Rank Revealing QR factorization

F. Guyomarc'h, D. Mezher and B. Philippe

Outline

- Introduction
- Classical Algorithms
  - ⋆ Full matrices
  - ⋆ Sparse matrices
- Rank-Revealing QR
- Conclusion

Situation of the problem

See Bjorck SIAM96 : Numerical Methods for Least Squares Problems.

Data : $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

Problem $\mathcal{P}$: Find $x \in \mathcal{S}$ such that $\|x\|$ minimum
where $\mathcal{S} = \{x \in \mathbb{R}^n \mid \|b - Ax\| \text{ minimum } \}$.

$(\|.\| \equiv \|.\|_2)$

The solution is unique : $\hat{x} = A^+ b$.

Property :

$$\begin{aligned} x \in \mathcal{S} \quad &\text{iff} \quad A^T(b - Ax) = 0, \\ &\text{iff} \quad A^T A x = A^T b. \end{aligned}$$

The last equation is consistent since $\mathcal{R}(A^T A) = \mathcal{R}(A^T)$.

Rank-Revealing QR factorization

Theorem : $A \in \mathbb{R}^{m \times n}$ with rank$(A) = r$ $(r \leq \min(m,n))$.

There exist Q, E, R_{11} and R_{12} such that

- $Q \in \mathbb{R}^{m \times m}$ is orthogonal,
- $E \in \mathbb{R}^{n \times n}$ is a permutation,
- $R_{11} \in \mathbb{R}^{r \times r}$ is upper-triangular with positive diagonal entries,
- $R_{12} \in \mathbb{R}^{r \times (n-r)}$,

and $AE = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix}$.

RRQR

The factorization is not unique. Let RRQR be any of them.

Actually, if we consider a complete orthogonal decomposition

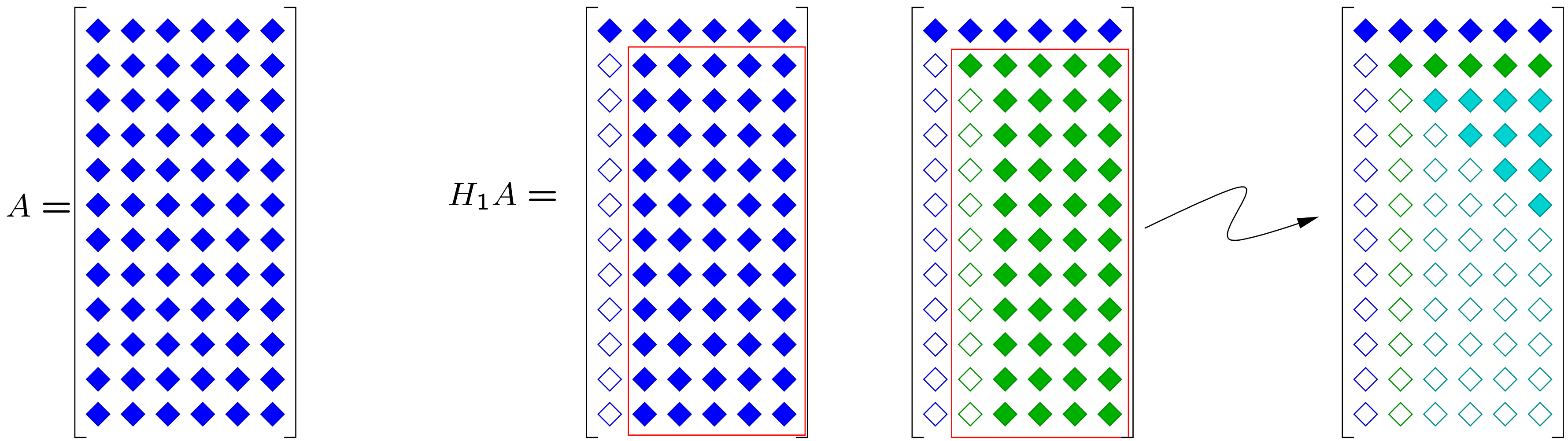
$$A = Q \begin{pmatrix} T & 0 \\ 0 & 0 \end{pmatrix} V^T$$

where Q and V are orthogonal and T triangular with positive diagonal entries, we have $A^+ = V \begin{pmatrix} T^{-1} & 0 \\ 0 & 0 \end{pmatrix} Q^T$.

Such a factorization can be obtained from the previous one by performing a QR factorization of $\begin{pmatrix} R_{11}^T \\ R_{12}^T \end{pmatrix}$

Column pivoting strategy

Householder QR factorization using Householder reflections :



$v = A_{1..12,1} + ||A_{1..12,1}||e_1$
$H_1 = I_{12} - 2\frac{vv^t}{v^t v}$

$v = A_{2..12,2} + ||A_{2..12,2}||e_1$
$H_2 = I_{11} - 2\frac{vv^t}{v^t v}$

Sparse QR factorization

Factorizing a sparse matrix implies fill-in in the factors.

The situation is worse with QR than with LU since when updating the tailing matrix :

- LU : the elementary transformation $x \longrightarrow y = (I - \alpha v e_k^T)x$ keeps x invariant when $x_k = 0$.
- QR : the elementary transformation $x \longrightarrow y = (I - \alpha v v^T)x$ keeps x invariant when $x \perp v$.

Since $A^T A = R^T R$, the QR factorization A is related to the Cholesky factorization of $A^T A$. It is known that a symmetric permutation on a sparse s.p.d. matrix changes the level of fill-in.

Therefore, a permutation of the columns of A changes the fill-in of R.

$\Rightarrow$ there is conflict between pivoting to minimize fill-ins and pivoting associated with numerical properties.

We choose to decouple the sparse factorization phase and the rank-revealing phase

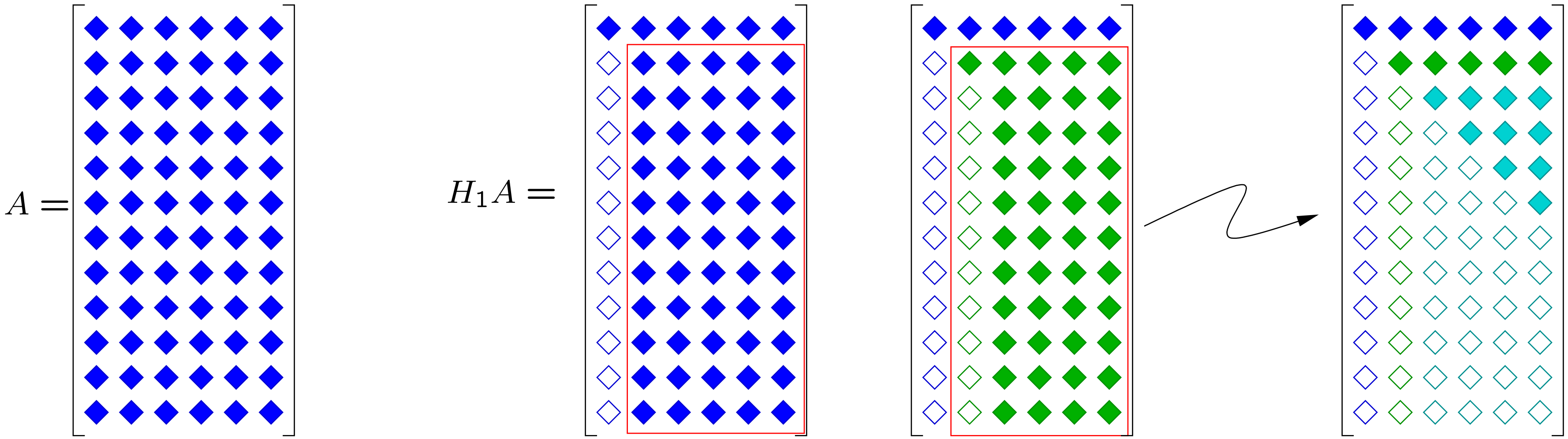
For a standard QR factorization of a sparse matrix :

Multi-frontal QR method [Amestoy-Duff-Puglisi '94]

Routine MA49AD in Library HSL.

Column pivoting strategy

Householder QR factorization using Householder reflections :



$v = A_{1..12,1} + ||A_{1..12,1}||e_1$
$H_1 = I_{12} - 2\frac{vv^t}{v^t v}$

$v = A_{2..12,2} + ||A_{2..12,2}||e_1$
$H_2 = I_{11} - 2\frac{vv^t}{v^t v}$

Householder QR with Column Pivoting (Businger and Golub) :

At step k, the column $j_k \geq k$ defining the Householder transformation is chosen so that $\|A(k:m, j_k)\| \geq \|A(k:m, j)\|$ for $j \geq k$.

Some properties on R_{11}

At step k :

$$A^{(k)} \equiv H_k \cdots H_1 A E(k) = \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix}.$$

Any column a of $\begin{pmatrix} R_{12}^{(k)}(k,:) \\ A_{22}^{(k)} \end{pmatrix}$ satisfies $R_{11}^{(k)}(k,k) \geq \|a\|$.

Moreover $\|A\| \geq \|R_{11}^{(k)}\| \geq R_{11}^{(k)}(1,1)$
$R_{11}^{(k)}(k,k) \geq \sigma_{\min}(R_{11}^{(k)}) \geq \sigma_{\min}(A)$,

This implies that : $cond(A) \geq cond(R_{11}^{(k)}) \geq \frac{R_{11}^{(k)}(1,1)}{R_{11}^{(k)}(k,k)}$.

Bad new

The quantity $\frac{R_{11}^{(k)}(1,1)}{R_{11}^{(k)}(k,k)}$ cannot be considered as an estimator of $cond(R_{11}^{(k)})$ since it can be of different order of magnitude :

Example (Kahan's matrix of order n) :

for $\theta \in (0, \frac{\pi}{2})$, $c = \cos\theta$ and $s = \arcsin\theta$:

`MATLAB :` d=c .^ [0:n-1] ; M=diag(d)*(eye(n)-s*triu(ones(n),1));

n=100 ; $\theta = \arcsin(0.2)$:

$$\sigma_{\min}(R_{11}) = 3.7e-9 \ll R_{11}(100,100) = 1.3e-1$$

Therefore a better estimate of $cond(R_{11}^{(k)})$ is needed.

Incremental Condition Estimator (ICE) [Bischof 90]
which is implemented in LAPACK :
it estimates $cond(R_{11}^{(k)})$ from an estimation of $cond(R_{11}^{(k-1)})$.

However, the strategy which consists in stopping the factorization when

$$cond(R_{11}^{(k)}) < \frac{1}{\epsilon} \leq cond(R_{11}^{(k+1)})$$

may fail in very special situations :
Counterexample :
M = Kahan's matrix of order 30 with $\theta = \arccos(0.2)$

$$cond(R_{11}^{(16)}) < \frac{1}{\epsilon} < cond(R_{11}^{(17)})$$

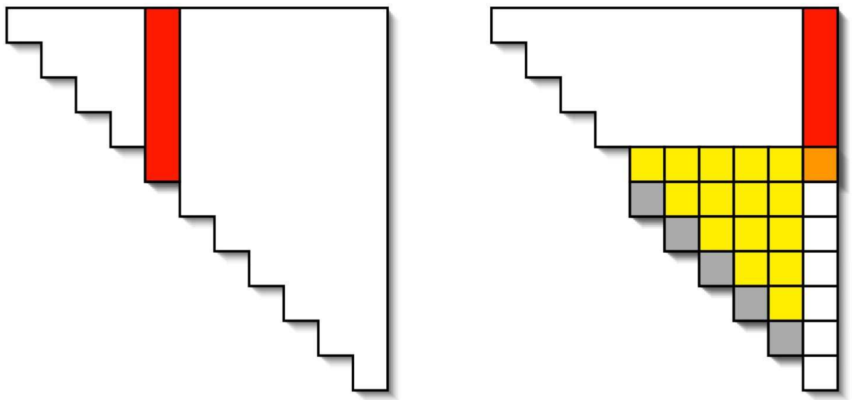
indicates a numerical rank equal to 16 although the numerical rank of M computed by SVD is 22.

Pivoting strategies

Convert R to reveal the rank by pushing the singularities towards the right end

- Apply an Incremental Condition Estimator to evaluate σ_{max} and σ_{min} of $R_{1\to i,1\to i}$,

- if $\sigma_{\text{max}}/\sigma_{\text{min}} > \tau$, move column i towards the right end and re-orthogonalise R using Givens rotations



Pivoting strategies

- $RRQR$ uses a set of thresholds to overcome numerical singularities,

- A predifined set of thresholds might fail

$$A = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 10^{-6} & 1 & 1 \\ 0 & 0 & 10^{-3} & -10^{-3} \end{bmatrix}$$

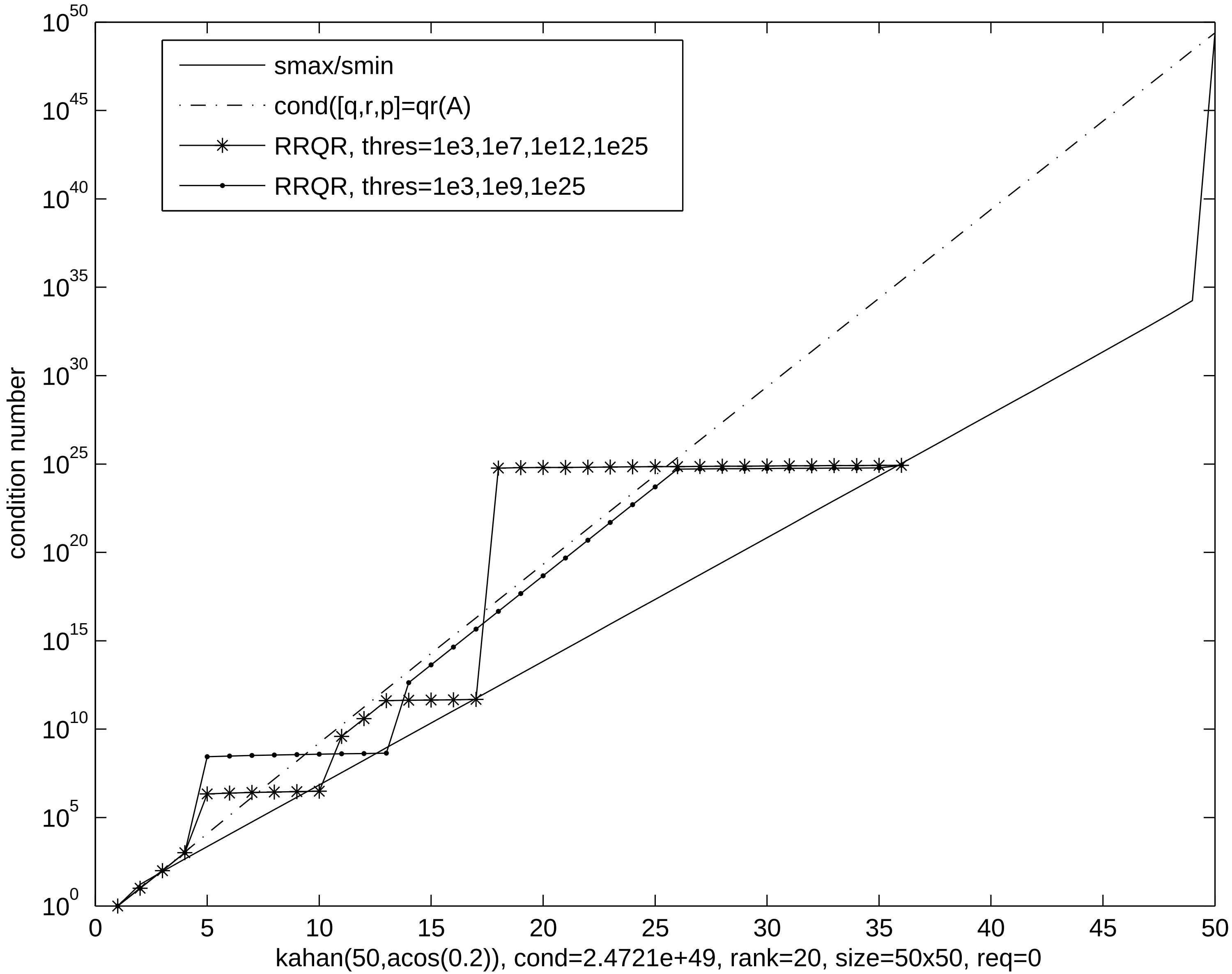
  rank=2 if $\tau = \{10^7\}$, rank=3 if $\tau = \{10^4, 10^7\}$

- $RRQR$ adapts the thresholds to avoid failure

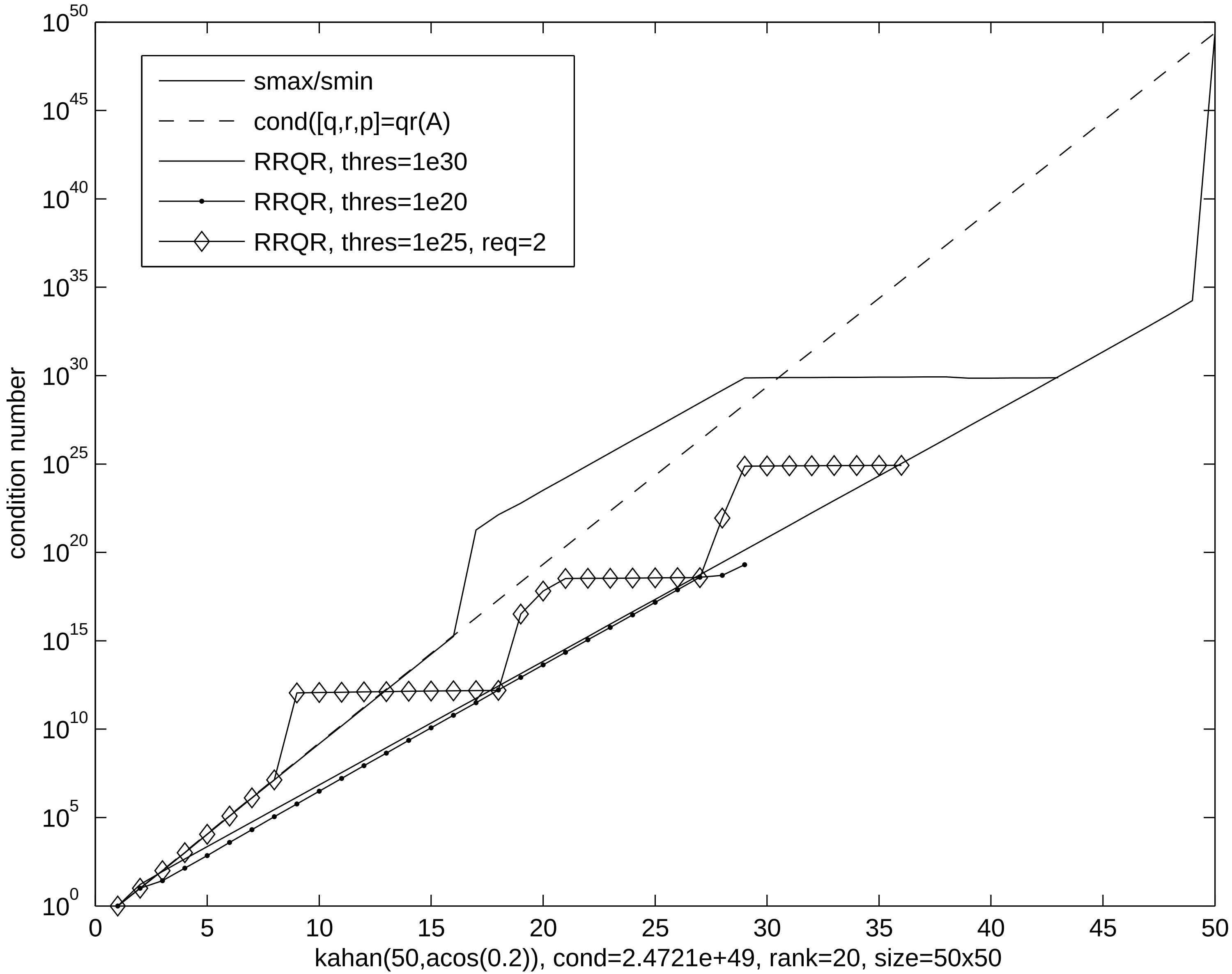
  ⋆ Upon completion, check if $\left\|R_{22}\right\| < \frac{\sigma_{\max}}{\tau} \simeq \sigma_{\min}$

  ⋆ On failure, insert new thresholds and restart algorithm

Numerical results

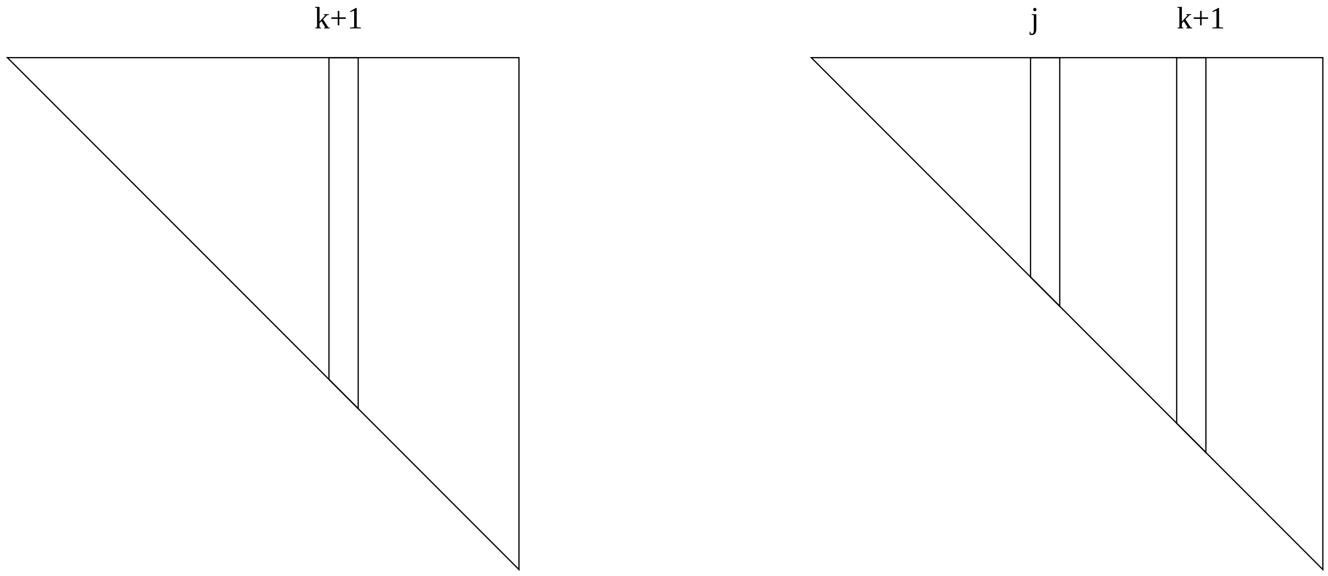


Numerical results



Second strategy

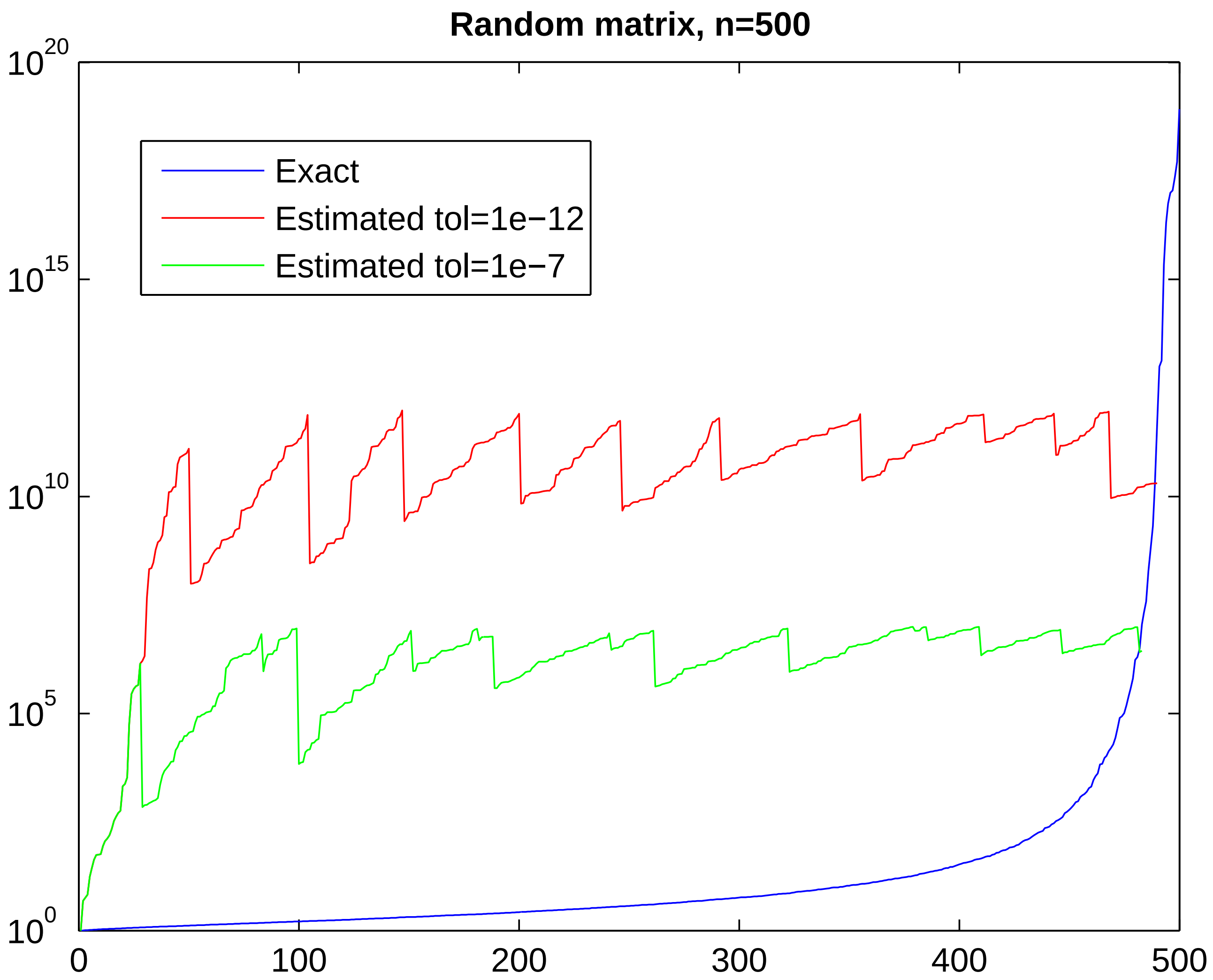
The $k+1$ column is not the worst usually for the condition number.
Can we select the worst one for the conditionement of R_{11} ?
Then we reject this worst column to the end.



We need to recompute the singular values and vectors estimates.

Reverse ICE

Second strategy



Conclusion

Work is still on progress :

Case where R is sparse

- ICE might fail, so does the Reverse ICE.
- Using the elimination tree, we can try to keep R as sparse as possible.