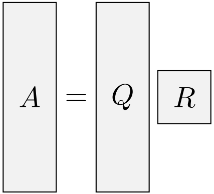
QR factorisation

For any $A \in \mathbb{C}^{m\times n}$, $\exists$ factorisation



$Q \in \mathbb{R}^{m\times n}$: orthonormal, $R \in \mathbb{R}^{n\times n}$: upper triangular

- Many algorithms available: Gram-Schmidt, Householder, CholeskyQR, ...
- various applications: least-squares, orthogonalisation, computing SVD, manifold retraction...
- With Householder, pivoting $A = QRP$ not needed for numerical stability
  - but pivoting gives rank-revealing QR (nonexaminable)

QR via Gram-Schmidt

Gram-Schmidt: Given $A = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^{m \times n}$ (assume full rank rank$(A) = n$), find orthonormal $[q_1, \ldots, q_n]$ s.t. span$(q_1, \ldots, q_n) =$ span$(a_1, \ldots, a_n)$

G-S process: $q_1 = \frac{a_1}{\|a_1\|}$, then $\tilde{q}_2 = a_2 - q_1 q_1^T a_2$, $q_2 = \frac{\tilde{q}_2}{\|\tilde{q}_2\|}$,
repeat for $j = 3, \ldots, n$: $\tilde{q}_j = a_j - \sum_{i=1}^{j-1} q_i q_i^T a_j$, $q_j = \frac{\tilde{q}_j}{\|\tilde{q}_j\|}$.

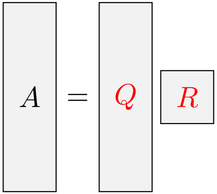
QR via Gram-Schmidt

Gram-Schmidt: Given $A = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^{m \times n}$ (assume full rank $\text{rank}(A) = n$), find orthonormal $[q_1, \ldots, q_n]$ s.t. $\text{span}(q_1, \ldots, q_n) = \text{span}(a_1, \ldots, a_n)$

G-S process: $q_1 = \frac{a_1}{\|a_1\|}$, then $\tilde{q}_2 = a_2 - q_1 q_1^T a_2$, $q_2 = \frac{\tilde{q}_2}{\|\tilde{q}_2\|}$,
repeat for $j = 3, \ldots, n$: $\tilde{q}_j = a_j - \sum_{i=1}^{j-1} q_i q_i^T a_j$, $q_j = \frac{\tilde{q}_j}{\|\tilde{q}_j\|}$.

This gives QR! Let $r_{ij} = q_i^T a_j$ $(i \neq j)$ and $r_{jj} = \|a_j - \sum_{i=1}^{j-1} r_{ij} q_i\|$,

$$q_1 = \frac{a_1}{r_{11}}$$
$$q_2 = \frac{a_2 - r_{12} q_1}{r_{22}} \quad \Leftrightarrow \quad \begin{aligned} a_1 &= r_{11} q_1 \\ a_2 &= r_{12} q_1 + r_{22} q_2 \\ a_j &= r_{1j} q_1 + r_{2j} q_2 + \cdots + r_{jj} q_j \end{aligned} \quad \Leftrightarrow$$
$$q_j = \frac{a_j - \sum_{i=1}^{j-1} r_{ij} q_i}{r_{jj}}$$

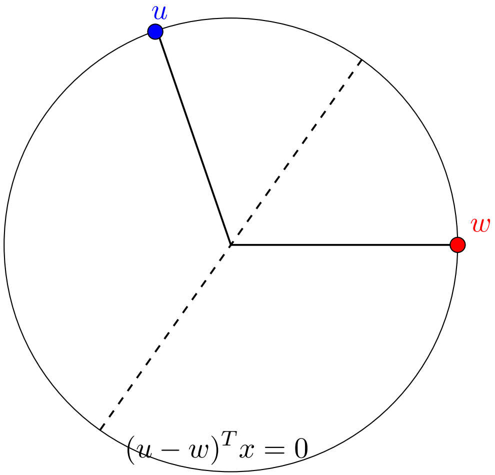


- But this isn't the recommended way to do QR; numerically unstable

Householder reflectors

$$H = I - 2vv^T, \qquad \|v\| = 1$$

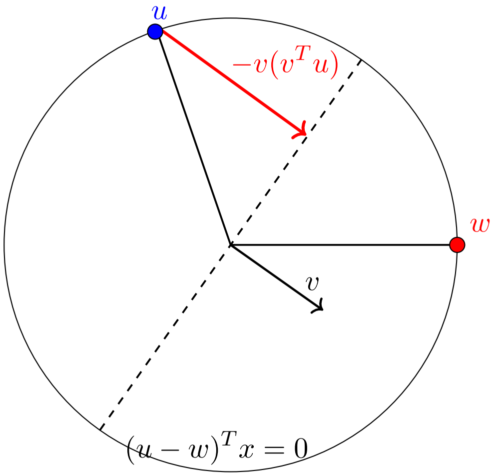
- H orthogonal and symmetric: $H^T H = H^2 = I$, eigvals 1 ($n-1$ copies) and -1 (1 copy)
- For any given $u, w \in \mathbb{R}^n$ s.t. $\|u\| = \|w\|$ and $u \neq v$, $H = I - 2vv^T$ with $v = \frac{w-u}{\|w-u\|}$ gives $Hu = w$ ($\Leftrightarrow u = Hw$, thus 'reflector')
- We'll use this mostly for $w = [*, 0, 0, \ldots, 0]^T$



Householder reflectors

$$H = I - 2vv^T, \qquad \|v\| = 1$$

- H orthogonal and symmetric: $H^T H = H^2 = I$, eigvals 1 ($n-1$ copies) and -1 (1 copy)
- For any given $u, w \in \mathbb{R}^n$ s.t. $\|u\| = \|w\|$ and $u \neq v$, $H = I - 2vv^T$ with $v = \frac{w-u}{\|w-u\|}$ gives $Hu = w$ ($\Leftrightarrow u = Hw$, thus 'reflector')
- We'll use this mostly for $w = [*, 0, 0, \ldots, 0]^T$



Householder reflectors for QR

Householder reflectors:

$$H = I - 2vv^T, \qquad v = \frac{x - \|x\|_2 e}{\|x - \|x\|_2 e\|_2}, \qquad e = [1, 0, \ldots, 0]^T$$

satisfies $Hx = [\|x\|, 0, \ldots, 0]^T$

Householder reflectors for QR

Householder reflectors:

$$H = I - 2vv^T, \qquad v = \frac{x - \|x\|_2 e}{\|x - \|x\|_2 e\|_2}, \qquad e = [1, 0, \ldots, 0]^T$$

satisfies $Hx = [\|x\|, 0, \ldots, 0]^T$

$\Rightarrow$ To do QR, find H_1 s.t. $H_1 a_1 = \begin{bmatrix} \|a_1\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$,

repeat to get $H_n \cdots H_2 H_1 A = R$ upper triangular, then
$A = (H_1 \cdots H_{n-1} H_n) R = QR$

Householder QR factorisation, diagram

$$A = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}$$

Apply sequence of Householder reflectors

$$H_1 A = (I - 2v_1 v_1^T)A = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{bmatrix}, \qquad H_2 H_1 A = (I - 2v_2 v_2^T) H_1 A = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \\ & & * & * \end{bmatrix},$$

$$H_3 H_2 H_1 A = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \\ & & & * \end{bmatrix}, \qquad H_n \cdots H_3 H_2 H_1 A = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \\ & & & \end{bmatrix},$$

Note $v_k = [\underbrace{0, 0, \ldots, 0}_{k-1 \text{ 0's}}, *, *, \ldots, *]^T$

Householder QR factorisation

$$H_n \cdots H_2 H_1 A = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \\ & & & \\ & & & \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$\Leftrightarrow A = (H_1^T \cdots H_{n-1}^T H_n^T) \begin{bmatrix} R \\ 0 \end{bmatrix} =: Q_F \begin{bmatrix} R \\ 0 \end{bmatrix}$ (**full** QR; Q_F is square orthogonal)

Writing $Q_F = [Q \ Q_\perp]$ where $Q \in \mathbb{R}^{m \times n}$ orthonormal, $A = QR$ (**'thin'** QR or just QR)

Properties

- Cost $\frac{4}{3}n^3$ flops with Householder-QR (twice that of LU)
- Unconditionally backward stable: $\hat{Q}\hat{R} = A + \Delta A$, $\|\hat{Q}^T\hat{Q} - I\|_2 = \epsilon$ (next lec)
- Constructive proof for $A = QR$ existence
- To solve $Ax = b$, solve $Rx = Q^T b$ via triangle solve.

  $\rightarrow$ Excellent method, but twice slower than LU (so rarely used)

Givens rotation

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1$$

Designed to 'zero' one element at a time. E.g. QR for upper Hessenberg matrix

$$A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \\ & & & * & * \end{bmatrix}, \quad G_1 A = \begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \\ & & & * & * \end{bmatrix}, G_2 G_1 A = \begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \\ & & * & * & * \\ & & & * & * \end{bmatrix},$$

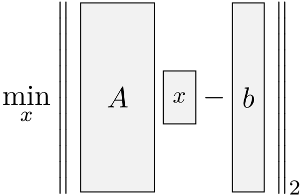
$$G_3 G_2 G_1 A = \begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \\ & & & * & * \\ & & & * & * \end{bmatrix}, G_4 G_3 G_2 G_1 A = \begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & & * & * & * \\ & & & * & * \\ & & & & * \end{bmatrix} =: R$$

$\Leftrightarrow A = G_1^T G_2^T G_3^T G_4^T R$ is the QR factorisation.

- G acts locally on two rows (two columns if right-multiplied)
- Non-neighboring rows/cols allowed

Least-squares problem

Given $A \in \mathbb{R}^{m\times n}, m \geq n$ and $b \in \mathbb{R}^m$, find $x \in \mathbb{R}^n$ s.t.



- More data than degrees of freedom
- 'Overdetermined' linear system; $Ax = b$ usually impossible
- Thus minimise $\|Ax - b\|$; usually $\|Ax - b\|_2$ but sometimes e.g. $\|Ax - b\|_1$ of interest (we focus on $\|Ax - b\|_2$)
- Assume full rank rank$(A) = n$; this makes solution unique

Least-squares problem via QR

$$\min_x \|Ax - b\|_2, \qquad A \in \mathbb{R}^{m \times n}, m \geq n$$

Least-squares problem via QR

$$\min_x \|Ax - b\|_2, \qquad A \in \mathbb{R}^{m\times n}, m \geq n$$

Let $A = [Q\ Q_\perp]\left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right] = Q_F\left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right]$ be 'full' QR factorization. Then

$$\|Ax - b\|_2 = \|Q_F^T(Ax - b)\|_2 = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} x - \begin{bmatrix} Q^T b \\ Q_\perp^T b \end{bmatrix} \right\|_2$$

so $x = R^{-1}Q^T b$ is the solution. This also gives algorithm:

Least-squares problem via QR

$$\min_x \|Ax-b\|_2, \qquad A\in\mathbb{R}^{m\times n}, m\geq n$$

Let $A=[Q\ Q_\perp]\left[\begin{smallmatrix}R\\0\end{smallmatrix}\right]=Q_F\left[\begin{smallmatrix}R\\0\end{smallmatrix}\right]$ be 'full' QR factorization. Then

$$\|Ax-b\|_2=\|Q_F^T(Ax-b)\|_2=\left\|\begin{bmatrix}R\\0\end{bmatrix}x-\begin{bmatrix}Q^Tb\\Q_\perp^Tb\end{bmatrix}\right\|_2$$

so $x=R^{-1}Q^Tb$ is the solution. This also gives algorithm:

1. Compute **thin** QR factorization $A=QR$
2. Solve linear system $Rx=Q^Tb$.

Least-squares problem via QR

$$\min_x \|Ax-b\|_2, \qquad A\in\mathbb{R}^{m\times n}, m\geq n$$

Let $A = [Q\ Q_\perp]\left[\begin{smallmatrix}R\\0\end{smallmatrix}\right] = Q_F\left[\begin{smallmatrix}R\\0\end{smallmatrix}\right]$ be 'full' QR factorization. Then

$$\|Ax-b\|_2 = \|Q_F^T(Ax-b)\|_2 = \left\|\begin{bmatrix}R\\0\end{bmatrix}x - \begin{bmatrix}Q^Tb\\Q_\perp^Tb\end{bmatrix}\right\|_2$$

so $x = R^{-1}Q^Tb$ is the solution. This also gives algorithm:

1. Compute **thin** QR factorization $A = QR$
2. Solve linear system $Rx = Q^Tb$.

- This is backward stable: computed $\hat{x}$ solution for $\min_x \|(A+\Delta A)x+(b+\Delta b)\|_2$ (see Higham's book Ch.20)
- Unlike square system $Ax=b$, one really needs QR: LU won't do the job

Normal equation: Cholesky-based least-squares solver

$$\min_x \|Ax - b\|_2, \qquad A \in \mathbb{R}^{m\times n}, m \geq n$$

$x = R^{-1}Q^T b$ is the solution $\Leftrightarrow$ x solution for $n \times n$ **normal equation**

$$(A^T A)x = A^T b$$

- $A^T A \succeq 0$ (always) and $A^T A \succ 0$ if $\text{rank}(A) = n$; then PD linear system; use Cholesky to solve.
- Fast! but NOT backward stable; $\kappa_2(A^T A) = (\kappa_2(A))^2$ where $\kappa_2(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$ **condition number** (next lecture)

Application: regression/function approximation

Given function $f : [-1, 1] \to \mathbb{R}$,

Consider approximating via polynomial $f(x) \approx p(x) = \sum_{i=0} c_i x^i$.

Very common technique: **Regression**

1. Sample f at points $\{z_i\}_{i=1}^m$, and
2. Find coefficients c defined by Vandermonde system $Ac \approx f$,

$$\begin{bmatrix} 1 & z_1 & \cdots & z_1^n \\ 1 & z_2 & \cdots & z_2^n \\ \vdots & \vdots & & \vdots \\ 1 & z_m & \cdots & z_m^n \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix} \approx \begin{bmatrix} f(z_1) \\ f(z_2) \\ \vdots \\ f(z_m) \end{bmatrix}.$$

- Numerous applications, e.g. in statistics, numerical analysis, approximation theory, data analysis!