Rank revealing factorizations and low rank approximations

L. Grigori

EPFL and PSI

November 19, 2024

Plan

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Plan

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Rank revealing QR factorization

Given A of size $m \times n, m \geq n$, consider the decomposition

$$A\Pi_c = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}, \tag{1}$$

where R_{11} is $k \times k$, permutation matrix Π_c and k are chosen such that $||R_{22}||_2$ is small and R_{11} is well-conditioned.

- By the interlacing property of singular values [Golub, Van Loan, 4th edition, page 487],

  $$\sigma_i(R_{11}) \leq \sigma_i(A) \text{ and } \sigma_j(R_{22}) \geq \sigma_{k+j}(A)$$

  for $1 \leq i \leq k$ and $1 \leq j \leq n-k$.
- $\sigma_{k+1}(A) \leq \sigma_{max}(R_{22}) = ||R_{22}||_2$

Rank revealing QR factorization

Given A of size $m \times n$, consider the decomposition

$$A\Pi_c = QR = Q \left[\begin{array}{cc} R_{11} & R_{12} \\ & R_{22} \end{array} \right]. \tag{2}$$

If $||R_{22}||_2$ is small,

- $Q(:, 1:k)$ forms an approximate orthogonal basis for the range of A,

$$\begin{aligned} A(:,j) &= \sum_{i=1}^{min(j,k)} R(i,j) Q(:,i) \quad \in span\{Q(:,1), \ldots Q(:,k)\} \\ Range(A) &\in span\{Q(:,1), \ldots Q(:,k)\} \end{aligned}$$

- $\Pi_c \begin{bmatrix} -R_{11}^{-1} R_{12} \\ I \end{bmatrix}$ is an approximate right null space of A.

Rank revealing QR factorization

The factorization from equation (1) is rank revealing if

$$1 \le \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \le \gamma_1(n,k),$$

for $1 \le i \le k$ and $1 \le j \le \min(m,n) - k$, where

$$\sigma_{max}(A) = \sigma_1(A) \ge \ldots \ge \sigma_{min}(A) = \sigma_n(A)$$

It is **strong** rank revealing [Gu and Eisenstat, 1996] if in addition

$$||R_{11}^{-1} R_{12}||_{max} \le \gamma_2(n,k),$$

where $\gamma_1(n,k), \gamma_2(n,k)$ are low degree polynomials in n and k.

Low rank approximation with strong RRQR

Given $A \in \mathbb{R}^{m \times n}$ and $R_{11} \in \mathbb{R}^{k \times k}$,

$$
\begin{aligned}
A\Pi_c &= QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}, \\
A_{qr} &= Q_1 \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} \Pi_c^T = Q_1 Q_1^T A = \mathcal{P}^o A
\end{aligned}
$$

- It can be shown that

$$\sigma_j(R_{22}) = \sigma_j(A - A_{qr})$$

- [Gu and Eisenstat, 1996] show that given k and f, there exists permutation $\Pi_c \in \mathbb{R}^{n \times n}$ such that the factorization satisfies,

$$
\begin{aligned}
1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} &\leq \gamma(n,k), \quad \gamma(n,k) = \sqrt{1 + f^2 k(n-k)} \\
||R_{11}^{-1} R_{12}||_{max} &\leq f
\end{aligned}
$$

  for $1 \leq i \leq k$ and $1 \leq j \leq \min(m,n) - k$.
- Cost: $4mnk$ (QRCP) plus $O(mnk)$ flops and $O(k \log_2 P)$ messages.

$\rightarrow$ A_{qr} with strong RRQR is $(k, \gamma(n,k))$ spectrum preserving and kernel approximation of A

strong RRQR (contd)

Given $A \in \mathbb{R}^{m \times n}$ and $R_{11} \in \mathbb{R}^{k \times k}$,

$$\begin{aligned} A\Pi_c &= QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}, \\ A_{qr} &= Q_1 \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} \Pi_c^T = Q_1 Q_1^T A = \mathcal{P}^o A \end{aligned}$$

- We show that

$$\sigma_j(R_{22}) = \sigma_j(A - A_{qr})$$

$$\sigma_j(A - A_{qr}) = \sigma_j(A - Q_1 Q_1^T A) = \sigma_j(Q_2 Q_2^T A) = \sigma_j(Q_2(0\ R_{22})\Pi_c^{-1}) = \sigma_j(R_{22})$$

QR with column pivoting [Businger and Golub, 1965]

Idea:

- At first iteration, trailing columns decomposed into parallel part to first column (or e_1) and orthogonal part (in rows $2:m$).
- The column of maximum norm is the column with largest component orthogonal to the first column.

Implementation:

- Find at each step of the QR factorization the column of maximum norm.
- Permute it into leading position.
- If rank(A) = k, at step $k+1$ the maximum norm is 0.
- No need to compute the column norms at each step, but just update them since

$$Q^T v = w = \begin{bmatrix} w_1 \\ w(2:n) \end{bmatrix}, \;\; ||w(2:n)||_2^2 = ||v||_2^2 - w_1^2$$

QR with column pivoting [Businger and Golub, 1965]

Sketch of the algorithm

column norm vector: $colnrm(j) = ||A(:,j)||_2, j = 1 : n$.
for j = 1 : n **do**
  Find column p of largest norm
  if $colnrm[p] > \epsilon$ **then**
    1. Pivot: swap columns j and p in A and modify $colnrm$.
    2. Compute Householder matrix H_j s.t. $H_j A(j : m, j) = \pm ||A(j : m, j)||_2 e_1$.
    3. Update $A(j : m, j+1 : n) = H_j A(j : m, j+1 : n)$.
    4. Norm downdate $colnrm(j+1 : n)^2 -= A(j, j+1 : n)^2$.
  else Break
  end if
end for

If algorithm stops after k steps

$$\sigma_{max}(R_{22}) \leq \sqrt{n-k} \max_{1 \leq j \leq n-k} ||R_{22}(:,j)||_2 \leq \sqrt{n-k}\epsilon$$

Strong RRQR [Gu and Eisenstat, 1996]

Since

$$det(R_{11}) = \prod_{i=1}^{k} \sigma_i(R_{11}) = \sqrt{det(A^T A)} / \prod_{i=1}^{n-k} \sigma_i(R_{22})$$

a strong RRQR is related to a large $det(R_{11})$. The following algorithm interchanges columns that increase $det(R_{11})$, given f and k.

Compute a strong RRQR factorization, given k:

Compute $A\Pi_c = QR$ by using QRCP
while there exist i and j such that $det(\tilde{R}_{11})/det(R_{11}) > f$, where
$R_{11} = R(1:k, 1:k)$, $\Pi_{i,j+k}$ permutes columns i and $j+k$,
$R\Pi_{i,j+k} = \tilde{Q}\tilde{R}, \tilde{R}_{11} = \tilde{R}(1:k, 1:k)$ **do**
Find i and j
Compute $R\Pi_{i,j+k} = \tilde{Q}\tilde{R}$ and $\Pi_c = \Pi_c \Pi_{i,j+k}$
end while

Strong RRQR (contd)

It can be shown that

$$\frac{det(\tilde{R}_{11})}{det(R_{11})} = \sqrt{\left(R_{11}^{-1}R_{12}\right)^2_{i,j} + \rho_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} \tag{3}$$

for any $1 \leq i \leq k$ and $1 \leq j \leq n-k$ (the 2-norm of the j-th column of A is $\chi_j(A)$, and the 2-norm of the j-th row of A^{-1} is $\rho_j(A)$).

Compute a strong RRQR factorization, given k:

Compute $A\Pi_c = QR$ by using QRCP

while $\max_{1\leq i\leq k, 1\leq j\leq n-k} \sqrt{\left(R_{11}^{-1}R_{12}\right)^2_{i,j} + \rho_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} > f$ **do**

Find i and j such that $\sqrt{\left(R_{11}^{-1}R_{12}\right)^2_{i,j} + \rho_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} > f$

Compute $R\Pi_{i,j+k} = \tilde{Q}\tilde{R}$ and $\Pi_c = \Pi_c\Pi_{i,j+k}$

end while

Strong RRQR (contd)

- $det(R_{11})$ strictly increases with every permutation, no permutation repeats, hence there is a finite number of permutations to be performed.

Strong RRQR (contd)

Theorem

[Gu and Eisenstat, 1996] If the QR factorization with column pivoting as in equation (1) *satisfies inequality*

$$\sqrt{\left(R_{11}^{-1}R_{12}\right)_{i,j}^{2} + \rho_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} < f$$

for any $1 \leq i \leq k$ *and* $1 \leq j \leq n-k$*, then*

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \sqrt{1 + f^2k(n-k)},$$

for any $1 \leq i \leq k$ *and* $1 \leq j \leq \min(m,n) - k$.

Sketch of the proof ([Gu and Eisenstat, 1996])

Assume A is full column rank. Let $\alpha = \sigma_{max}(R_{22})/\sigma_{min}(R_{11})$, and let

$$R = \begin{bmatrix} R_{11} & \\ & R_{22}/\alpha \end{bmatrix} \begin{bmatrix} I_k & R_{11}^{-1}R_{12} \\ & \alpha I_{n-k} \end{bmatrix} = \tilde{R}_1 W_1.$$

We have

$$\sigma_i(R) \leq \sigma_i(\tilde{R}_1)||W_1||_2, 1 \leq i \leq n.$$

Since $\sigma_{min}(R_{11}) = \sigma_{max}(R_{22}/\alpha)$, then $\sigma_i(\tilde{R}_1) = \sigma_i(R_{11})$, for $1 \leq i \leq k$.

$$\begin{aligned} ||W_1||_2^2 &\leq 1 + ||R_{11}^{-1}R_{12}||_2^2 + \alpha^2 = 1 + ||R_{11}^{-1}R_{12}||_2^2 + ||R_{22}||_2^2||R_{11}^{-1}||_2^2 \\ &\leq 1 + ||R_{11}^{-1}R_{12}||_F^2 + ||R_{22}||_F^2||R_{11}^{-1}||_F^2 \\ &= 1 + \sum_{i=1}^{k}\sum_{j=1}^{n-k}\left((R_{11}^{-1}R_{12})_{i,j}^2 + \rho_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)\right) \leq 1 + f^2k(n-k) \end{aligned}$$

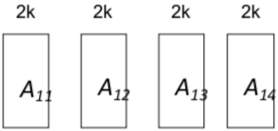
We obtain,

$$\frac{\sigma_i(A)}{\sigma_i(R_{11})} \leq \sqrt{1 + f^2k(n-k)}$$

Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)

- 1D column block partition of A, select k cols from each block with strong RRQR



$$
\begin{array}{cccc}
(\; A_{11} & A_{12} & A_{13} & A_{14} \;) \\
\| & \| & \| & \| \\
(\; Q_{00} R_{00} \Pi_{c00}^T & Q_{10} R_{10} \Pi_{c10}^T & Q_{20} R_{20} \Pi_{c20}^T & Q_{30} R_{30} \Pi_{c30}^T \\
\downarrow & \downarrow & \downarrow & \downarrow \\
I_{00} & I_{10} & I_{20} & I_{30}
\end{array}
$$

- Reduction tree to select k cols from sets of $2k$ cols,

$$
\begin{array}{cc}
(\; A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30});\;) \\
\| & \| \\
(\; Q_{01} R_{01} \Pi_{c01}^T & Q_{11} R_{11} \Pi_{c11}^T \;) \\
\downarrow & \downarrow \\
I_{01} & I_{11}
\end{array}
$$

$$
A(:, I_{01} \cup I_{11}) = Q_{02} R_{02} \Pi_{c02}^T \rightarrow I_{02}
$$

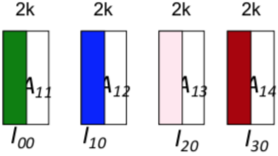
- Return selected columns $A(:, I_{02})$

Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)

- 1D column block partition of A, select k cols from each block with strong RRQR

$$
\begin{array}{cccc}
(\; A_{11} & A_{12} & A_{13} & A_{14} \;) \\
\| & \| & \| & \| \\
(\; Q_{00}R_{00}\Pi_{c00}^T & Q_{10}R_{10}\Pi_{c10}^T & Q_{20}R_{20}\Pi_{c20}^T & Q_{30}R_{30}\Pi_{c30}^T \\
\downarrow & \downarrow & \downarrow & \downarrow \\
I_{00} & I_{10} & I_{20} & I_{30}
\end{array}
$$



- Reduction tree to select k cols from sets of $2k$ cols,

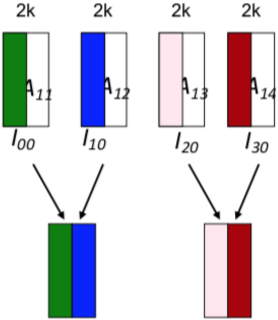
$$
\begin{array}{cc}
(\; A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30}); \;) \\
\| & \| \\
(\; Q_{01}R_{01}\Pi_{c01}^T & Q_{11}R_{11}\Pi_{c11}^T \;) \\
\downarrow & \downarrow \\
I_{01} & I_{11}
\end{array}
$$

$$A(:, I_{01} \cup I_{11}) = Q_{02}R_{02}\Pi_{c02}^T \rightarrow I_{02}$$

- Return selected columns $A(:, I_{02})$

Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)



- 1D column block partition of A, select k cols from each block with strong RRQR

$$\begin{array}{cccc} (A_{11} & A_{12} & A_{13} & A_{14}) \\ \| & \| & \| & \| \\ (Q_{00}R_{00}\Pi_{c00}^T & Q_{10}R_{10}\Pi_{c10}^T & Q_{20}R_{20}\Pi_{c20}^T & Q_{30}R_{30}\Pi_{c30}^T \\ \downarrow & \downarrow & \downarrow & \downarrow \\ I_{00} & I_{10} & I_{20} & I_{30} \end{array}$$

- Reduction tree to select k cols from sets of $2k$ cols,

$$\begin{array}{cc} (A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30});) \\ \| & \| \\ (Q_{01}R_{01}\Pi_{c01}^T & Q_{11}R_{11}\Pi_{c11}^T) \\ \downarrow & \downarrow \\ I_{01} & I_{11} \end{array}$$

$$A(:, I_{01} \cup I_{11}) = Q_{02}R_{02}\Pi_{c02}^T \rightarrow I_{02}$$

- Return selected columns $A(:, I_{02})$

Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)

- 1D column block partition of A, select k cols from each block with strong RRQR

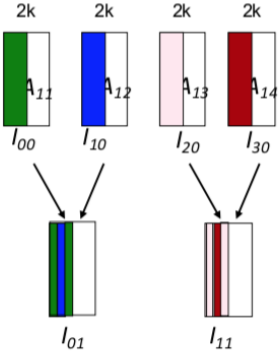
$$\begin{array}{cccc} (A_{11} & A_{12} & A_{13} & A_{14}) \\ \| & \| & \| & \| \\ (Q_{00}R_{00}\Pi_{c00}^T & Q_{10}R_{10}\Pi_{c10}^T & Q_{20}R_{20}\Pi_{c20}^T & Q_{30}R_{30}\Pi_{c30}^T) \\ \downarrow & \downarrow & \downarrow & \downarrow \\ I_{00} & I_{10} & I_{20} & I_{30} \end{array}$$

- Reduction tree to select k cols from sets of $2k$ cols,

$$\begin{array}{cc} (A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30});) \\ \| & \| \\ (Q_{01}R_{01}\Pi_{c01}^T & Q_{11}R_{11}\Pi_{c11}^T) \\ \downarrow & \downarrow \\ I_{01} & I_{11} \end{array}$$

$$A(:, I_{01} \cup I_{11}) = Q_{02}R_{02}\Pi_{c02}^T \rightarrow I_{02}$$

- Return selected columns $A(:, I_{02})$



Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)

- 1D column block partition of A, select k cols from each block with strong RRQR

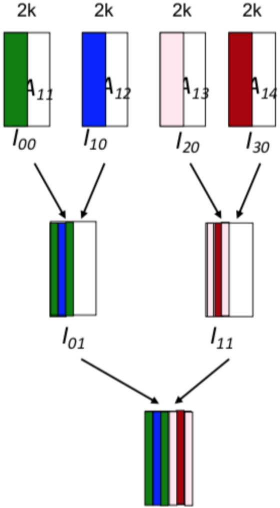
$$
\begin{array}{cccc}
(\; A_{11} & A_{12} & A_{13} & A_{14} \;) \\
\| & \| & \| & \| \\
(\; Q_{00}R_{00}\Pi_{c_{00}}^T & Q_{10}R_{10}\Pi_{c_{10}}^T & Q_{20}R_{20}\Pi_{c_{20}}^T & Q_{30}R_{30}\Pi_{c_{30}}^T \;) \\
\downarrow & \downarrow & \downarrow & \downarrow \\
I_{00} & I_{10} & I_{20} & I_{30}
\end{array}
$$

- Reduction tree to select k cols from sets of $2k$ cols,

$$
\begin{array}{cc}
(\; A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30}); \;) \\
\| & \| \\
(\; Q_{01}R_{01}\Pi_{c_{01}}^T & Q_{11}R_{11}\Pi_{c_{11}}^T \;) \\
\downarrow & \downarrow \\
I_{01} & I_{11}
\end{array}
$$

$$A(:, I_{01} \cup I_{11}) = Q_{02}R_{02}\Pi_{c_{02}}^T \rightarrow I_{02}$$

- Return selected columns $A(:, I_{02})$



Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)

- 1D column block partition of A, select k cols from each block with strong RRQR

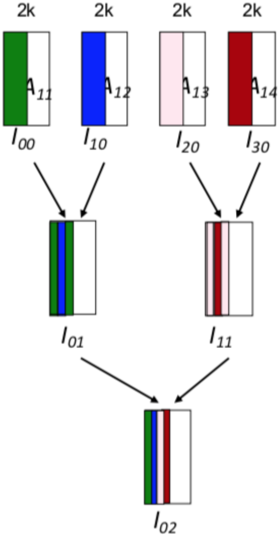
$$
\begin{array}{cccc}
(\; A_{11} & A_{12} & A_{13} & A_{14} \;) \\
\| & \| & \| & \| \\
(\; Q_{00}R_{00}\Pi_{c_{00}}^T & Q_{10}R_{10}\Pi_{c_{10}}^T & Q_{20}R_{20}\Pi_{c_{20}}^T & Q_{30}R_{30}\Pi_{c_{30}}^T \;) \\
\downarrow & \downarrow & \downarrow & \downarrow \\
I_{00} & I_{10} & I_{20} & I_{30}
\end{array}
$$

- Reduction tree to select k cols from sets of $2k$ cols,

$$
\begin{array}{cc}
(\; A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30}); \;) \\
\| & \| \\
(\; Q_{01}R_{01}\Pi_{c_{01}}^T & Q_{11}R_{11}\Pi_{c_{11}}^T \;) \\
\downarrow & \downarrow \\
I_{01} & I_{11}
\end{array}
$$

$$A(:, I_{01} \cup I_{11}) = Q_{02}R_{02}\Pi_{c_{02}}^T \rightarrow I_{02}$$

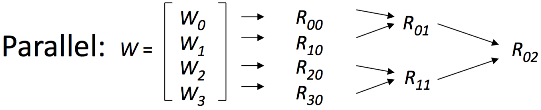
- Return selected columns $A(:, I_{02})$



Select k columns from a tall and skinny matrix

Given W of size $m \times 2k$, $m >> k$, k columns are selected as:

$W = QR_{02}$ using TSQR
$R_{02}\Pi_c = Q_2 R_2$ using QRCP
Return $W\Pi_c(:, 1:k)$



Reduction trees

Any shape of reduction tree can be used during CA_RRQR, depending on the underlying architecture.

- Flat tree:

$$\begin{tikzcd}
A_{00} \arrow[d] & A_{10} \arrow[ddl] & A_{20} \arrow[dddll] & A_{30} \arrow[ddddlll] \\
f(A_{00}) \arrow[d] & & & \\
f(A_{01}) \arrow[d] & & & \\
f(A_{02}) \arrow[d] & & & \\
f(A_{03}) & & &
\end{tikzcd}$$

- Binary tree:

$$\begin{tikzcd}
A_{00} \arrow[d] & & A_{10} \arrow[d] & & A_{20} \arrow[d] & & A_{30} \arrow[d] \\
f(A_{00}) \arrow[dr] & & f(A_{10}) \arrow[dl] & & f(A_{20}) \arrow[dr] & & f(A_{30}) \arrow[dl] \\
& f(A_{01}) \arrow[drr] & & & & f(A_{11}) \arrow[dll] & \\
& & & f(A_{02}) & & &
\end{tikzcd}$$

Notation: at each node of the reduction tree, $f(A_{ij})$ returns the first b columns obtained after performing (strong) RRQR of A_{ij}.

Rank revealing properties of tournament pivoting

It is shown in [Demmel et al., 2015] that the column permutation computed by CA-RRQR satisfies

$$\chi_j^2\left(R_{11}^{-1}R_{12}\right) + \left(\chi_j\left(R_{22}\right)/\sigma_{\min}(R_{11})\right)^2 \leq F_{TP}^2, \text{ for } j = 1, \ldots, n-k. \quad (4)$$

where F_{TP} depends on k, f, n, the shape of reduction tree used during tournament pivoting, and the number of iterations of CARRQR.

CA-RRQR - bounds for one tournament

Selecting k columns by using tournament pivoting reveals the rank of A with the following bounds:

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \sqrt{1 + F_{TP}^2(n-k)},$$

$$||R_{11}^{-1}R_{12}||_{max} \leq F_{TP}$$

- Binary tree of depth $\log_2(n/k)$,

$$F_{TP} \leq \frac{1}{\sqrt{2k}} (n/k)^{\log_2\left(\sqrt{2}fk\right)}. \tag{5}$$

  The upper bound is a decreasing function of k when $k > \sqrt{n/(\sqrt{2}f)}$.

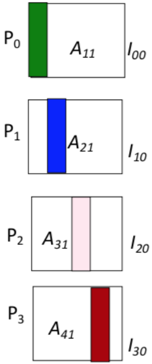
- Flat tree of depth n/k,

$$F_{TP} \leq \frac{1}{\sqrt{2k}} \left(\sqrt{2}fk\right)^{n/k}. \tag{6}$$

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00}R_{00}\Pi_{c_{00}}^{-1} \\ Q_{10}R_{10}\Pi_{c_{10}}^{-1} \\ Q_{20}R_{20}\Pi_{c_{20}}^{-1} \\ Q_{30}R_{30}\Pi_{c_{30}}^{-1} \end{pmatrix} \begin{matrix} \rightarrow \text{select k cols } I_{00} \\ \rightarrow \text{select k cols } I_{10} \\ \rightarrow \text{select k cols } I_{20} \\ \rightarrow \text{select k cols } I_{30} \end{matrix}$$



- Apply 1D-TP on sets of 2k sub-columns

$$\begin{pmatrix} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} (:, I_{00} \cup I_{10}) \\ \hline \begin{pmatrix} A_{31} \\ A_{41} \end{pmatrix} (:, I_{20} \cup I_{30}) \end{pmatrix} = \begin{pmatrix} Q_{01}R_{01}\Pi_{c_{01}}^{-1} \\ Q_{11}R_{11}\Pi_{c_{11}}^{-1} \end{pmatrix} \begin{matrix} \rightarrow I_{01} \\ \rightarrow I_{11} \end{matrix}$$

$$A(:, I_{01} \cup I_{11}) = \left(Q_{02}R_{02}\Pi_{c_{02}}^{-1}\right) \rightarrow I_{02}$$

- Return columns $A(:, I_{02})$

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

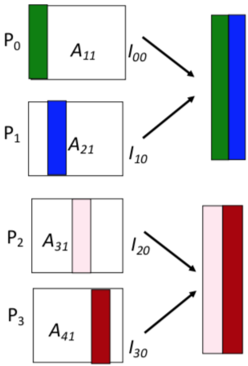
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} \Pi_{c_{00}}^{-1} \\ Q_{10} R_{10} \Pi_{c_{10}}^{-1} \\ Q_{20} R_{20} \Pi_{c_{20}}^{-1} \\ Q_{30} R_{30} \Pi_{c_{30}}^{-1} \end{pmatrix} \begin{array}{l} \rightarrow \text{select k cols } I_{00} \\ \rightarrow \text{select k cols } I_{10} \\ \rightarrow \text{select k cols } I_{20} \\ \rightarrow \text{select k cols } I_{30} \end{array}$$

- Apply 1D-TP on sets of 2k sub-columns

$$\begin{pmatrix} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} (:, I_{00} \cup I_{10}) \\ \hline \begin{pmatrix} A_{31} \\ A_{41} \end{pmatrix} (:, I_{20} \cup I_{30}) \end{pmatrix} = \begin{pmatrix} Q_{01} R_{01} \Pi_{c_{01}}^{-1} \\ Q_{11} R_{11} \Pi_{c_{11}}^{-1} \end{pmatrix} \begin{array}{l} \rightarrow I_{01} \\ \rightarrow I_{11} \end{array}$$

$$A(:, I_{01} \cup I_{11}) = \left(Q_{02} R_{02} \Pi_{c_{02}}^{-1} \right) \quad \rightarrow I_{02}$$

- Return columns $A(:, I_{02})$



Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

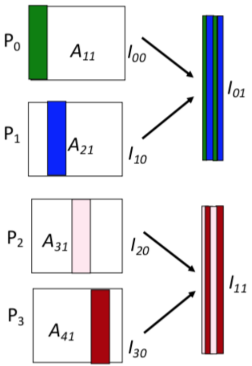
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} \Pi_{c_{00}}^{-1} \\ Q_{10} R_{10} \Pi_{c_{10}}^{-1} \\ Q_{20} R_{20} \Pi_{c_{20}}^{-1} \\ Q_{30} R_{30} \Pi_{c_{30}}^{-1} \end{pmatrix} \begin{matrix} \rightarrow \text{select k cols } I_{00} \\ \rightarrow \text{select k cols } I_{10} \\ \rightarrow \text{select k cols } I_{20} \\ \rightarrow \text{select k cols } I_{30} \end{matrix}$$

- Apply 1D-TP on sets of 2k sub-columns

$$\begin{pmatrix} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} (:, I_{00} \cup I_{10}) \\ \hline \begin{pmatrix} A_{31} \\ A_{41} \end{pmatrix} (:, I_{20} \cup I_{30}) \end{pmatrix} = \begin{pmatrix} Q_{01} R_{01} \Pi_{c_{01}}^{-1} \\ Q_{11} R_{11} \Pi_{c_{11}}^{-1} \end{pmatrix} \begin{matrix} \rightarrow I_{01} \\ \rightarrow I_{11} \end{matrix}$$

$$A(:, I_{01} \cup I_{11}) = \left(Q_{02} R_{02} \Pi_{c_{02}}^{-1} \right) \rightarrow I_{02}$$

- Return columns $A(:, I_{02})$



Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

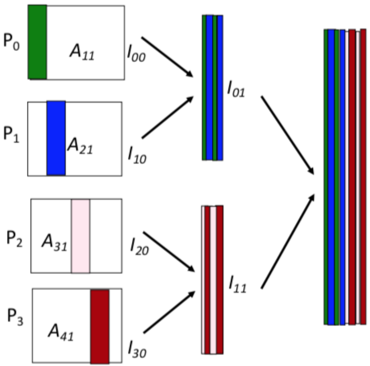
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} \Pi_{c_{00}}^{-1} \\ Q_{10} R_{10} \Pi_{c_{10}}^{-1} \\ Q_{20} R_{20} \Pi_{c_{20}}^{-1} \\ Q_{30} R_{30} \Pi_{c_{30}}^{-1} \end{pmatrix} \begin{matrix} \rightarrow \text{select k cols } I_{00} \\ \rightarrow \text{select k cols } I_{10} \\ \rightarrow \text{select k cols } I_{20} \\ \rightarrow \text{select k cols } I_{30} \end{matrix}$$

- Apply 1D-TP on sets of 2k sub-columns

$$\begin{pmatrix} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} (:, I_{00} \cup I_{10}) \\ \hline \begin{pmatrix} A_{31} \\ A_{41} \end{pmatrix} (:, I_{20} \cup I_{30}) \end{pmatrix} = \begin{pmatrix} Q_{01} R_{01} \Pi_{c_{01}}^{-1} \\ Q_{11} R_{11} \Pi_{c_{11}}^{-1} \end{pmatrix} \begin{matrix} \rightarrow I_{01} \\ \rightarrow I_{11} \end{matrix}$$

$$A(:, I_{01} \cup I_{11}) = \left(Q_{02} R_{02} \Pi_{c_{02}}^{-1} \right) \rightarrow I_{02}$$

- Return columns $A(:, I_{02})$



Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

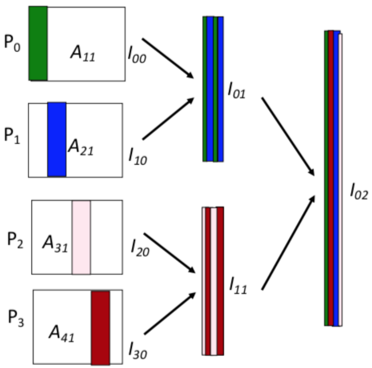
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} \Pi_{c_{00}}^{-1} \\ Q_{10} R_{10} \Pi_{c_{10}}^{-1} \\ Q_{20} R_{20} \Pi_{c_{20}}^{-1} \\ Q_{30} R_{30} \Pi_{c_{30}}^{-1} \end{pmatrix} \begin{array}{l} \rightarrow \text{select k cols } I_{00} \\ \rightarrow \text{select k cols } I_{10} \\ \rightarrow \text{select k cols } I_{20} \\ \rightarrow \text{select k cols } I_{30} \end{array}$$

- Apply 1D-TP on sets of 2k sub-columns

$$\begin{pmatrix} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} (:, I_{00} \cup I_{10}) \\ \hline \begin{pmatrix} A_{31} \\ A_{41} \end{pmatrix} (:, I_{20} \cup I_{30}) \end{pmatrix} = \begin{pmatrix} Q_{01} R_{01} \Pi_{c_{01}}^{-1} \\ Q_{11} R_{11} \Pi_{c_{11}}^{-1} \end{pmatrix} \begin{array}{l} \rightarrow I_{01} \\ \rightarrow I_{11} \end{array}$$

$$A(:, I_{01} \cup I_{11}) = \left(Q_{02} R_{02} \Pi_{c_{02}}^{-1} \right) \rightarrow I_{02}$$

- Return columns $A(:, I_{02})$



CA-RRQR : 2D tournament pivoting

- A distributed on $\Pi_r \times \Pi_c$ procs as e.g.

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \end{pmatrix}$$

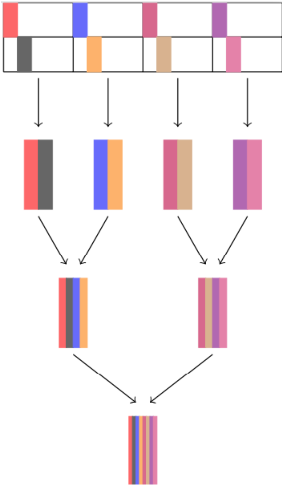
- Select k cols from each column block by 1Dr-TP,

$$\begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \quad \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} \quad \begin{pmatrix} A_{13} \\ A_{23} \end{pmatrix} \quad \begin{pmatrix} A_{14} \\ A_{24} \end{pmatrix}$$
$$\downarrow \quad\quad \downarrow \quad\quad \downarrow \quad\quad \downarrow$$
$$I_{00} \quad\quad I_{10} \quad\quad I_{20} \quad\quad I_{30}$$

- Apply 1Dc-TP on sets of k selected cols,

$$A(:, I_{00}) \quad A(:, I_{10}) \quad A(:, I_{20}) \quad A(:, I_{30})$$

- Return columns selected by 1Dc-TP $A(:, I_{02})$



CA-RRQR : 2D tournament pivoting

- A distributed on $\Pi_r \times \Pi_c$ procs as e.g.

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \end{pmatrix}$$

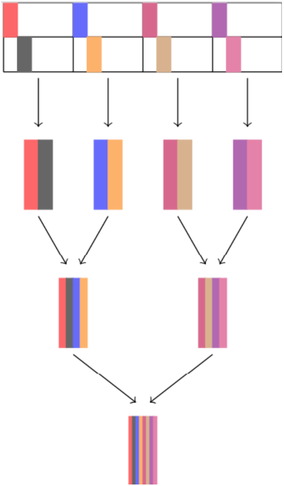
- Select k cols from each column block by 1Dr-TP,

$$\begin{matrix} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} & \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} & \begin{pmatrix} A_{13} \\ A_{23} \end{pmatrix} & \begin{pmatrix} A_{14} \\ A_{24} \end{pmatrix} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ I_{00} & I_{10} & I_{20} & I_{30} \end{matrix}$$

- Apply 1Dc-TP on sets of k selected cols,

$$A(:, I_{00}) \quad A(:, I_{10}) \quad A(:, I_{20}) \quad A(:, I_{30})$$

- Return columns selected by 1Dc-TP $A(:, I_{02})$



CA-RRQR : 2D tournament pivoting

- A distributed on $\Pi_r \times \Pi_c$ procs as e.g.

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \end{pmatrix}$$

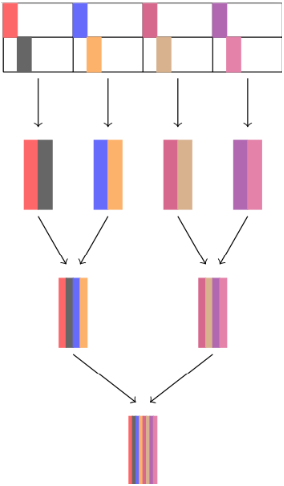
- Select k cols from each column block by 1Dr-TP,

$$\begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \quad \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} \quad \begin{pmatrix} A_{13} \\ A_{23} \end{pmatrix} \quad \begin{pmatrix} A_{14} \\ A_{24} \end{pmatrix}$$
$$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$$
$$I_{00} \qquad I_{10} \qquad I_{20} \qquad I_{30}$$

- Apply 1Dc-TP on sets of k selected cols,

$$A(:, I_{00}) \quad A(:, I_{10}) \quad A(:, I_{20}) \quad A(:, I_{30})$$

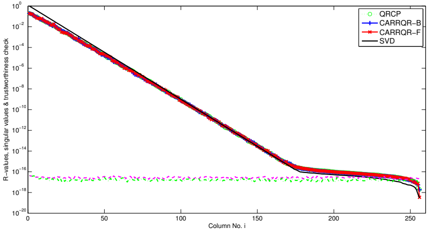
- Return columns selected by 1Dc-TP $A(:, I_{02})$

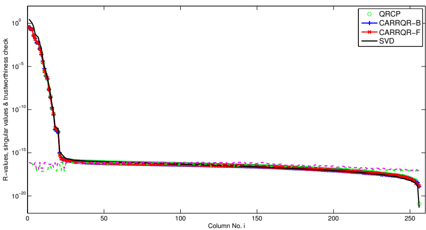


Numerical results

- Stability close to QRCP for many tested matrices.
- Absolute value of diagonals of R referred to as R-values.
- Methods compared
  - RRQR: QR with column pivoting
  - CA-RRQR-B with tournament pivoting 1Dc-TP based on binary tree
  - CA-RRQR-F with tournament pivoting 1Dc-TP based on flat tree
  - SVD

Numerical results (contd)



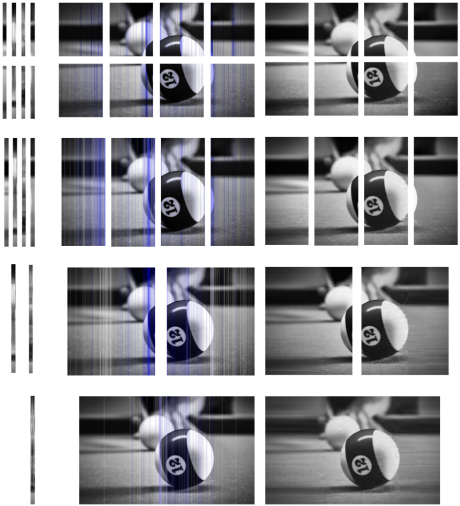


- Left: exponent - exponential Distribution, $\sigma_1 = 1$, $\sigma_i = \alpha^{i-1}$ $(i = 2, \ldots, n)$, $\alpha = 10^{-1/11}$ [Bischof, 1991]
- Right: shaw - 1D image restoration model [Hansen, 2007]

$$\epsilon \min\{||(A\Pi_0)(:,i)||_2\,, ||(A\Pi_1)(:,i)||_2\,, ||(A\Pi_2)(:,i)||_2\} \tag{7}$$
$$\epsilon \max\{||(A\Pi_0)(:,i)||_2\,, ||(A\Pi_1)(:,i)||_2\,, ||(A\Pi_2)(:,i)||_2\} \tag{8}$$

where $\Pi_j (j = 0, 1, 2)$ are the permutation matrices obtained by QRCP, CARRQR-B, and CARRQR-F, and ϵ is the machine precision.

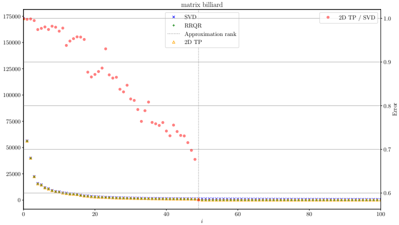
CA-RRQR : 2D tournament pivoting



Numerical experiments

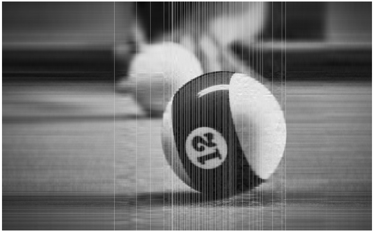
Original image, size 1190×1920

Singular values and ratios



Rank-10 approx, 2D TP 8×8 procs

Rank-50 approx, 2D TP 8×8 procs



- Image source: `https://pixabay.com/photos/billiards-ball-play-number-half-4345870/`

Plan

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Low rank approximation based on LU factorization

- Given desired rank k, the factorization has the form

$$\Pi_r A \Pi_c = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix}, \tag{9}$$

where $A \in \mathbb{R}^{m \times n}$, $\bar{A}_{11} \in \mathbb{R}^{k,k}$, $S(\bar{A}_{11}) = \bar{A}_{22} - \bar{A}_{21}\bar{A}_{11}^{-1}\bar{A}_{12}$.

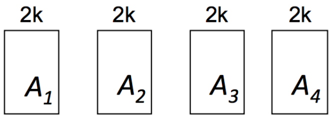
- The rank-k approximation matrix A_k is

$$A_k = \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{-1} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} = \begin{pmatrix} \bar{A}_{11} \\ \bar{A}_{21} \end{pmatrix} \bar{A}_{11}^{-1} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix}. \tag{10}$$

- $\bar{A}_{11}^{-1}$ is never formed, its factorization is used when A_k is applied to a vector.

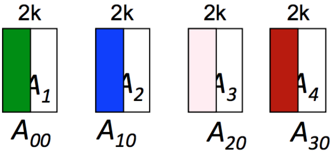
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



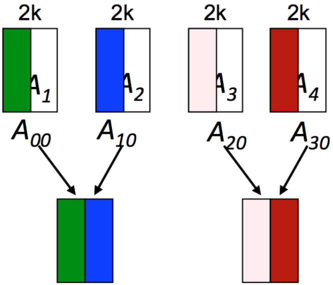
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



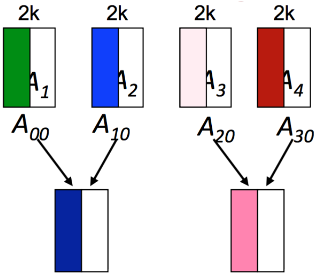
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



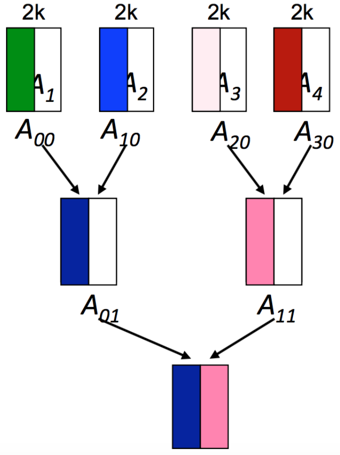
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



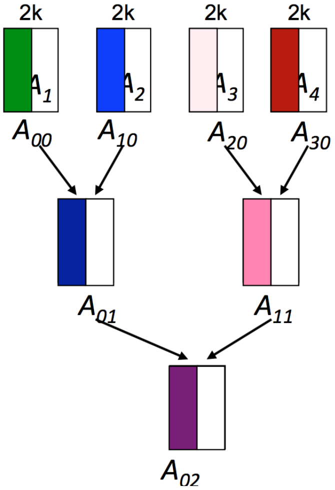
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



LU_CRTP factorization - one block step

One step of truncated block LU based on column/row tournament pivoting on matrix A of size $m \times n$:

1. Select k columns by using tournament pivoting, permute them in front, bounds for s.v. governed by $q_1(n, k)$

$$A\Pi_c = Q \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$$

2. Select k rows from $(Q_{11};\ Q_{21})^T$ of size $m \times k$ by using tournament pivoting,

$$\Pi_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix}$$

such that $||\bar{Q}_{21}\bar{Q}_{11}^{-1}||_{max} \leq F_{TP}$ and bounds for s.v. governed by $q_2(m, k)$.

Orthogonal matrices

Given orthogonal matrix $Q \in \mathbb{R}^{m \times m}$ and its partitioning

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}, \tag{11}$$

the selection of k cols by tournament pivoting from $(Q_{11}; Q_{21})^T$ leads to the factorization

$$\Pi_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \tag{12}$$

where $S(\bar{Q}_{11}) = \bar{Q}_{22} - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{Q}_{12} = \bar{Q}_{22}^{-T}$ since

$$\begin{aligned} S(\bar{Q}_{11})\bar{Q}_{22}^T &= \bar{Q}_{22}\bar{Q}_{22}^T - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{Q}_{12}\bar{Q}_{22}^T = I - \bar{Q}_{21}\bar{Q}_{21}^T - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{Q}_{12}\bar{Q}_{22}^T \\ &= I - \bar{Q}_{21}(\bar{Q}_{21}^T - \bar{Q}_{11}^{-1}\bar{Q}_{11}\bar{Q}_{21}^T) = I \end{aligned}$$

Orthogonal matrices (contd)

The factorization

$$\Pi_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \tag{13}$$

satisfies:

$$\begin{aligned} \|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|_{max} &\leq F_{TP}, & (14)\\ \frac{1}{q_2(m,k)} &\leq \sigma_i(\bar{Q}_{11}) \leq 1, & (15)\\ \sigma_{min}(\bar{Q}_{11}) &= \sigma_{min}(\bar{Q}_{22}) & (16) \end{aligned}$$

for all $1 \leq i \leq k$, $1 \leq j \leq m-k$, where $q_2(m,k) = \sqrt{1 + F_{TP}^2(m-k)}$.
Exercice: show that $\sigma_{min}(\bar{Q}_{11}) = \sigma_{min}(\bar{Q}_{22})$ by considering unit vectors $x \in \mathbb{R}^k, y \in \mathbb{R}^{m-k}$

$$1 = ||\bar{Q}_{11}x||^2 + ||\bar{Q}_{21}x||^2,\ 1 = ||\bar{Q}_{22}^T y||^2 + ||\bar{Q}_{21}^T y||^2$$

and showing $\min_{||x||=1} ||\bar{Q}_{11}x||^2 = \min_{||y||=1} ||\bar{Q}_{22}^T y||^2$

Sketch of the proof

$$
\begin{aligned}
\Pi_r A \Pi_c &= \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} \\
&= \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix} \qquad (17)
\end{aligned}
$$

where

$$
\begin{aligned}
\bar{Q}_{21}\bar{Q}_{11}^{-1} &= \bar{A}_{21}\bar{A}_{11}^{-1}, \\
S(\bar{A}_{11}) &= S(\bar{Q}_{11}) R_{22} = \bar{Q}_{22}^{-T} R_{22}.
\end{aligned}
$$

Sketch of the proof (contd)

$$\begin{aligned} \bar{A}_{11} &= \bar{Q}_{11} R_{11}, & (18) \\ S(\bar{A}_{11}) &= S(\bar{Q}_{11}) R_{22} = \bar{Q}_{22}^{-T} R_{22}. & (19) \end{aligned}$$

We obtain

$$\sigma_i(A) \geq \sigma_i(\bar{A}_{11}) \geq \sigma_{min}(\bar{Q}_{11}) \sigma_i(R_{11}) \geq \frac{1}{q_1(n,k) q_2(m,k)} \sigma_i(A),$$

We also have that

$$\begin{aligned} \sigma_{k+j}(A) \leq \sigma_j(S(\bar{A}_{11})) &= \sigma_j(S(\bar{Q}_{11}) R_{22}) \leq ||S(\bar{Q}_{11})||_2 \sigma_j(R_{22}) \\ &\leq q_1(n,k) q_2(m,k) \sigma_{k+j}(A), \end{aligned}$$

where $q_1(n,k) = \sqrt{1 + F_{TP}^2 (n-k)}$, $q_2(m,k) = \sqrt{1 + F_{TP}^2 (m-k)}$.

LU_CRTP factorization - bounds if $rank = k$

Given A of size $m \times n$, one step of LU_CRTP computes the decomposition

$$\bar{A} = \Pi_r A \Pi_c = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} \tag{20}$$

where $\bar{A}_{11}$ is of size $k \times k$ and

$$S(\bar{A}_{11}) = \bar{A}_{22} - \bar{A}_{21}\bar{A}_{11}^{-1}\bar{A}_{12} = \bar{A}_{22} - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{A}_{12}. \tag{21}$$

It satisfies the following properties:

$$\|\bar{A}_{21}\bar{A}_{11}^{-1}\|_{max} \quad = \quad \|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|_{max} \leq F_{TP}, \tag{22}$$

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(\bar{A}_{11})}, \frac{\sigma_j(S(\bar{A}_{11}))}{\sigma_{k+j}(A)} \leq q(m,n,k), \tag{23}$$

for any $1 \leq l \leq m-k$, $1 \leq i \leq k$, and $1 \leq j \leq \min(m,n) - k$,
$q(m,n,k) = q_1(n,k)q_2(m,k) = \sqrt{\left(1 + F_{TP}^2(n-k)\right)\left(1 + F_{TP}^2(m-k)\right)}$.

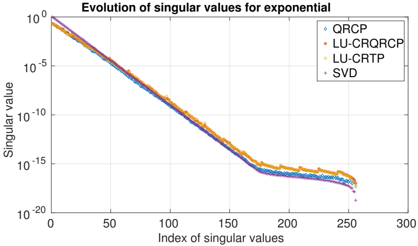
Plan

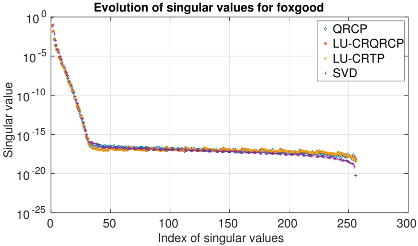
Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Numerical results

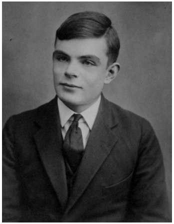




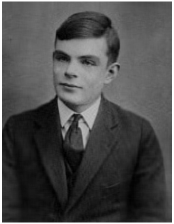
- Left: exponent - exponential Distribution, $\sigma_1 = 1$, $\sigma_i = \alpha^{i-1}$ $(i = 2, \ldots, n)$, $\alpha = 10^{-1/11}$ [Bischof, 1991]
- Right: foxgood - Severely ill-posed test problem of the 1st kind Fredholm integral equation used by Fox and Goodwin

Results for image of size 919 × 707

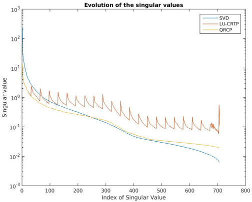
Original image



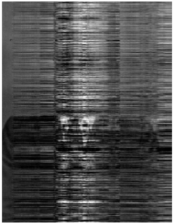
Rank-38 approx, SVD



Singular value distribution

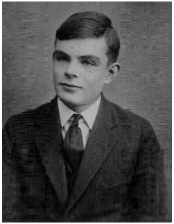


Rank-38 approx, LUPP



Rank-38 approx, LU_CRTP

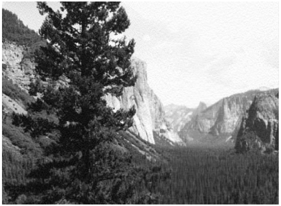
Rank-75 approx, LU_CRTP



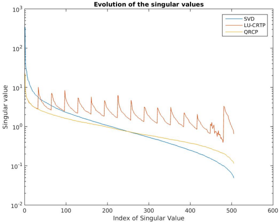
Results for image of size 691 × 505

Original image

Rank-105 approx, SVD

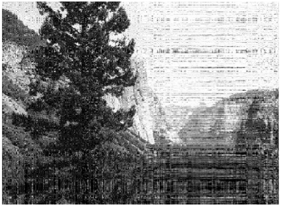


Singular value distribution

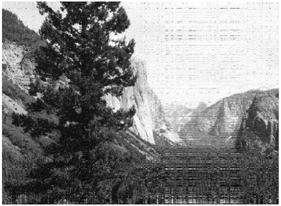


Rank-105 approx, LUPP

Rank-105 approx, LU_CRTP



Rank-209 approx, LU_CRTP



More details on CA deterministic algorithms

- [Demmel et al., 2015] Communication avoiding rank revealing QR factorization with column pivoting Demmel, Grigori, Gu, Xiang, SIAM J. Matrix Analysis and Applications, 2015.
- Low rank approximation of a sparse matrix based on LU factorization with column and row tournament pivoting, with S. Cayrols and J. Demmel, Inria TR 8910.

References (1)

Bischof, C. H. (1991).
A parallel QR factorization algorithm with controlled local pivoting.
SIAM J. Sci. Stat. Comput., 12:36–57.

Businger, P. A. and Golub, G. H. (1965).
Linear least squares solutions by Householder transformations.
Numer. Math., 7:269–276.

Civril, A. and Magdon-Ismail, M. (2013).
Exponential inapproximability of selecting a maximum volume sub-matrix.
Algorithmica, 65(1):159–176.

Demmel, J., Grigori, L., Gu, M., and Xiang, H. (2015).
Communication-avoiding rank-revealing qr decomposition.
SIAM Journal on Matrix Analysis and its Applications, 36(1):55–89.

Demmel, J., Grigori, L., and Rusciano, A. (2023).
An improved analysis and unified perspective on deterministic and randomized low-rank matrix approximation.
SIAM Journal on Matrix Analysis and Applications, 44(2):559–591.

Duersch, J. A. and Gu, M. (2017).
Randomized QR with column pivoting.
SIAM Journal on Scientific Computing, 39(4):C263–C291.

Eckart, C. and Young, G. (1936).
The approximation of one matrix by another of lower rank.
Psychometrika, 1:211–218.

References (2)

Eisenstat, S. C. and Ipsen, I. C. F. (1995).
Relative perturbation techniques for singular value problems.
SIAM J. Numer. Anal., 32(6):1972–1988.

Goreinov, S. and Tyrtshnikov, E. (2001).
The maximal-volume concept in approximation by low-rank matrices.
Contemporary Mathematics, 280:47–52.

Gu, M. and Eisenstat, S. C. (1996).
Efficient algorithms for computing a strong rank-revealing QR factorization.
SIAM J. Sci. Comput., 17(4):848–869.

Hansen, P. C. (2007).
Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems.
Numerical Algorithms, (46):189–194.

Martinsson, P.-G. and Voronin, S. (2016).
A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices.
SIAM Journal on Scientific Computing, 38(5):S485–S507.

Xiao, J., Gu, M., and Langou, J. (2017).
Fast Parallel Randomized QR with Column Pivoting Algorithms for Reliable Low-rank Matrix Approximations.
2017 IEEE 24th International Conference on High Performance Computing (HiPC), pages 233–242.
arXiv: 1804.05138.

Results used in the proofs

- Interlacing property of singular values [Golub, Van Loan, 4th edition, page 487]
  Let $A = [a_1| \dots |a_n]$ be a column partitioning of an $m \times n$ matrix with $m \geq n$. If $A_r = [a_1| \dots |a_r]$, then for $r = 1 : n - 1$

  $$\sigma_1(A_{r+1}) \geq \sigma_1(A_r) \geq \sigma_2(A_{r+1}) \geq \dots \geq \sigma_r(A_{r+1}) \geq \sigma_r(A_r) \geq \sigma_{r+1}(A_{r+1}).$$

- Given $n \times n$ matrix B and $n \times k$ matrix C, then ([Eisenstat and Ipsen, 1995], p. 1977)

  $$\sigma_{min}(B)\sigma_j(C) \leq \sigma_j(BC) \leq \sigma_{max}(B)\sigma_j(C), j = 1, \dots, k.$$