u_k\|$
  $A_{k:n,k:n} \leftarrow A_{k:n,k:n} - 2u_k u_k^T A_{k:n,k:n}$
end for
$R \leftarrow A$
Return R

(a) Suppose you perform the operation mentioned above in the following order

$$A_{k:n,k:n} \leftarrow A_{k:n,k:n} - 2(u_k u_k^T) A_{k:n,k:n}$$

that is you first multiply the vectors u_k and u_k^T. Calculate the total # of flops required by the pseudocode above in terms of n.

(b) Now suppose you first multiply $u_k^T A_{k:n,k:n}$, that is

$$A_{k:n,k:n} \leftarrow A_{k:n,k:n} - 2u_k(u_k^T A_{k:n,k:n}).$$

Calculate the total # of flops required by the pseudocode in terms of n.

5. Implement a Matlab routine `qrfactor.m` to compute the QR factorization of A by Householder reflectors. Your routine must take one input parameter, that is the matrix A for which the QR factorization is sought. It must return two output parameters Q and R, the orthogonal and upper triangular factors of A satisfying $A = QR$. Make sure that your implementation requires $O(n^3)$ operations for an $n \times n$ matrix A.

A matlab routine `constructq.m` is provided on the course website to construct the orthogonal matrix Q from the Householder vectors $u_1, u_2, \ldots, u_{n-1}$. Specifically if you create an $n \times (n-1)$ matrix U such that $U(1 : n-k+1, k) = u_k$ for $k = 1, \ldots, n-1$. Then `Q = constructq(U)` forms the orthogonal factor Q from U. It is up to you whether to use `constructq.m` or not, however if you prefer to form Q on your own, make sure that the computational complexity does not exceed $O(n^3)$. In particular you should definitely avoid the multiplication of two $n \times n$ orthogonal matrices, since this would require $2n^3$ flops and performing $n-2$ such matrix-matrix multiplications would require $2n^3(n-2)$.

Test your implementation on two random matrices of sizes 3×3 and 7×7, which you can create by typing `>> A = randn(3)` and `>> A = randn(7)`, respectively. Verify that the computed factors $\hat{Q}, \hat{R}$ are accurate by checking that the norm of $A - \hat{Q}\hat{R}$ is close to the machine precision $\epsilon_{\text{mach}} \approx 10^{-16}$.
(Note : To view the computed results in full precision you should type `>> format long e`)