Utrecht, 3 oktober 2017

Matrix factorizations

Gerard Sleijpen

Universiteit Utrecht

Department of Mathematics

http://www.staff.science.uu.nl/∼sleij101/

Program Lecture 3

- Factorizations
- Factorizations for linear problems

  LU-decomposition
- Intermezzo: orthonormal matrices
- Factorizations for linear problems

  QR-decomposition

Program Lecture 3

- Factorizations
- Factorizations for linear problems

  LU-decomposition
- Intermezzo: orthonormal matrices
- Factorizations for linear problems

  QR-decomposition

Factorizations

$$\mathbf{A} = \mathbf{PQR},$$

where

— $\mathbf{A}$ is a given matrix

— $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ are to be constructed and have attractive properties

NLA = Factorisations

Factorizations

$$\mathbf{A} = \mathbf{PQR},$$

where

— **A** is a given matrix

— **P**, **Q** and **R** are to be constructed and
have attractive properties

'one-sided' factorisations as $\mathbf{A} = \mathbf{LU}$ for linear systems

similarity transforms as $\mathbf{A} = \mathbf{VDV}^{-1}$
for eigenvalue problems

congruency transforms as $\mathbf{A} = \mathbf{VRV}^{*}$
for linear systems and eigenvalue problems

Why not one-sided for eigenvalue problems?

Factorizations

$$\mathbf{A} = \mathbf{PQR},$$

where

— $\mathbf{A}$ is a given matrix

— $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ are to be constructed and have attractive properties

- **LU-decomposition**: $\mathbf{A} = \mathbf{LU}$, $\quad \mathbf{PA} = \mathbf{LU}$,

  Cholesky decomposition: (if $\mathbf{A}$ is PD) $\mathbf{A} = \mathbf{CC}^*$
- **QR-factorization**: $\mathbf{A} = \mathbf{QR}$
- **Eigenvalue decomposition**: $\mathbf{A} = \mathbf{VDV}^{-1}$
- **Schur decomposition**: $\mathbf{A} = \mathbf{QSQ}^*$
- **Singular value decomposition**: $\mathbf{A} = \mathbf{VDQ}^*$

Program Lecture 3

- Factorizations
- Factorizations for linear problems

  LU-decomposition
- Intermezzo: orthonormal matrices
- Factorizations for linear problems

  QR-decomposition

LU-decomposition

$\mathbf{A}$ is a non-singular $n \times n$ matrix.

Assigment. Solve $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$.

Strategy.

- Use Gaussian elimination to obtain

$$\mathbf{A} = \mathbf{LU}$$

with $\mathbf{L}$ lower-Δ with $\text{diag}(\mathbf{L}) = \mathbf{I}$, and
$\mathbf{U}$ upper-Δ.

- Solve $\mathbf{Ly} = \mathbf{b}$ for $\mathbf{y}$,
- Solve $\mathbf{Ux} = \mathbf{y}$ for $\mathbf{x}$.

LU-decomposition, costs

A is $n \times n$. Solve $\mathbf{Ax} = \mathbf{b}$: $\mathbf{A} = \mathbf{LU}$, $\mathbf{Ly} = \mathbf{b}$, $\mathbf{Ux} = \mathbf{y}$

Costs (i.e., # flops) depend on the sparsity structure.

If **A** is full: $\frac{2}{3}n^3$ flop
If **A** has bandwidth p (i.e., $a_{ij} = 0$ if $|i - j| > p$): $2p^2\,n$ flop.

Costs may be much less if **A** has an 'arrowhead' structure.

Use a **pivoting** strategy to improve **A**'s structure, i.e., find a row permutation $\mathbf{P}_r$ and a column permutation $\mathbf{P}_c$ such that $\mathbf{P}_r\mathbf{A}\mathbf{P}_c$ has a more favourable structure (smaller bandwidth, longer 'arrows', ...).

Solve $\mathbf{Ax} = \mathbf{b}$:

$$\mathbf{P}_r\mathbf{A}\mathbf{P}_c = \mathbf{LU}, \quad \mathbf{Uy} = \mathbf{P}_r\mathbf{b}, \quad \mathbf{Uz} = \mathbf{y}, \quad \mathbf{x} = \mathbf{P}_c^\top\mathbf{z}.$$

LU-decomposition, costs

A is $n \times n$. Solve $\mathbf{Ax} = \mathbf{b}$: $\mathbf{A} = \mathbf{LU}$, $\mathbf{Ly} = \mathbf{b}$, $\mathbf{Ux} = \mathbf{y}$

Costs (i.e., # flops) depend on the sparsity structure.

If **A** is full: $\frac{2}{3}n^3$ flop
If **A** has bandwidth p (i.e., $a_{ij} = 0$ if $|i-j| > p$): $2p^2\,n$ flop.

Costs may be much less if **A** has an 'arrowhead' structure.

Rule of the thumb. Gaussion elimination may not be feasibe (i.e., costs are too high)
if $n > 30\,000$ for full matrices
if $n > 300\,000$ for sparse matrices (even with pivoting)

If Gaussian elimination is unfeasable, use some iterative method: see the following lectures

LU-decomposition, stability

$\mathbf{A}$ is $n \times n$. Solve $\mathbf{A}\mathbf{x} = \mathbf{b}$: $\mathbf{A} = \mathbf{L}\mathbf{U}$, $\mathbf{L}\mathbf{y} = \mathbf{b}$, $\mathbf{U}\mathbf{x} = \mathbf{y}$

With $\widehat{\mathbf{L}}$,$\widehat{\mathbf{U}}$, $\widehat{\mathbf{y}}$, and $\widehat{\mathbf{x}}$ the computed quantities:

Theorem. $\quad (\mathbf{A} + \Delta_A)\widehat{\mathbf{x}} = \mathbf{b}$ with

$$|\Delta_A| \leq 3\, p\, \mathbf{u}\, |\widehat{\mathbf{L}}|\, |\widehat{\mathbf{U}}| \approx 3\, p\, \mathbf{u}\, |\mathbf{L}|\, |\mathbf{U}|.$$

Here $|\cdot|$ and $\leq$ matrix-entry-wise, p bandwidth of $\mathbf{A}$.

Δ_A is the **backward error** of Gaussian elimination.
This leads to following bound on the **forward error**:

$$\frac{\|\widehat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \leq \mu \equiv p\, \mathbf{u}\, \mathcal{C}(\mathbf{A})\, 3\, \rho, \quad \text{where} \quad \rho \equiv \frac{\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|}{\|\mathbf{A}\|}.$$

Here $\|\cdot\|$ is a vector norm.

Recall: $\mathrm{fl}(\mathbf{A}\mathbf{x}) = (\mathbf{A} + \Delta_A)\mathbf{x}$ with $|\Delta_A| \leq p\, \mathbf{u}\, |\mathbf{A}|$.

LU-decomposition, stability

$\mathbf{A}$ is $n \times n$. Solve $\mathbf{Ax} = \mathbf{b}$: $\mathbf{A} = \mathbf{LU}$, $\mathbf{Ly} = \mathbf{b}$, $\mathbf{Ux} = \mathbf{y}$

Stability Gaussian elimination involves an "extra" factor

$$3\rho \equiv 3(\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|)/\|\mathbf{A}\|$$

- Note that

$$\rho_\infty \equiv \frac{\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|_\infty}{\|\mathbf{A}\|_\infty} = \frac{\|\,|\mathbf{L}|\,(|\mathbf{U}|\mathbf{1})\,\|_\infty}{\|\mathbf{A1}\|_\infty} \leq p^2\,\|\mathbf{L}\|_{\max}\,\frac{\|\mathbf{U}\|_{\max}}{\|\mathbf{A}\|_{\max}}.$$

Here, $\|\mathbf{A}\|_{\max} = \|(a_{i,j})\|_{\max} \equiv \max_{i,j}|a_{i,j}|$.

- Extra factor ρ_∞ can be large (2^{n-1}) even if $\|\mathbf{L}\|_{\max} = 1$.

LU-decomposition, stability

$\mathbf{A}$ is $n \times n$. Solve $\mathbf{Ax} = \mathbf{b}$: $\mathbf{A} = \mathbf{LU}$, $\mathbf{Ly} = \mathbf{b}$, $\mathbf{Ux} = \mathbf{y}$

Stability Gaussian elimination involves an "extra" factor

$$3\rho \equiv 3(\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|)/\|\mathbf{A}\|$$

- Note that

$$\rho_\infty \equiv \frac{\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|_\infty}{\|\mathbf{A}\|_\infty} = \frac{\|\,|\mathbf{L}|\,(|\mathbf{U}|\mathbf{1})\,\|_\infty}{\|\mathbf{A1}\|_\infty} \leq p^2\,\|\mathbf{L}\|_{\max}\,\frac{\|\mathbf{U}\|_{\max}}{\|\mathbf{A}\|_{\max}}.$$

- Extra factor ρ_∞ can be large (2^{n-1}) even if $\|\mathbf{L}\|_{\max} = 1$.

Wilkinson's Miracle [±1960]. In practice, almost always,

$$\|\mathbf{L}\|_{\max} = 1 \quad \Rightarrow \quad \frac{\|\mathbf{U}\|_{\max}}{\|\mathbf{A}\|_{\max}} \leq 16.$$

LU-decomposition, stability

$\mathbf{A}$ is $n \times n$. Solve $\mathbf{A}\mathbf{x} = \mathbf{b}$: $\mathbf{A} = \mathbf{L}\mathbf{U}$, $\mathbf{L}\mathbf{y} = \mathbf{b}$, $\mathbf{U}\mathbf{x} = \mathbf{y}$

Stability Gaussian elimination involves an "extra" factor

$$3\rho \equiv 3(\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|)/\|\mathbf{A}\|$$

- Note that

$$\rho_\infty \equiv \frac{\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|_\infty}{\|\mathbf{A}\|_\infty} = \frac{\|\,|\mathbf{L}|\,(|\mathbf{U}|\mathbf{1})\,\|_\infty}{\|\mathbf{A}\mathbf{1}\|_\infty} \leq p^2\,\|\mathbf{L}\|_{\max}\,\frac{\|\mathbf{U}\|_{\max}}{\|\mathbf{A}\|_{\max}}.$$

With so-called **Partial Pivoting**, we find a row permutation $\mathbf{P}$ such that $\|\mathbf{L}\|_{\max} = 1$ (i.e., $\mathbf{P}_r = \mathbf{P}$ and $\mathbf{P}_c = \mathbf{I}$).

LU-decomposition, stability

$\mathbf{A}$ is $n \times n$. Solve $\mathbf{Ax} = \mathbf{b}$: $\mathbf{A} = \mathbf{LU}$, $\mathbf{Ly} = \mathbf{b}$, $\mathbf{Ux} = \mathbf{y}$

Stability Gaussian elimination involves an "extra" factor

$$3\rho \equiv 3(\| \, |\mathbf{L}| \, |\mathbf{U}| \, \|)/\|\mathbf{A}\|$$

- Note that

$$\rho_\infty \equiv \frac{\| \, |\mathbf{L}| \, |\mathbf{U}| \, \|_\infty}{\|\mathbf{A}\|_\infty} = \frac{\| \, |\mathbf{L}| \, (|\mathbf{U}|\mathbf{1}) \, \|_\infty}{\|\mathbf{A1}\|_\infty} \leq p^2 \, \|\mathbf{L}\|_{\mathsf{max}} \, \frac{\|\mathbf{U}\|_{\mathsf{max}}}{\|\mathbf{A}\|_{\mathsf{max}}}.$$

Note. In practice, partial pivoting may spoil sparsity:

balans efficiency and stability.

For large n and sparse $\mathbf{A}$, partial pivotting may even be unfeasible and Gaussian elimination may not be sufficiently stable.

Strategy for solving $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$ $(*)$.

1) Apply **row scaling** to $(*)$ (to reduce $\mathcal{C}(\mathbf{A})$, that is, try to reduce the forward error of $(*)$), i.e., solve

$$(\mathbf{D}^{-1}\mathbf{A})\mathbf{x} = \mathbf{D}^{-1}\mathbf{b} \quad \text{for} \quad \mathbf{x} \qquad (**)$$

Here $\mathbf{D} = \mathbf{D}_r = (d_{ij})$ is a diagonal matrix with $d_{ii} = \|\mathbf{A}^*\mathbf{e}_i\|$, the norm of the ith row of $\mathbf{A}$.

Notes. • Is cheap, preserves sparsity, destroys symmetry.

• Column scaling reduces the error on $\mathbf{D}_c\mathbf{x}$ (rather than on $\mathbf{x}$).

• Row scaling changes may lead to larger errors on $\mathbf{b}$.

• $(**)$ is an instance of a more general strategy to improve the conditioning, called **preconditioning**: $\mathbf{M}^{-1}\mathbf{Ax} = \widetilde{\mathbf{b}} \equiv \mathbf{M}^{-1}\mathbf{b}$ where systems as $\mathbf{M}\widetilde{\mathbf{b}} = \mathbf{b}$ are easy to solve and $\mathcal{C}(\mathbf{M}^{-1}\mathbf{A})$ is smaller than $\mathcal{C}(\mathbf{A})$.

For ease of notation,
we assume $\mathbf{A}$ to be replaced by $\mathbf{D}^{-1}\mathbf{A}$ and $\mathbf{b}$ by $\mathbf{D}^{-1}\mathbf{b}$.

Strategy for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ $(*)$.

1) Apply **row scaling** to $(*)$.

2) **If feasible** find appropriate permutations $\mathbf{P}_r$ and $\mathbf{P}_c$ and LU-factors $\mathbf{L}$ and $\mathbf{U}$: $\mathbf{P}_r \mathbf{A} \mathbf{P}_c = \mathbf{L}\mathbf{U}$.

'**Feasible**', that is, if costs permit.

Notes. • For optimal stability, use partial pivoting. This, however, may destroy a favourable structure that $\mathbf{A}$ may have (sparsity or symmetry or ...).

• Feasibility may require another pivoting strategy.

• Computation of $\mathbf{L}$ and $\mathbf{U}$ may be unfeasible for any pivoting strategy (if $\mathbf{A}$ is dense, n is huge).

For ease of notation,
we assume $\mathbf{A}$ to be replaced by $\mathbf{P}_r \mathbf{A} \mathbf{P}_c$ and $\mathbf{b}$ by $\mathbf{P}_r\mathbf{b}$,
we denote the computed L and U factors by $\mathbf{L}$ and $\mathbf{U}$.

Strategy for solving $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$ $(*)$.

1) Apply **row scaling** to $(*)$.

2) **If feasible** find appropriate permutations $\mathbf{P}_r$ and $\mathbf{P}_c$ and LU-factors $\mathbf{L}$ and $\mathbf{U}$: $\mathbf{P}_r\,\mathbf{A}\,\mathbf{P}_c = \mathbf{L}\mathbf{U}$.

3) Estimate $\mu \equiv 3p\,\mathbf{u}\,\mathcal{C}(\mathbf{A})\rho$ by, say, $\widehat{\mu}$.
Recall that $\|\widehat{\mathbf{x}} - \mathbf{x}\| \leq \mu\,\|\mathbf{x}\|$ and $\rho \equiv (\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|)/\|\,|\mathbf{A}|\,\|$.
If $\widehat{\mu}$ is sufficient small, do 4) else do 5).

4) Solve $\mathbf{Ly} = \mathbf{b}$, $\mathbf{Ux} = \mathbf{y}$ and undo the row permutation on $\mathbf{x}$.

5) If $\widehat{\mu} \ll 1$

a) apply a few steps of iterative refinement

else

b) consider using a QR-decomposition to solve $(*)$.

Details on 5.a) and 5.b) on the next transparancies.

5.a) If $\mu \ll 1$ (e.g., $\mu \approx 10^{-2}$) apply a few steps of

iterative refinement

(on the row-scaled, permuted, system)

```
x_0 = 0
for  j = 0, 1, ...  do
    break if  x_j is sufficiently accurate
    compute the residual  r_j ≡ b − A x_j ,
    solve  A u_j = r_j  for  u_j
            using the L and U factors of A
    update x:   x_{j+1} = x_j + û_j
```

$$\mathbf{x}_0 = \mathbf{0}$$

for $j = 0, 1, \ldots$ do

break if $\mathbf{x}_j$ is sufficiently accurate

compute the residual $\mathbf{r}_j \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_j$,

solve $\mathbf{A}\,\mathbf{u}_j = \mathbf{r}_j$ for $\mathbf{u}_j$ using the $\mathbf{L}$ and $\mathbf{U}$ factors of $\mathbf{A}$

update $\mathbf{x}$: $\mathbf{x}_{j+1} = \mathbf{x}_j + \widehat{\mathbf{u}}_j$

Theorem. $\|\mathbf{x}_j - \mathbf{x}\| \lesssim \mu^j \|\mathbf{x}\|$:

the forward error is reduced by a factor μ per step.

Note that the expensive part, row-scaling, pivoting, computing $\mathbf{L}$ and $\mathbf{U}$ has to be done only once.

Iterative refinement is an instance of the **basic iterative scheme**

```
Select x_0
x = x_0,  r = b - Ax
for j = 1 : j_max
    break if ||r|| <= tol
    Compute an approximate solution û of
        Au = r
    x <- x + û
    r <- r - Aû
```

If $\mathbf{x}_j$ is some approximate solution of $\mathbf{Ax} = \mathbf{b}$ with error $\mathbf{u}_j$, i.e., $\mathbf{x} = \mathbf{x}_j + \mathbf{u}_j$, then $\mathbf{u}_j$ satisfies

$$\mathbf{Au}_j = \mathbf{r}_j \equiv \mathbf{b} - \mathbf{Ax}_j$$

If $\mathbf{x}_{j+1} = \mathbf{x}_j + \widehat{\mathbf{u}}_j$ then $\mathbf{r}_{j+1} = \mathbf{b} - \mathbf{Ax}_{j+1} = \mathbf{r}_j - \mathbf{A}\widehat{\mathbf{u}}_j$.

5.b) If $\mu \equiv 3\,p\,\mathbf{u}\,\mathcal{C}(\mathbf{A})\,\frac{\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|}{\|\mathbf{A}\|} \not\ll 1$ use a

QR-decomposition

Before discussion QR-decomposition some comments.

μ can be non-small if the Gaussian elimination process is unstable (which is unlikely when combined with partial pivoting).

QR-decompositions spoil sparsity and are not feasible for square systems of high dimension as $n \geq 20\,000$.

QR-decompositions are useful for “solving” (in some adjusted way) ill-conditioned systems of modest dimension as $n \leq 20\,000$

5.b) If $\mu \equiv 3\,p\,\mathbf{u}\,\mathcal{C}(\mathbf{A})\,\frac{\|\,|\mathbf{L}|\,|\mathbf{U}|\,\|}{\|\mathbf{A}\|} \not\ll 1$ use a

QR-decomposition

Before discussion QR-decomposition some comments.

QR-decompositions of $n \times k$ systems (i.e., **A** is an $n \times k$ matrix), with k modest, as $k < 200$, allow stable computations and form the backbone of many methods for high dimensional problems with $n > 10^7$.

Program Lecture 3

- Factorizations
- Factorizations for linear problems

  LU-decomposition
- Intermezzo: orthonormal matrices
- Factorizations for linear problems

  QR-decomposition

Intermezzo: orthonormal matrices

Suppose $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_q]$ is orthonormal.
The column vector $\mathbf{v}_i$ form an orthonormal basis of

$$\mathcal{V} \equiv \text{span}(\mathbf{V}) = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}.$$

$\mathbf{P} \equiv \mathbf{V}\mathbf{V}^*$ is an **orthogonal projection** onto $\mathcal{V}$: [Ex.3.7]
$\mathbf{P}\mathbf{x} \in \mathcal{V}$ ($\mathbf{x} \in \mathbb{C}^n$), $\mathbf{P}\mathbf{x} = \mathbf{x}$ ($\mathbf{x} \in \mathcal{V}$), $\mathbf{x} - \mathbf{P}\mathbf{x} \perp \mathbf{P}\mathbf{x}$ ($\mathbf{x} \in \mathbb{C}^n$)

$\mathbf{I} - \mathbf{V}\mathbf{V}^*$ is an orthogonal projection onto $\mathcal{V}^\perp$.

Householder reflections. $\mathbf{H} \equiv \mathbf{I} - 2\mathbf{V}\mathbf{V}^*$ is unitary, [ex.3.8]
a **reflection** wrt the 'mirror space' $\mathcal{V}^\perp$:
if $\mathbf{x} = \mathbf{x}_\mathcal{V} + \mathbf{x}_{\mathcal{V}^\perp}$ then $\mathbf{H}\mathbf{x} = -\mathbf{x}_\mathcal{V} + \mathbf{x}_{\mathcal{V}^\perp}$. ($\mathbf{x}_\mathcal{V} \in \mathcal{V}$, $\mathbf{x}_{\mathcal{V}^\perp} \in \mathcal{V}^\perp$).

Exercise. Determine # flop to compute $\mathbf{x}_\mathcal{V}$, $\mathbf{x}_{\mathcal{V}^\perp}$, $\mathbf{H}\mathbf{x}$

Program Lecture 3

- Factorizations
- Factorizations for linear problems

  LU-decomposition
- Intermezzo: orthonormal matrices
- Factorizations for linear problems

  QR-decomposition

QR-factorization

Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ be an $n \times k$ matrix.

$$\mathbf{A} = \mathbf{QR}$$

with $\mathbf{Q}$ orthonormal, $\mathbf{R}$ upper-Δ, matching dimensions:
— $\mathbf{Q} \equiv \mathbf{Q}_1$ $n \times n$ (Unitary) & $\mathbf{R} \equiv \mathbf{R}_1$ $n \times k$
— $\mathbf{Q} \equiv \mathbf{Q}_0$ $n \times k$ & $\mathbf{R} \equiv R_0$ $k \times k$ (economical form).

Matlab: `[Q1,R1]=qr(A);` `[Q0,R0]=qr(A,'0');`

QR-factorization

Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ be an $n \times k$ matrix.

$$\mathbf{A} = \mathbf{QR}$$

with $\mathbf{Q}$ orthonormal, $\mathbf{R}$ upper-Δ, matching dimensions:
— $\mathbf{Q} \equiv \mathbf{Q}_1$ $n \times n$ (Unitary) & $\mathbf{R} \equiv \mathbf{R}_1$ $n \times k$
— $\mathbf{Q} \equiv \mathbf{Q}_0$ $n \times k$ & $\mathbf{R} \equiv R_0$ $k \times k$ (economical form).

We may expect good stability properties since

$$\frac{\| \, |\mathbf{Q}| \, |\mathbf{R}| \, \|_2}{\|\mathbf{A}\|_2} \leq n \qquad \left(\frac{\|\mathbf{Q}\|_2 \, \|\mathbf{R}\|_2}{\|\mathbf{A}\|_2} = 1 \right).$$

Recall that with $\mathbf{A} = \mathbf{LU}$ the stability is determined by the size of $\frac{\| \, |\mathbf{L}| \, |\mathbf{U}| \, \|}{\|\mathbf{A}\|}$.

QR-factorization

Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ be an $n \times k$ matrix.

$$\mathbf{A} = \mathbf{QR}$$

with $\mathbf{Q}$ orthonormal, $\mathbf{R}$ upper-Δ, matching dimensions:
— $\mathbf{Q} \equiv \mathbf{Q}_1$ $n \times n$ (Unitary) & $\mathbf{R} \equiv \mathbf{R}_1$ $n \times k$
— $\mathbf{Q} \equiv \mathbf{Q}_0$ $n \times k$ & $\mathbf{R} \equiv R_0$ $k \times k$ (economical form).

We may expect good stability properties since

$$\frac{\|\,|\mathbf{Q}|\,|\mathbf{R}|\,\|_2}{\|\mathbf{A}\|_2} \leq n \qquad \left(\frac{\|\mathbf{Q}\|_2\,\|\mathbf{R}\|_2}{\|\mathbf{A}\|_2} = 1 \right).$$

Costs. $\mathbf{Q}_0$ is a full matrix. $\mathbf{Q}_1^{-1} = \mathbf{Q}_1^*$.

QR-factorization

Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ be an $n \times k$ matrix.

$$\mathbf{A} = \mathbf{Q}\mathbf{R}$$

with $\mathbf{Q}$ orthonormal, $\mathbf{R}$ upper-Δ, matching dimensions:
— $\mathbf{Q} \equiv \mathbf{Q}_1$ $n \times n$ (Unitary) & $\mathbf{R} \equiv \mathbf{R}_1$ $n \times k$
— $\mathbf{Q} \equiv \mathbf{Q}_0$ $n \times k$ & $\mathbf{R} \equiv R_0$ $k \times k$ (economical form).

We may expect good stability properties since

$$\frac{\| \, |\mathbf{Q}| \, |\mathbf{R}| \, \|_2}{\|\mathbf{A}\|_2} \leq n \qquad \left(\frac{\|\mathbf{Q}\|_2 \, \|\mathbf{R}\|_2}{\|\mathbf{A}\|_2} = 1 \right).$$

[Ass.3.1]

Existence. Exists (unconditionally).

Proof: Gram–Schmidt.

The columns $\mathbf{q}_1, \ldots, \mathbf{q}_k$ of $\mathbf{Q}$ form
an orthonormal basis of Range($\mathbf{A}$) = span($\mathbf{A}$).

Constructing a QR-factorization

(classical) Gram–Schmidt:

Normalise: $\mathbf{q}_1 = \mathbf{a}_1 / \|\mathbf{a}_1\|_2$

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_2)$

Normalise: $\mathbf{q}_2 = \tilde{\mathbf{q}}_2/\|\tilde{\mathbf{q}}_2\|_2$

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\widetilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \widetilde{\mathbf{q}}_3 / \|\widetilde{\mathbf{q}}_3\|_2$

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

Then $\mathbf{A} = \mathbf{Q}R$ (economical form),

$$[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] \begin{bmatrix} \|\mathbf{a}_1\|_2 & \mathbf{q}_1^*\mathbf{a}_2 & \mathbf{q}_1^*\mathbf{a}_3 \\ 0 & \|\tilde{\mathbf{q}}_2\|_2 & \mathbf{q}_2^*\mathbf{a}_3 \\ 0 & 0 & \|\tilde{\mathbf{q}}_3\|_2 \end{bmatrix}$$

GS constructs an orthonormal basis $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ for span($\mathbf{A}$).
The upper triangular factor R appears as a side product.
The QR-factorization shows up in economical form.

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\widetilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \widetilde{\mathbf{q}}_3/\|\widetilde{\mathbf{q}}_3\|_2$

modified Gram–Schmidt:

Orthogonalise: $\widetilde{\mathbf{q}} = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3)$, $\widetilde{\mathbf{q}}_3 = \widetilde{\mathbf{q}} - \mathbf{q}_2(\mathbf{q}_2^*\widetilde{\mathbf{q}})$

Normalise: $\mathbf{q}_3 = \widetilde{\mathbf{q}}_3/\|\widetilde{\mathbf{q}}_3\|_2$

Note that $\mathbf{q}_2^*\widetilde{\mathbf{q}} = \mathbf{q}_2^*\mathbf{a}_3$, because $\mathbf{q}_2^*\mathbf{q}_1 = 0$.

In exact arithmetic: clasGS = modGS.

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

modified Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}} = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3)$, $\tilde{\mathbf{q}}_3 = \tilde{\mathbf{q}} - \mathbf{q}_2(\mathbf{q}_2^*\tilde{\mathbf{q}})$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

Householder-QR:

find $\mathbf{v}_1$ such that $\|\mathbf{v}_1\|_2 = 1$ and

$(\mathbf{I} - 2\mathbf{v}_1\mathbf{v}_1^*)\mathbf{a}_1 = \tau_1\mathbf{e}_1, \qquad \mathbf{A}^{(1)} = (\mathbf{I} - 2\mathbf{v}_1\mathbf{v}_1^*)\,\mathbf{A}.$

If $\mathbf{x}$ and $\mathbf{y}$ are normalised, then $(\mathbf{I} - 2\mathbf{v}\mathbf{v}^*)\mathbf{x} = \tau\mathbf{y}$
for some scalar τ, if $\mathbf{v} \equiv \tilde{\mathbf{v}}/\|\tilde{\mathbf{v}}\|_2$ with $\tilde{\mathbf{v}} \equiv \mathbf{x} \pm \mathbf{y}$.
For optimal stability, select sign to have $\|\tilde{\mathbf{v}}\|_2$ largest.

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

modified Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}} = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3)$, $\tilde{\mathbf{q}}_3 = \tilde{\mathbf{q}} - \mathbf{q}_2(\mathbf{q}_2^*\tilde{\mathbf{q}})$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

Householder-QR:

find $\mathbf{v}_2$ such that $\|\mathbf{v}_2\|_2 = 1$, $\mathbf{e}_1^*\mathbf{v}_2 = 0$, and

$(\mathbf{I} - 2\mathbf{v}_2\mathbf{v}_2^*)\mathbf{a}_2^{(1)} = \tau_2\mathbf{e}_2, \qquad \mathbf{A}^{(2)} = (\mathbf{I} - 2\mathbf{v}_2\mathbf{v}_2^*)\,\mathbf{A}^{(1)}.$

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

modified Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}} = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3)$, $\tilde{\mathbf{q}}_3 = \tilde{\mathbf{q}} - \mathbf{q}_2(\mathbf{q}_2^*\tilde{\mathbf{q}})$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

Householder-QR:

find $\mathbf{v}_3$ such that $\|\mathbf{v}_3\|_2 = 1$, $\mathbf{e}_1^*\mathbf{v}_3 = 0$, $\mathbf{e}_2^*\mathbf{v}_3 = 0$, and

$(\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)\mathbf{a}_3^{(2)} = \tau_3\mathbf{e}_3, \quad \mathbf{A}^{(3)} = (\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)\,\mathbf{A}^{(2)}.$

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\widetilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \widetilde{\mathbf{q}}_3/\|\widetilde{\mathbf{q}}_3\|_2$

modified Gram–Schmidt:

Orthogonalise: $\widetilde{\mathbf{q}} = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3)$, $\widetilde{\mathbf{q}}_3 = \widetilde{\mathbf{q}} - \mathbf{q}_2(\mathbf{q}_2^*\widetilde{\mathbf{q}})$

Normalise: $\mathbf{q}_3 = \widetilde{\mathbf{q}}_3/\|\widetilde{\mathbf{q}}_3\|_2$

Householder-QR:

find $\mathbf{v}_3$ such that $\|\mathbf{v}_3\|_2 = 1$, $\mathbf{e}_1^*\mathbf{v}_3 = 0$, $\mathbf{e}_2^*\mathbf{v}_3 = 0$, and

$(\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)\mathbf{a}_3^{(2)} = \tau_3\mathbf{e}_3$, $\quad \mathbf{A}^{(3)} = (\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)\,\mathbf{A}^{(2)}$.

Then $\quad [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] = \mathbf{Q}\mathbf{R}$, $\quad$ with $\mathbf{Q}$ unitary,

$\mathbf{R} \equiv \mathbf{A}^{(3)}$, $\quad n \times 3$ upper-Δ,

$\mathbf{Q} \equiv \Big((\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)(\mathbf{I} - 2\mathbf{v}_2\mathbf{v}_2^*)(\mathbf{I} - 2\mathbf{v}_1\mathbf{v}_1^*)\Big)^*$, $\; n \times n$.

Constructing a QR-factorization

(classical) Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^*\mathbf{a}_3)$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

modified Gram–Schmidt:

Orthogonalise: $\tilde{\mathbf{q}} = \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_3)$, $\tilde{\mathbf{q}}_3 = \tilde{\mathbf{q}} - \mathbf{q}_2(\mathbf{q}_2^*\tilde{\mathbf{q}})$

Normalise: $\mathbf{q}_3 = \tilde{\mathbf{q}}_3/\|\tilde{\mathbf{q}}_3\|_2$

Householder-QR:

find $\mathbf{v}_3$ such that $\|\mathbf{v}_3\|_2 = 1$, $\mathbf{e}_1^*\mathbf{v}_3 = 0$, $\mathbf{e}_2^*\mathbf{v}_3 = 0$, and

$(\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)\mathbf{a}_3^{(2)} = \tau_3\mathbf{e}_3, \quad \mathbf{A}^{(3)} = (\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*)\,\mathbf{A}^{(2)}.$

Then $[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] = \mathbf{Q}\mathbf{R}$, with $\mathbf{Q}$ unitary,

$\mathbf{R} \equiv \mathbf{A}^{(3)}, \quad n \times 3$ upper-Δ,

$\mathbf{Q} \equiv (\mathbf{I} - 2\mathbf{v}_1\mathbf{v}_1^*)(\mathbf{I} - 2\mathbf{v}_2\mathbf{v}_2^*)(\mathbf{I} - 2\mathbf{v}_3\mathbf{v}_3^*), \quad n \times n.$

QR-factorization, stability

For the computed factors $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$, we have

$$\mathbf{A} + \Delta_A = \widehat{\mathbf{Q}}\,\widehat{\mathbf{R}}$$

for some $n \times k$ Δ_A with

- $\widehat{\mathbf{R}}$ upper triangular,
- $\|\Delta_A\|_F \leq \kappa\,\mathbf{u}\,\|\mathbf{A}\|_F,$ with κ modest:

  $\kappa =$ $4k^2$ (clasGS), $4k^2$ (modGS), $4kn$ (Householder-QR).

QR-factorization, stability

For the computed factors $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$, we have

$$\mathbf{A} + \Delta_A = \widehat{\mathbf{Q}}\,\widehat{\mathbf{R}}$$

for some $n \times k$ Δ_A with

- $\widehat{\mathbf{R}}$ upper triangular,
- $\|\Delta_A\|_F \leq \kappa\,\mathbf{u}\,\|\mathbf{A}\|_F$, with κ modest,
- $\|\widehat{\mathbf{Q}}^*\widehat{\mathbf{Q}} - I\|_2 \approx \kappa\,\mathbf{u}\,(\mathcal{C}_2(\mathbf{A}))^i$ with κ of order $\sqrt{kn}$ and

$i \geq 2$ **for clasGS** (conjecture: $i = 2$)

$i = 1$ for modGS

$i = 0$ **for Householder-QR**

QR-factorization, stability

For the computed factors $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$, we have

$$\mathbf{A} + \Delta_A = \widehat{\mathbf{Q}}\,\widehat{\mathbf{R}}$$

for some $n \times k$ Δ_A with

- $\widehat{\mathbf{R}}$ upper triangular,
- $\|\Delta_A\|_F \leq \kappa\,\mathbf{u}\,\|\mathbf{A}\|_F$, with κ modest,
- $\|\widehat{\mathbf{Q}}^*\widehat{\mathbf{Q}} - I\|_2 \approx \kappa\,\mathbf{u}\,(\mathcal{C}_2(\mathbf{A}))^i$ with κ of order $\sqrt{kn}$ and

$i \geq 2$ **for clasGS** (conjecture: $i = 2$)

$i = 1$ for modGS

$i = 0$ **for Householder-QR**

Questions. Costs?
Why differences in loss of orthogonality?
Why worry about loss of orthogonality?

Intermezzo: condition numbers

For a general (possibly non-square) matrix $\mathbf{A}$, we define

$$\sigma_{\max} \equiv \max \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}, \quad \sigma_{\min} \equiv \min \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}, \quad \text{and} \quad \mathcal{C}(\mathbf{A}) \equiv \frac{\sigma_{\max}}{\sigma_{\min}},$$

where we take the max. and min. over all non-trival vectors $\mathbf{x}$ (or, equivalently, over all $\mathbf{x}$ with $\|\mathbf{x}\| = 1$).
$\mathcal{C}(\mathbf{A})$ is called the **condition number** if $\mathbf{A}$.

Note. $\sigma_{\max} = \|\mathbf{A}\|$. If $\mathbf{A}$ is square and non-singular, then

$$\sigma_{\min} = 1/\|\mathbf{A}^{-1}\| \quad \text{and} \quad \mathcal{C}(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|.$$

In case of the 2-norm,
$\sigma_{\min}$ ($\sigma_{\max}$) is the smallest (largest) **singular value** of $\mathbf{A}$.

QR-factorization, costs

Costs in case $k \ll n$ (neglecting lower order terms)

$2k^2n$ for clasGS, modGS as well as Householder QR

For Householder-QR it is assumed that $\mathbf{Q}$ is used and stored in factorized form as a product of the Householder reflections (store the $\mathbf{v}_i$). **Forming the Q by explicitly performing the product, will make Householder-QR twice as expensive** and less stable. (*Recall that in LU-factorization, forming* **L** *from the factors* $\mathbf{I} - \ell_i \mathbf{e}_i^*$ *is trivial*).

• Hence, if the vectors $\mathbf{q}_i$ are required, clasGS or modGS, are preferred over Householder QR.

• classGS allows parallelisation.

Costs in case $k = n$ (neglecting lower order terms)

for Housholder QR: $\frac{4}{3}n^3$ (twice the costs of LU fact.).

for clasGS and modGS: $2n^3$ (thrice LU).

Loss of orthogonality: Householder-QR

Householder-QR gives a unitary matrix since the Householder reflections are unitary regardless the accuracy of the vectors $\mathbf{v}_i$:
keep Q in factorized form and work with its factors.

Loss of orthogonality: GS

GS can lose orthogonality already in orthonormalizing one vector against another, say $\mathbf{a}_2$ against $\mathbf{q}_1$:

$$\widetilde{\mathbf{q}}_2 = \mathbf{a}_2 - \mathbf{q}_1(\mathbf{q}_1^*\mathbf{a}_2), \qquad \mathbf{q}_2 = \widetilde{\mathbf{q}}_2/\|\widetilde{\mathbf{q}}_2\|_2.$$

Let $\widehat{\mathbf{q}}_2 = \mathbf{q}_2 + \Delta_q$ be the computed $\mathbf{q}_2$. If δ is the error in $\mathbf{q}_1^*\mathbf{a}_2$ then $\Delta_q = \delta\,\mathbf{q}_1/\|\widetilde{\mathbf{q}}_2\|_2$ (plus other error terms):

$$\|\Delta_q\|_2 \le \frac{n\,\mathbf{u}\,\|\mathbf{a}_2\|_2}{\|\widetilde{\mathbf{q}}_2\|_2} \approx \frac{n\,\mathbf{u}}{\sin\angle(\mathbf{a}_2,\mathbf{q}_1)}.$$

Conclusion. Orthogonality is (likely to be) lost if the angle between the two vectors is small.

Remedy. If $\widetilde{\mathbf{q}}_2$ is not numerically $\mathbf{0}$ ($\mathbf{q}_1$ and $\mathbf{a}_2$ are not numerically orthogonal), then repeat the orthogonalisation:

$$\widetilde{\mathbf{q}}_2 = \widehat{\mathbf{q}}_2 - \mathbf{q}_1(\mathbf{q}_1^*\widehat{\mathbf{q}}_2), \qquad \mathbf{q}_2 = \widetilde{\mathbf{q}}_2/\|\widetilde{\mathbf{q}}_2\|_2.$$

Theorem. Twice is enough.

Loss of orthogonality: Gram-Schmidt

The strategy of GS for orthonormalizing a vector $\mathbf{a}_{k+1}$ against $\mathbf{q}_1, \ldots, \mathbf{q}_k$ relies on the assumption that $\mathbf{q}_1, \ldots, \mathbf{q}_k$ is an orthonormal system. If this assumption is not correct, then the loss of orthognality is amplified in the next vector.

Remedy. Repeat the orthogonalisation against all $\mathbf{q}_1, \ldots, \mathbf{q}_k$.

When to repeat?

DGKS: If $\angle\ \mathbf{a}_{k+1}$ and span$(\mathbf{q}_1, \ldots, \mathbf{q}_k)$ is $< 45^\circ$.

Is twice enough?
In practise, Repeated GS as stable as Householder QR.

modGS can be viewed (also in rounded arithmetic) as Householder-QR on a matrix extended at the top with a $k \times k$ block of zeros, where $\mathbf{A}$ is $n \times k$. This insight can be exploited to prove that modGS has a better orthonormalisation property than classGS

Effects of loss of orthogonality

Consider the case where $\mathbf{A}$ is square.
Let $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$ be the computed QR factors.
Put $\mathbf{E} \equiv \widehat{\mathbf{Q}}^* \widehat{\mathbf{Q}} - \mathbf{I}$ and assume $\|\mathbf{E}\|_2 < 1$.

Using the QR factors, $\mathbf{A}\mathbf{x} = \mathbf{b}$ will be solved as

$$\mathbf{y} = \widehat{\mathbf{Q}}^* \mathbf{b}, \quad \text{solve } \widehat{\mathbf{R}}\mathbf{x} = \mathbf{y} \text{ for } \mathbf{x}.$$

whereas $\mathbf{y}$ should be $\mathbf{y} = \widehat{\mathbf{Q}}^{-1} \mathbf{b}$ (given the QR factors).

Since $(\mathbf{I} + \mathbf{E})^{-1} \widehat{\mathbf{Q}}^* \widehat{\mathbf{Q}} = \mathbf{I}$, we see that

$$\widehat{\mathbf{Q}}^{-1} = (\mathbf{I} + \mathbf{E})^{-1} \widehat{\mathbf{Q}}^* \approx (\mathbf{I} - \mathbf{E}) \widehat{\mathbf{Q}}^*.$$

Hence,

$$\|\widehat{\mathbf{Q}}^* \mathbf{b} - \widehat{\mathbf{Q}}^{-1} \mathbf{b}\|_2 \approx \|\mathbf{E} \widehat{\mathbf{Q}}^* \mathbf{b}\|_2 \le \|\mathbf{E}\|_2 \|\mathbf{b}\|_2.$$

$\mathbf{E}$ could be computed,
but would make the methods more expensive!

Effects of loss of orthogonality

Consider the case where $\mathbf{A}$ is square.
Let $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$ be the computed QR factors.
Put $\mathbf{E} \equiv \widehat{\mathbf{Q}}^* \widehat{\mathbf{Q}} - \mathbf{I}$ and assume $\|\mathbf{E}\|_2 < 1$.

Using the QR factors, $\mathbf{A}\mathbf{x} = \mathbf{b}$ will be solved as

$$\mathbf{y} = \widehat{\mathbf{Q}}^* \mathbf{b}, \quad \text{solve } \widehat{\mathbf{R}}\mathbf{x} = \mathbf{y} \text{ for } \mathbf{x}.$$

whereas $\mathbf{y}$ should be $\mathbf{y} = \widehat{\mathbf{Q}}^{-1} \mathbf{b}$ (given the QR factors).

Since $(\mathbf{I} + \mathbf{E})^{-1} \widehat{\mathbf{Q}}^* \widehat{\mathbf{Q}} = \mathbf{I}$, we see that

$$\widehat{\mathbf{Q}}^{-1} = (\mathbf{I} + \mathbf{E})^{-1} \widehat{\mathbf{Q}}^* \approx (\mathbf{I} - \mathbf{E}) \widehat{\mathbf{Q}}^*.$$

Hence,

$$\|\widehat{\mathbf{Q}}^* \mathbf{b} - \widehat{\mathbf{Q}}^{-1} \mathbf{b}\|_2 \approx \|\mathbf{E} \widehat{\mathbf{Q}}^* \mathbf{b}\|_2 \leq \|\mathbf{E}\|_2 \|\mathbf{b}\|_2.$$

Effectively, the **repeated GS** variant works with $(\mathbf{I} - \mathbf{E}) \widehat{\mathbf{Q}}^*$ rather than with $\widehat{\mathbf{Q}}^*$ (or $\widehat{\mathbf{Q}}^{-1}$): $(\mathbf{I} - \widehat{\mathbf{Q}} \widehat{\mathbf{Q}}^*)^2 = \mathbf{I} - \widehat{\mathbf{Q}} (\mathbf{I} - \mathbf{E}) \widehat{\mathbf{Q}}^*$.

QR-factorisation, least square

Application. If $k < n$, then generally

solution $\mathbf{x}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$ does not exists!! [Ex.3.14]

Alternative:

$$\mathbf{x} = \text{argmin}\|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2,$$

minimising over all $\mathbf{y} \in \mathbb{C}^k$.

QR-factorisation, least square

Application. If $k < n$, then generally

solution $\mathbf{x}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$ does not exists!! [Ex.3.14]

Alternative:

$$\mathbf{x} = \text{argmin}\|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2,$$

minimising over all $\mathbf{y} \in \mathbb{C}^k$.

Terminology. 'solve $\mathbf{A}\mathbf{x} = \mathbf{b}$ in **least square sense**'.

QR-factorisation, least square

Application. If $k < n$, then generally

solution $\mathbf{x}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$ does not exists!! [Ex.3.14]

Alternative:

$$\mathbf{x} = \text{argmin}\|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2,$$

minimising over all $\mathbf{y} \in \mathbb{C}^k$.

Application areas. Computerized Tomography
X-ray CT, SPECT, old MRI, . . .
Seismography, . . .

QR-factorisation, least square

Application. If $k < n$, then generally

$$\text{solution } \mathbf{x} \text{ of } \mathbf{A}\mathbf{x} = \mathbf{b} \text{ does not exists!!}$$ [Ex.3.14]

Alternative:

$$\mathbf{x} = \text{argmin}\|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2,$$

minimising over all $\mathbf{y} \in \mathbb{C}^k$.

Lemma. $\mathcal{V}$ k-dim subspace $\mathbb{C}^n$.

$$\mathbf{b}_0 = \text{argmin}_{\mathbf{v}\in\mathcal{V}} \|\mathbf{b} - \mathbf{v}\|_2 \quad \Leftrightarrow \quad \mathbf{s} \equiv \mathbf{b} - \mathbf{b}_0 \perp \mathcal{V}$$

QR-factorisation, least square

Application. If $k < n$, then generally

$$\text{solution } \mathbf{x} \text{ of } \mathbf{A}\mathbf{x} = \mathbf{b} \text{ does not exists!!}$$ [Ex.3.14]

Alternative:

$$\mathbf{x} = \text{argmin}\|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2,$$

minimising over all $\mathbf{y} \in \mathbb{C}^k$.

Lemma. $\mathcal{V}$ k-dim subspace $\mathbb{C}^n$.

$$\mathbf{b}_0 = \text{argmin}_{\mathbf{v}\in\mathcal{V}} \|\mathbf{b} - \mathbf{v}\|_2 \quad \Leftrightarrow \quad \mathbf{s} \equiv \mathbf{b} - \mathbf{b}_0 \perp \mathcal{V}$$

$$\tan(\angle(\mathbf{b}, \mathcal{V})) = \frac{\|\mathbf{b}-\mathbf{b}_0\|_2}{\|\mathbf{b}_0\|_2}, \qquad \cos(\angle(\mathbf{b}, \mathcal{V})) = \frac{\|\mathbf{b}_0\|_2}{\|\mathbf{b}\|_2}$$

QR-factorisation, least square

Application. If $k < n$, then generally

solution $\mathbf{x}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$ does not exists!! [Ex.3.14]

Alternative:

$$\mathbf{x} = \text{argmin}\|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2,$$

minimising over all $\mathbf{y} \in \mathbb{C}^k$.

Lemma. $\mathcal{V}$ k-dim subspace $\mathbb{C}^n$.

$$\mathbf{b}_0 = \text{argmin}_{\mathbf{v} \in \mathcal{V}} \|\mathbf{b} - \mathbf{v}\|_2 \quad \Leftrightarrow \quad \mathbf{s} \equiv \mathbf{b} - \mathbf{b}_0 \perp \mathcal{V}$$

Normal equations.

$$\mathbf{x} = \text{argmin}_{\mathbf{y}}\|\mathbf{b} - \mathbf{A}\mathbf{y}\| \quad \Leftrightarrow \quad \mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}.$$

Least square, stability

$\mathbf{A}$ square, $\quad (\mathbf{A}+\Delta_A)(\mathbf{x}+\Delta_x) = \mathbf{b}+\Delta_b \;\Rightarrow$

$$\|\Delta_x\|_2 \lesssim \|\mathbf{A}^{-1}\|_2 \left(\|\Delta_b\|_2 + \|\Delta_A\|_2\,\|\mathbf{x}\|_2\right)$$

$\mathbf{A}$ is non-square, $\mathbf{x}$ solves $\mathbf{Ax}=\mathbf{b}$ in least square sense.

$$(\mathbf{A}+\Delta_A)(\mathbf{x}+\Delta_x) = \mathbf{b}+\Delta_b \qquad \text{least square}$$

$\Rightarrow$

$$\|\Delta_x\|_2 \lesssim \frac{1}{\sigma_{\text{min}}}(\|\Delta_b\|_2 + \|\Delta_A\|_2\,\|\mathbf{x}\|_2) + \frac{1}{\sigma_{\text{min}}^2}\|\Delta_A\|_2\,\|\mathbf{s}\|_2$$

Normal eq. $\quad (\mathbf{A}^*\mathbf{A}+\widetilde{\Delta}_A)(\mathbf{x}+\Delta_x) = \mathbf{A}^*\mathbf{b}+\widetilde{\Delta}_b$

$\Rightarrow$
$$\frac{1}{\sigma_{\text{min}}(\mathbf{A}^*\mathbf{A})} = \frac{1}{\sigma_{\text{min}}(\mathbf{A})^2}$$

$$\|\Delta_x\|_2 \lesssim \frac{1}{\sigma_{\text{min}}^2}(\|\widetilde{\Delta}_b\|_2 + \|\widetilde{\Delta}_A\|_2\,\|\mathbf{x}\|_2)$$

Normal equations

- **Stability.** $\mathcal{C}_2(\mathbf{A})^2$ determines stability; $\mathcal{C}_2(\mathbf{A}) = \frac{\sigma_{\max}}{\sigma_{\min}}$.
- **Costs.** Formation $\mathbf{A}^*\mathbf{A}$ forms 'extra' costs.

Alternative. $\mathbf{A} = \mathbf{Q}_0\mathbf{R}_0 \;\Rightarrow$

$$\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{R}_0^*\mathbf{R}_0\mathbf{x} = \mathbf{R}_0^*\mathbf{Q}_0^*\mathbf{b} \;\Rightarrow \qquad \boxed{\mathbf{R}_0\mathbf{x} = \mathbf{Q}_0^*\mathbf{b}}$$

Note. $\mathbf{R}_0^*\mathbf{R}_0$ is Cholesky's decomposition $\mathbf{A}^*\mathbf{A}$

Normal equations

− **Stability.** $\mathcal{C}_2(\mathbf{A})^2$ determines stability; $\mathcal{C}_2(\mathbf{A}) = \frac{\sigma_{\max}}{\sigma_{\min}}$.

− **Costs.** Formation $\mathbf{A}^*\mathbf{A}$ forms 'extra' costs.

Alternative. $\mathbf{A} = \mathbf{Q}_0\mathbf{R}_0 \;\Rightarrow$

$\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{R}_0^*\mathbf{R}_0\mathbf{x} = \mathbf{R}_0^*\mathbf{Q}_0^*\mathbf{b} \;\Rightarrow$ $\boxed{\mathbf{R}_0\mathbf{x} = \mathbf{Q}_0^*\mathbf{b}}$

\+ **Stability.** $\mathcal{C}(\mathbf{R}_0)$ determines stability: $\mathcal{C}(\mathbf{R}_0) = \mathcal{C}(\mathbf{A})$ except for the component $\mathbf{b} - \mathbf{Q}_0\mathbf{Q}_0^*\mathbf{b} = \mathbf{b} - \mathbf{A}\mathbf{x}$:
Optimal stability.

\+ **Costs.**

QR versus LU

For small ($n < 10000$), dense systems:

LU. + easy and cheap to compute

+ easy and cheap to work with

− stability requires permutation (and scaling)

QR. o easy and cheap to compute, but 2$\times$ the costs LU

o easy and cheap to work with, but 1.5$\times$ the costs LU

+ stable

For large n, sparse systems
both factorizations destroy sparsity structure. However,
LU: + $\exists$ effective incomplete LU with sparsity structure,
QR: − no effective incomplete QR with sparsity structure.