## Orthogonality – The Gram-Schmidt algorithm

1. Two vectors $u$ and $v$ are orthogonal if $(u, v) = 0$.
2. A system of vectors $\{v_1, \ldots, v_n\}$ is orthogonal if $(v_i, v_j) = 0$ for $i \neq j$; and orthonormal if $(v_i, v_j) = \delta_{ij}$
3. A matrix is orthogonal if its columns are orthonormal

➤ Notation: $V = [v_1, \ldots, v_n]$ == matrix with column-vectors $v_1, \ldots, v_n$.

IMPORTANT: From now on, we will reserve the term unitary for square matrices. The term 'orthonormal matrix' is not used. Even 'orthogonal' is often used for square matrices.

**Problem:** Given $X = [x_1, \ldots, x_n]$, compute $Q = [q_1, \ldots, q_n]$ which is orthonormal and s.t. $\mathrm{span}(Q) = \mathrm{span}(X)$.



## The QR factorization and Least-Squares Systems

- Orthogonality
- The Gram-Schmidt and Modified Gram-Schmidt processes. Text: 5.2.7, 5.2.8
- Least-squares systems. Text: 5.3
- The Householder QR and the Givens QR. Text: 5.1, 5.2.

➤ Lines 5 and 7-8 show that

$$x_j = r_{1j}q_1 + r_{2j}q_2 + \ldots + r_{jj}q_j$$

➤ If $X = [x_1, x_2, \ldots, x_n]$, $Q = [q_1, q_2, \ldots, q_n]$, and if $R$ is the $n \times n$ upper triangular matrix

$$R = \{r_{ij}\}_{i,j=1,\ldots,n}$$

then the above relation can be written as

$$X = QR$$

➤ $R$ is upper triangular, $Q$ is orthogonal. This is called the QR factorization of $X$.

✍ What is the cost of the factorization when $X \in \mathbb{R}^{m\times n}$?



**ALGORITHM : 1. Classical Gram-Schmidt**

1. For $j = 1, \ldots, n$ Do:
2. Set $\hat{q} := x_j$
3. Compute $r_{ij} := (\hat{q}, q_i)$ , for $i = 1, \ldots, j-1$
4. For $i = 1, \ldots, j-1$ Do :
5. Compute $\hat{q} := \hat{q} - r_{ij}q_i$
6. EndDo
7. Compute $r_{jj} := \|\hat{q}\|_2$ ,
8. If $r_{jj} = 0$ then Stop, else $q_j := \hat{q}/r_{jj}$
9. EndDo

➤ All $n$ steps can be completed iff $x_1, x_2, \ldots, x_n$ are linearly independent.

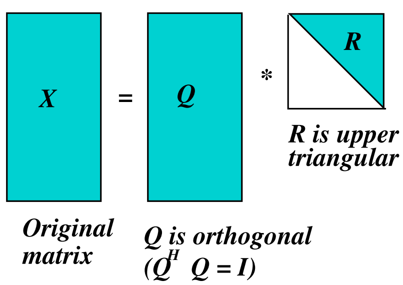
➤ **Better algorithm: Modified Gram-Schmidt.**

**ALGORITHM : 2. Modified Gram-Schmidt**

1. For $j = 1, \ldots, n$ Do:
2.     Define $\hat{q} := x_j$
3.     For $i = 1, \ldots, j-1$, Do:
4.         $r_{ij} := (\hat{q}, q_i)$
5.         $\hat{q} := \hat{q} - r_{ij} q_i$
6.     EndDo
7.     Compute $r_{jj} := \|\hat{q}\|_2$,
8.     If $r_{jj} = 0$ then Stop, else $q_j := \hat{q}/r_{jj}$
9. EndDo

**Only difference: inner product uses the accumulated sub-sum instead of original $\hat{q}$**






**Another decomposition:**

**A matrix $X$, with linearly independent columns, is the product of an orthogonal matrix $Q$ and a upper triangular matrix $R$.**

➤ Modified Gram-Schmidt algorithm is much more stable than classical Gram-Schmidt in general. [A few examples easily show this].

Suppose MGS is applied to $A$ yielding computed matrices $\hat{Q}$ and $\hat{R}$. Then there are constants $c_i$ (depending on $(m, n)$) such that

$$A + E_1 = \hat{Q}\hat{R} \qquad \|E_1\|_2 \le c_1 \, \underline{\mathbf{u}} \; \|A\|_2$$

$$\|\hat{Q}^T\hat{Q} - I\|_2 \le c_2 \, \underline{\mathbf{u}} \; \kappa_2(A) + O((\underline{\mathbf{u}}\, \kappa_2(A))^2)$$

for a certain perturbation matrix $E_1$, and there exists an orthonormal matrix $Q$ such that

$$A + E_2 = Q\hat{R} \qquad \|E_2(:, j)\|_2 \le c_3 \underline{\mathbf{u}} \, \|A(:, j)\|_2$$

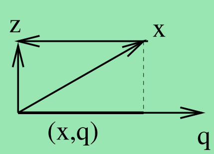
for a certain perturbation matrix $E_2$.



The operations in lines 4 and 5 can be written as

$$\hat{q} := ORTH(\hat{q}, q_i)$$

Where $ORTH(x, q)$ denotes the operation of orthogonalizing a vector $x$ against a unit vector $q$.




Result of $z = ORTH(x, q)$

*Example:* **Orthonormalize the system of vectors:**

$$X = [x_1, x_2, x_3] \;=\; \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 4 \end{pmatrix}$$

**Answer:**

$$q_1 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \;;\quad \hat{q}_2 = x_2 - (x_2, q_1) q_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} - 1 \times \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

$$\hat{q}_2 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} ; \quad q_2 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

➤ **An equivalent version:**

ALGORITHM : 3. **Modified Gram-Schmidt - 2 -**

1. For $j = 1, \ldots, n$ Do:
2. Compute $r_{jj} := \|\hat{x}_j\|_2$,
3. If $r_{jj} = 0$ then Stop, else $q_j := \hat{x}_j / r_{jj}$
4. For $i = j + 1, \ldots, n$, Do:
5. $r_{ji} := (x_i, q_j)$
6. $x_i := x_i - r_{ji} q_j$
7. EndDo
8. EndDo

➤ **Does exactly the same computation as previous algorithm, but in a different order.**

For this example: compute $Q^TQ$.

➤ Result is the identity matrix.

Recall: For any orthogonal matrix $Q$, we have

$$\boxed{Q^TQ = I}$$

(In complex case: $Q^HQ = I$).

Consequence: For an $n \times n$ orthogonal matrix $\boxed{Q^{-1} = Q^T}$. ($Q$ is unitary)



$$\hat{q}_3 = x_3 - (x_3, q_1)q_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 4 \end{pmatrix} - 2 \times \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ -2 \\ 3 \end{pmatrix}$$

$$\hat{q}_3 = \hat{q}_3 - (\hat{q}_3, q_2)q_2 = \begin{pmatrix} 0 \\ -1 \\ -2 \\ 3 \end{pmatrix} - (-1) \times \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -2.5 \\ 2.5 \end{pmatrix}$$

$$\|\hat{q}_3\|_2 = \sqrt{13} \rightarrow q_3 = \frac{\hat{q}_3}{\|\hat{q}_3\|_2} = \frac{1}{\sqrt{13}} \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -2.5 \\ 2.5 \end{pmatrix}$$

## Least-Squares systems

➤ Given: an $m \times n$ matrix $n < m$. Problem: find $x$ which minimizes:

$$\|b - Ax\|_2$$

➤ Good illustration: Data fitting.

Typical problem of data fitting: We seek an unknwon function as a linear combination $\phi$ of $n$ known functions $\phi_i$ (e.g. polynomials, trig. functions). Experimental data (not accurate) provides measures $\beta_1, \ldots, \beta_m$ of this unknown function at points $t_1, \ldots, t_m$. Problem: find the 'best' possible approximation $\phi$ to this data.

$$\phi(t) = \sum_{i=1}^{n} \xi_i \phi_i(t) \quad , \text{ s.t. } \quad \phi(t_j) \approx \beta_j, j = 1, \ldots, m$$



**Application:** another method for solving linear systems.

$$Ax = b$$

$A$ is an $n \times n$ nonsingular matrix. Compute its QR factorization.

➤ Multiply both sides by $Q^T \to Q^T Q R x = Q^T b \to$

$$Rx = Q^T b$$

Method:

➤ Compute the QR factorization of $A$, $A = QR$.

➤ Solve the upper triangular system $Rx = Q^T b$.

✍ Cost??

Define

$$F = [f_1, f_2, \ldots, f_n], \quad x = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}$$

- We want to find $x$ to minimize $\boxed{\|b - Fx\|_2}$.
- Least-squares linear system. $F$ is $m \times n$, with $m > n$.

**THEOREM.** The vector $x_*$ mininizes $\|b - Fx\|_2$ if and only if it is the solution of the normal equations:

$$F^T F x = F^T b$$



- Question: Close in what sense?
- Least-squares approximation: Find $\phi$ such that

$$\phi(t) = \sum_{i=1}^{n} \xi_i \phi_i(t), \;\; \& \;\; \sum_{j=1}^{m} |\phi(t_j) - \beta_j|^2 = \text{Min}$$

- Translated in linear algebra terms: find 'best' approximation vector to a vector $b$ from linear combinations of vectors $f_i$, $i = 1, \ldots, n$, where

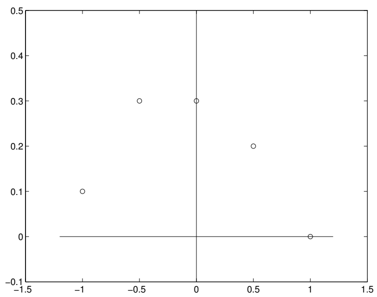
$$b = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix}, \quad f_i = \begin{pmatrix} \phi_i(t_1) \\ \phi_i(t_2) \\ \vdots \\ \phi_i(t_m) \end{pmatrix}$$

- We want to find $x = \{\xi_i\}_{i=1,\ldots,n}$ such that

$$\left\| \sum_{i=1}^{n} \xi_i f_i - b \right\|_2 \quad \text{Minimum}$$

***Example:***

| Points: | $t_1 = -1$ | $t_2 = -1/2$ | $t_3 = 0$ | $t_4 = 1/2$ | $t_5 = 1$ |
|---|---|---|---|---|---|
| Values: | $\beta_1 = 0.1$ | $\beta_2 = 0.3$ | $\beta_3 = 0.3$ | $\beta_4 = 0.2$ | $\beta_5 = 0.0$ |





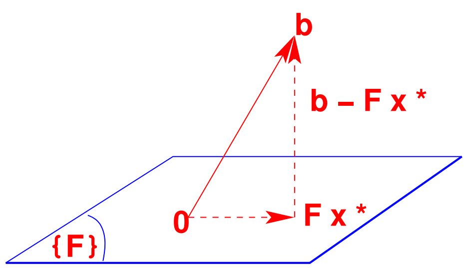
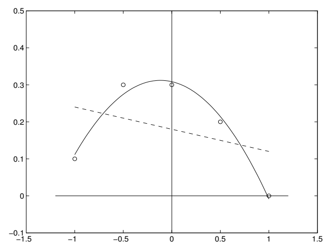




**Illustration of theorem:** $x^*$ **is the best approximation to the vector** $b$ **from the subspace** $\text{span}\{F\}$ **if and only if** $b - Fx^*$ **is** $\perp$ **to the whole subspace** $\text{span}\{F\}$**. This in turn is equivalent to** $F^T(b - Fx^*) = 0$ ➤ **Normal equations.**

## 2) Approximation by polynomials of degree 2:

- $\phi_1(t) = 1, \phi_2(t) = t, \phi_3(t) = t^2.$
- **Best polynomial found:**
  $0.3085714285 - 0.06 \ \times t \ - 0.2571428571 \ \times t^2$



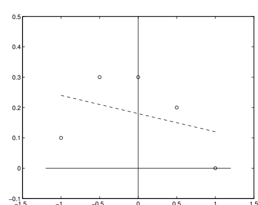


## 1) Approximations by polynomials of degree one:

- $\phi_1(t) = 1, \phi_2(t) = t.$

$$F = \begin{pmatrix} 1.0 & -1.0 \\ 1.0 & -0.5 \\ 1.0 & 0 \\ 1.0 & 0.5 \\ 1.0 & 1.0 \end{pmatrix} \qquad F^T F = \begin{pmatrix} 5.0 & 0 \\ 0 & 2.5 \end{pmatrix} \quad F^T b = \begin{pmatrix} 0.9 \\ -0.15 \end{pmatrix}$$

- **Best approximation is** $\boxed{\phi(t) = 0.18 - 0.06t}$.



## Another derivation:

- Recall: $\text{span}(Q) = \text{span}(X)$
- So $\|b - Ax\|_2$ is minimum when $b - Ax \perp \text{span}\{Q\}$
- Therefore solution $x$ must satisfy $Q^T(b - Ax) = 0 \rightarrow$

$$Q^T(b - QRx) = 0 \rightarrow Rx = Q^T b$$

$$x = R^{-1}Q^T b$$



## Use of the QR factorization

**Problem:** $Ax \approx b$ **in least-squares sense**

$A$ is an $m \times n$ (full-rank) matrix. Let

$$A = QR$$

the QR factorization of $A$ and consider the normal equations:

$$A^T Ax = A^T b \;\rightarrow\; R^T Q^T QRx = R^T Q^T b \;\rightarrow$$

$$R^T Rx = R^T Q^T b \rightarrow Rx = Q^T b$$

($R^T$ is an $n \times n$ nonsingular matrix). Therefore,

$$x = R^{-1}Q^T b$$

**Method:**

- Compute the QR factorization of $A$, $A = QR$.
- Compute the right-hand side $f = Q^T b$
- Solve the upper triangular system $Rx = f$.
- $x$ is the least-squares solution

➤ As a rule it is not a good idea to form $A^T A$ and solve the normal equations. Methods using the QR factorization are better.

✍ Total cost??

✍ Using matlab find the parabola that fits the data in previous example in L.S. sense [verify that the result found is correct.]



➤ Also observe that for any vector $w$

$$w = QQ^T w + (I - QQ^T)w$$

and that $w = QQ^T w \quad \perp \quad (I - QQ^T)w \ \rightarrow$

➤ Pythagoras theorem:

$$\|w\|_2^2 = \|QQ^T w\|_2^2 + \|(I - QQ^T)w\|_2^2$$

$$\begin{aligned}\|b - Ax\|^2 &= \|b - QRx\|^2 \\ &= \|(I - QQ^T)b + Q(Q^T b - Rx)\|^2 \\ &= \|(I - QQ^T)b\|^2 + \|Q(Q^T b - Rx)\|^2 \\ &= \|(I - QQ^T)b\|^2 + \|Q^T b - Rx\|^2\end{aligned}$$

➤ Min is reached when 2nd term of r.h.s. is zero.

## A few simple properties:

- $P$ is symmetric (real for $w$ real) – It is also unitary (for real $w$)
- In the complex case $P = I - 2ww^H$ is Hermitian and unitary.
- $P$ can be written as $P = I - \beta vv^T$ with $\beta = 2/\|v\|_2^2$, where $v$ is a multiple of $w$. [storage: $v$ and $\beta$]
- $Px$ can be evaluated $x - \beta(x^T v) \times v$ (op count?)
- Similarly: $PA = A - vz^T$ where $z^T = \beta * v^T * A$

➤ NOTE: we work in $\mathbb{R}^m$, so all vectors are of length $m$, $P$ is of size $m \times m$, etc.

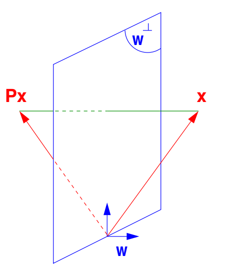


## Householder QR

➤ Householder reflectors are matrices of the form

$$P = I - 2ww^T,$$

where $w$ is a unit vector (a vector of 2-norm unity)




Geometrically, $Px$ represents a mirror image of $x$ with respect to the hyperplane $\mathrm{span}\{w\}^\perp$.

➤ Should verify that both signs work, i.e., that in both cases we indeed get $Px = \alpha e_1$ [exercise]

➤ Which sign is best? To reduce cancellation, the resulting $x - \alpha e_1$ should not be small. So, $\alpha = -\text{sign}(\xi_1)\|x\|_2$.

$$v = x + \text{sign}(\xi_1)\|x\|_2 e_1 \text{ and } \beta = 2/\|v\|_2^2$$

$$v = \begin{pmatrix} \hat{\xi}_1 \\ \xi_2 \\ \vdots \\ \xi_{m-1} \\ \xi_m \end{pmatrix} \quad \text{with} \quad \hat{\xi}_1 = \begin{cases} \xi_1 + \|x\|_2 \text{ if } \xi_1 > 0 \\ \xi_1 - \|x\|_2 \text{ if } \xi_1 \le 0 \end{cases}$$

➤ OK, but will yield a negative multiple of $e_1$ if $\xi_1 > 0$.



**Problem 1:** Given a vector $x \neq 0$, find $w$ such that

$$(I - 2ww^T)x = \alpha e_1,$$

where $\alpha$ is a (free) scalar.

Writing $(I - \beta vv^T)x = \alpha e_1$ yields

$$\beta(v^T x)\ v = x - \alpha e_1. \tag{1}$$

➤ Desired $w$ is a multiple of $x - \alpha e_1$, i.e., we can take

$$v = x - \alpha e_1$$

➤ To determine $\alpha$ we just recall that

$$\|(I - 2ww^T)x\|_2 = \|x\|_2$$

➤ As a result: $|\alpha| = \|x\|_2$, or

$$\alpha = \pm\|x\|_2$$

**Alternative:**

- Define $\sigma = \sum_{i=2}^{m} \xi_i^2$.
- Always set $\hat{\xi}_1 = \xi_1 - \|x\|_2$. Update OK when $\xi_1 \leq 0$
- When $\xi_1 > 0$ compute $\hat{x}_1$ as

$$\hat{\xi}_1 = \xi_1 - \|x\|_2 = \frac{\xi_1^2 - \|x\|_2^2}{\xi_1 + \|x\|_2} = \frac{-\sigma}{\xi_1 + \|x\|_2}$$

So: $$\hat{\xi}_1 = \begin{cases} \frac{-\sigma}{\xi_1 + \|x\|_2} & \text{if } \xi_1 > 0 \\ \xi_1 - \|x\|_2 & \text{if } \xi_1 \leq 0 \end{cases}$$

- It is customary to compute a vector $v$ such that $v_1 = 1$. So $v$ is scaled by its first component.
- If $\sigma$ is zero, procedure will return $v = [1; x(2:m)]$ and $\beta = 0$.
- Matlab function:

.. Show that $(I - \beta v v^T)x = \alpha e_1$ when $v = x - \alpha e_1$ and $\alpha = \pm\|x\|_2$.

- Equivalent to showing that

$$x - (\beta x^T v)v = \alpha e_1 \leftrightarrow x - \alpha e_1 = (\beta x^T v)v$$

but recall that $v = x - \alpha e_1$ so we need to show that

$$\beta x^T v = 1 \quad \text{i.e., that} \quad \frac{2x^T v}{\|x - \alpha e_1\|_2^2} = 1$$

- Denominator $= \|x\|_2^2 + \alpha^2 - 2\alpha e_1^T x = 2(\|x\|_2^2 - \alpha e_1^T x)$
- Numerator $= 2x^T v = 2x^T(x - \alpha e_1) = 2(\|x\|_2^2 - \alpha x^T e_1)$

Numerator/ Denominator = 1. Done

## Problem 2: Generalization.

Given an $m \times n$ matrix $X$, find $w_1, w_2, \ldots, w_n$ such that

$$(I - 2w_n w_n^T) \cdots (I - 2w_2 w_2^T)(I - 2w_1 w_1^T)X = R$$

where $r_{ij} = 0$ for $i > j$

➤ First step is easy : select $w_1$ so that the first column of $X$ becomes $\alpha e_1$

➤ Second step: select $w_2$ so that $x_2$ has zeros below 2nd component.

➤ etc.. After $k - 1$ steps: $X_k \equiv P_{k-1} \ldots P_1 X$ has the following shape:



```
function [v,bet] = house (x)
%% computes the householder vector for x
m = length(x);
v = [1 ; x(2:m)];
sigma = v(2:m)' * v(2:m);
if (sigma == 0)
   bet = 0;
else
   xnrm = sqrt(x(1)^2 + sigma) ;
   if (x(1) <= 0)
      v(1) = x(1) - xnrm;
   else
      v(1) = -sigma / (x(1) + xnrm) ;
   end
   bet = 2 / (1+sigma/v(1)^2);
   v = v/v(1) ;
end
```

➤ To leave the first $k-1$ columns unchanged $w$ must have zeros in positions 1 through $k-1$.

$$P_k = I - 2w_k w_k^T, \quad w_k = \frac{v}{\|v\|_2},$$

where the vector $v$ can be expressed as a Householder vector for a shorter vector using the matlab function house,

$$v = \begin{pmatrix} 0 \\ house(X(k:m,k)) \end{pmatrix}$$

➤ The result is that work is done on the $(k:m, k:n)$ submatrix.



$$X_k = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & \cdots & \cdots & x_{1n} \\ & x_{22} & x_{23} & \cdots & \cdots & \cdots & x_{2n} \\ & & x_{33} & \cdots & \cdots & \cdots & x_{3n} \\ & & & \ddots & \cdots & \cdots & \vdots \\ & & & & x_{kk} & \cdots & \vdots \\ & & & & x_{k+1,k} & \cdots & x_{k+1,n} \\ & & & & \vdots & \vdots & \vdots \\ & & & & x_{m,k} & \cdots & x_{m,n} \end{pmatrix}.$$

➤ To do: transform this matrix into one which is upper triangular up to the $k$-th column...

➤ ... while leaving the previous columns untouched.

Yields the factorization:

$$X = QR$$

where

$$Q = P_1 P_2 \dots P_n \text{ and } R = X_n$$

MAJOR difference with Gram-Schmidt: $Q$ is $m \times m$ and $R$ is $m \times n$ (same as $X$). The matrix $R$ has zeros below the $n$-th row. Note also : this factorization always exists.

Cost of Householder QR? Compare with Gram-Schmidt

**Question:** How to obtain $X = Q_1 R_1$ where $Q_1 =$ same size as $X$ and $R_1$ is $n \times n$ (as in MGS)?



ALGORITHM : 4. **Householder QR**

For $k = 1 : n$ do<br>
    $[v, \beta] = house(X(k : m, k))$<br>
    $X(k : m, k : n) = (I - \beta v v^T) X(k : m, k : n)$<br>
    If $(k < m)$<br>
        $X(k + 1 : m, k) = v(2 : m - k + 1)$<br>
    end<br>
end

➤ In the end:

$$X_n = P_n P_{n-1} \dots P_1 X = \text{upper triangular}$$

## The rank-deficient case

➤ Result of Householder QR: $Q_1$ and $R_1$ such that $Q_1 R_1 = X$. In the rank-deficient case, can have $\text{span}\{Q_1\} \neq \text{span}\{X\}$ because $R_1$ may be singular.

➤ Remedy: Householder QR with column pivoting. Result will be:

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix}$$

➤ $R_{11}$ is nonsingular. So rank$(X)$ = size of $R_{11}$ = rank$(Q_1)$ and $Q_1$ and $X$ span the same subspace.

➤ $\Pi$ permutes columns of $X$.



**Answer:** simply use the partitioning

$$X = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \quad \rightarrow \quad X = Q_1 R_1$$

➤ Referred to as the "thin" QR factorization (or "economy-size QR" factorization in matlab)

➤ How to solve a least-squares problem $Ax = b$ using the Householder factorization?

➤ Answer: no need to compute $Q_1$. Just apply $Q^T$ to $b$.

➤ This entails applying the successive Householder reflections to $b$

## Properties of the QR factorization

Consider the 'thin' factorization $A = QR$, (size($Q$) = [m,n] = size ($A$)). Assume $r_{ii} > 0$, $i = 1, \ldots, n$

1. When $A$ is of full column rank this factorization exists and is unique

2. It satisfies:

$$\text{span}\{a_1, \cdots, a_k\} = \text{span}\{q_1, \cdots, q_k\}, \quad k = 1, \ldots, n$$

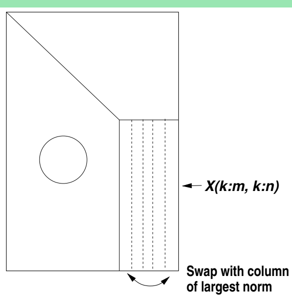
3. $R$ is identical with the Cholesky factor $G^T$ of $A^TA$.

➤ When $A$ in rank-deficient and Householder with pivoting is used, then

$$Ran\{Q_1\} = Ran\{A\}$$



**Algorithm:** At step $k$, active matrix is $X(k:m, k:n)$. Swap $k$-th column with column of largest 2-norm in $X(k:m, k:n)$. If all the columns have zero norm, stop.




**Practical Question:** how to implement this ???

**Main idea of Givens rotations** consider $y = Gx$ then

$$y_i = c * x_i + s * x_k$$
$$y_k = -s * x_i + c * x_k$$
$$y_j = x_j \quad \text{for} \quad j \neq i, k$$

- Can make $y_k = 0$ by selecting

$$s = x_k/t; \quad c = x_i/t; \quad t = \sqrt{x_i^2 + x_k^2}$$

- This is used to introduce zeros in the first column of a matrix $A$ (for example $G(m-1, m)$, $G(m-2, m-1)$ etc..$G(1, 2)$ )..
- See text for details



## Givens Rotations

- Matrices of the form

$$G(i,k,\theta) = \begin{array}{cc} \begin{pmatrix} 1 & \dots & 0 & & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \dots & 0 & & \dots & & 1 \end{pmatrix} & \begin{array}{c} \\ \\ i \\ \\ k \\ \\ \\ \end{array} \end{array}$$

with $c = \cos\theta$ and $s = \sin\theta$

- represents a rotation in the span of $e_i$ and $e_k$.

**Proof.** (a), (b) are trivial

**(c):** Clearly $Ran(P) = \{x|\ x = QQ^Ty, y \in \mathbb{R}^m\} \subseteq \mathcal{X}$. Any $x \in \mathcal{X}$ is of the form $x = Qy, y \in \mathbb{R}^m$. Take $Px = QQ^T(Qy) = Qy = x$. Since $x = Px$, $x \in Ran(P)$ so $\mathcal{X} \subseteq Ran(P)$. In the end $\mathcal{X} = Ran(P)$.

**(d):** Need to show inclusion both ways.

- $x \in Null(P) \leftrightarrow Px = 0 \leftrightarrow (I-P)x = x \rightarrow x \in Ran(I-P)$
- $x \in Ran(I-P) \leftrightarrow \exists y \in \mathbb{R}^m | x = (I-P)y \rightarrow Px = P(I-P)y = 0 \rightarrow x \in Null(P)$

**(e):** $x \in \mathcal{X}^\perp \leftrightarrow (x,y) = 0, \forall y \in \mathcal{X} \leftrightarrow (x, Qz) = 0, \forall z \in \mathbb{R}^r \leftrightarrow (Q^Tx, z) = 0, \forall z \in \mathbb{R}^r \leftrightarrow Q^Tx = 0 \leftrightarrow QQ^Tx = 0 \leftrightarrow Px = 0$



## Orthogonal projectors and subspaces

Notation: Given a supspace $\mathcal{X}$ or $\mathbb{R}^m$ define

$$\mathcal{X}^\perp = \{y\ |y \perp x, \quad \forall\ x\ \in \mathcal{X}\}$$

➤ Let $Q = [q_1, \cdots, q_r]$ an orthonormal basis of $\mathcal{X}$

✍ How would you obtain such a basis?

➤ Then define orthogonal projector $P = QQ^T$

**Properties**

(a) $P^2 = P$ (b) $(I-P)^2 = I - P$
(c) $Ran(P) = \mathcal{X}$ (d) $Ran(I-P) = Null(P)$
(e) $Null(P) = \mathcal{X}^\perp$ ( $= Ran(I-P)$)

➤ Note that (b) means that $I - P$ is also a projector

## Four fundamental supspaces - URV decomposition

Let $A \in \mathbb{R}^{m\times n}$ and consider $\mathrm{Ran}(A)^{\perp}$

**Property 1:** $\mathrm{Ran}(A)^{\perp} = Null(A^T)$

Proof: $x \in \mathrm{Ran}(A)^{\perp}$ iff $(Ay, x) = 0$ for all $y$ iff $(y, A^T x) = 0$ for all $y$ ...

**Property 2:** $\mathrm{Ran}(A^T) = Null(A)^{\perp}$

- Take $\mathcal{X} = \mathrm{Ran}(A)$ in orthogonal decomoposition
- Result:

$$\mathbb{R}^m = Ran(A) \oplus Null(A^T)$$
$$\mathbb{R}^n = Ran(A^T) \oplus Null(A)$$

4 fundamental subspaces

$Ran(A)$ $Null(A)$,
$Ran(A^T)$ $Null(A^T)$



## Orthogonal decomposition

**Result:** Any $x \in \mathbb{R}^m$ can be written in a unique way as

$$x = x_1 + x_2, \quad x_1 \in \mathcal{X}, \quad x_2 \in \mathcal{X}^{\perp}$$

- Just set $x_1 = Px, \quad x_2 = (I - P)x$
- In other words $\mathbb{R}^m = P\mathbb{R}^m \oplus (I - P)\mathbb{R}^m$ or
$$\mathbb{R}^m = Ran(P) \oplus Ran(I - P)$$
$$\mathbb{R}^m = Ran(P) \oplus Null(P)$$
- Can complete basis $\{q_1, \cdots, q_r\}$ into orthonormal basis of $\mathbb{R}^m$, $q_{r+1}, \cdots, q_m$
- $\{q_{r+1}, q_{r+2}, \cdots, q_m\}$ = basis of $\mathcal{X}^{\perp}$. $\rightarrow$
$$dim(\mathcal{X}^{\perp}) = m - r.$$

- Far from unique.

✍ Show how you can get a decomposition in which $C$ is lower (or upper) triangular, from the above factorization.

- Can select decomposition so that $R$ is upper triangular $\rightarrow$ URV decomposition.

- Can select decomposition so that $R$ is lower triangular $\rightarrow$ ULV decomposition.

- SVD = special case of URV where $R$ = diagonal

✍ How can you get the ULV decomposition by using only the Householder QR factorizatoin (possibly with pivoting)? [Hint: you must use Householder twice]



- Express the above with bases for $\mathbb{R}^m$ :

$$[\underbrace{u_1, u_2, \cdots, u_r}_{Ran(A)}, \underbrace{u_{r+1}, u_{r+2}, \cdots, u_m}_{Null(A^T)}]$$

and for $\mathbb{R}^n$

$$[\underbrace{v_1, v_2, \cdots, v_r}_{Ran(A^T)}, \underbrace{v_{r+1}, v_{r+2}, \cdots, v_n}_{Null(A)}]$$

- Observe $u_i^T A v_j = 0$ for $i > r$ or $j > r$. Therefore

$$U^T A V = R = \begin{pmatrix} C & 0 \\ 0 & 0 \end{pmatrix}_{m \times n} \quad C \in \mathbb{R}^{r \times r} \quad \longrightarrow$$

$$\boxed{A = URV^T}$$

- General class of URV decompositions