GIAN Course on Solving Linear Systems and Computing Generalized Inverses Using Recurrent Neural Networks

June 09-19, 2025, IIT Indore,

(The Least Squares Problem and QR Decompostion)

Overview

Topics Covered:

1. Description of the Least Squares Problem
2. Rotators, Reflectors, and the QR Decomposition
3. Solving Least Squares via QR Decomposition
4. Gram-Schmidt Orthonormalization
    - Relation to QR decomposition
    - Computational variants
5. Theoretical Foundations (postponed for clarity)
6. Updating the QR Decomposition
    - When rows/columns are added or deleted

Motivation

Why Least Squares?

- Many real-world problems involve inconsistent systems of equations
- We seek an approximate solution that minimizes the error
- Leads to the problem:

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

Motivation

Why Least Squares?

- Many real-world problems involve inconsistent systems of equations
- We seek an approximate solution that minimizes the error
- Leads to the problem:

$$\min_{x} \|Ax - b\|_2$$

Applications:

- Data fitting
- Signal processing
- Machine learning
- Control systems

The Discrete Least Squares Problem

- Given a set of data points (t_i, y_i) for $i = 1, \ldots, n$, we seek a line

$$p(t) = a_0 + a_1 t$$

  that fits the data.
- In general, the data does not lie exactly on a line $\Rightarrow$ no exact solution.
- We define the residuals:

$$r_i = y_i - p(t_i)$$

  and consider the residual vector $\mathbf{r} = [r_1, \ldots, r_n]^T$.

Least Squares Formulation

- We seek $p(t)$ that minimizes the residual norm $\|\mathbf{r}\|$.
- Different norms lead to different problems:
  - 1-norm: $\|\mathbf{r}\|_1 = \sum |r_i|$
  - ∞-norm: $\|\mathbf{r}\|_\infty = \max |r_i|$
  - **2**-norm (Euclidean): $\|\mathbf{r}\|_2 = \sqrt{\sum r_i^2}$
- The most widely used and best understood is the **least squares** approach:

$$\min_{a_0, a_1} \|\mathbf{r}\|_2^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 t_i)^2$$

Statistical Justification for Least Squares

- If measurement errors in y_i are:
  - Independent,
  - Normally distributed,
  - Mean zero, constant variance σ^2,

  then minimizing $\|r\|_2^2$ gives the **Maximum Likelihood Estimator (MLE)**.
- Justifies use of 2-norm in least squares problems.

Polynomial Approximation

- A straight line: $p(t) = a_0 + a_1 t$ is a degree-1 polynomial.
- Sometimes higher-degree polynomials fit data better.
- General form (degree $< m$):

$$p(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_{m-1} t^{m-1}$$

- Called the **Discrete Least Squares Problem** because data points (t_i, y_i) are finite.

Polynomial Basis and Vector Space

- Set of polynomials of degree $< m$ forms a vector space of dimension m.
- Standard basis: $\phi_1(t) = 1, \phi_2(t) = t, \ldots, \phi_m(t) = t^{m-1}$.
- Any $p(t)$ can be written as:

$$p(t) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_m\phi_m(t)$$

- Alternate bases may improve numerical stability.

Matrix Form of the Problem

- For n data points and m basis functions:

$$Ax \approx b$$

  where:
  - $A \in \mathbb{R}^{n \times m}$, $A_{ij} = \phi_j(t_i)$,
  - $x \in \mathbb{R}^m$: coefficient vector,
  - $b \in \mathbb{R}^n$: vector of y_i values.
- If $n > m$, this is an **overdetermined system**.

Least Squares Formulation

- We seek x minimizing the residual:

$$r = b - Ax$$

- Least squares problem:

$$\min_{x \in \mathbb{R}^m} \|r\|_2^2 = \min_x \|b - Ax\|_2^2$$

- Includes fitting with:
  - Polynomials,
  - Trigonometric functions,
  - Exponentials,
  - Any basis functions $\phi_j(t)$.

What Comes Next

- We will solve the least squares problem using:
  - Normal equations: $A^T Ax = A^T b$
  - Orthogonal transformations: QR decomposition
- These techniques ensure numerical stability and efficient computation.

Rotation in $\mathbb{R}^2$

- A rotation through angle θ is a linear transformation:

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- Acts on any vector $x \in \mathbb{R}^2$ by rotating it counterclockwise.
- This matrix is called a **rotator**.

GIAN GLOBAL INITIATIVE OF ACADEMIC NETWORKS

Properties of Rotators

- $Q^T Q = I$ Q is **orthogonal**.
- $\det(Q) = 1$
- $Q^{-1} = Q^T$: inverse of a rotator is a rotation through $-\theta$.
- Rotators preserve:
  - Vector norms: $\|Qx\| = \|x\|$
  - Angles between vectors

Using Rotators to Create Zeros

- Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, with $x_1 \neq 0$.
- Find rotator Q such that:

$$Q^T x = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

- Choose:

$$\cos\theta = \frac{x_1}{\sqrt{x_1^2 + x_2^2}}, \quad \sin\theta = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}$$

- This ensures $\cos^2\theta + \sin^2\theta = 1$

Geometric Interpretation

- For every $x \in \mathbb{R}^2$, there exists a rotation matrix Q such that:

$$Q^T x = \begin{bmatrix} \|x\| \\ 0 \end{bmatrix}$$

- Interpretation: Rotating vector x onto the x-axis.
- This is a basic step in constructing the QR decomposition.

Application to Matrices

- Let $A \in \mathbb{R}^{2\times 2}$. Then:
$$Q^T A = R$$
where R is upper triangular.
- Generalizes to $A \in \mathbb{R}^{n\times n}$: for such A, there exists an orthogonal matrix Q and an upper triangular matrix R such that:
$$Q^T A = R$$
- This is the foundation of the **QR decomposition**.

QR Decomposition

- For any matrix $A \in \mathbb{R}^{m \times n}$, there exist:
  - $Q \in \mathbb{R}^{m \times m}$ orthogonal
  - $R \in \mathbb{R}^{m \times n}$ upper triangular

  such that:

$$A = QR$$

- $Q^T A = R \Rightarrow$ transformation to upper triangular form
- Fundamental in solving least squares and linear systems

Solving Linear Systems Using QR Decomposition

- Given a QR decomposition: $A = QR$, where
  - Q is orthogonal: $Q^T Q = I$
  - R is upper triangular
- To solve $Ax = b$, rewrite as:

$$QRx = b$$

- Let $y = Rx \Rightarrow Qy = b \Rightarrow y = Q^T b$
- Now solve $Rx = y$ via **back substitution**.

Summary of the QR Method

1. Compute the QR decomposition: $A = QR$
2. Compute $y = Q^T b$
3. Solve $Rx = y$ using back substitution

Advantage

QR decomposition is especially useful when A is full-rank but not square or poorly conditioned for LU.

Two Viewpoints

- Multiply both sides of $Ax = b$ by Q^T:

$$Q^T Ax = Q^T b \quad \Rightarrow \quad Rx = c$$

- Or, write $A = QR$, and:

$$QRx = b \quad \Rightarrow \quad Q(Rx) = b$$

- In both cases:

$$c = Q^T b, \quad Rx = c$$

- **Same method from two equivalent perspectives.**

Example: Solving a System Using QR Decomposition

We want to solve the system:

$$Ax = b, \quad \text{where} \quad A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Step 1: Construct Q and R.

Using vector $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Now

$$\cos\theta = \frac{1}{\sqrt{2}}, \quad \sin\theta = \frac{1}{\sqrt{2}} \Rightarrow Q = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Then:

$$R = Q^T A = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

Step 2: Solve $Q^T b = c$

$$c = Q^T b = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ -\sqrt{2} \end{bmatrix}$$

Step 3: Solve $Rz = c$ by back substitution

$$\sqrt{2} z_1 = \sqrt{2} \Rightarrow z_1 = 1$$
$$\sqrt{2} z_2 = -\sqrt{2} \Rightarrow z_2 = -1$$

Solution: $x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Plane Rotators in $\mathbb{R}^n$ (Givens Rotators)

A **plane rotator** is an $n \times n$ matrix that looks like the identity, except for a 2×2 rotation block in rows and columns i and j.

Definition:

$$Q(i,j,\theta) = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos\theta & \cdots & -\sin\theta & \\ & & \vdots & \ddots & \vdots & \\ & & \sin\theta & \cdots & \cos\theta & \\ & & & & & \ddots \end{bmatrix} \in \mathbb{R}^{n \times n}$$

where the 2×2 rotation is in rows and columns i and j.

Properties:

- Q is orthogonal: $Q^T Q = I$
- $\det(Q) = 1$
- Applying Q or Q^T to a vector alters only the i-th and j-th components

Using a Plane Rotator to Zero an Entry

Given a vector

$$x = \begin{bmatrix} \vdots \\ x_i \\ \vdots \\ x_j \\ \vdots \end{bmatrix} \in \mathbb{R}^n$$

we choose c and s such that:

$$c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad s = \frac{x_j}{\sqrt{x_i^2 + x_j^2}} \Rightarrow Q^T x \text{ has a zero in position } j$$

If $x_i = x_j = 0$, use $c = 1$, $s = 0$.

Only the i-th and j-th entries of x are modified.

Geometric Interpretation of a Plane Rotator

A plane rotator acts on vectors in $\mathbb{R}^n$ by rotating only in the $\{x_i, x_j\}$-plane.

Key Idea: A vector $x \in \mathbb{R}^n$ can be uniquely decomposed as:

$$x = p + p^{\perp}$$

- p lies in the $x_i x_j$-plane
- $p^{\perp}$ is orthogonal to the $x_i x_j$-plane

Action of the Plane Rotator Q:

- Rotates p through angle θ in the $x_i x_j$-plane
- Leaves $p^{\perp}$ unchanged

Visualization

Only components of x along axes i and j are affected. All other components remain unchanged.

Theorem: QR Decomposition Using Rotators

Theorem: Let $A \in \mathbb{R}^{n \times n}$. Then there exists an orthogonal matrix Q and an upper triangular matrix R such that

$$A = QR.$$

Proof Sketch:

- Construct Q as a product of *plane rotators* (Givens rotations).
- Apply rotators $Q_{21}, Q_{31}, \ldots, Q_{n1}$ to zero out entries below a_{11} in column 1.
- These rotators only affect rows below the current pivot and preserve existing zeros above.
- Proceed to column 2:
  - Apply $Q_{32}, Q_{42}, \ldots, Q_{n2}$ to zero entries below a_{22}.
- Repeat this process for columns 3 through $n - 1$.

$$Q = Q_{n,n-1} \cdots Q_{32} Q_{n1} \cdots Q_{21}, \quad R = Q^\top A$$

Since Q is a product of orthogonal matrices, Q itself is orthogonal.
Exercise Cost of QR Decomposition via Rotators **Exercise:** Show that the algorithm sketched in the proof of Theorem takes $\mathcal{O}(n^3)$ flops to transform A to R.

Analysis:

- For each column $k = 1$ to $n-1$, we apply Givens rotations to eliminate entries below the diagonal.
- Number of Givens rotations per column: $n-k$.
- Each Givens rotation affects only two rows $\Rightarrow$ it requires about $2(n-k)$ flops.

Total flop count:

$$\sum_{k=1}^{n-1}(n-k)\cdot 2(n-k) = 2\sum_{k=1}^{n-1}(n-k)^2 = 2\sum_{j=1}^{n-1} j^2 = 2\cdot\frac{(n-1)n(2n-1)}{6}$$

$$\Rightarrow \mathcal{O}(n^3) \text{ flops}$$

Reflectors: Geometric Definition

Goal: Construct a matrix Q that reflects any vector $x \in \mathbb{R}^2$ across a line ℓ through the origin.

Let:

- v be a nonzero vector on the line ℓ
- u be a unit vector orthogonal to ℓ
- Any $x \in \mathbb{R}^2$ can be written as $x = \alpha u + \beta v$

Then:

$$\text{Reflection through } \ell : \quad x \mapsto -\alpha u + \beta v$$

Matrix Formulation:

- Let $P = uu^T$ where u is unit vector ($\|u\|_2 = 1$)
- Define $Q = I - 2P$

$Qx = (I - 2uu^T)x$ is the reflection of x across the hyperplane orthogonal to u

Properties

- Q is symmetric: $Q = Q^T$
- Q is orthogonal: $Q^T Q = I$
- $\det(Q) = -1$

Proposition: Reflector Matrix

Let $u \in \mathbb{R}^n$ be a nonzero vector. Define:

$$Q = I - \frac{2}{\|u\|_2^2} uu^T$$

Then Q is a **reflector** with the following properties:

(a) $Qu = -u$ *(Reflection inverts u)*
(b) $Qv = v$ for all v such that $u^T v = 0$ *(Orthogonal vectors remain unchanged)*

Proof Sketch:

- Normalize u so that $\|u\| = 1 \Rightarrow Q = I - 2uu^T$
- Then:

$$Qu = (I - 2uu^T)u = u - 2u = -u$$

- If $u^T v = 0$:

$$Qv = (I - 2uu^T)v = v - 2u(u^T v) = v$$

Conclusion: Q reflects vectors through the hyperplane orthogonal to u

Theorem : Reflecting x to y

Statement: Let $x, y \in \mathbb{R}^n$ with $x \neq y$ and $\|x\|_2 = \|y\|_2$. Then there exists a unique **reflector** Q such that:

$$Qx = y$$

Theorem : Reflecting x to y

Statement: Let $x, y \in \mathbb{R}^n$ with $x \neq y$ and $\|x\|_2 = \|y\|_2$. Then there exists a unique **reflector** Q such that:

$$Qx = y$$

Construction: Let $u = x - y$, then define:

$$Q = I - \frac{2}{\|u\|^2} uu^T$$

Theorem : Reflecting x to y

Statement: Let $x, y \in \mathbb{R}^n$ with $x \neq y$ and $\|x\|_2 = \|y\|_2$. Then there exists a unique **reflector** Q such that:

$$Qx = y$$

Construction: Let $u = x - y$, then define:

$$Q = I - \frac{2}{\|u\|^2} uu^T$$

Proof Sketch:

- Decompose x as:

$$x = \frac{1}{2}(x + y) + \frac{1}{2}(x - y)$$

- Note: $x - y = u$ and $x + y$ is orthogonal to u since:

$$(x - y)^T (x + y) = \|x\|^2 - \|y\|^2 = 0$$

- Apply Q:

$$Q(x - y) = -u = y - x$$

$$Q(x + y) = x + y \quad \text{(no change)}$$

- Thus:

$$Qx = Q\left(\frac{x+y}{2} + \frac{x-y}{2}\right) = \frac{x+y}{2} - \frac{x-y}{2} = y$$

Reflector Mapping

Statement: Let $x \in \mathbb{R}^n$ be any nonzero vector. Then there exists a **reflector** Q such that:

$$Qx = y = \begin{bmatrix} \pm\|x\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Reflector Mapping

Statement: Let $x \in \mathbb{R}^n$ be any nonzero vector. Then there exists a **reflector** Q such that:

$$Qx = y = \begin{bmatrix} \pm\|x\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Proof:

- Let $y = [-r, 0, \ldots, 0]^T$ with $r = \pm\|x\|_2$.
- Choose the sign of r so that $x \neq y$.
- Then $\|x\|_2 = \|y\|_2$.
- By Theorem 3.2.30, there exists a reflector Q such that $Qx = y$.

Reflector Mapping

Statement: Let $x \in \mathbb{R}^n$ be any nonzero vector. Then there exists a **reflector** Q such that:

$$Qx = y = \begin{bmatrix} \pm\|x\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Proof:

- Let $y = [-r, 0, \ldots, 0]^T$ with $r = \pm\|x\|_2$.
- Choose the sign of r so that $x \neq y$.
- Then $\|x\|_2 = \|y\|_2$.
- By Theorem 3.2.30, there exists a reflector Q such that $Qx = y$.

Note: Any nonzero vector can be reflected to a scalar multiple of the first standard basis vector. This is useful in QR factorization via Householder transformations.

Example: Stable Computation of $\|x\|_2$

Problem: Compute $\|x\|_2$ on a computer that underflows at 10^{-10}. Assume each component of x is smaller than 10^{-10} $\Rightarrow$ all are set to zero.

Example: Stable Computation of $\|x\|_2$

Problem: Compute $\|x\|_2$ on a computer that underflows at 10^{-10}. Assume each component of x is smaller than 10^{-10} $\Rightarrow$ all are set to zero.

Result:

$$\|x\|_2 = 0 \quad \text{(incorrect!)}$$

True norm: nonzero $\rightarrow$ error 5

Example: Stable Computation of $\|x\|_2$

Problem: Compute $\|x\|_2$ on a computer that underflows at 10^{-10}. Assume each component of x is smaller than 10^{-10} $\Rightarrow$ all are set to zero.

Result:

$$\|x\|_2 = 0 \quad \text{(incorrect!)}$$

True norm: nonzero $\rightarrow$ error 5

Solution: Scaling Procedure

- Let $\beta = \max_{1 \le i \le n} |x_i|$
- If $\beta = 0$, then $\|x\|_2 = 0$
- Else, scale: $x = \frac{1}{\beta} z$
- Then compute: $\|z\|_2 = \beta \cdot \|x\|_2$

GIAN
GLOBAL INITIATIVE OF ACADEMIC NETWORKS

Example: Stable Computation of $\|x\|_2$

Problem: Compute $\|x\|_2$ on a computer that underflows at 10^{-10}. Assume each component of x is smaller than 10^{-10} $\Rightarrow$ all are set to zero.

Result:

$$\|x\|_2 = 0 \quad \text{(incorrect!)}$$

True norm: nonzero $\rightarrow$ error 5

Solution: Scaling Procedure

- Let $\beta = \max_{1 \leq i \leq n} |x_i|$
- If $\beta = 0$, then $\|x\|_2 = 0$
- Else, scale: $x = \frac{1}{\beta} z$
- Then compute: $\|z\|_2 = \beta \cdot \|x\|_2$

Why It Works:

- $|x_i| < 1 \rightarrow$ avoids overflow
- Tiny terms may still underflow but can be ignored safely
- Final norm is rescaled $\rightarrow$ correct magnitude

Efficient Application of a Reflector Q

Context: When computing QR decomposition using reflectors, we apply

$$Q = I - \gamma uu^T \quad \text{to a matrix } B \in \mathbb{R}^{n \times m}$$

Efficient Application of a Reflector Q

Context: When computing QR decomposition using reflectors, we apply

$$Q = I - \gamma u u^T \quad \text{to a matrix } B \in \mathbb{R}^{n \times m}$$

Goal: Compute $QB = (I - \gamma u u^T)B = B - \gamma u u^T B$ efficiently without forming Q explicitly.

Efficient Application of a Reflector Q

Context: When computing QR decomposition using reflectors, we apply

$$Q = I - \gamma uu^T \quad \text{to a matrix } B \in \mathbb{R}^{n\times m}$$

Goal: Compute $QB = (I - \gamma uu^T)B = B - \gamma uu^T B$ efficiently without forming Q explicitly.

Efficient Strategy:

- Let $v^T = \gamma u^T \Rightarrow QB = B - uv^T B$
- Compute $v^T B \in \mathbb{R}^{1\times m}$
- Then compute outer product: $u(v^T B) \in \mathbb{R}^{n\times m}$
- Subtract: $QB = B - u(v^T B)$

Exercise: Efficient Computation of $QB = (I - \gamma uu^T)B$

Let $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times m}$. We compare flop counts for different computational strategies.

Exercise: Efficient Computation of $QB = (I - \gamma uu^T)B$

Let $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times m}$. We compare flop counts for different computational strategies.

(a) Compute $(uv^T)B$**:**

- $uv^T \in \mathbb{R}^{n \times n}$ costs n^2 flops
- Multiply with $B \in \mathbb{R}^{n \times m}$ costs $2n^2m$ flops
- **Total:** $\approx 2n^2m$ flops
- **Requires** full $n \times n$ intermediate matrix

Exercise: Efficient Computation of $QB = (I - \gamma uu^T)B$

Let $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times m}$. We compare flop counts for different computational strategies.

(a) Compute $(uv^T)B$**:**

- $uv^T \in \mathbb{R}^{n \times n}$ costs n^2 flops
- Multiply with $B \in \mathbb{R}^{n \times m}$ costs $2n^2m$ flops
- **Total:** $\approx 2n^2m$ flops
- **Requires** full $n \times n$ intermediate matrix

(b) Compute $u(v^TB)$**:**

- $v^TB \in \mathbb{R}^{1 \times m}$ costs nm flops
- $u(v^TB) \in \mathbb{R}^{n \times m}$ costs $2nm$ flops
- **Total:** $nm + 2nm = 3nm$ flops
- **Much cheaper in both time and storage**

(c) Compute $QB = B - \gamma u(v^T B)$:

- From (b), $u(v^T B)$: 3nm flops
- Subtraction: $B - (\cdot)$: nm flops
- **Total:** $3nm + nm = 4nm$ flops

(c) Compute $QB = B - \gamma u(v^T B)$:

- From (b), $u(v^T B)$: 3nm flops
- Subtraction: $B - (\cdot)$: nm flops
- **Total:** $3nm + nm = 4nm$ flops

(d) Compute QB using full matrix $Q \in \mathbb{R}^{n \times n}$:

- Full matrix multiplication: QB costs $2n^2 m$ flops
- **Much more expensive**

Theorem: Uniqueness of the QR Decomposition

Theorem 3.2.46: Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix. Then there exist unique matrices $Q, R \in \mathbb{R}^{n \times n}$ such that:

- Q is orthogonal ($Q^\top Q = I$)
- R is upper triangular with positive diagonal entries
- $A = QR$

Proof Sketch: Existence

- By Theorem 3.2.20, we know that $A = \tilde{Q}\tilde{R}$ for orthogonal $\tilde{Q}$ and upper triangular $\tilde{R}$ (diagonal entries may not be positive).
- Define a diagonal matrix D such that $D_{ii} = \text{sign}(\tilde{R}_{ii})$.
- Then define:

$$Q = \tilde{Q}D, \quad R = D^{-1}\tilde{R}$$

- D is orthogonal (since $D^{-1} = D^{\top} = D$), so Q remains orthogonal and R is upper triangular with positive diagonal entries.

Proof Sketch: Uniqueness

Assume $A = Q_1 R_1 = Q_2 R_2$ where:

- Q_1, Q_2 are orthogonal
- R_1, R_2 are upper triangular with positive diagonal entries

Then:

$$A^\top A = R_1^\top Q_1^\top Q_1 R_1 = R_1^\top R_1$$

$$A^\top A = R_2^\top R_2$$

- $A^\top A$ is symmetric positive definite
- R_1 and R_2 are both Cholesky factors of $A^\top A$
- By uniqueness of the Cholesky decomposition: $R_1 = R_2$
- Then $Q_1 = Q_2$ follows from $Q = AR^{-1}$

QR Decomposition with MATLAB

MATLAB's `qr` function performs QR decomposition:

$$A = QR$$

`Q` is orthogonal, `R` is upper triangular.

Matlab Code n = 7; A = randn(n); [Q, R] = qr(A);
Q'*Q norm(eye(n) - Q'*Q) norm(A - Q*R)

Observations from Output

- **Orthogonality:**

$$Q^\top Q \approx I \Rightarrow \texttt{norm(eye(n) - Q'*Q)} \ll 1$$

- **Accuracy of Factorization:**

$$A \approx QR \Rightarrow \texttt{norm(A - Q*R)} \ll 1$$

- **Diagonal of R:** Entries on the diagonal of R are *not* necessarily positive.
  - MATLAB does not enforce this by default
  - Positive diagonals are needed only for uniqueness

Conclusion

- MATLAB's `qr` is efficient and numerically stable.
- Orthogonality and reconstruction errors are typically small.
- Diagonal signs in R are not fixed by MATLAB.

Explore further: help qr

Numerical Stability of Rotators and Reflectors

Wilkinson's analysis (see Wilkinson [81, pp. 126–162]) shows that:

- Both rotators (Givens rotations) and reflectors (Householder matrices) are numerically stable.
- They are used to construct orthogonal matrices Q for QR decomposition.
- When applied to a matrix A, they produce a small backward error:

$$\widehat{QA} = Q(A + E), \quad \text{with} \quad \frac{\|E\|_2}{\|A\|_2} \ll 1$$

Interpretation: The computed result is the exact result for a slightly perturbed input.

Stability Under Repeated Application

Applying multiple orthogonal transformations:

$$\widehat{Q_2 Q_1 A} = Q_2 Q_1 A + E, \quad \text{where } \frac{\|E\|_2}{\|A\|_2} \ll 1$$

This follows from:

$$\widehat{Q_1 A} = Q_1(A + E_1), \quad \widehat{Q_2(Q_1 A)} = Q_2 Q_1 A + Q_2 E_1 + E_2$$

Then:

$$E = Q_2 E_1 + E_2, \quad \Rightarrow \|E\|_2 \leq \|E_1\|_2 + \|E_2\|_2$$

Conclusion: The backward error remains small after multiple applications.

Note on Stability

- Both Givens and Householder transformations are **normwise backward stable**.
- Errors accumulate slowly: $\|E\|_2/\|A\|_2$ remains small.
- QR decomposition using orthogonal transformations is highly reliable numerically.

Implication: QR-based methods are robust and suitable for solving least squares problems.

Definition of Complex Rotator

Let $z \in \mathbb{C}$, $z \neq 0$. Define

$$U = \frac{1}{|z|} \begin{bmatrix} \bar{z} & -|z| \\ |z| & z \end{bmatrix}$$

- Goal: Show that U is **unitary**
- and $\det(U) = 1$

(a) U is Unitary

We compute:

$$U^* U = \left(\frac{1}{|z|} \begin{bmatrix} \bar{z} & -|z| \\ |z| & z \end{bmatrix} \right)^* \left(\frac{1}{|z|} \begin{bmatrix} \bar{z} & -|z| \\ |z| & z \end{bmatrix} \right)$$

$$= \frac{1}{|z|^2} \begin{bmatrix} z & |z| \\ -|z| & \bar{z} \end{bmatrix} \begin{bmatrix} \bar{z} & -|z| \\ |z| & z \end{bmatrix} = \frac{1}{|z|^2} \begin{bmatrix} |z|^2 + |z|^2 & 0 \\ 0 & |z|^2 + |z|^2 \end{bmatrix} = I$$

Conclusion: U is unitary.

(b) Determinant of U

Use the formula for determinant of a 2×2 matrix:

$$\det(U) = \frac{1}{|z|^2} \det \begin{bmatrix} \bar{z} & -|z| \\ |z| & z \end{bmatrix} = \frac{1}{|z|^2}(\bar{z} \cdot z + |z|^2) = \frac{1}{|z|^2}(|z|^2 + |z|^2) = 1$$

Conclusion: $\det(U) = 1$

Extension to $\mathbb{C}^{n\times n}$

Note: Complex rotators are building blocks for stable algorithms in the complex domain.

(a) Existence of Complex Reflector

Given $x, y \in \mathbb{C}^n$, with $x \neq y$, $\|x\|_2 = \|y\|_2$, and $(x, y) \in \mathbb{R}$.
Claim: There exists a unitary matrix Q of the form:

$$Q = I - \beta u u^*, \quad \beta \in \mathbb{C}, \ u \in \mathbb{C}^n$$

such that $Qx = y$.

Theorem (Complex QR Decomposition)

Let $A \in \mathbb{C}^{n \times n}$ be nonsingular.

Then: There exist unique matrices $Q, R \in \mathbb{C}^{n \times n}$ such that:

- Q is **unitary**: $Q^* Q = I$,
- R is **upper triangular** with **real, positive** entries on the diagonal,
- $A = QR$.

Theorem (Complex QR Decomposition)

Let $A \in \mathbb{C}^{n\times n}$ be nonsingular.

Then: There exist unique matrices $Q, R \in \mathbb{C}^{n\times n}$ such that:

- Q is **unitary**: $Q^*Q = I$,
- R is **upper triangular** with **real, positive** entries on the diagonal,
- $A = QR$.

Uniqueness: If $A = Q_1R_1 = Q_2R_2$ with both R_1, R_2 having real, positive diagonals and Q_1, Q_2 unitary, then:

$$Q_1 = Q_2, \quad R_1 = R_2$$

MATLAB: QR of Complex Matrix

Try the following MATLAB commands:

Code

```
n = 4;
A = randn(n) + 1i * randn(n);  % Complex matrix
[Q, R] = qr(A);                % QR decomposition

Q'                             % Conjugate transpose of Q
Q'*Q                          % Should be the identity
norm(eye(n) - Q'*Q)          % Should be near zero
norm(A - Q*R)                % Should be near zero
```

MATLAB: QR of Complex Matrix

Try the following MATLAB commands:

Code

```
n = 4;
A = randn(n) + 1i * randn(n);  % Complex matrix
[Q, R] = qr(A);                % QR decomposition

Q'                             % Conjugate transpose of Q
Q'*Q                          % Should be the identity
norm(eye(n) - Q'*Q)          % Should be near zero
norm(A - Q*R)                % Should be near zero
```

Observations:

- Q is **unitary**: $Q^*Q \approx I$
- $A \approx QR$: small residual $\|A - QR\|$
- MATLAB handles complex matrices *naturally*

Theorem 3.3.3 (Rectangular QR Decomposition)

Let $A \in \mathbb{R}^{n \times m}$ with $n \geq m$ (i.e., a tall matrix).

Then there exist:

- An **orthogonal** matrix $Q \in \mathbb{R}^{n \times n}$, such that $Q^T Q = I$,
- A matrix $R \in \mathbb{R}^{n \times m}$, of the form:

$$R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \quad \text{where } \hat{R} \in \mathbb{R}^{m \times m} \text{ is upper triangular}$$

such that:

$$A = QR$$

Theorem 3.3.3 (Rectangular QR Decomposition)

Let $A \in \mathbb{R}^{n\times m}$ with $n \geq m$ (i.e., a tall matrix).

Then there exist:

- An **orthogonal** matrix $Q \in \mathbb{R}^{n\times n}$, such that $Q^T Q = I$,
- A matrix $R \in \mathbb{R}^{n\times m}$, of the form:

$$R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \quad \text{where } \hat{R} \in \mathbb{R}^{m\times m} \text{ is upper triangular}$$

such that:

$$A = QR$$

Summary:

- Q: orthogonal basis for $\mathbb{R}^n$
- R: upper-trapezoidal (first m rows upper triangular, rest zero)

Exercise: Flop Count for QR via Reflectors

Goal: Show that the flop count for computing the QR decomposition of an $n \times m$ matrix A using Householder reflectors is approximately:

$$\boxed{\text{Flops} \approx 2nm^2 - \frac{2}{3}m^3}$$

Exercise: Flop Count for QR via Reflectors

Goal: Show that the flop count for computing the QR decomposition of an $n \times m$ matrix A using Householder reflectors is approximately:

$$\boxed{\text{Flops} \approx 2nm^2 - \frac{2}{3}m^3}$$

Sketch of Derivation:

- For each of the m Householder steps:
  - Reflector formation: $\sim 2(n-k+1)$ flops
  - Apply reflector to trailing submatrix of size $(n-k+1) \times (m-k)$
  - Cost per step: $\sim 2(n-k+1)(m-k)$
- Total flops:

$$\sum_{k=1}^{m} 2(n-k+1)(m-k) \approx 2nm^2 - \frac{2}{3}m^3$$

Exercise: Flop Count for QR via Reflectors

Goal: Show that the flop count for computing the QR decomposition of an $n \times m$ matrix A using Householder reflectors is approximately:

$$\boxed{\text{Flops} \approx 2nm^2 - \frac{2}{3}m^3}$$

Sketch of Derivation:

- For each of the m Householder steps:
  - Reflector formation: $\sim 2(n-k+1)$ flops
  - Apply reflector to trailing submatrix of size $(n-k+1) \times (m-k)$
  - Cost per step: $\sim 2(n-k+1)(m-k)$
- Total flops:

$$\sum_{k=1}^{m} 2(n-k+1)(m-k) \approx 2nm^2 - \frac{2}{3}m^3$$

Interpretation:

$$\text{If } n \gg m, \quad \text{then} \quad \boxed{\text{Flops} \approx 2nm^2}$$

This is linear in n, and quadratic in m.

QR Decomposition and Full Rank

Let $A \in \mathbb{R}^{n \times m}$ with $n \geq m$. The QR decomposition:

$$A = QR$$

with $Q \in \mathbb{R}^{n \times n}$ orthogonal and $R \in \mathbb{R}^{n \times m}$ (upper trapezoidal), helps in solving the least squares problem:

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

QR Decomposition and Full Rank

Let $A \in \mathbb{R}^{n \times m}$ with $n \geq m$. The QR decomposition:

$$A = QR$$

with $Q \in \mathbb{R}^{n \times n}$ orthogonal and $R \in \mathbb{R}^{n \times m}$ (upper trapezoidal), helps in solving the least squares problem:

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

Key Insight: Rank and Usefulness

- $\text{rank}(A) = \text{rank}(R)$
- $R = Q^T A \Rightarrow \text{rank}(R) \leq \text{rank}(A)$
- $A = QR \Rightarrow \text{rank}(A) \leq \text{rank}(R)$
- Therefore, $\boxed{\text{rank}(A) = \text{rank}(R)}$

QR Decomposition and Full Rank

Let $A \in \mathbb{R}^{n \times m}$ with $n \geq m$. The QR decomposition:

$$A = QR$$

with $Q \in \mathbb{R}^{n \times n}$ orthogonal and $R \in \mathbb{R}^{n \times m}$ (upper trapezoidal), helps in solving the least squares problem:

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

Key Insight: Rank and Usefulness

- $\text{rank}(A) = \text{rank}(R)$
- $R = Q^T A \Rightarrow \text{rank}(R) \leq \text{rank}(A)$
- $A = QR \Rightarrow \text{rank}(A) \leq \text{rank}(R)$
- Therefore, $\boxed{\text{rank}(A) = \text{rank}(R)}$

Conclusion:

- A has full rank $\Leftrightarrow$ R is nonsingular (i.e., invertible)

Theorem (Least Squares via QR)

Let $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, with $n > m$, and suppose that A has full rank.

Then the least squares problem:

$$\min_{x \in \mathbb{R}^m} \|Ax - b\|_2$$

has a **unique solution**, given as follows:

- Compute the QR decomposition: $A = QR$, where

$$Q \in \mathbb{R}^{n \times n} \text{ (orthogonal)}, \quad R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \quad \hat{R} \in \mathbb{R}^{m \times m} \text{ (upper triangular)}$$

- Let $c = Q^T b$, and define $\hat{c} \in \mathbb{R}^m$ as the first m entries of c
- Solve the system:

$$\hat{R}x = \hat{c}$$

Theorem (Least Squares via QR)

Let $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$, with $n > m$, and suppose that A has full rank.

Then the least squares problem:

$$\min_{x\in\mathbb{R}^m} \|Ax - b\|_2$$

has a **unique solution**, given as follows:

- Compute the QR decomposition: $A = QR$, where

$$Q \in \mathbb{R}^{n\times n} \text{ (orthogonal)}, \quad R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \quad \hat{R} \in \mathbb{R}^{m\times m} \text{ (upper triangular)}$$

- Let $c = Q^T b$, and define $\hat{c} \in \mathbb{R}^m$ as the first m entries of c
- Solve the system:

$$\hat{R}x = \hat{c}$$

Conclusion: The solution to the least squares problem is

$$\boxed{x = \hat{R}^{-1}\hat{c}}$$

MATLAB Example: Least Squares via QR

Given: Overdetermined system $Ax = b$, where $A \in \mathbb{R}^{5\times 3}$

MATLAB Code

```
n = 5; m = 3;
A = randn(n, m);
b = randn(n, 1);

[Q, R_full] = qr(A);
R = R_full(1:m, 1:m);
c = Q' * b;
c_hat = c(1:m);

x = R \ c_hat;
residual_norm = norm(A*x - b)
```

Reference Books

- **Lloyd N. Trefethen and David Bau III**, *Numerical Linear Algebra*, SIAM, 1997.
- **Gene H. Golub and Charles F. Van Loan**, *Matrix Computations*, 4th Edition, Johns Hopkins University Press, 2013.
- **David S. Watkins**, *Fundamentals of Matrix Computations*, 3rd Edition, Wiley, 2010.
- **James W. Demmel**, *Applied Numerical Linear Algebra*, SIAM, 1997.
- **Yousef Saad**, *Numerical Methods for Large Eigenvalue Problems*, SIAM, 2011.
- **Gilbert Strang**, *Linear Algebra and Its Applications*, 4th Edition, Cengage Learning, 2006.
- **Alan J. Laub**, *Matrix Analysis for Scientists and Engineers*, SIAM, 2005.

GLOBAL INITIATIVE OF ACADEMIC NETWORKS

Thank You !