Mixed Precision Algorithms for High Performance Scientific Computing

Today (MS170)

2:15 PM	Theo Mary	Mixed Precision Low Rank Compression and its Application to BLR Matrix Factorization
2:35 PM	Hatem Ltaief	Tile-Centric Mixed Precision Computations for HPC Applications
2:55 PM	Florent Lopez	Mixed Precision LU Factorization using GPU Tensor Cores
3:15 PM	Françoise Tisseur	Mixed Precision Cholesky-QR Algorithm with Applications
3:35 PM	Hiroyuki Ootomo	TSQR on Tensor Cores with Error Correction

Tomorrow (MS233)

9:45 AM	Srikara Pranesh	Three-Precision GMRES-Based Iterative Refinement for Least Squares Problems
10:05 AM	Azzam Haidar	How NVIDIA Tensor Cores can Help HPC Scientific Application Unleash the Power of GPUs using Mixed Precision Solvers
10:25 AM	Bastien Vieublé	Iterative Refinement in up to Five Precisions for the Solution of Large Linear Systems
10:45 AM	Thomas Gruetzmacher	Compressed Basis GMRES on High Performance GPUs
11:05 AM	Daichi Mukunoki	DGEMM using Tensor Cores

SIAM CSE 2021

March 3, 2021

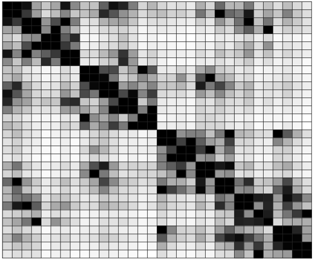
Mixed Precision Low Rank Compression and its Application to BLR Matrix Factorization

Theo Mary

Sorbonne Université, CNRS, LIP6

https://www-pequan.lip6.fr/~tmary/

Slides available at https://bit.ly/cse21mixLR



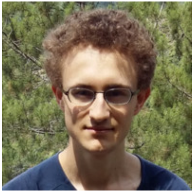
Joint work with

Patrick Amestoy

Olivier Boiteau

Alfredo Buttari

Matthieu Gerest



Fabienne Jézéquel

Jean-Yves L'Excellent

Low precision arithmetics

	Bits Signif. (t)	Exp.	Range	$u = 2^{-t}$
bfloat16	8	8	$10^{\pm 38}$	4×10^{-3}
fp16	11	5	$10^{\pm 5}$	5×10^{-4}
fp32	24	8	$10^{\pm 38}$	6×10^{-8}
fp64	53	11	$10^{\pm 308}$	1×10^{-16}
fp128	113	15	$10^{\pm 4932}$	1×10^{-34}

Half precision increasingly supported by hardware:

- Fp16 used by NVIDIA GPUs, AMD Radeon Instinct MI25 GPU, ARM NEON, Fujitsu A64FX ARM
- Bfloat16 used by Google TPU, NVIDIA GPUs, Arm, Intel

Low precision arithmetics

	Bits			
	Signif. (t)	Exp.	Range	$u = 2^{-t}$
bfloat16	8	8	$10^{\pm 38}$	4×10^{-3}
fp16	11	5	$10^{\pm 5}$	5×10^{-4}
fp32	24	8	$10^{\pm 38}$	6×10^{-8}
fp64	53	11	$10^{\pm 308}$	1×10^{-16}
fp128	113	15	$10^{\pm 4932}$	1×10^{-34}

Great benefits:

- Reduced **storage**, data movement, and communications
- Increased **speed**, e.g., with GPU Tensor Cores:
  fp32 $\rightarrow$ fp16 speedup evolution:
  P100: 2× V100: 8× A100: 16×
- Reduced **energy** consumption ($5\times$ with fp16, $9\times$ with bfloat16!)

Mixed precision algorithms

- **Low precision $\Rightarrow$ correspondingly low accuracy !**

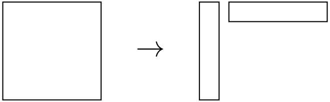
- Mixed precision algorithms: combine low and high precisions strategically
  - Better performance than high prec. algs.
  - Better accuracy and/or robustness than low prec. algs.

- Mixed precision algs. highly successful in NLA / HPC
  - This MS and its part II (MS233, tomorrow 9:45 AM)
  - MS171 and MS200 (today, 2:15–5:55 PM)
  - Three talks in MS21 (Monday)

- This talk: mixed precision low rank approximations

Low rank compression

$$\begin{matrix} A & \approx & X & Y^T \\ n \times n & & n \times r & r \times n \end{matrix}$$



- ε-rank of A:

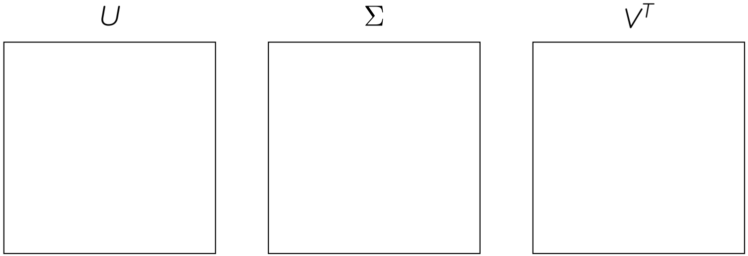
  smallest r_ε such that $\exists T, \quad \text{rank}(T) = r_\varepsilon, \quad \|A - T\| \le \varepsilon \|A\|$

- Optimal ε-approximation given by truncated SVD (Eckart–Young)

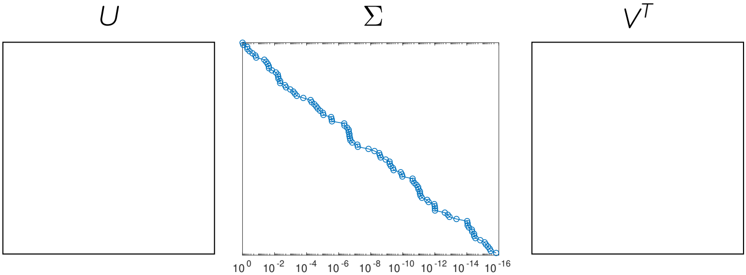
$$A = U \Sigma V^T \quad \Rightarrow \quad T = U_\varepsilon \Sigma_\varepsilon V_\varepsilon^T = \sum_{i=1}^{r_\varepsilon} u_i \sigma_i v_i^T$$

- **What precision should we store T in ?**
- Naive answer: a precision with unit roundoff safely smaller than ε (e.g., fp64 if $\varepsilon < u_{\text{fp32}} \approx 6 \times 10^{-8}$)

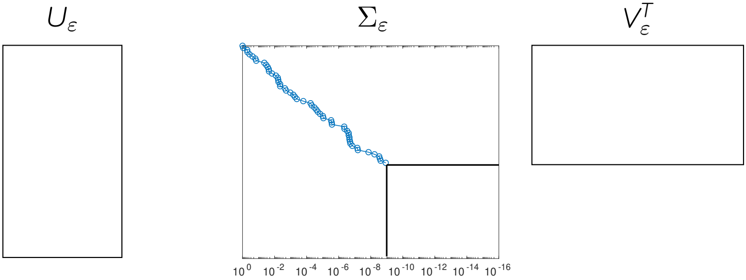
Mixed precision SVD: an example



Mixed precision SVD: an example

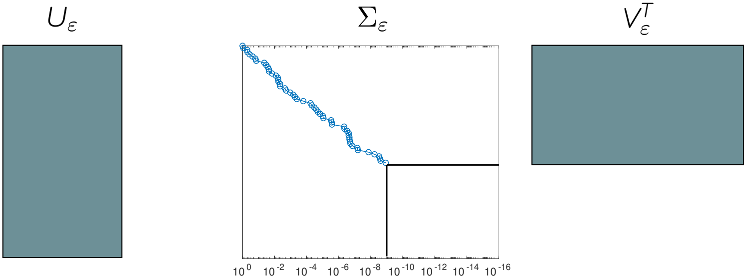


Mixed precision SVD: an example

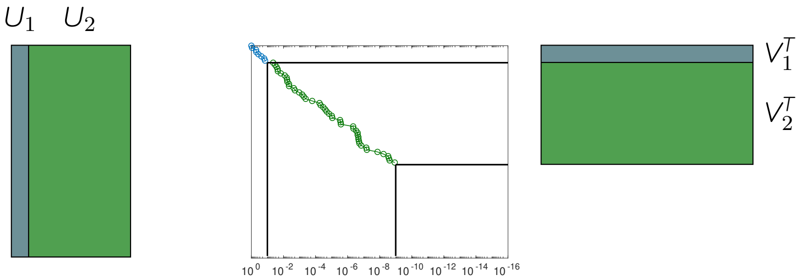


- Assume $\varepsilon = 10^{-9} \Rightarrow \|A - U_\varepsilon \Sigma_\varepsilon V_\varepsilon^T\| \leq \varepsilon \|A\|$

Mixed precision SVD: an example

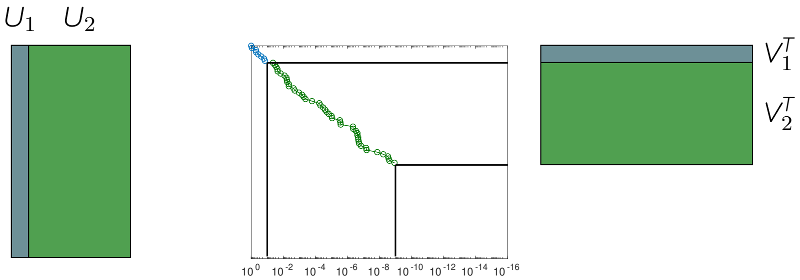


- Assume $\varepsilon = 10^{-9} \Rightarrow \|A - U_\varepsilon \Sigma_\varepsilon V_\varepsilon^T\| \le \varepsilon \|A\|$
- Naive approach: use **double precision** because $u_{\text{fp32}} > \varepsilon$



- Assume $\varepsilon = 10^{-9} \Rightarrow \|A - U_\varepsilon \Sigma_\varepsilon V_\varepsilon^T\| \le \varepsilon \|A\|$
- Naive approach: use **double precision** because $u_{\mathrm{fp32}} > \varepsilon$
- Our idea: let $U_\varepsilon = [U_1 \; U_2]$, $\Sigma_\varepsilon = \mathrm{diag}(\Sigma_1, \Sigma_2)$, and $V_\varepsilon = [V_1 \; V_2]$. Converting U_2 and V_2 to **single precision** introduces an error of order $u_{\mathrm{fp32}} \, \|\Sigma_2\|$

Mixed precision SVD: an example



- Assume $\varepsilon = 10^{-9} \Rightarrow \|A - U_\varepsilon \Sigma_\varepsilon V_\varepsilon^T\| \le \varepsilon \|A\|$
- Naive approach: use **double precision** because $u_{\mathrm{fp32}} > \varepsilon$
- Our idea: let $U_\varepsilon = [U_1 \ U_2]$, $\Sigma_\varepsilon = \mathrm{diag}(\Sigma_1, \Sigma_2)$, and $V_\varepsilon = [V_1 \ V_2]$. Converting U_2 and V_2 to **single precision** introduces an error of order $u_{\mathrm{fp32}} \|\Sigma_2\|$

$\Rightarrow$ Need to partition Σ such that $\|\Sigma_2\| \le \varepsilon / u_{\mathrm{fp32}} \approx 2 \times 10^{-2}$

Error analysis

- Can use any number of precisions $u_1 \le \varepsilon < u_2 < \ldots < u_p$
- Partition the SVD into p groups $U_k \Sigma_k V_k$ such that

$$\|\Sigma_k\| \le \varepsilon \|A\| / u_k$$

and let $\widehat{U}_k$ and $\widehat{V}_k$ be stored in precision u_k.

Error analysis

- Can use any number of precisions $u_1 \leq \varepsilon < u_2 < \ldots < u_p$
- Partition the SVD into p groups $U_k \Sigma_k V_k$ such that

$$\|\Sigma_k\| \leq \varepsilon \|A\| / u_k$$

  and let $\widehat{U}_k$ and $\widehat{V}_k$ be stored in precision u_k.
- Then

$$\|U_k \Sigma_k V_k^T - \widehat{U}_k \Sigma_k \widehat{V}_k^T\| \leq (2u_k + u_k^2)\|\Sigma_k\| \leq (2 + u_k)\varepsilon\|A\|$$

  and so
$$\boxed{\|A - \widehat{T}\| \leq \Big(2p - 1 + \sum_{k=2}^{p} u_k\Big)\varepsilon\|A\| = O(\varepsilon)\|A\|}$$

Error analysis

- Can use any number of precisions $u_1 \leq \varepsilon < u_2 < \ldots < u_p$
- Partition the SVD into p groups $U_k \Sigma_k V_k$ such that

$$\|\Sigma_k\| \leq \varepsilon \|A\| / u_k$$

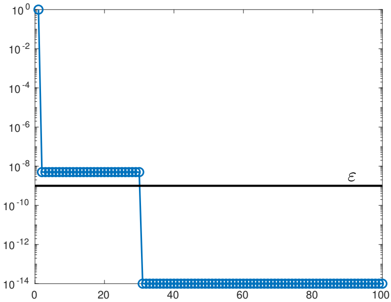
  and let $\widehat{U}_k$ and $\widehat{V}_k$ be stored in precision u_k.
- Then

$$\|U_k \Sigma_k V_k^T - \widehat{U}_k \Sigma_k \widehat{V}_k^T\| \leq (2u_k + u_k^2)\|\Sigma_k\| \leq (2 + u_k)\varepsilon \|A\|$$

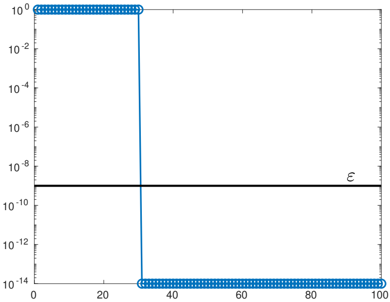
  and so
$$\boxed{\|A - \widehat{T}\| \leq \Big(2p - 1 + \sum_{k=2}^{p} u_k\Big)\varepsilon \|A\| = O(\varepsilon)\|A\|}$$

- Applicable to any low rank matrix $XY^T = \sum_{i=1}^{r_\varepsilon} x_i y_i^T$ with decaying $\|x_i y_i^T\|$. Example: $AP \approx Q_\varepsilon R_\varepsilon = Q_1 R_1 + \ldots + Q_p R_p$

Both matrices have ε-rank 30 (with $\varepsilon = 10^{-9}$) but present very different potential for mixed precision

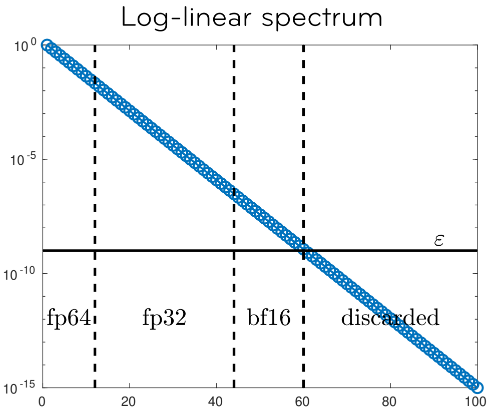


Large gain
(almost all in lower precision)



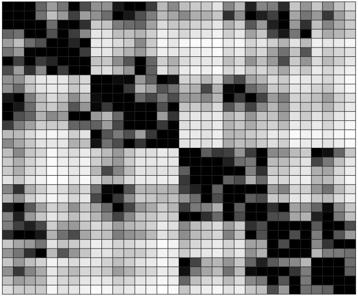
No gain
(all in higher precision)

Examples of spectrum



BLR matrices

Block low rank (BLR) matrices use a flat 2D block partitioning

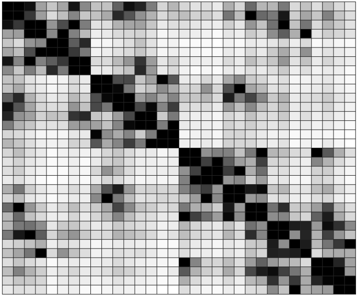


Example of a BLR matrix (Schur complement of a 64^3 Poisson problem with block size 128)

- Diagonal blocks are full rank
- Off-diagonal blocks A_{ij} are approximated by low-rank blocks T_{ij} satisfying $\|A_{ij} - T_{ij}\| \leq \varepsilon \|A\|$ (**global compression**)
- $\varepsilon = 10^{-15} \rightarrow$ 50% entries kept

BLR matrices

Block low rank (BLR) matrices use a flat 2D block partitioning

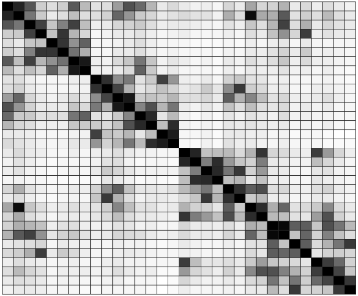


Example of a BLR matrix (Schur complement of a 64^3 Poisson problem with block size 128)

- Diagonal blocks are full rank
- Off-diagonal blocks A_{ij} are approximated by low-rank blocks T_{ij} satisfying $\|A_{ij} - T_{ij}\| \le \varepsilon \|A\|$ (**global compression**)
- $\varepsilon = 10^{-15} \rightarrow$ 50% entries kept
- $\varepsilon = 10^{-12} \rightarrow$ 36% entries kept

BLR matrices

Block low rank (BLR) matrices use a flat 2D block partitioning

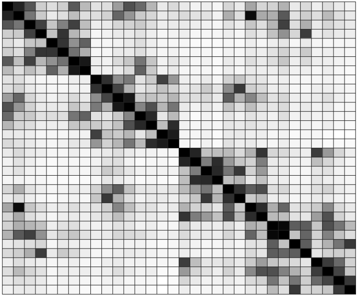


Example of a BLR matrix (Schur complement of a 64^3 Poisson problem with block size 128)

- Diagonal blocks are full rank
- Off-diagonal blocks A_{ij} are approximated by low-rank blocks T_{ij} satisfying $\|A_{ij} - T_{ij}\| \leq \varepsilon \|A\|$ (**global compression**)
- $\varepsilon = 10^{-15} \rightarrow$ 50% entries kept
- $\varepsilon = 10^{-12} \rightarrow$ 36% entries kept
- $\varepsilon = 10^{-9} \rightarrow$ 23% entries kept

BLR matrices

Block low rank (BLR) matrices use a flat 2D block partitioning

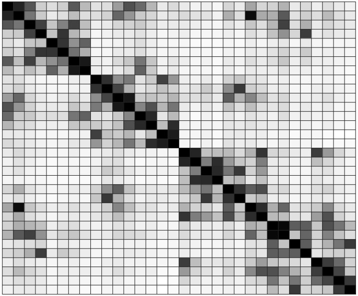


Example of a BLR matrix (Schur complement of a 64^3 Poisson problem with block size 128)

- Diagonal blocks are full rank
- Off-diagonal blocks A_{ij} are approximated by low-rank blocks T_{ij} satisfying $\|A_{ij} - T_{ij}\| \le \varepsilon \|A\|$ (**global compression**)
- $\varepsilon = 10^{-15} \rightarrow$ 50% entries kept
- $\varepsilon = 10^{-12} \rightarrow$ 36% entries kept
- $\varepsilon = 10^{-9} \rightarrow$ 23% entries kept
- Rapid decay $\Rightarrow$ high potential for mixed precision compression

BLR matrices

Block low rank (BLR) matrices use a flat 2D block partitioning



Example of a BLR matrix (Schur complement of a 64^3 Poisson problem with block size 128)

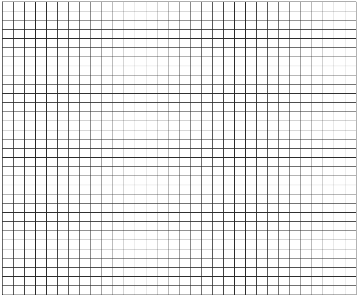
- Diagonal blocks are full rank
- Off-diagonal blocks A_{ij} are approximated by low-rank blocks T_{ij} satisfying $\|A_{ij} - T_{ij}\| \le \varepsilon \|A\|$ (**global compression**)
- $\varepsilon = 10^{-15} \rightarrow$ 50% entries kept
- $\varepsilon = 10^{-12} \rightarrow$ 36% entries kept
- $\varepsilon = 10^{-9} \rightarrow$ 23% entries kept
- Rapid decay $\Rightarrow$ high potential for mixed precision compression

Hierarchical data sparse matrices ($\mathcal{H}$, HSS, ...) not covered in this talk, but could also benefit from mixed precision

Cf. talk of J.-Y. L'Excellent (MS343, Friday, 10:40 AM)

Mixed precision BLR matrices

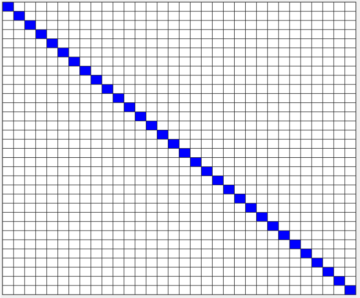
(Poisson, $\varepsilon = 10^{-10}$)



With two precisions (**double + single**):

Mixed precision BLR matrices

(Poisson, $\varepsilon = 10^{-10}$)

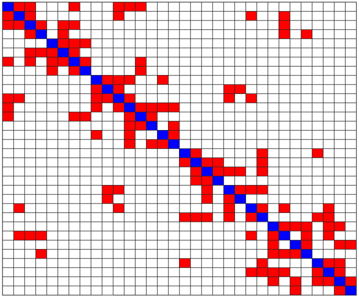


With two precisions (**double + single**):

- **double**

Mixed precision BLR matrices

(Poisson, $\varepsilon = 10^{-10}$)

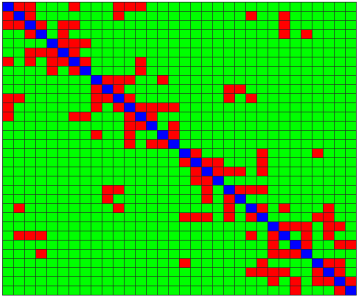


With two precisions (**double + single**):

- **double**
- **double/single**

Mixed precision BLR matrices

(Poisson, $\varepsilon = 10^{-10}$)



With two precisions (**double + single**):

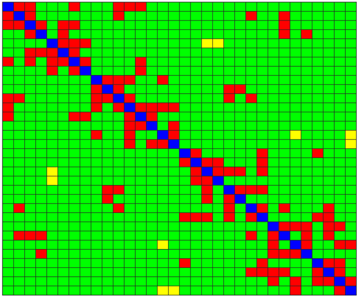
- **double**
- **double/single**
- **single**

- If $\|A_{ij}\| \leq \varepsilon\|A\|/u_{\text{fp32}}$, block can be stored entirely in single, no need for double

$\Rightarrow$ Without LR approximations, mixed precision can be still be used (Abdulah et al.)

Mixed precision BLR matrices

(Poisson, $\varepsilon = 10^{-10}$)



With three precisions (**double + single + half**):

- **double**
- **double/single/half**
- **single/half**
- **half**

- If $\|A_{ij}\| \leq \varepsilon \|A\| / u_{\text{fp32}}$, block can be stored entirely in single, no need for double

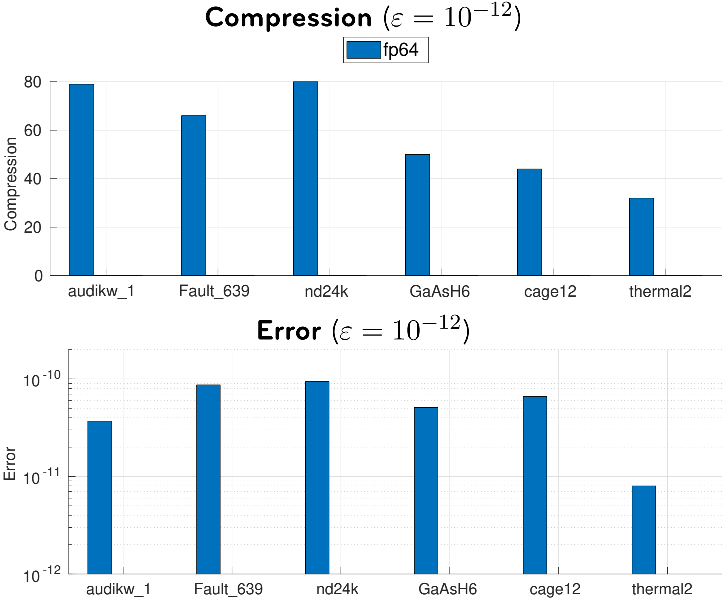
$\Rightarrow$ Without LR approximations, mixed precision can be still be used (Abdulah et al.)

Test matrices

- Dense matrices obtained from the root separator (Schur complement) of sparse matrices

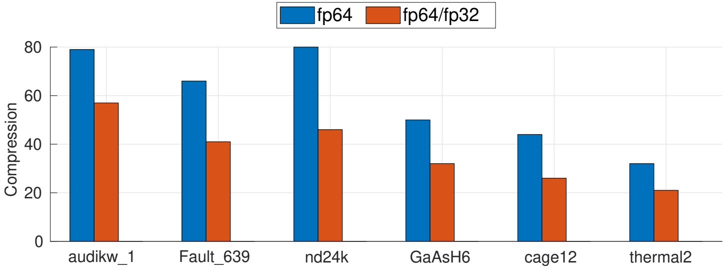
Matrix	Application	n
audikw_1	Structural	3768
Fault_639	Structural	7983
nd24k	2D/3D	7785
GaAsH6	Chemistry	6232
cage12	Graph	7323
thermal2	Thermal	1382

- Block size is set to 128

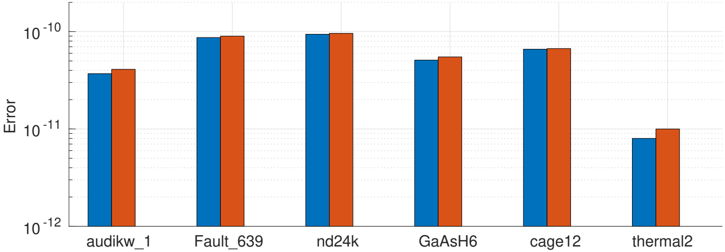

Compression ($\varepsilon = 10^{-12}$)
fp64
Compression
0
20
40
60
80
audikw_1
Fault_639
nd24k
GaAsH6
cage12
thermal2
Error ($\varepsilon = 10^{-12}$)
Error
10^{-10}
10^{-11}
10^{-12}
audikw_1
Fault_639
nd24k
GaAsH6
cage12
thermal2

Numerical results

Compression ($\varepsilon = 10^{-12}$)

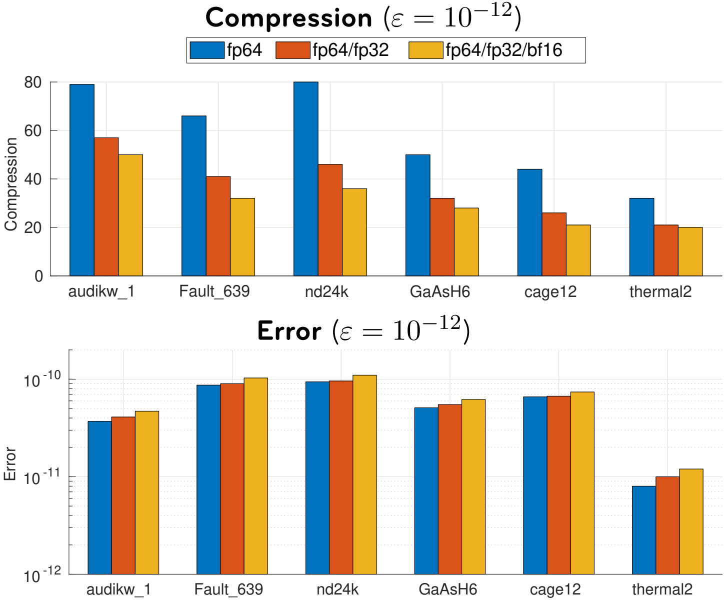


Error ($\varepsilon = 10^{-12}$)



Up to $1.7\times$ storage reduction with almost no error increase

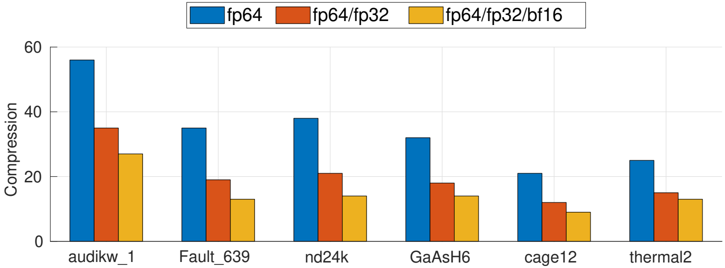
Numerical results



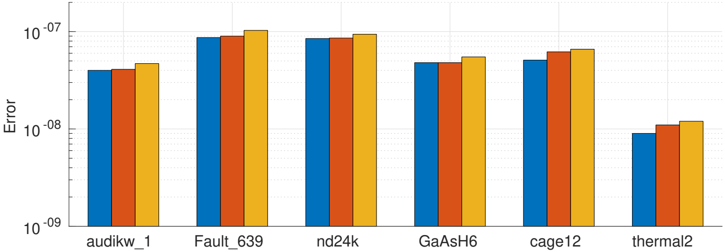
Up to $2.2\times$ storage reduction with almost no error increase

Numerical results

Compression ($\varepsilon = 10^{-9}$)



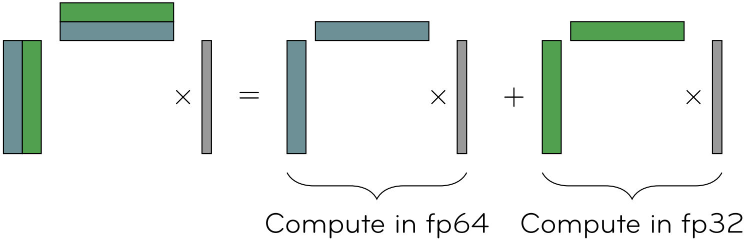
Error ($\varepsilon = 10^{-9}$)



Up to $2.7\times$ storage reduction with almost no error increase

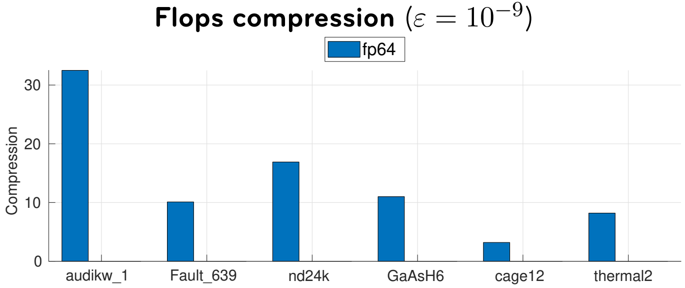
Mixed precision BLR factorization

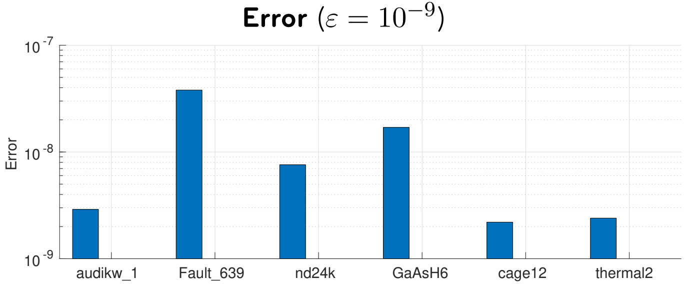
- Mixed precision can also be used to accelerate the BLR factorization. Example: multiplication of a mixed precision low rank matrix with a vector:



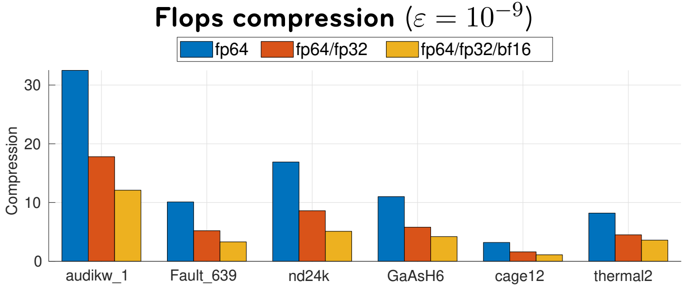
- Other operations of the BLR factorization can also be accelerated
- Full details (and rigorous error analysis) in forthcoming preprint

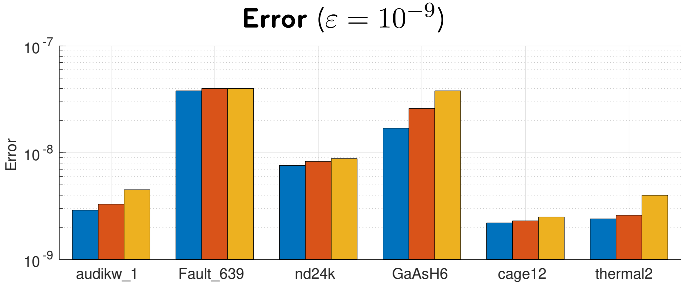
Preliminary flops results (no timings yet)





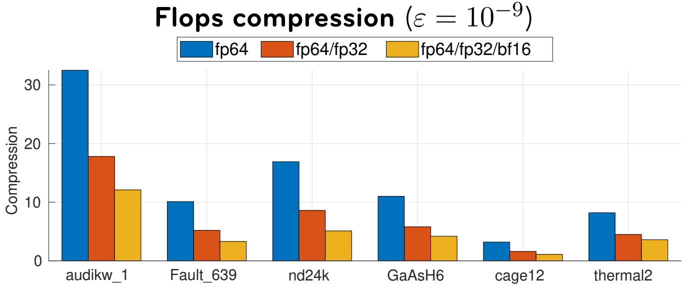
Preliminary flops results (no timings yet)

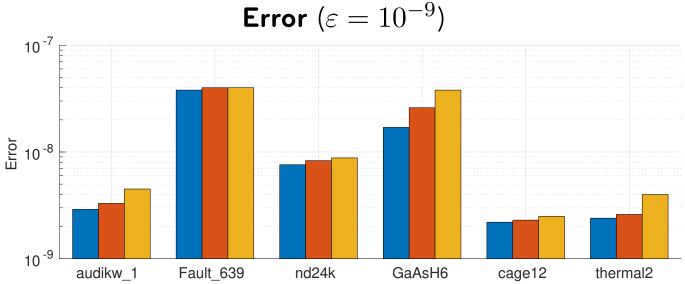




Up to $3.3\times$ flops reduction with almost no error increase

Preliminary flops results (no timings yet)





Up to $3.3\times$ flops reduction with almost no error increase

$\to 3.3\times$ time reduction??

Conclusions

Mixed precision low rank compression

- Given a matrix A and a target accuracy ε, in what precision should we represent A?
- Naive answer: a precision with unit roundoff less than ε
- Our answer: **it depends on its singular values!**

$\Rightarrow$ If rapidly decaying, lower precisions can be used

- Also applicable to QR and many other low rank decompositions

Mixed precision BLR factorization

- BLR matrices are an ideal application
- Achieved up to $2.7\times$ storage reduction with fp64/fp32/bfloat16
- Can also accelerate factorization, up to $3.3\times$ flops reduction

$\Rightarrow$ Much work still needed to transform flops into time reduction!

Slides available at `https://bit.ly/cse21mixLR`