Mathematics, Pusan National University

NUMIRICAL LINEAR ALGEBRA

Lecture 10. Householder Triangularization

Taehyeong Kim th_kim@pusan.ac.kr

September 14, 2020

Content

Householder and Gram-Schmidt

In Lecture 8, by the Gram-Schmidt iteration,

$$A \underbrace{R_1 R_2 \cdots R_n}_{\hat{R}^{-1}} = \hat{Q}$$

where $\hat{Q}$ is orthogonal and each R_k is upper triangular.

$$\left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right] \begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{bmatrix} = \left[\begin{array}{c|c|c|c} q_1 & v_2^{(2)} & \cdots & v_n^{(2)} \end{array}\right]$$

Householder and Gram-Schmidt

In constrast, the Householder method applies

$$\underbrace{Q_n \cdots Q_2 Q_1}_{Q^*} A = R$$

where R is uppertriangular and each Q_k is unitary matrix.

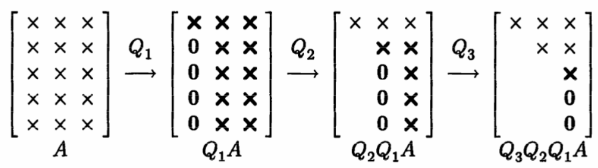
$$\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1 A}{\begin{bmatrix} \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}} \xrightarrow{Q_2} \underset{Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \end{bmatrix}} \xrightarrow{Q_3} \underset{Q_3 Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \mathbf{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}}$$

Householder and Gram-Schmidt

The two methods can thus be summarized as follows:

Gram-Schmidt : triangular orthogonalization,
Householder : orthogonal Triangularization.

Householder Reflectors



The standard approach is as follows.

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$$ where $I : (k-1) \times (k-1)$identity, $F : (m-k+1) \times (m-k+1)$unitary matrix.

Householder Reflectors

The Householder algorithm chooses F to be a particular matrix called a *House holder reflector*.

$$x = \begin{bmatrix} \times \\ \times \\ \times \\ \vdots \\ \times \end{bmatrix} \quad \xrightarrow{F} \quad Fx = \begin{bmatrix} \|x\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1$$

Householder Reflectors

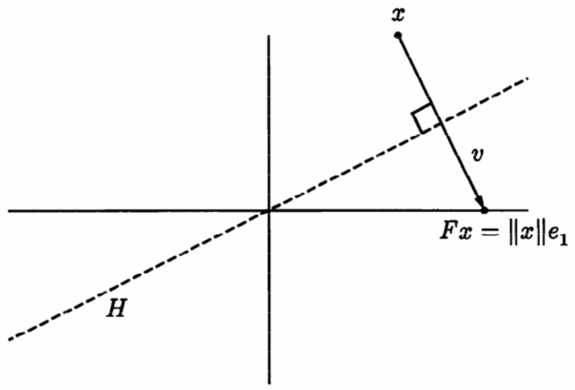
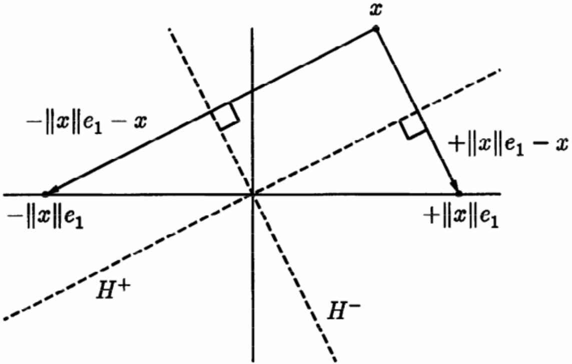


Figure: Householder reflection

The Better of Two Reflectors



For numerical stability, it is important to choose the one that moves x the larger distance.

The Better of Two Reflectors

Difference between Projector and Reflector

In section 6, we studied about projection with orthogonal basis and arbitrary nonzero vector. For arbitrary nonzero vector a, the formulas are

$$P_{\perp a} = I - \frac{aa^*}{a^*a}$$

The reflector Fy should therefore be

$$Fy = \left(I - 2\frac{vv^*}{v^*v}\right)y = y - 2\frac{vv^*}{v^*v}y.$$

Hence the matrix F is

$$F = I - 2\frac{vv^*}{v^*v}$$

The Algorithm

Algorithm 1: Householder QR Factorization

Input: $A : m \times n$ matrix

for $k = 1$ **to** n **do**

$x = A_{k:m,k}$

$v_k = \text{sign}(x_1)\|x\|_2 e_1 + x$

$x_k = v_k / \|v_k\|_2$

$A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^* A_{k:m,k:n})$

end

Applying of Forming Q

Calculate Q^*b where

$$Q^* = Q_n \cdots Q_2 Q_1$$

Algorithm 2: Implicit Calculation of a Product Q^*b

Input: $b : m \times 1$ vector

for $k = 1$ **to** n **do**

$$b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$$

end

Applying of Forming Q

Calculate Qx where

$$Q = Q_1 Q_2 \cdots Q_n$$

Algorithm 3: Implicit Calculation of a Product Qx

Input: $x : m \times 1$ vector

for $k = n$ **downto** 1 **do**

$$x_{k:m} = x_{k:m} - 2v_k(v_k^* x_{k:m})$$

end

Operation Count

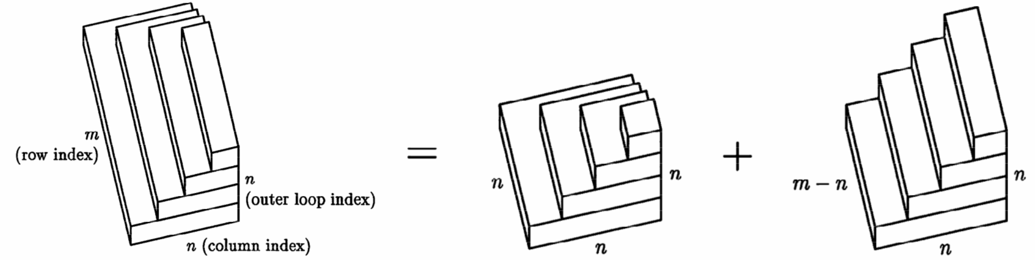
In $A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^* A_{k:m,k:n})$, the FLOPS(FLoating point Operations Per Second) is

- $(2(m-k+1)-1)(n-k+1)$ flops for product $v_k^*(A_{k:m,k:n})$
- $(m-k+1)(n-k+1)$ flops for outer product with v_k
- $(m-k+1)(n-k+1)$ flops for subtraction form $A_{k:m,k:n}$

sum is roughly $4(m-k+1)(n-k+1)$ flops.

$$\sum_{k=1}^{n} 4(m-k+1)(n-k+1)$$

Operation Count



Work for Householder orthogonalization $\sim 2mn^2 - \frac{2}{3}n^3$ flops

Thank you!