THE UNIVERSITY OF TEXAS
AT AUSTIN

CS383C NUMERICAL ANALYSIS

Homework 03

Edited by LaTeX

Department of Computer Science

STUDENT
Jimmy Lin
xl5224

COURSE COORDINATOR
Robert A. van de Geijn

UNIQUE NUMBER
53180

RELEASE DATE
Sep. 18 2014

DUE DATE
May. 23 2014

TIME SPENT
5 hours

September 21, 2014

Exercise 2. Show that if H is a reflector, then

2.1 $HH = I$

Since H is a reflector, we have

$$H = I - 2uu^H \tag{1}$$

where u is unit vector ($||u||_2 = 1$).

$$\begin{aligned} HH &= (I - 2uu^H)(I - 2uu^H) \\ &= I \cdot I - 2uu^H - 2uu^H + 4(uu^H)(uu^H) \\ &= I - 4uu^H + 4||u||_2 uu^H \\ &= I - 4uu^H + 4uu^H \\ &= I \end{aligned} \tag{2}$$

Lemma 1. $(uu^H)(uu^H) = ||u||_2 uu^H$

Proof.

$$uu^H = \tag{3}$$

□

2.2 $H = H^H$

$$\begin{aligned} H^H &= (I - 2uu^H)^H \\ &= I^H - (2uu^H)^H \\ &= I - 2(u^H)^H u^H \\ &= I - 2uu^H \\ &= H \end{aligned} \tag{4}$$

2.3 $HH^H = I$

In terms of (4), multiply both sides with H and then we have

$$H^H H = HH \tag{5}$$

Then by (2), we have

$$H^H H = I \tag{6}$$

Exercise 4. Show that if $x \in \mathbb{R}^n$, $v = x \mp ||x||_2 e_0$ and $\tau = v^T v/2$, then $(I - \frac{1}{\tau} vv^T) = \pm||x||_2 e_0$

We start from the reflector $I - \frac{1}{\tau} vv^T$ on x,

$$
\begin{aligned}
& (I - \frac{1}{\tau} vv^T)x \\
= & \left(I - \frac{2vv^T}{v^T v}\right)x \\
= & \left(I - \frac{2(x \mp ||x||_2 e_0)(x \mp ||x||_2 e_0)^T}{(x \mp ||x||_2 e_0)^T (x \mp ||x||_2 e_0)}\right)x \\
= & \left(I - 2\frac{xx^T \mp ||x||_2 (xe_0^T + e_0 x^T) + ||x||_2^2 e_0 e_0^T}{2||x||_2^2 \mp 2||x||_2 e_0^T x}\right)x \\
= & \left(I - \frac{xx^T \mp ||x||_2 (xe_0^T + e_0 x^T) + ||x||_2^2 e_0 e_0^T}{||x||_2^2 \mp ||x||_2 e_0^T x}\right)x \\
= & \frac{||x||_2^2 \mp ||x||_2 e_0^T x - xx^T \pm ||x||_2 (xe_0^T + e_0 x^T) - ||x||_2^2 e_0 e_0^T}{||x||_2^2 \mp ||x||_2 e_0^T x} x \\
= & \frac{||x||_2^2 x \mp ||x||_2 e_0^T x x - xx^T x \pm ||x||_2 (xe_0^T + e_0 x^T)x - ||x||_2^2 e_0 e_0^T x}{||x||_2^2 \mp ||x||_2 e_0^T x} \\
= & \frac{(||x||_2^2 x - xx^T x) + (\mp||x||_2 e_0^T x x \pm ||x||_2 x e_0^T x) \pm ||x||_2 e_0 x^T x - ||x||_2^2 e_0 e_0^T x}{||x||_2^2 \mp ||x||_2 e_0^T x} \\
= & \frac{\pm||x||_2 e_0 x^T x - ||x||_2^2 e_0 e_0^T x}{||x||_2^2 \mp ||x||_2 e_0^T x} \\
= & \frac{\pm e_0 x^T x - ||x||_2 e_0 e_0^T x}{||x||_2 \mp e_0^T x} \\
= & \frac{\pm||x||_2^2 e_0 - ||x||_2 (e_0^T x) e_0}{||x||_2 \mp e_0^T x} \\
= & \frac{\pm(||x||_2 - (e_0^T x))||x||_2 e_0}{||x||_2 \mp e_0^T x} \\
= & \pm||x||_2 e_0
\end{aligned}
\tag{7}
$$

Note that above derivation frequently makes use of the following lemma.

Lemma 2. *For arbitrary vector $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $ab^T c = (b^T c)a$.*

Proof.

$$
ab^T c = \begin{pmatrix} a_0 b^T \\ a_1 b^T \\ \vdots \\ a_{n-1} b^T \end{pmatrix} c = \begin{pmatrix} a_0 b^T c \\ a_1 b^T c \\ \vdots \\ a_{n-1} b^T c \end{pmatrix} = \begin{pmatrix} (b^T c) a_0 \\ (b^T c) a_1 \\ \vdots \\ (b^T c) a_{n-1} \end{pmatrix} = (b^T c) \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = (b^T c) a \tag{8}
$$

□

Exercise 5. Complex

Exercise 6. Matrix Equivalence

We start from right hand side

$$\begin{aligned} RHS &= I - \frac{1}{\tau_1}\left(\begin{array}{c} 0 \\ \hline 1 \\ \hline u_2 \end{array}\right)\left(\begin{array}{c} 0 \\ \hline 1 \\ \hline u_2 \end{array}\right)^H \\ &= \left(I - \frac{1}{\tau_1}\left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & \begin{pmatrix} 1 \\ \hline u_2 \end{pmatrix}\begin{pmatrix} 1 \\ \hline u_2 \end{pmatrix}^H \end{array}\right)\right) \\ &= \left(I - \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & \frac{1}{\tau_1}\begin{pmatrix} 1 \\ \hline u_2 \end{pmatrix}\begin{pmatrix} 1 \\ \hline u_2 \end{pmatrix}^H \end{array}\right)\right) \\ &= \left(\begin{array}{c|c} I & 0 \\ \hline 0 & I - \frac{1}{\tau_1}\begin{pmatrix} 1 \\ \hline u_2 \end{pmatrix}\begin{pmatrix} 1 \\ \hline u_2 \end{pmatrix}^H \end{array}\right) \\ &= LHS \end{aligned} \tag{9}$$

Exercise 11. Expensive Algorithm

As indicatd by **Theorem 10** in the note, we have cost of the algorithm in Figure 6 for $A \in \mathbb{C}^{m\times n}$

$$C_{FormQ}(m,n) = 2mn^2 - \frac{2}{3}n^3 \tag{10}$$

For $m = n$, the cost can be simplified as

$$C_{FormQ}(A) = \frac{4}{3}n^3 = \mathbb{O}(n^3) \tag{11}$$

However, if we accumulate Q by using n householder transformation with

$$Q = (\ldots((IH_0)H_1)\ldots H_{n-1}) \tag{12}$$

Then the cost we have is at least

$$C_{accumulation}(A) = n^3 \cdot (n-1) = \mathbb{O}(n^4) \tag{13}$$

where n^3 comes from each one matrix multiplication, and $n-1$ comes from the total number of householder matrix H_i ($i \in \mathbb{Z}, i \in [0, n-1]$).
Comparing formula (11) and (13), it is obvious that the accumulation method is much more expensive than algorithm in Figure 6.