A New Preconditioner for Low-Accuracy Block Low-Rank Multifrontal Solvers

Theo Mary, joint work with Nick Higham
University of Manchester, School of Mathematics

Sparse Days, Toulouse, 27-28 September 2018

M\cr NA
Manchester Numerical Analysis

Context

Objective

- Compute solution to linear system $Ax = b$
- $A \in \mathbb{R}^{n \times n}$ is ill conditioned

LU-based preconditioner

1. Compute approximate factorization $A = \widehat{L}\widehat{U} + \Delta A$
   - Half-precision factorization
   - Incomplete LU factorization
   - Structured matrix factorization: Block Low-Rank, $\mathcal{H}$, HSS,...
2. Solve $\Pi_{LU} Ax = \Pi_{LU} b$ with $\Pi_{LU} = \widehat{U}^{-1}\widehat{L}^{-1}$ via some iterative method

- Convergence to solution may be slow or fail

$\Rightarrow$ **Objective: accelerate convergence**

Outline

1. **A new preconditioner for approximate factorizations**

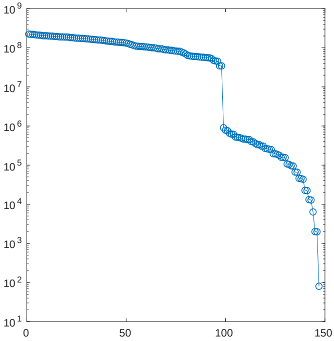
   N. J. Higham and T. Mary, *A New Preconditioner that Exploits Low-Rank Approximations to Factorization Error*, MIMS EPrint 2018.10.

2. **Application to low-accuracy BLR multifrontal solvers**

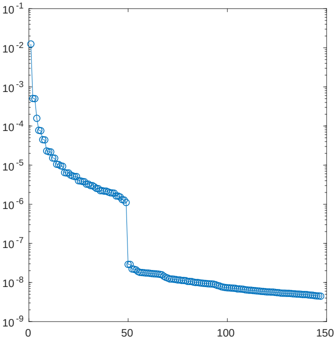
A new preconditioner for approximate factorizations

Key observation

Matrix lund_a ($n = 147$, $\kappa(A) = 2.8e{+}06$)



SVD of A



SVD of A^{-1}

- Often, A is ill conditioned due to a small number of small singular values
- Then, A^{-1} is numerically low-rank

Key idea

Factorization error might be low-rank?

Let the error $E = \widehat{U}^{-1}\widehat{L}^{-1}A - I = \widehat{U}^{-1}\widehat{L}^{-1}(\widehat{L}\widehat{U} + \Delta A) - I$

$$= \widehat{U}^{-1}\widehat{L}^{-1}\Delta A \approx A^{-1}\Delta A$$

Does E retain the low-rank property of A^{-1}?

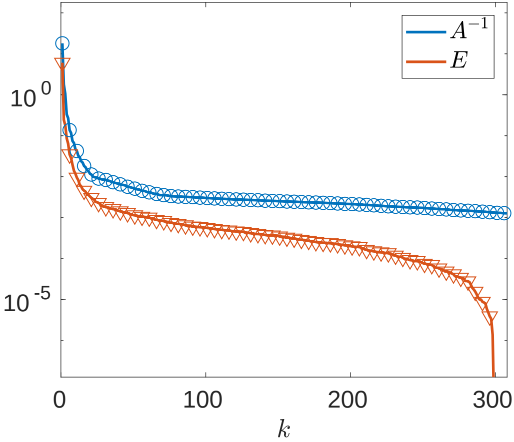
A novel preconditioner

Consider the preconditioner

$$\Pi_{E_k} = (I + E_k)^{-1}\Pi_{LU}$$

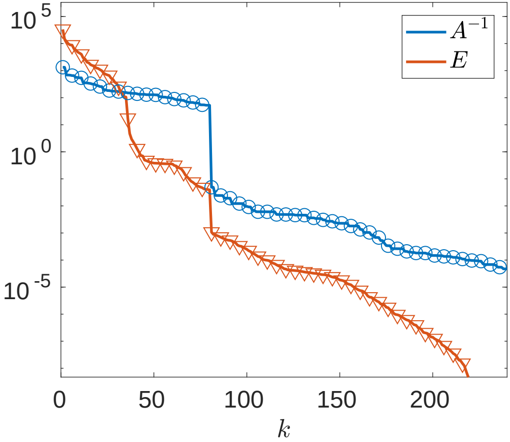
with E_k a rank-k approximation to E.

- If $E = E_k$, $\Pi_{E_k} = A^{-1}$
- If $E \approx E_k$ for some small k, Π_{E_k} can be computed cheaply



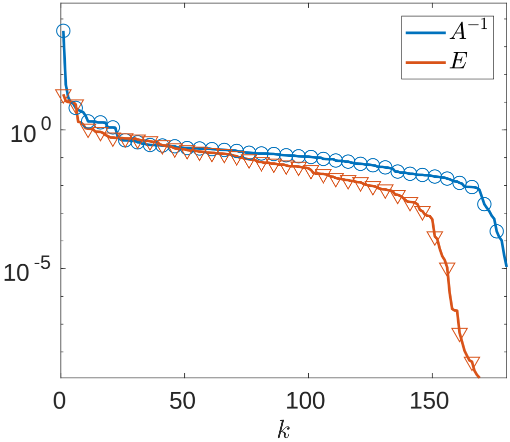
Matrix cz308

Typical SV distributions of A^{-1} and E



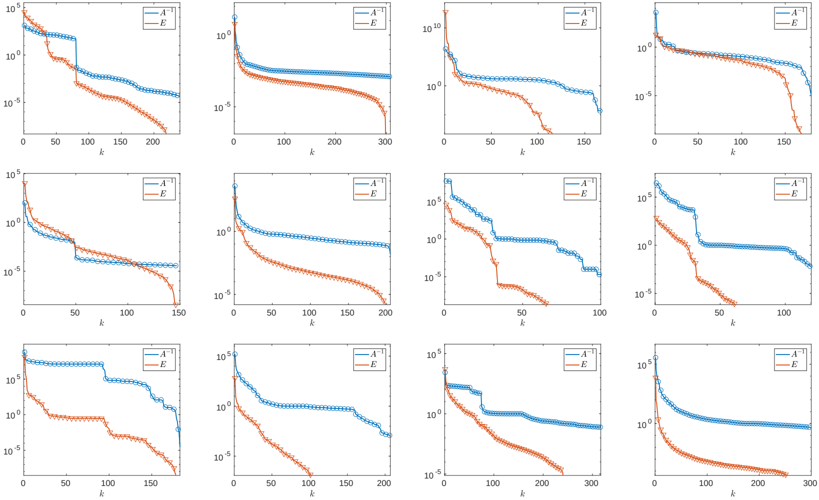
Matrix steam1

Typical SV distributions of A^{-1} and E



Matrix rajat14

Typical SV distributions of A^{-1} and E



We did **not** specifically select matrices for which A^{-1} is low-rank!

Computing E_k

We need to compute a rank-k approximation of

$$E = \widehat{U}^{-1}\widehat{L}^{-1}A - I$$

E cannot be built explicitly! $\Rightarrow$ use **randomized** method

---

Algorithm 1 Randomized SVD via direct SVD of $V^T E$.

1: Sample E: $S = E\Omega$, with Ω a $n \times (k+p)$ random matrix.
2: Orthonormalize S: $V = \text{qr}(S)$. $\{\Rightarrow E \approx VV^T E.\}$
3: Compute truncated SVD $V^T E \approx X_k \Sigma_k Y_k^T$.
4: $E_k \approx (VX_k)\Sigma_k Y_k^T$.

Preliminary experiments on small matrices

- Three types of approximate LU factorization:
  - Half-precision
  - Incomplete LU with drop tolerance $10^{-5} \le \tau \le 10^{-1}$
  - Block Low-Rank with low-rank threshold $10^{-9} \le \tau \le 10^{-1}$

Preliminary experiments on small matrices

- Three types of approximate LU factorization:
  - Half-precision
  - Incomplete LU with drop tolerance $10^{-5} \leq \tau \leq 10^{-1}$
  - Block Low-Rank with low-rank threshold $10^{-9} \leq \tau \leq 10^{-1}$

- Iterative solver is GMRES-based iterative refinement (Carson & Higham, 2017, 2018) with three precisions
  - FP64 working precision and residual is computed in FP128
  - Max nb of GMRES iterations per IR step is 100
  - Max nb of IR steps is 10

Preliminary experiments on small matrices

- Three types of approximate LU factorization:
  - Half-precision
  - Incomplete LU with drop tolerance $10^{-5} \leq \tau \leq 10^{-1}$
  - Block Low-Rank with low-rank threshold $10^{-9} \leq \tau \leq 10^{-1}$
- Iterative solver is GMRES-based iterative refinement (Carson & Higham, 2017, 2018) with three precisions
  - FP64 working precision and residual is computed in FP128
  - Max nb of GMRES iterations per IR step is 100
  - Max nb of IR steps is 10
- Large set of real-life but small matrices
  - $53 \leq n \leq 494$ and $10^3 \leq \kappa(A) \leq 10^{14}$
  - Most come from SuiteSparse collection, but treated as dense
  - 149 tests on 40 different matrices

Preliminary experiments on small matrices

- Three types of approximate LU factorization:
  - Half-precision
  - Incomplete LU with drop tolerance $10^{-5} \le \tau \le 10^{-1}$
  - Block Low-Rank with low-rank threshold $10^{-9} \le \tau \le 10^{-1}$

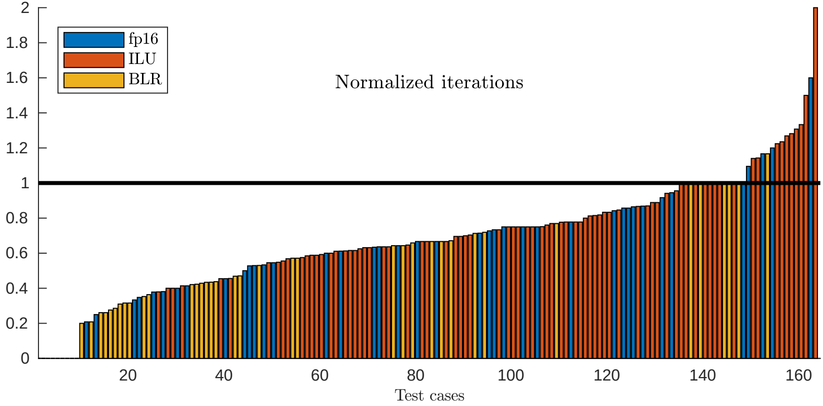
- Iterative solver is GMRES-based iterative refinement (Carson & Higham, 2017, 2018) with three precisions
  - FP64 working precision and residual is computed in FP128
  - Max nb of GMRES iterations per IR step is 100
  - Max nb of IR steps is 10

- Large set of real-life but small matrices
  - $53 \le n \le 494$ and $10^3 \le \kappa(A) \le 10^{14}$
  - Most come from SuiteSparse collection, but treated as dense
  - 149 tests on 40 different matrices

- MATLAB code running on laptop
  - We only measure number of iterations

Results with black-box setting

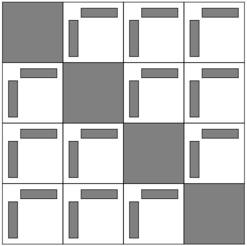
Black-box setting: use $p = 10$ and $\varepsilon = 10^{-7}$



Application to low-accuracy BLR multifrontal solvers

BLR matrices

Key principle: build approximated factorization $\mathbf{A}_\varepsilon = \mathbf{L}_\varepsilon \mathbf{U}_\varepsilon$ at accuracy ε controlled by the user



Each off-diagonal block B is approximated by a low-rank matrix $\widetilde{B}$:

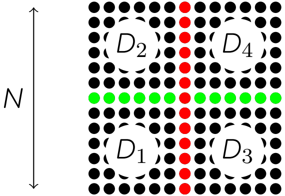
$$\|B - \widetilde{B}\| \le \varepsilon \text{ with } \operatorname{rank}(\widetilde{B}) = k_\varepsilon$$

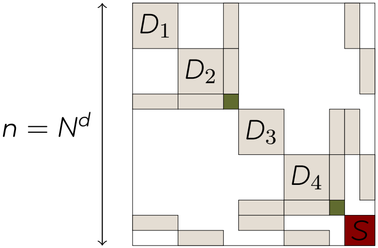
If $k_\varepsilon \ll \text{size}(B) \Rightarrow$ memory and flops can be reduced with a controlled loss of accuracy ($\le \varepsilon$)

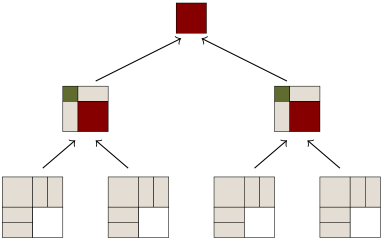
Block Low-Rank (BLR) matrix

Applicative contexts: integral equations, discretized PDEs, covariance matrices, ...

Multifrontal factorization with nested dissection







Two operations:

- Partial factorization of fronts
- Assembly of contribution blocks

Low-accuracy BLR solver: classical preconditioner

Results with the BLR-MUMPS solver
Time includes preconditioner setup (factorization) and iterative solve with GMRES (with relative stopping tolerance 10^{-9})

Matrix	n	Time (s)		Storage (GB)	
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-8}$
audikw_1	1.0M	1163	69	5	10
Bump_2911	2.9M	–	282	34	56
Emilia_923	0.9M	304	63	7	12
Fault_639	0.6M	–	45	5	9
Ga41As41H72	0.3M	–	76	12	17
Hook_1498	1.5M	902	75	6	11
Si87H76	0.2M	–	62	10	14

Low-accuracy BLR solvers:
- ☹ are slower and less robust
- ☺ but require much less storage

Results with new preconditioner

Results for $\varepsilon = 10^{-2}$:

Matrix	Π_{LU}		Π_{E_k}	
	Iter.	Time	Iter.	Time
audikw_1	691	1163	331	625
Bump_2911	–	–	284	1708
Emilia_923	174	304	136	267
Fault_639	–	–	294	345
Ga41As41H72	–	–	135	143
Hook_1498	417	902	356	808
Si87H76	–	–	131	116

$\Rightarrow$ **performance and robustness improvement**

Results with new preconditioner

Results for $\varepsilon = 10^{-2}$:

Matrix	Π_{LU} Iter.	Π_{LU} Time	Π_{E_k} Iter.	Π_{E_k} Time
audikw_1	691	1163	331	625
Bump_2911	–	–	284	1708
Emilia_923	174	304	136	267
Fault_639	–	–	294	345
Ga41As41H72	–	–	135	143
Hook_1498	417	902	356	808
Si87H76	–	–	131	116

$\Rightarrow$ **performance and robustness improvement**

But what about storage?

What is the storage overhead of the Π_{E_k} preconditioner?

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices

$\Rightarrow$ but only needed after factorization

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices
$\Rightarrow$ but only needed after factorization

Traditional multifrontal storage is $S_A + S_{LU} + S_{CB}$

- S_A = storage for matrix A
- S_{LU} = storage for (BLR) LU factors
- S_{CB} = storage for contribution blocks $\Rightarrow$ temporary storage during factorization

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices
$\Rightarrow$ but only needed after factorization

Traditional multifrontal storage is $S_A + S_{LU} + S_{CB}$

- S_A = storage for matrix A
- S_{LU} = storage for (BLR) LU factors
- S_{CB} = storage for contribution blocks $\Rightarrow$ temporary storage during factorization

Thus, S_{CB} and S_{E_k} do not overlap!

- Factorization storage: $S_A + S_{LU} + S_{CB}$
- Solution storage: $S_A + S_{LU} + S_{E_k}$

$\Rightarrow$ Total storage: $S_A + S_{LU} + \max(S_{CB}, S_{E_k})$

Storage overhead: formula

We need to store E_k: two dense $n \times k$ matrices

$\Rightarrow$ but only needed after factorization

Traditional multifrontal storage is $S_A + S_{LU} + S_{CB}$

- S_A = storage for matrix A
- S_{LU} = storage for (BLR) LU factors
- S_{CB} = storage for contribution blocks $\Rightarrow$ temporary storage during factorization

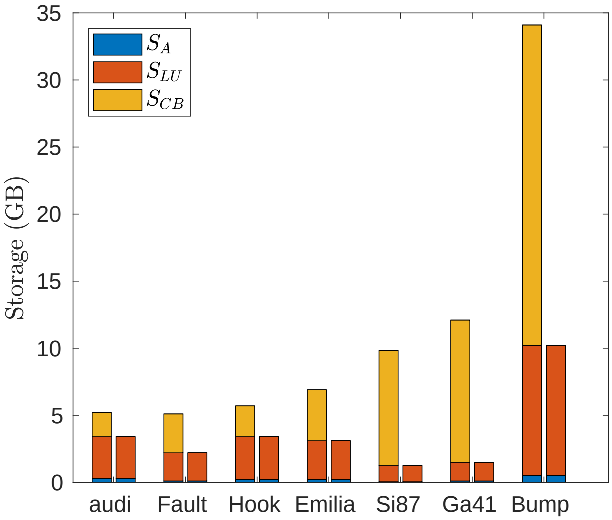
Thus, S_{CB} and S_{E_k} do not overlap!

- Factorization storage: $S_A + S_{LU} + S_{CB}$
- Solution storage: $S_A + S_{LU} + S_{E_k}$

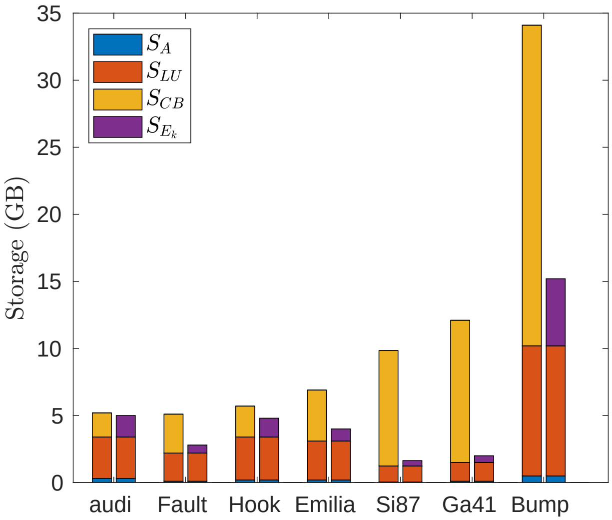
$\Rightarrow$ Total storage: $S_A + S_{LU} + \max(S_{CB}, S_{E_k})$

If $S_{E_k} \leq S_{CB}$, zero storage overhead!

Storage overhead: results



Storage overhead: results



$\Rightarrow$ **zero storage overhead on all matrices**

Conclusion

A new preconditioner

- Ill-conditioned matrices often have a numerically low-rank inverse
- Novel preconditioner based on a low-rank approximation to the error to accelerate linear systems solution

Application to BLR low-accuracy preconditioners

- Low-accuracy BLR solvers require very little storage
- Our new preconditioner improves both their performance and robustness, with zero storage overhead in the multifrontal context

Slides and paper available here

`bit.ly/theomary`

References

- N. J. Higham and T. Mary, *A New Preconditioner that Exploits Low-Rank Approximations to Factorization Error*, MIMS EPrint 2018.10.
- N. J. Higham and T. Mary, *Accelerating Linear Systems Solution by Exploiting Low-Rank Approximations to Factorization Error*, IMA Conference on Numerical Linear Algebra and Optimization (2018).
- P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary, *Performance and Scalability of the BLR Multifrontal Factorization on Multicore Architectures*, ACM Trans. Math. Soft. (2018).
- T. Mary, *Block Low-Rank multifrontal solvers: complexity, performance, and scalability*, PhD thesis (2017).
- E. Carson and N. J. Higham, *A New Analysis of Iterative Refinement and Its Application to Accurate Solution of Ill-Conditioned Sparse Linear Systems*, SIAM J. Sci. Comp. (2017).
- E. Carson and N. J. Higham, *Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions*, SIAM J. Sci. Comp. (2018).