Lecture 10
Householder Triangularization

NLA Reading Group Spring '13

by Onur Güngör

Householder and Gram-Schmidt

Gram-Schmidt: triangular orthogonalization

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

Householder: orthogonal triangularization

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

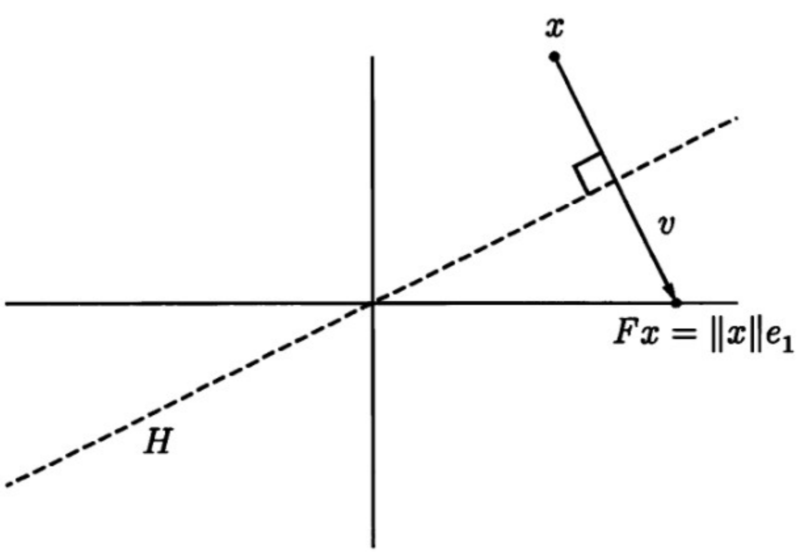
Triangularization by Introducing Zeros

$$
\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}}
\xrightarrow{Q_1}
\underset{Q_1 A}{\begin{bmatrix} \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}}
\xrightarrow{Q_2}
\underset{Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \end{bmatrix}}
\xrightarrow{Q_3}
\underset{Q_3 Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \mathbf{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}}
\qquad (10.1)
$$

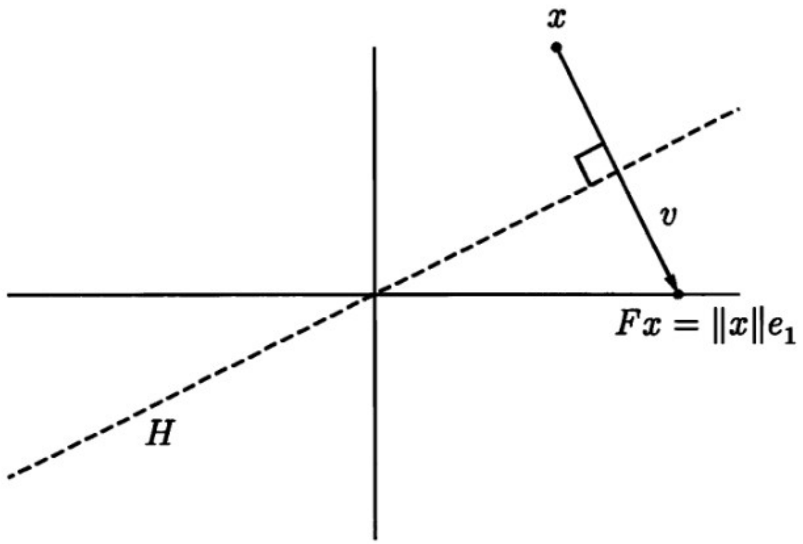
$$
Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}
$$

Householder Reflectors

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



Householder Reflectors



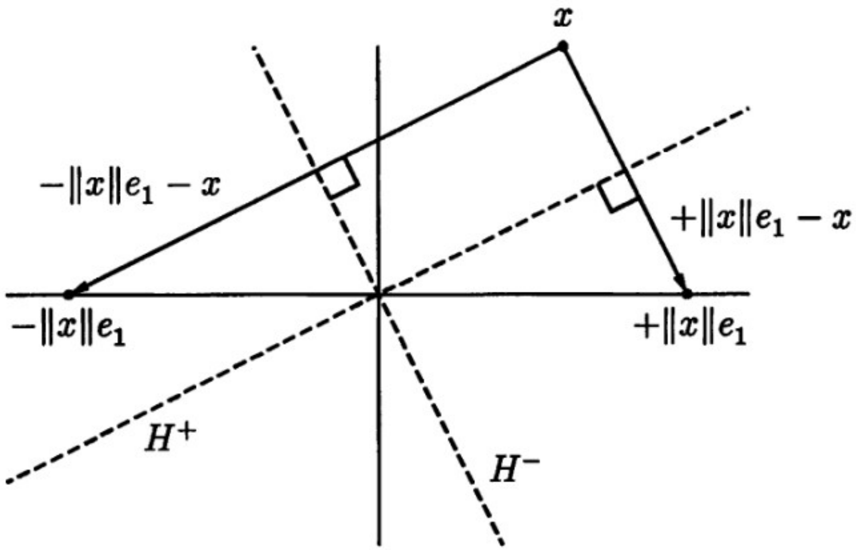
$$v = \|x\|e_1 - x.$$

$$Py \;=\; \left(I - \frac{vv^*}{v^*v}\right)y \;=\; y - v\left(\frac{v^*y}{v^*v}\right)$$

P is the projector onto the space H

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

Householder Reflectors



Instead of $v = \|x\|e_1 - x.$

We use $v = \operatorname{sign}(x_1)\|x\|e_1 + x.$

for numerical stability.

Householder Algorithm

Algorithm 10.1. Householder QR Factorization

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

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

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

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

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

Applying Q

Algorithm 10.2. Implicit Calculation of a Product Q^*b

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

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

This will be employed while solving least squares problems using QR factorization.

Forming Q

Algorithm 10.3. Implicit Calculation of a Product Qx

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

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

Q can be formed by calculating Qe_1, Qe_2, … and Qe_m.

Operation Count

Algorithm 10.1. Householder QR Factorization

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

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

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

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

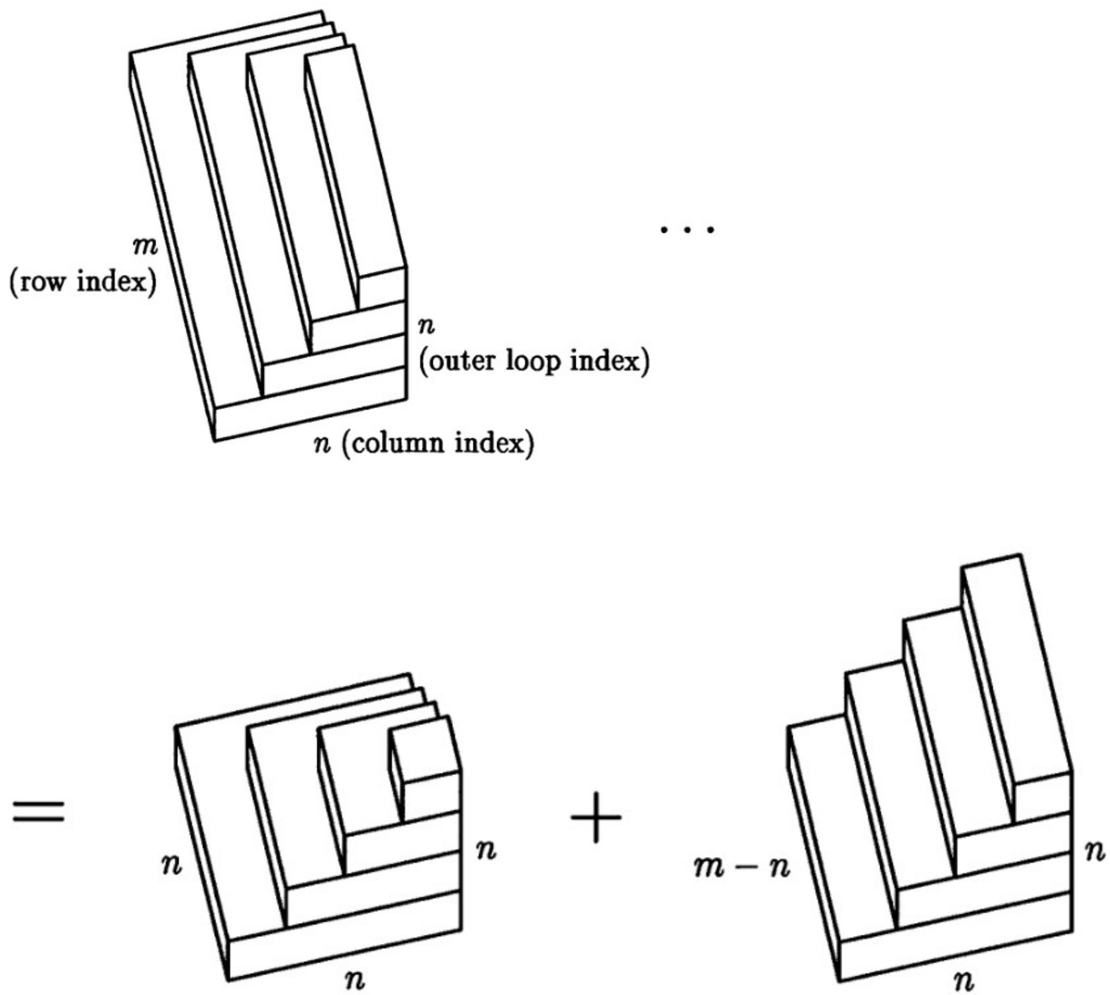
$A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^* A_{k:m,k:n})$

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

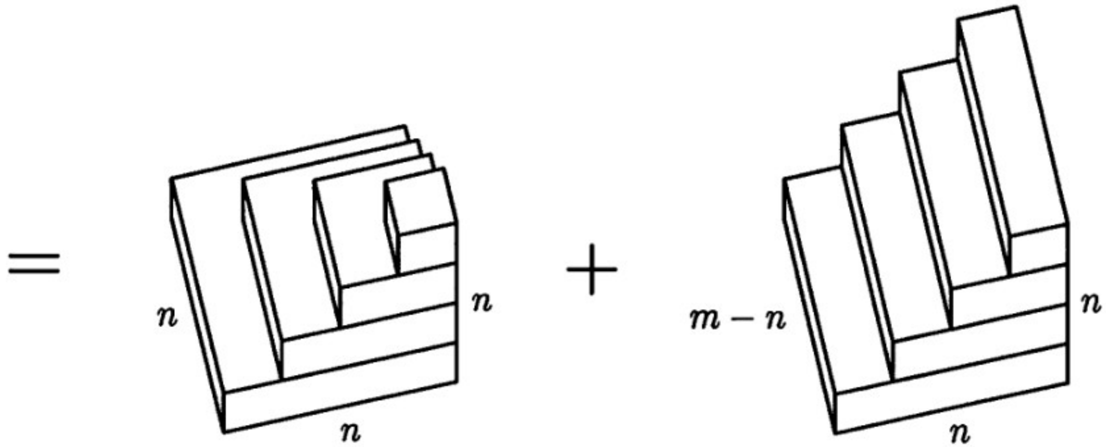
Let $l = m - k + 1$

Each vector requires $4l - 1 \sim 4l$ flops.

Operation Count



Operation Count



$$\sim 2mn^2 - \frac{2}{3}n^3 \text{ flops.}$$

Lecture 11
Least Squares Problems

NLA Reading Group Spring ’13

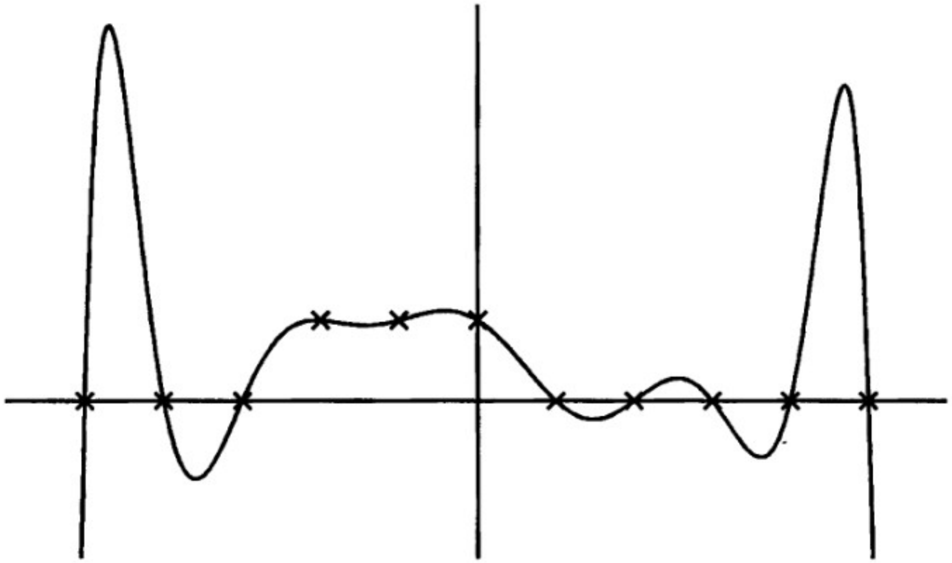
by Onur Güngör

Definition

$$\text{Given } A \in \mathbb{C}^{m\times n},\ m \geq n,\ b \in \mathbb{C}^{m},$$
$$\text{find } x \in \mathbb{C}^{n} \text{ such that } \|b - Ax\|_2 \text{ is minimized.}$$

Polynomial Interpolation

$$\begin{bmatrix} 1 & x_1 & x_1^2 & & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & & x_3^{m-1} \\ & \vdots & & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{m-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

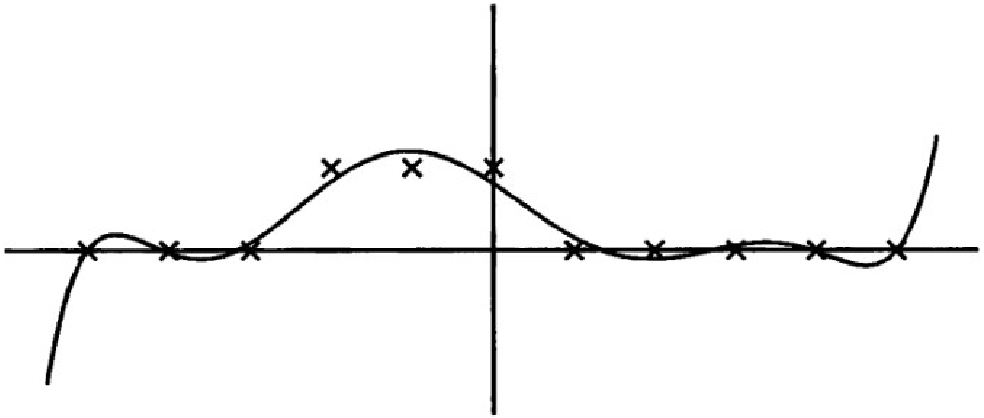


Polynomial Least Squares Fitting

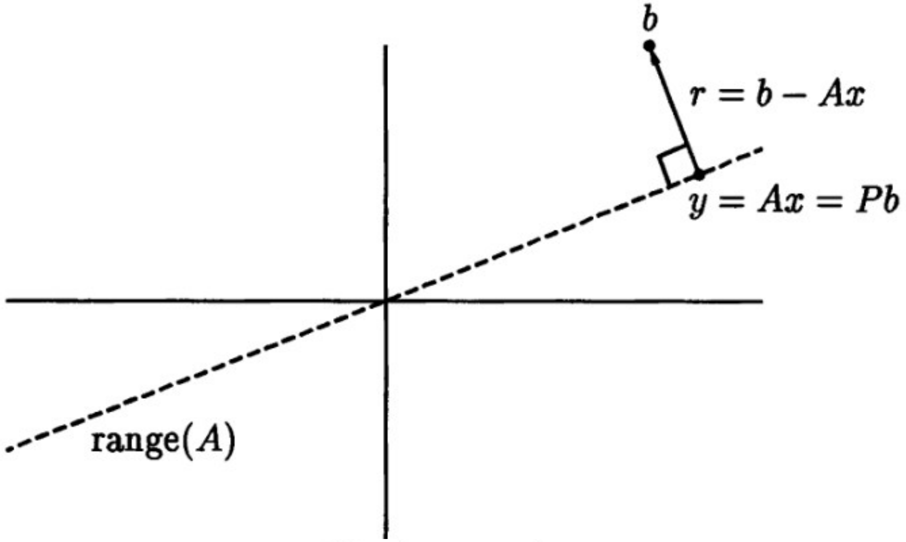
$$\begin{bmatrix} 1 & x_1 & & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ 1 & x_3 & & x_3^{n-1} \\ & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

Solve by minimizing

$$\sum_{i=1}^{m} |p(x_i) - y_i|^2$$



Orthogonal Projection



Theorem 11.1. *Let $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$ and $b \in \mathbb{C}^m$ be given. A vector $x \in \mathbb{C}^n$ minimizes the residual norm $\|r\|_2 = \|b - Ax\|_2$, thereby solving the least squares problem* (11.2), *if and only if $r \perp$* range(A), *that is,*

$$A^*r = 0, \tag{11.8}$$

or equivalently,

$$A^*Ax = A^*b, \tag{11.9}$$

or again equivalently,

$$Pb = Ax, \tag{11.10}$$

where $P \in \mathbb{C}^{m\times m}$ is the orthogonal projector onto range(A). *The $n \times n$ system of equations* (11.9), *known as the* normal equations, *is nonsingular if and only if A has full rank. Consequently the solution x is unique if and only if A has full rank.*

Pseudoinverse and Normal Equations

$$A^*Ax = A^*b,$$

$$A^+ = (A^*A)^{-1}A^* \in \mathbb{C}^{n,m}.$$

$$x = A^+b$$

Least Squares via Normal Equations

$$A^*Ax = A^*b,$$

$$R^*Rx = A^*b.$$

Algorithm 11.1. Least Squares via Normal Equations

1. Form the matrix A^*A and the vector A^*b.
2. Compute the Cholesky factorization $A^*A = R^*R$.
3. Solve the lower-triangular system $R^*w = A^*b$ for w.
4. Solve the upper-triangular system $Rx = w$ for x.

$$\text{Work for Algorithm 11.1:} \quad \sim mn^2 + \frac{1}{3}n^3 \text{ flops.}$$

Least Squares via QR Factorization

$$y = Pb = \hat{Q}\hat{Q}^*b.$$

$$\hat{Q}\hat{R}x = \hat{Q}\hat{Q}^*b,$$

$$\hat{R}x = \hat{Q}^*b.$$

Algorithm 11.2. Least Squares via QR Factorization

1. Compute the reduced QR factorization $A = \hat{Q}\hat{R}$.
2. Compute the vector $\hat{Q}^*b$.
3. Solve the upper-triangular system $\hat{R}x = \hat{Q}^*b$ for x.

$$\text{Work for Algorithm 11.2:} \quad \sim 2mn^2 - \frac{2}{3}n^3 \text{ flops.}$$

Least Squares via SVD

$$y = Pb = \hat{U}\hat{U}^*b,$$

$$\hat{U}\hat{\Sigma}V^*x = \hat{U}\hat{U}^*b$$

$$\hat{\Sigma}V^*x = \hat{U}^*b.$$

Algorithm 11.3. Least Squares via SVD

1. Compute the reduced SVD $A = \hat{U}\hat{\Sigma}V^*$.
2. Compute the vector $\hat{U}^*b$.
3. Solve the diagonal system $\hat{\Sigma}w = \hat{U}^*b$ for w.
4. Set $x = Vw$.

Work for Algorithm 11.3: $\sim 2mn^2 + 11n^3$ flops,