Communication Avoiding: The Past Decade and the New Challenges

L. Grigori
and collaborators

Alpines
Inria Paris and LJLL, Sorbonne University

February 27, 2019

Plan

Motivation of our work

Short overview of results from CA dense linear algebra
- TSQR factorization

Preconditioned Krylov subspace methods
- Enlarged Krylov methods
- Robust multilevel additive Schwarz preconditioner

Unified perspective on low rank matrix approximation
- Generalized LU decomposition

Prospects for the future: tensors in high dimensions
- Hierarchical low rank tensor approximation

Conclusions

The communication wall: compelling numbers

Time/flop 59% annual improvement up to 2004[1]

2008 Intel Nehalem 3.2GHz$\times$4 cores (51.2 GFlops/socket)	1x
2017 Intel Skylake XP 2.1GHz$\times$28 cores (1.8 TFlops/socket)	35x in 9 years

DRAM latency: 5.5% annual improvement up to 2004[1]

DDR2 (2007)	120 ns	1x
DDR4 (2014)	45 ns	2.6x in 7 years
Stacked memory	similar to DDR4	

Network latency: 15% annual improvement up to 2004[1]

Interconnect (example one machine today): 0.25 μs to 3.7 μs MPI latency

Sources:

1. Getting up to speed, The future of supercomputing 2004, data from 1995-2004
2. G. Bosilca (UTK), S. Knepper (Intel), J. Shalf (LBL)

Can we have both scalable and robust methods ?

Difficult ... but crucial ...

since complex and large scale applications very often challenge existing methods

Focus on increasing scalability by reducing/minimizing coummunication while preserving robustness in linear algebra

- Dense linear algebra: ensuring backward stability
- Iterative solvers and preconditioners: bounding the condition number of preconditioned matrix
- Matrix approximation: attaining a prescribed accuracy

Can we have both scalable and robust methods ?

Difficult ... but crucial ...
since complex and large scale applications very often challenge existing methods

Focus on increasing scalability by reducing/minimizing coummunication while preserving robustness in linear algebra

- Dense linear algebra: ensuring backward stability
- Iterative solvers and preconditioners: bounding the condition number of preconditioned matrix
- Matrix approximation: attaining a prescribed accuracy

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, tech report 2008, SISC 2012

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

And to almost all direct linear algebra

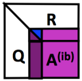
[Ballard, Demmel, Holtz, Schwartz, 09]

2D Parallel algorithms and communication bounds

If memory per processor $= n^2/P$, the lower bounds on communication are

$$\#\text{words\_moved} \geq \Omega(n^2/\sqrt{P}), \quad \#\text{messages} \geq \Omega(\sqrt{P})$$

Most classical algorithms (ScaLAPACK) attain
- lower bounds on #words_moved
- but do not attain lower bounds on #messages



	ScaLAPACK	CA algorithms
LU	partial pivoting	tournament pivoting [LG, Demmel, Xiang, 08] [Khabou, Demmel, LG, Gu, 12]
QR	column based Householder	reduction based Householder [Demmel, LG, Hoemmen, Langou, 08] [Ballard, Demmel, LG, Jacquelin, Nguyen, Solomonik, 14]
RRQR	column pivoting	tournament pivoting [Demmel, LG, Gu, Xiang 13]

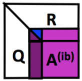
Only several references shown, ScaLAPACK and communication avoiding algorithms

2D Parallel algorithms and communication bounds

If memory per processor $= n^2/P$, the lower bounds on communication are

$$\#\text{words\_moved} \geq \Omega(n^2/\sqrt{P}), \quad \#\text{messages} \geq \Omega(\sqrt{P})$$

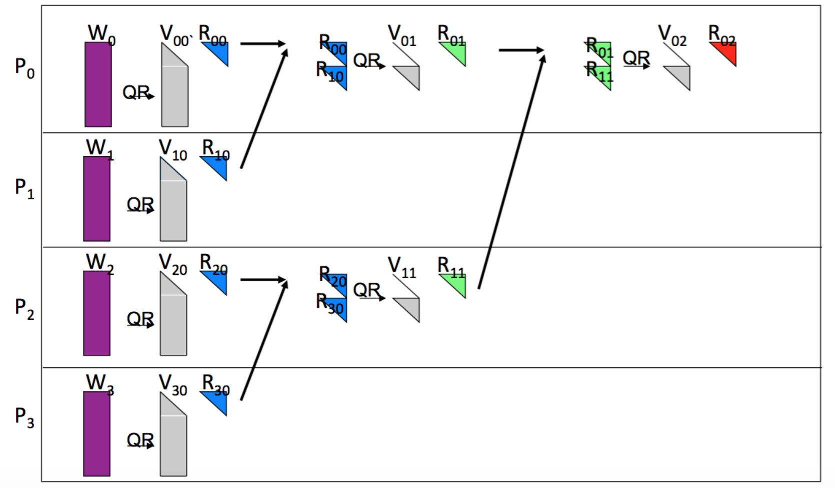
Most classical algorithms (ScaLAPACK) attain
lower bounds on #words_moved
but do not attain lower bounds on #messages



	ScaLAPACK	CA algorithms
LU	partial pivoting	tournament pivoting [LG, Demmel, Xiang, 08] [Khabou, Demmel, LG, Gu, 12]
QR	column based Householder	reduction based Householder [Demmel, LG, Hoemmen, Langou, 08] [Ballard, Demmel, LG, Jacquelin, Nguyen, Solomonik, 14]
RRQR	column pivoting	tournament pivoting [Demmel, LG, Gu, Xiang 13]

Only several references shown, ScaLAPACK and communication avoiding algorithms

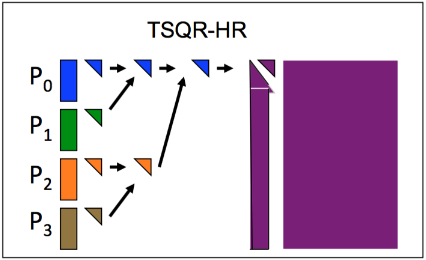
TSQR: CA QR factorization of a tall skinny matrix



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

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

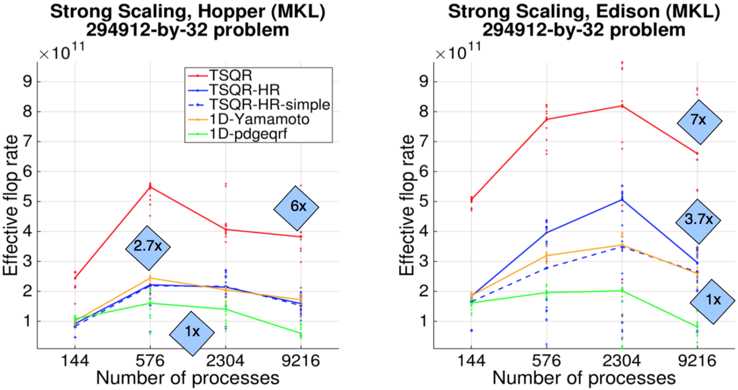
TSQR: CA QR factorization of a tall skinny matrix



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

Ballard, Demmel, LG, Jacquelin, Nguyen, Solomonik, 14

Strong scaling of TSQR



- 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 of our work

Short overview of results from CA dense linear algebra
- TSQR factorization

Preconditioned Krylov subspace methods
- Enlarged Krylov methods
- Robust multilevel additive Schwarz preconditioner

Unified perspective on low rank matrix approximation
- Generalized LU decomposition

Prospects for the future: tensors in high dimensions
- Hierarchical low rank tensor approximation

Conclusions

Challenge in getting scalable and robust solvers

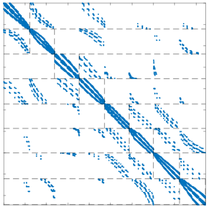
On large scale computers, Krylov solvers reach less than 2% of the peak performance.

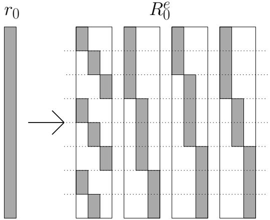
- Typically, each iteration of a Krylov solver requires
  - Sparse matrix vector product
    $\rightarrow$ point-to-point communication
  - Dot products for orthogonalization
    $\rightarrow$ global communication
- When solving complex linear systems most of the highly parallel preconditioners lack robustness
  - wrt jumps in coefficients / partitioning into irregular subdomains, e.g. one level DDM methods (Additive Schwarz, RAS)
  - A few small eigenvalues hinder the convergence of iterative methods

Focus on increasing scalability by reducing coummunication/increasing arithmetic intensity while dealing with small eigenvalues

Enlarged Krylov methods [LG, Moufawad, Nataf, 14]

- Partition the matrix into N domains
- Split the residual r_0 into t vectors corresponding to the N domains,





- Generate t new basis vectors, obtain an **enlarged** Krylov subspace

$$\mathcal{K}_{t,k}(A, r_0) = \text{span}\{R_0^e, AR_0^e, A^2 R_0^e, ..., A^{k-1} R_0^e\}$$

$$\mathcal{K}_k(A, r_0) \subset \mathcal{K}_{t,k}(A, r_0)$$

- Search for the solution of the system $Ax = b$ in $\mathcal{K}_{t,k}(A, r_0)$

Enlarged Krylov subspace methods based on CG

Defined by the subspace $\mathcal{K}_{t,k}$ and the following two conditions:

1. Subspace condition: $x_k \in x_0 + \mathcal{K}_{t,k}$
2. Orthogonality condition: $r_k \perp \mathcal{K}_{t,k}$

- At each iteration, the new approximate solution x_k is found by minimizing $\phi(x) = \frac{1}{2}(x^t A x) - b^t x$ over $x_0 + \mathcal{K}_{t,k}$:

$$\phi(x_k) = min\{\phi(x), \forall x \in x_0 + \mathcal{K}_{t,k}(A, r_0)\}$$

- Can be seen as a particular case of a block Krylov method
  - $AX = S(b)$, such that $S(b)ones(t,1) = b$; $R_0^e = AX_0 - S(b)$
  - Orthogonality condition involves the block residual $R_k \perp \mathcal{K}_{t,k}$

Enlarged Krylov subspace methods based on CG

Defined by the subspace $\mathcal{K}_{t,k}$ and the following two conditions:

1. Subspace condition: $x_k \in x_0 + \mathcal{K}_{t,k}$
2. Orthogonality condition: $r_k \perp \mathcal{K}_{t,k}$

- At each iteration, the new approximate solution x_k is found by minimizing $\phi(x) = \frac{1}{2}(x^t A x) - b^t x$ over $x_0 + \mathcal{K}_{t,k}$:

$$\phi(x_k) = min\{\phi(x), \forall x \in x_0 + \mathcal{K}_{t,k}(A, r_0)\}$$

- Can be seen as a particular case of a block Krylov method
    - $AX = S(b)$, such that $S(b) ones(t, 1) = b$; $R_0^e = AX_0 - S(b)$
    - Orthogonality condition involves the block residual $R_k \perp \mathcal{K}_{t,k}$

Convergence analysis

Given

- A is an SPD matrix, x^* is the solution of $Ax = b$
- $||x^* - \overline{x}_k||_A$ is the k^{th} error of CG, $e_0 = x^* - x_0$
- $||x^* - x_k||_A$ is the k^{th} error of ECG

Result

CG

$$||x^* - \overline{x}_k||_A \leq 2||e_0||_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k$$

where $\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$

ECG

$$||x^* - x_k||_A \leq 2||\hat{e}_0||_A \left(\frac{\sqrt{\kappa_t}-1}{\sqrt{\kappa_t}+1}\right)^k$$

where $\kappa_t = \frac{\lambda_{max}(A)}{\lambda_t(A)}$, $\hat{e}_0 \equiv E_0(\Phi_1^\top E_0)^{-1}\begin{pmatrix}0\\ \vdots\\ 0\\ 1\end{pmatrix}$, Φ_1 denotes the t eigenvectors associated to the smallest eigenvalues, and E_0 is the initial enlarged error.

From here on, results on enlarged CG obtained with O. Tissot

Classical CG *vs.* Enlarged CG derived from Block CG

Algorithm 1 Classical CG

$p_1 = r_0(r_0^\top A r_0)^{-1/2}$
while $||r_{k-1}||_2 > \varepsilon ||b||_2$ **do**
    $\alpha_k = p_k^\top r_{k-1}$
    $x_k = x_{k-1} + p_k \alpha_k$
    $r_k = r_{k-1} - A p_k \alpha_k$
    $z_{k+1} = r_k - p_k (p_k^\top A r_k)$
    $p_{k+1} = z_{k+1}(z_{k+1}^\top A z_{k+1})^{-1/2}$
end while

Cost per iteration

\# flops $= O(\frac{n}{P}) \leftarrow$ BLAS 1 & 2
\# words $= O(1)$
\# messages $= O(1)$ from SpMV + $O(logP)$ from dot prod + norm

Algorithm 2 ECG

$P_1 = R_0^e (R_0^{e\top} A R_0^e)^{-1/2}$
while $|| \sum_{i=1}^{\top} R_k^{(i)} ||_2 < \varepsilon ||b||_2$ **do**
    $\alpha_k = P_k^\top R_{k-1}$ ▷ $t \times t$ matrix
    $X_k = X_{k-1} + P_k \alpha_k$
    $R_k = R_{k-1} - A P_k \alpha_k$
    $Z_{k+1} = A P_k - P_k (P_k^\top A A P_k) - P_{k-1}(P_{k-1}^\top A A P_k)$
    $P_{k+1} = Z_{k+1}(Z_{k+1}^\top A Z_{k+1})^{-1/2}$
end while
$x = \sum_{i=1}^{\top} X_k^{(i)}$

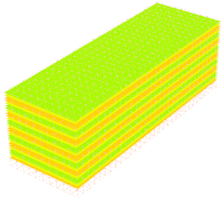
Cost per iteration

\# flops $= O(\frac{nt^2}{P}) \leftarrow$ BLAS 3
\# words $= O(t^2) \leftarrow$ Fit in the buffer
\# messages $= O(1)$ from SpMV + $O(logP)$ from A-ortho

Test cases

- 3 of 5 largest SPD matrices of Tim Davis' collection
- Heterogeneous linear elasticity problem discretized with FreeFem++ using $\mathbb{P}_1$ FE

$$
\begin{aligned}
\operatorname{div}(\sigma(u)) + f &= 0 && \text{on } \Omega,\\
u &= u_D && \text{on } \partial\Omega_D,\\
\sigma(u)\cdot n &= g && \text{on } \partial\Omega_N,
\end{aligned}
$$



- $u \in \mathbb{R}^d$ is the unknown displacement field, f is some body force.
- Young's modulus E and Poisson's ratio ν, $(E_1, \nu_1) = (2 \cdot 10^{11}, 0.25)$, and $(E_2, \nu_2) = (10^7, 0.45)$.

Name	Size	Nonzeros	Problem
Hook_1498	1,498,023	59,374,451	Structural problem
Flan_1565	1,564,794	117,406,044	Structural problem
Queen_4147	4,147,110	316,548,962	Structural problem
Ela_4	4,615,683	165,388,197	Linear elasticity

Enlarged CG: dynamic reduction of search directions

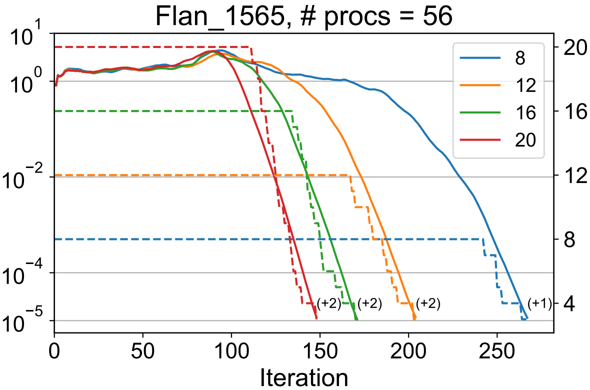


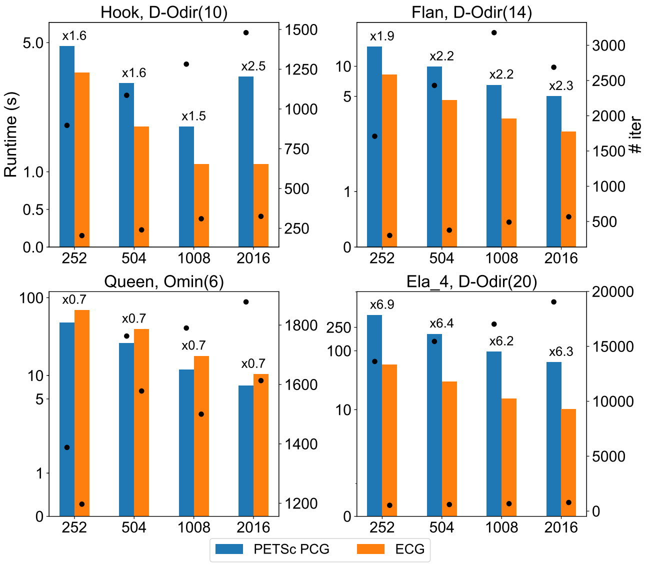
Figure : solid line: normalized residual (scale on the left), dashed line: number of search directions (scale on the right)

→ **Reduction occurs when the convergence has started**

Strong scalability

- Run on Kebnekaise, Umeå University (Sweden) cluster, 432 nodes with Broadwell processors (28 cores per node)
- Compiled with Intel Suite 18
- PETSc 3.7.6 (linked with the MKL)
- Pure MPI (no threading)
- Stopping criterion tolerance is set to 10^{-5} (PETSc default value)
- Block diagonal preconditioner, number blocks equals number of MPI processes
  - Cholesky factorization on the block with MKL-PARDISO solver

Strong scalability



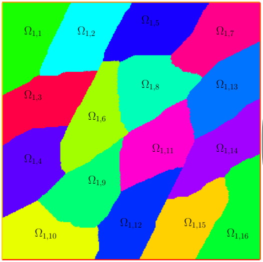
Additive Schwarz methods

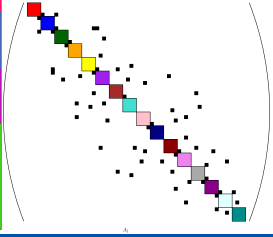
Solve $M^{-1}Ax = M^{-1}b$, where $A \in \mathbb{R}^{n\times n}$ is SPD
Original idea from Schwarz algorithm at the continuous level (Schwarz 1870)

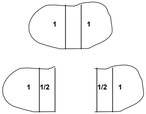
- Symmetric formulation, Additive Schwarz (1989)

$$M_{AS,1}^{-1} := \sum_{j=1}^{N_1} R_{1j}^T A_{1j}^{-1} R_{1j}$$

- DOFs partitioned into N_1 domains of dimensions $n_{11}, n_{12}, \ldots n_{1,N_1}$
- $R_{1j} \in \mathbb{R}^{n_{1j}\times n}$: restriction operator
- $A_{1j} \in \mathbb{R}^{n_{1j}\times n_{1j}}$: matrix associated to domain j, $A_{1j} = R_{1j} A R_{1j}^T$
- $(D_{1j})_{j=1:N_1}$: algebraic partition of unity







Upper bound for the eigenvalues of $M_{AS,1}^{-1}A$

Let k_c be number of distinct colours to colour the subdomains of A. The following holds (e.g. Chan and Mathew 1994)

$$\lambda_{max}(M_{AS,1}^{-1}A) \leq k_c$$

$\rightarrow$ Two level preconditioners are required

- Early references: [Nicolaides 87], [Morgan 92], [Chapman, Saad 92], [Kharchenko, Yeremin 92], [Burrage, Ehrel, and Pohl, 93]
- Our work uses the theoretical framework of the Fictitious space lemma (Nepomnyaschikh 1991).

Construction of the coarse space for 2nd level

Consider the generalized eigenvalue problem for each domain j, for given τ:

$$\text{Find } (u_{1jk},\ \lambda_{1jk})\ \in \mathbb{R}^{n_{i,1}} \times \mathbb{R}, \lambda_{1jk} \leq 1/\tau$$
$$\text{such that } R_{1j}\tilde{A}_{1j}R_{1j}^T u_{1jk} = \lambda_{1jk} D_{1j} A_{1j} D_{1j} u_{1jk}$$

where $\tilde{A}_{1j}$ is a local SPSD splitting of A suitably permuted, V_1 basis of S_1,

$$\mathcal{S}_1 \quad := \quad \bigoplus_{j=1}^{N_1} D_{1j} R_{1j}^\top Z_{1j}, \quad Z_{1j} = \text{span}\left\{u_{1jk} \mid \lambda_{1jk} < 1/\tau\right\}$$

$$M_{AS,2}^{-1} \quad := \quad V_1\left(V_1^T A V_1\right)^{-1} V_1^T + \sum_{j=1}^{N_1} R_{1j}^T A_{1j}^{-1} R_{1j}$$

Theorem (H. Al Daas, LG, 2018)

$$\kappa\left(M_{AS,2_{ALSP}}^{-1} A\right) \leq (k_c + 1)\left(2 + (2k_c + 1)k_m\tau\right)$$

where k_c is the number of distinct colors required to color the graph of A, $k_m \leq N_1$, where N_1 is the number of subdomains

Construction of the coarse space for 2nd level

Consider the generalized eigenvalue problem for each domain j, for given τ:

$$\begin{aligned} &\text{F}\mathit{ind}\ (u_{1jk},\ \lambda_{1jk})\ \in \mathbb{R}^{n_{i,1}} \times \mathbb{R}, \lambda_{1jk} \leq 1/\tau \\ &\mathit{such\ that}\ R_{1j}\tilde{A}_{1j}R_{1j}^{T} u_{1jk} = \lambda_{1jk} D_{1j} A_{1j} D_{1j} u_{1jk} \end{aligned}$$

where $\tilde{A}_{1j}$ is a local SPSD splitting of A suitably permuted, V_1 basis of S_1,

$$\begin{aligned} \mathcal{S}_1 \quad &:= \quad \bigoplus_{j=1}^{N_1} D_{1j} R_{1j}^{\top} Z_{1j}, \quad Z_{1j} = \mathrm{span}\left\{u_{1jk} \mid \lambda_{1jk} < 1/\tau\right\} \\ M_{AS,2}^{-1} \quad &:= \quad V_1 \left(V_1^T A V_1\right)^{-1} V_1^T + \sum_{j=1}^{N_1} R_{1j}^T A_{1j}^{-1} R_{1j} \end{aligned}$$

- Generalization of Geneo theory fulfilled by standard FE and bilinear forms [Spillane, Dolean, Hauret, Nataf, Pechstein, Scheichl'13]
- $k_m =$ max number of domains that share a common vertex
- $\tilde{A}_{1j}$ is the Neumann matrix of domain j, D_{1j} is nonsingular.

Local SPSD splitting of A wrt a subdomain

- For each domain j, a local SPSD splitting is a decomposition $A = \tilde{A}_{1j} + C$, where $\tilde{A}_{1j}$ and C are SPSD
- Ideally $\tilde{A}_{1j}$ is local
- Consider domain 1, where A_{11} corresponds to interior DOFs, A_{22} the DOFs at the interface of 1 with all other domains, and A_{33} the rest of DOFs:

$$A = \begin{pmatrix} A_{11} & A_{12} & \\ A_{21} & A_{22} & A_{23} \\ & A_{32} & A_{33} \end{pmatrix}$$

- We note $S(A_{22})$ the Schur complement with respect to A_{22},

$$S(A_{22}) = A_{22} - A_{21}A_{11}^{-1}A_{12} - A_{23}A_{33}^{-1}A_{32}.$$

Algebraic local SPSD splitting lemma

Let $A \in \mathbb{R}^{n\times n}$, an SPD matrix, and $\tilde{A}_{11} \in \mathbb{R}^{n\times n}$ be partitioned as follows

$$A = \begin{pmatrix} A_{11} & A_{12} & \\ A_{21} & A_{22} & A_{23} \\ & A_{32} & A_{33} \end{pmatrix}, \quad \tilde{A}_{11} = \begin{pmatrix} A_{11} & A_{12} & \\ A_{21} & \bar{A}_{22} & \\ & & 0 \end{pmatrix}$$

where $A_{ii} \in \mathbb{R}^{m_i\times m_i}$ is non trivial matrix for $i \in \{1,2,3\}$. If $\bar{A}_{22} \in \mathbb{R}^{m_2\times m_2}$ is a symmetric matrix verifying the following inequalities

$$u^T A_{21} A_{11}^{-1} A_{12} u \le u^T \bar{A}_{22} u \le u^T \left(A_{22} - A_{23} A_{33}^{-1} A_{32}\right) u, \quad \forall u \in \mathbb{R}^{m_2},$$

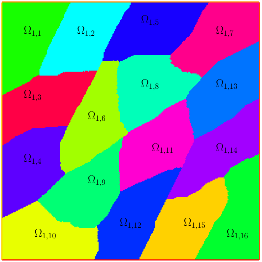
then $A - \tilde{A}_{11}$ is SPSD, that is the following inequality holds

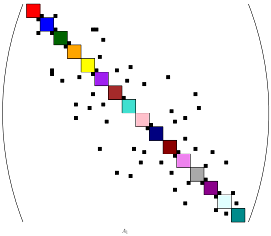
$$0 \le u^T \tilde{A}_{11} u \le u^T A u, \quad \forall u \in \mathbb{R}^n.$$

- Remember: $S(A_{22}) = A_{22} - A_{23} A_{33}^{-1} A_{32} - A_{21} A_{11}^{-1} A_{12}$.
- The left and right inequalities are optimal

Multilevel Additive Schwarz M_{MAS}

with H. Al Daas, P. Jolivet, P. H. Tournier





for level $i = 1$ and each domain $j = 1 : N_1$ in parallel $(A = A_1)$ **do**

$A_{1j} = R_{1j} A_1 R_{1j}^T$ *(local matrix associated to domain j)*

$\tilde{A}_{1j}$ is Neumann matrix of domain j (local SPSD splitting)

Solve Gen EVP, set $Z_{1j} = \text{span}\left\{ u_{1jk} \mid \lambda_{1jk} < \frac{1}{\tau} \right\}$

Find $(u_{1jk}, \lambda_{1jk}) \in \mathbb{R}^{n_{1j}} \times \mathbb{R}$

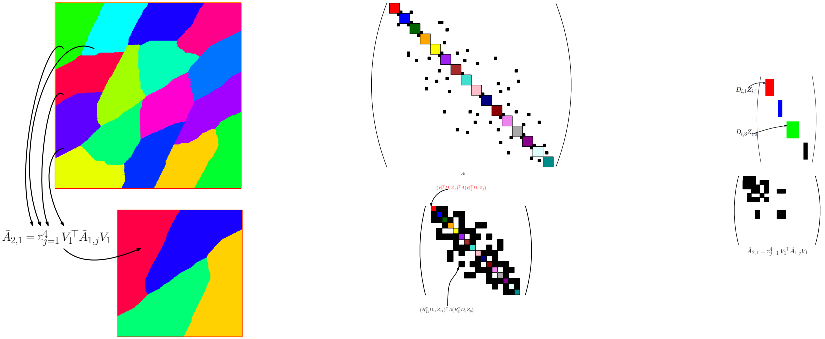
$R_{1j} \tilde{A}_{1j} R_{1j}^\top u_{1jk} = \lambda_{1jk} D_{1j} A_{1j} D_{1j} u_{1jk}$.

Let $\mathcal{S}_1 = \bigoplus_{j=1}^{N_1} D_{1j} R_{1j}^\top Z_{1j}$, V_1 basis of S_1, $A_2 = V_1^T A_1 V_1$, $A_2 \in \mathbb{R}^{n_2 \times n_2}$

end for

Preconditioner defined as: $M_{A_1,MAS}^{-1} = V_1 A_2^{-1} V_1^T + \sum_{j=1}^{N_1} R_{1j}^\top A_{1j}^{-1} R_{1j}$

Multilevel Additive Schwarz M_{MAS}



for level $i = 2$ to $\log_d N_i$ **do**

for each domain $j = 1 : N_i$ in parallel **do**

$\tilde{A}_{ij} = \sum_{k=(j-1)d+1}^{jd} V_{i-1}^T \tilde{A}_{i-1,k} V_{i-1}$ (local SPSD splitting)

$A_{ij} = R_{ij} A_i R_{ij}^T$ (local matrix associated to domain j)

Solve Gen EVP, $Z_{ij} = \text{span}\left\{ u_{ijk} \mid \lambda_{ijk} < \frac{1}{\tau} \right\}$

Find $(u_{ijk}, \lambda_{ijk}) \in \mathbb{R}^{n_{ij}} \times \mathbb{R}$

$R_{ij} \tilde{A}_{ij} R_{ij}^\top u_{ijk} = \lambda_{ijk} D_{ij} A_{ij} D_{ij} u_{ijk}$

$$M_{A_i,MAS}^{-1} = V_i A_{i+1}^{-1} V_i^T + \sum_{j=1}^{N_i} R_{ij}^\top A_{ij}^{-1} R_{ij}$$

Let $\mathcal{S}_i = \bigoplus_{j=1}^{N_i} D_{ij} R_{ij}^\top Z_{ij}$, V_i basis of S_i, $A_{i+1} = V_i^T A_i V_i$, $A_{i+1} \in \mathbb{R}^{n_{i+1} \times n_{i+1}}$

end for end for

Robustness and efficiency of multilevel AS

Theorem (Al Daas, LG, Jolivet, Tournier)

Given the multilevel preconditioner defined at each level $i = 1 : \log_d N_1$ as

$$M^{-1}_{A_i,MAS} = V_i A^{-1}_{i+1} V_i^T + \sum_{j=1}^{N_i} R_{ij}^\top A_{ij}^{-1} R_{ij}$$

then $M^{-1}_{MAS} = M^{-1}_{A_1,MAS}$ and,

$$\kappa(M^{-1}_{A_i,MAS} A_i) \leq (k_{ic} + 1)\left(2 + (2k_{ic} + 1) k_i \tau\right),$$

where k_{ic} = number of distinct colours to colour the graph of A,
k_i = max number of domains that share a common vertex.

Communication efficiency

- Construction of M^{-1}_{MAS} preconditioner requires $O(\log_d N_1)$ messages.
- Application of M^{-1}_{MAS} preconditioner requires $O((\log_2 N_1)^{\log_d N_1})$ messages per iteration.

Robustness and efficiency of multilevel AS

Theorem (Al Daas, LG, Jolivet, Tournier)

Given the multilevel preconditioner defined at each level $i = 1 : \log_d N_1$ as

$$M^{-1}_{A_i,MAS} = V_i A^{-1}_{i+1} V_i^T + \sum_{j=1}^{N_i} R_{ij}^\top A_{ij}^{-1} R_{ij}$$

then $M^{-1}_{MAS} = M^{-1}_{A_1,MAS}$ and,

$$\kappa(M^{-1}_{A_i,MAS} A_i) \leq (k_{ic} + 1)\,(2 + (2k_{ic} + 1)k_i\tau)\,,$$

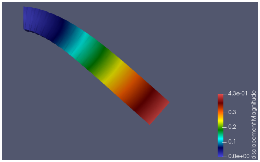
where k_{ic} = number of distinct colours to colour the graph of A,
k_i = max number of domains that share a common vertex.

Communication efficiency

- Construction of M^{-1}_{MAS} preconditioner requires $O(\log_d N_1)$ messages.
- Application of M^{-1}_{MAS} preconditioner requires $O((\log_2 N_1)^{\log_d N_1})$ messages per iteration.

Parallel performance for linear elasticity

- Machine: IRENE (Genci), Intel Skylake 8168, 2,7 GHz, 24 cores each
- Stopping criterion: 10^{-5} (10^{-2} for 3rd level)
- Young's modulus E and Poisson's ratio ν take two values, $(E_1, \nu_1) = (2 \cdot 10^{11}, 0.35)$, and $(E_2, \nu_2) = (10^7, 0.45)$

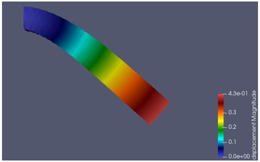


Linear elasticity, $121x10^6$ **unknowns, PETSc versus GenEO HPDDM**

# P	PETSc GAMG			HPDDM				
	PCSetUp	KSPSolve	Total	Deflation subspace	Domain factor	Coarse matrix	Solve	Total
1,024	39.9	85.7	125.7	185.8	26.8	3.0	62.0	277.7
2,048	42.1	21.2	63.3	76.1	8.5	4.2	28.5	117.3
4,096	95.1	182.8	277.9	32.0	3.6	8.5	18.1	62.4

More details in P. Jolivet's talk, MS 199, this morning

Parallel performance for linear elasticity



- Machine: IRENE (Genci), Intel Skylake 8168, 2,7 GHz, 24 cores each
- Stopping criterion: 10^{-5} (10^{-2} for 3rd level)
- Young's modulus E and Poisson's ratio ν take two values, $(E_1, \nu_1) = (2 \cdot 10^{11}, 0.35)$, and $(E_2, \nu_2) = (10^7, 0.45)$

Linear elasticity, $616 \cdot 10^6$ **unknowns, GenEO versus GenEO multilevel**

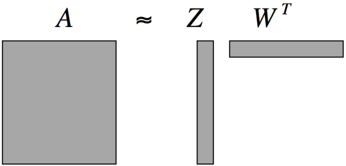
# P	Deflation subspace	Domain factor	Coarse matrix	Solve	Total	# iter
			GenEO			
8192	113.3	11.9	27.5	52.0	152.8	53
			GenEO multilevel			
8192	113.3	11.9	13.2	52.0	138.5	53

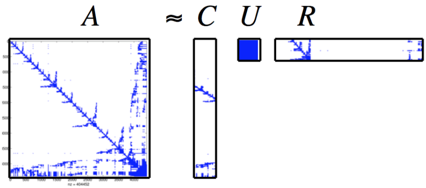
A_2 of dimension $328 \cdot 10^3 \times 328 \cdot 10^3$, A_3 of dimension 5120×5120.
More details in P. Jolivet's talk, MS 199, this morning

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 $k \times n$.



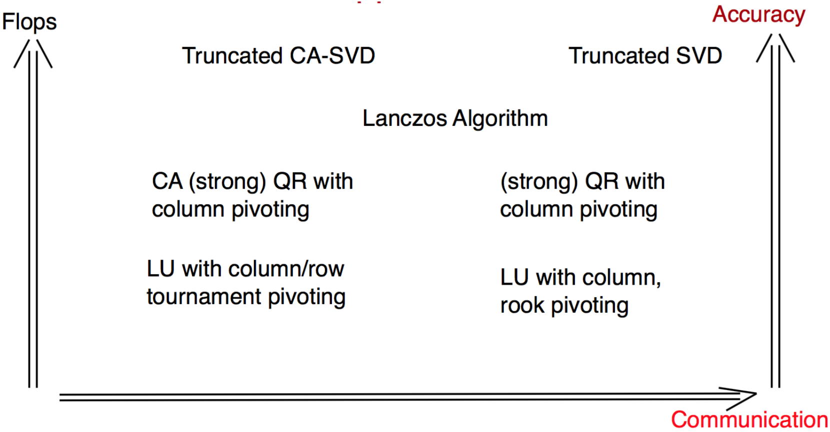


- Best rank-k approximation $A_k = U_k \Sigma_k V_k$ is rank-k truncated SVD of A [Eckart and Young, 1936]

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

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

Low rank matrix approximation: trade-offs



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

$$\# \text{ messages} = \Omega(\log P).$$

Deterministic rank-k matrix approximation

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, $U_1 \in \mathbb{R}^{l'\times m}$, $V_1 \in \mathbb{R}^{n\times l}$, $k \le l \le l'$.

$$\begin{aligned} UAV &= \bar{A} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} \\ &= \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{+} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} = U \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}, \end{aligned}$$

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

$$\tilde{A}_k = U^{-1} \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{+} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} V^{-1}$$

- QR computes the approximation

$$\tilde{A}_k = Q_1 \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} V^{-1} = Q_1 Q_1^T A, \text{ where } Q_1 \text{ is orth basis for } (AV_1)$$

Unified perspective: generalized LU factorization

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

$$\tilde{A}_k = AV_1(U_1 AV_1)^{-1} U_1 A$$

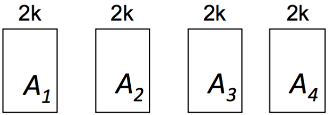
Deterministic algorithms V_1 column permutation and ...	**Randomized algorithms*** V_1 random matrix and ...
QR with column selection (a.k.a. strong rank revealing QR) $U_1 = Q_1^T$, $\tilde{A}_k = Q_1 Q_1^T A$ $\|\|R_{11}^{-1} R_{12}\|\|_{max}$ is bounded	Randomized QR (a.k.a. randomized SVD) $U_1 =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

with J. Demmel, A. Rusciano

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

Deterministic column selection: tournament pivoting

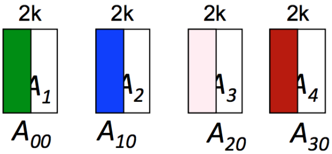
- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$,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



[Demmel, LG, Gu, Xiang 13], [LG, Cayrols, Demmel 18]

Deterministic column selection: tournament pivoting

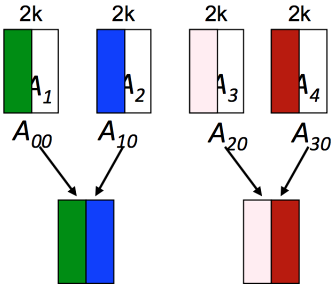
- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$,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



[Demmel, LG, Gu, Xiang 13], [LG, Cayrols, Demmel 18]

Deterministic column selection: tournament pivoting

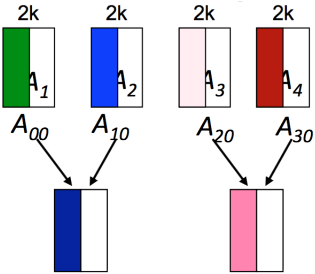
- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$,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



[Demmel, LG, Gu, Xiang 13], [LG, Cayrols, Demmel 18]

Deterministic column selection: tournament pivoting

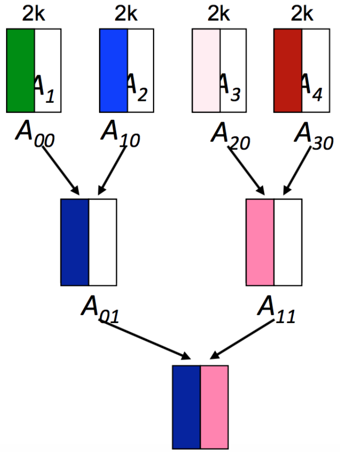
- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$,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



[Demmel, LG, Gu, Xiang 13], [LG, Cayrols, Demmel 18]

Deterministic column selection: tournament pivoting

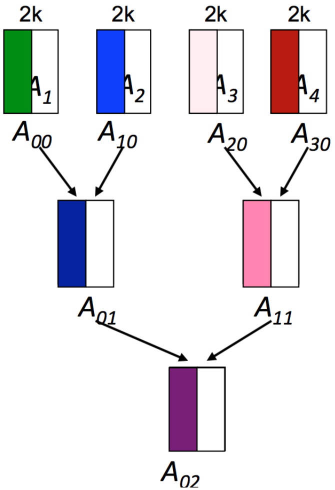
- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$,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



[Demmel, LG, Gu, Xiang 13], [LG, Cayrols, Demmel 18]

Deterministic column selection: 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}



[Demmel, LG, Gu, Xiang 13], [LG, Cayrols, Demmel 18]

Deterministic guarantees for rank-k approximation

- CA QR with column selection based on binary tree tournament pivoting:

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \sqrt{1 + F_{TP}^2(n-k)}, \quad F_{TP} \leq \frac{1}{\sqrt{2k}} (n/k)^{\log_2(\sqrt{2}fk)}$$

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

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

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

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

Deterministic guarantees for rank-k approximation

- CA QR with column selection based on binary tree tournament pivoting:

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \sqrt{1 + F_{TP}^2(n-k)}, \quad F_{TP} \leq \frac{1}{\sqrt{2k}} (n/k)^{\log_2\left(\sqrt{2}fk\right)}$$

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

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

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

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

Probabilistic guarantees

- Combine deterministic guarantees with sketching ensembles satisfying Johnson-Lindenstrauss properties $\rightarrow$ **better bounds**
- Consider $U_1 \in \mathbb{R}^{l' \times m}$, $V_1 \in \mathbb{R}^{n \times l}$ are Subsampled Randomized Hadamard Transforms (SRHT), $l' > l$.
  - Compute $\tilde{A}_k$ through generalized LU costs $O(mn \log_2 l')$ flops

Let $U_1 \in \mathbb{R}^{l' \times m}$ and $V_1 \in \mathbb{R}^{n \times l}$ be drawn from SRHT ensembles,
$l = 10\epsilon^{-1}(\sqrt{k} + \sqrt{8\log(n/\delta)})^2 \log(k/\delta)$, $l \geq log^2(n/\delta)$,
$l' = 10\epsilon^{-1}(\sqrt{l} + \sqrt{8\log(m/\delta)})^2 \log(k/\delta)$, $l' \geq log^2(m/\delta)$.
With probability $1 - 5\delta$, the **generalized LU** approximation $\tilde{A}_k$ satisfies

$$\|A - \tilde{A}_k\|_2^2 = O(1)\sigma_{k+1}^2(A) + O(\frac{\log(n/\delta)}{l} + \frac{\log(m/\delta)}{l'})(\sigma_{k+1}^2(A) + \dots \sigma_n^2(A))$$

Probabilistic guarantees

- Combine deterministic guarantees with sketching ensembles satisfying Johnson-Lindenstrauss properties $\rightarrow$ **better bounds**
- Consider $U_1 \in \mathbb{R}^{l' \times m}$, $V_1 \in \mathbb{R}^{n \times l}$ are Subsampled Randomized Hadamard Transforms (SRHT), $l' > l$.
  - Compute $\tilde{A}_k$ through generalized LU costs $O(mn \log_2 l')$ flops

Let $U_1 \in \mathbb{R}^{l' \times m}$ and $V_1 \in \mathbb{R}^{n \times l}$ be drawn from SRHT ensembles,
$l = 10\epsilon^{-1}(\sqrt{k} + \sqrt{8 \log(n/\delta)})^2 \log(k/\delta)$, $l \geq log^2(n/\delta)$,
$l' = 10\epsilon^{-1}(\sqrt{l} + \sqrt{8 \log(m/\delta)})^2 \log(k/\delta)$, $l' \geq log^2(m/\delta)$.
With probability $1 - 5\delta$, the **generalized LU** approximation $\tilde{A}_k$ satisfies

$$\|A - \tilde{A}_k\|_2^2 = O(1)\sigma_{k+1}^2(A) + O(\frac{\log(n/\delta)}{l} + \frac{\log(m/\delta)}{l'})(\sigma_{k+1}^2(A) + \ldots \sigma_n^2(A))$$

Growth factor in Gaussian elimination

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

S_k is Schur complement obtained at iteration k

Deterministic algorithms

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

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

Randomized algorithms

U, V Haar distributed matrices,

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

Plan

Motivation of our work

Short overview of results from CA dense linear algebra
- TSQR factorization

Preconditioned Krylov subspace methods
- Enlarged Krylov methods
- Robust multilevel additive Schwarz preconditioner

Unified perspective on low rank matrix approximation
- Generalized LU decomposition

Prospects for the future: tensors in high dimensions
- Hierarchical low rank tensor approximation

Conclusions

Prospects for the future: 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 ALS, DMRG, 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

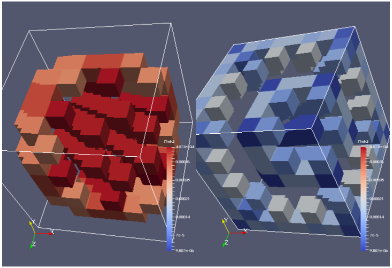
For an overview, see Kolda and Bader, SIAM Review 2009

Hierarchical low rank tensor approximation

- Decompose $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots n_d}$ in subtensors $\mathcal{A}_{1j} \in \mathbb{R}^{n_1/2 \times \dots n_d/2}$, $j = 1 : 2^d$.
- Decompose recursively each subtensor $\mathcal{A}_{1j}$ until depth L

Input: $\mathcal{A}$, 2^{Ld} subtensors $\mathcal{A}_{ij}$, $i = 1 : L$, tree T with 2^{Ld} leaves and height L
Output: $\tilde{\mathcal{A}}$ in hierarchical format
Ensure: $||\mathcal{A} - \tilde{\mathcal{A}}||_F < \varepsilon$
for each level i from L to 1 **do**
  for each node j with merge allowed **do**
    Compute $\tilde{\mathcal{A}}_{ij}$ s.t. $||\mathcal{A}_{ij} - \tilde{\mathcal{A}}_{ij}||_F < \varepsilon / 2^{di}$
    if storage($\tilde{\mathcal{A}}_{ij}$) < storage (children approx.) in T **then**
      keep $\mathcal{A}_{ij}$ approximation in $\tilde{\mathcal{A}}$
    else keep children approx. in $\tilde{\mathcal{A}}$
      merge of ancestors not allowed **endif**
  endfor
endfor

Coulomb potential, 512^3,
$V(x, y, z) = \frac{1}{|x-y|} + \frac{1}{|y-z|} + \frac{1}{|x-z|}$
hierarchical format requires 7% of storing $\mathcal{A}$ for $\varepsilon = 10^{-5}$



with V. Ehrlacher and D. Lombardi

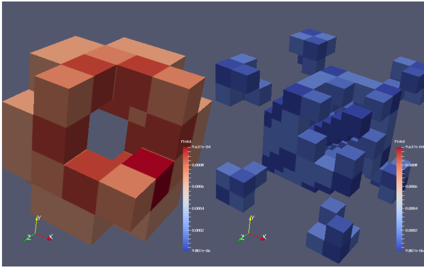
Hierarchical low rank tensor approximation

- Decompose $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots n_d}$ in subtensors $\mathcal{A}_{1j} \in \mathbb{R}^{n_1/2 \times \dots n_d/2}$, $j = 1 : 2^d$.
- Decompose recursively each subtensor $\mathcal{A}_{1j}$ until depth L

Input: $\mathcal{A}$, 2^{Ld} subtensors $\mathcal{A}_{ij}$, $i = 1 : L$, tree T with 2^{Ld} leaves and height L
Output: $\tilde{\mathcal{A}}$ in hierarchical format
Ensure: $||\mathcal{A} - \tilde{\mathcal{A}}||_F < \varepsilon$
for each level i from L to 1 **do**
  for each node j with merge allowed **do**
    Compute $\tilde{\mathcal{A}}_{ij}$ s.t. $||\mathcal{A}_{ij} - \tilde{\mathcal{A}}_{ij}||_F < \varepsilon/2^{di}$
    if storage($\tilde{\mathcal{A}}_{ij}$) $<$ storage (children approx.) in T
then
      keep $\mathcal{A}_{ij}$ approximation in $\tilde{\mathcal{A}}$
    else keep children approx. in $\tilde{\mathcal{A}}$
      merge of ancestors not allowed **endif**
  endfor
endfor

with V. Ehrlacher and D. Lombardi

Coulomb potential, 512^3,
$V(x,y,z) = \frac{1}{|x-y|} + \frac{1}{|y-z|} + \frac{1}{|x-z|}$
hierarchical format requires 7% of storing $\mathcal{A}$ for $\varepsilon = 10^{-5}$

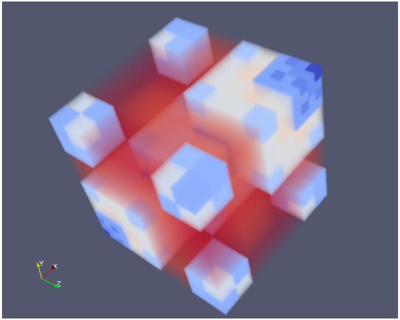


Hierarchical low rank tensor approximation

- Decompose $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots n_d}$ in subtensors $\mathcal{A}_{1j} \in \mathbb{R}^{n_1/2 \times \dots n_d/2}$, $j = 1 : 2^d$.
- Decompose recursively each subtensor $\mathcal{A}_{1j}$ until depth L

Input: $\mathcal{A}$, 2^{Ld} subtensors $\mathcal{A}_{ij}$, $i = 1 : L$, tree T with 2^{Ld} leaves and height L
Output: $\tilde{\mathcal{A}}$ in hierarchical format
Ensure: $||\mathcal{A} - \tilde{\mathcal{A}}||_F < \varepsilon$
for each level i from L to 1 **do**
  for each node j with merge allowed **do**
    Compute $\tilde{\mathcal{A}}_{ij}$ s.t. $||\mathcal{A}_{ij} - \tilde{\mathcal{A}}_{ij}||_F < \varepsilon/2^{di}$
    if storage($\tilde{\mathcal{A}}_{ij}$) $<$ storage (children approx.) in T **then**
      keep $\mathcal{A}_{ij}$ approximation in $\tilde{\mathcal{A}}$
    else keep children approx. in $\tilde{\mathcal{A}}$
      merge of ancestors not allowed **endif**
  endfor
endfor

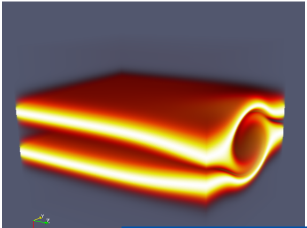
Coulomb potential, 512^3,
$V(x, y, z) = \frac{1}{|x-y|} + \frac{1}{|y-z|} + \frac{1}{|x-z|}$
hierarchical format requires 7% of storing $\mathcal{A}$ for $\varepsilon = 10^{-5}$

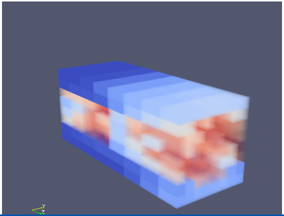


with V. Ehrlacher and D. Lombardi

Compressing the solution of Vlasov-Poisson equation

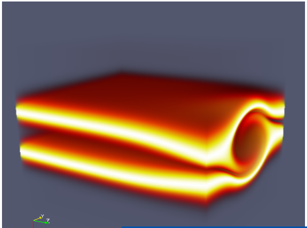
- Hierarchical tensors in the spirit of hierarchical matrices (Hackbusch et al), but no information on the represented function required. Speed, velocity, time 512x256x160, compression factor of 350 for $\varepsilon = 10^{-3}$.

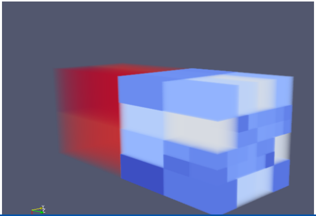




Compressing the solution of Vlasov-Poisson equation

- Hierarchical tensors in the spirit of hierarchical matrices (Hackbusch et al), but no information on the represented function required. Speed, velocity, time 512x256x160, compression factor of 350 for $\varepsilon = 10^{-3}$.





Plan

Conclusions

Most of the methods discussed available in libraries:

- Dense CA linear algebra
  - progressively in LAPACK/ScaLAPACK and some vendor libraries
- Iterative methods:
  preAlps library `https://github.com/NLAFET/preAlps`:
  - Enlarged CG: Reverse Communication Interface
  - Enlarged GMRES will be available as well
- Multilevel Additive Schwarz
  - will be available in HPDDM as multilevel Geneo (P. Jolivet)

Acknowledgements

- NLAFET H2020 european project, ANR
- Total

Prospects for the future

- Multilevel Additive Schwarz
  - from theory to practice, find an efficient local algebraic splitting that allows to solve the Gen. EVP locally on each processor
- 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: G. Ballard, S. Cayrols, H. Al Daas, J. Demmel, M. Hoemmen, P. Jolivet, N. Knight, S. Moufawad, F. Nataf, D. Nguyen, J. Langou, E. Solomonik, A. Rusciano, P. H. Tournier, O. Tissot.

References (1)

Ashby, S. F., Manteuffel, T. A., and Saylor, P. E. (1990).
A taxonomy for conjugate gradient methods.
SIAM Journal on Numerical Analysis, 27(6):1542–1568.

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

Grigori, L., Moufawad, S., and Nataf, F. (2016).
Enlarged Krylov Subspace Conjugate Gradient Methods for Reducing Communication.
SIAM Journal on Scientific Computing, 37(2):744–773.
Also as INRIA TR 8266.

Hestenes, M. R. and Stiefel, E. (1952).
Methods of conjugate gradients for solving linear systems.
Journal of research of the National Bureau of Standards., 49:409–436.

OLeary, D. P. (1980).
The block conjugate gradient algorithm and related methods.
Linear Algebra and Its Applications, 29:293–322.