SAND2019-7464C

Photos placed in horizontal position with even amount of white space between photos and header

SpaND: An Algebraic Sparsified Nested Dissection Algorithm Using Low-Rank Approximations

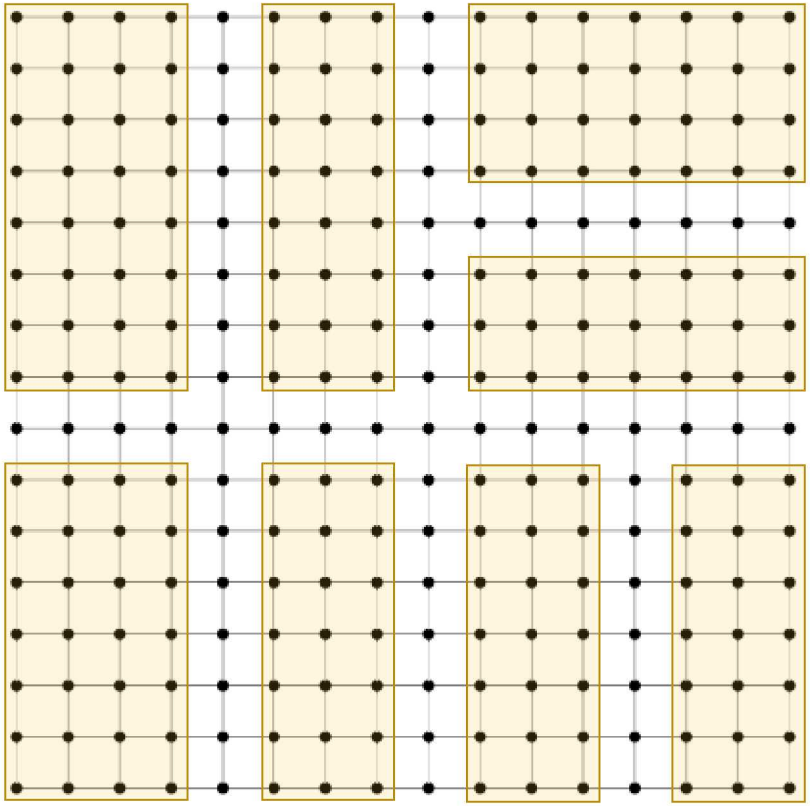
Leopold Cambier, Chao Chen, Erik Boman, Siva Rajamanickam, Ray Tuminaro, Eric Darve

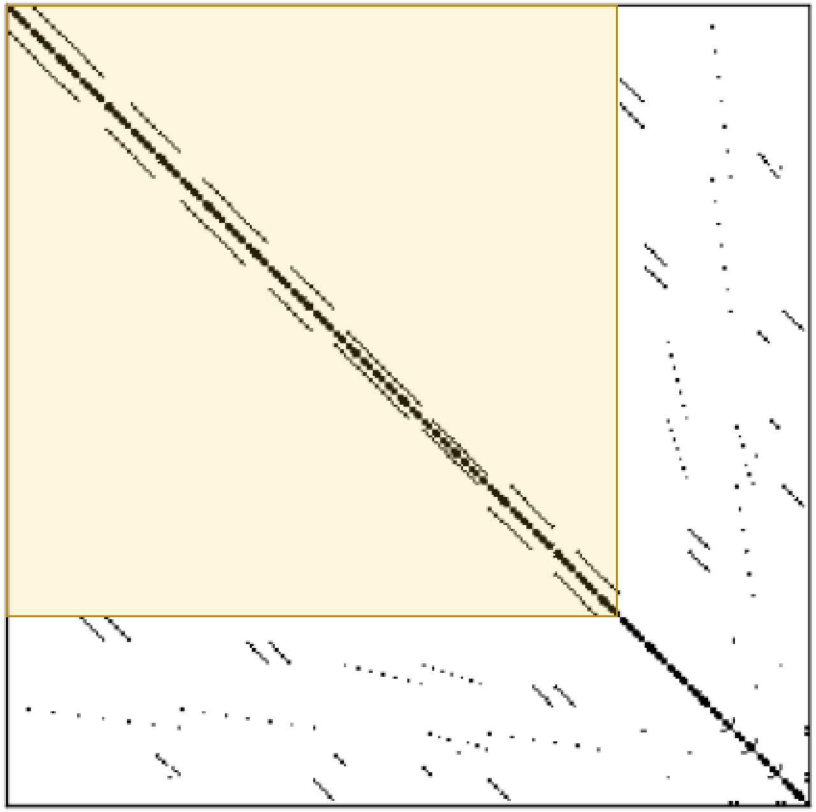
Preconditioning'19, July 1-3 2019

Motivation

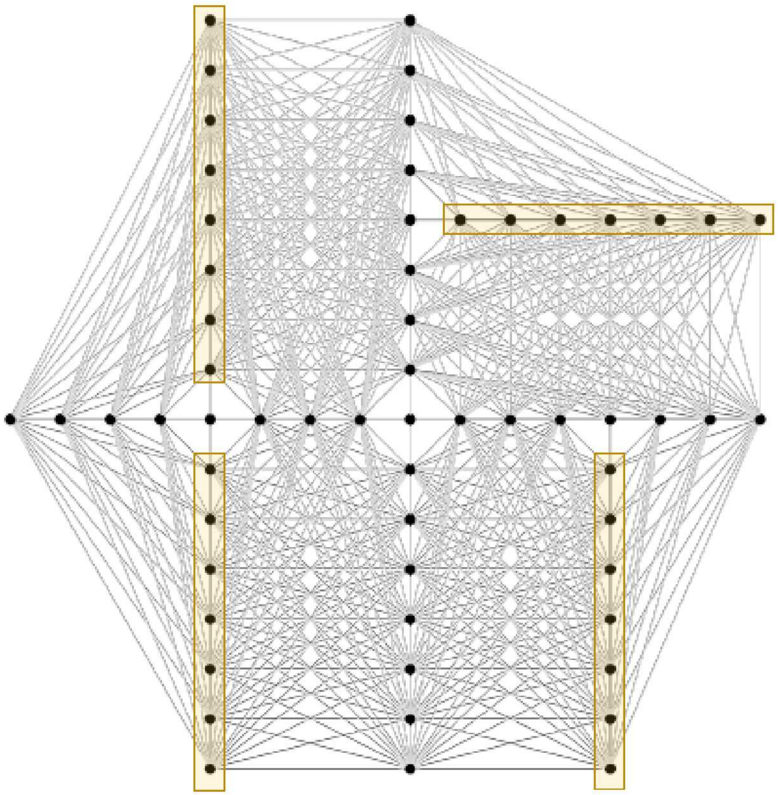
- Sparse direct factorization is too expensive in 3D
- Want robust “black-box” approximate factorization
  - Use as preconditioner
  - Allow trade-off fill versus quality
- Current methods are not scalable or not robust
  - Incomplete factorizations
  - Schur complement methods
  - Sparse approximate inverses
- SpaND: Collaboration with Stanford (Darve, Cambier, Chen)
  - Similar to HIF method (Ho & Ying)

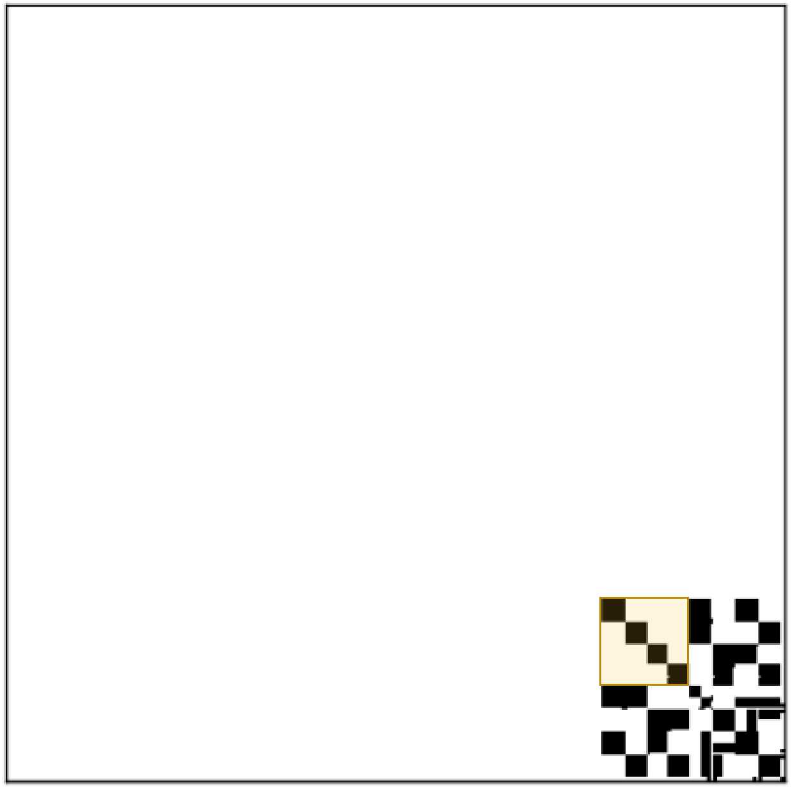
Sparse Factorization (1)



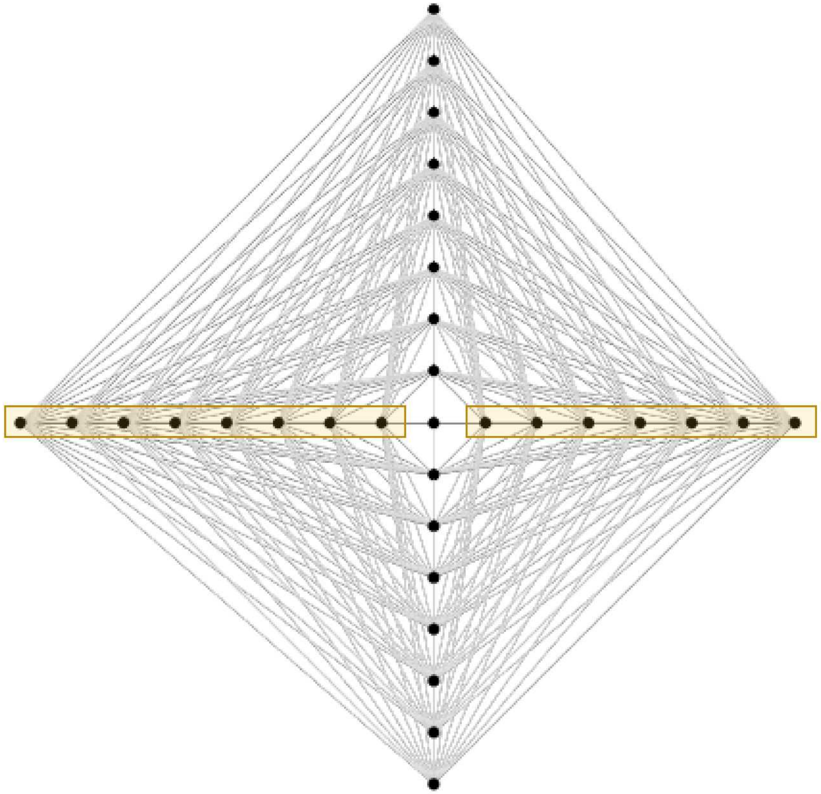


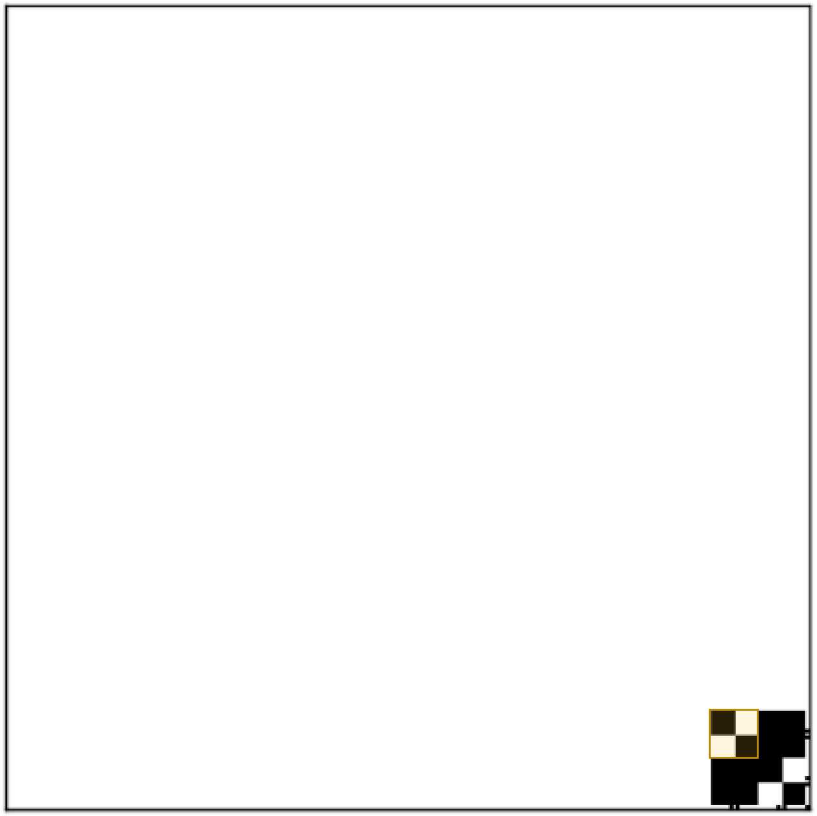
Sparse Factorization (2)



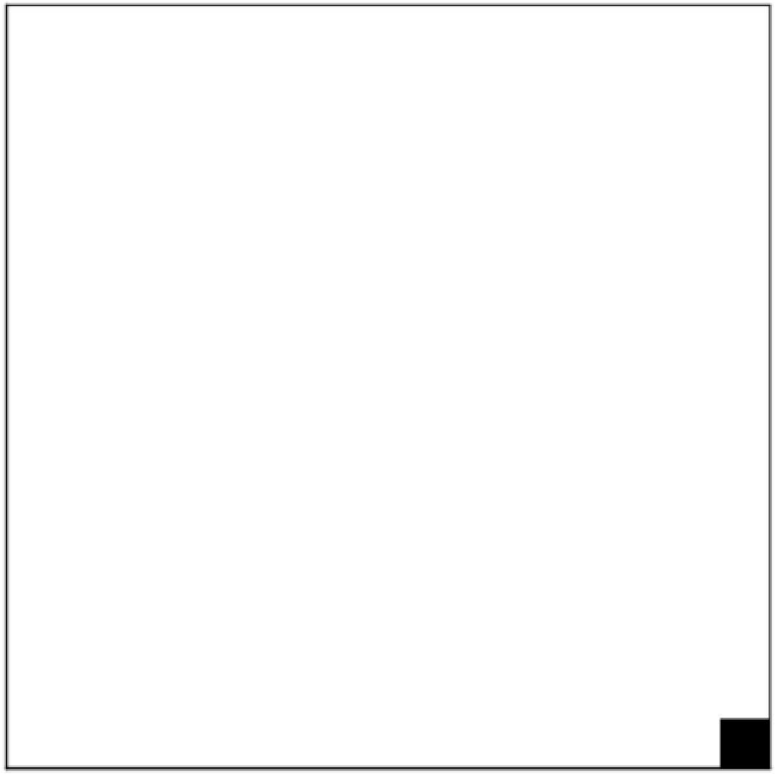


Sparse Factorization (3)

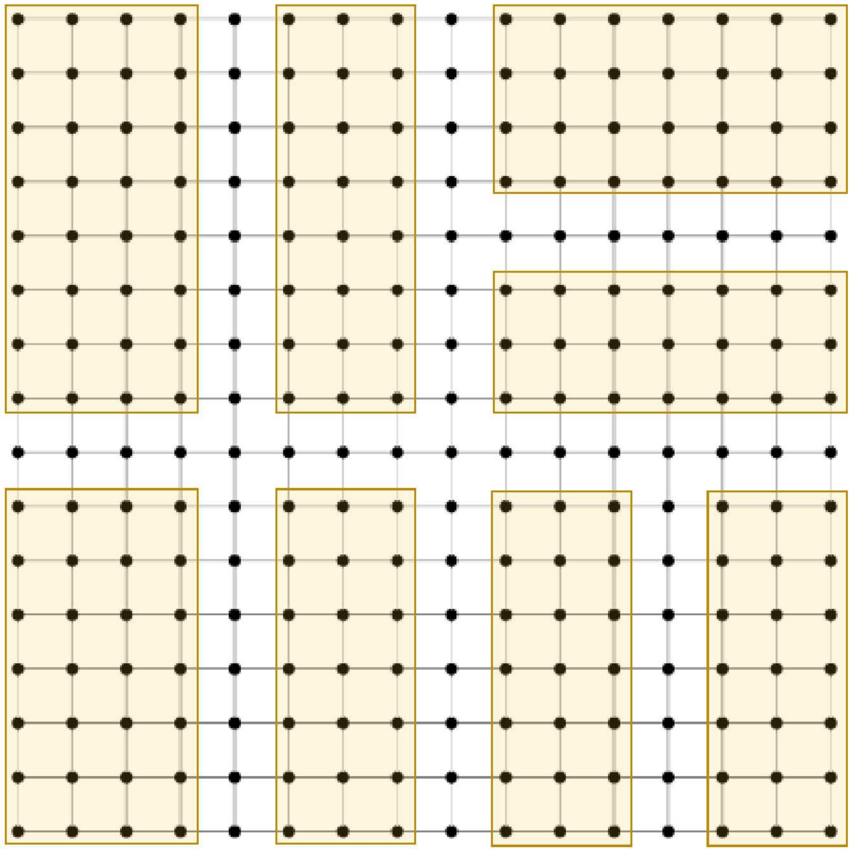


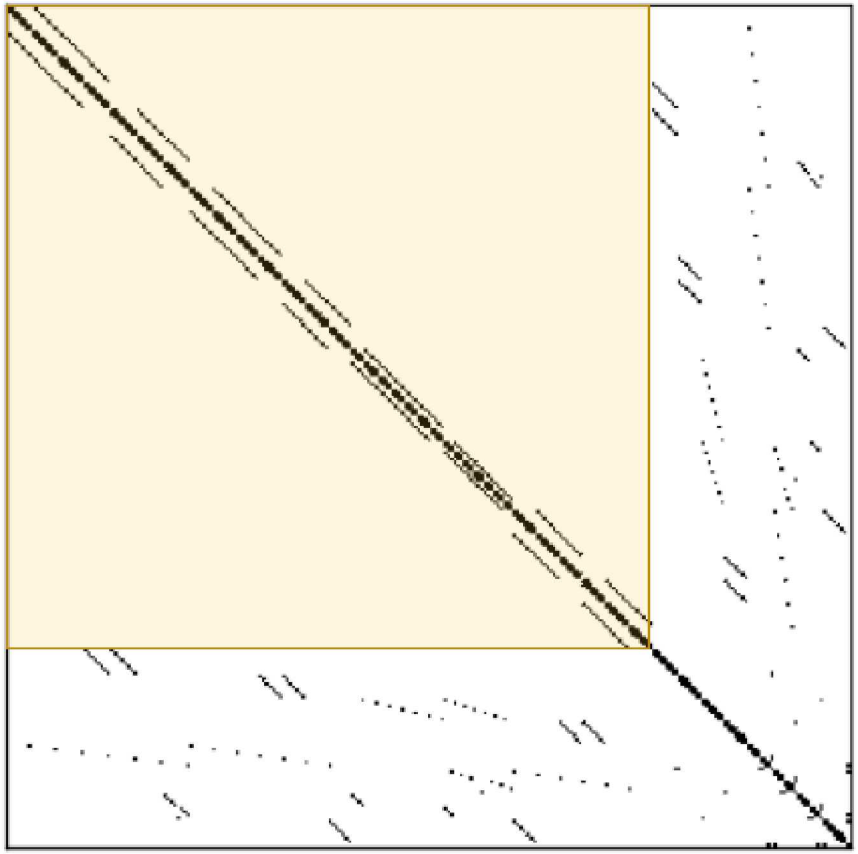


Sparse Factorization (4)

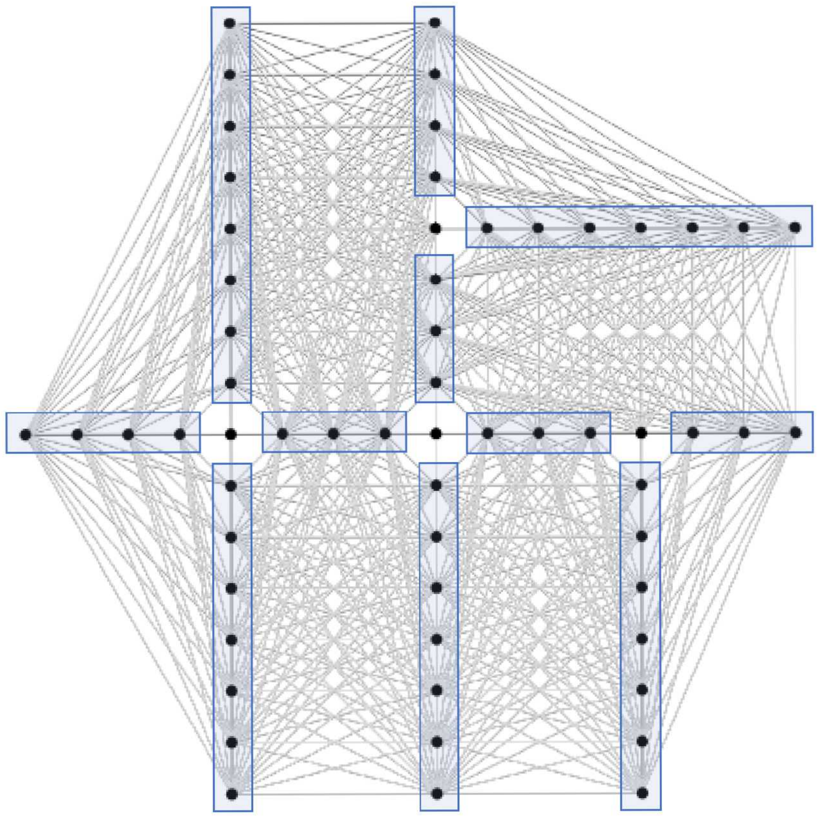


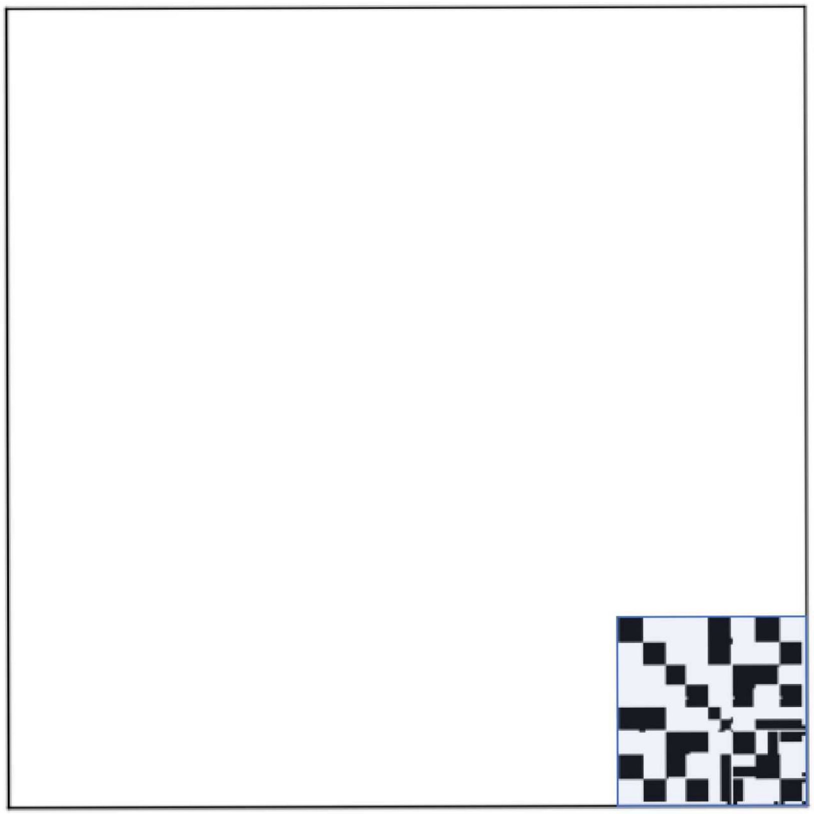
Sparsified Approx. Factorization (1)





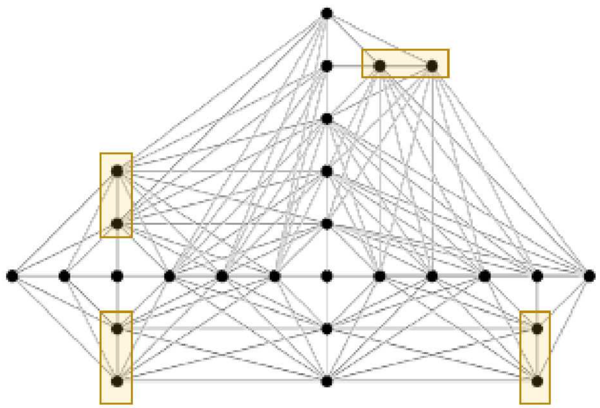
Sparsified Approx. Factorization (2)

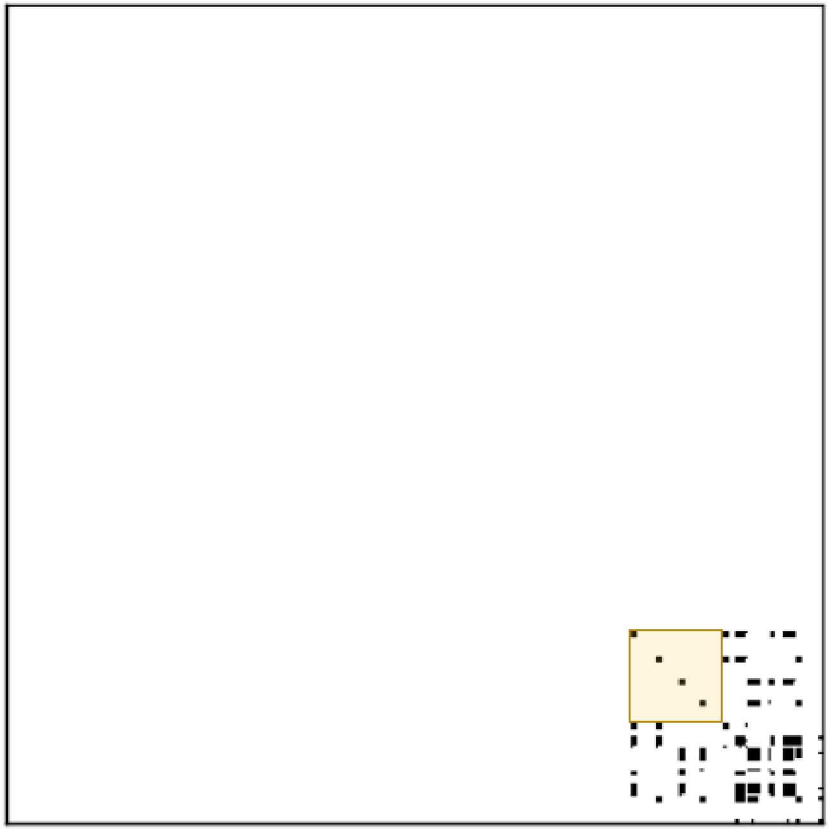




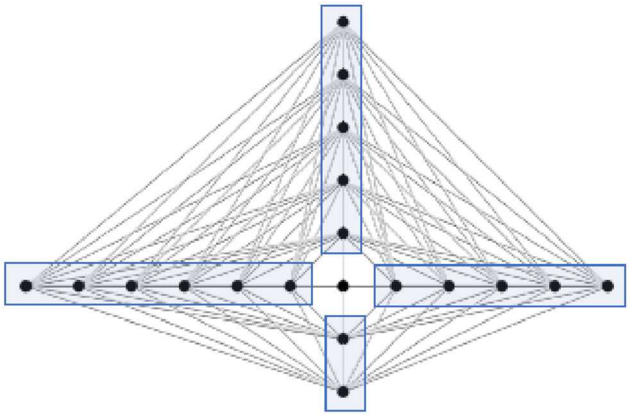
Sparsified Approx. Factorization (3)

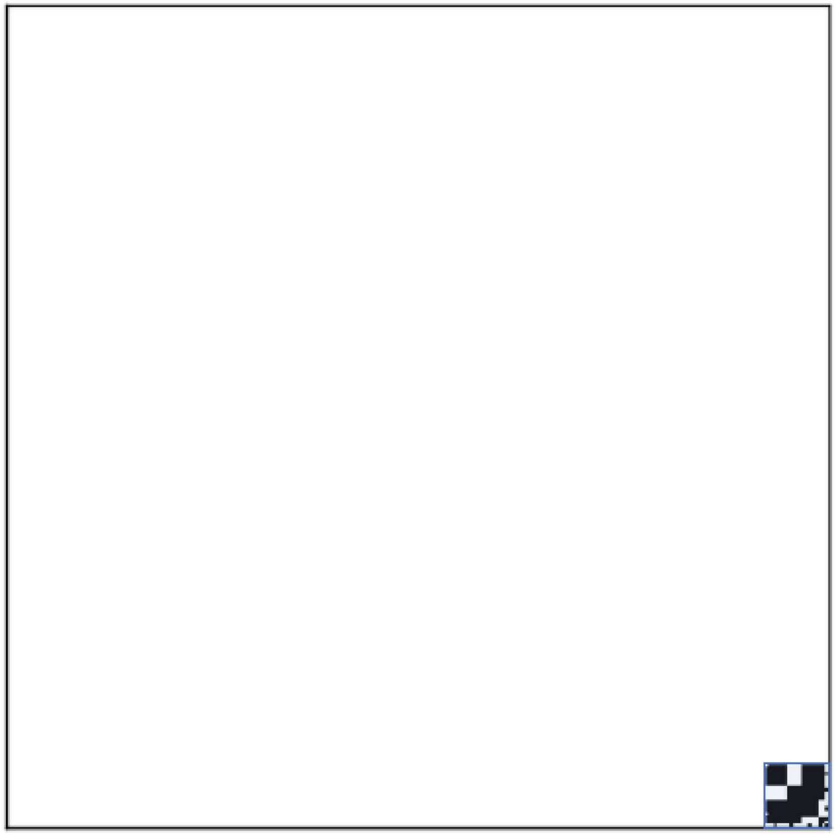
≈





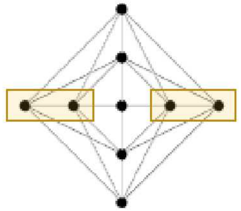
Sparsified Approx. Factorization (4)

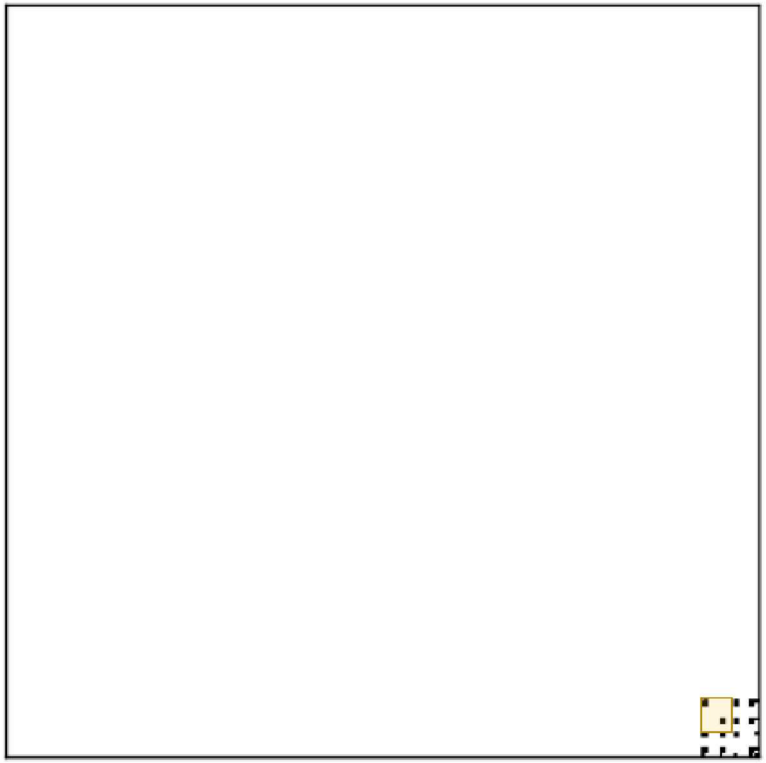




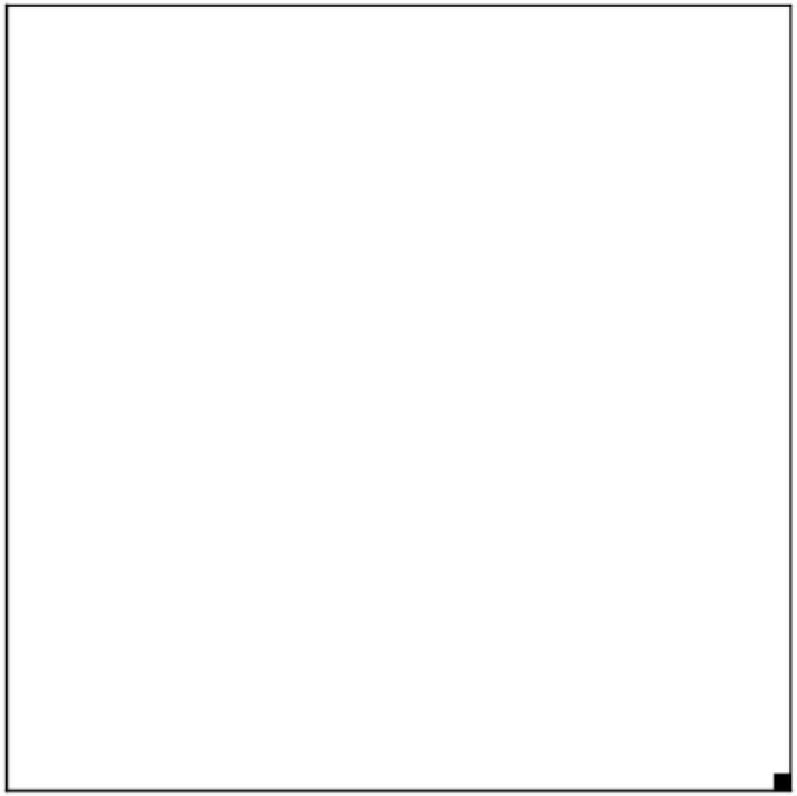
Sparsified Approx. Factorization (5)

$\approx$



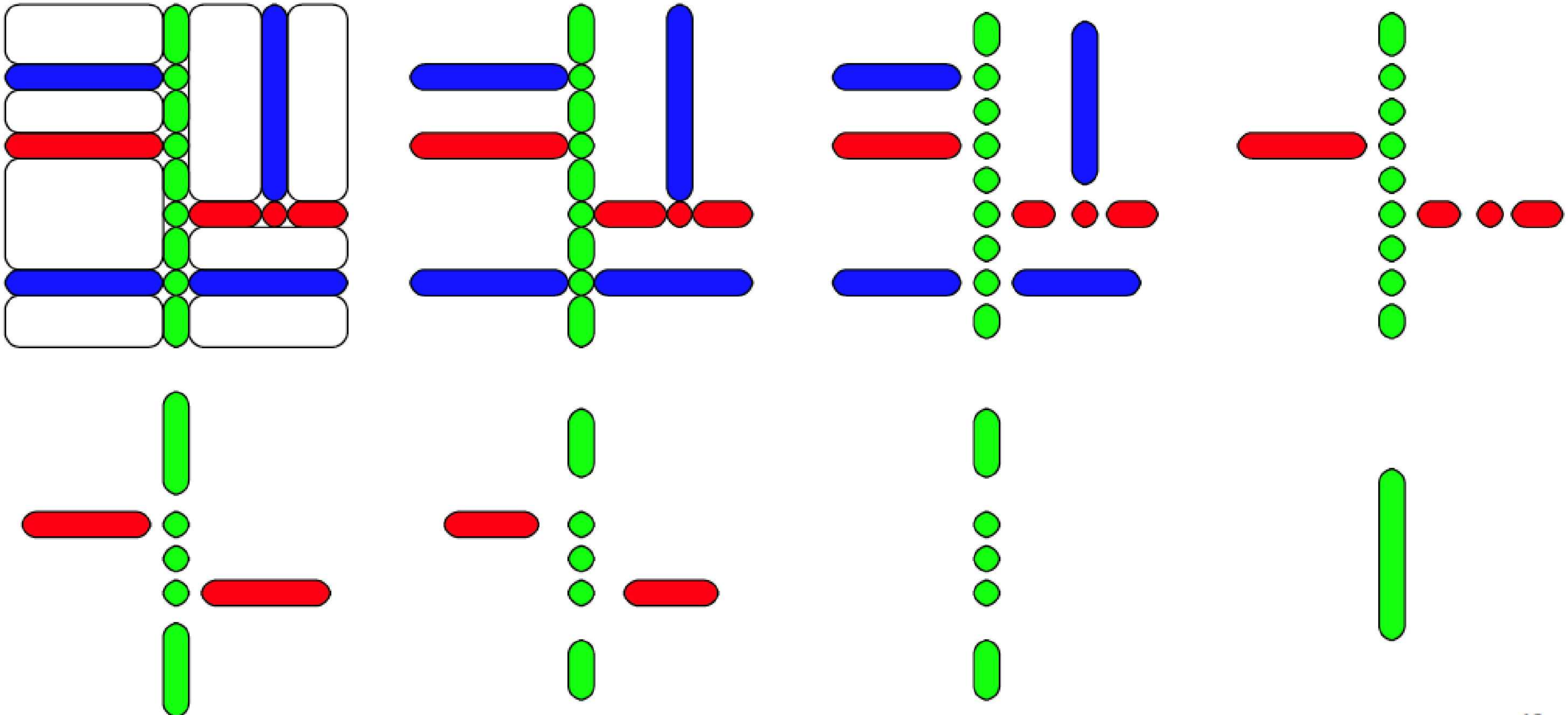


Sparsified Approx. Factorization (6)



SpaND Summary

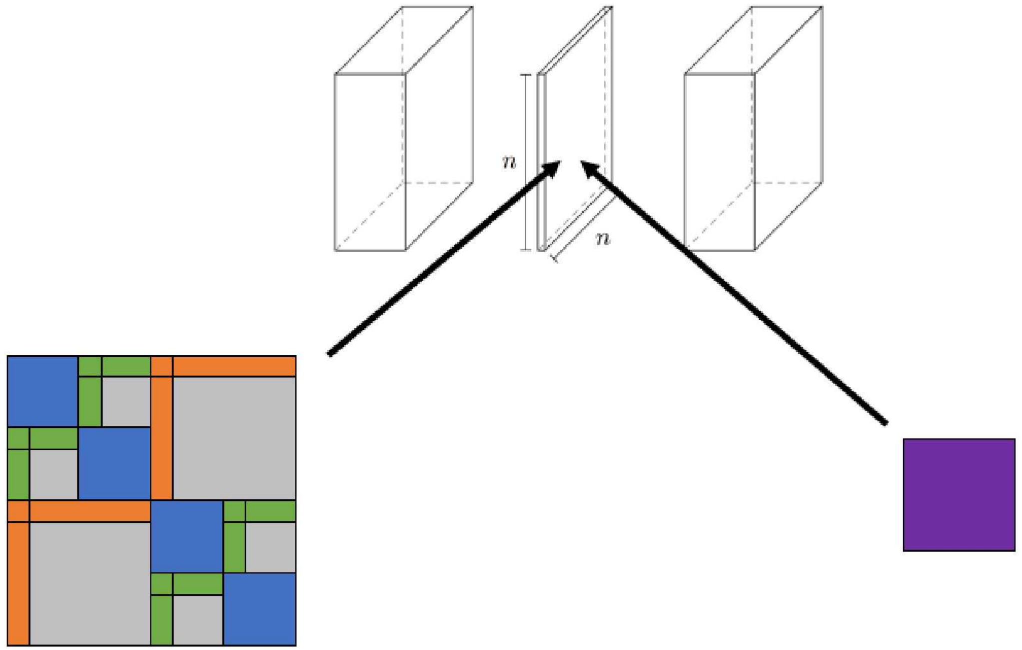
- Sparsify separators (low-rank compression) during elimination



12

Different from fast-algebra on dense

- Common approach: Fast algebra (H/HSS/BLR) on dense blocks
  - Ex: Strumpack, MUMPS, PasTix, etc.
- Instead we reduce the size of the separator blocks!



Sparsification Step

- Block scaling, low-rank elimination, drop negligible blocks

$$\underbrace{\begin{bmatrix} L_{ss}^{-1} & & \\ & I & \\ & & I \end{bmatrix}\begin{bmatrix} A_{ss} & & A_{sn} \\ & A_{ww} & A_{wn} \\ A_{ns} & A_{nw} & A_{nn} \end{bmatrix}\begin{bmatrix} L_{ss}^{-\top} & & \\ & I & \\ & & I \end{bmatrix}}$$

$$\underbrace{\begin{bmatrix} Q^{\top} & & \\ & I & \\ & & I \end{bmatrix}\begin{bmatrix} I & & A_{sn} \\ & A_{ww} & A_{wn} \\ A_{ns} & A_{nw} & A_{nn} \end{bmatrix}\begin{bmatrix} Q & & \\ & I & \\ & & I \end{bmatrix}}$$

$$\begin{bmatrix} I & & & \varepsilon \\ & I & & W_{cn} \\ & & A_{ww} & A_{wn} \\ \varepsilon & W_{cn}^{\top} & A_{nw} & A_{nn} \end{bmatrix}$$

Sparsification via Low-rank Approx.

We need low-rank approximation of off-diagonal (rectangular) block.

1. Interpolative decomposition (ID)
   - Use RRQR (QRCP)
   - Aka skeletonization
2. Orthogonal transform
   - More stable, but more expensive

Sparsification 1: ID

(1) We start with

$$\begin{bmatrix} A_{ss} & & A_{sn} \\ & A_{ww} & A_{wn} \\ A_{ns} & A_{nw} & A_{nn} \end{bmatrix}$$

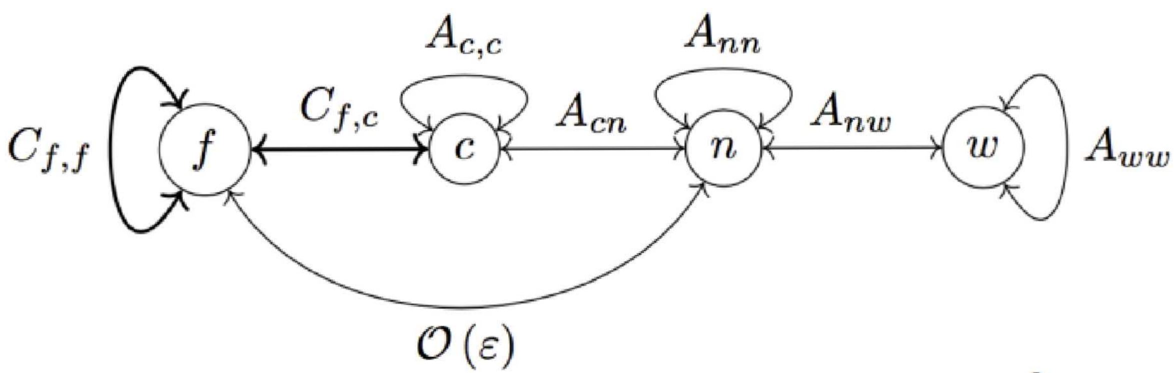
(2) We then approximate

$$A_{sn} = \begin{pmatrix} T_{fc} \\ I \end{pmatrix} A_{cn} + \varepsilon$$

$$s = f \cup c$$

(3) We end up with

$$\begin{bmatrix} C_{ff} & C_{fc} & & \varepsilon \\ C_{cf} & A_{cc} & & A_{cn} \\ & & A_{ww} & A_{wn} \\ \varepsilon & A_{nc} & A_{nw} & A_{nn} \end{bmatrix}$$



8

Sparsification 2: Orthogonal

(1) We start with

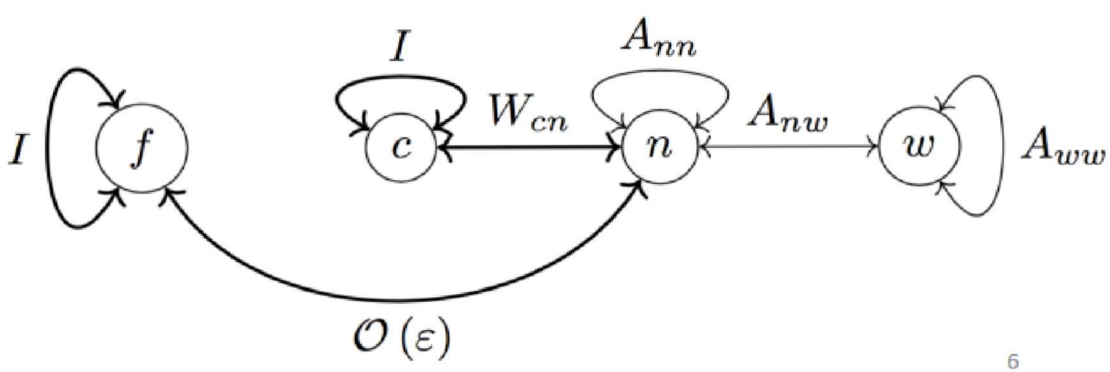
$$\begin{bmatrix} I & & A_{sn} \\ & A_{ww} & A_{wn} \\ A_{ns} & A_{nw} & A_{nn} \end{bmatrix}$$

(2) We then approximate

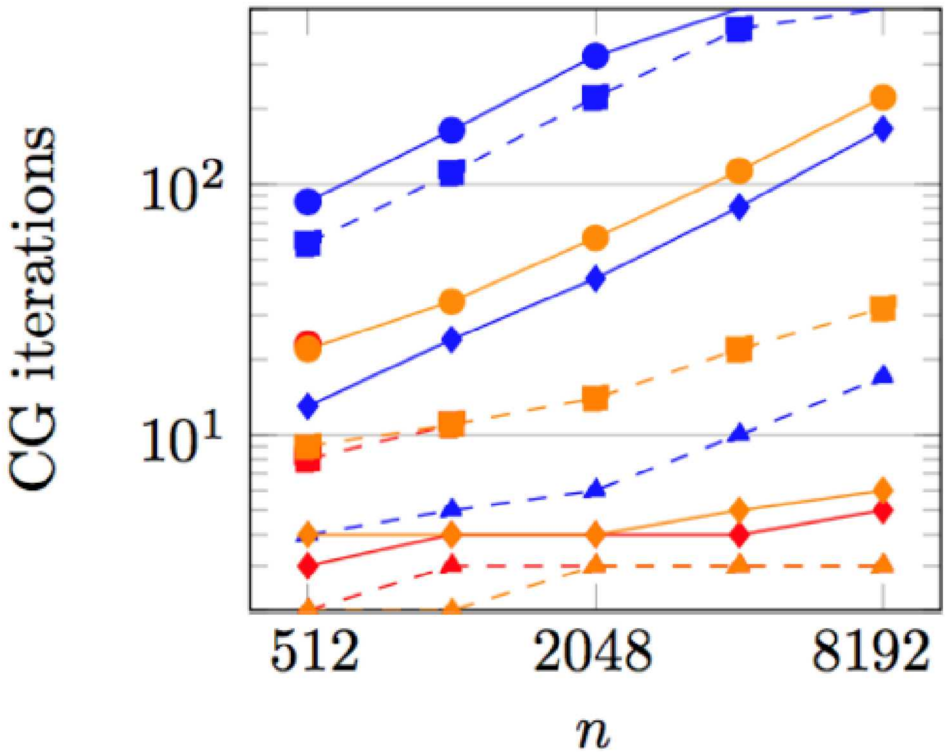
$$A_{sn} = Q_{sc} W_{cn} + \varepsilon$$
$$Q^{\top} s = f \cup c$$

(3) We end up with

$$\begin{bmatrix} I & & & \varepsilon \\ & I & & W_{cn} \\ & & A_{ww} & A_{wn} \\ \varepsilon & W_{cn}^{\top} & A_{nw} & A_{nn} \end{bmatrix}$$



Results: 2D Laplacians



Interpolative, no scaling

Interpolative, with scaling

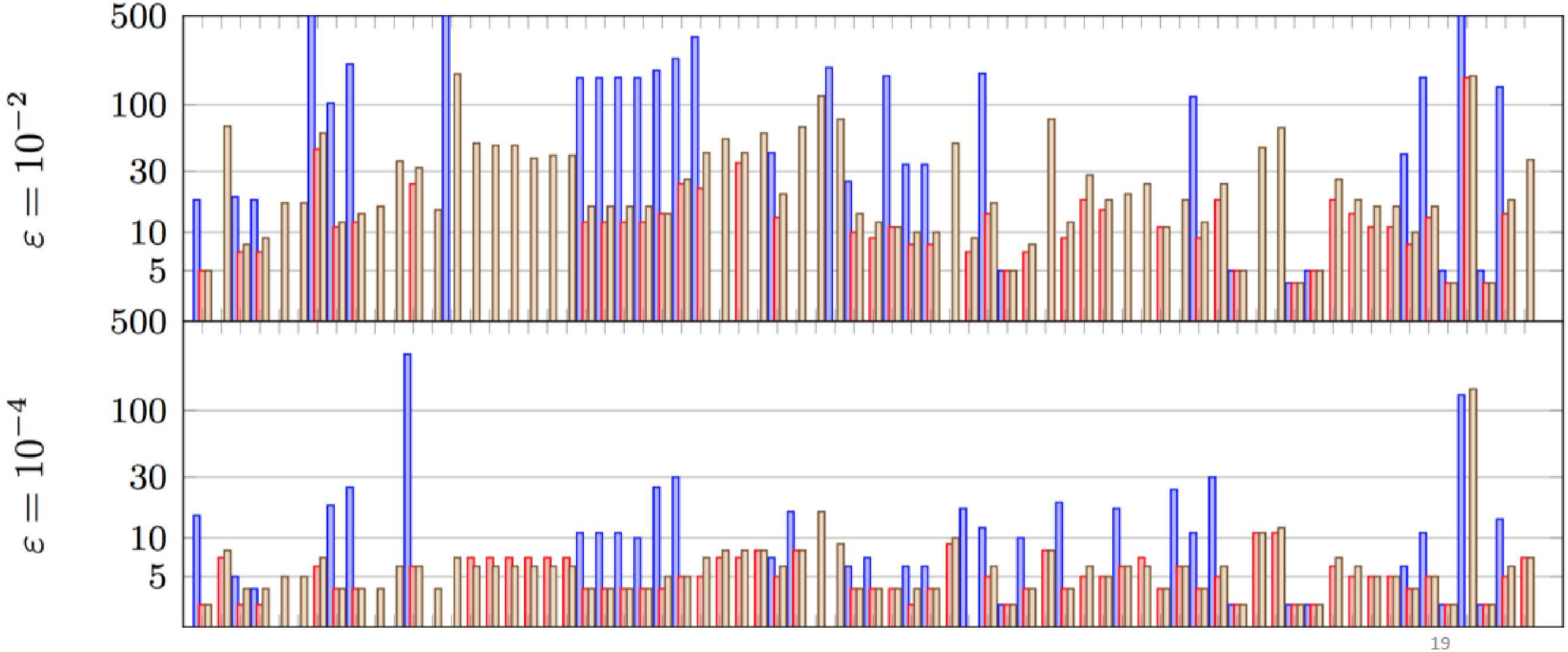
Orthogonal, with scaling

$$\varepsilon = 10^{-1} \rightarrow 10^{-6}$$

Results: SuiteSparse Collection

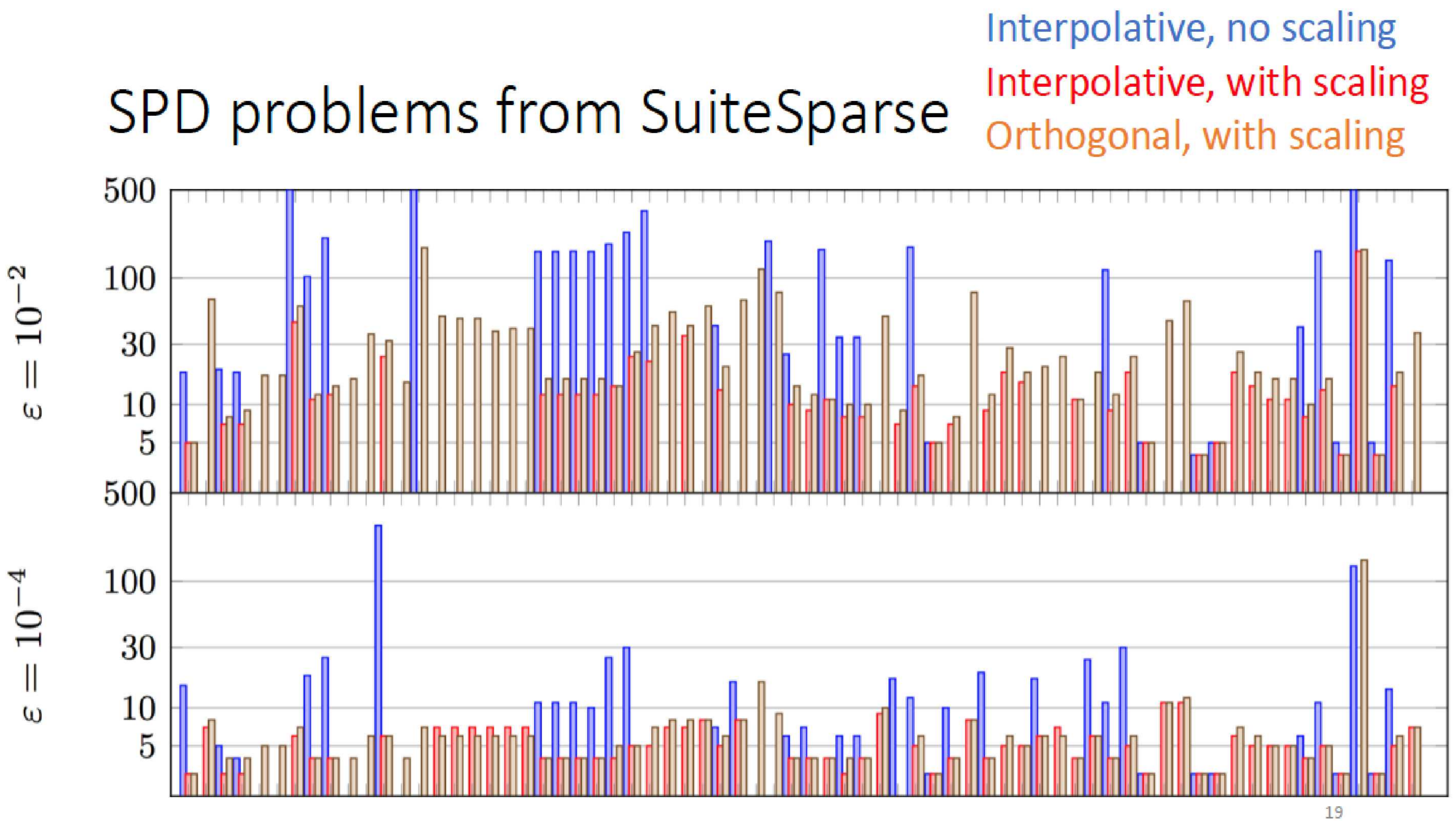
SPD problems from SuiteSparse

Interpolative, no scaling
Interpolative, with scaling
Orthogonal, with scaling



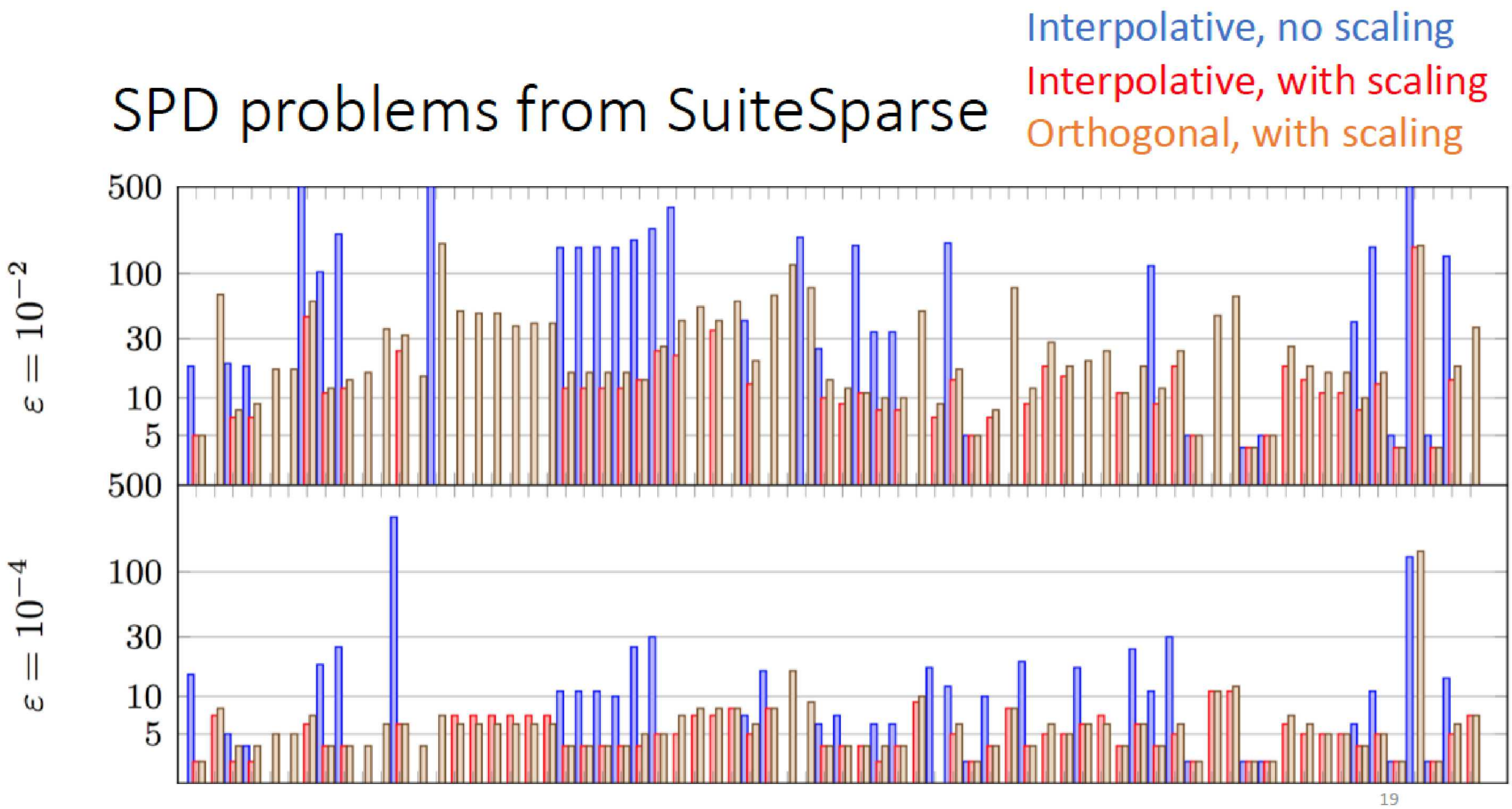
Results: Performance Profile

SPD problems from SuiteSparse



Results:

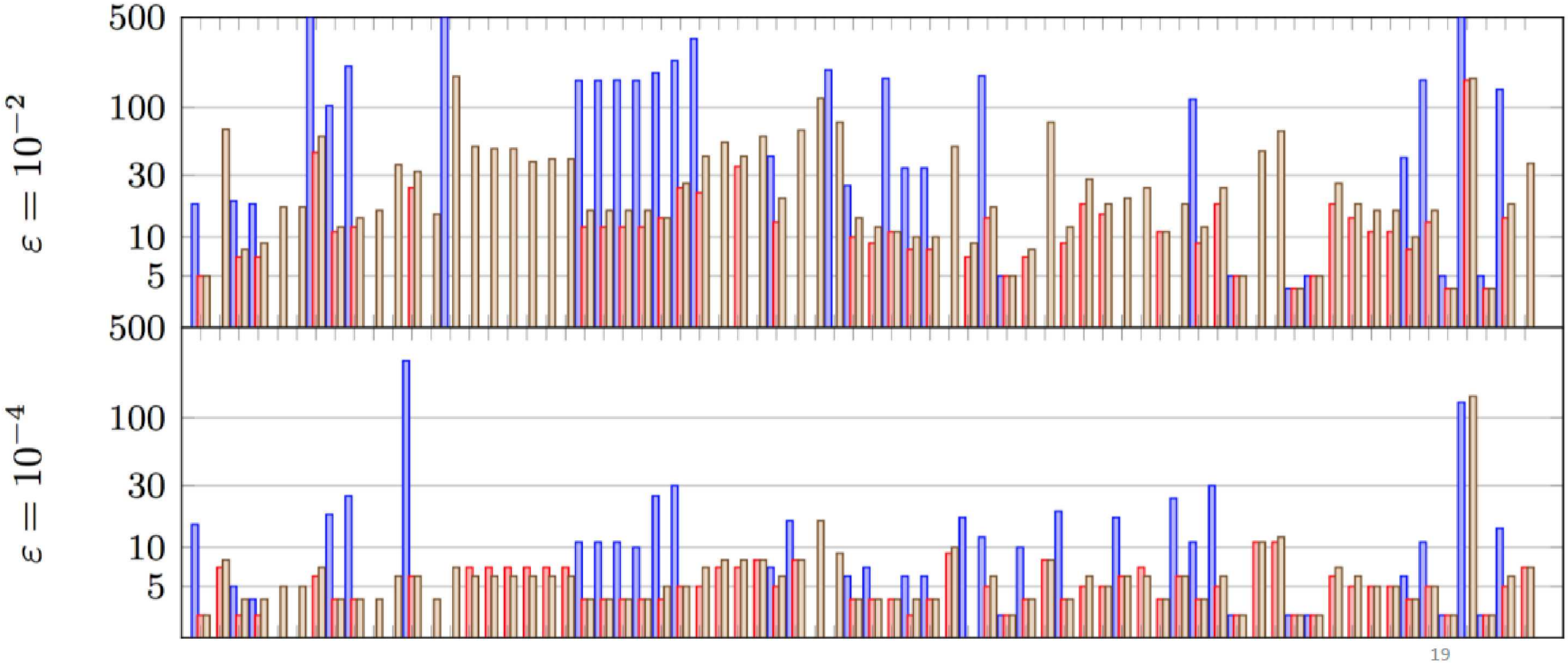
SPD problems from SuiteSparse



Results: SPE

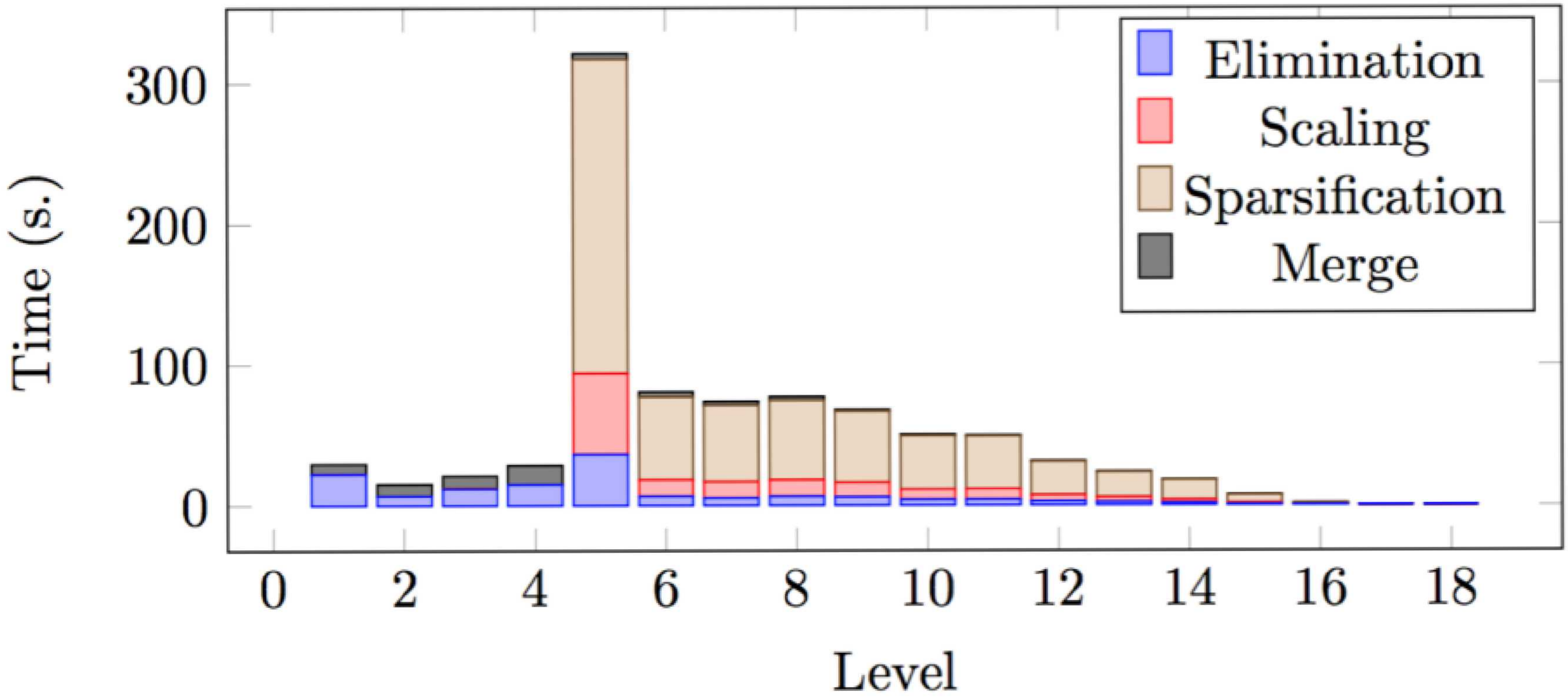
SPD problems from SuiteSparse

Interpolative, no scaling
Interpolative, with scaling
Orthogonal, with scaling



Profiling

- Most expensive part is sparsification (RRQR)
- Skip sparsification on bottom levels (no benefit)



Conclusions

- SpaND is an approximate factorization
  - combines features from sparse direct and hierarchical matrices
- Tunable trade-off factorization cost and preconditioner quality
  - Observed near-linear scaling on many problems
- Based on HIF but several improvements
- We focused on SPD case (Cholesky) but
  - Method can be generalized to nonsymmetric (LU)

References

- SpaND
- HIF