Solutions for Homework 4 Foundations of Computational Math 1 Fall 2011

Problem 4.1

Recall that an elementary reflector has the form $Q = I + \alpha x x^T \in \mathbb{R}^{n\times n}$ with $\|x\|_2 \neq 0$.

4.1.a. Show that Q is orthogonal if and only if

$$\alpha = \frac{-2}{x^T x} \text{ or } \alpha = 0$$

4.1.b. Given $v \in \mathbb{R}^n$, let $\gamma = \pm\|v\|$ and $x = v + \gamma e_1$. Assuming that $x \neq v$ show that

$$\frac{x^T x}{x^T v} = 2$$

4.1.c. Using the definitions and results above show that $Qv = -\gamma e_1$

Solution: We have

$$\begin{aligned} Q^T Q &= (I + \alpha x x^T)^T (I + \alpha x x^T) = (I + \alpha x x^T)(I + \alpha x x^T) \\ &= I + 2\alpha x x^T + \alpha^2 x (x^T x) x^T = I + (2\alpha + \alpha^2 x^T x) x x^T \end{aligned}$$

Since x is arbitrary we must have

$$(2\alpha + \alpha^2 x^T x) = 0 \to \alpha = \frac{-2}{x^T x}$$

Now taking $x = v + \gamma e_1$, we have

$$\begin{gathered} x^T v = v^T v + \gamma e_1^T v = \gamma^2 + \gamma \nu_1 \\ x^T x = (v + \gamma e_1)^T (v + \gamma e_1) = v^T v + 2\gamma\nu_1 + \gamma^2 = 2(\gamma^2 + \gamma\nu_1) \\ \therefore \frac{x^T x}{x^T v} = 2 \end{gathered}$$

Finally,

$$\begin{gathered} Qv = (I + \alpha x x^T) v = v + \alpha (x^T v) x = v + \alpha(x^T v) v + \alpha (x^T v) \gamma e_1 \\ \alpha(x^T v) = \frac{-2 x^T v}{x^T x} = -1 \\ \therefore Qv = -\gamma e_1 \end{gathered}$$

Problem 4.2

4.2.a

This part of the problem concerns the computational complexity question of operation count.

For both LU factorization and Householder reflector-based orthogonal factorization, we have used elementary transformations, T_i, that can be characterized as rank-1 updates to the identity matrix, i.e.,

$$T_i = I + x_i y_i^T, \;\; x_i \in \mathbb{R}^n \text{ and } y_i \in \mathbb{R}^n$$

Gauss transforms and Householder reflectors differ in the definitions of the vectors x_i and y_i. Maintaining computational efficiency in terms of a reasonable operation count usually implies careful application of associativity and distribution when combining matrices and vectors.

Suppose we are to evaluate

$$z = T_3 T_2 T_1 v = (I + x_3 y_3^T)(I + x_2 y_2^T)(I + x_1 y_1^T)v$$

where $v \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$. Show that by using the properties of matrix-matrix multiplication and matrix-vector multiplication, the vector z can be evaluated in $O(n)$ computations (a good choice of version for an algorithm) or $O(n^3)$ computations (a very bad choice of version for an algorithm).

Solution: First we derive the bad choice:

$$\begin{gathered}
\text{compute } T_1 = (I + x_1 y_1^T) \rightarrow O(n^2) \text{ operations} \\
\text{compute } T_2 = (I + x_2 y_2^T) \rightarrow O(n^2) \text{ operations} \\
\text{compute } T_3 = (I + x_3 y_3^T) \rightarrow O(n^2) \text{ operations} \\
\text{compute } T = (T_3(T_2 T_1)) \rightarrow O(n^3) \text{ operations from two matrix-matrix products} \\
\text{compute } z = Tv \rightarrow O(n^2) \text{ operations} \\
O(n^3) \text{ operations in total as desired}
\end{gathered}$$

Second we derive the best choice:

$$\begin{gathered}
z = (I + x_3 y_3^T)(I + x_2 y_2^T)(I + x_1 y_1^T)v \\
\text{compute } v_1 = v + x_1(y_1^T v) \rightarrow O(n) \text{ operations from an inner product and a vector triad} \\
\text{compute } v_2 = v_1 + x_2(y_2^T v_1) \rightarrow O(n) \text{ operations from an inner product and a vector triad} \\
\text{compute } z = v_2 + x_3(y_3^T v_2) \rightarrow O(n) \text{ operations from an inner product and a vector triad} \\
O(n) \text{ operations in total as desired}
\end{gathered}$$

For completeness we note an $O(n^2)$ version also exists:

$$\begin{gathered}
\text{compute } T_1 = (I + x_1 y_1^T) \to O(n^2) \text{ operations} \\
\text{compute } T_2 = (I + x_2 y_2^T) \to O(n^2) \text{ operations} \\
\text{compute } T_3 = (I + x_3 y_3^T) \to O(n^2) \text{ operations} \\
\text{compute } v_1 = T_1 v \to O(n^2) \text{ operations} \\
\text{compute } v_2 = T_2 v_1 \to O(n^2) \text{ operations} \\
\text{compute } z = T_3 v \to O(n^2) \text{ operations} \\
O(n^2) \text{ operations in total}
\end{gathered}$$

4.2.b

This part of the problem concerns the computational complexity question of storage space.

Recall, that we discussed and programmed an **in-place** implementation of LU factorization that was very efficient in terms of storage space. An array with n^2 entries initialized with $array(I, J) = \alpha_{ij}$ could be used to store the n^2 entries needed to specify L and U, i.e., λ_{ij} for $j < i$, $2 \leq i \leq n$ and $1 \leq j \leq n-1$, and μ_{ij} for $i < j$, $1 \leq i \leq n$ and $1 \leq j \leq n$.

Let $A \in \mathbb{R}^{n \times k}$, $n \geq k$, and $rank(A) = k$. Consider the use of Householder reflectors, H_i, $1 \leq i \leq k$, to transform A to upper trapezoidal form, i.e.,

$$H_k H_{k-1} \cdots H_2 H_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

$$R \in \mathbb{R}^{k \times k} \text{ nonsingular upper triangular}$$

Suppose you are given an array with $n \times k$ entries initialized with $array(I, J) = \alpha_{ij}$ and you are to implement your algorithm using minimal storage.

(i) Are you able to store all of the information needed to specify the H_i, $1 \leq i \leq k$ and R within the array with $n \times k$ entries? Justify your answer.

(ii) If you are not able to store all of the information in the array, how much extra storage do you need and what do you store in it?

Solution: If the entries of R are stored in-place in the array then only the positions where 0 values are created in the transformed A are available to store information specifying the H_i. Let $H_i = I + \alpha_i x_i x_i^T$. H_i transforms $n-i$ elements in column i of the matrix to 0 values. To do this it requires α_i and $n-i+1$ nonzero entries in x_i. So, a total of $n-i+2$ scalars are needed to specify H_i.

Therefore, there are

$$\sum_{i=1}^{k} (n-i) = kn - \frac{k(k+1)}{2}$$

positions with 0 values that are available in the array. The H_i require

$$\sum_{i=1}^{k} (n-i+2) = kn - \frac{k(k+1)}{2} + 2k$$

storage locations. We need $2k$ extra locations to store α_i and one of the nonzero elements of x_i, typically the topmost in the i-th position of x_i for $1 \leq i \leq k$.

This can be reduced to k extra locations if the form

$$H_i = I - 2u_i u_i^T$$

where $\|u_i\|_2 = 1$ is used. This is not typically used in libraries for reasons related to possible overflow.

Problem 4.3

Let $A \in \mathbb{R}^{n\times k}$ have full column rank. Describe an efficient algorithm based on Householder reflectors, H_i, $1 \le i \le k$ that computes a matrix $Q \in \mathbb{R}^{n\times k}$ with orthonormal columns such that

$$\mathcal{R}(A) = \mathcal{R}(Q)$$

i.e., A and Q have the same range space.

Solution:

We can construct a series of Householder reflectors so that

$$H_k \cdots H_2 H_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}$$
$$H^T = \begin{pmatrix} Q & Q_\perp \end{pmatrix}$$
$$H^T = H_1 H_2 \cdots H_k$$
$$Q \in \mathbb{R}^{n\times k} \text{ and } \mathcal{R}(A) = \mathcal{R}(Q)$$

It follows that

$$H^T e_i = Q e_i = H_1 H_2 \cdots H_k e_i \quad 1 \le i \le k$$

Therefore, Q can be evaluated by applying H^T in factored form to $e_i \quad 1 \le i \le k$. Some computations can be saved by noting that due to the structure in the vector u_i that defines H_i we have $H_i e_j = e_j$ with $j < i$.

Problem 4.4

Consider a Householder reflector, H, in $\mathbb{R}^2$. Show that

$$H = \begin{pmatrix} -\cos(\phi) & -\sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

where ϕ is some angle.

Solution:

We have the basic identities

$$\begin{aligned} \sin^2(\theta) &= \frac{1}{2}(1 - \cos(2\theta)) \\ \cos^2(\theta) &= \frac{1}{2}(1 + \cos(2\theta)) \\ \cos(\theta)\sin(\theta) &= \frac{1}{2}\sin(2\theta). \end{aligned}$$

Any unit length vector in two space must be of the form

$$\begin{pmatrix} \gamma \\ \sigma \end{pmatrix}$$

where $\gamma = \cos(\theta)$ and $\sigma = \sin(\theta)$ for some angle θ.

$$-2\begin{pmatrix} \gamma \\ \sigma \end{pmatrix}\begin{pmatrix} \gamma & \sigma \end{pmatrix} = \begin{pmatrix} -2\gamma^2 & -2\gamma\sigma \\ -2\gamma\sigma & -2\sigma^2 \end{pmatrix}$$

Therefore

$$H = \begin{pmatrix} 1 - 2\gamma^2 & -2\gamma\sigma \\ -2\gamma\sigma & 1 - 2\sigma^2 \end{pmatrix}$$

Applying the identities in the problem yields

$$H = \begin{pmatrix} -\cos(2\theta) & -\sin(2\theta) \\ -\sin(2\theta) & \cos(2\theta) \end{pmatrix}$$

and the proof is completed by setting $\phi = 2\theta$.