Communication avoiding low rank matrix approximation, a unified perspective on deterministic and randomized approaches

L. Grigori
and collaborators

Alpines
Inria Paris and LJLL, Sorbonne University

Slides available at https://who.rocq.inria.fr/Laura.Grigori/Slides/ENLA20_Grigori.pdf

July 8, 2020

Plan

Motivation of our work

Unified perspective on low rank matrix approximation
- Generalized LU decomposition

Recent deterministic algorithms and bounds
- CA RRQR with 2D tournament pivoting
- CA LU with column/row tournament pivoting

Randomized generalized LU and bounds

Approximation of tensors
- Parallel HORRQR

Conclusions

The communication challenge

- Cost of data movement dominates the cost of arithmetics: time and energy consumption
  - Per socket flop performance continues to increase: increase of number of cores per socket and/or number of flops per cycle

    2008 Intel Nehalem 3.2GHz$\times$4 cores (51.2 GFlops/socket DP)

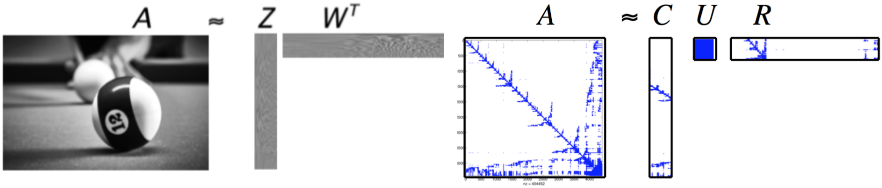
    2020 A64FX 2.2GHz$\times$48 cores (3.37 TFlops/socket DP)[1] 66x in 12 years
  - Interconnect latency: few μs MPI latency

Our focus: increasing scalability by reducing/minimizing coummunication while controlling the loss of information in low rank matrix (and tensor) approximation.

[1] Fugaku supercomputer `https://www.top500.org/system/179807/`

Low rank matrix approximation

- Problem: given $A \in \mathbb{R}^{m \times n}$, compute rank-k approximation ZW^T, where $Z \in \mathbb{R}^{m \times k}$ and $W^T \in \mathbb{R}^{k \times n}$.



- Problem ubiquitous in scientific computing and data analysis
  - column subset selection, linear dependency analysis, fast solvers for integral equations, H-matrices,
  - principal component analysis, image processing, data in high dimensions, ...

Low rank matrix approximation

- Best rank-k approximation $A_{opt,k} = \hat{U}_k \Sigma_k \hat{V}_k^T$ is rank-k truncated SVD of A [Eckart and Young, 1936], with $\sigma_{max}(A) = \sigma_1(A) \geq \ldots \geq \sigma_{min}(A) = \sigma_{\min(m,n)}(A)$

$$\min_{rank(\tilde{A}_k) \leq k} ||A - \tilde{A}_k||_2 = ||A - A_{opt,k}||_2 = \sigma_{k+1}(A)$$

$$\min_{rank(\tilde{A}_k) \leq k} ||A - \tilde{A}_k||_F = ||A - A_{opt,k}||_F = \sqrt{\sum_{j=k+1}^{n} \sigma_j^2(A)}$$

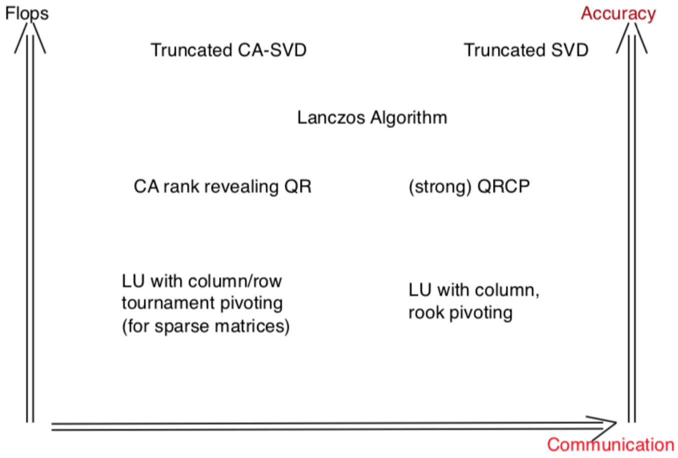
Image, size 1190×1920

Rank-10 approximation, SVD

Rank-50 approximation, SVD

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

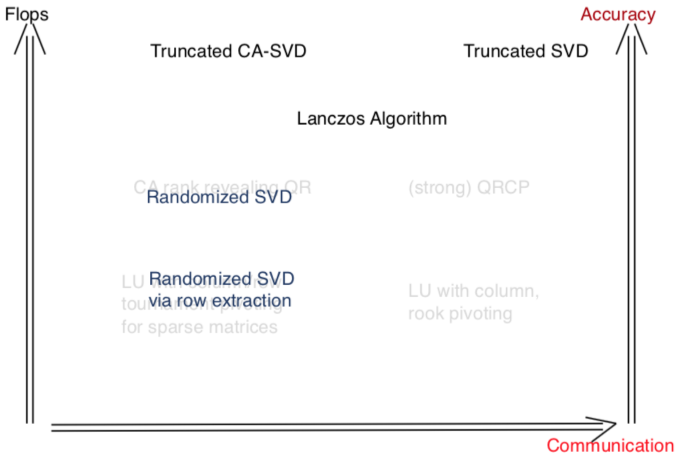
Low rank matrix approximation: trade-offs



Communication optimal if computing a rank-k approximation on P processors requires

$$\# \text{ messages} = \Omega\left(\log_2 P\right).$$

Low rank matrix approximation: trade-offs



Communication optimal if computing a rank-k approximation on P processors requires

$$\# \text{ messages} = \Omega\left(\log_2 P\right).$$

Idea underlying many algorithms

Compute $\tilde{A}_k = \mathcal{P}A$, where $\mathcal{P} = \mathcal{P}^o$ or $\mathcal{P} = \mathcal{P}^{so}$ is obtained as:

1. Construct a low dimensional subspace $X = range(AV_1)$, $V_1 \in \mathbb{R}^{n \times l}$ that approximates well the range of A, e.g.

$$\|A - \mathcal{P}^o A\|_2 \leq \gamma \sigma_{k+1}(A), \text{ for some } \gamma \geq 1,$$

where Q_1 is orth. basis of (AV_1)

$$\mathcal{P}^o = AV_1(AV_1)^+ = Q_1 Q_1^T, \text{ or equiv } \mathcal{P}^o a_j := arg \min_{x \in X} \|x - a_j\|_2$$

2. Select a semi-inner product $\langle U_1 \cdot, U_1 \cdot \rangle_2$, $U_1 \in \mathbb{R}^{l' \times m}$ $l' \geq l$, define

$$\mathcal{P}^{so} = AV_1(U_1 AV_1)^+ U_1, \text{ or equiv } \mathcal{P}^{so} a_j := arg \min_{x \in X} \|U_1(x - a_j)\|_2$$

with O. Balabanov, 2020

Idea underlying many algorithms

Compute $\tilde{A}_k = \mathcal{P}A$, where $\mathcal{P} = \mathcal{P}^o$ or $\mathcal{P} = \mathcal{P}^{so}$ is obtained as:

1. Construct a low dimensional subspace $X = range(AV_1)$, $V_1 \in \mathbb{R}^{n \times l}$ that approximates well the range of A, e.g.

$$\|A - \mathcal{P}^o A\|_2 \leq \gamma \sigma_{k+1}(A), \text{ for some } \gamma \geq 1,$$

   where Q_1 is orth. basis of (AV_1)

$$\mathcal{P}^o = AV_1(AV_1)^+ = Q_1 Q_1^T, \text{ or equiv } \mathcal{P}^o a_j := arg \min_{x \in X} \|x - a_j\|_2$$

2. Select a semi-inner product $\langle U_1 \cdot, U_1 \cdot \rangle_2$, $U_1 \in \mathbb{R}^{l' \times m}$ $l' \geq l$, define

$$\mathcal{P}^{so} = AV_1(U_1 AV_1)^+ U_1, \text{ or equiv } \mathcal{P}^{so} a_j := arg \min_{x \in X} \|U_1(x - a_j)\|_2$$

with O. Balabanov, 2020

Unified perspective: generalized LU factorization

Given $A \in \mathbb{R}^{m \times n}$, $U = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \in \mathbb{R}^{m,m}$, $V = \begin{pmatrix} V_1 & V_2 \end{pmatrix} \in \mathbb{R}^{n,n}$, U, V invertible, $U_1 \in \mathbb{R}^{l' \times m}$, $V_1 \in \mathbb{R}^{n \times l}$, $k \le l \le l'$.

$$\begin{aligned} UAV &= \bar{A} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} \\ &= \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{+} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} \end{aligned}$$

where $\bar{A}_{11} \in \mathbb{R}^{l',l}$, $\bar{A}_{11}^{+}\bar{A}_{11} = I$, $S(\bar{A}_{11}) = \bar{A}_{22} - \bar{A}_{21}\bar{A}_{11}^{+}\bar{A}_{12}$.

- Generalized LU computes the approximation

$$\begin{aligned} \tilde{A}_{glu} &= U^{-1} \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{+} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} V^{-1} \\ &= [U_1^{+}(I - (U_1AV_1)(U_1AV_1)^{+}) + (AV_1)(U_1AV_1)^{+}][U_1A] \end{aligned}$$

with J. Demmel and A. Rusciano, 2019

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \le l < l'$, rank-k approximation,

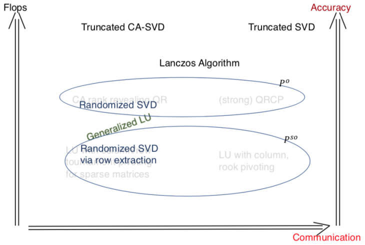
$$\tilde{A}_{glu} \;=\; [U_1^+(I - (U_1AV_1)(U_1AV_1)^+) + (AV_1)(U_1AV_1)^+][U_1A]$$

Unification for many existing algorithms:

- If $k \le l = l'$ and $U_1 = Q_1^T$, then $\tilde{A}_{glu} = Q_1 Q_1^T A = \mathcal{P}^o A$
- If $k \le l = l'$ then $\tilde{A}_{glu} = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A$

Approximation result: If $k \le l < l'$,

$$\|A - \mathcal{P}^{so}A\|_F^2 = \|A - \tilde{A}_{glu}\|_F^2 + \|\tilde{A}_{glu} - \mathcal{P}^{so}A\|_F^2$$



Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l < l'$, rank-k approximation,

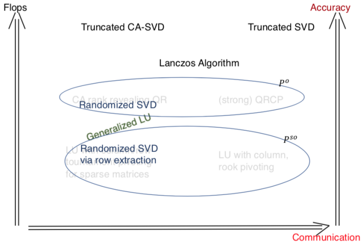
$$\tilde{A}_{glu} \quad = \quad [U_1^{+}(I - (U_1 A V_1)(U_1 A V_1)^{+}) + (AV_1)(U_1 A V_1)^{+}][U_1 A]$$

Unification for many existing algorithms:

- If $k \leq l = l'$ and $U_1 = Q_1^T$, then $\tilde{A}_{glu} = Q_1 Q_1^T A = \mathcal{P}^o A$
- If $k \leq l = l'$ then $\tilde{A}_{glu} = AV_1(U_1 A V_1)^{-1} U_1 A = \mathcal{P}^{so} A$

Approximation result: If $k \leq l < l'$,

$$\|A - \mathcal{P}^{so} A\|_F^2 = \|A - \tilde{A}_{glu}\|_F^2 + \|\tilde{A}_{glu} - \mathcal{P}^{so} A\|_F^2$$



Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l < l'$, rank-k approximation,

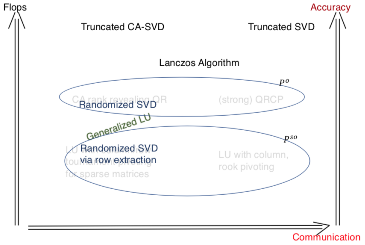
$$\tilde{A}_{glu} \quad = \quad [U_1^+(I - (U_1AV_1)(U_1AV_1)^+) + (AV_1)(U_1AV_1)^+][U_1A]$$

Unification for many existing algorithms:

- If $k \leq l = l'$ and $U_1 = Q_1^T$, then $\tilde{A}_{glu} = Q_1 Q_1^T A = \mathcal{P}^o A$
- If $k \leq l = l'$ then $\tilde{A}_{glu} = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A$

Approximation result: If $k \leq l < l'$,

$$\|A - \mathcal{P}^{so}A\|_F^2 = \|A - \tilde{A}_{glu}\|_F^2 + \|\tilde{A}_{glu} - \mathcal{P}^{so}A\|_F^2$$



Desired properties of low rank matrix approximation

1. $\tilde{A}_k$ is (k, γ) *low-rank approximation* of A if it satisfies

$$\|A - \tilde{A}_k\|_2 \leq \gamma \sigma_{k+1}(A), \text{ for some } \gamma \geq 1.$$

$\rightarrow$ Focus of both deterministic and randomized approaches

2. $\tilde{A}_k$ is (k, γ) *spectrum preserving* of A if

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(\tilde{A}_k)} \leq \gamma, \text{ for all } i = 1, \ldots, k \text{ and some } \gamma \geq 1$$

$\rightarrow$ Focus of deterministic approaches

3. $\tilde{A}_k$ is (k, γ) *kernel approximation* of A if

$$1 \leq \frac{\sigma_j(A - \tilde{A}_k)}{\sigma_{k+j}(A)} \leq \gamma, \text{ for all } i = 1, \ldots, \min(m, n) - k \text{ and some } \gamma \geq 1$$

$\rightarrow$ Focus of deterministic approaches

Goal γ is a low degree polynomial in k and the number of columns and/or rows of A for some of the most efficient algorithms.

Desired properties of low rank matrix approximation

1. $\tilde{A}_k$ is (k, γ) *low-rank approximation* of A if it satisfies

$$\|A - \tilde{A}_k\|_2 \leq \gamma \sigma_{k+1}(A), \text{ for some } \gamma \geq 1.$$

   $\rightarrow$ Focus of both deterministic and randomized approaches

2. $\tilde{A}_k$ is (k, γ) *spectrum preserving* of A if

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(\tilde{A}_k)} \leq \gamma, \text{ for all } i = 1, \ldots, k \text{ and some } \gamma \geq 1$$

   $\rightarrow$ Focus of deterministic approaches

3. $\tilde{A}_k$ is (k, γ) *kernel approximation* of A if

$$1 \leq \frac{\sigma_j(A - \tilde{A}_k)}{\sigma_{k+j}(A)} \leq \gamma, \text{ for all } i = 1, \ldots, \min(m, n) - k \text{ and some } \gamma \geq 1$$

   $\rightarrow$ Focus of deterministic approaches

Goal γ is a low degree polynomial in k and the number of columns and/or rows of A for some of the most efficient algorithms.

Desired properties of low rank matrix approximation

1. $\tilde{A}_k$ is (k, γ) *low-rank approximation* of A if it satisfies

$$\|A - \tilde{A}_k\|_2 \leq \gamma \sigma_{k+1}(A), \text{ for some } \gamma \geq 1.$$

   $\rightarrow$ Focus of both deterministic and randomized approaches

2. $\tilde{A}_k$ is (k, γ) *spectrum preserving* of A if

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(\tilde{A}_k)} \leq \gamma, \text{ for all } i = 1, \dots, k \text{ and some } \gamma \geq 1$$

   $\rightarrow$ Focus of deterministic approaches

3. $\tilde{A}_k$ is (k, γ) *kernel approximation* of A if

$$1 \leq \frac{\sigma_j(A - \tilde{A}_k)}{\sigma_{k+j}(A)} \leq \gamma, \text{ for all } i = 1, \dots, \min(m, n) - k \text{ and some } \gamma \geq 1$$

   $\rightarrow$ Focus of deterministic approaches

Goal γ is a low degree polynomial in k and the number of columns and/or rows of A for some of the most efficient algorithms.

Plan

Strong rank revealing QR (RRQR) factorization

Given $A \in \mathbb{R}^{m \times n}$, consider the QRCP decomposition with $R_{11} \in \mathbb{R}^{k \times k}$, [Golub, 1965, Businger and Golub, 1965],

$$\begin{aligned} AV &= QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}, \\ \tilde{A}_{qr} &= Q_1 \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} V^{-1} = Q_1 Q_1^T A = \mathcal{P}^o A \end{aligned}$$

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

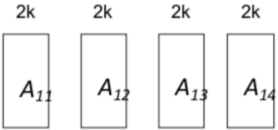
$$\begin{aligned} 1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} &\leq \gamma(n,k), \quad \gamma(n,k) = \sqrt{1 + f^2 k(n-k)} \\ ||R_{11}^{-1} R_{12}||_{max} &\leq f \end{aligned}$$

  for $1 \leq i \leq k$ and $1 \leq j \leq \min(m,n) - k$, and $\sigma_j(R_{22}) = \sigma_j(A - \tilde{A}_{qr})$
- Cost: $4mnk$ (QRCP) plus $O(mnk)$ flops and $O(k \log_2 P)$ messages.

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

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

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

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

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

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

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

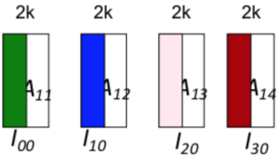
[Demmel, LG, Gu, Xiang'15]

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

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



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

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

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

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

[Demmel, LG, Gu, Xiang'15]

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

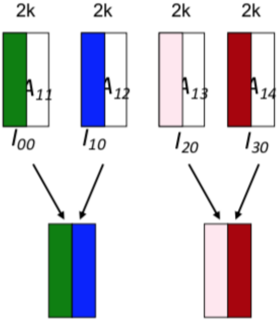
$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ \| & \| & \| & \| \\ Q_{00}R_{00}V_{00}^T & Q_{10}R_{10}V_{10}^T & Q_{20}R_{20}V_{20}^T & Q_{30}R_{30}V_{30}^T \\ \downarrow & \downarrow & \downarrow & \downarrow \\ I_{00} & I_{10} & I_{20} & I_{30} \end{pmatrix}$$

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

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

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

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



[Demmel, LG, Gu, Xiang'15]

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

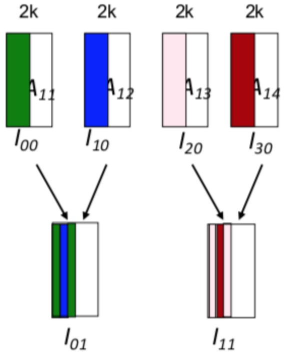
$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ \| & \| & \| & \| \\ Q_{00}R_{00}V_{00}^T & Q_{10}R_{10}V_{10}^T & Q_{20}R_{20}V_{20}^T & Q_{30}R_{30}V_{30}^T \\ \downarrow & \downarrow & \downarrow & \downarrow \\ I_{00} & I_{10} & I_{20} & I_{30} \end{pmatrix}$$

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

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

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

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



[Demmel, LG, Gu, Xiang'15]

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

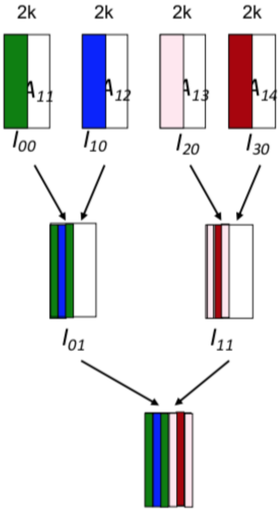
$$\begin{array}{cccc} (\; A_{11} & A_{12} & A_{13} & A_{14} \;) \\ \| & \| & \| & \| \\ (\; Q_{00}R_{00}V_{00}^T & Q_{10}R_{10}V_{10}^T & Q_{20}R_{20}V_{20}^T & Q_{30}R_{30}V_{30}^T \;) \\ \downarrow & \downarrow & \downarrow & \downarrow \\ I_{00} & I_{10} & I_{20} & I_{30} \end{array}$$

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

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

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

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



[Demmel, LG, Gu, Xiang'15]

Deterministic column selection: tournament pivoting

1D tournament pivoting (1Dc-TP)

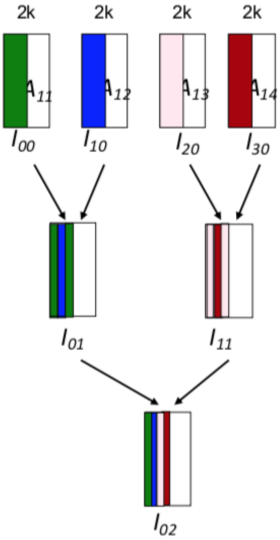
- 1D column block partition of A, select k cols from each block with strong RRQR

$$
\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \end{pmatrix}
$$
$$
\| \quad \| \quad \| \quad \|
$$
$$
\begin{pmatrix} Q_{00}R_{00}V_{00}^T & Q_{10}R_{10}V_{10}^T & Q_{20}R_{20}V_{20}^T & Q_{30}R_{30}V_{30}^T \end{pmatrix}
$$
$$
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow
$$
$$
I_{00} \quad I_{10} \quad I_{20} \quad I_{30}
$$

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

$$
\begin{pmatrix} A(:, I_{00} \cup I_{10}) & A(:, I_{20} \cup I_{30}); \end{pmatrix}
$$
$$
\| \quad \|
$$
$$
\begin{pmatrix} Q_{01}R_{01}V_{01}^T & Q_{11}R_{11}V_{11}^T \end{pmatrix}
$$
$$
\downarrow \quad \downarrow
$$
$$
I_{01} \quad I_{11}
$$

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



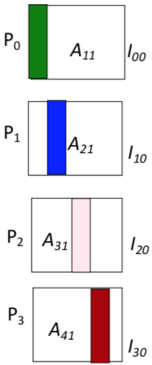
- Return selected columns $A(:, I_{02})$

[Demmel, LG, Gu, Xiang'15]

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

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



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

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

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

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

with M. Beaupère, Inria

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

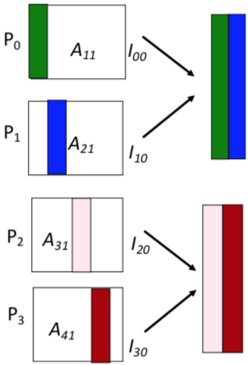
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} V_{00}^{-1} \\ Q_{10} R_{10} V_{10}^{-1} \\ Q_{20} R_{20} V_{20}^{-1} \\ Q_{30} R_{30} V_{30}^{-1} \end{pmatrix} \begin{matrix} \to \text{select k cols } I_{00} \\ \to \text{select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

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

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

$$A(:, I_{01} \cup I_{11}) = \begin{pmatrix} Q_{02} R_{02} V_{02}^{-1} \end{pmatrix} \quad \to I_{02}$$

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



with M. Beaupère, Inria

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

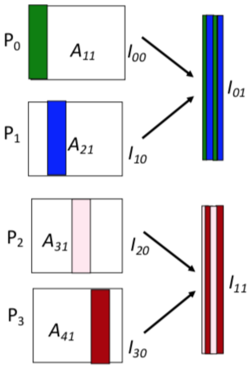
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} V_{00}^{-1} \\ Q_{10} R_{10} V_{10}^{-1} \\ Q_{20} R_{20} V_{20}^{-1} \\ Q_{30} R_{30} V_{30}^{-1} \end{pmatrix} \begin{matrix} \rightarrow \text{select k cols } I_{00} \\ \rightarrow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

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

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

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

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



with M. Beaupère, Inria

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

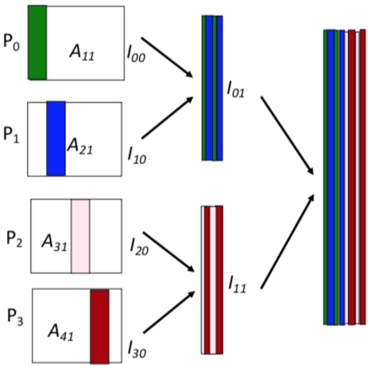
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} V_{00}^{-1} \\ Q_{10} R_{10} V_{10}^{-1} \\ Q_{20} R_{20} V_{20}^{-1} \\ Q_{30} R_{30} V_{30}^{-1} \end{pmatrix} \begin{matrix} \rightarrow \text{select k cols } I_{00} \\ \rightarrow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

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

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

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

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



with M. Beaupère, Inria

Tournament pivoting for 1D row partitioning - 1Dr TP

- Row block partition A as e.g.

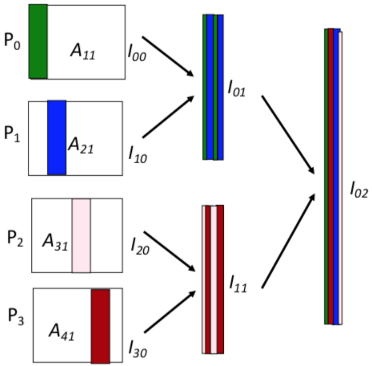
$$A = \begin{pmatrix} A_{11} \\ \hline A_{21} \\ \hline A_{31} \\ \hline A_{41} \end{pmatrix} = \begin{pmatrix} Q_{00} R_{00} V_{00}^{-1} \\ Q_{10} R_{10} V_{10}^{-1} \\ Q_{20} R_{20} V_{20}^{-1} \\ Q_{30} R_{30} V_{30}^{-1} \end{pmatrix} \begin{matrix} \rightarrow \text{select k cols } I_{00} \\ \rightarrow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

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

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

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

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

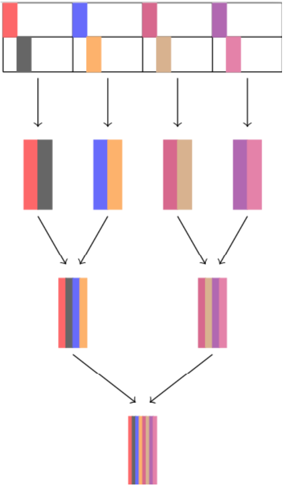


with M. Beaupère, Inria

CA-RRQR : 2D tournament pivoting

- A distributed on $P_r \times P_c$ 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,

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

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

with M. Beaupère, Inria

CA-RRQR : 2D tournament pivoting

- A distributed on $P_r \times P_c$ procs as e.g.

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

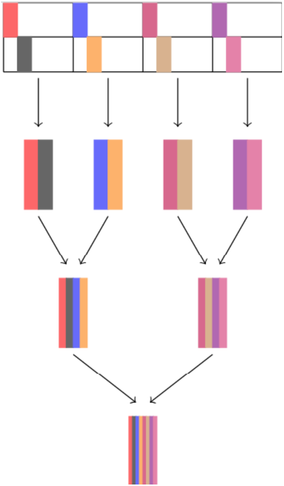
- Select k cols from each column block by 1Dr-TP,

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

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



with M. Beaupère, Inria

CA-RRQR : 2D tournament pivoting

- A distributed on $P_r \times P_c$ procs as e.g.

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

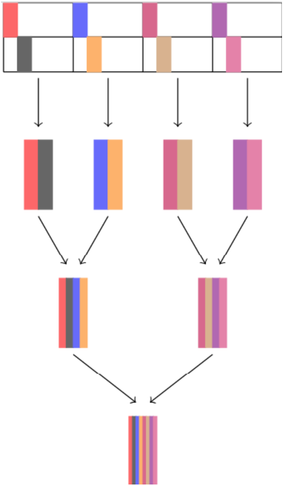
- Select k cols from each column block by 1Dr-TP,

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

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



with M. Beaupère, Inria

CA-RRQR - bounds for 2D tournament pivoting

Bounds when selecting k columns from $A \in \mathbb{R}^{m \times n}$ distributed on $P = P_r \times P_c$ processors by using 2D tournament pivoting:

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \gamma_1(n,k), \gamma_1(n,k) = \sqrt{1 + F_{2D-TP}^2(n-k)},$$

$$||(R_{11}^{-1} R_{12})(:,l)||_2 \leq F_{2D-TP}$$

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

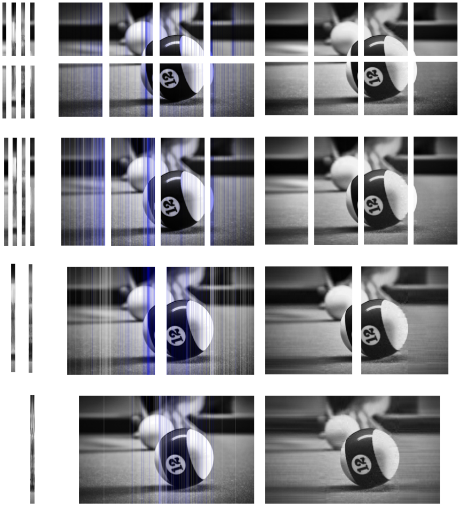
- 1Dr-TP with binary tree of depth $\log_2 P_r$ followed by 1Dc-TP with binary tree of depth $\log_2 P_c$,

$$F_{2D-TP} \leq P k^{\log_2 P + 1/2} f^{\log_2 P_c + 1}$$

- Cost: $O(\frac{mnk}{P})$flops, $(1 + log_2 P_r) log_2 P$ messages , $O(\frac{mk}{P_r} \log_2 P_c)$ words

$\rightarrow$ $\tilde{A}_{qr}$ with 2D TP is $(k, \gamma_1(n,k))$ spectrum preserving and kernel approximation of A

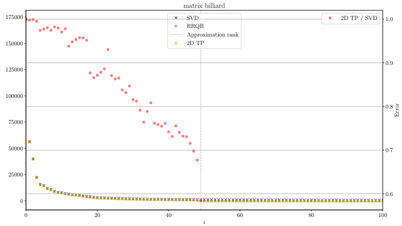
CA-RRQR : 2D tournament pivoting



Numerical experiments

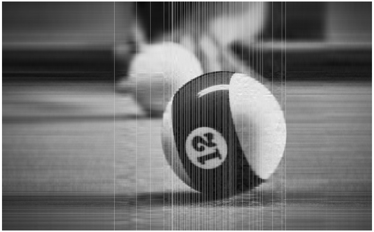
Original image, size 1190×1920

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/`

LU_CRTP: LU with column/row tournament pivoting

Compute rank-k approx. $\tilde{A}_{lu}$ of $A \in \mathbb{R}^{m \times n}$, $k = l = l'$,

$$\tilde{A}_{lu} = \begin{pmatrix} \bar{A}_{11} \\ \bar{A}_{21} \end{pmatrix} \bar{A}_{11}^{-1} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A \qquad (1)$$

1. Select k columns by using TP, bounds for s.v. governed by $\gamma_1(n,k)$

$$AV = Q \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$$

2. Select k rows from $Q_1 \in \mathbb{R}^{m \times k}$ by using TP,

$$U_1 Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}, \text{ hence } \bar{A}_{11} = \bar{Q}_{11} R_{11},$$

s.t. $||\bar{Q}_{21}\bar{Q}_{11}^{-1}||_{max}$ is bounded and bounds for s.v. governed by $\gamma_2(m,k)$,

$$\frac{1}{\gamma_2(m,k)} \leq \sigma_i(\bar{Q}_{11}) \leq 1.$$

with S. Cayrols, J. Demmel, 2018

Deterministic guarantees for rank-k approximation

- CA LU_CRTP with column/row selection with binary tree tournament pivoting:

$$
\begin{aligned}
1 \leq \frac{\sigma_i(A)}{\sigma_i(\bar{A}_{11})}, \frac{\sigma_j(S(\bar{A}_{11}))}{\sigma_{k+j}(A)} &\leq \sqrt{(1 + F_{TP}^2(n-k))}/\sigma_{min}(\bar{Q}_{11}) \\
&\leq \sqrt{(1 + F_{TP}^2(n-k))\,(1 + F_{TP}^2(m-k))} \\
&= \gamma_1(n,k)\gamma_2(m,k),
\end{aligned}
$$

for any $1 \leq i \leq k$, and $1 \leq j \leq \min(m,n) - k$, $U_1 Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$, and $\sigma_j(A - \tilde{A}_{lu}) = \sigma_j(S(\bar{A}_{11}))$.

$\rightarrow$ $\tilde{A}_{lu}$ is $(k, \gamma_1(n,k)\gamma_2(m,k))$ spectrum preserving and kernel approximation of A

Performance results

Selection of 256 columns by tournament pivoting

- Edison, Cray XC30 (NERSC): 2x12-core Intel Ivy Bridge (2.4 GHz)
- Tournament pivoting uses SPQR (T. Davis) + dGEQP3 (Lapack), time in secs

Matrices: dimension at leaves on 32 procs

- Parab_fem: 528825×528825 — 528825×16432
- Mac_econ: 206500×206500 — 206500×6453

	Time 2k cols	*Time leaves 32procs SPQR + dGEQP3*	*Number of MPI processes* 16	32	64	128	256	512	1024
Parab_fem	0.26	0.26 + 1129	46.7	24.5	13.7	8.4	5.9	4.8	4.4
Mac_econ	0.46	25.4 + 510	132.7	86.3	111.4	59.6	27.2	–	–

Plan

Typical randomized SVD

1. Compute an approximate basis for the range of $A \in \mathbb{R}^{m \times n}$
   Sample $V_1 \in \mathbb{R}^{n \times l}$, $l = p + k$, with independent mean-zero, unit-variance Gaussian entries.
   Compute $Y = AV_1$, $Y \in \mathbb{R}^{m \times l}$ expected to span column space of A.
   - Cost of multiplying AV_1: $2mnl$ flops
2. With Q_1 being orthonormal basis of Y, approximate A as:

$$\tilde{A}_k = Q_1 Q_1^T A = \mathcal{P}^o A$$

   - Cost of multiplying $Q_1^T A$: $2mnl$ flops

Source: Halko et al, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decomposition*, SIREV 2011.

Cost of randomized SVD for dense matrices

→ To have lower arithmetic complexity than deterministic approaches, the costs of multiplying AV_1 and $Q_1^T A$ need to be reduced:

1. Take V_1 a fast Johnson-Lindenstrauss transform, e.g. a subsampled randomized Hadamard transform (SRHT), AV_1 costs $2mn\log_2(l+1)$
   References: Ailon and Chazelle'06, Liberty, Rokhlin, Tygert and Woolfe'06, Sarlos'06.

2. Use a different projector than $\mathcal{P}^o$, e.g. pick U_1 and compute

$$\tilde{A}_k = \mathcal{P}^{so} A = AV_1(U_1 AV_1)^+ U_1 A$$

   Examples: randomized SVD via row extraction, Clarkson and Woodruff approximation in input sparsity time.

Cost of randomized SVD for dense matrices

$\rightarrow$ To have lower arithmetic complexity than deterministic approaches, the costs of multiplying AV_1 and $Q_1^T A$ need to be reduced:

1. Take V_1 a fast Johnson-Lindenstrauss transform, e.g. a subsampled randomized Hadamard transform (SRHT), AV_1 costs $2mn \log_2(l+1)$
   References: Ailon and Chazelle'06, Liberty, Rokhlin, Tygert and Woolfe'06, Sarlos'06.

2. Use a different projector than $\mathcal{P}^o$, e.g. pick U_1 and compute

$$\tilde{A}_k = \mathcal{P}^{so} A = AV_1(U_1 AV_1)^+ U_1 A$$

   Examples: randomized SVD via row extraction, Clarkson and Woodruff approximation in input sparsity time.

Cost of randomized SVD for dense matrices

$\rightarrow$ To have lower arithmetic complexity than deterministic approaches, the costs of multiplying AV_1 and $Q_1^T A$ need to be reduced:

1. Take V_1 a fast Johnson-Lindenstrauss transform, e.g. a subsampled randomized Hadamard transform (SRHT), AV_1 costs $2mn \log_2(l+1)$
   References: Ailon and Chazelle'06, Liberty, Rokhlin, Tygert and Woolfe'06, Sarlos'06.

2. Use a different projector than $\mathcal{P}^o$, e.g. pick U_1 and compute

$$\tilde{A}_k = \mathcal{P}^{so} A = AV_1 (U_1 AV_1)^+ U_1 A$$

   Examples: randomized SVD via row extraction, Clarkson and Woodruff approximation in input sparsity time.

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l = l'$, rank-k approximation,

$$\tilde{A}_k = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A$$

Deterministic algorithms V_1 column permutation and ...	**Randomized algorithms*** V_1 random matrix and ...
QR with column selection (a.k.a. strong rank revealing QR) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$ $\|\|R_{11}^{-1}R_{12}\|\|_{max}$ is bounded	Randomized QR (a.k.a. randomized SVD) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$
LU with column/row selection (a.k.a. rank revealing LU) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ is bounded	Randomized LU with row selection (a.k.a. randomized SVD via Row extraction) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ bounded
	Randomized LU approximation U_1 random matrix

with J. Demmel, A. Rusciano

* For a review, see Halko et al., SIAM Review 11

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l = l'$, rank-k approximation,

$$\tilde{A}_k = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A$$

Deterministic algorithms V_1 column permutation and ...	**Randomized algorithms*** V_1 random matrix and ...
QR with column selection (a.k.a. strong rank revealing QR) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$ $\|\|R_{11}^{-1}R_{12}\|\|_{max}$ is bounded	Randomized QR (a.k.a. randomized SVD) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$
LU with column/row selection (a.k.a. rank revealing LU) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ is bounded	Randomized LU with row selection (a.k.a. randomized SVD via Row extraction) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ bounded
	Randomized LU approximation U_1 random matrix

with J. Demmel, A. Rusciano

* For a review, see Halko et al., SIAM Review 11

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l = l'$, rank-k approximation,

$$\tilde{A}_k = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A$$

Deterministic algorithms V_1 column permutation and ...	**Randomized algorithms*** V_1 random matrix and ...
QR with column selection (a.k.a. strong rank revealing QR) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$ $\|\|R_{11}^{-1}R_{12}\|\|_{max}$ is bounded	Randomized QR (a.k.a. randomized SVD) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$
LU with column/row selection (a.k.a. rank revealing LU) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix}\bar{Q}_{11}\\ \bar{Q}_{21}\end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ is bounded	Randomized LU with row selection (a.k.a. randomized SVD via Row extraction) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix}\bar{Q}_{11}\\ \bar{Q}_{21}\end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ bounded
	Randomized LU approximation U_1 random matrix

with J. Demmel, A. Rusciano

* For a review, see Halko et al., SIAM Review 11

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l = l'$, rank-k approximation,

$$\tilde{A}_k = AV_1(U_1AV_1)^{-1}U_1A = \mathcal{P}^{so}A$$

Deterministic algorithms V_1 column permutation and ...	**Randomized algorithms*** V_1 random matrix and ...
QR with column selection (a.k.a. strong rank revealing QR) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$ $\|\|R_{11}^{-1}R_{12}\|\|_{max}$ is bounded	Randomized QR (a.k.a. randomized SVD) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1Q_1^TA = \mathcal{P}^oA$
LU with column/row selection (a.k.a. rank revealing LU) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ is bounded	Randomized LU with row selection (a.k.a. randomized SVD via Row extraction) U_1 row permutation s.t. $U_1Q_1 = \begin{pmatrix} \bar{Q}_{11} \\ \bar{Q}_{21} \end{pmatrix}$ $\|\|\bar{Q}_{21}\bar{Q}_{11}^{-1}\|\|_{max}$ bounded
	Randomized LU approximation U_1 random matrix

with J. Demmel, A. Rusciano

* For a review, see Halko et al., SIAM Review 11

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \leq l < l'$, rank-k approximation,

$$\begin{aligned}\tilde{A}_{glu} &= U^{-1} \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{+} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} V^{-1} \\ &= [U_1^{+}(I - (U_1AV_1)(U_1AV_1)^{+}) + (AV_1)(U_1AV_1)^{+}][U_1A] \neq \mathcal{P}^{so}A\end{aligned}$$

Approximation result: When $k \leq l < l'$, the approximation $\tilde{A}_{glu}$ is more accurate than $\mathcal{P}^{so}A$,

$$\|A - \mathcal{P}^{so}A\|_F^T = \|A - \tilde{A}_{glu}\|_F^2 + \|\tilde{A}_{glu} - \mathcal{P}^{so}A\|_F^2$$

Deterministic guarantee: Let $AV = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$, then

$$\begin{aligned}\sigma_j(A - \mathcal{P}^{o}A) &= \sigma_j(R_{22}) \\ \sigma_j^2(A - \tilde{A}_{glu}) &\leq \sigma_j^2(R_{22}) + \|(U_1Q_1)^{+}(U_1Q_2)(R_{22} - (R_{22})_{opt,j-1})\|_2^2\end{aligned}$$

Unified perspective: generalized LU factorization

Given U_1, A, V_1, Q_1 orth. basis of (AV_1), $k \le l < l'$, rank-k approximation,

$$\begin{aligned} \tilde{A}_{glu} &= U^{-1} \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{+} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} V^{-1} \\ &= [U_1^{+}(I - (U_1AV_1)(U_1AV_1)^{+}) + (AV_1)(U_1AV_1)^{+}][U_1A] \neq \mathcal{P}^{so}A \end{aligned}$$

Approximation result: When $k \le l < l'$, the approximation $\tilde{A}_{glu}$ is more accurate than $\mathcal{P}^{so}A$,

$$\|A - \mathcal{P}^{so}A\|_F^T = \|A - \tilde{A}_{glu}\|_F^2 + \|\tilde{A}_{glu} - \mathcal{P}^{so}A\|_F^2$$

Deterministic guarantee: Let $AV = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$, then

$$\begin{aligned} \sigma_j(A - \mathcal{P}^{o}A) &= \sigma_j(R_{22}) \\ \sigma_j^2(A - \tilde{A}_{glu}) &\le \sigma_j^2(R_{22}) + \|(U_1Q_1)^{+}(U_1Q_2)(R_{22} - (R_{22})_{opt,j-1})\|_2^2 \end{aligned}$$

Unified perspective: generalized LU factorization

Given $U_1, A, V_1,$ Q_1 orth. basis of (AV_1), $k \leq l < l'$, rank-k approximation,

$$\begin{aligned}\tilde{A}_{glu} &= U^{-1} \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{+} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} V^{-1} \\ &= [U_1^{+}(I - (U_1 AV_1)(U_1 AV_1)^{+}) + (AV_1)(U_1 AV_1)^{+}][U_1 A] \neq \mathcal{P}^{so} A\end{aligned}$$

Approximation result: When $k \leq l < l'$, the approximation $\tilde{A}_{glu}$ is more accurate than $\mathcal{P}^{so} A$,

$$\|A - \mathcal{P}^{so} A\|_F^T = \|A - \tilde{A}_{glu}\|_F^2 + \|\tilde{A}_{glu} - \mathcal{P}^{so} A\|_F^2$$

Deterministic guarantee: Let $AV = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$, then

$$\begin{aligned}\sigma_j(A - \mathcal{P}^{o} A) &= \sigma_j(R_{22}) \\ \sigma_j^2(A - \tilde{A}_{glu}) &\leq \sigma_j^2(R_{22}) + \|(U_1 Q_1)^{+}(U_1 Q_2)(R_{22} - (R_{22})_{opt,j-1})\|_2^2\end{aligned}$$

Oblivious subspace embedding

- A (k, ϵ, δ) oblivious subspace embedding (OSE) from $\mathbb{R}^n$ to $\mathbb{R}^l$ is a distribution $U_1 \sim \mathbb{D}$ over $l \times n$ matrices. It satisfies with probability $1 - \delta$

$$1 - \epsilon \leq \sigma^2_{\min}(U_1 Q_1) \leq \sigma^2_{\max}(U_1 Q_1) \leq 1 + \epsilon \quad (2)$$

  for any given orthogonal $n \times k$ matrix Q_1. We assume $l \geq k$ and $\epsilon < 1/6$.
- $U_1 \in \mathbb{R}^{l \times n}$ is (ϵ, δ, n) multiplication approximating, if for any A, B having n rows, it satisfies with probability $1 - \delta$,

$$\|A^T U_1^T U_1 B - A^T B\|_F^2 \leq \epsilon \|A\|_F^2 \|B\|_F^2 \quad (3)$$

- Let $U_1 \in \mathbb{R}^{l \times n}$ be subsampled random Hadamard transform (SRHT) obtained by uniform sampling without replacement,
  - With appropriate choices of ϵ, δ, l, U_1 satisfies OSE property (2) (Lemma 4.1 from [Boutsidis and Gittens, 2013]) and the multiplication property (3).

Oblivious subspace embedding

- A (k, ϵ, δ) oblivious subspace embedding (OSE) from $\mathbb{R}^n$ to $\mathbb{R}^l$ is a distribution $U_1 \sim \mathbb{D}$ over $l \times n$ matrices. It satisfies with probability $1 - \delta$

$$1 - \epsilon \leq \sigma^2_{\min}(U_1 Q_1) \leq \sigma^2_{\max}(U_1 Q_1) \leq 1 + \epsilon \tag{2}$$

  for any given orthogonal $n \times k$ matrix Q_1. We assume $l \geq k$ and $\epsilon < 1/6$.
- $U_1 \in \mathbb{R}^{l \times n}$ is (ϵ, δ, n) multiplication approximating, if for any A, B having n rows, it satisfies with probability $1 - \delta$,

$$\|A^T U_1^T U_1 B - A^T B\|_F^2 \leq \epsilon \|A\|_F^2 \|B\|_F^2 \tag{3}$$

- Let $U_1 \in \mathbb{R}^{l \times n}$ be subsampled random Hadamard transform (SRHT) obtained by uniform sampling without replacement,
  - With appropriate choices of ϵ, δ, l, U_1 satisfies OSE property (2) (Lemma 4.1 from [Boutsidis and Gittens, 2013]) and the multiplication property (3).

Probabilistic guarantees

- Combine deterministic guarantees with sketching ensembles satisfying oblivious subspace embedding properties → **better bounds**

- Consider $U_1 \in \mathbb{R}^{l' \times m}$, $V_1 \in \mathbb{R}^{n \times l}$ are SRHT, $l' > l$
  - Compute $\mathcal{P}^o A$ costs $O(mnl)$ flops
  - Compute $\tilde{A}_{glu}$ through generalized LU costs $O(mn \log_2 l')$ flops

Let ρ be the rank of A,
$l = O(1)\epsilon^{-1}(\sqrt{k} + \sqrt{8\log(n/\delta)})^2 \log(k/\delta)$, $l \geq log(n/\delta)log(\rho/\delta)$,
$l' = O(1)\epsilon^{-1}(\sqrt{l} + \sqrt{8\log(m/\delta)})^2 \log(k/\delta)$, $l' \geq log(m/\delta)log(\rho/\delta)$.
With probability $1 - 5\delta$,

$$\begin{aligned} \sigma_j^2(A - \mathcal{P}^o A) &\leq O(1)\sigma_{k+j}^2(A) + O(\frac{\log(\rho/\delta)}{l})(\sigma_{k+j}^2(A) + \dots \sigma_n^2(A)) \\ \sigma_j^2(A - \tilde{A}_{glu}) &\leq O(1)\sigma_{k+j}^2(A) + O(\frac{\log(\rho/\delta)}{l})(\sigma_{k+j}^2(A) + \dots \sigma_n^2(A)). \end{aligned}$$

→ Randomized $\mathcal{P}^o A$ and $\tilde{A}_{glu}$ are kernel approximations (upper bound) of A

Probabilistic guarantees

- Combine deterministic guarantees with sketching ensembles satisfying oblivious subspace embedding properties $\rightarrow$ **better bounds**

- Consider $U_1 \in \mathbb{R}^{l' \times m}$, $V_1 \in \mathbb{R}^{n \times l}$ are SRHT, $l' > l$
  - Compute $\mathcal{P}^o A$ costs $O(mnl)$ flops
  - Compute $\tilde{A}_{glu}$ through generalized LU costs $O(mn \log_2 l')$ flops

Let ρ be the rank of A,
$l = O(1)\epsilon^{-1}(\sqrt{k} + \sqrt{8\log(n/\delta)})^2 \log(k/\delta)$, $l \geq log(n/\delta)log(\rho/\delta)$,
$l' = O(1)\epsilon^{-1}(\sqrt{l} + \sqrt{8\log(m/\delta)})^2 \log(k/\delta)$, $l' \geq log(m/\delta)log(\rho/\delta)$.
With probability $1 - 5\delta$,

$$\begin{aligned} \sigma_j^2(A - \mathcal{P}^o A) &\leq O(1)\sigma_{k+j}^2(A) + O(\frac{\log(\rho/\delta)}{l})(\sigma_{k+j}^2(A) + \dots \sigma_n^2(A)) \\ \sigma_j^2(A - \tilde{A}_{glu}) &\leq O(1)\sigma_{k+j}^2(A) + O(\frac{\log(\rho/\delta)}{l})(\sigma_{k+j}^2(A) + \dots \sigma_n^2(A)). \end{aligned}$$

$\rightarrow$ Randomized $\mathcal{P}^o A$ and $\tilde{A}_{glu}$ are kernel approximations (upper bound) of A

Growth factor in Gaussian elimination

$$\rho(A) \quad := \quad \frac{\max_k ||S_k||_{max}}{||A||_{max}}, \text{ where } A \in \mathbb{R}^{m \times n},$$

S_k is Schur complement obtained at iteration k

Deterministic algorithms, k steps of LU

- LU with partial pivoting: $\rho(A) \leq 2^k$
- CA LU with column/row selection with binary tree tournament pivoting:

$$||S_k(\bar{A}_{11})||_{max} \quad \leq \quad \min((1 + F_{TP}\sqrt{k})||A||_{max}, F_{TP}\sqrt{1 + F_{TP}^2(m-k)}\sigma_k(A))$$

Randomized algorithms

U, V Haar distributed matrices, complete LU factorization,

$$\mathbb{E}[\log(\rho(UAV))] = O(\log(n))$$

Plan

Approximation of tensors

Let $\mathcal{A}$ be a d-order tensor, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$.

- CANDECOMP/PARAFAC (CP) [Hitchcock'27] approximates $\mathcal{A}$ as the sum of k rank-1 tensors, where $q_{1,i} \circ q_{2,i}$ is outer product of $q_{1,i}$ and $q_{2,i}$,

$$\tilde{\mathcal{A}} = \sum_{i=1}^{k} q_{1,i} \circ q_{2,i} \circ \ldots \circ q_{d,i}$$

- Tucker decomposition [Tucker, 1963], computes a rank-$(k_1, \ldots k_d)$ approximation e.g. by using HOSVD and ALS,

$$\begin{aligned} \tilde{\mathcal{A}} &= \mathcal{C} \times_1 Q_1 \times_2 Q_2 \ldots \times_d Q_d \\ &= \sum_{s_1=1}^{k_1} \sum_{s_2=1}^{k_2} \ldots \sum_{s_d=1}^{k_d} \mathcal{C}(s_1, \ldots, s_d) Q_1(:, s_1) \circ \ldots \circ Q_d(:, s_d) \end{aligned}$$

  where $\mathcal{C} \in \mathbb{R}^{k_1 \times k_2 \times \ldots \times k_d}$, $Q_i \in \mathbb{R}^{n_i \times k_i}$, $i = 1, \ldots d$.

- Tensor train or tensor networks for high dimensions

For an overview, see Kolda and Bader, SIAM Review 2009

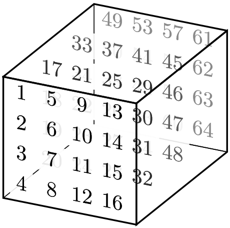
HOSVD for computing a Tucker decomposition

HOSVD for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1 \dots d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \rightarrow RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$



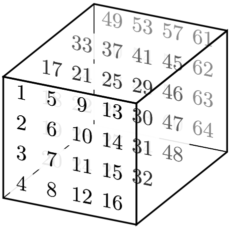
HOSVD for computing a Tucker decomposition

HOSVD for computing a $rank-(k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \rightarrow RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$



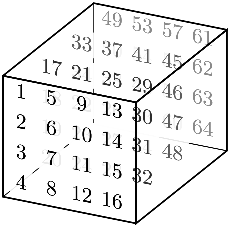
HOSVD for computing a Tucker decomposition

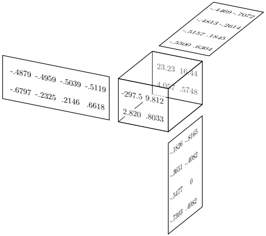
HOSVD for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \rightarrow RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$





HOSVD for computing a Tucker decomposition

HOSVD for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \rightarrow RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$

Error bound:

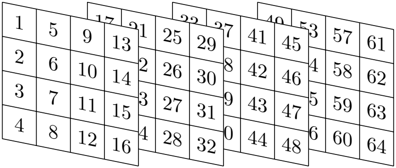
If Q_i are the leading left singular vectors of unfolding A_i, then:

$$\|\mathcal{A} - \tilde{\mathcal{A}}\|_F \leq \sqrt{d} \|\mathcal{A} - \mathcal{A}_{best}\|_F,$$

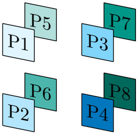
where $\mathcal{A}_{best}$ is the best rank-$k_1, \dots, k_d$ approximation of $\mathcal{A}$.

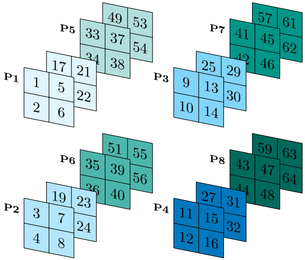
Partitioning for parallel HO-RRQR

- Consider a d-order tensor $\mathcal{A} \in \mathbb{R}^{n \times \ldots \times n}$ ($n = 4, d = 3$ in the example),

$\mathcal{A} =$ 

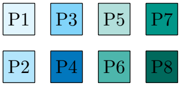
- Partition $\mathcal{A}$ into $\sqrt[d]{P} \times \ldots \times \sqrt[d]{P}$ subtensors $\mathcal{A}_{i_1..i_d} \in \mathbb{R}^{n/\sqrt[d]{P} \times \ldots \times n/\sqrt[d]{P}}$ distributed on $\sqrt[d]{P} \times \ldots \times \sqrt[d]{P}$ processor tensor,

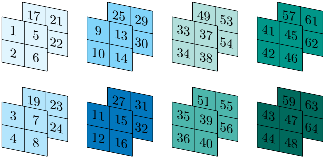




Partitioned unfolding

- Consider 1-mode unfolding of the $2 \times 2 \times 2$ processor tensor,





- Followed on each processor by 1-mode unfolding of its subtensor,

$$A_{1^2} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\ 2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\ \hline 3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\ 4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64 \end{array}\right]$$

- The 1-mode unfolding of $\mathcal{A}$ is:

$$A_1 = \left[\begin{array}{cccccccccccccccc} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{array}\right]$$

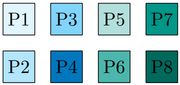
- For any i-mode unfolding, there is a permutation Π_i such that

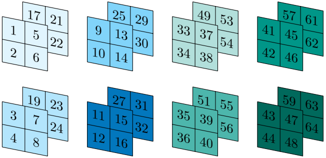
$$A_{i^2} = A_i \Pi_i$$

with M. Beaupère and D. Frenkiel

Partitioned unfolding

- Consider 1-mode unfolding of the $2 \times 2 \times 2$ processor tensor,





- Followed on each processor by 1-mode unfolding of its subtensor,

$$A_{1^2} = \left[\begin{array}{cccc|cccc|cccc|cccc}
1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\
2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\
\hline
3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\
4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64
\end{array}\right]$$

- The 1-mode unfolding of $\mathcal{A}$ is:

$$A_1 = \left[\begin{array}{cccccccccccccccc}
1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\
2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\
3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\
4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64
\end{array}\right]$$

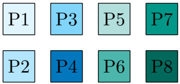
- For any i-mode unfolding, there is a permutation Π_i such that

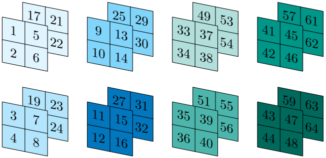
$$A_{i^2} = A_i \Pi_i$$

with M. Beaupère and D. Frenkiel

Partitioned unfolding

- Consider 1-mode unfolding of the $2 \times 2 \times 2$ processor tensor,





- Followed on each processor by 1-mode unfolding of its subtensor,

$$A_{1^2} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\ 2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\ \hline 3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\ 4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64 \end{array}\right]$$

- The 1-mode unfolding of $\mathcal{A}$ is:

$$A_1 = \left[\begin{array}{cccccccccccccccc} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{array}\right]$$

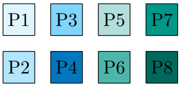
- For any i-mode unfolding, there is a permutation Π_i such that

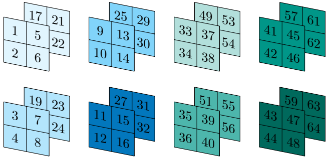
$$A_{i^2} = A_i \Pi_i$$

with M. Beaupère and D. Frenkiel

Partitioned unfolding

- Consider 1-mode unfolding of the $2 \times 2 \times 2$ processor tensor,





- Followed on each processor by 1-mode unfolding of its subtensor,

$$A_{1^2} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\ 2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\ \hline 3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\ 4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64 \end{array}\right]$$

- The 1-mode unfolding of $\mathcal{A}$ is:

$$A_1 = \left[\begin{array}{cccccccccccccccc} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{array}\right]$$

- For any i-mode unfolding, there is a permutation Π_i such that

$$A_{i^2} = A_i \Pi_i$$

with M. Beaupère and D. Frenkiel

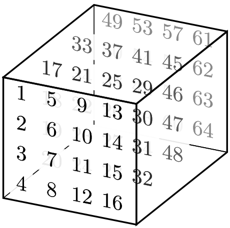
Parallel HO-RRQR

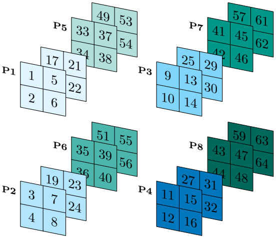
HO-RRQR for computing a $rank-(k_1,\ldots k_d)$ approximation

1. **Input**: Partitioned tensor $\mathcal{A}\in\mathbb{R}^{n\times\ldots\times n}$ on a $\sqrt[d]{P}\times\ldots\times\sqrt[d]{P}$ processor tensor, ranks $k_1,\ldots k_d$
2. For every partitioned unfolding A_{i^2} along mode $i=1\ldots d$, compute factor matrices $Q_i\in\mathbb{R}^{n\times k_i}$ using 2D tournament pivoting (2D TP) on $A_{i^2}^T$:

$$A_{1^2}=\left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\ 2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\ \hline 3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\ 4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64 \end{array}\right]\rightarrow 2D\ TP\begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C}=\mathcal{A}\times_1 Q_1^T\times_2 Q_2^T\ldots\times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}}=\mathcal{C}\times_1 Q_1\ldots\times_d Q_d=\mathcal{A}\times_1 Q_1Q_1^T\ldots\times_d Q_dQ_d^T$





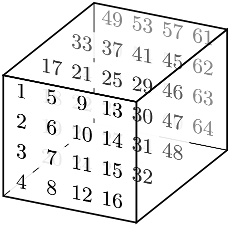
Parallel HO-RRQR

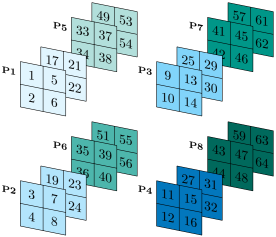
HO-RRQR for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Partitioned tensor $\mathcal{A} \in \mathbb{R}^{n \times \dots \times n}$ on a $\sqrt[d]{P} \times \dots \times \sqrt[d]{P}$ processor tensor, ranks $k_1, \dots k_d$
2. For every partitioned unfolding A_{i^2} along mode $i = 1 \dots d$, compute factor matrices $Q_i \in \mathbb{R}^{n \times k_i}$ using 2D tournament pivoting (2D TP) on $A_{i^2}^T$:

$$A_{1^2} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\ 2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\ \hline 3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\ 4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64 \end{array}\right] \rightarrow 2D\ TP \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$





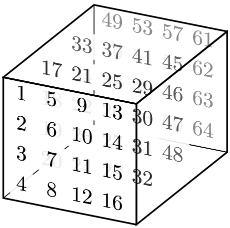
Parallel HO-RRQR

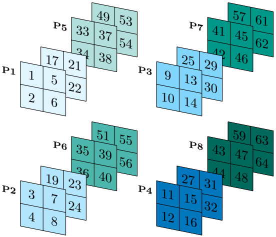
HO-RRQR for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Partitioned tensor $\mathcal{A} \in \mathbb{R}^{n \times \dots \times n}$ on a $\sqrt[d]{P} \times \dots \times \sqrt[d]{P}$ processor tensor, ranks $k_1, \dots k_d$
2. For every partitioned unfolding A_{i^2} along mode $i = 1 \dots d$, compute factor matrices $Q_i \in \mathbb{R}^{n \times k_i}$ using 2D tournament pivoting (2D TP) on $A_{i^2}^T$:

$$A_{1^2} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 5 & 17 & 21 & 9 & 13 & 25 & 29 & 33 & 37 & 49 & 53 & 41 & 45 & 57 & 61 \\ 2 & 6 & 18 & 22 & 10 & 14 & 26 & 30 & 34 & 38 & 50 & 54 & 42 & 46 & 58 & 62 \\ \hline 3 & 7 & 19 & 23 & 11 & 15 & 27 & 31 & 35 & 39 & 51 & 55 & 43 & 47 & 59 & 63 \\ 4 & 8 & 20 & 24 & 12 & 16 & 28 & 32 & 36 & 40 & 52 & 56 & 44 & 48 & 60 & 64 \end{array}\right] \rightarrow 2D\ TP \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$





Parallel HO-RRQR: cost and bounds

HO-RRQR for computing a $rank-(k_1, \dots k_d)$ approximation

1. **Input**: Partitioned tensor $\mathcal{A} \in \mathbb{R}^{n \times \dots \times n}$ on a $\sqrt[d]{P} \times \dots \times \sqrt[d]{P}$ processor tensor, ranks $k_1, \dots k_d$
2. For every partitioned unfolding $A_{i^2} \in \mathbb{R}^{n \times n^{d-1}}$, $i = 1 \dots d$, compute factor matrices $Q_i \in \mathbb{R}^{n \times k_i}$ using 2D tournament pivoting (2D TP) on $A_{i^2}^T$:
   \# messages $\approx d \log_2^2 P$
   Conjecture: can be decreased to $\log_2^2 P$ with a unique reduction tree used by 2D TP on the different unfoldings
3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\tilde{\mathcal{A}} = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$

Error bound:

If factor matrices Q_i are obtained from 2D TP on $A_{i^2}^T$, then:

$$\|\mathcal{A} - \tilde{\mathcal{A}}\|_F \leq \sqrt{1 + \max_i (F_{i,2D-TP}^2 (n - k_i))} \sqrt{d} \|\mathcal{A} - \mathcal{A}_{best}\|_F, \text{ where}$$

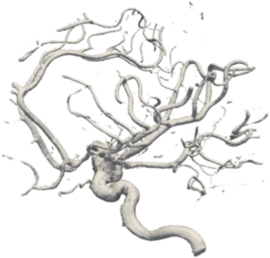
$$F_{i,2D-TP} \leq P k_i^{\log_2 P + 1/2} f^{(1-1/d)\log_2 P + 1}$$

where $\mathcal{A}_{best}$ is the best rank-$k_1, \dots, k_d$ approximation of $\mathcal{A}$.

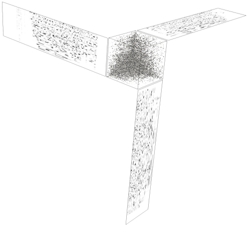
Parallel HO-RRQR: numerical experiments

Isosurface view of $256 \times 256 \times 256$ aneurism:

Original tensor



Core tensor $64 \times 64 \times 64$, 2D TP, 8 procs



Reconstructed image from core tensor $64 \times 64 \times 64$

- Image source: `https://tc18.org/3D_images.html` x-ray scan of the arteries of the right half of a human head with aneurism.

Plan

Open questions for tensors

Many open questions - only a few recalled

Communication bounds few existing results

- Symmetric tensor contractions [Solomonik et al, 18]
- Bound for volume of communication for matricized tensor times Khatri-Rao product [Ballard et al, 17]

Approximation algorithms

- Algorithms as DMRG are intrinsically sequential in the number of modes
- Dynamically adapt the rank to a given error
- Approximation of high rank tensors
  - but low rank in large parts, e.g. due to stationarity in the model the tensor describes

Prospects for the future

- Tensors in high dimensions
  - ERC Synergy project *Extreme-scale Mathematically-based Computational Chemistry project (EMC2)*, with E. Cances, Y. Maday, and J.-P. Piquemal.

Collaborators: O. Balabanov, M. Beaupère, S. Cayrols, J. Demmel, D. Frenkiel, A. Rusciano.

Funding:

- This project has received funding from the European Commission under the Horizon 2020 research and innovation programme Grant agreement No. 810367
- H2020 NLAFET project, ANR

References

Results from following papers:

1. Papers in preparation with O. Balabanov, M. Beaupère, D. Frenkiel on 2D tournament pivoting, parallel HOSVD.
2. J. Demmel, L. Grigori, A. Rusciano, *An improved analysis and unified perspective on deterministic and randomized low rank matrix approximations*, October 2019.
3. V. Ehrlacher, L. Grigori, D. Lombardi, H. Song, *Adaptive hierarchical subtensor partitioning for tensor compression*, SIAM J. Sci. Comput., 2020, in revision.
4. L. Grigori, S. Cayrols, and J. Demmel, *Low rank approximation of a sparse matrix based on LU factorization with column and row tournament pivoting* , SIAM J. Sci. Comput., 40 (2):C181-C209, 2018.
5. J. Demmel, L. Grigori, M. Gu, and H. Xiang, *Communication-Avoiding Rank-Revealing QR Factorization with Column Pivoting*, SIAM Journal on Matrix Analysis and Applications, Vol. 36, No. 1, pp. 55-89, 2015.

References (1)

Boutsidis, C. and Gittens, A. (2013).
Improved matrix algorithms via the subsampled randomized hadamard transform.
SIAM J. Matrix Analysis Applications, 34:1301–1340.

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

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

Golub, G. H. (1965).
Numerical methods for solving linear least squares problems.
Numer. Math., 7:206–216.

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.

Hastad, J. (1990).
Tensor rank is NP-complete.
J. of Algorithms, 11:644–654.

Tucker, L. R. (1963).
Implications of factor analysis of three-way matrices for measurement of change.
In Harris, C. W., editor, *Problems in Measuring Change*, pages 122–137. University of Wisconsin Press.