Parallelization and Pivoting in a Block-Low Rank Multifrontal Solver

Patrick Amestoy[*] Cleve Ashcraft[†] Olivier Boiteau[‡] Alfredo Buttari[§]
Jean-Yves L'Excellent[¶] Clement Weisbecker[†]

SIAM Parallel Processing '14, February 21st, 2014

[*]INPT-ENSEEIHT-IRIT [†] LSTC [‡] EDF [§] CNRS-IRIT [¶] INRIA-LIP

LSTC
Livermore Software Technology Corp.

Large sparse linear system $A\,x = b$

Often a keystone in scientific computing (discretization of PDEs, step of an optimization method, regression...).

Sparsity

- A sparse matrix is "any matrix with enough zeros that it pays to take advantage of them" (Wilkinson).

Multifrontal direct methods

- rely on dense kernels
- goal: factorization of $A = LU$, $A = LDL^T$ or $A = LL^T$
- then: forward and backward substitutions $Ly = b$ and $Ux = y$

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

34 days on the world's 14th most powerful supercomputer! (and 200K$!)

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

34 days on the world's 14th most powerful supercomputer! (and 200K$!)

$\Rightarrow$critical to improve direct methods while maintaining:

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

34 days on the world's 14th most powerful supercomputer! (and 200K$!)

$\Rightarrow$critical to improve direct methods while maintaining:

- out-of-core (OOC)

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

34 days on the world's 14th most powerful supercomputer! (and 200K$!)

$\Rightarrow$critical to improve direct methods while maintaining:

- out-of-core (OOC)
- algebraic ($\Rightarrow$ "black-box")

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

34 days on the world's 14th most powerful supercomputer! (and 200K$!)

$\Rightarrow$critical to improve direct methods while maintaining:

- out-of-core (OOC)
- algebraic ($\Rightarrow$ "black-box")
- pivoting for numerical stability

Solution of sparse linear systems with direct methods

Regular problems	2D $N \times N$ grid	3D $N \times N \times N$ grid
Nonzeros in factors	$O(N^2 log\ n)$	$O(N^4)$
Floating-point ops	$O(N^3)$	$O(N^6)$

$\Rightarrow$ extrapolation on a $1000 \times 1000 \times 1000$ grid:

50 exaflops (10^{18}) and 700 Tbytes for factors!

34 days on the world's 14th most powerful supercomputer! (and 200K$!)

$\Rightarrow$critical to improve direct methods while maintaining:

- out-of-core (OOC)
- algebraic ($\Rightarrow$ "black-box")
- pivoting for numerical stability
- distributed-memory parallelism

Low-rank approximations

"A low-rank matrix is any matrix with enough very small singular values that it pays off to ignore them" ($\Rightarrow$ data sparse)

Low-rank approximations

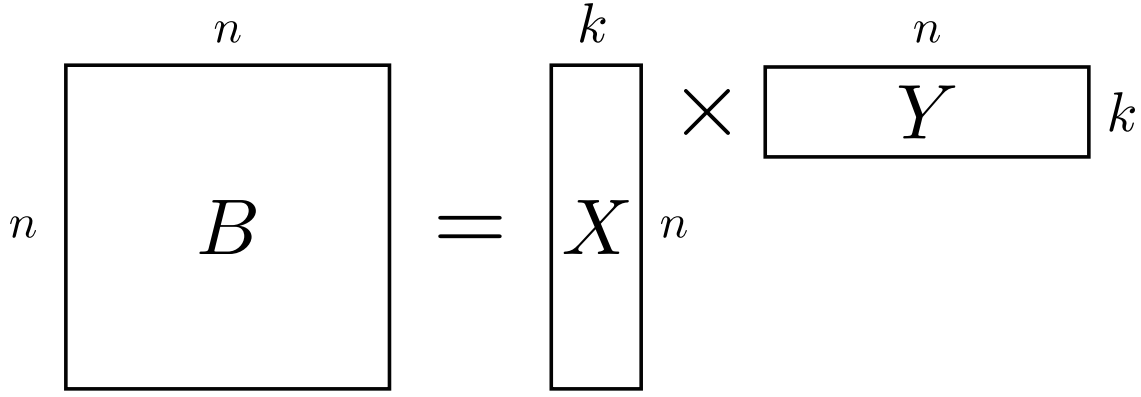
"A low-rank matrix is any matrix with enough very small singular values that it pays off to ignore them" ($\Rightarrow$ data sparse)

$\Rightarrow$ "very small" depends on the application

Low-rank approximations

"A low-rank matrix is any matrix with enough very small singular values that it pays off to ignore them" ($\Rightarrow$ data sparse)

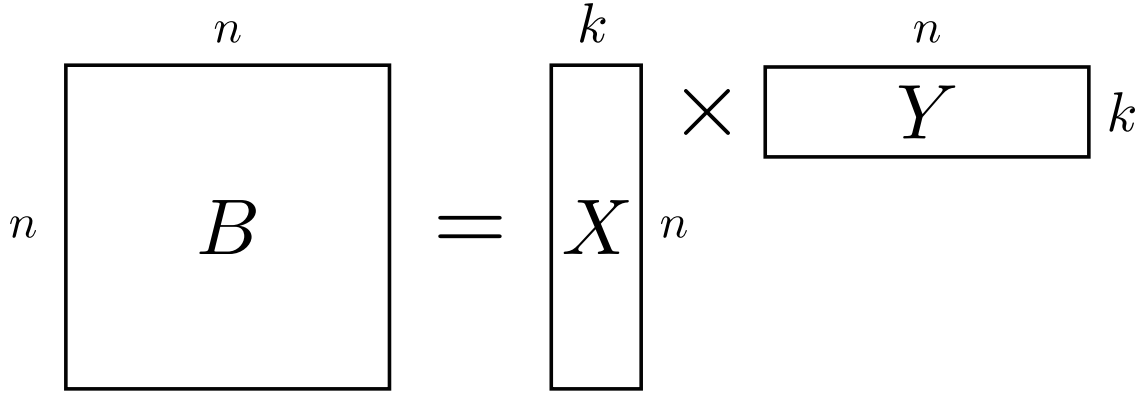
$\Rightarrow$ "very small" depends on the application



Low-rank approximations

"A low-rank matrix is any matrix with enough very small singular values that it pays off to ignore them" ($\Rightarrow$ data sparse)

$\Rightarrow$ "very small" depends on the application



- memory and operations reduction
- numerical parameter ε for the approximation
- theory [Bebendorf, Chandrasekaran, Hackbusch, ...] available

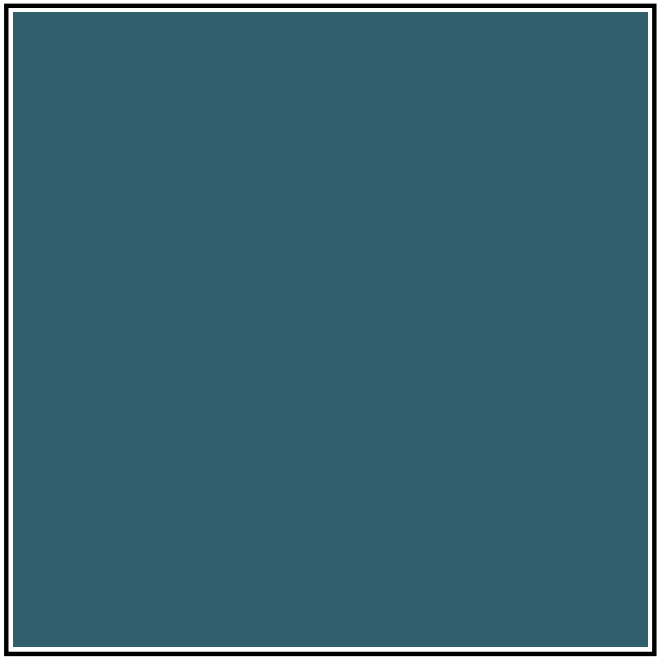
Part I

Block Low-Rank format

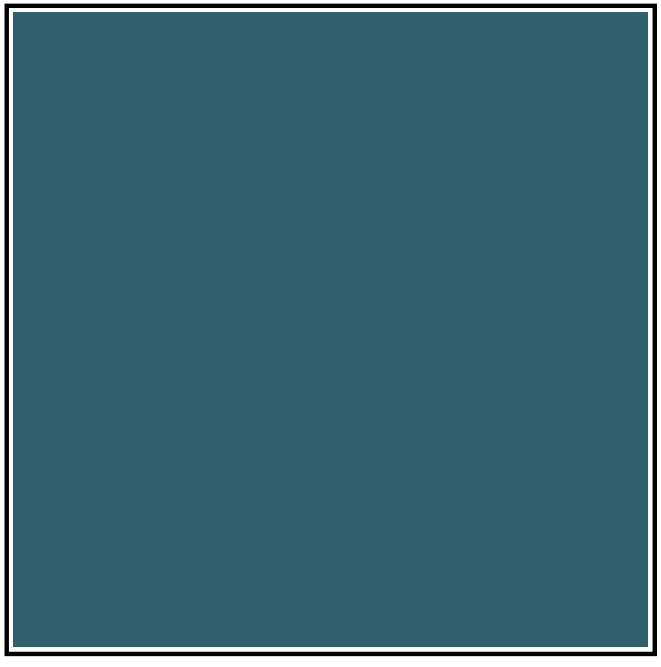
Low-rank approximations – representations

- Many representations: $\mathcal{H}, \mathcal{H}^2$ [Bebendof, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...]
- Simpler representations apply to broader classes of problems but provide less potential gain in memory/operations.

Hierarchically Semi-Separable (HSS):



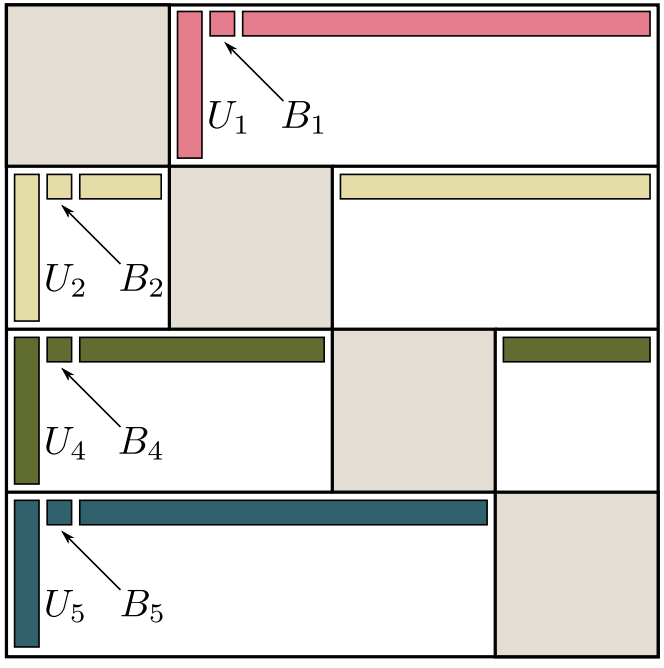
Block Low-Rank (BLR):



Low-rank approximations – representations

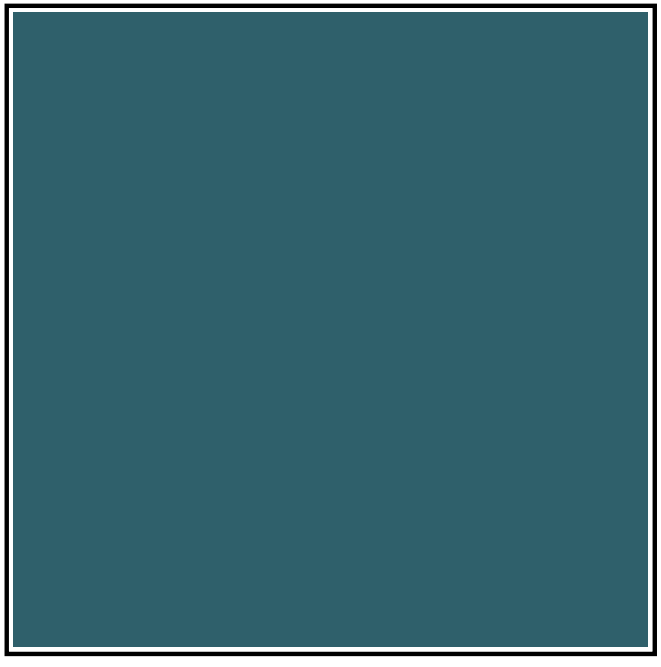
- Many representations: $\mathcal{H}, \mathcal{H}^2$ [Bebendof, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...]
- Simpler representations apply to broader classes of problems but provide less potential gain in memory/operations.

Hierarchically Semi-Separable (HSS):



Recursive partitioning where off-diag blocks are low-rank and nested.

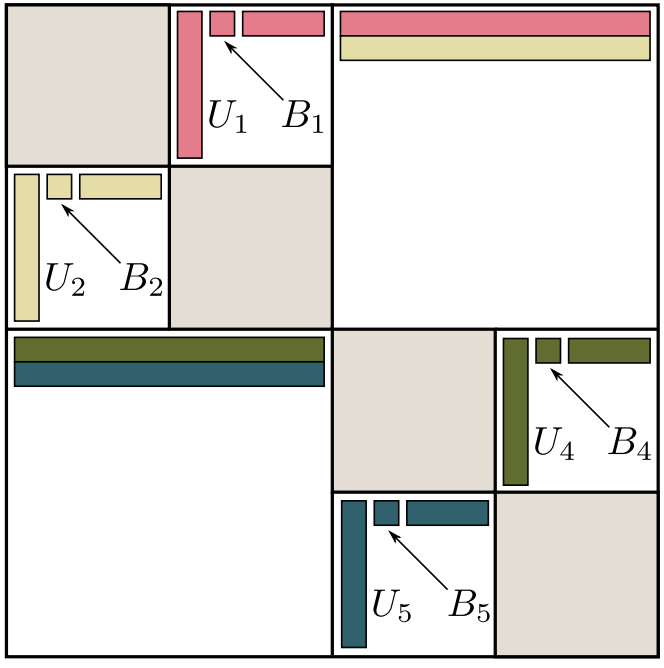
Block Low-Rank (BLR):



Low-rank approximations – representations

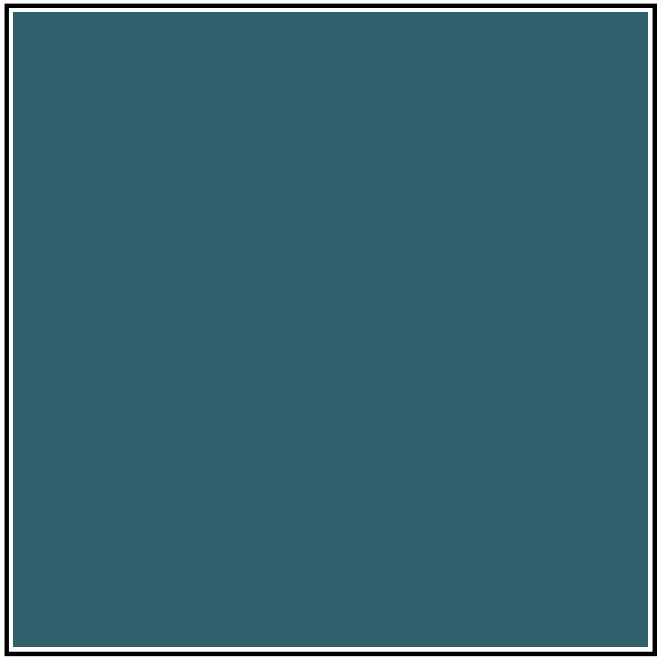
- Many representations: $\mathcal{H}, \mathcal{H}^2$ [Bebendof, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...]
- Simpler representations apply to broader classes of problems but provide less potential gain in memory/operations.

Hierarchically Semi-Separable (HSS):



Recursive partitioning where off-diag blocks are low-rank and nested.

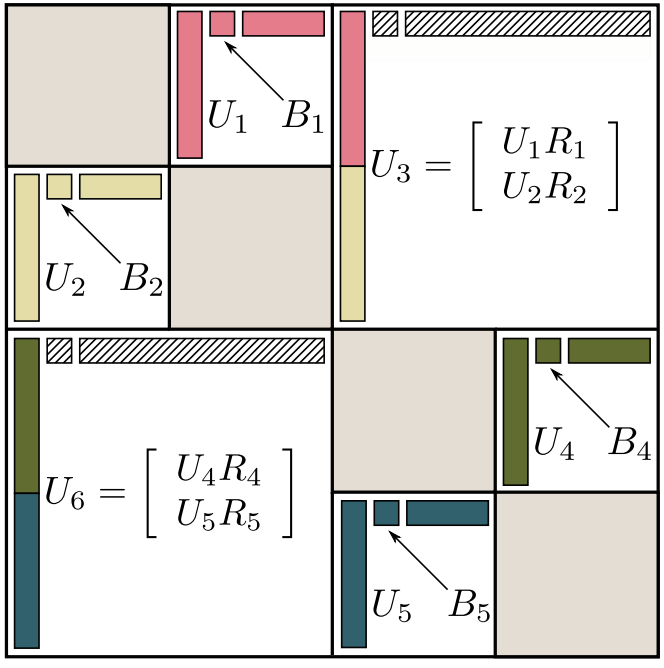
Block Low-Rank (BLR):



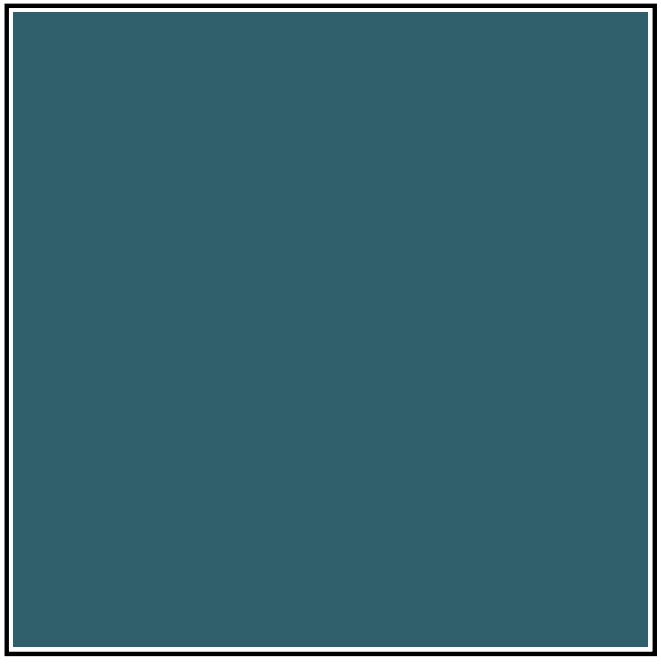
Low-rank approximations – representations

- Many representations: $\mathcal{H}, \mathcal{H}^2$ [Bebendof, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...]
- Simpler representations apply to broader classes of problems but provide less potential gain in memory/operations.

Hierarchically Semi-Separable (HSS):



Block Low-Rank (BLR):

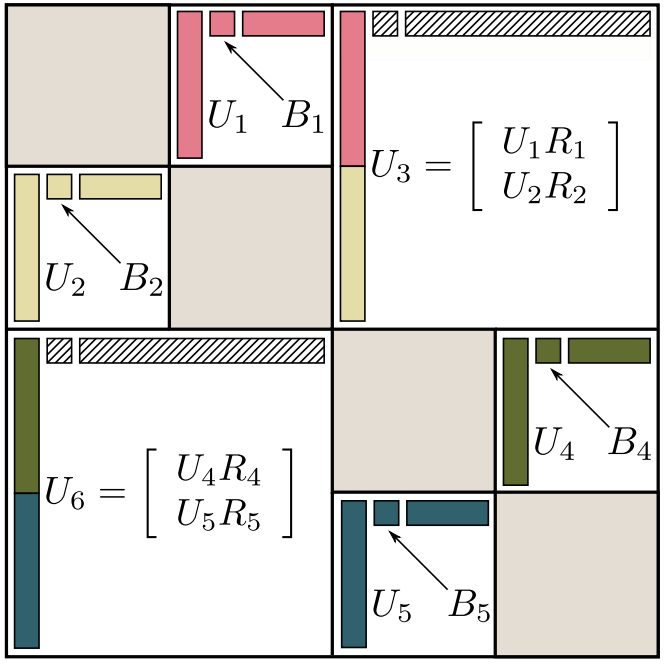


Recursive partitioning where off-diag blocks are low-rank and nested.

Low-rank approximations – representations

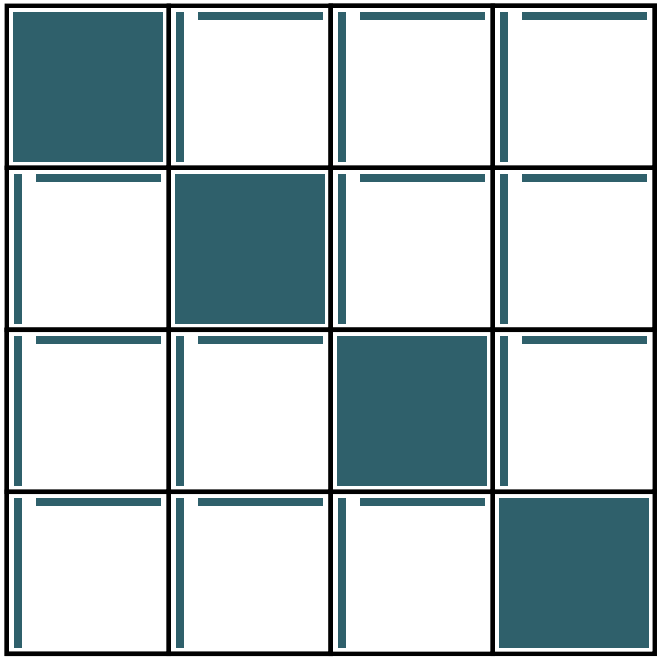
- Many representations: $\mathcal{H}, \mathcal{H}^2$ [Bebendof, Börm, Hackbush, Grasedyck,...], HSS/SSS [Chandrasekaran, Dewilde, Gu, Li, Xia,...]
- Simpler representations apply to broader classes of problems but provide less potential gain in memory/operations.

Hierarchically Semi-Separable (HSS):



Recursive partitioning where off-diag blocks are low-rank and nested.

Block Low-Rank (BLR):

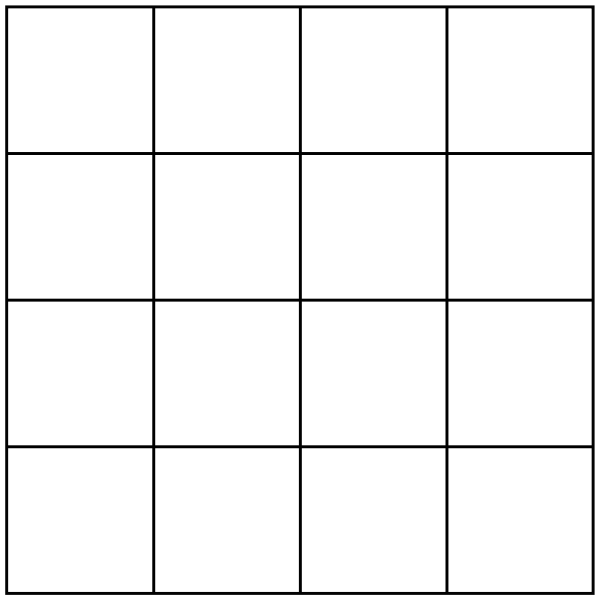


2D partitioning where off-diag blocks are low-rank and do not interact.

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$

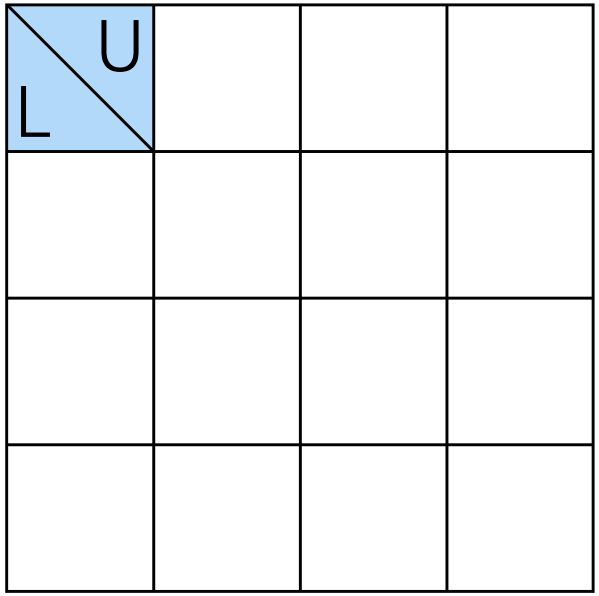
(r=block size, k=rank)



Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$

(r=block size, k=rank)

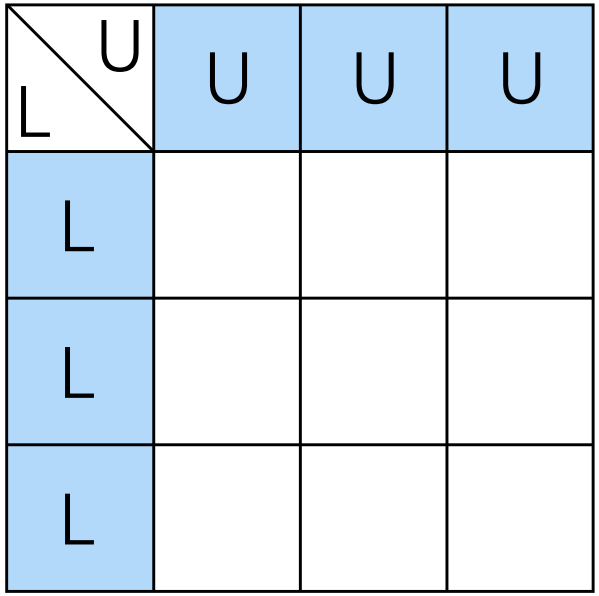


- **FACTOR**
- SOLVE
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$

(r=block size, k=rank)

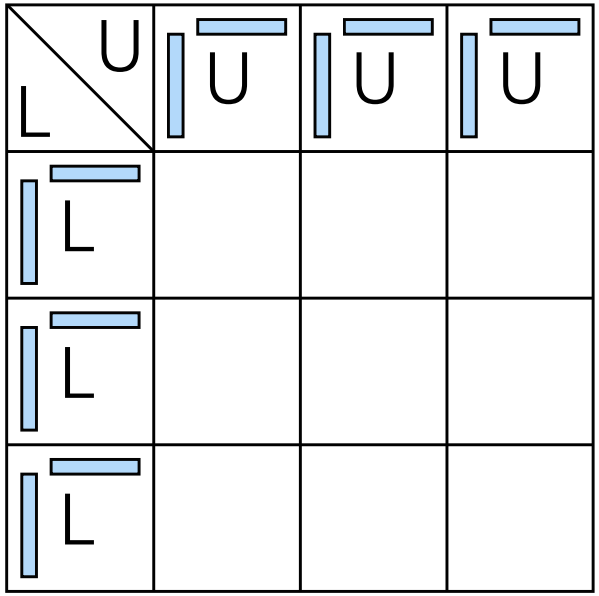


- FACTOR
- **SOLVE**
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

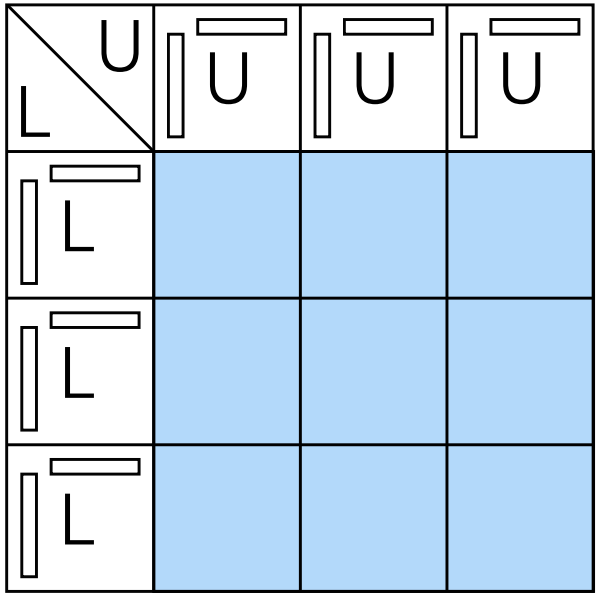


- FACTOR
- SOLVE
- **COMPRESS**
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

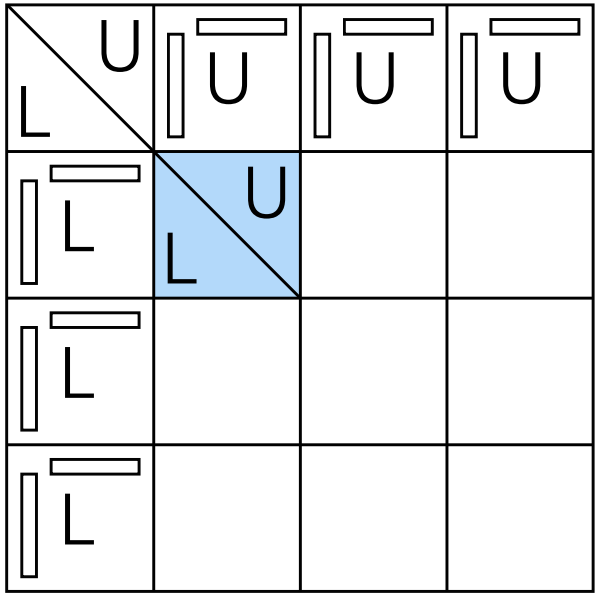


- FACTOR
- SOLVE
- COMPRESS
- **UPDATE**

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

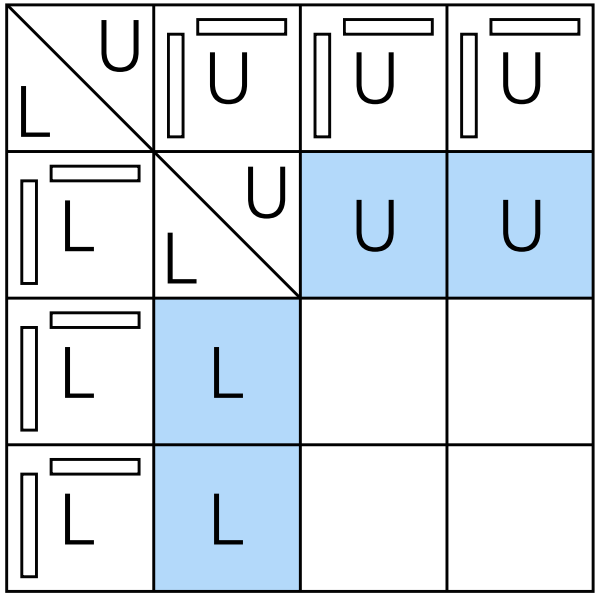


- **FACTOR**
- SOLVE
- COMPRESS
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

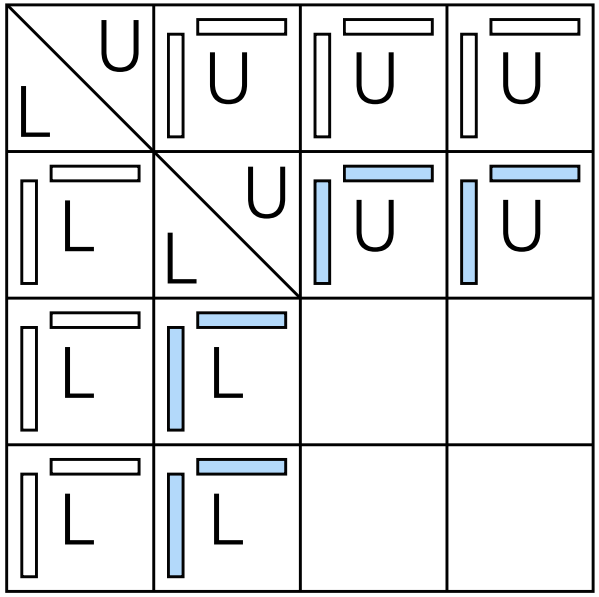


- FACTOR
- **SOLVE**
- COMPRESS
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

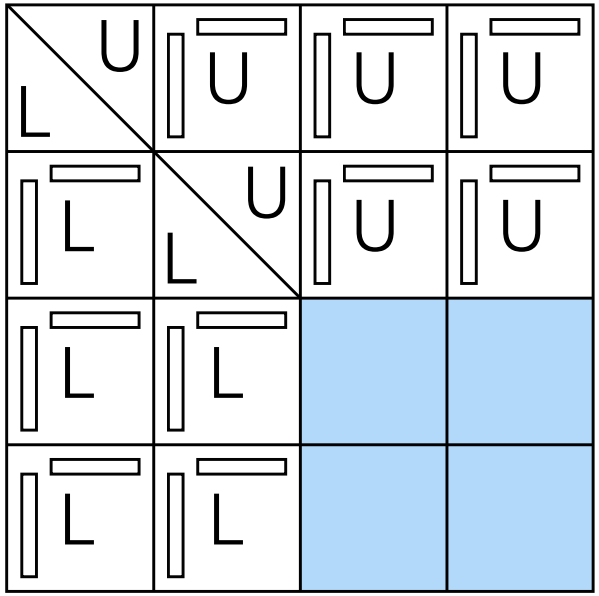


- FACTOR
- SOLVE
- **COMPRESS**
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

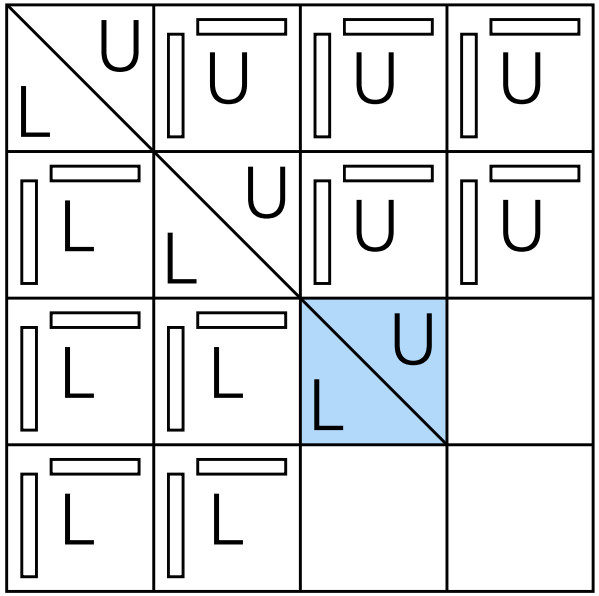


- FACTOR
- SOLVE
- COMPRESS
- **UPDATE**

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

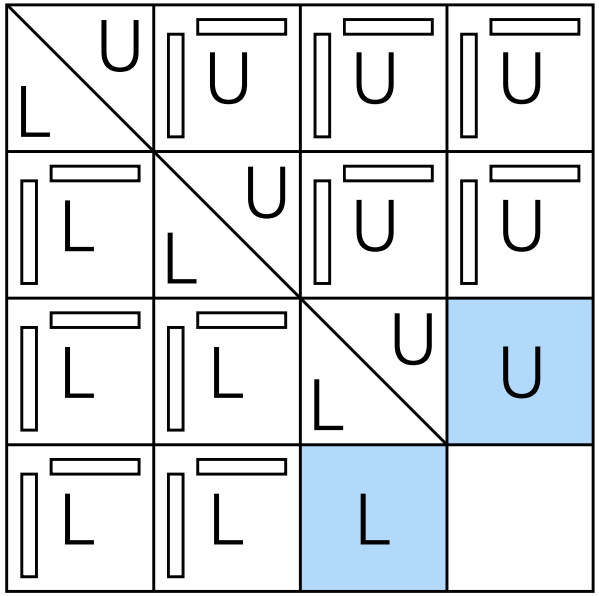


- **FACTOR**
- SOLVE
- COMPRESS
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

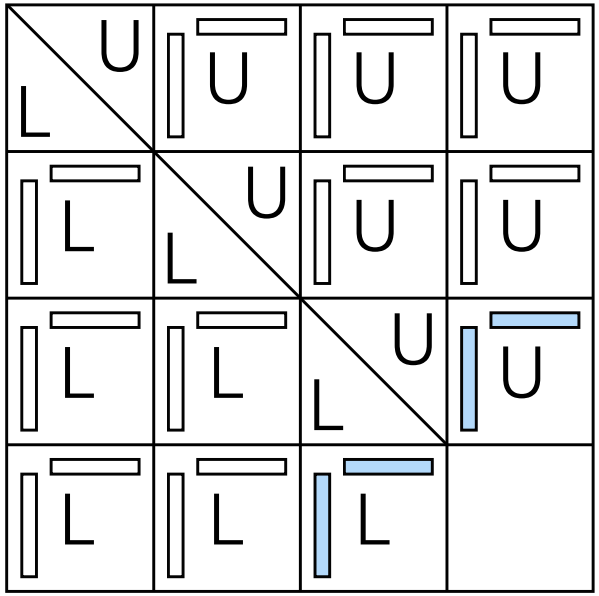


- FACTOR
- **SOLVE**
- COMPRESS
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

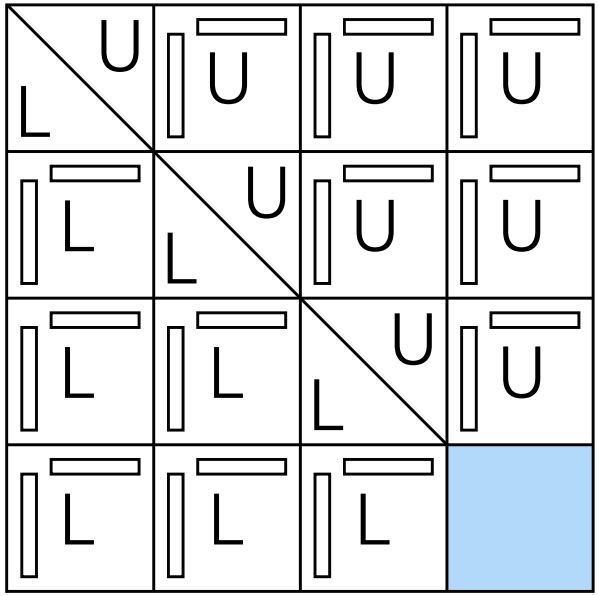


- FACTOR
- SOLVE
- **COMPRESS**
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

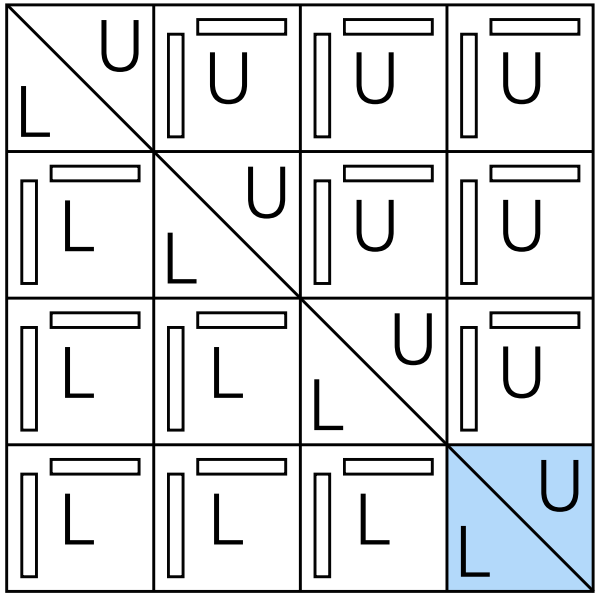


- FACTOR
- SOLVE
- COMPRESS
- **UPDATE**

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)

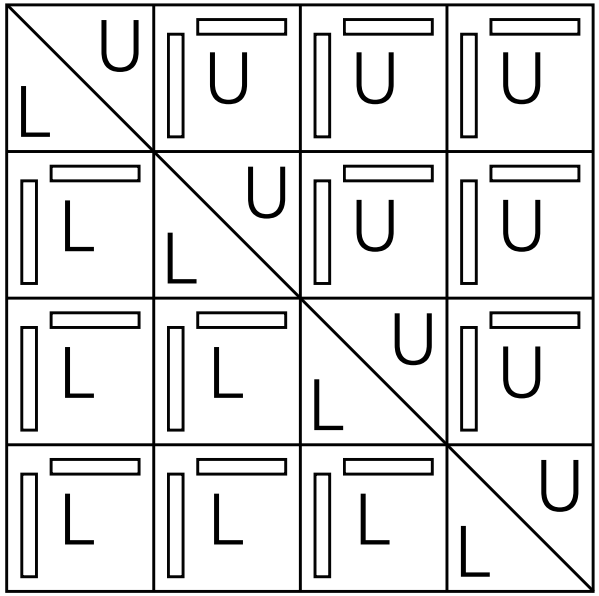


- **FACTOR**
- SOLVE
- COMPRESS
- UPDATE

Dense BLR LU factorization

task	operation type	dense	low-rank
Factor (F)	$B = LU^T$	$(2/3)r^3$	$(2/3)r^3$
Solve (S)	$D = X(YL^{-1})$	r^3	kr^2
Update (U)	$D = D - X_1(Y_1X_2)Y_2$	$2r^3$	$2kr^2$
Compress (C)	$C = XY$	kr^2	—

(r=block size, k=rank)



- **F**ACTOR
- **S**OLVE
- **C**OMPRESS
- **U**PDATE

[Demmel et al. 1995] proposed a bound for $\|\Delta A\| = \|A - \hat{L}\hat{U}\|$ in the FR case:

$$\|\Delta A\| \leq u\left(\delta(n,r)\|A\| + \theta(n,r)\|\hat{L}\|\|\hat{U}\|\right) + O(u^2)$$

where r is the block size, $\delta(n,r)$ and $\theta(n,r)$ are polynomial constants, and u the unit roundoff.

For the FSCU algorithm, we proved that:

$$\begin{aligned}\|\Delta A\| \leq\ & u\left(\delta(n,r)\|A\| + \theta(n,r)\|\hat{L}\|\|\hat{U}\|\right) + O(u^2) \\ +\ & u\left(\varepsilon\|\hat{U}\| + \varepsilon\|\hat{L}\| + \varepsilon^2\right) + r\varepsilon \max\left(\|\hat{L}\|, \|\hat{U}\|\right)\end{aligned}$$

[Demmel et al. 1995] proposed a bound for $\|\Delta A\| = \|A - \hat{L}\hat{U}\|$ in the FR case:

$$\|\Delta A\| \leq u\left(\delta(n,r)\|A\| + \theta(n,r)\|\hat{L}\|\|\hat{U}\|\right) + O(u^2)$$

where r is the block size, $\delta(n,r)$ and $\theta(n,r)$ are polynomial constants, and u the unit roundoff.

For the FSCU algorithm, we proved that:

$$\|\Delta A\| \leq u\left(\delta(n,r)\|A\| + \theta(n,r)\|\hat{L}\|\|\hat{U}\|\right) + O(u^2)$$
$$+ \; u\left(\varepsilon\|\hat{U}\| + \varepsilon\|\hat{L}\| + \varepsilon^2\right) + r\varepsilon \max\left(\|\hat{L}\|, \|\hat{U}\|\right)$$

[Demmel et al. 1995] proposed a bound for $\|\Delta A\| = \|A - \hat{L}\hat{U}\|$ in the FR case:

$$\|\Delta A\| \leq u\left(\delta(n,r)\|A\| + \theta(n,r)\|\hat{L}\|\|\hat{U}\|\right) + O(u^2)$$

where r is the block size, $\delta(n,r)$ and $\theta(n,r)$ are polynomial constants, and u the unit roundoff.

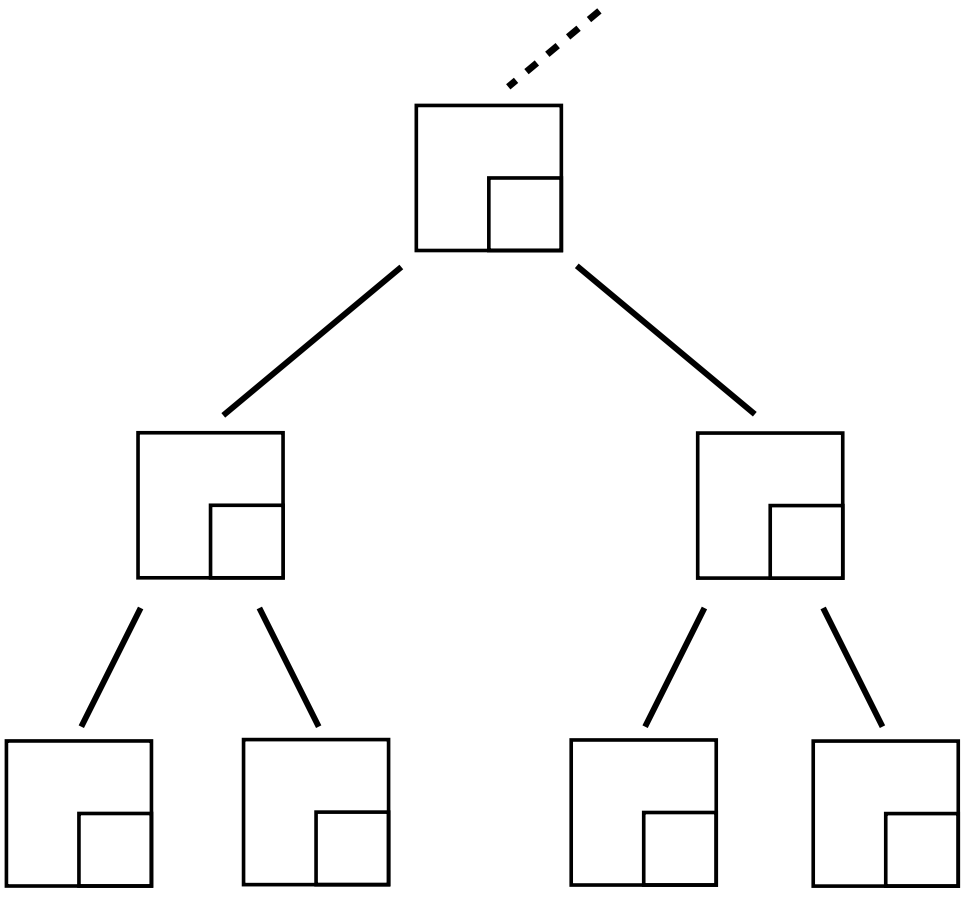
For the FSCU algorithm, we proved that:

$$\|\Delta A\| \leq u\left(\delta(n,r)\|A\| + \theta(n,r)\|\hat{L}\|\|\hat{U}\|\right) + O(u^2)$$
$$+ u\left(\varepsilon\|\hat{U}\| + \varepsilon\|\hat{L}\| + \varepsilon^2\right) + r\varepsilon\max\left(\|\hat{L}\|, \|\hat{U}\|\right)$$

Part II

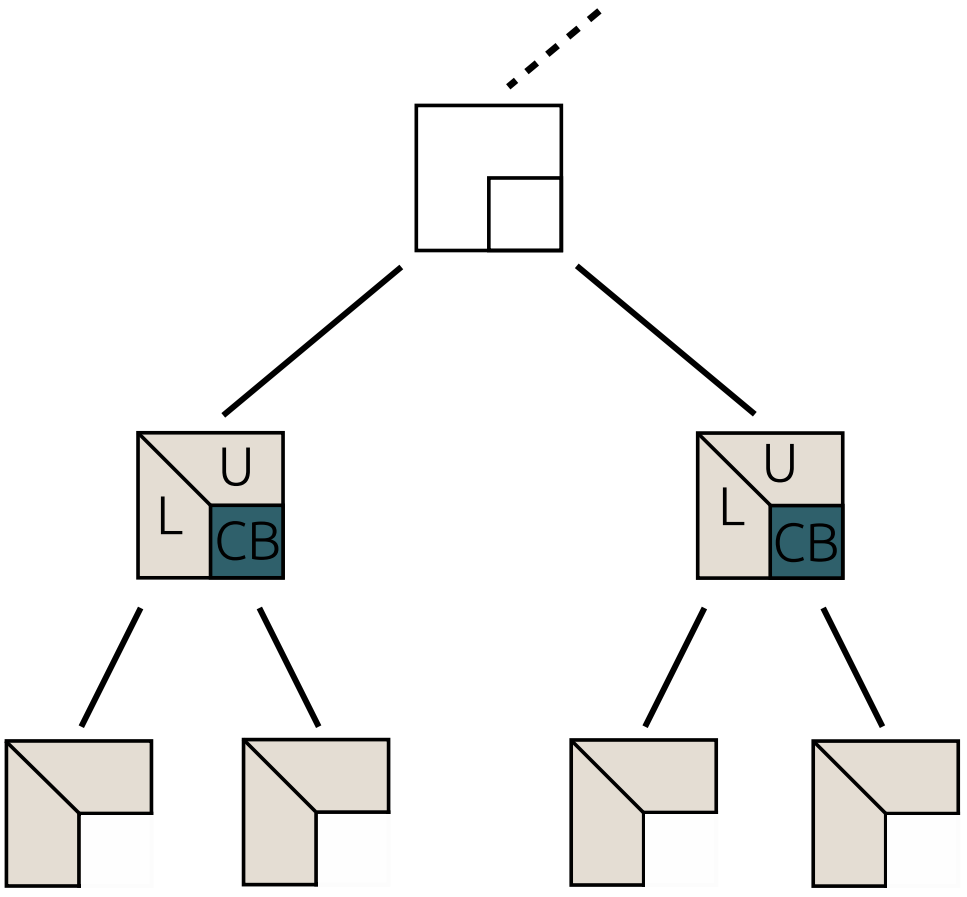
Block Low-Rank multifrontal method

The multifrontal method [Duff & Reid '83]



Elimination tree

The multifrontal method [Duff & Reid '83]

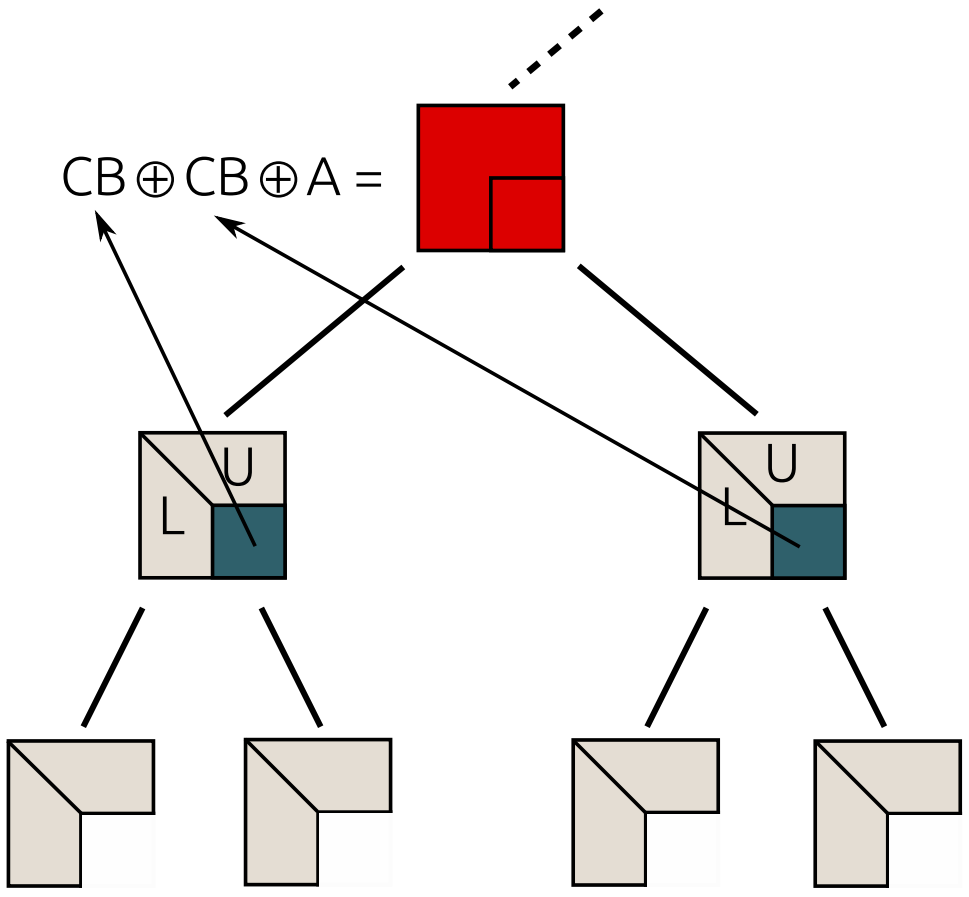


Elimination tree

Partial factorizations yield

- rows of U
- columns of L
- contribution blocks (CB)

The multifrontal method [Duff & Reid '83]

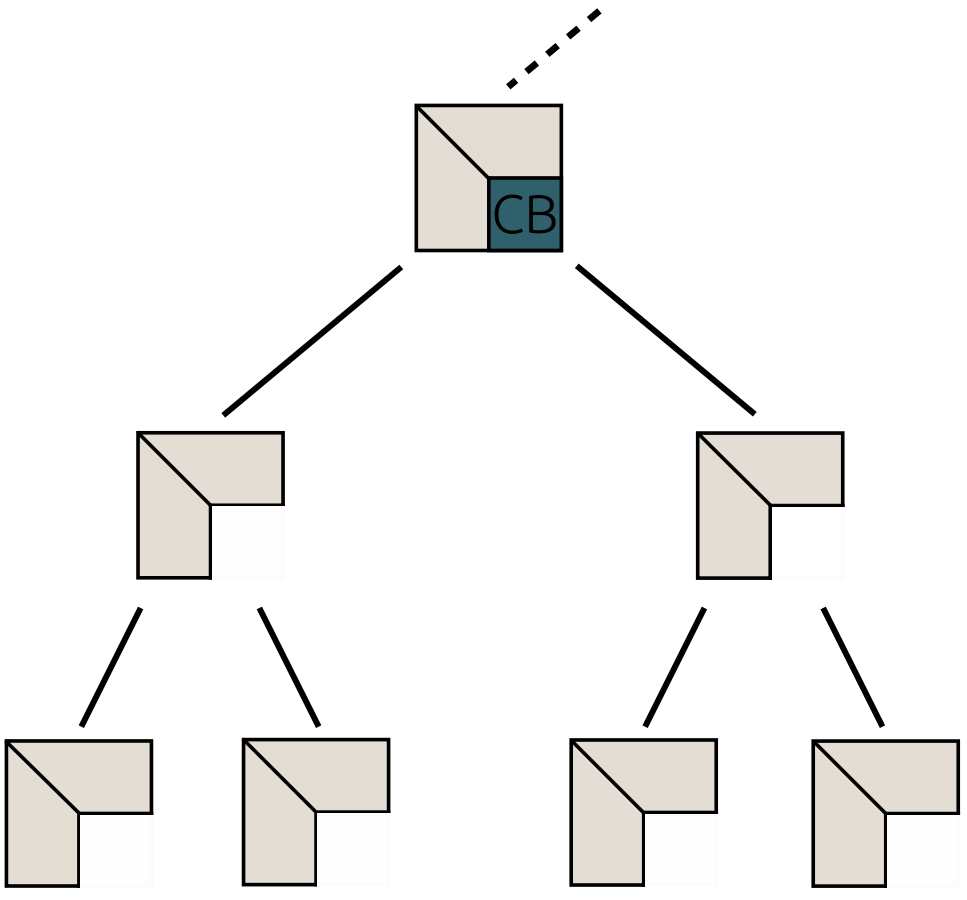


Elimination tree

Assembly:

CBs are "summed" with values in A to form a new front

The multifrontal method [Duff & Reid '83]

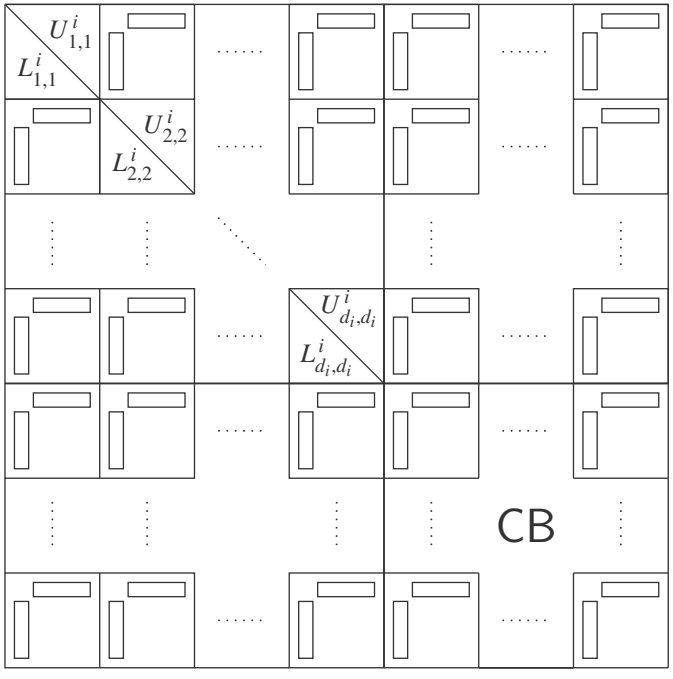


Elimination tree

A new front is processed

General idea

Fronts are not low-rank but have low-rank subblocks [Chandrasekaran et al. 2010, Hackbusch 1999]

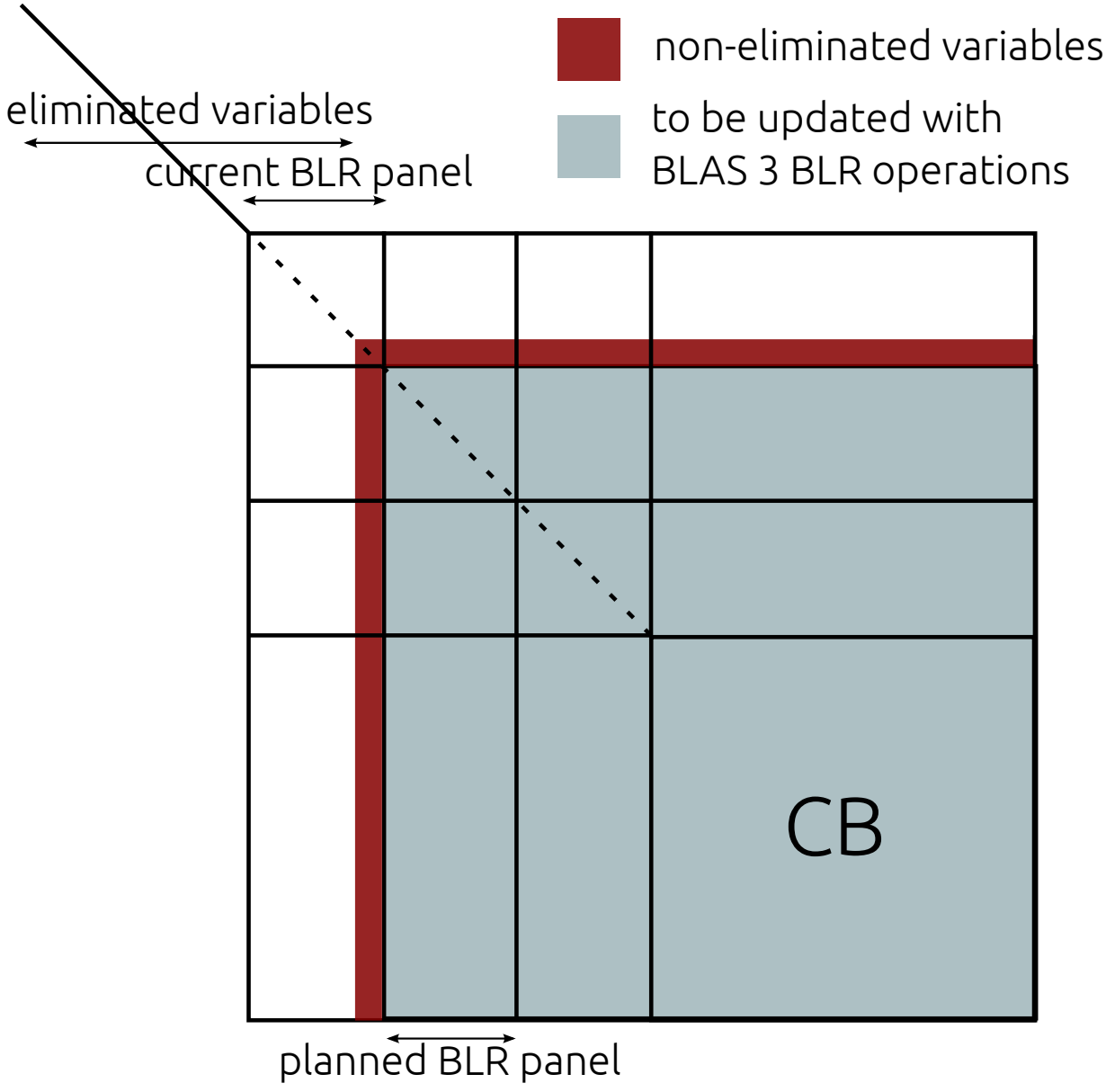


⇒ blocks must be numerically meaningful to be low-rank

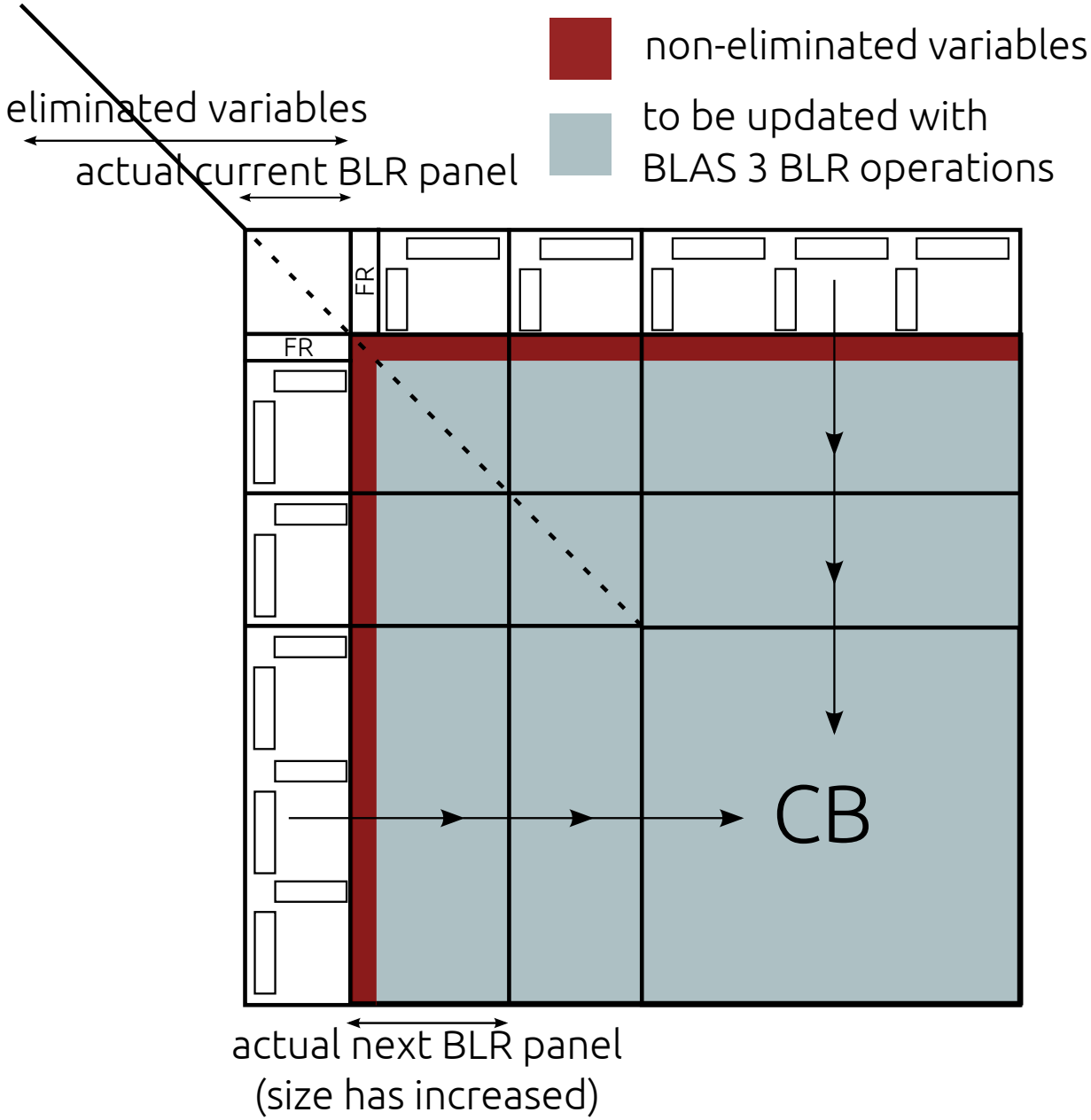
⇒ dense algorithm (FSCU) adapts to partial factorization

⇒ only larger fronts are processed with BLR

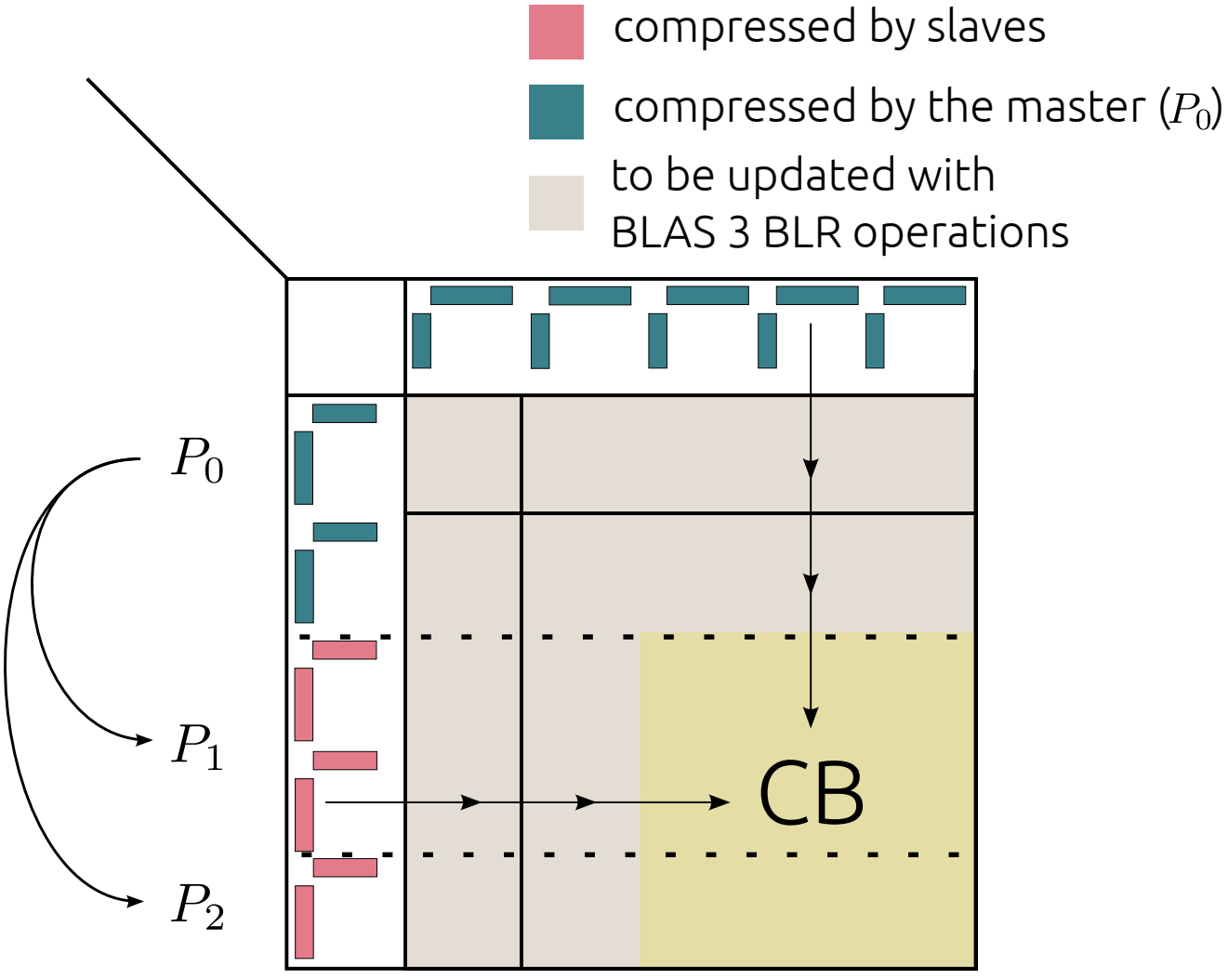
Threshold partial pivoting with BLR

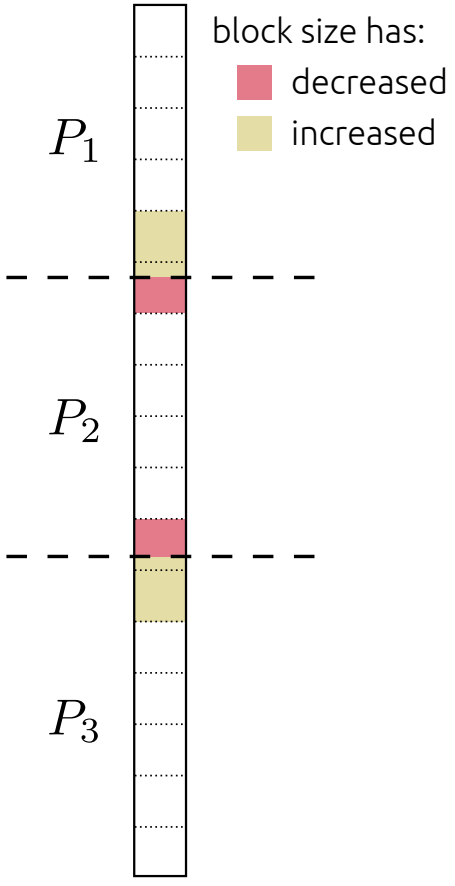


Threshold partial pivoting with BLR



Distributed memory parallelism with BLR





$\Rightarrow$ clustering adapted dynamically to the distribution among processors of MUMPS

Part III

Results

Experimental environment

metrics:

- $|\mathbf{L}|$: fraction of FR factors storage obtained with BLR (%)
- **ops**: operations for BLR factorization (including compression)
- **time**: time for BLR factorization (including compression)

Standard operators:

- `poisson`: $\Delta u = f$ with Dirichlet conditions (sym, real)
- `helmholtz`: $\left(-\Delta - \frac{\omega^2}{v(x)^2}\right)\ u(x,\omega) = s(x,\omega)$ (unsym, complex)

Other matrices:

- applicative problems from Prof. Davis's collection and EDF

Accuracy:

- CSR = Componentwise Scaled Residual $= \max_i \left(\frac{|A\tilde{x}-b|_i}{(|A||\tilde{x}|+|b|)_i}\right)$

Distributed-memory parallelism

- preliminary results on a small problem (128^3)

#	Poisson	Helmholtz
procs	|L|	
1	53.3%	72.0%
4	54.5%	n/a
8	54.5%	n/a
16	54.6%	73.6%
32	54.9%	73.6%
64	55.3%	73.7%
128	56.0%	74.1%

$\Rightarrow$ factor compression well maintained (slight increase)

Distributed-memory parallelism

- preliminary results on a small problem (128^3)

# procs	ops	execution time	
		BLR	MUMPS
1	52.3%	n/a	n/a
16	56.4%	3133s.	4905s.
32	56.5%	2125s.	2648s.
64	56.7%	1409s.	1537s.
128	57.3%	1014s.	979s.

(Helmholtz only)

$\Rightarrow$ ops compression well maintained (slight increase)

$\Rightarrow$ timings need to be improved (+ too many procs)

Pivoting

name	struct	n ($\times 10^6$)	NZ	application
`cont-300`	sym	0.18	0.53	optimization (linear, UFL)
`kkt_power`	sym	2	8	optimal power flow (nonlinear, UFL)
`d-plan-inco`	nonsym	2	21	nonlinear mechanics (EDF)

- three different applications
- very large amount of pivoting
- static pivoting: if $|a_{k,k}| < \sqrt{\epsilon_{mach}} \cdot \|A\|$, then $a_{k,k} \leftarrow \sqrt{\epsilon_{mach}} \cdot \|A\|$

- Why is partial pivoting needed with BLR applications ?

	cont-300	kkt_power	d-plan-inco

Pivoting

	cont-300	kkt_power	d-plan-inco
MUMPS FR partial	5.68E-8	3.05E-10	1.90E-12
MUMPS FR static	5.71E-3	4.66E-4	8.88E-7

- partial pivoting needed in FR

Pivoting

	cont-300	kkt_power	d-plan-inco
MUMPS FR partial	5.68E-8	3.05E-10	1.90E-12
MUMPS FR static	5.71E-3	4.66E-4	8.88E-7
BLR (10^{-14}) static	5.91E-3	7.90E-5	1.04E-6
Compressions ($\|L\|$ - f)	98 - 116	75 - 64	46 - 8
BLR (10^{-14}) partial	2.06E-8	2.65E-9	2.37E-12
Compressions ($\|L\|$ - f)	92 - 82	70 - 57	47 - 8

- partial pivoting needed in FR
- pivoting does not decrease the compression rates

Pivoting

	cont-300	kkt_power	d-plan-inco
MUMPS FR partial	5.68E-8	3.05E-10	1.90E-12
MUMPS FR static	5.71E-3	4.66E-4	8.88E-7
BLR (10^{-14}) static	5.91E-3	7.90E-5	1.04E-6
Compressions ($\|L\|$ - f)	98 - 116	75 - 64	46 - 8
BLR (10^{-14}) partial	2.06E-8	2.65E-9	2.37E-12
Compressions ($\|L\|$ - f)	92 - 82	70 - 57	47 - 8
BLR (ε) partial	1.59E-3(10^{-9})	9.77E-4(10^{-8})	6.28E-7(10^{-7})
Compressions ($\|L\|$ - f)	86 - 67	54 - 31	44 - 7

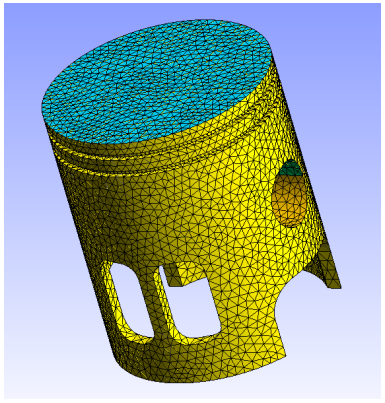
- partial pivoting needed in FR
- pivoting does not decrease the compression rates
- two choices: better solution or same solution but faster

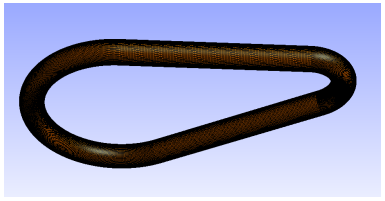
Pivoting

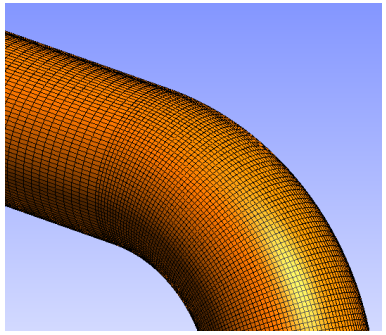
	cont-300	kkt_power	d-plan-inco
MUMPS FR partial	5.68E-8	3.05E-10	1.90E-12
MUMPS FR static	5.71E-3	4.66E-4	8.88E-7
BLR (10^{-14}) static	5.91E-3	7.90E-5	1.04E-6
Compressions ($\|L\|$ - f)	98 - 116	75 - 64	46 - 8
BLR (10^{-14}) partial	2.06E-8	2.65E-9	2.37E-12
Compressions ($\|L\|$ - f)	92 - 82	70 - 57	47 - 8
BLR (ε) partial	1.59E-3(10^{-9})	9.77E-4(10^{-8})	6.28E-7(10^{-7})
Compressions ($\|L\|$ - f)	86 - 67	54 - 31	44 - 7
Time MUMPS FR partial	4 s.	4090 s.	321 s.
Time BLR partial (10^{-14})	3 s.	560 s.	82 s.
Time BLR partial (ε)	2 s. (10^{-9})	484 s. (10^{-8})	78s. (10^{-7})

- partial pivoting needed in FR
- pivoting does not decrease the compression rates
- two choices: better solution or same solution but faster

Preconditioning in Code_Aster: set of problems







	n	NZ	Cond.	application
`piston`	1.3E+6	54.7E+6	5.1E+5	external pressure force on the top
`perf`	2.0E+6	75.8E+6	1.5E+11	"cavity" hook subjected to internal pressure force (challenging for EDF)

- CG preconditioned with `MUMPS` single precision with BLR

Preconditioning in Code_Aster: results with piston (1)

FR SP = 3 iterations

	BLR SP			BLR DP		
ε	#it	|L|	ops	#it	|L|	ops
10^{-8}	4	66.4%	31.0%	2	65.6%	29.5%
10^{-7}	3	63.4%	27.5%	2	62.5%	26.2%
10^{-6}	3	60.1%	24.0%	3	59.1%	22.9%
10^{-5}	4	56.3%	20.5%	4	55.1%	19.6%
10^{-4}	6	51.7%	17.0%	7	50.4%	16.3%
10^{-3}	66	45.9%	13.2%	24	44.5%	12.8%
10^{-2}	no convergence in 1000s.					

$\Rightarrow$ convergence slightly improved in DP with similar compressions

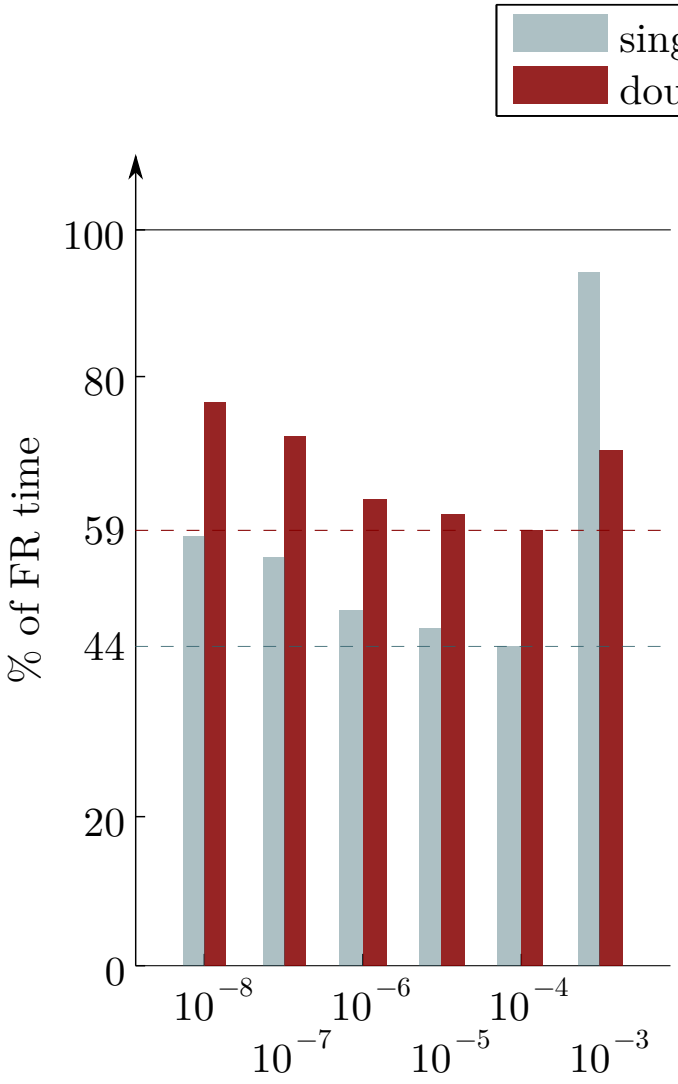
Preconditioning in Code_Aster: results with perf (1)

FR SP = 68 iterations

	BLR SP			BLR DP		
ε	#it	|L|	ops	#it	|L|	ops
10^{-8}	67	59.4%	26.7%	3	58.7%	25.5%
10^{-7}	68	56.9%	24.3%	4	56.4%	23.4%
10^{-6}	66	52.4%	20.2%	8	51.9%	19.7%
10^{-5}	67	49.1%	17.1%	19	48.6%	17.0%
10^{-4}	81	45.1%	14.1%	68	44.4%	14.1%
10^{-3}	no convergence in 1000s.					
10^{-2}						

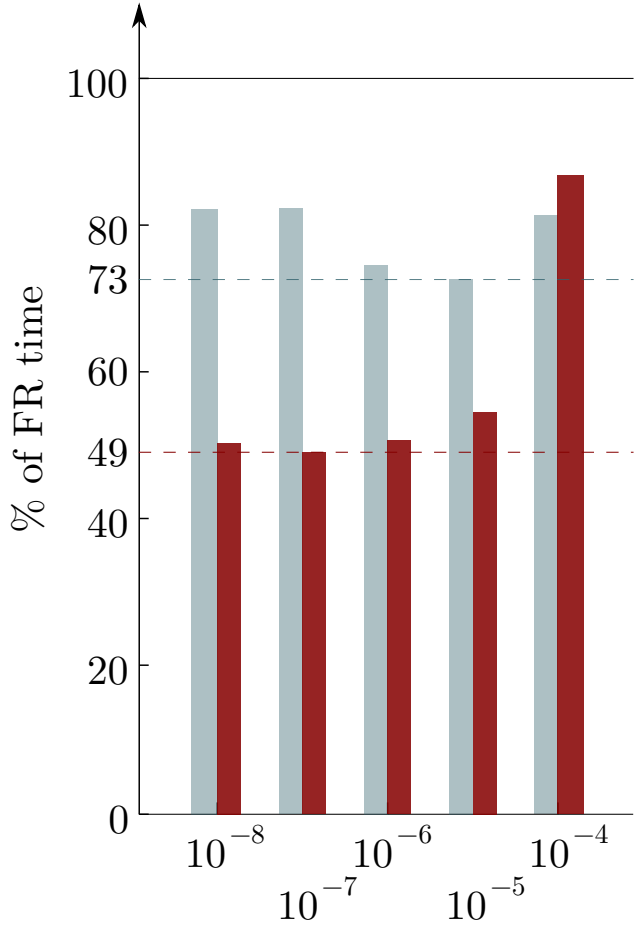
$\Rightarrow$ convergence substantially improved in DP with similar compressions

Preconditioning in Code_Aster: timings



piston / FR SP = 373s.

$\Longrightarrow$ best is SP!



perf / FR SP = 815s.

$\Longrightarrow$ best is DP!

Conclusion

A BLR multifrontal solver that is ...

- algebraic, parallel, general purpose
- numerically stable
- efficient (extrapolation on a $1000 \times 1000 \times 1000$ problem: 40 times less operations, 3 times less factor entries, even more for active memory)

Perspectives and challenges

- improve parallelism
- BLR forward and backward substitutions
- further study preconditioning
- BLAS3 efficiency

Thank you for your attention!

[weisbecker.perso.enseeiht.fr]
[clement@lstc.com]

Acknowledgements

- S. Gratton, X.S. Li, A. Napov, M. Ngom, F.-H. Rouet, N. Tardieu, D. Titley-Peloquin, LBNL, EDF

More details in ...

- "*Improving multifrontal methods by means of block low-rank representations*", Amestoy et al., 2012, submitted to SIAM SISC
- "*Improving multifrontal solvers by means of algebraic block low-rank representations*", Weisbecker, 2013, Ph.D. thesis