Numerical Methods for 
Computational Science and Engineering

Autumn Semester 2018

Prof. Rima Alaifari, SAM, ETH Zurich

QR decomposition (Find one that is numerically stable?)

Gram-Schmidt revisited:

\[
Q = [q^1 \ldots q^n] \in \mathbb{R}^{m,n} \text{ with orthonormal columns such that:}
q^1 = t_{11}a^1
q^2 = t_{12}a^1 + t_{22}a^2
q^3 = t_{13}a^1 + t_{23}a^2 + t_{33}a^3
\vdots
q^k = t_{1k}a^1 + t_{2k}a^2 + \cdots + t_{nk}a^n.
\]

For a matrix \( A = [a^1, a^2, \ldots, a^n] \in \mathbb{R}^{m,n}:

Step 1:

\[
\begin{bmatrix}
| & | & | & | \\
| a^1 & a^2 & \cdots & a^n | & | t_{11} & 0 & \cdots & 0 |\\
| & 0 & 1 & 0 & \cdots & 0 |\\
| a^2 & a^3 & \cdots & a^n | & | 0 & 1 & & & |\\
| & 0 & 0 & 1 & & |\\
| \vdots & \vdots & \ddots & & | & | \ddots & & & | & | \ddots & & & |\\
| 0 & 0 & \cdots & 1 | & | 1 & t_{12} & 0 & \cdots & 0 |\\
\end{bmatrix}
\]

Step 2:

\[
\begin{bmatrix}
| & | & | & | \\
| a^1 & a^2 & \cdots & a^n | & | t_{11}a^1 & a^2 & \cdots & a^n |\\
| & 0 & 1 & 0 & \cdots & 0 |\\
| a^2 & a^3 & \cdots & a^n | & | 0 & 0 & 1 & & |\\
| & 0 & 0 & 1 & & |\\
| \vdots & \vdots & \ddots & & | & | \ddots & & & | & | \ddots & & & |\\
| 0 & 0 & \cdots & 1 | & | t_{11}a^1 & t_{12}a^1 + t_{22}a^2 & \cdots & a^n |\\
\end{bmatrix}
\]

At each step: multiplication from the right by upper triangular matrix
\[ A = QT \]

\[ R = T^{-1} \] is also upper triangular

\[ \Rightarrow A = QR \]

**QR decomposition via orthogonal transformations**

Consider reflection of vector \( a \) w.r.t. vector \( v \)

Orthogonal transformations on vectors: \( \langle u, v \rangle = \langle Qu, Qv \rangle \)

- preserve their length: \( \|u\| = \|Qv\| \)
- preserve angles between vectors

are only combinations of rotations & reflections

**rotation only**: \( \det Q_{rot} = +1 \)

**reflection only**: \( \det Q_{ref} = -1 \)

Householder reflections

**Idea**: use reflections only

Vector \( a \) reflected at \( v \) gives \( b \)

\[ b = a - 2r = a + 2\text{proj}_v a \]
\[ = a - 2 \frac{v^T a}{\|v\|_2^2} v \]
\[ = a - 2 \frac{vv^T}{\|v\|_2^2} a = (I_m - 2 \frac{vv^T}{v^Tv}) a \]
\[ (v^Ta)v = v(v^Ta) \]

Now we can define the Householder matrix \( H_v \) as
\[ H_v := I_m - \frac{2vv^T}{v^Tv} \]
so that \( H_v a = b \).

Goal: Find vector \( v_1, \ldots, v_n \) such that
\[ H_{v_n} \cdots H_{v_1} A = R = Q^T \]

1st step: \[ H_{v_1} a = \begin{pmatrix} c \\ 0 \\ \vdots \\ 0 \end{pmatrix} = ce^1 \]

unit vector
Find $v^4$ and c s.t. 

$$H_{v^4}a = a - 2 \frac{(v^4)^T a}{\|v^4\|^2} v^4 = c e^4$$

$$\Rightarrow v^4 = (a - c e^4) \frac{\|v^4\|^2}{2 (v^4)^T a}$$

$$\Rightarrow v^4 \text{ must parallel to } a - c e^4$$

Note: Rescaling of $v^4$ doesn't affect the above eqn:

$$c v^4 = (a - c e^4) \frac{\|v^4\|^2}{2 (c v^4)^T a} \checkmark$$

We set $v^4 = a - c e^4$

This implies: 

$$\frac{\|v^4\|^2}{2 (v^4)^T a} = 1$$

Use this to determine $c$:

$$\|a - c e^4\|^2 = \|v^4\|^2 = 2 (v^4)^T a = 2 (a - c e^4)^T a$$

$$\|a\|^2 = 2 a^T (c e^4) + c^2 = 2 \|a\|^2 - 2 a^T (c e^4)$$

$$\Rightarrow c^2 = \|a\|^2$$

$$\Rightarrow c = \pm \|a\|$$

Note: Cancellation can occur in $v^4 = a - c e^4$ + amplification because we divide by $\|v^4\|^2$

Good news: can happen only for one of the possible choices

$$c = \|a\|$$

or

$$c = -\|a\|$$
Choose depending on first entry of $a^i$:

\[
v^i = \begin{cases} 
\frac{1}{2}(a^i - \|a^i\| e^i) & \text{if } (a^i)_1 < 0 \\
\frac{1}{2}(a^i + \|a^i\| e^i) & \text{if } (a^i)_1 > 0
\end{cases}
\]

Choose $v^d = \begin{bmatrix} 0 \\
0 \\
\vdots \\
0 \\
\hat{a}_2^d \\
\end{bmatrix} + c_{\hat{g}} e^d$

with $c_{\hat{g}} = + \|\hat{a}_2^d\|$

How does a general HH step look like?

Consider $j$-th step:

Goal: create $m-j$ zeros in a vector $\tilde{a}^d$

\[
\tilde{a}^d = \begin{bmatrix} 
\tilde{a}_1^d \\
\tilde{a}_2^d \\
\vdots \\
\tilde{a}_m^d \\
\end{bmatrix} \in \mathbb{R}^{m-j+1}
\]

Compute

\[
H_{v^d} \tilde{a}^d = \tilde{a}^d - 2 \frac{(\tilde{a}^d)^T v^d}{\|v^d\|^2} v^d = \tilde{a}^d - v^d
\]

\[
= 1
\]

\[
(\tilde{a}^d)^T v^d = (\tilde{a}_2^d)^T (\tilde{a}_2^d + c_{\hat{g}} \begin{bmatrix} 0 \\
\vdots \\
0 \\
1 \\
\end{bmatrix})
\]

\[
= \|\tilde{a}_2^d\|^2 + (\tilde{a}_2^d)_1 \cdot c_{\hat{g}}
\]

Can we find $v^d$ s.t.:

\[
H_{v^d} \tilde{a}^d = \begin{bmatrix} 
0 \\
0 \\
\vdots \\
0 \\
\hat{a}_2^d \\
\end{bmatrix} \quad m-j \text{ zeros}
\]
\[(v^i)^T v^j = (a_2^i + (c_j^i))^T (a_2^i + (c_j^i)) = \|a_2^i\|^2 + 2a_2^i : c_j^i + c_j^i \|
\]

\[= 2 \left\{ \frac{\|a_2^i\|^2}{2} + (a_2^i : c_j^i) \right\}\]

\[\left(\frac{(a_2^i)^T v^j}{(v^i)^T v^j}\right) = \frac{1}{2}
\]

What happens to previous columns \(r^k\), \(k < j\) as we apply \(H_{v^j}\):

\[H_{v^j} r^k = r^k - 2 \frac{(v^j)^T r^k}{\|v^j\|^2} v^j = r^k - \frac{(v^j)^T r^k}{\|v^j\|^2} v^j = r^k\]

\[\begin{bmatrix} (v^j)^T \gamma^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ (v^j)^T \gamma^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ (r^k)^T \gamma^k \end{bmatrix} = 0\]

\[H_{v^j} a^j = a^j - v^j = \begin{bmatrix} a_2^j \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_2^j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_2^j - c_j^j \end{bmatrix} \in \mathbb{R}^{m-j+1} \]

\[\text{Coincide on indices } j+1 \text{ to } m\]

\[= \begin{bmatrix} r^j \\ 0 \\ \vdots \\ 0 \\ m-j \text{ zeros} \\ 0 \end{bmatrix}\]

Altogether: \(H_{v^1} \ldots H_{v^n} A = R\)

\[Q = H_{v^1}^T \ldots H_{v^n}^T\]

Instead of forming and storing \(Q\): in practice only the vectors \(v^1, \ldots, v^n\) are stored as one
triangular matrix \((v^j : \text{first } j-1 \text{ entries are zero})\) or a "reduced / thin" QR decomposition.

Remark:

For a "tall" matrix \(A \in \mathbb{R}^{m \times n}\) (\(m > n\))

one can either compute the "full" QR decomposition,

\[
A = Q R
\]

or a reduced QR computation:

\[
\tilde{A} = \tilde{Q} \tilde{R}
\]

In Gram-Schmidt: starting from \(a^1, \ldots, a^n\) columns, we compute an orthogonal set of vectors \(\tilde{q}^1, \ldots, \tilde{q}^n\) — reduced QR.

How to obtain a full QR?

- augment \(\tilde{R}\) by zero rows
- find columns \(q^{i+1}, \ldots, q^n\) such that \(\{q^1, \ldots, q^n\}\) ONB for \(\mathbb{R}^m\)
With Householder reflections

either compute full QR decomposition:

\[ Q = H_1^T \cdots H_n^T \]

(recall: \( H_n \) are \( m \times m \) matrices)

or: take "full" identity matrix

\[ I_{m \times n} = \begin{pmatrix} I_m & 0 \\ 0 & 0 \end{pmatrix} \]

and compute reflections successively.

Example: Instead of constructing

\[ H_n^T = I_m - 2 \frac{v_n(v_n)^T}{\|v_n\|^2} \]

we simply compute the reflection for each column

\[ \tilde{e}_i^T = \tilde{e}_i - \begin{pmatrix} (v_n)^T e_i \end{pmatrix} \frac{1}{\|v_n\|^2} \]

Next, apply w.r.t. \( v^{n-1} \) on \( \tilde{q}_n \) ...

This way: thin QR

\[ \tilde{Q} = \begin{pmatrix} \tilde{q}_1 & \cdots & \tilde{q}_n \end{pmatrix} \]

In EIGEN:

\[ Q = qr.householderQ(); \quad \text{apply all HH reflections \( m \) times} \]

\[ Q_{thin} = (qr.householderQ() \ast MatrixXd::Identity(m, n)); \quad \text{apply all HH reflections \( n \) times} \]

In general: HH reflections are directly applied to \( A \) without constructing \( Q \).
Complexity of H^H for m > n?

1 reflection applied to 1 vector: \( \Theta(m) \)

1 reflection applied to \( A \): \( \Theta(mn) \)

\( n \) reflections applied to \( A \): \( \Theta(mn^2) \)

Then \( \| Ax - b \|_2 = \| Q(Rx - Q^Hb) \|_2 = \| Rx - Q^Hb \|_2 \)

\( Q \) preserves length

Thus solving \( Ax = b \) in least-squares sense

\( \iff \) to solving \( Rx = \tilde{b} \) in least-sq sense

Solving least-squares problem with

QR decomposition

Suppose we have computed decomposition \( A = QR \).

\( (\text{recall } Q^HQ = I) \)
The zero rows can never be fulfilled:

\[
x = R_0^r \begin{bmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_m \end{bmatrix}
\]

\[R_0 = (R)_{1:n,1:n}\]

We can also compute the residual:

\[
r = Q \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tilde{b}_{m+1} \\ \vdots \\ \tilde{b}_m \end{bmatrix}
\]

\[
= \|r\|_2 = \left( \sum_{i=n+1}^m \tilde{b}_i^2 \right)^{1/2}
\]

Alternatively to HT reflections: Given rotations are based on rotations only.

---

### 3.4. The Singular Value Decomposition

A different orthogonal decomposition

\[A = U \Sigma V^T\]

**Example:**
- How to compress an image?
- How to extract common features from a data set?
- How to solve \(Ax = b\) approximately?

1. Construction of Moore-Penrose inverse \(A^+\)
2. What can be done if \(\text{cond}(A)\) is large?
Theorem 3.4.1 (Singular value decomposition)
For any $A \in K^{m \times n}$, there are unitary matrices $U \in K^{m \times m}$, $V \in K^{n \times n}$ and a (generalized) diagonal (*) matrix $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p) \in K^{m \times n}$, $p := \min\{m, n\}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, such that

$$A = U \Sigma V^H.$$
Reduced SVD for $m > n$:

$$A = U \Sigma V^H$$

Case $m < n$: reduced SVD: $\Sigma \in \mathbb{K}^{m,m}$

$$V \in \mathbb{K}^{n,m}$$

Facts from linear algebra:

Lemma 3.4.1. The squares $\sigma_i^2$ of the non-zero singular values of $A$ are the non-zero eigenvalues of $A^HA$ and $AA^H$ with associated eigenvectors $(V)_i, 1, \ldots, (V)_p$ and $(U)_i, 1, \ldots, (U)_p$, respectively.

Lemma 3.4.2 (SVD and rank of a matrix). If, for some $1 \leq r \leq p := \min\{m, n\}$, the singular values of $A \in \mathbb{K}^{m,n}$ satisfy $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0$, then

- $\text{rank}(A) = r$ (no. of non-zero singular values),
- $\mathcal{N}(A) = \text{Span}\{(V)_r+1, \ldots, (V)_n\}$,
- $\mathcal{R}(A) = \text{Span}\{(U)_1, \ldots, (U)_r\}$.

Implication: If SVD determines $\text{rank}(A)$

- with $U$ and $V$ we obtain ONB for $\mathcal{R}(A)$ & $\mathcal{N}(A)$!

$\Rightarrow$ useful for construction of Moore–Penrose inverse.
In EIGEN: jacobi SVD is numerically stable

least squares: \( \min \| A x - b \|_2 \)

either compute \( \Sigma \), values only or

add flags: ComputeThinU, ComputeThinV

ComputeFullU, ComputeFullV

cost of thin SVD: \( O(mn^2) \) \( m > n \).

Generalized solutions by SVD

Assume \( A \in \mathbb{R}^{m \times n} \), \( m \geq n \), \( \text{rank}(A) = r \leq n \)

\[ A = [U_1 \ U_2] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix} \]

\[ \Sigma_r = \text{diag}(\sigma_1, \ldots, \sigma_r) \]

\[ \| A x - b \|_2^2 = \| \Sigma_r V_1^H x - b \|_2^2 = \| U_r^H (U \Sigma_r V_1^H x - b) \|_2^2 \]

\[ = \| \Sigma_r V_1^H x - U_r^H b \|_2^2 \]

\[ = \| \begin{bmatrix} \Sigma_r V_1^H x \\ 0 \end{bmatrix} - \begin{bmatrix} U_1^H b \\ U_2^H b \end{bmatrix} \|_2^2 \]

\[ = \| \begin{bmatrix} \Sigma_r V_1^H x - U_1^H b \\ -U_2^H b \end{bmatrix} \|_2^2 \]

\[ = \| \Sigma_r V_1^H x - U_1^H b \|_2^2 + \| U_2^H b \|_2^2 \]

\[ \frac{\| \Sigma_r V_1^H x - U_1^H b \|_2^2}{\| U_2^H b \|_2^2} \]

\[ \text{fixed} \]

\[ \text{equivalently: } \min_x \| \Sigma_r V_1^H x - U_1^H b \|_2^2 \]
\[ \Sigma_r V_1^H x = U_1^H b \]

LSE \quad r \times n

Theorem (Pseudo-inverse and SVD):

If \( A \in \mathbb{K}^{m \times n} \) has the SVD \( A = U \Sigma V^H \), then its Moore-Penrose Pseudo-inverse is given by

\[ A^+ = V_1 \Sigma_r^{-1} U_1^H \]

If \( r < n \): non-uniqueness from \( \mathcal{N}(V_1^H) \)

\[ \mathcal{N}(V_1^H) = \mathbb{R}(V_1)^\perp \]

Choose \( x \in \mathbb{R}(V_1) \)

\[ x = V_1 z \quad \text{for some} \quad z \in \mathbb{K}^r \]

\[ \Sigma_r V_1^H V_1 z = U_1^H b \]

\[ = I \]

\[ \Sigma_r z = U_1^H b \]

\[ \Rightarrow \quad z = \Sigma_r^{-1} U_1^H b \]

Generalized \( x \): \( V_1 z = V_1 \Sigma_r^{-1} U_1^H b \)

SVD-based optimization & approximation

Norm-based Extrema

Consider the following minimization problem:

Given \( A \in \mathbb{K}^{m \times n} \), \( m \geq n \)

Find \( x \in \mathbb{K}^n \) s.t.

\[ \| Ax \|_2 \rightarrow \text{min}, \quad 1 \| x \|_2 = 1 \]
The SVD: \( A = U \Sigma V^H \)

\[
\min_{k \leq n, r \leq m} \| A_{k \times r} \|_F = \min \| U \Sigma_k V^H \|_F = \min_k \| U \Sigma_k V^H \|_F
\]

where

\[
\Sigma_k = \begin{bmatrix}
\sigma_1 & \cdots & \sigma_k \\
0 & \cdots & 0
\end{bmatrix}
\]

is the best low-rank approximation of \( A \).

For a given matrix \( A \), find a low-rank matrix \( \tilde{A} \) that is the closest to \( A \) and has a desired rank \( k \).

Find a matrix \( \tilde{A} \in \mathbb{R}^{m \times n} \) with \( \text{rank}(\tilde{A}) \leq k \) such that

\[
\| A - \tilde{A} \|_F \rightarrow \text{min} \quad \text{over all rank-} k \text{ matrices}
\]

\( \| \cdot \|_F \): Frobenius norm for \( A \in \mathbb{R}^{m \times n} \)

\[
\| A \|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}
\]

\( \| A \|_F \) can be computed from the singular values

\[
\| A \|_F^2 = \sum_{i=1}^n \sigma_i^2 \\
\quad (\text{rank}(A) = r)
\]

Minimal value is attained by \( \gamma = e_n \)

\[
\Rightarrow V^H x = y = e_n
\]

\[
\Rightarrow x = V e_n = (V)_{1:n}
\]
Desirable: \( k < \min \{m,n\} \)

Applications: • data compression
  • modelling
  • PCA

Use SVD: \( A = U \Sigma V^H \)

\[
A = \sum_{i=1}^{k} \sigma_i (U)_i \cdot (V)_i^H
\]

Take \( \tilde{A} = \sum_{i=1}^{k} \sigma_i (U)_i \cdot (V)_i^H \)

\( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \)

Turns out: This is the best rank-\( k \) approximation to \( A \).

\[
\mathcal{R}_k(m,n) = \{ F \in \mathbb{C}^{m \times n} : \text{rank}(F) \leq k \}
\]

\( \tilde{A} \) is the closest element in \( \mathcal{R}_k(m,n) \) to \( A \).

in 2-norm & Frobenius norm

Theorem:

Let \( A = U \Sigma V^H \) be the SVD of \( A \in \mathbb{C}^{m \times n} \).

For \( k \in \{1, \ldots, \text{rank}(A)\} \), set

\[
U_k = [ (U)_1, \ldots, (U)_{i_k} ] \in \mathbb{C}^{m \times k}
\]

\[
V_k = [ (V)_{i_k}, \ldots, (V)_{n_k} ] \in \mathbb{C}^{n \times k}
\]

\[
\Sigma_k = \text{diag} (\sigma_1, \ldots, \sigma_k)
\]
Then:

$$\| A - U_k \Sigma_k (V_k)^H \| \leq \| A - F \| \quad \forall \ F \in \mathbb{R}^k(m \times n)$$

This holds for

$$\| \cdot \|_F = \| \cdot \|_F$$

$\| \cdot \|_2 = \| \cdot \|_2$

Best low-rank approximation is useful e.g. for compression

For best $k$-rank approximation: reduce required memory from $m \times n$ to $k(m+n-1)$.

How good is the approximation:

$$A - \tilde{A} = \sum_{i=k+1}^{r} \sigma_i(U_i V_i^T)$$

Since values of $A - \tilde{A}$: $\sigma_{k+1} \geq \ldots \geq \sigma_r$

$$\| A - \tilde{A} \|_F = \sigma_{k+1}$$

$$\| A - \tilde{A} \|_2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Example: Image compression

Original image

Compressed image (50 singular values)

Dim: $530 \times 600$
Original image: $530 \cdot 600 \approx 3 \cdot 10^5$

Compressed image: $50 \cdot \frac{1129}{m \mu m^{-1}} \approx 5 \cdot 10^4$

Note: In practice, other methods are used
(e.g. ridgelets / curvelets)

Principle Component Analysis (PCA)