Communication avoiding rank revealing factorizations, and low rank approximations

L. Grigori

Inria Paris / sabbatical at UC Berkeley

April 2015

Talk based on the papers

- [Demmel et al., 2012] Communication-optimal parallel and sequential QR and LU factorizations, J. W. Demmel, L. Grigori, M. Hoemmen, and J. Langou, SIAM Journal on Scientific Computing, Vol. 34, No 1, 2012.
- [Demmel et al., 2015] Communication avoiding rank revealing QR factorization with column pivoting Demmel, Grigori, Gu, Xiang, SIAM J. Matrix Analysis and Applications, 2015.
- Low rank approximation of a sparse matrix based on LU factorization with column and row tournament pivoting, with S. Cayrols and J. Demmel. 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_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Motivation - the communication wall

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

Annual improvements			
Time/flop		Bandwidth	Latency
59%	Network	26%	15%
	DRAM	23%	5%

Getting up to speed, The future of supercomputing 2004, data from 1995-2004.
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 — 1x
- 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 μs to 3.7 μs MPI latency, 8GB/sec MPI bandwidth

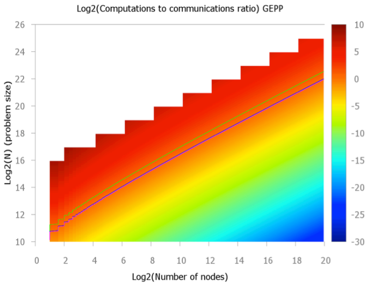
Approaches for reducing communication

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

Communication Complexity of Dense Linear Algebra

- Matrix multiply, using $2n^3$ flops (sequential or parallel)
  - Hong-Kung (1981), Irony/Tishkin/Toledo (2004)
  - Lower bound on Bandwidth $= \Omega(\# flops/M^{1/2})$
  - Lower bound on Latency $= \Omega(\# flops/M^{3/2})$
- Same lower bounds apply to LU using reduction
  - Demmel, LG, Hoemmen, Langou 2008

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

- And to almost all direct linear algebra
  - Ballard, Demmel, Holtz, Schwartz, 2009

2D Parallel algorithms and communication bounds

- Memory per processor $= n^2/P$, the lower bounds become $\#words\_moved \geq \Omega(n^2/P^{1/2})$, $\#messages \geq \Omega(P^{1/2})$

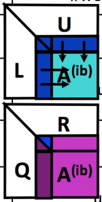
Algorithm	Minimizing #words (not #messages)	Minimizing #words and #messages
Cholesky	ScaLAPACK	ScaLAPACK
LU	ScaLAPACK uses partial pivoting	[LG, Demmel, Xiang, 08] [Khabou, Demmel, LG, Gu, 12] uses tournament pivoting
QR	ScaLAPACK	[Demmel, LG, Hoemmen, Langou, 08] uses different representation of Q
RRQR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

- 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

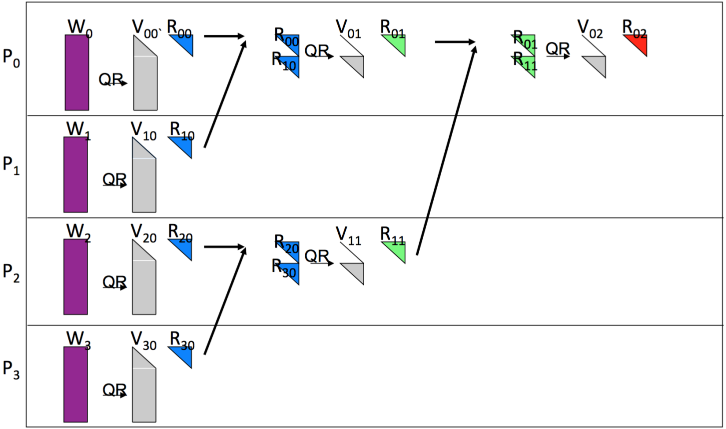
- Memory per processor $= n^2/P$, the lower bounds become
  $\#words\_moved \geq \Omega(n^2/P^{1/2}), \quad \#messages \geq \Omega(P^{1/2})$

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



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

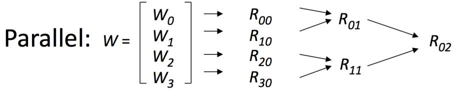
TSQR: QR factorization of a tall skinny matrix



J. Demmel, LG, M. Hoemmen, J. Langou, 08

References: Golub, Plemmons, Sameh 88, Pothen, Raghavan, 89, Da Cunha, Becker, Patterson, 02

Algebra of TSQR

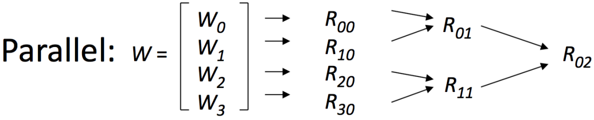


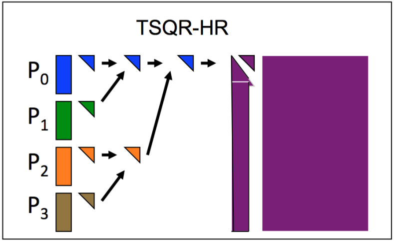
$$W = \begin{pmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \end{pmatrix} = \begin{pmatrix} Q_{00}R_{00} \\ Q_{10}R_{10} \\ Q_{20}R_{20} \\ Q_{30}R_{30} \end{pmatrix} = \begin{pmatrix} Q_{00} & & & \\ & Q_{10} & & \\ & & Q_{20} & \\ & & & Q_{30} \end{pmatrix} \cdot \begin{pmatrix} R_{00} \\ R_{10} \\ R_{20} \\ R_{30} \end{pmatrix}$$

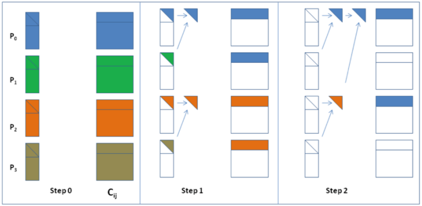
$$\begin{pmatrix} R_{00} \\ R_{10} \\ R_{20} \\ R_{30} \end{pmatrix} = \begin{pmatrix} Q_{01}R_{01} \\ Q_{11}R_{11} \end{pmatrix} = \begin{pmatrix} Q_{01} & \\ & Q_{11} \end{pmatrix} \cdot \begin{pmatrix} R_{01} \\ R_{11} \end{pmatrix} \qquad \begin{pmatrix} R_{01} \\ R_{11} \end{pmatrix} = Q_{02}R_{02}$$

- Classic QR: $W = QR_{02} = (I - YTY^T)R_{02}$
- Q is represented implicitly as a product
- Output: $Q_{00}, Q_{10}, Q_{00}, Q_{20}, Q_{30}, Q_{01}, Q_{11}, Q_{02}, R_{02}$

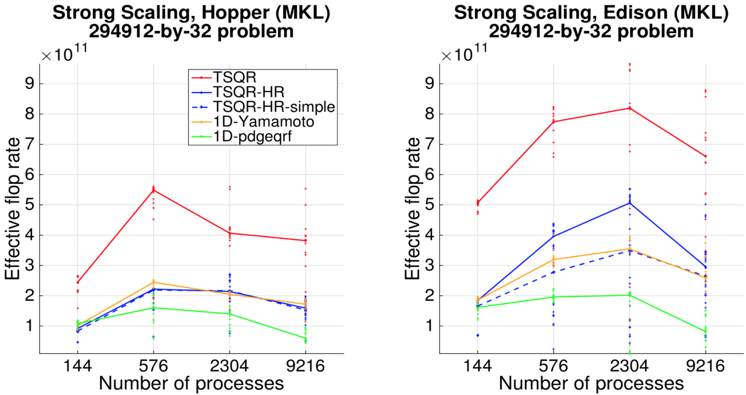
Algebra of TSQR







Strong scaling

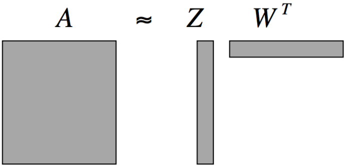


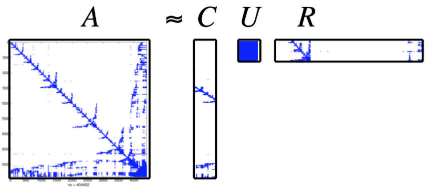
- Hopper: Cray XE6 (NERSC): 2 x 12-core AMD Magny-Cours (2.1 GHz)
- Edison: Cray CX30 (NERSC): 2 x 12-core Intel Ivy Bridge (2.4 GHz)
- Effective flop rate, computed by dividing $2mn^2 - 2n^3/3$ by measured runtime
- Ballard, Demmel, LG, Jacquelin, Knight, Nguyen, and Solomonik, 2015.

Plan

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, ...

$$Ax \rightarrow ZW^T x$$

$$Flops \quad 2mn \rightarrow 2(m+n)k$$

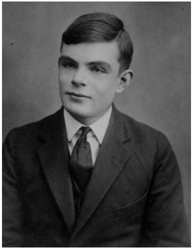
Low rank matrix approximation

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

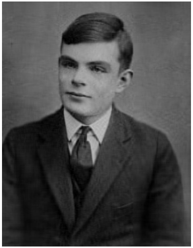
$$\min_{rank(\tilde{A}_k) \leq k} ||A - \tilde{A}_k||_2 = ||A - A_k||_2 = \sigma_{k+1}(A) \quad (2)$$

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

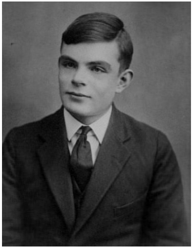
Original image of size 919×707



Rank-38 approximation, SVD

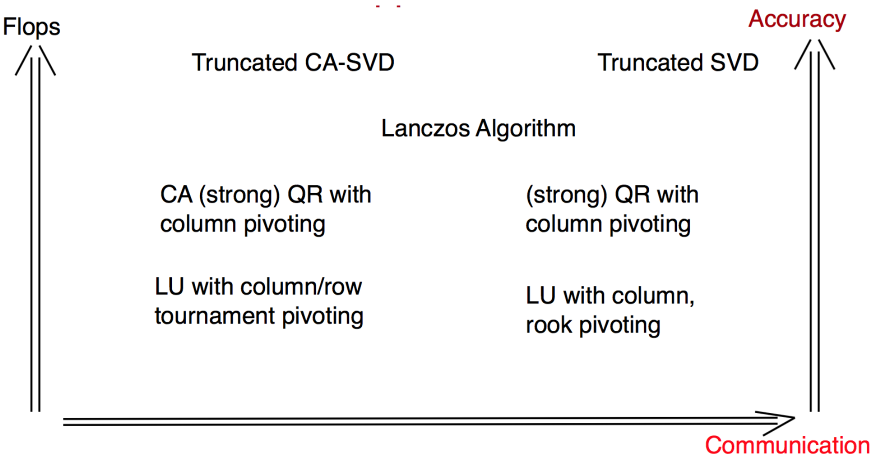


Rank-75 approximation, SVD



- Image source: `https://upload.wikimedia.org/wikipedia/commons/a/a1/Alan_Turing_Aged_16.jpg`

Low rank matrix approximation: trade-offs



Plan

Rank revealing QR factorization

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

$$AP_c = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}, \tag{4}$$

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

Rank revealing QR factorization

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

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

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

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

- Gu and Eisenstat show that given k and f, there exists a P_c such that $q_1(k,n) = \sqrt{1 + f^2 k(n-k)}$ 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: $colnrm(j) = ||A(:,j)||_2, j = 1:n$.
for j = 1 : n **do**

1. Pivot, choose column p of largest norm,
   swap columns j and p in A and modify $colnrm$.
2. Compute Householder matrix H_j s.t.
   $H_j A(j:m,j) = \pm||A(j:m,j)||_2 e_1$.
3. Update $A(j:m,j+1:n) = H_j A(j:m,j+1:n)$.
4. Norm downdate $colnrm(j+1:n)^2 -= A(j,j+1:n)^2$.

end for

Lower bounds on communication for dense LA

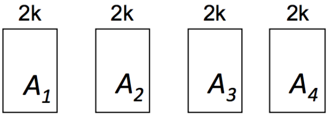
Matrix of size $n \times n$ distributed over P processors.

$$\#\text{ words} \geq \Omega\left(\frac{n^2}{\sqrt{P}}\right), \;\; \#\text{ messages} \geq \Omega\left(\sqrt{P}\right). \qquad (5)$$

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



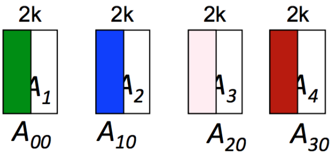
$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

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

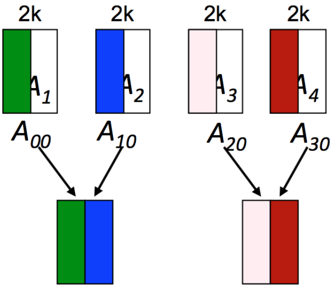
$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$



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



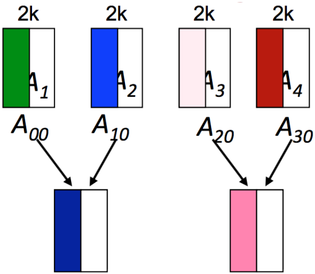
$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

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

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

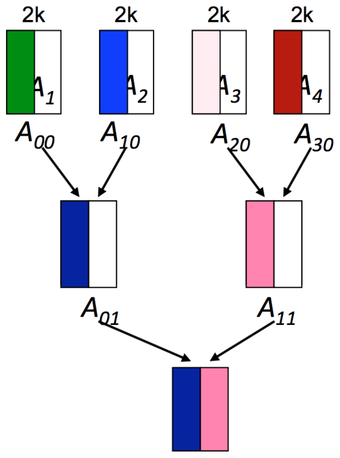


Tournament pivoting [Demmel et al., 2015]

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

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

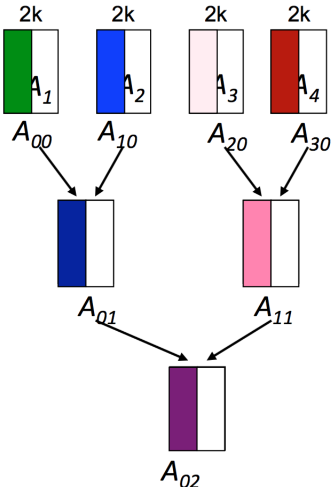


Tournament pivoting [Demmel et al., 2015]

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

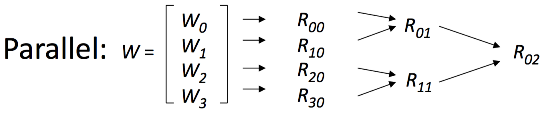
$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$



Select b columns from a tall and skinny matrix

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

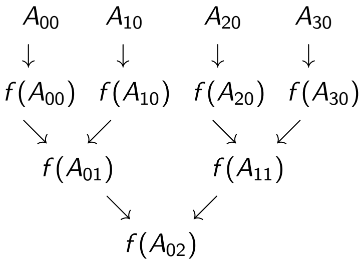
$W = QR_{02}$ using TSQR
$R_{02}P_c = Q_2R_2$ using QRCP
Return $WP_c(:, 1:b)$



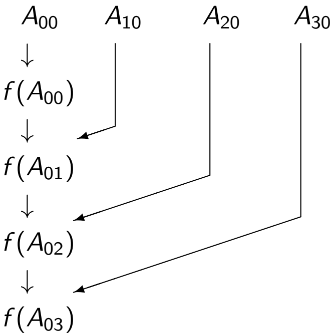
Reduction trees

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

- Binary tree:



- Flat tree:



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

CA-RRQR - bounds for one tournament

Selecting b columns by using tournament pivoting reveals the rank of A (for $k = b$) with the following bounds:

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

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

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

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

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

$$F_{TP} \leq \frac{1}{\sqrt{2b}} \left(\sqrt{2}fb\right)^{n/b} . \tag{7}$$

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

$$\begin{array}{llll} Flops: & 4n^3/P + O(n^2 blogP/\sqrt{P}) & vs & (4/3)n^3/P \\ Bandwidth: & O(n^2 \log P/\sqrt{P}) & vs & same \\ Latency: & O(n \log P/b) & vs & O(n \log P) \end{array}$$

Communication optimal, modulo polylogarithmic factors, by choosing

$$b = \frac{1}{2log^2 P} \frac{n}{\sqrt{P}}$$

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

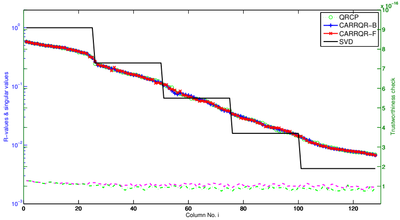
Devil's stairs (Stewart), a matrix with multiple gaps in the singular values.

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
$s(Length * Nst : end) = -0.6 * (Nst - 1);$
$s = 10. \wedge s;$
A = orth(rand(n)) * diag(s) * orth(randn(n));

QLP decomposition (Stewart)

$$\begin{aligned} AP_{c_1} &= Q_1 R_1 \text{ using ca\_rrqr} \\ R_1^T &= Q_2 R_2 \end{aligned}$$



Numerical results - devil's stairs

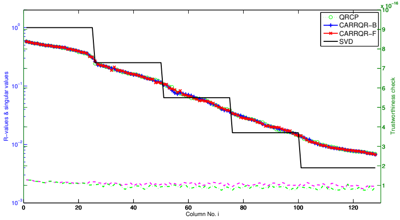
Devil's stairs (Stewart), a matrix with multiple gaps in the singular values.

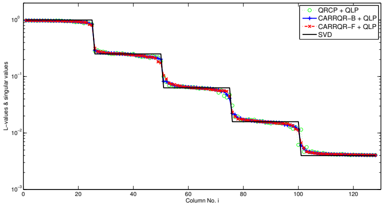
Matlab code:

```
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));
```

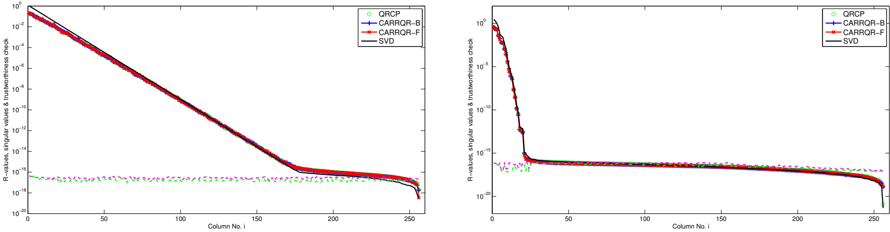
QLP decomposition (Stewart)

$$\begin{aligned} AP_{c_1} &= Q_1 R_1 \text{ using ca\_rrqr} \\ R_1^T &= Q_2 R_2 \end{aligned}$$





Numerical results (contd)



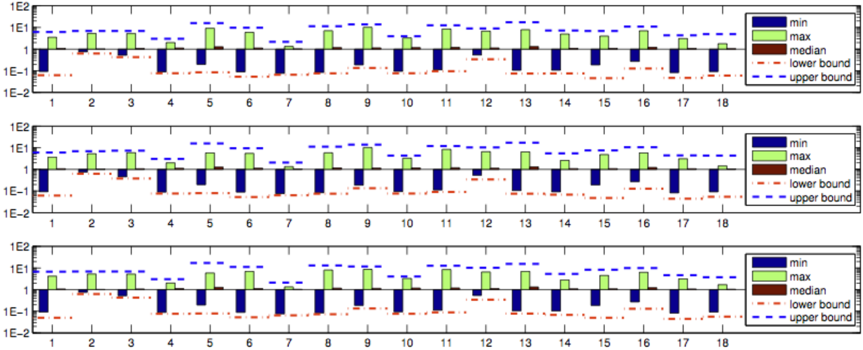
- Left: exponent - exponential Distribution, $\sigma_1 = 1$, $\sigma_i = \alpha^{i-1}$ $(i = 2, \dots, n)$, $\alpha = 10^{-1/11}$ [Bischof, 1991]
- Right: shaw - 1D image restoration model [Hansen, 2007]

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

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

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

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

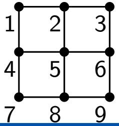
Experimental results, LU_CRTP

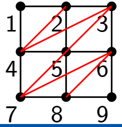
LU versus QR - filled graph $G^+(A)$

- 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).

$$A = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & x & x & & x & & & & & \\ 2 & x & x & x & & x & & & & \\ 3 & & x & x & & & x & & & \\ 4 & x & & & x & x & & x & & \\ 5 & & x & & x & x & x & & x & \\ 6 & & & x & & x & x & & & x \\ 7 & & & & x & & & x & x & \\ 8 & & & & & x & & x & x & x \\ 9 & & & & & & x & & x & x \end{array}$$

$$L + L^T = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & x & x & & x & & & & & \\ 2 & x & x & x & x & x & & & & \\ 3 & & x & x & x & x & x & & & \\ 4 & x & x & x & x & x & & x & & \\ 5 & & x & x & x & x & x & x & x & \\ 6 & & & x & & x & x & x & x & x \\ 7 & & & & x & x & x & x & x & \\ 8 & & & & & x & x & x & x & x \\ 9 & & & & & & x & & x & x \end{array}$$





LU versus QR

Filled column intersection graph $G_\cap^+(A)$

- Graph of the Cholesky factor of $A^T A$
- $G(R) \subseteq G_\cap^+(A)$
- $A^T A$ 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

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

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

Low rank approximation based on LU factorization

- Given desired rank k, the factorization has the form

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

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

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

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

- $\bar{A}_{11}^{-1}$ is never formed, its factorization is used when $\tilde{A}_k$ is applied to a vector.
- In randomized algorithms, $U = C^+ A R^+$, where C^+, R^+ are Moore-Penrose generalized inverses.

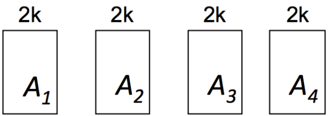
Design space

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.

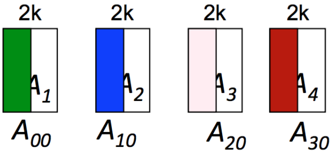
Select k cols using tournament pivoting

- 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}



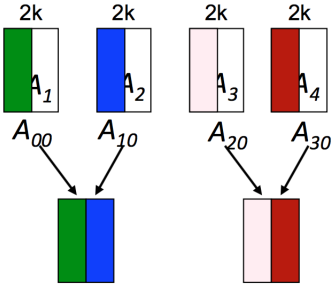
Select k cols using tournament pivoting

- 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}



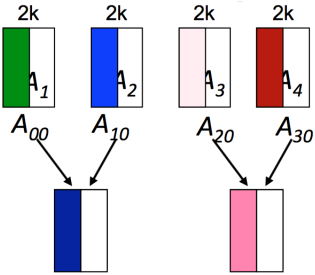
Select k cols using tournament pivoting

- 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}



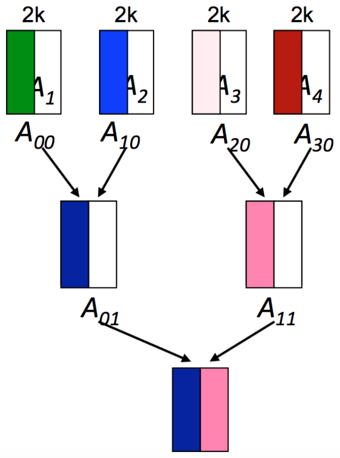
Select k cols using tournament pivoting

- 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}



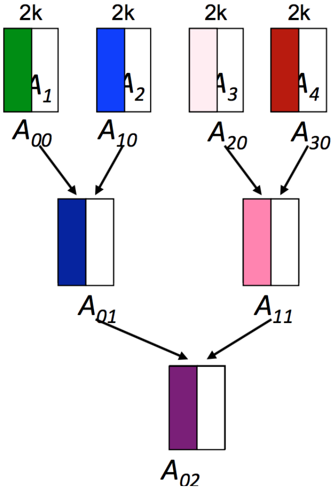
Select k cols using tournament pivoting

- 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}



Our LU_CRTP factorization - one block step

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

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

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

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

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

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

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}, \tag{13}$$

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

$$P_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \tag{14}$$

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

$$P_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \tag{15}$$

satisfies:

$$\rho_j(\bar{Q}_{21}\bar{Q}_{11}^{-1}) \leq F_{TP}, \tag{16}$$

$$\frac{1}{q_2(k,m)} \leq \sigma_i(\bar{Q}_{11}) \leq 1, \tag{17}$$

$$\sigma_{min}(\bar{Q}_{11}) = \sigma_{min}(\bar{Q}_{22}) \tag{18}$$

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

$$
\begin{aligned}
P_r A P_c &= \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} \\
&= \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix} \qquad (19)
\end{aligned}
$$

where

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

Sketch of the proof (contd)

$$\bar{A}_{11} = \bar{Q}_{11} R_{11}, \tag{20}$$
$$S(\bar{A}_{11}) = S(\bar{Q}_{11}) R_{22} = \bar{Q}_{22}^{-T} R_{22}. \tag{21}$$

We obtain

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

We also have that

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

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

LU_CRTP factorization - bounds if $rank = k$

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

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

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

$$S(\bar{A}_{11}) = \bar{A}_{22} - \bar{A}_{21}\bar{A}_{11}^{-1}\bar{A}_{12} = \bar{A}_{22} - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{A}_{12}. \qquad (23)$$

It satisfies the following properties:

$$\begin{aligned} \rho_l(\bar{A}_{21}\bar{A}_{11}^{-1}) &= \rho_l(\bar{Q}_{21}\bar{Q}_{11}^{-1}) \le F_{TP}, & (24) \\ ||S(\bar{A}_{11})||_{max} &\le \min((1 + F_{TP}\sqrt{k})||A||_{max}, F_{TP}\sqrt{1 + F_{TP}^2(m-k)}\sigma_k(A)) \end{aligned}$$

$$1 \le \frac{\sigma_i(A)}{\sigma_i(\bar{A}_{11})}, \frac{\sigma_j(S(\bar{A}_{11}))}{\sigma_{k+j}(A)} \le q(m, n, k), \qquad (25)$$

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

LU_CRTP factorization - bounds if $rank = K = Tk$

Consider T block steps of LU_CRTP factorization

$$P_r A P_c = \begin{pmatrix} I & & & & \\ L_{21} & I & & & \\ \vdots & \vdots & \ddots & & \\ L_{T1} & L_{T2} & \dots & I & \\ L_{T+1,1} & L_{T+1,2} & \dots & L_{T+1,T} & I \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1T} & U_{1,T+1} \\ & U_{22} & \dots & U_{2T} & U_{2,T+1} \\ & & \ddots & \vdots & \vdots \\ & & & U_{TT} & U_{T,T+1} \\ & & & & U_{T+1,T+1} \end{pmatrix} \quad (2$$

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:

$$\begin{aligned} \rho_l(L_{i+1,j}) &\leq F_{TP}, \\ ||U_K||_{max} &\leq \min\left((1 + F_{TP}\sqrt{k})^{K/k} ||A||_{max}, q_2(m,k) q(m,n,k)^{K/k-1} \sigma_K(A)\right), \end{aligned}$$

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 $rank = K = Tk$

Consider $T = K/k$ block steps of our LU_CRTP factorization

$$P_r A P_c = \begin{pmatrix} I & & & & \\ L_{21} & I & & & \\ \vdots & \vdots & \ddots & & \\ L_{T1} & L_{T2} & \dots & I & \\ L_{T+1,1} & L_{T+1,2} & \dots & L_{T+1,T} & I \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1T} & U_{1,T+1} \\ & U_{22} & \dots & U_{2T} & U_{2,T+1} \\ & & \ddots & \vdots & \vdots \\ & & & U_{TT} & U_{T,T+1} \\ & & & & U_{T+1,T+1} \end{pmatrix} \quad (2$$

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

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

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

$$flops(TP_{FT}) \leq O(nnz(A)k^{3/2})$$
$$flops(TP_{BT}) \leq O(\frac{nnz(A)}{P}k^{3/2} \log\frac{n}{k})$$

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 is best rank-k approximation.

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

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

Tournament pivoting for sparse matrices

Arithmetic complexity

A has arbitrary sparsity structure

$$flops(TP_{FT}) \leq 2nnz(A)k^2$$

$$flops(TP_{BT}) \leq 8\frac{nnz(A)}{P}k^2 \log \frac{n}{k}$$

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

$$flops(TP_{FT}) \leq O(nnz(A)k^{3/2})$$

$$flops(TP_{BT}) \leq O(\frac{nnz(A)}{P}k^{3/2} \log \frac{n}{k})$$

Randomized algorithm by Clarkson and Woodruff, STOC'13

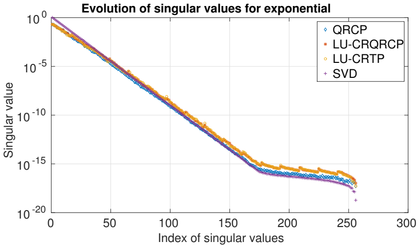
- Given $n \times n$ matrix A, it computes LDW^T, where D is $k \times k$ such that $||A - LDW^T||_F \leq (1+\epsilon)||A - A_k||_F$, A_k is best rank-k approximation.

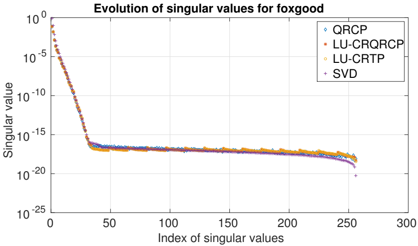
$$flops \leq O(nnz(A)) + n\epsilon^{-4} log^{O(1)}(n\epsilon^{-4})$$

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

Plan

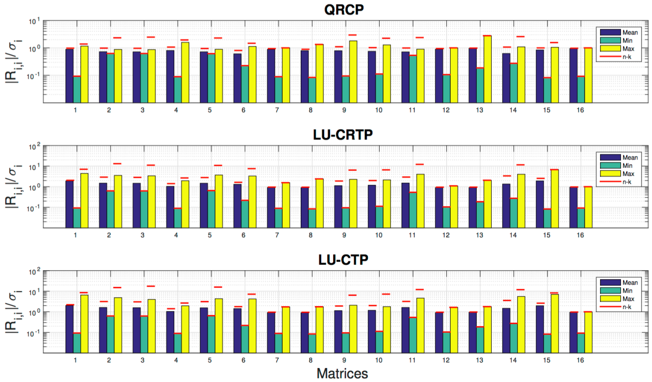
Numerical results





- 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

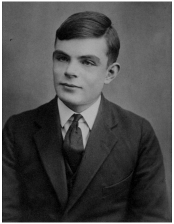
Numerical results



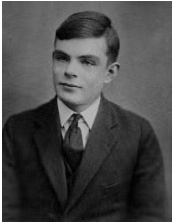
- Here $k = 16$ and the factorization is truncated at $K = 128$ (bars) or $K = 240$ (red lines).
- LU_CTP: Column tournament pivoting + partial pivoting
- All singular values smaller than machine precision, ϵ, are replaced by ϵ.
- The number along x-axis represents the index of test matrices.

Results for image of size 919×707

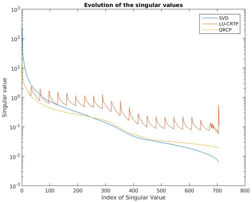
Original image



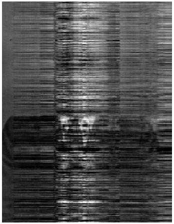
Rank-38 approx, SVD



Singular value distribution

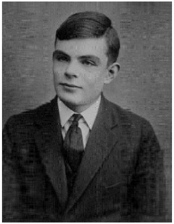


Rank-38 approx, LUPP



Rank-38 approx, LU_CRTP

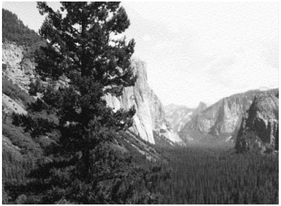
Rank-75 approx, LU_CRTP



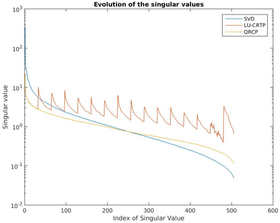
Results for image of size 691 × 505

Original image

Rank-105 approx, SVD

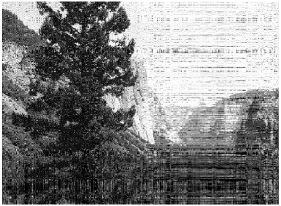


Singular value distribution

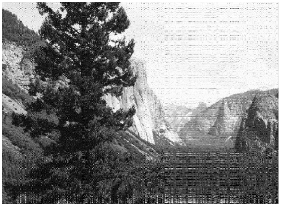


Rank-105 approx, LUPP

Rank-105 approx, LU_CRTP



Rank-209 approx, LU_CRTP



Comparing nnz in the factors L, U versus Q, R

Name/size	*Nnz* $A(:, 1:K)$	*Rank* K	*Nnz QRCP/* *Nnz LU_CRTP*	*Nnz LU_CRTP/* *Nnz LUPP*
*gemat*11	1232	128	2.1	2.2
4929	4895	512	3.3	2.6
	9583	1024	11.5	3.2
*wang*3	896	128	3.0	2.1
26064	3536	512	2.9	2.1
	7120	1024	2.9	1.2
Rfdevice	633	128	10.0	1.1
74104	2255	512	82.6	0.9
	4681	1024	207.2	0.0
Parab_fem	896	128	–	0.5
525825	3584	512	–	0.3
	7168	1024	–	0.2
Mac_econ	384	128	–	0.3
206500	1535	512	–	0.3
	5970	1024	–	0.2

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	–	–

References (1)

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

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

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

Demmel, J. W., Grigori, L., Hoemmen, M., and Langou, J. (2012).
Communication-optimal parallel and sequential QR and LU factorizations.
SIAM Journal on Scientific Computing, (1):206–239.
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.

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

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.

References (2)

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

Results used in the proofs

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

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

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

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