

Utility for solving a consistent system $A \vec{x} = \vec{b}$

Suppose we have the task of solving $A \vec{x} = \vec{b}$ for multiple \vec{b} vectors

Suppose A is $m \times n$ with rank r

1. we compute $A = QR$ once so that

$$QR \vec{x} = \vec{b}$$

2. left-multiply both sides by Q^T :

$$Q^TQR \vec{x} = Q^T \vec{b}$$

3. from $Q^T Q = I_r$ we conclude:

$$R \vec{x} = Q^T \vec{b} \text{ (note we do not need to invert } Q)$$

4. computation for any new \vec{b} reduces to

- matrix-vector multiplication
- solving an upper-triangular system

As we showed before, QR separates transformation by A into

- R : pure deformation inside $\text{Col}(A)$
- Q : rotation or reflection that positions that deformation in \mathbb{R}^m

When solving $A \vec{x} = \vec{b}$, multiplying by Q^T removes the rotational part, reducing the problem to solving only the deformation $R \vec{x} = Q^T \vec{b}$

Suppose A is rank-deficient ($r < n$)

then R is $r \times n$ and the system $R \vec{x} = Q^T \vec{b}$

represents r independent equations in n unknowns

The original system $A \vec{x} = \vec{b}$ is still solvable
if (and only if) \vec{b} lies in $\text{Col}(A)$



QRF vs row reduction

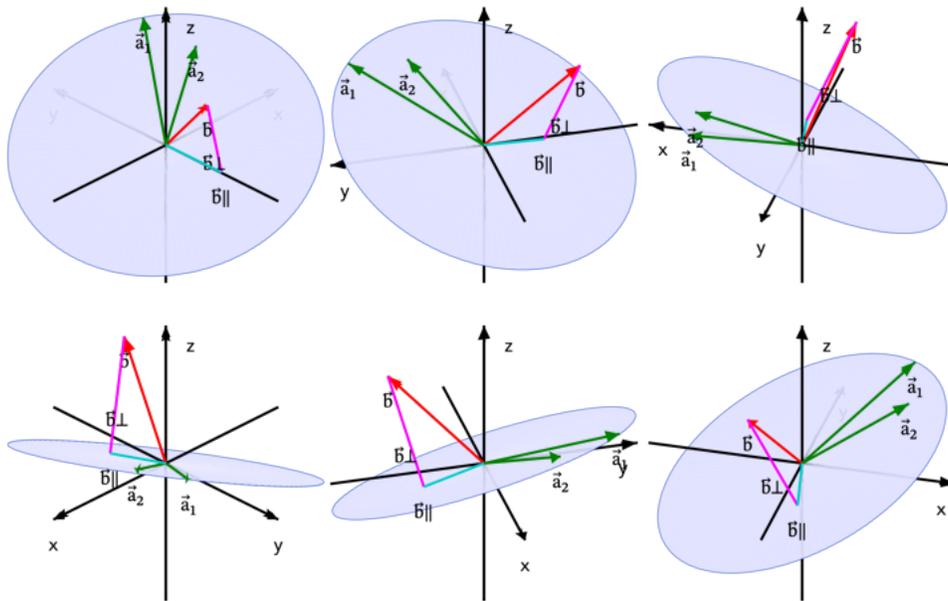
	REF & RREF matrix	Q matrix in QRF
Target	Rows of A	columns of A
Tools	<ul style="list-style-type: none"> • Subtracts multiples of rows • Scaling is not required 	<ul style="list-style-type: none"> • Subtracts precisely chosen multiples $(a_i, q_j) q_j$ to eliminate directional components & achieve orthogonality • Always scales to normalize
Goal	Subtraction is done in a way that guarantees linearly independent pivot rows	Subtraction is done in a way that guarantees orthogonal columns (orthogonality \Rightarrow independence)
Row space	Preserves by replacing rows with linear combinations of other rows	Alters
Column space	Alters	Preserves by replacing columns with linear combinations of other columns
Solution set	Preserves	Solution set transfers to R via $R \vec{x} = Q^T \vec{b}$
Geometry	Alters	<p style="text-align: center;">Preserves by</p> <ul style="list-style-type: none"> • requiring orthogonality • normalizing
Numerical stability	<p style="text-align: center;">May amplify rounding errors: subtracting nearly parallel rows</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">producing very small remainders</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">producing small pivots for later steps</p>	While numerical errors may arise when adjacent columns are nearly parallel, errors do not affect later columns



Utility for solving an inconsistent system $A \vec{x} = \vec{b}$

Compare two examples of solving an inconsistent system $A \vec{x} = \vec{b}$ where

- $A = [\vec{a}_1 \mid \vec{a}_2]$
- \vec{a}_1 & \vec{a}_2 are vectors in \mathbb{R}^3 , linearly independent, but not orthogonal
- \vec{b} cannot be written as a linear combination of \vec{a}_1 & \vec{a}_2



① As shown in the 'Four Subspaces' chapter, we can

1. rewrite the system as

$$A \vec{x} = \vec{b}_{||} + \vec{b}_{\perp} \text{ where}$$

- \vec{b}_{\perp} is orthogonal to vectors \vec{a}_1 and \vec{a}_2
- $\vec{b}_{||} \in \text{col}(A)$

2. left-multiply both sides by A^T to obtain

$$A^T A \vec{x} = A^T \vec{b}_{||} + A^T \vec{b}_{\perp}$$

3. recall that

- \vec{b}_{\perp} is orthogonal to vectors \vec{a}_1 and \vec{a}_2

\Leftrightarrow

$$\vec{b}_{\perp} \in \ell\text{-null}(A)$$

- ℓ -null(A) is equivalent to $\text{null}(A^T)$

\Leftrightarrow

$$A^T \vec{b}_\perp = \vec{0}$$

4. substitute it into

$$A^T A \vec{x} = A^T \vec{b}_\parallel + A^T \vec{b}_\perp = A^T \vec{b}_\parallel$$

5. solve $A^T A \vec{x} = A^T \vec{b}_\parallel$ as

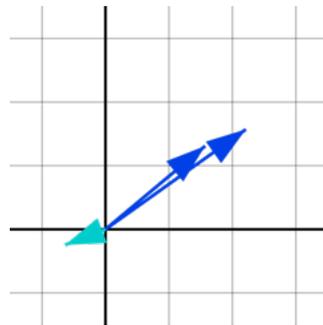
$$\vec{x} = (A^T A)^{-1} A^T \vec{b}$$

So far, this is a recap of what we have seen

Next, we solve $A^T A \vec{x} = A^T \vec{b}$ from the standard 'point of view':

\mathbb{R}^2 with standard canonical basis

- the two columns of the 2×2 matrix $A^T A$ are drawn in blue (note that they are not the same as vectors \vec{a}_1 and \vec{a}_2 seen in 3D)
- the target 2D vector $A^T \vec{b}$ is drawn in cyan
- solving the equation means finding coefficients x to represent the $A^T \vec{b}$ as a linear combination of the columns of $A^T A$



- ② The alternative and the preferred way to solve it is to perform QR factorization on A first

Assume A has full column rank and $A = QR$ where

- Q has orthonormal columns, so $Q^T Q = I_r$
- R is $r \times r$ upper triangular

1. start from $A^T A \vec{x} = A^T \vec{b}$

$$\vec{x} = (A^T A)^{-1} A^T \vec{b}$$

2. substitute $A = Q R$

3. obtain

$$A^T A = (Q R)^T (Q R) = (R^T Q^T)(Q R) = R^T (Q^T Q) R = R^T R$$

4. obtain

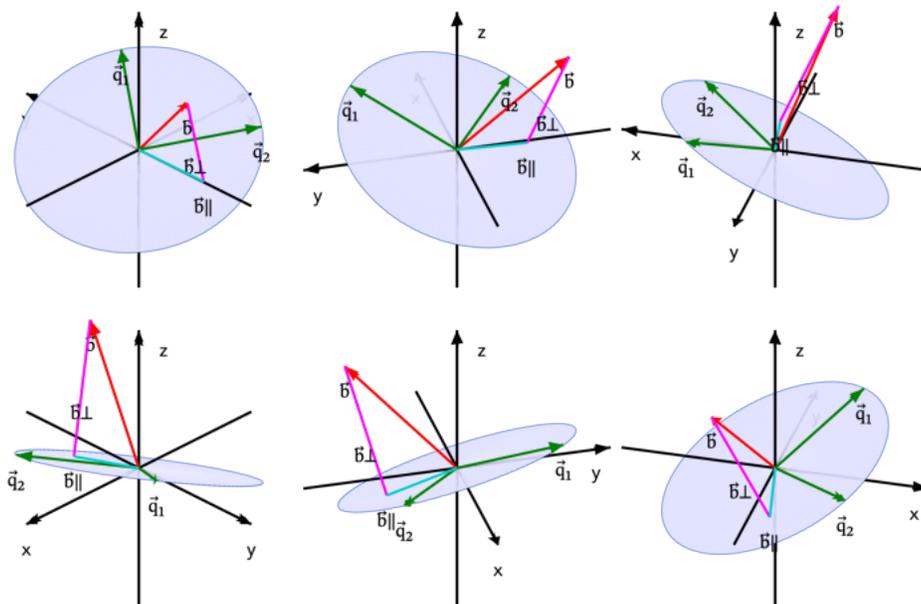
$$A^T \vec{b} = (Q R)^T \vec{b} = R^T Q^T \vec{b}$$

we arrive at

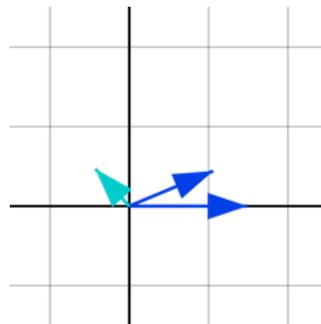
$$R \vec{x} = Q^T \vec{b}$$

in place of

$$A^T A \vec{x} = A^T \vec{b}$$



- the orthogonal columns of the 2×3 matrix Q on the 3D image are drawn in green
- the target 3D vector $\vec{b}_{||} = Q Q^T \vec{b}$ is drawn in cyan



- the columns of the upper-triangular 2×2 matrix R on the 2D image are drawn in blue
 - note that they are not the same vectors as \vec{q}_1 and \vec{q}_2
 - also note the 'hierarchy' of R with first vector being aligned with \vec{e}_1
- the target 2D vector $Q^T \vec{b}$ (different from the 3D vector $b_{\parallel} = Q Q^T \vec{b}$) is drawn in cyan

Unlike the previous system,

$$R \vec{x} = Q^T \vec{b}$$

is solved by back-substitution, which is computationally efficient and more numerically stable

(the discussion of numerical stability is beyond the scope of this chapter)



Normal equations vs QR method

Both solve the same inconsistent system:

$$A \vec{x} = \vec{b}$$

by left-multiplying both sides by A^T :

$$A^T A \vec{x} = A^T \vec{b}$$

$$(r = \text{rank}(A))$$

Normal equations	QR method
Multiplies by A^T explicitly	Factors $A = Q R$ first to solve simplified system
Solves as $A^T A \vec{x} = A^T \vec{b}$	Solves as $R \vec{x} = Q^T \vec{b}$
Solves symmetric $r \times r$ system	Solves an upper-triangular $r \times r$ system by back-substitution
Distorts left-side by $A^T A$	Isolates distortion in R
Distorts right-side by A^T	Multiplies right-side by Q^T without distortion

