Communication avoiding rank revealing factorizations, and low rank approximations

L. Grigori
Inria Paris / sabbatical at UC Berkeley

April 2015
Talk based on the papers


- Low rank approximation of a sparse matrix based on LU factorization with column and row tournament pivoting, with S. Cayrols and J. Demmel. Soon on arxiv.
Plan

Motivation

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP
Plan

Motivation

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTTP
Motivation - the communication wall

- Time to move data $\gg$ time per flop
  - Gap steadily and exponentially growing over time

<table>
<thead>
<tr>
<th>Annual improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Time/flop</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>DRAM</td>
</tr>
</tbody>
</table>


We are going to hit the memory wall, unless something basic changes, [W. Wulf, S. McKee, 95].
Compelling numbers (1)

**DRAM bandwidth:**
- Mid 90’s 0.2 bytes/flop - 1 byte/flop
- Past few years 0.02 to 0.05 bytes/flop

**DRAM latency:**
- DDR2 (2007) 120 ns
- DDR4 (2014) 45 ns 2.6x in 7 years
- Stacked memory similar to DDR4

**Time/flop:**
- 2006 Intel Yonah 2GHz x 2 cores (16 GFlops/chip) 1x
- 2015 Intel Haswell 3GHz x 24 cores (288 GFlops/chip) 18x in 9 years

Source: J. Shalf
Compelling numbers (2)

Fetch from DRAM 1 byte of data
- 1988: compute 6 flops
- 2004: compute over 100 flops
- 2015: compute 920 flops

Receive from another processor 1 byte of data
- 2015: compute 4600 - 13616 flops

Example of a supercomputer today:
- Intel Haswell: 8 flops per cycle per core
- Interconnect: 0.25 $\mu$s to 3.7 $\mu$s MPI latency, 8GB/sec MPI bandwidth
Approaches for reducing communication

Tuning

- Overlap communication and computation, at most a factor of 2 speedup

Same numerical algorithm, different schedule of the computation

- Block algorithms for NLA
  - Barron and Swinnerton-Dyer, 1960
  - ScaLAPACK, Blackford et al 97

- Cache oblivious algorithms for NLA
  - Gustavson 97, Toledo 97, Frens and Wise 03, Ahmed and Pingali 00
Approaches for reducing communication

Same algebraic framework, different numerical algorithm

- The approach used in CA algorithms
- More opportunities for reducing communication, may affect stability
Matrix multiply, using $2n^3$ flops (sequential or parallel)
- Lower bound on Bandwidth $= \Omega(\#\text{flops}/M^{1/2})$
- Lower bound on Latency $= \Omega(\#\text{flops}/M^{3/2})$

Same lower bounds apply to LU using reduction
- Demmel, LG, Hoemmen, Langou 2008

\[
\begin{pmatrix}
I & B \\
A & I \\
I & I
\end{pmatrix}
= \begin{pmatrix}
I & I \\
A & I \\
I & I
\end{pmatrix}
\begin{pmatrix}
I & -B \\
I & AB \\
I & I
\end{pmatrix} \tag{1}
\]

And to almost all direct linear algebra
- Ballard, Demmel, Holtz, Schwartz, 2009
2D Parallel algorithms and communication bounds

- Memory per processor $= \frac{n^2}{P}$, the lower bounds become
  $\# \text{words}_\text{moved} \geq \Omega\left(\frac{n^2}{P^{1/2}}\right)$,  $\# \text{messages} \geq \Omega\left(P^{1/2}\right)$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Minimizing #words (not #messages)</th>
<th>Minimizing #words and #messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cholesky</td>
<td>ScaLAPACK</td>
<td>ScaLAPACK</td>
</tr>
<tr>
<td>LU</td>
<td>ScaLAPACK uses partial pivoting</td>
<td>[LG, Demmel, Xiang, 08]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Khabou, Demmel, LG, Gu, 12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uses tournament pivoting</td>
</tr>
<tr>
<td>QR</td>
<td>ScaLAPACK</td>
<td>[Demmel, LG, Hoemmen, Langou, 08]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uses different representation of Q</td>
</tr>
<tr>
<td>RRQR</td>
<td>ScaLAPACK uses column pivoting</td>
<td>[Demmel, LG, Gu, Xiang 13]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uses tournament pivoting, 3x flops</td>
</tr>
</tbody>
</table>

- Only several references shown, block algorithms (ScaLAPACK) and communication avoiding algorithms
- CA algorithms exist also for SVD and eigenvalue computation
2D Parallel algorithms and communication bounds

- Memory per processor = \( \frac{n^2}{P} \), the lower bounds become
- \( \# \text{words moved} \geq \Omega\left(\frac{n^2}{P^{1/2}}\right) \), \( \# \text{messages} \geq \Omega\left(P^{1/2}\right) \)

- Only several references shown, block algorithms (ScaLAPACK) and communication avoiding algorithms
- CA algorithms exist also for SVD and eigenvalue computation
TSQR: QR factorization of a tall skinny matrix

J. Demmel, LG, M. Hoemmen, J. Langou, 08
References: Golub, Plemmons, Sameh 88, Pothen, Raghavan, 89, Da Cunha, Becker, Patterson, 02
Algebra of TSQR

Classic QR: $W = QR_{02} = (I - YT YT^T)R_{02}$

- $Q$ is represented implicitly as a product
- Output: $Q_{00}, Q_{10}, Q_{00}, Q_{20}, Q_{30}, Q_{01}, Q_{11}, Q_{02}, R_{02}$
Algebra of TSQR

Parallel: $w = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \end{bmatrix}$

$\rightarrow \quad R_{00}$

$\rightarrow \quad R_{01}$

$\rightarrow \quad R_{02}$

$\quad R_{10}$

$\rightarrow \quad R_{11}$

$\rightarrow \quad R_{20}$

$\rightarrow \quad R_{11}$

$\rightarrow \quad R_{20}$

$\rightarrow \quad R_{30}$

$\rightarrow \quad R_{11}$

TSQR-HR

Step 0

$C_{ij}$

Step 1

Step 2
Strong scaling

- **Hopper**: Cray XE6 (NERSC): 2 x 12-core AMD Magny-Cours (2.1 GHz)
- **Edison**: Cray CX30 (NERSC): 2 x 12-core Intel Ivy Bridge (2.4 GHz)
- Effective flop rate, computed by dividing $2mn^2 - 2n^3/3$ by measured runtime
- Ballard, Demmel, LG, Jacquelin, Knight, Nguyen, and Solomonik, 2015.
Plan

Motivation

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP
Low rank matrix approximation

- Problem: given \( m \times n \) matrix \( A \), compute rank-\( k \) approximation \( ZW^T \), where \( Z \) is \( m \times k \) and \( W^T \) is \( k \times n \).

- Problem with diverse applications
  - from scientific computing: fast solvers for integral equations, H-matrices
  - to data analytics: principal component analysis, image processing, ...

\[
Ax \rightarrow ZW^Tx
\]

\[
\text{Flops} \quad 2mn \rightarrow 2(m + n)k
\]
Low rank matrix approximation

- Best rank-k approximation $A_k = U_k \Sigma_k V_k$ is rank-k truncated SVD of $A$ [Eckart and Young, 1936]

$$\min_{\text{rank}(\tilde{A}_k) \leq k} \|A - \tilde{A}_k\|_2 = \|A - A_k\|_2 = \sigma_{k+1}(A) \quad (2)$$

$$\min_{\text{rank}(\tilde{A}_k) \leq k} \|A - \tilde{A}_k\|_F = \|A - A_k\|_F = \sqrt{\sum_{j=k+1}^{n} \sigma_j^2(A)} \quad (3)$$

Original image of size $919 \times 707$

Rank-38 approximation, SVD

Rank-75 approximation, SVD

- Image source: https://upload.wikimedia.org/wikipedia/commons/a/a1/Alan_Turing_Aged_16.jpg
Low rank matrix approximation: trade-offs

Flops

Truncated CA-SVD

Lanczos Algorithm

CA (strong) QR with column pivoting
LU with column/row tournament pivoting

Communication

Accuracy

Truncated SVD

(strong) QR with column pivoting
LU with column, rook pivoting
Plan

Motivation

Low rank matrix approximation

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 $m \times n$, consider the decomposition

$$AP_c = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix},$$

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

- $Q(:, 1 : k)$ forms an approximate orthogonal basis for the range of $A$,
- $P_c \begin{bmatrix} R_{11}^{-1} & R_{12} \\ -I & -I \end{bmatrix}$ is an approximate right null space of $A$. 
The factorization from equation (4) is rank revealing if

\[ 1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq q_1(k, n), \]

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

\[ \sigma_{\text{max}}(A) = \sigma_1(A) \geq \ldots \geq \sigma_{\text{min}}(A) = \sigma_n(A) \]

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

\[ \|R_{11}^{-1} R_{12}\|_{\text{max}} \leq q_2(k, n) \]

- Gu and Eisenstat show that given \(k\) and \(f\), there exists a \(P_c\) such that \(q_1(k, n) = \sqrt{1 + f^2k(n - k)}\) and \(q_2(k, n) = f\).
- Factorization computed in \(O(mnk)\) flops.
QR with column pivoting [Businger and Golub, 1965]

Sketch of the algorithm

column norm vector: \(\text{colnrm}(j) = \|A(:,j)\|_2, j = 1 : n.\)

for \(j = 1 : n\) do

1. Pivot, choose column \(p\) of largest norm, swap columns \(j\) and \(p\) in \(A\) and modify \(\text{colnrm}\).
2. Compute Householder matrix \(H_j\) s.t.
   \(H_jA(j : m, j) = \pm\|A(j : m, j)\|_2e_1.\)
3. Update \(A(j : m, j + 1 : n) = H_jA(j : m, j + 1 : n).\)
4. Norm downdate \(\text{colnrm}(j + 1 : n)^2 - = A(j, j + 1 : n)^2.\)

end for

Lower bounds on communication for dense LA
Matrix of size \(n \times n\) distributed over \(P\) processors.

\[
\# \text{ words} \geq \Omega \left( \frac{n^2}{\sqrt{P}} \right), \quad \# \text{ messages} \geq \Omega \left( \sqrt{P} \right).
\] (5)
Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select $k$ columns

- 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
- Permute $A_{ji}$ in leading positions, compute QR with no pivoting

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ * & * \end{pmatrix}$$
Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select \( k \) columns

- 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
- Permute \( A_{ji} \) in leading positions, compute QR with no pivoting

\[
AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ * & * \end{pmatrix}
\]
Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select $k$ columns

- 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
- Permute $A_{ji}$ in leading positions, compute QR with no pivoting

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ * & * \end{pmatrix}$$
Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select $k$ columns

- 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
- Permute $A_{ji}$ in leading positions, compute QR with no pivoting

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ * & * \end{pmatrix}$$
One step of CA_RRQR, tournament pivoting used to select $k$ columns

- 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
- Permute $A_{ji}$ in leading positions, compute QR with no pivoting

\[
AP_{c1} = Q_1 \begin{pmatrix} R_{11} & \ast \\ \ast & \ast \end{pmatrix}
\]
Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select $k$ columns

- 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
- Permute $A_{ji}$ in leading positions, compute QR with no pivoting

\[
AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ * & \end{pmatrix}
\]
Select $b$ columns from a tall and skinny matrix

Given $W$ of size $m \times 2b$, $m \gg b$, $b$ columns are selected as:

$$W = QR_{02} \text{ using TSQR}$$

$$R_{02}P_c = Q_2R_2 \text{ using QRCP}$$

Return $WP_c(:, 1 : b)$

Parallel: $w = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \end{bmatrix} \rightarrow R_{00} \rightarrow R_{10} \rightarrow R_{20} \rightarrow R_{30} \rightarrow R_{01} \rightarrow R_{11} \rightarrow R_{02}$
Reduction trees

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

- **Binary tree:**

  \[
  \begin{array}{cccc}
  A_{00} & A_{10} & A_{20} & A_{30} \\
  \downarrow & \downarrow & \downarrow & \downarrow \\
  f(A_{00}) & f(A_{10}) & f(A_{20}) & f(A_{30}) \\
  \downarrow & \downarrow & \downarrow & \downarrow \\
  f(A_{01}) & f(A_{11}) \\
  \downarrow & \downarrow \\
  f(A_{02}) \\
  \end{array}
  \]

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

- **Flat tree:**

  \[
  \begin{array}{cccc}
  A_{00} & A_{10} & A_{20} & A_{30} \\
  \downarrow & \downarrow & \downarrow \\
  f(A_{00}) & f(A_{10}) & f(A_{20}) & f(A_{30}) \\
  \downarrow & \downarrow \\
  f(A_{01}) \\
  \downarrow \\
  f(A_{02}) \\
  \downarrow \\
  f(A_{03}) \\
  \end{array}
  \]
Selecting $b$ columns by using tournament pivoting reveals the rank of $A$ (for $k = b$) with the following bounds:

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

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

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

$$F_{TP} \leq \frac{1}{\sqrt{2b}} \left(\frac{n}{b}\right)^{\log_2(\sqrt{2fb})}.$$  \hfill (6)

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

- Flat tree of depth $n/b$,

$$F_{TP} \leq \frac{1}{\sqrt{2b}} \left(\sqrt{2fb}\right)^{n/b}.$$  \hfill (7)
Cost of CA-RRQR

Cost of CA-RRQR vs QR with column pivoting

\( n \times n \) matrix on \( \sqrt{P} \times \sqrt{P} \) processor grid, block size \( b \)

- **Flops**: \( 4n^3/P + O(n^2 \log P / \sqrt{P}) \) vs \( (4/3)n^3/P \)
- **Bandwidth**: \( O(n^2 \log P / \sqrt{P}) \) vs same
- **Latency**: \( O(n \log P / b) \) vs \( O(n \log P) \)

Communication optimal, modulo polylogarithmic factors, by choosing

\[
b = \frac{1}{2 \log^2 P} \frac{n}{\sqrt{P}}
\]
Numerical results

- Stability close to QRCP for many tested matrices.

- Absolute value of diagonals of R, L referred to as R-values, L-values.

- Methods compared
  - RRQR: QR with column pivoting
  - CA-RRQR-B with tournament pivoting based on binary tree
  - CA-RRQR-F with tournament pivoting based on flat tree
  - SVD
Numerical results - devil’s stairs

Devil’s stairs (Stewart), a matrix with multiple gaps in the singular values.

Matlab code:

```
Length = 20; s = zeros(n,1); Nst = floor(n/Length);
for i = 1 : Nst
do
    s(1+Length*(i-1):Length*i) = -0.6*(i-1);
end for
s(Length * Nst : end) = -0.6 * (Nst - 1);
s = 10. * s;
A = orth(rand(n)) * diag(s) * orth(randn(n));
```

QLP decomposition (Stewart)

\[ AP_{c_1} = Q_1 R_1 \quad \text{using ca_rqr} \]

\[ R_1^T = Q_2 R_2 \]
Devil’s stairs (Stewart), a matrix with multiple gaps in the singular values.

Matlab code:

```matlab
Length = 20; s = zeros(n,1); Nst = floor(n/Length);
for i = 1 : Nst
do
    s(1+Length*(i-1):Length*i) = -0.6*(i-1);
end for
s(Length * Nst : end) = 0.6 * (Nst - 1);
s = 10. \times s;
A = orth(rand(n)) * diag(s) * orth(randn(n));
```

QLP decomposition (Stewart)

\[ A P_{c_1} = Q_1 R_1 \text{ using ca_rrqr} \]

\[ R_1^T = Q_2 R_2 \]
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]

$$\epsilon \min \{ \| (A\Pi_0)(:, i) \|_2, \| (A\Pi_1)(:, i) \|_2, \| (A\Pi_2)(:, i) \|_2 \}$$  \hspace{1cm} (8)

$$\epsilon \max \{ \| (A\Pi_0)(:, i) \|_2, \| (A\Pi_1)(:, i) \|_2, \| (A\Pi_2)(:, i) \|_2 \}$$  \hspace{1cm} (9)

where $\Pi_j (j = 0, 1, 2)$ are the permutation matrices obtained by QRCP, CARRQR-B, and CARRQR-F, and $\epsilon$ is the machine precision.
Numerical results - a set of 18 matrices

- Ratios $|R(i, i)|/\sigma_i(R)$, for QRCP (top plot), CARRQR-B (second plot), and CARRQR-F (third plot).
- The number along x-axis represents the index of test matrices.
Plan

Motivation

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP
Consider $A$ is SPD and $A = LL^T$.

Given $G(A) = (V, E)$, $G^+(A) = (V, E^+)$ is defined as:
there is an edge $(i, j) \in G^+(A)$ iff there is a path from $i$ to $j$ in $G(A)$
going through lower numbered vertices.

$G(L + L^T) = G^+(A)$, ignoring cancellations.

Definition holds also for directed graphs (LU factorization).

$$A = \begin{pmatrix}
  x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x \\
  x & x & x & x & x & x \\
  x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x \\
\end{pmatrix}$$

$$L + L^T = \begin{pmatrix}
  x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x & x \\
  x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x & x & x & x & x \\
  x & x & x & x & x & x & x \\
\end{pmatrix}$$
LU versus QR

Filled column intersection graph $G_n^+(A)$

- Graph of the Cholesky factor of $A^T A$
- $G(R) \subseteq G_n^+(A)$
- $A^T A$ can have many more nonzeros than $A$
LU versus QR

Numerical stability

- Let $\hat{L}$ and $\hat{U}$ be the computed factors of the block LU factorization. Then

$$\hat{L}\hat{U} = A + E, \quad \|E\|_{max} \leq c_3(n)\epsilon \left(\|A\|_{max} + \|\hat{L}\|_{max}\|\hat{U}\|_{max}\right). \quad (10)$$

- For partial pivoting, $\|L\|_{max} \leq 1$, $\|U\|_{max} \leq 2^n\|A\|_{max}$

In practice, $\|U\|_{max} \leq \sqrt{n}\|A\|_{max}$
Given desired rank $k$, the factorization has the form

$$P_rAP_c = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \tilde{A}_{21}\tilde{A}_{11}^{-1} \\ \tilde{A}_{21} & I \end{pmatrix} \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{22} & S(\tilde{A}_{11}) \end{pmatrix},$$

(11)

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

The rank-$k$ approximation matrix $\tilde{A}_k$ is

$$\tilde{A}_k = \begin{pmatrix} I & \tilde{A}_{12} \\ \tilde{A}_{21}\tilde{A}_{11}^{-1} & \tilde{A}_{11} \end{pmatrix} \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{22} & S(\tilde{A}_{11}) \end{pmatrix} = \begin{pmatrix} \tilde{A}_{11}^{-1} \\ \tilde{A}_{21} \end{pmatrix} \begin{pmatrix} \tilde{A}_{11}^{-1} \tilde{A}_{11} & \tilde{A}_{11}^{-1} \tilde{A}_{12} \\ \tilde{A}_{22} & S(\tilde{A}_{11}) \end{pmatrix}. \quad (12)$$

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

In randomized algorithms, $U = C^+AR^+$, where $C^+$, $R^+$ are Moore-Penrose generalized inverses.
Design space

Non-exhaustive list for selecting $k$ columns and rows:

1. Select $k$ linearly independent columns of $A$ (call result $B$), by using
   1.1 (strong) QRCP/tournament pivoting using QR,
   1.2 LU / tournament pivoting based on LU, with some form of pivoting (column, complete, rook),
   1.3 randomization: premultiply $X = ZA$ where random matrix $Z$ is short and fat, then pick $k$ rows from $X^T$, by some method from 2) below,
   1.4 tournament pivoting based on randomized algorithms to select columns at each step.

2. Select $k$ linearly independent rows of $B$, by using
   2.1 (strong) QRCP / tournament pivoting based on QR on $B^T$, or on $Q^T$, the rows of the thin $Q$ factor of $B$,
   2.2 LU / tournament pivoting based on LU, with pivoting (row, complete, rook) on $B$,
   2.3 tournament pivoting based on randomized algorithms to select rows.
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 $A_{ji}$

2k
\[ A_1 \]
\[ A_0 \]
\[ 2k \]
\[ A_2 \]
\[ A_{10} \]
\[ 2k \]
\[ A_3 \]
\[ A_{20} \]
\[ 2k \]
\[ A_4 \]
\[ A_{30} \]
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 $A_{ji}$
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 $q_1(k, n, F_{TP})$

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

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

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

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

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

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

(13)

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

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

(14)

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

The factorization

\[ P_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \bar{Q}_{21} \bar{Q}_{11}^{-1} \\ \bar{Q}_{21} \bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ S(\bar{Q}_{11}) & \end{pmatrix} \]

satisfies:

\[ \rho_j(\bar{Q}_{21} \bar{Q}_{11}^{-1}) \leq F_{TP}, \]

\[ \frac{1}{q_2(k, m)} \leq \sigma_i(\bar{Q}_{11}) \leq 1, \]

\[ \sigma_{\min}(\bar{Q}_{11}) = \sigma_{\min}(\bar{Q}_{22}) \]

for all \( 1 \leq i \leq k, 1 \leq j \leq m - k \), where \( \rho_j(A) \) is the 2-norm of the j-th row of \( A \), \( q_2(k, m) = \sqrt{1 + F_{TP}^2(m - k)} \).
Sketch of the proof

\[ P_r A P_c = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} \bar{A}_{21} \bar{A}_{11}^{-1} & I \\ \bar{Q}_{21} & \bar{Q}_{11} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ S(\bar{A}_{11}) \end{pmatrix} \]

\[ = \begin{pmatrix} I \\ \bar{Q}_{21} \bar{Q}_{11}^{-1} \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ S(\bar{Q}_{11}) \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ R_{22} \end{pmatrix} \]  \hspace{1cm} (19)

where

\[ \bar{Q}_{21} \bar{Q}_{11}^{-1} = \bar{A}_{21} \bar{A}_{11}^{-1}, \]
\[ S(\bar{A}_{11}) = S(\bar{Q}_{11}) R_{22} = \bar{Q}_{22}^{-T} R_{22}. \]
Sketch of the proof (contd)

\[ \tilde{A}_{11} = \tilde{Q}_{11} R_{11}, \quad (20) \]
\[ S(\tilde{A}_{11}) = S(\tilde{Q}_{11}) R_{22} = \tilde{Q}_{22}^{-T} R_{22}. \quad (21) \]

We obtain

\[ \sigma_i(A) \geq \sigma_i(\tilde{A}_{11}) \geq \sigma_{\text{min}}(\tilde{Q}_{11}) \sigma_i(R_{11}) \geq \frac{1}{q_1(n, k) q_2(m, k)} \sigma_i(A), \]

We also have that

\[ \sigma_{k+j}(A) \leq \sigma_j(S(\tilde{A}_{11})) = \sigma_j(S(\tilde{Q}_{11}) R_{22}) \leq \|S(\tilde{Q}_{11})\|_2 \sigma_j(R_{22}) \leq q_1(n, k) q_2(m, k) \sigma_{k+j}(A), \]

where

\[ q_1(n, k) = \sqrt{1 + F_{TP}^2(n-k)}, \quad q_2(m, k) = \sqrt{1 + F_{TP}^2(m-k)}. \]
LU_CRTP factorization - bounds if \( \text{rank} = k \)

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

\[
\tilde{A} = P_r AP_c = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \end{pmatrix} \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{Q}_{21} \tilde{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ S(\tilde{A}_{11}) \end{pmatrix}
\]

where \( \tilde{A}_{11} \) is of size \( k \times k \) and

\[
S(\tilde{A}_{11}) = \tilde{A}_{22} - \tilde{A}_{21} \tilde{A}_{11}^{-1} \tilde{A}_{12} = \tilde{A}_{22} - \tilde{Q}_{21} \tilde{Q}_{11}^{-1} \tilde{A}_{12}.
\]

It satisfies the following properties:

\[
\rho_l(\tilde{A}_{21} \tilde{A}_{11}^{-1}) = \rho_l(\tilde{Q}_{21} \tilde{Q}_{11}^{-1}) \leq F_{TP},
\]

\[
\|S(\tilde{A}_{11})\|_{max} \leq \min((1 + F_{TP} \sqrt{k})\|A\|_{max}, F_{TP} \sqrt{1 + F_{TP}^2 (m - k) \sigma_k(A)})
\]

\[
1 \leq \frac{\sigma_i(A)}{\sigma_i(\tilde{A}_{11})}, \frac{\sigma_j(S(\tilde{A}_{11}))}{\sigma_k+j(A)} \leq q(m, n, k),
\]

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

\[
q(m, n, k) = \sqrt{(1 + F_{TP}^2 (n - k)) (1 + F_{TP}^2 (m - k))}.
\]
Consider \( T \) block steps of LU_CRTP factorization

\[
P_{r} A P_{c} = \begin{pmatrix}
I & & & \\
L_{21} & I & & \\
& \vdots & \ddots & \vdots \\
L_{T1} & L_{T2} & \ldots & I \\
L_{T+1,1} & L_{T+1,2} & \ldots & L_{T+1,T} & I
\end{pmatrix}
\begin{pmatrix}
U_{11} & U_{12} & \ldots & U_{1T} & U_{1,T+1} \\
& U_{22} & \ldots & U_{2T} & U_{2,T+1} \\
& & \ddots & \vdots & \vdots \\
& & & U_{TT} & U_{T,T+1} \\
& & & & U_{T+1,T+1}
\end{pmatrix}
\]

(26)

where \( U_{tt} \) is \( k \times k \) for \( 1 \leq t \leq T \), and \( U_{T+1,T+1} \) is \((m - Tk) \times (n - Tk)\). Then:

\[
\rho_{l}(L_{i+1,j}) \leq F_{TP},
\]

\[
\|U_{K}\|_{\max} \leq \min \left( (1 + F_{TP} \sqrt{k})^{K/k} \|A\|_{\max}, q_{2}(m, k) q(m, n, k)^{K/k-1} \sigma_{K}(A) \right),
\]

for any \( 1 \leq l \leq k \). \( q_{2}(m, k) = \sqrt{1 + F_{TP}^{2}(m - k)} \), and

\[
q(m, n, k) = \sqrt{(1 + F_{TP}^{2}(n - k))(1 + F_{TP}^{2}(m - k))}.
\]
LU_CRTP factorization - bounds if \( \text{rank} = K = Tk \)

Consider \( T = K/k \) block steps of our LU_CRTP factorization

\[
P_rAP_c = \begin{pmatrix}
I & & & \\
L_{21} & I & & \\
& \vdots & \ddots & \\
L_{T1} & L_{T2} & \cdots & I \\
L_{T+1,1} & L_{T+1,2} & \cdots & L_{T+1,T}
\end{pmatrix}
\begin{pmatrix}
U_{11} & U_{12} & \cdots & U_{1T} & U_{1,T+1} \\
U_{22} & \ddots & \cdots & \vdots & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
U_{TT} & \cdots & \cdots & U_{T,T+1} & U_{T+1,T+1}
\end{pmatrix}
\]

(2)

where \( U_{tt} \) is \( k \times k \) for \( 1 \leq t \leq T \), and \( U_{T+1,T+1} \) is \( (m-Tk) \times (n-Tk) \). Then:

\[
\frac{1}{\prod_{v=0}^{t-2} q(m-vk, n-vk, k)} \leq \frac{\sigma_{(t-1)k+i}(A)}{\sigma_i(U_{tt})} \leq q(m-(t-1)k, n-(t-1)k, k),
\]

\[
1 \leq \frac{\sigma_j(U_{T+1,T+1})}{\sigma_{K+j}(A)} \leq \prod_{v=0}^{K/k-1} q(m-vk, n-vk, k),
\]

for any \( 1 \leq i \leq k \), \( 1 \leq t \leq T \), and \( 1 \leq j \leq \min(m, n) - K \). Here

\( q_2(m, k) = \sqrt{1 + F_{TP}^2(m-k)} \), and

\( q(m, n, k) = \sqrt{(1 + F_{TP}^2(n-k))(1 + F_{TP}^2(m-k))} \).
Tournament pivoting for sparse matrices

Arithmetic complexity

A has arbitrary sparsity structure

\[ \text{flops}(TP_{FT}) \leq 2 \text{nnz}(A)k^2 \]
\[ \text{flops}(TP_{BT}) \leq 8 \frac{\text{nnz}(A)}{P} k^2 \log \frac{n}{k} \]

\( G(A^T A) \) is an \( n^{1/2} \)-separable graph

\[ \text{flops}(TP_{FT}) \leq O(\text{nnz}(A)k^{3/2}) \]
\[ \text{flops}(TP_{BT}) \leq O\left( \frac{\text{nnz}(A)}{P} k^{3/2} \log \frac{n}{k} \right) \]

Randomized algorithm by Clarkson and Woodruff, STOC’13

- Given \( n \times n \) matrix \( A \), it computes \( LDW^T \), where \( D \) is \( k \times k \) such that

\[ ||A - LDW^T||_F \leq (1 + \epsilon)||A - A_k||_F, \ A_k \text{ is best rank-}k \text{ approximation.} \]

\[ \text{flops} \leq O(\text{nnz}(A)) + n\epsilon^{-4} \log^{O(1)}(n\epsilon^{-4}) \]

- Tournament pivoting is faster if \( \epsilon \leq \frac{1}{(\text{nnz}(A)/n)^{1/4}} \)
or if \( \epsilon = 0.1 \) and \( \text{nnz}(A)/n \leq 10^4 \).
Tournament pivoting for sparse matrices

Arithmetic complexity

\( A \) has arbitrary sparsity structure \( G(A^T A) \) is an \( n^{1/2} \)-separable graph

\[
\text{flops}(TP_{FT}) \leq 2 \text{nnz}(A) k^2 \\
\text{flops}(TP_{BT}) \leq 8 \frac{\text{nnz}(A)}{P} k^2 \log \frac{n}{k}
\]

Randomized algorithm by Clarkson and Woodruff, STOC’13

- Given \( n \times n \) matrix \( A \), it computes \( LDW^T \), where \( D \) is \( k \times k \) such that
  \[
  \| A - LDW^T \|_F \leq (1 + \epsilon) \| A - A_k \|_F, \ A_k \text{ is best rank}-k \text{ approximation}.
  \]
  \[
  \text{flops} \leq O(\text{nnz}(A)) + n\epsilon^{-4} \log^{O(1)}(n\epsilon^{-4})
  \]

- Tournament pivoting is faster if \( \epsilon \leq \frac{1}{(\text{nnz}(A)/n)^{1/4}} \)
  or if \( \epsilon = 0.1 \) and \( \text{nnz}(A)/n \leq 10^4 \).
Plan

Motivation

Low rank matrix approximation

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
Here $k = 16$ and the factorization is truncated at $K = 128$ (bars) or $K = 240$ (red lines).

- **LU_CTP**: Column tournament pivoting + partial pivoting
- All singular values smaller than machine precision, $\epsilon$, are replaced by $\epsilon$.
- The number along x-axis represents the index of test matrices.
Results for image of size $919 \times 707$

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

Rank-105 approx, LUPP

Rank-105 approx, LU_CRTP

Rank-209 approx, LU_CRTP

Singular value distribution
Comparing nnz in the factors $L, U$ versus $Q, R$

<table>
<thead>
<tr>
<th>Name/size</th>
<th>$Nnz_{A(\cdot, 1:K)}$</th>
<th>Rank K</th>
<th>$Nnz_{QRCP}/Nnz_{LU_CRTP}$</th>
<th>$Nnz_{LU_CRTP}/Nnz_{LU_UPP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>gemat11</td>
<td>1232</td>
<td>128</td>
<td>2.1</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>4929</td>
<td>512</td>
<td>3.3</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>9583</td>
<td>1024</td>
<td>11.5</td>
<td>3.2</td>
</tr>
<tr>
<td>wang3</td>
<td>896</td>
<td>128</td>
<td>3.0</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>26064</td>
<td>512</td>
<td>2.9</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>7120</td>
<td>1024</td>
<td>2.9</td>
<td>1.2</td>
</tr>
<tr>
<td>Rfdevice</td>
<td>633</td>
<td>128</td>
<td>10.0</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>74104</td>
<td>512</td>
<td>82.6</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>4681</td>
<td>1024</td>
<td>207.2</td>
<td>0.0</td>
</tr>
<tr>
<td>Parab_fem</td>
<td>896</td>
<td>128</td>
<td>–</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>525825</td>
<td>512</td>
<td>–</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>7168</td>
<td>1024</td>
<td>–</td>
<td>0.2</td>
</tr>
<tr>
<td>Mac_econ</td>
<td>384</td>
<td>128</td>
<td>–</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>206500</td>
<td>512</td>
<td>–</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>5970</td>
<td>1024</td>
<td>–</td>
<td>0.2</td>
</tr>
</tbody>
</table>
### 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:**

- **Parab_fem:** $528825 \times 528825$  
- **Mac_econ:** $206500 \times 206500$

<table>
<thead>
<tr>
<th></th>
<th>Time $2k$ cols</th>
<th>Time leaves 32 procs $SPQR + dGEQP3$</th>
<th>Number of MPI processes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td><strong>Parab_fem</strong></td>
<td>0.26</td>
<td>0.26 + 1129</td>
<td>46.7</td>
</tr>
<tr>
<td><strong>Mac_econ</strong></td>
<td>0.46</td>
<td>25.4 + 510</td>
<td>132.7</td>
</tr>
</tbody>
</table>
A parallel QR factorization algorithm with controlled local pivoting.

Linear least squares solutions by Householder transformations.

Communication-avoiding rank-revealing qr decomposition.

Communication-optimal parallel and sequential QR and LU factorizations.
short version of technical report UCB/EECS-2008-89 from 2008.

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

Relative perturbation techniques for singular value problems.

Efficient algorithms for computing a strong rank-revealing QR factorization.
Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems.  
Results used in the proofs

- **Interlacing property of singular values** [Golub, Van Loan, 4th edition, page 487]
  Let \( A = [a_1|\ldots|a_n] \) be a column partitioning of an \( m \times n \) matrix with \( m \geq n \). If \( A_r = [a_1|\ldots|a_r] \), then for \( r = 1 : n - 1 \)
  \[
  \sigma_1(A_{r+1}) \geq \sigma_1(A_r) \geq \sigma_2(A_{r+1}) \geq \ldots \geq \sigma_r(A_{r+1}) \geq \sigma_r(A_r) \geq \sigma_{r+1}(A_{r+1}).
  \]

- **Given** \( n \times n \) matrix \( B \) and \( n \times k \) matrix \( C \), then ([Eisenstat and Ipsen, 1995], p. 1977)
  \[
  \sigma_{\min}(B)\sigma_j(C) \leq \sigma_j(BC) \leq \sigma_{\max}(B)\sigma_j(C), \ j = 1, \ldots, k.
  \]