ICME Stanford Institute for Computational & Mathematical Engineering

TOTAL

A General Sparsified Nested Dissection Algorithm

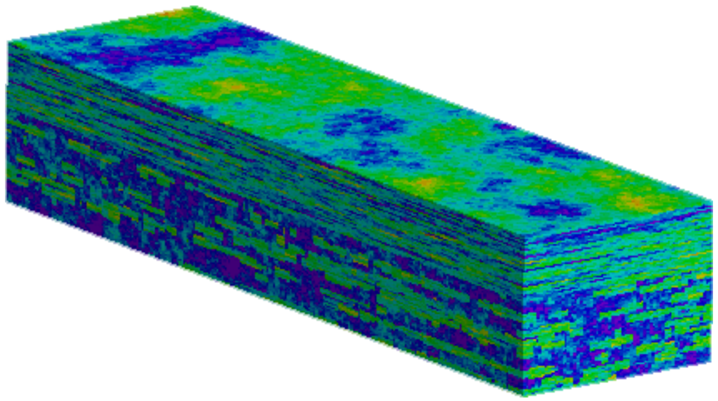
Léopold Cambier

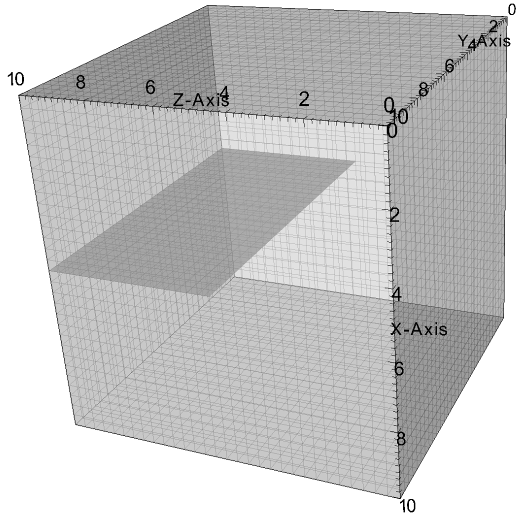
with B. Klockiewicz, E. Darve, C. Chen,
E. Boman, S. Rajamanickam, R. Tuminaro

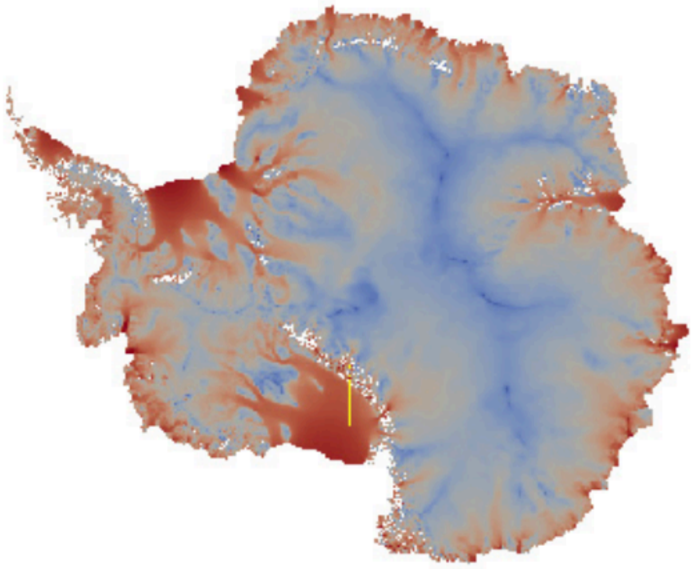
October 16, 2019

Problem and Motivation

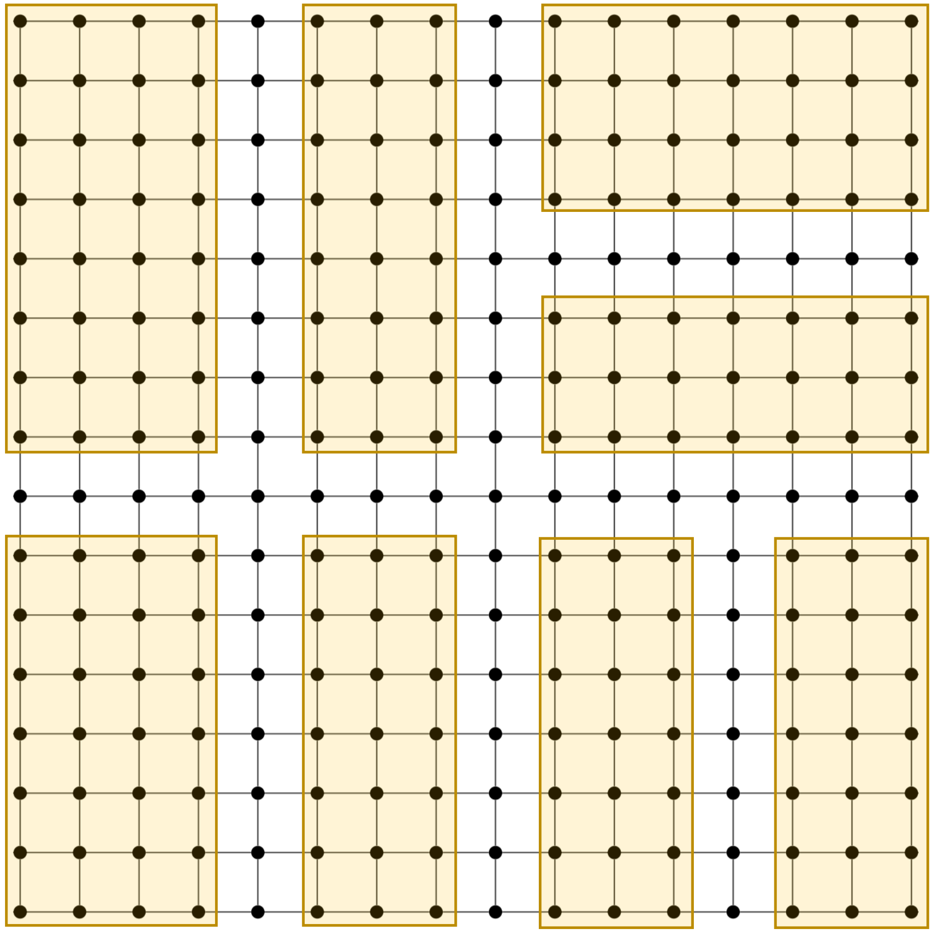
- Ax = b, A is sparse and (typically) from PDE's
- Generic and algebraic (ε only) and scalable (multilevels, O(N log N))
- Direct methods: too expensive
- Preconditioner: custom and sometimes hard to generalize

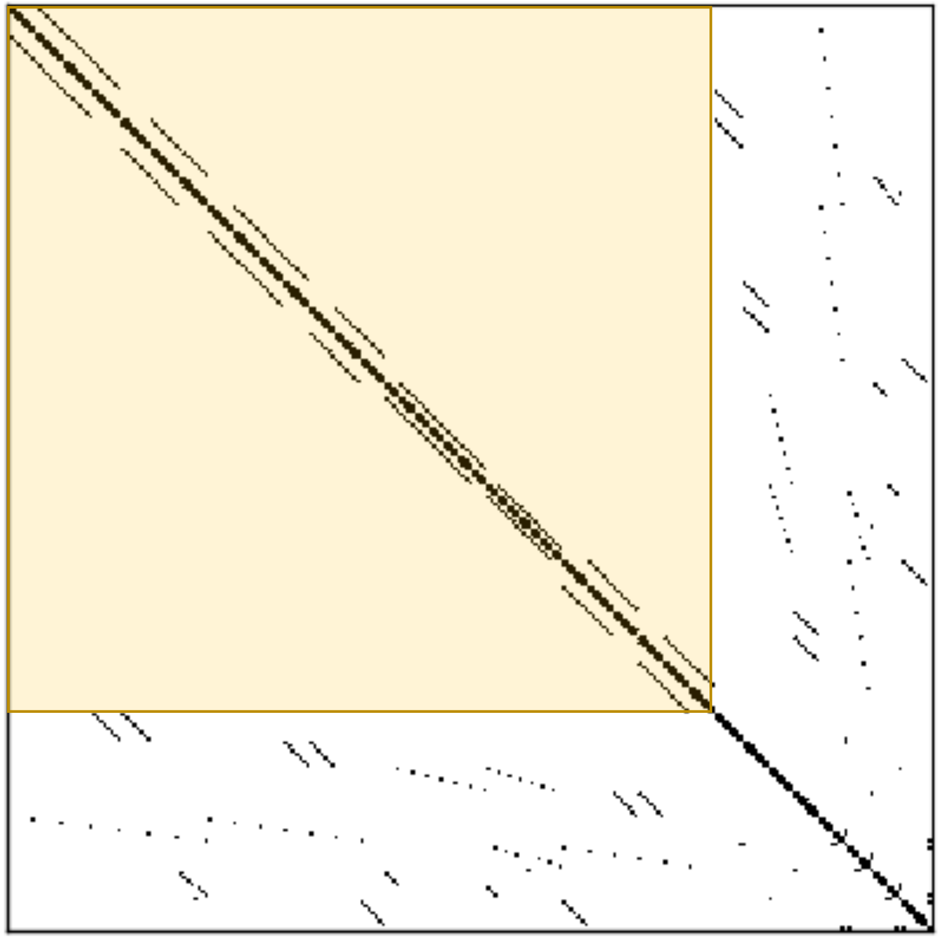




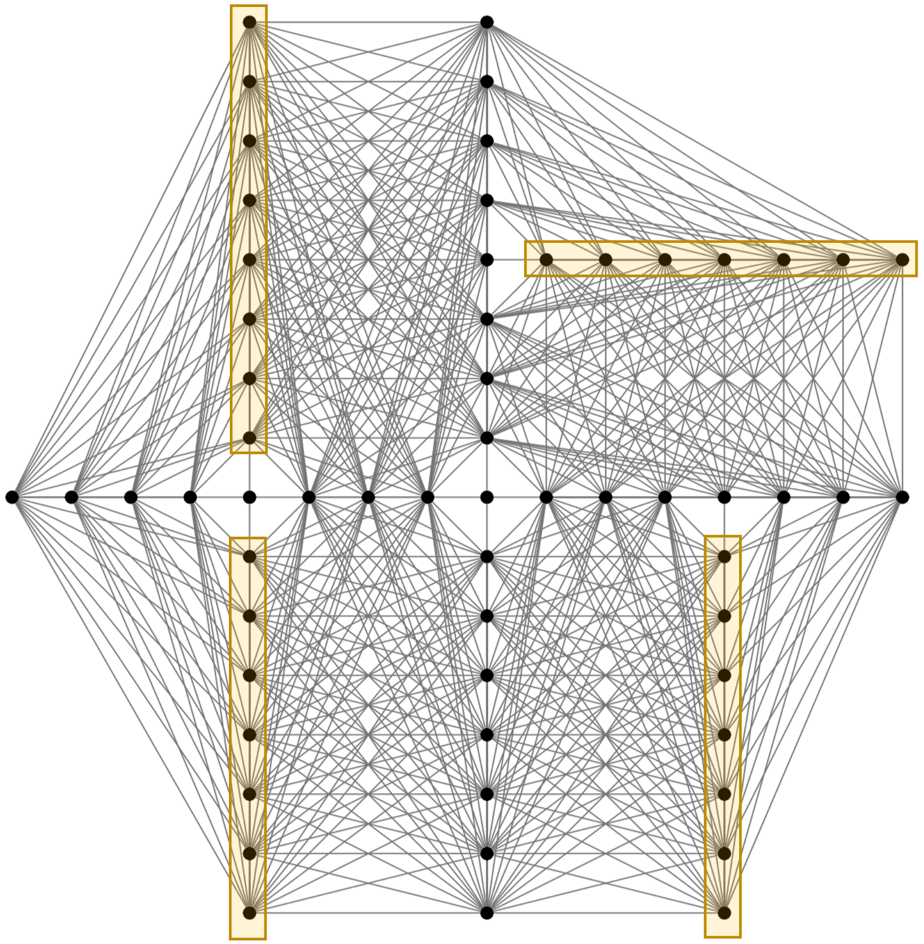


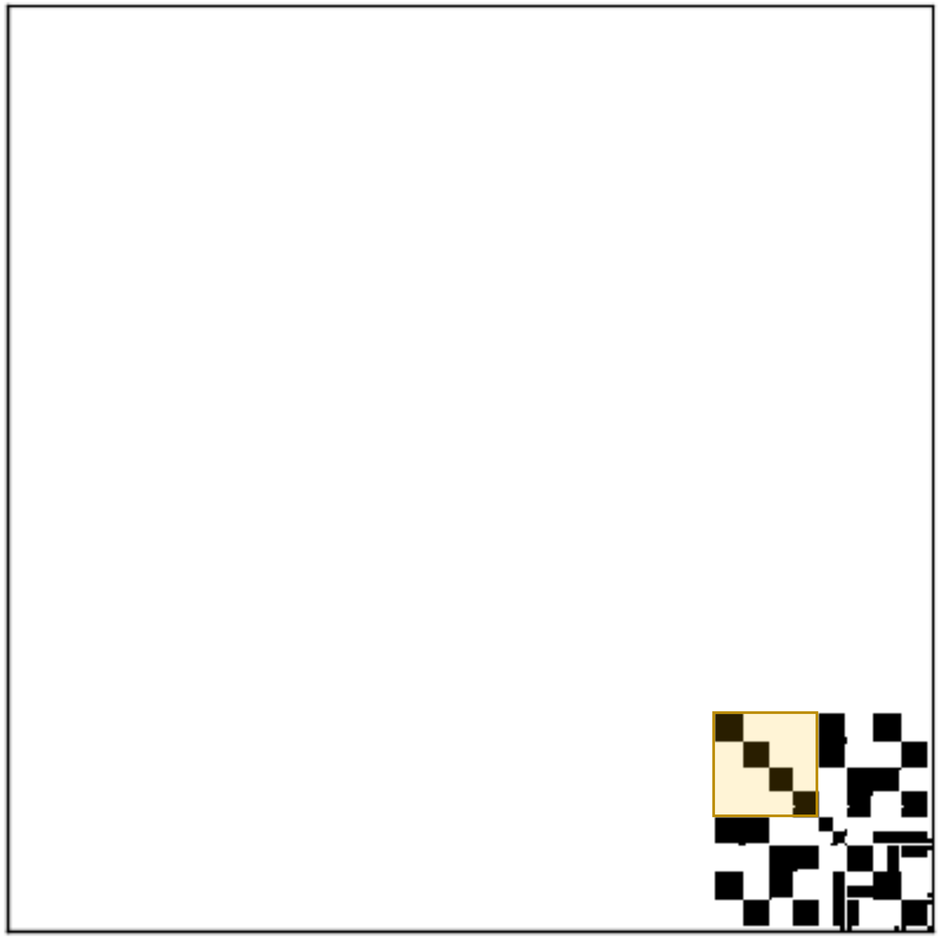
Sparse Linear Systems (1)



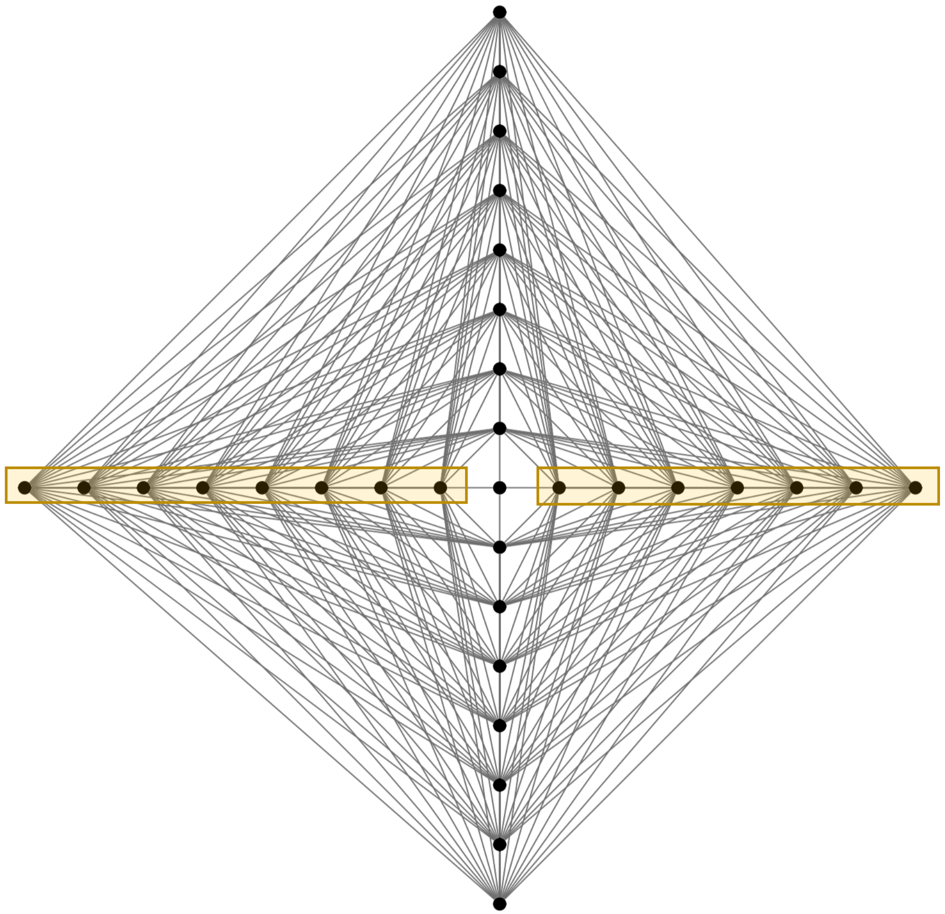


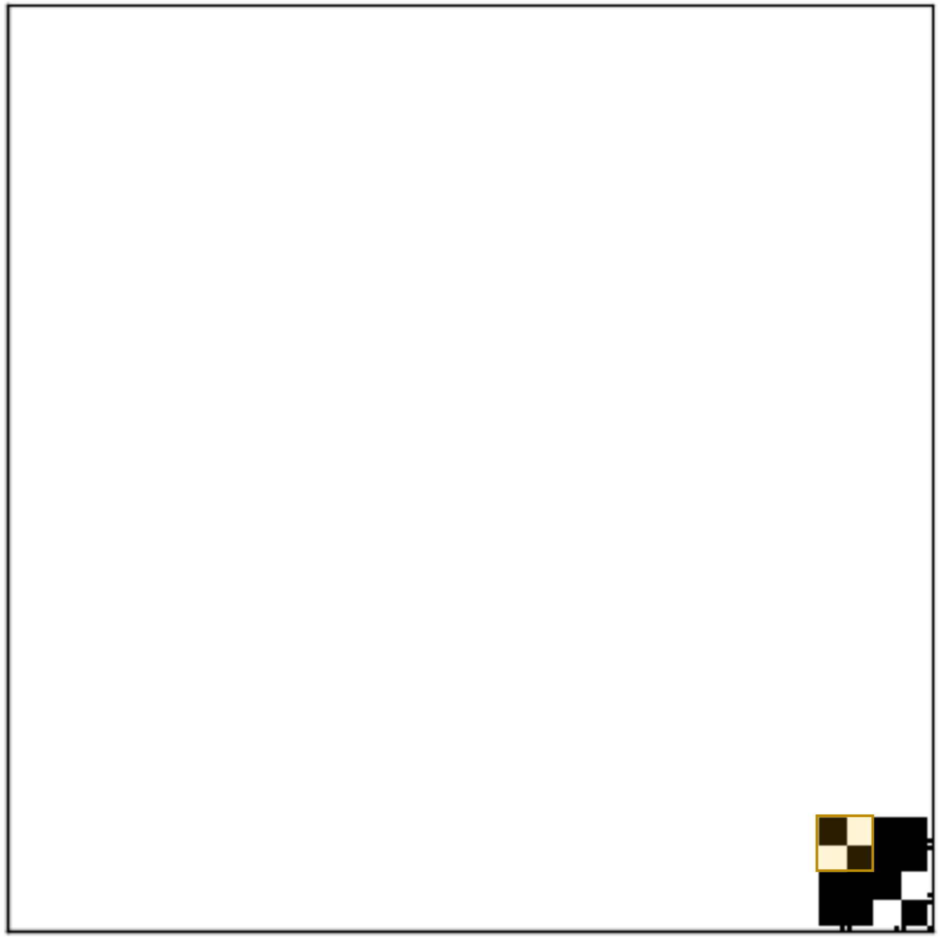
Sparse Linear Systems (2)



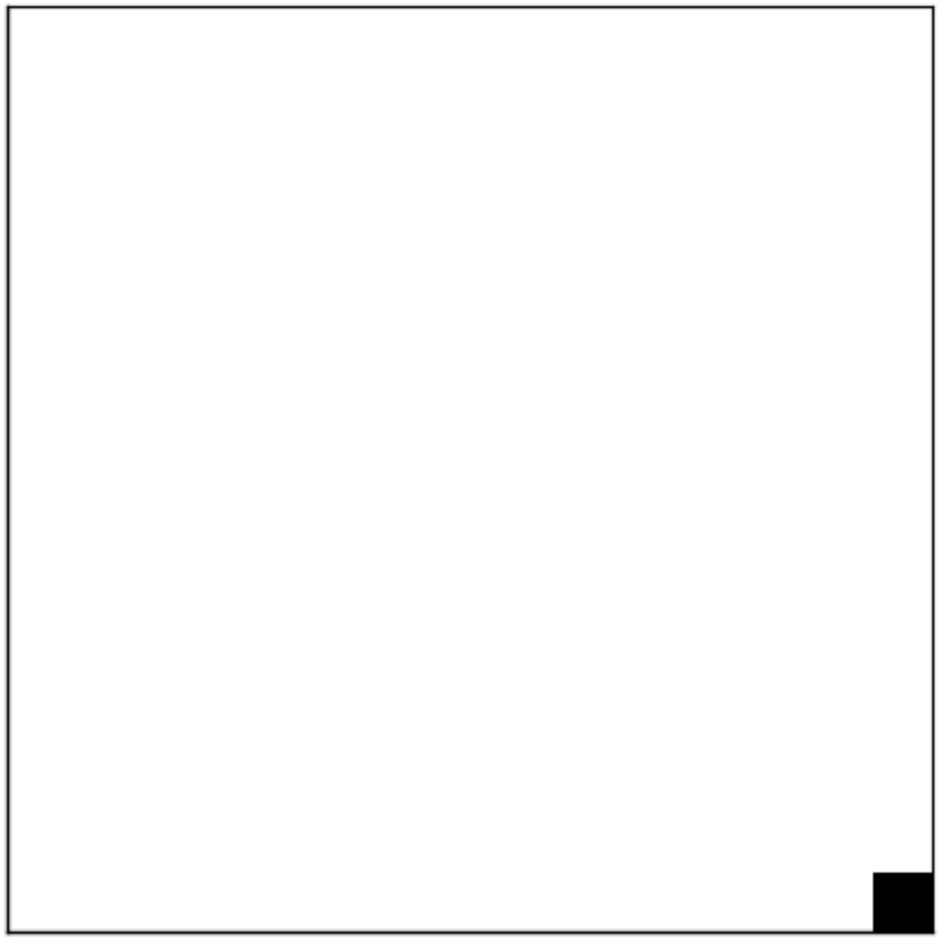


Sparse Linear Systems (3)

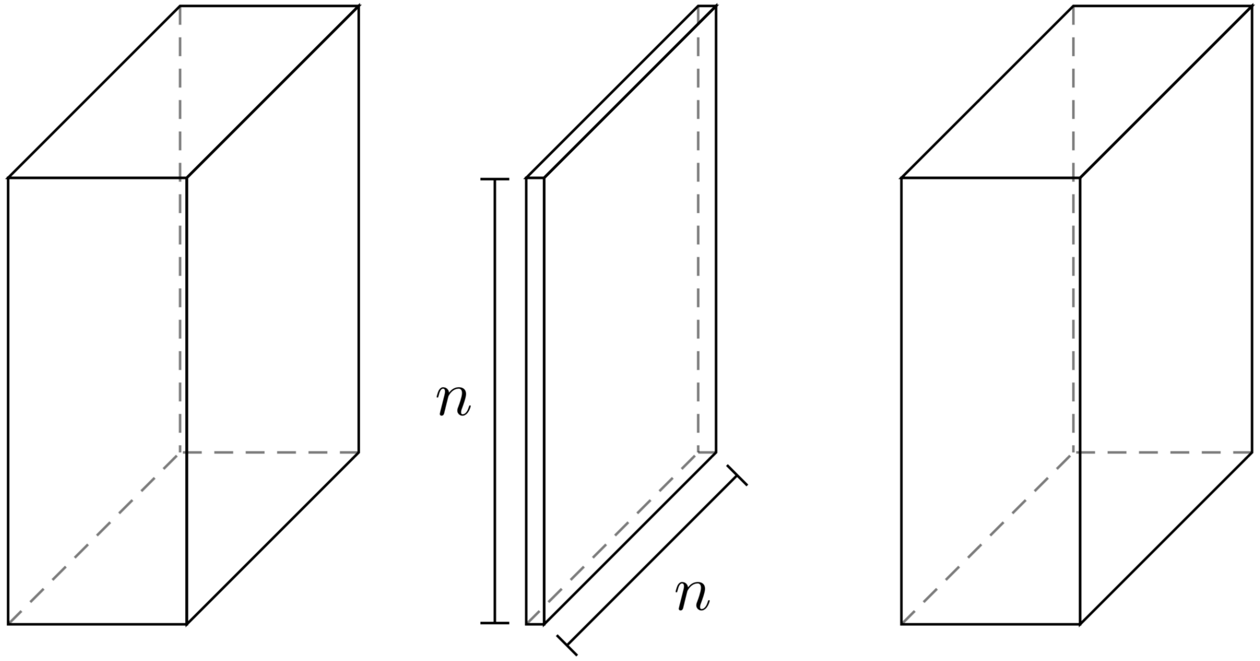




Sparse Linear Systems (4)



Nested Dissection

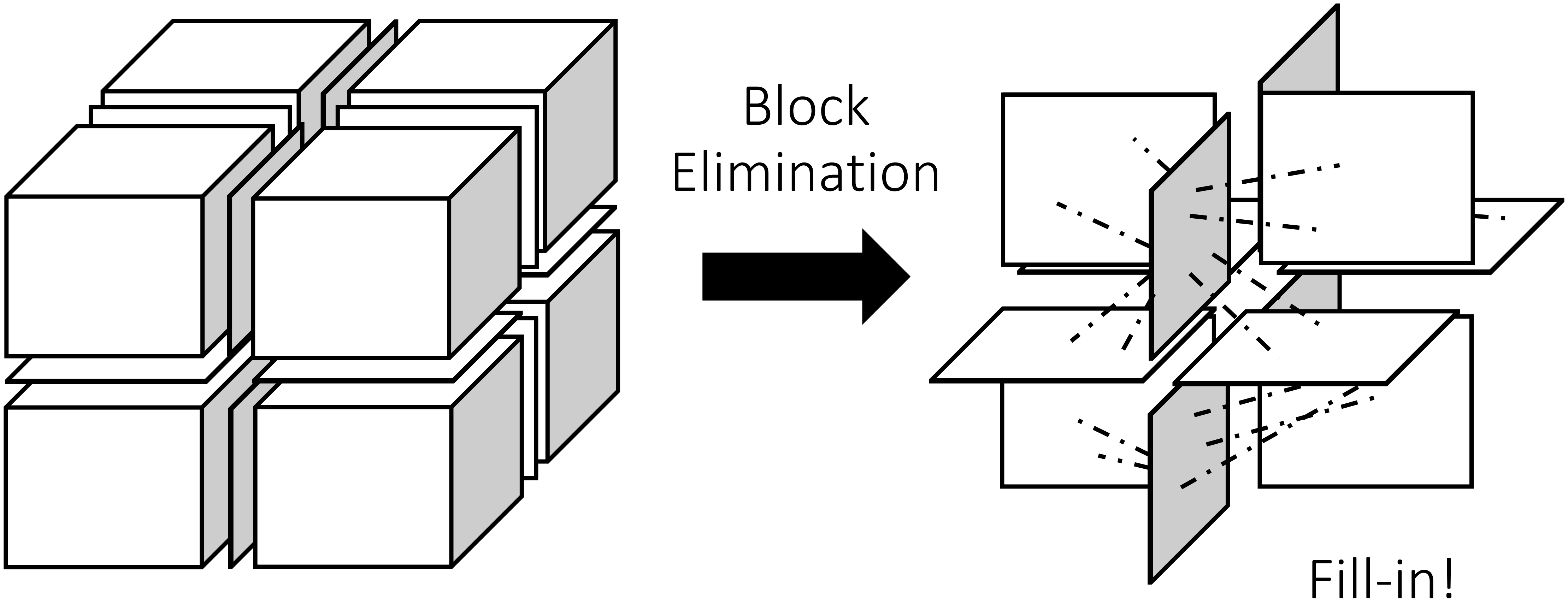


Too much fill-in!

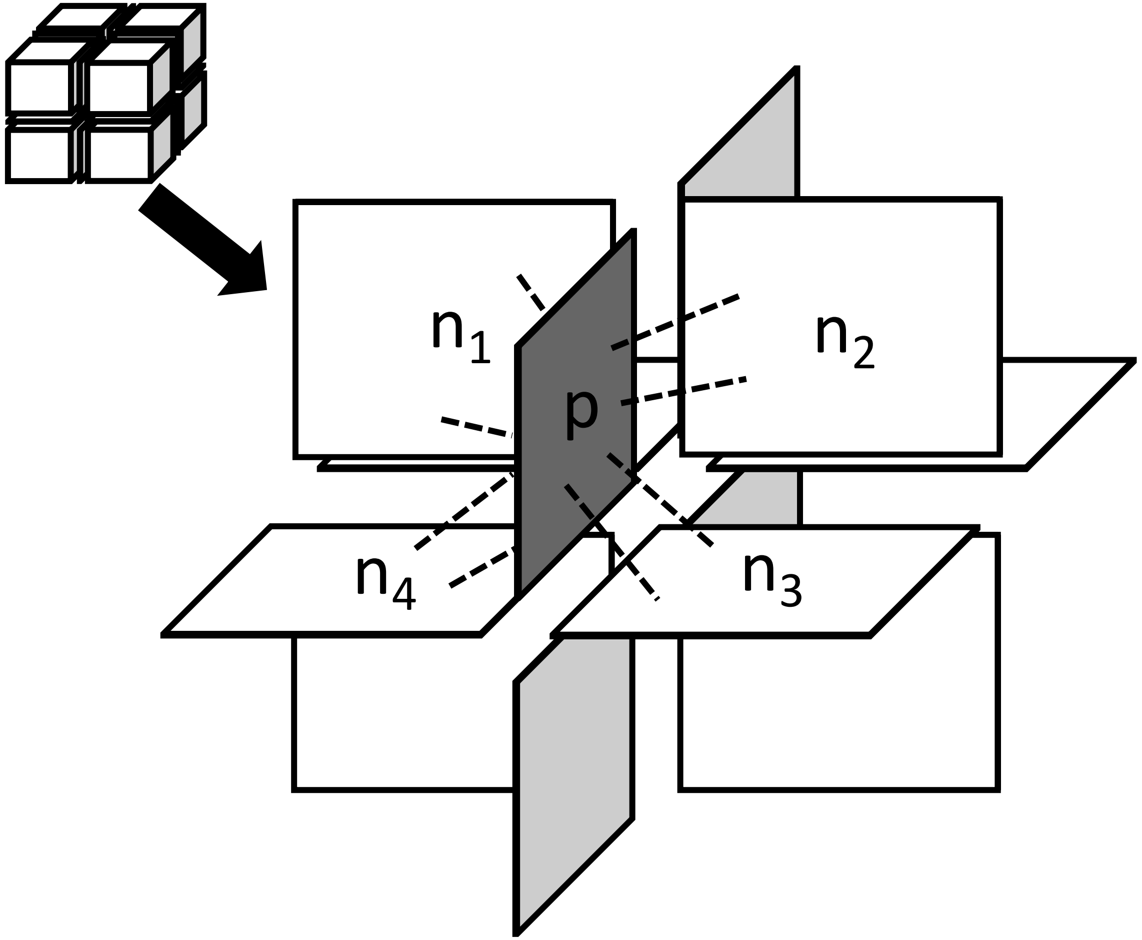
Factoring $n^2 = N^{\frac{2}{3}}$ takes $O(N^2)$

Nested Dissection Ellimination

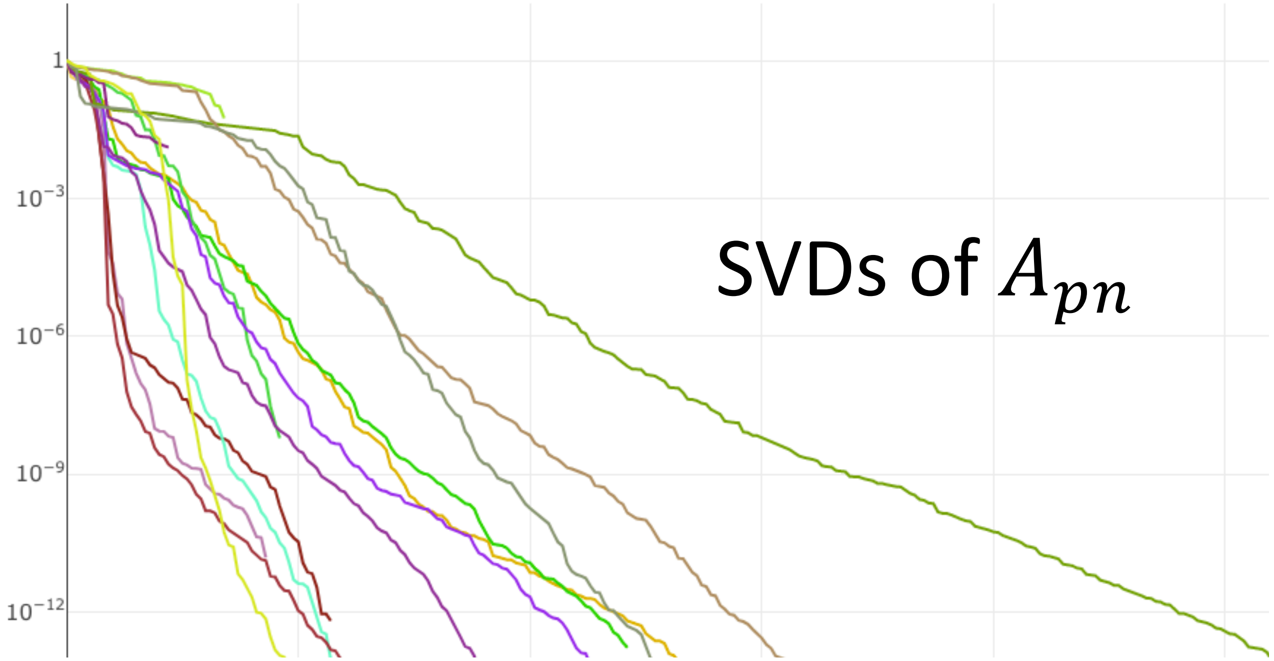
$$L^{-1}\begin{bmatrix} A_{pp} & A_{pn} & \\ A_{np} & A_{nn} & A_{nw} \\ & A_{wn} & A_{ww} \end{bmatrix} U^{-1} = \begin{bmatrix} I & & \\ & A_{nn} - A_{ns}A_{ss}^{-1}A_{sn} & A_{nw} \\ & A_{wn} & A_{ww} \end{bmatrix}$$



Sparsification



$$A_{pn} = [A_{pn_1} \quad A_{pn_2} \quad A_{pn_3} \quad \ldots]$$
$$= Q_c W_{cn} + Q_f W_{fn}$$
$$\approx Q_c W_{cn}$$

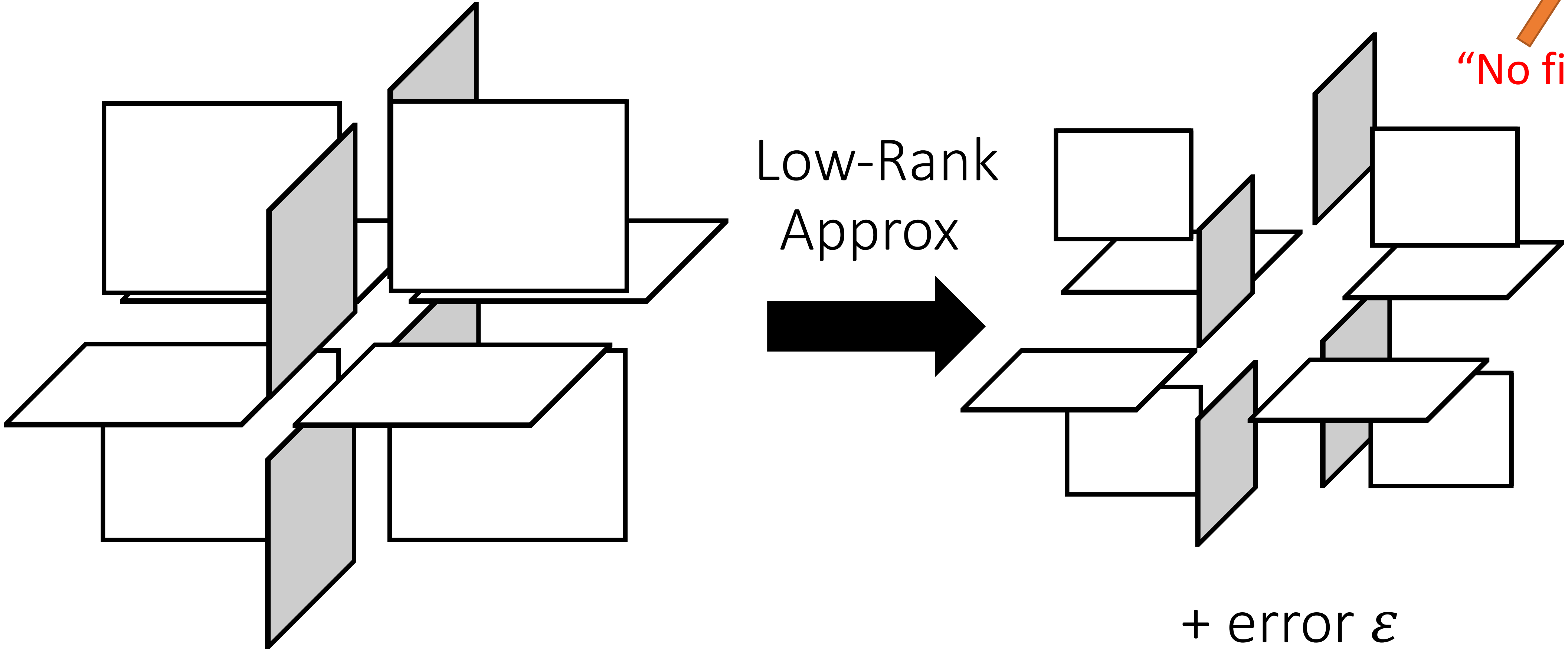


Sparsification

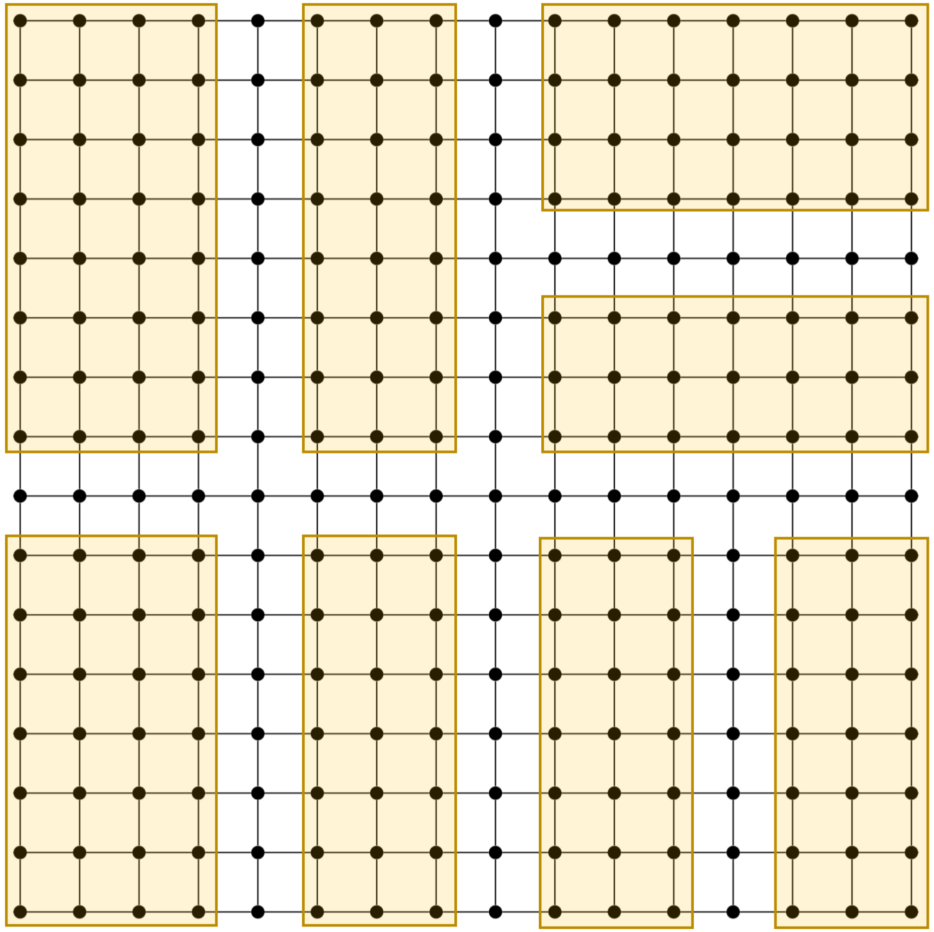
“Eliminated”

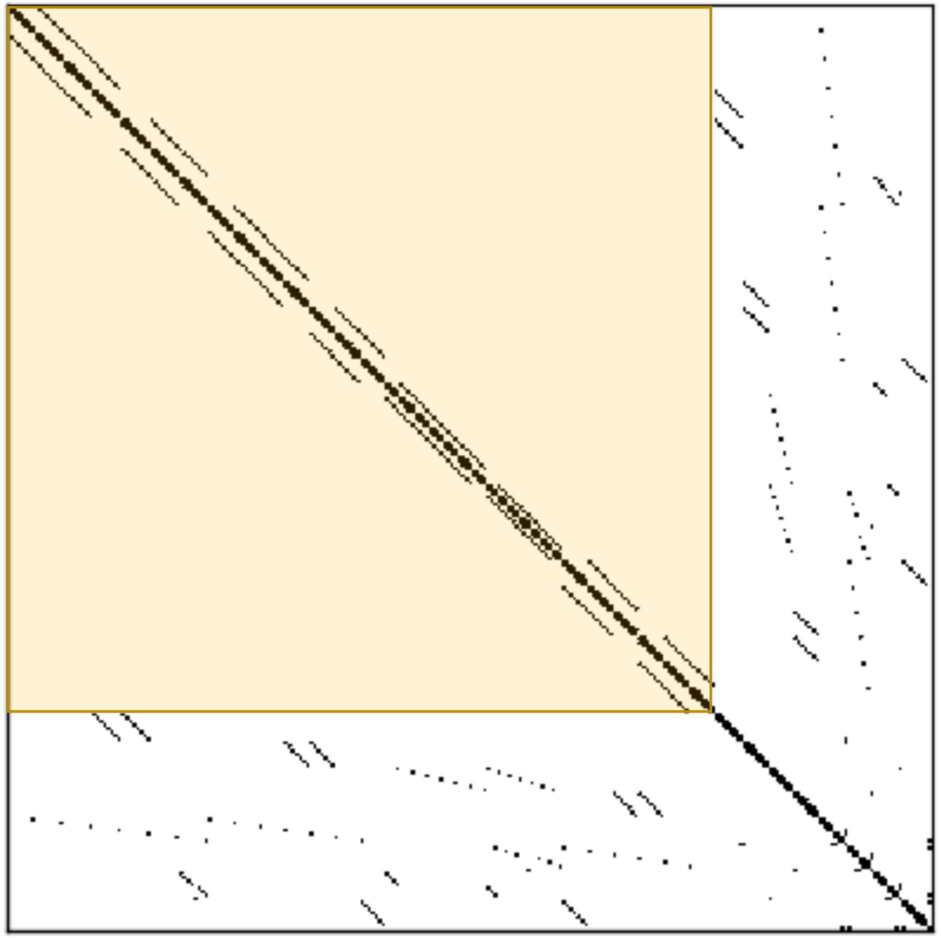
$$\begin{bmatrix} Q_p^T & \\ & I \end{bmatrix} \begin{bmatrix} I & A_{pn} \\ A_{np} & A_{nn} \end{bmatrix} \begin{bmatrix} Q_p & \\ & I \end{bmatrix} = \begin{bmatrix} I & & \times \\ & I & W_{cn} \\ \times & W_{nc} & A_{nn} \end{bmatrix}$$

“No fill-in!”

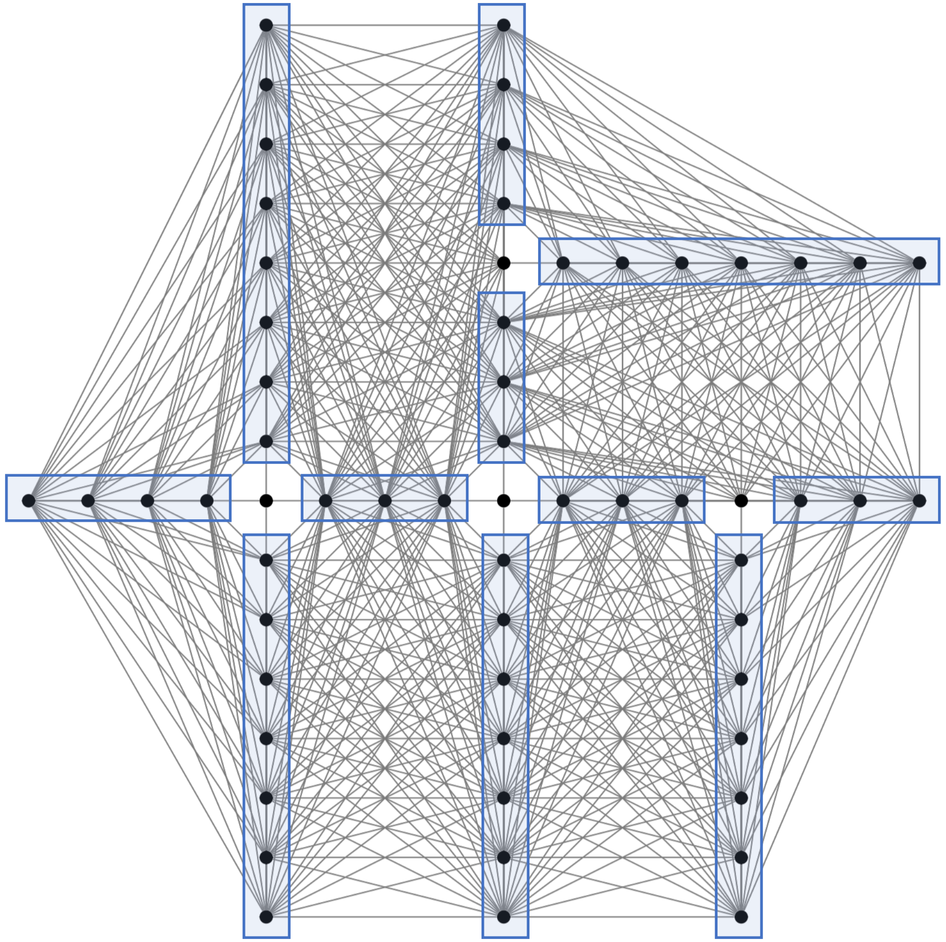


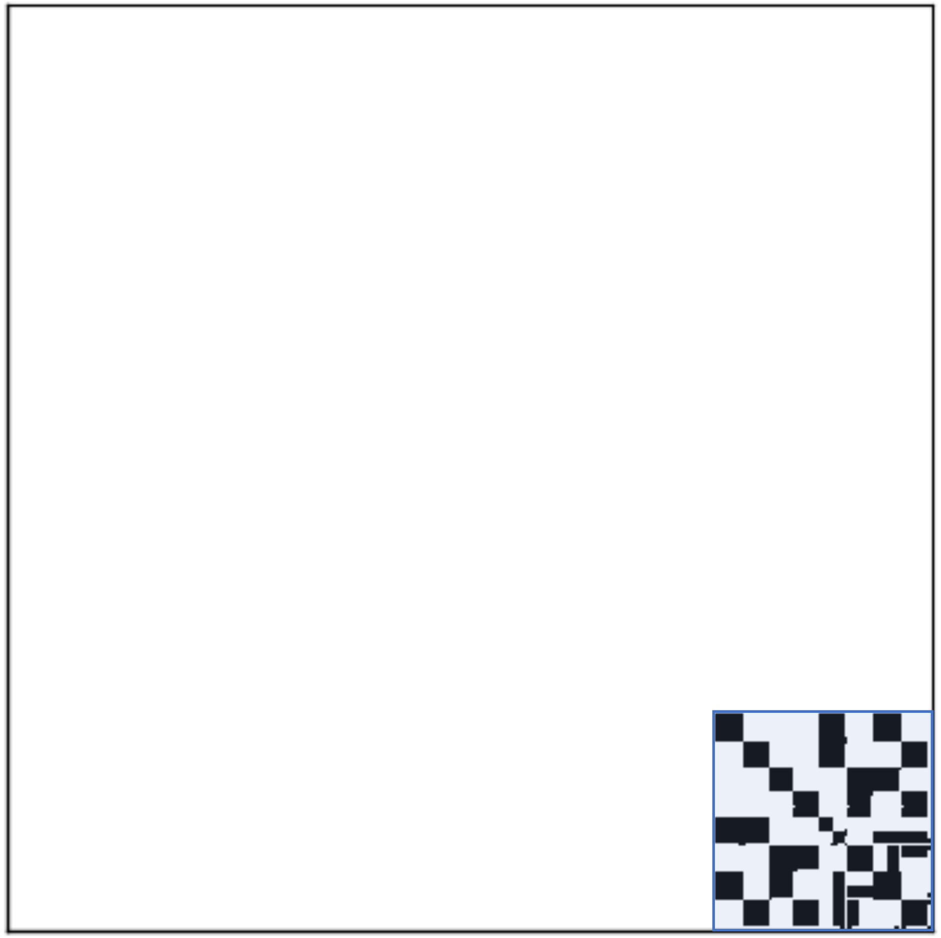
All together: spaND (1)





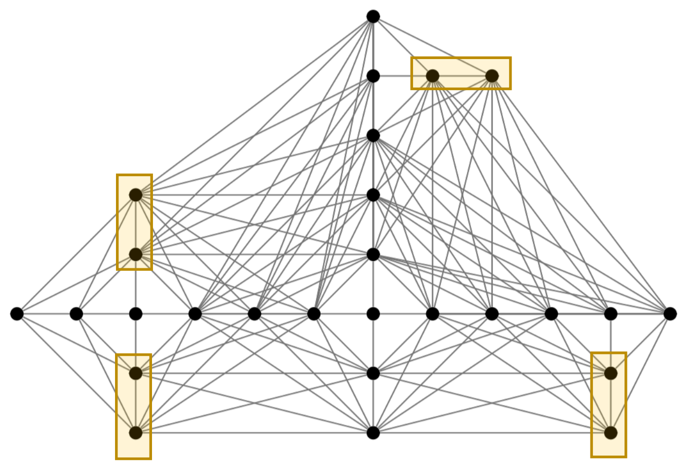
All together: spaND (2)

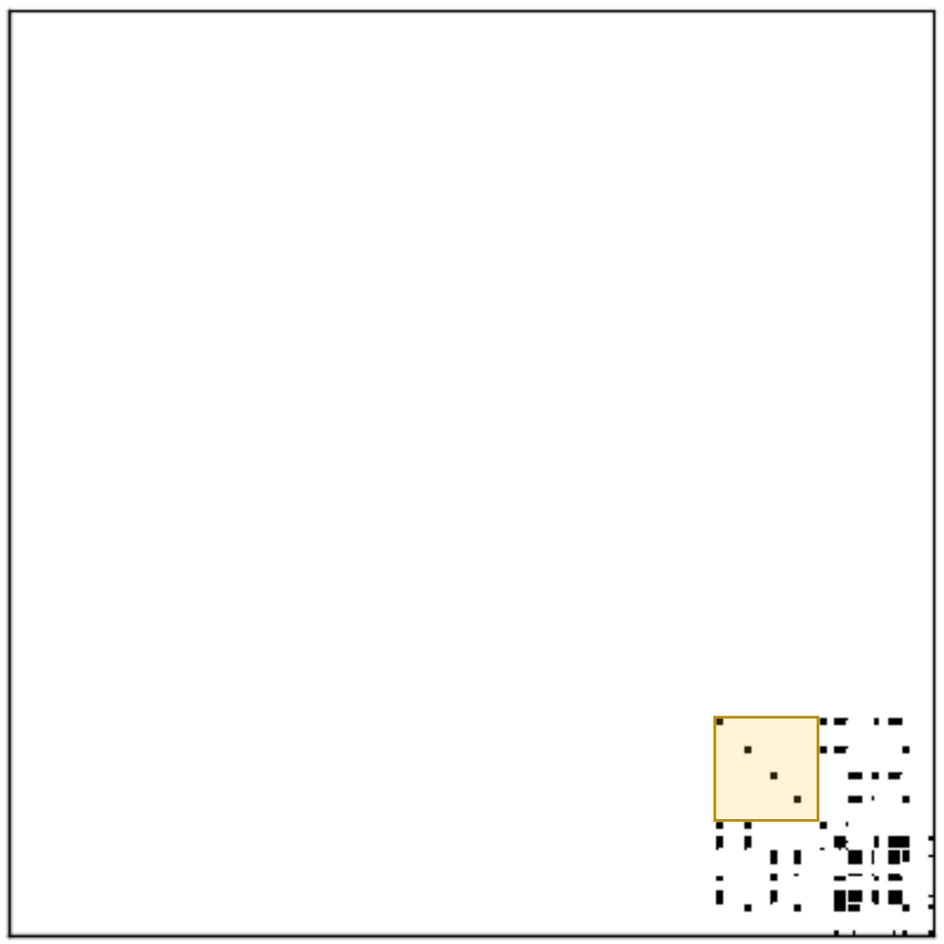




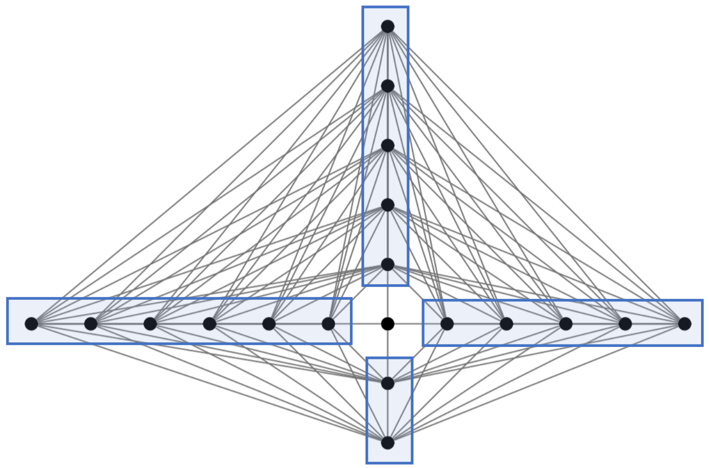
All together: spaND (3)

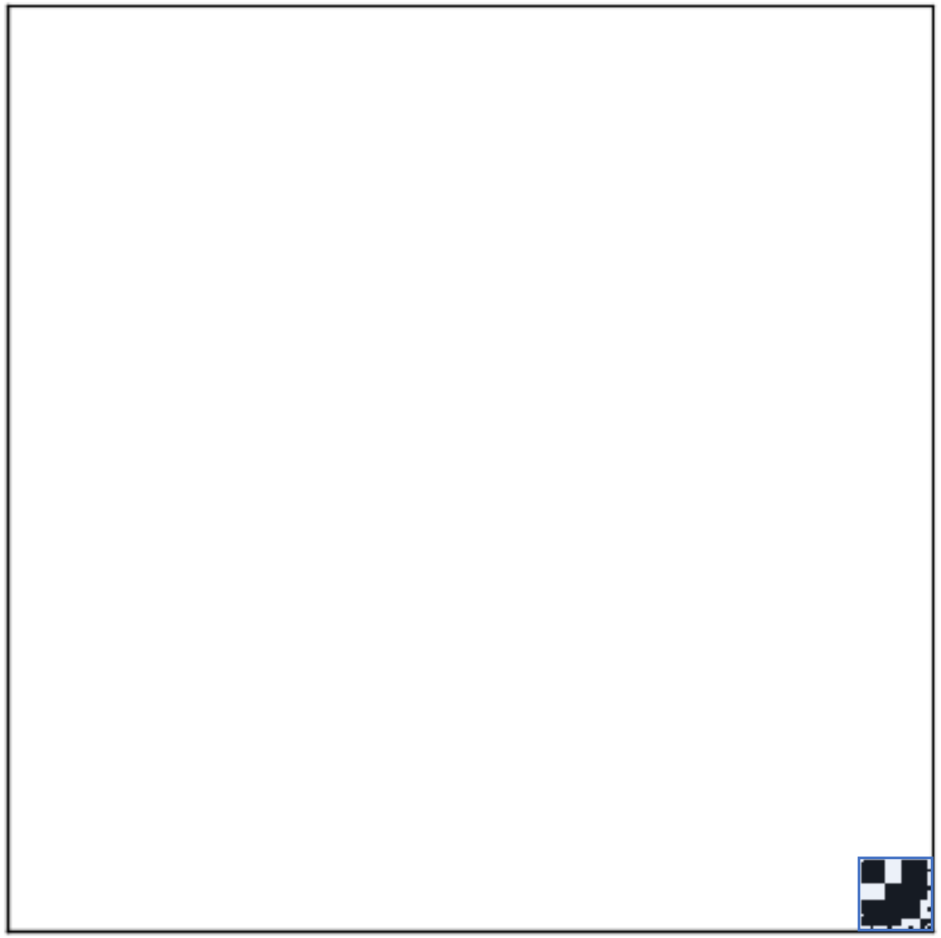
≈





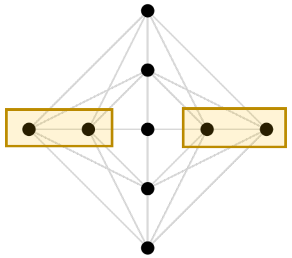
All together: spaND (4)

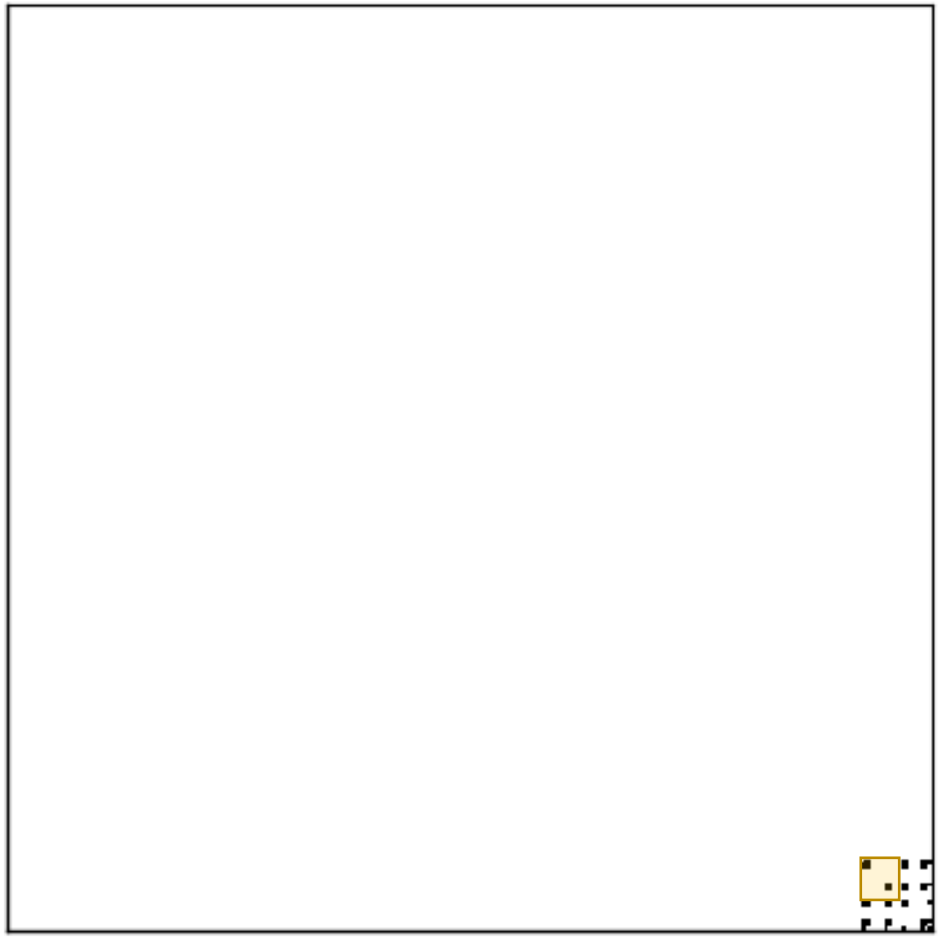




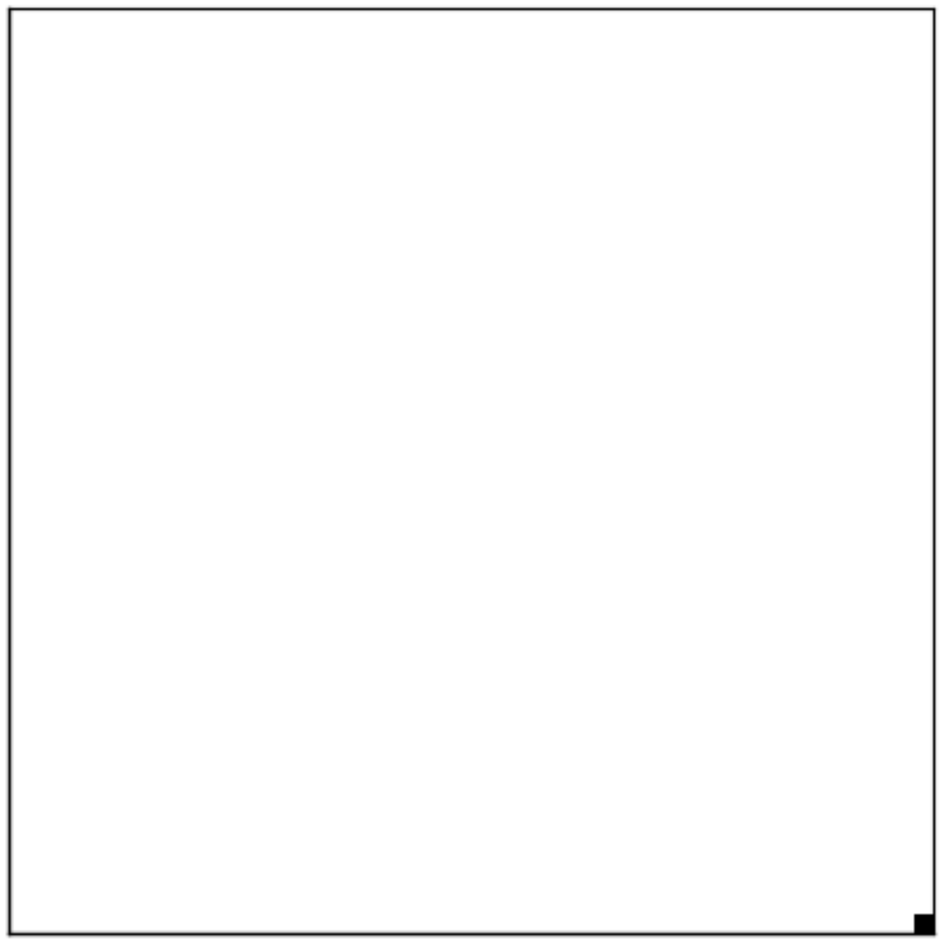
All together: spaND (5)

$\approx$





All together: spaND (6)



General spaND

For level k = 1, …, L

- Eliminate interiors (LL^T, LDL^T, PLU, PLUQ)

$$L^{-1}\begin{bmatrix} A_{pp} & A_{pn} & \\ A_{np} & A_{nn} & A_{nw} \\ & A_{wn} & A_{ww} \end{bmatrix} U^{-1} = \begin{bmatrix} I & & \\ & A_{nn} - A_{ns}A_{ss}^{-1}A_{sn} & A_{nw} \\ & A_{wn} & A_{ww} \end{bmatrix}$$

Fill-in;
limited by separators

- Scale interfaces (LL^T, LDL^T, PLU, PLUQ)

$$\begin{bmatrix} L^{-1} & \\ & I \end{bmatrix}\begin{bmatrix} A_{pp} & A_{pn} \\ A_{np} & A_{nn} \end{bmatrix}\begin{bmatrix} U^{-1} & \\ & I \end{bmatrix} = \begin{bmatrix} I & L^{-1}A_{pn} \\ A_{np}U^{-1} & A_{nn} \end{bmatrix}$$

- Sparsify interfaces (RRQR)

$$\begin{bmatrix} Q_p^T & \\ & I \end{bmatrix}\begin{bmatrix} I & A_{pn} \\ A_{np} & A_{nn} \end{bmatrix}\begin{bmatrix} Q_p & \\ & I \end{bmatrix} = \begin{bmatrix} I & & \varepsilon \\ & I & W_{cn} \\ \varepsilon & W_{nc} & A_{nn} \end{bmatrix} \approx \begin{bmatrix} I & & \\ & I & W_{cn} \\ & W_{nc} & A_{nn} \end{bmatrix}$$

No fill-in!

- Merge clusters

Stability

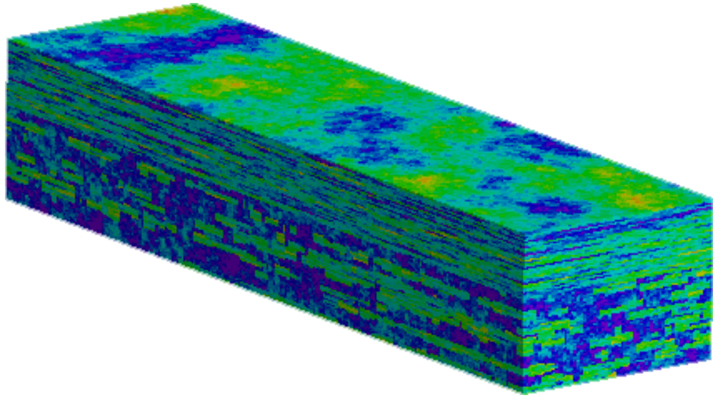
$$\begin{bmatrix} Q_p^T & \\ & I \end{bmatrix} \begin{bmatrix} I & A_{pn} \\ A_{np} & A_{nn} \end{bmatrix} \begin{bmatrix} Q_p & \\ & I \end{bmatrix} = \begin{bmatrix} I & & \varepsilon \\ & \begin{matrix} I & W_{cn} \\ W_{nc} & A_{nn} \end{matrix} & \\ \varepsilon & & \end{bmatrix}$$

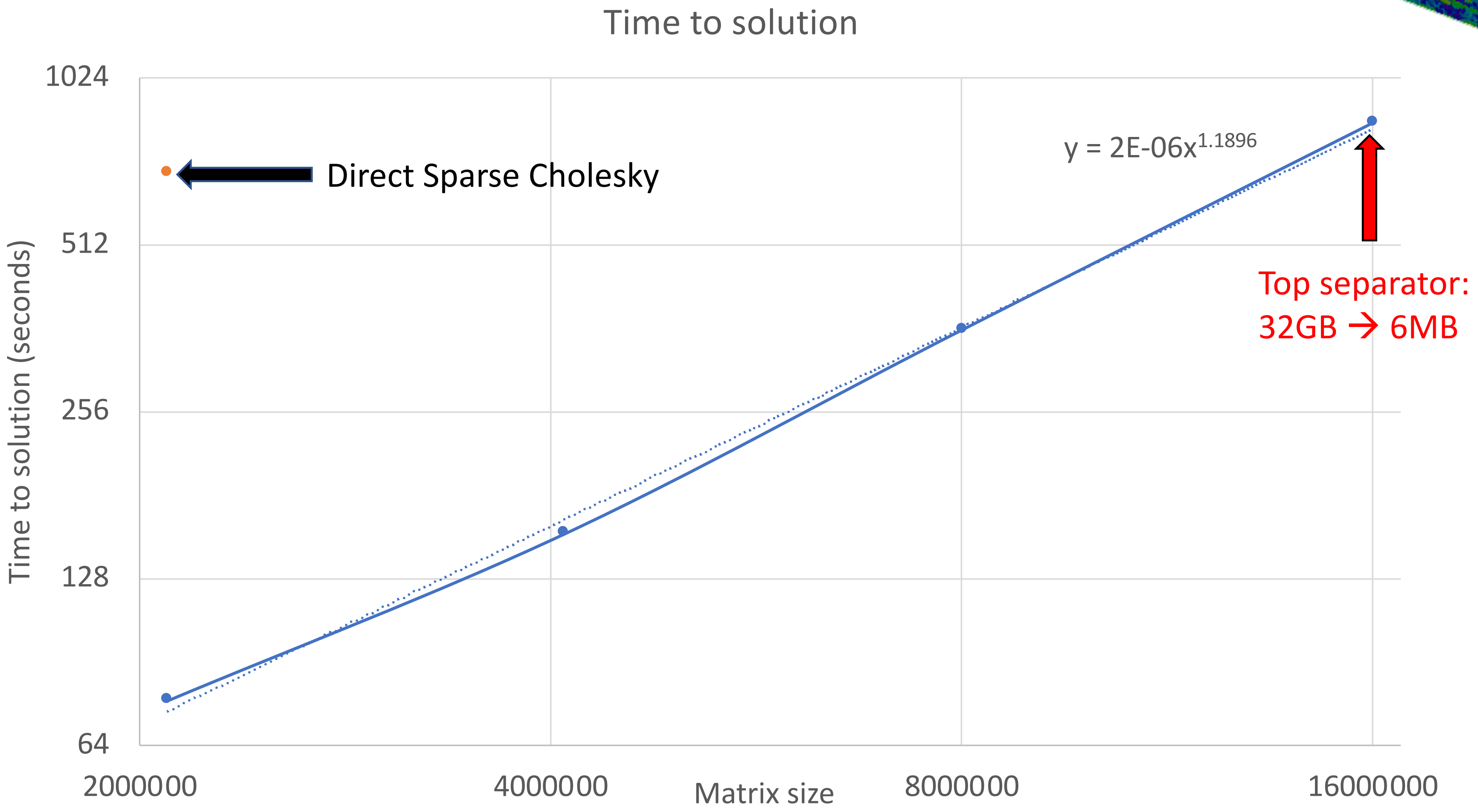
SPD

SPD

SPE10

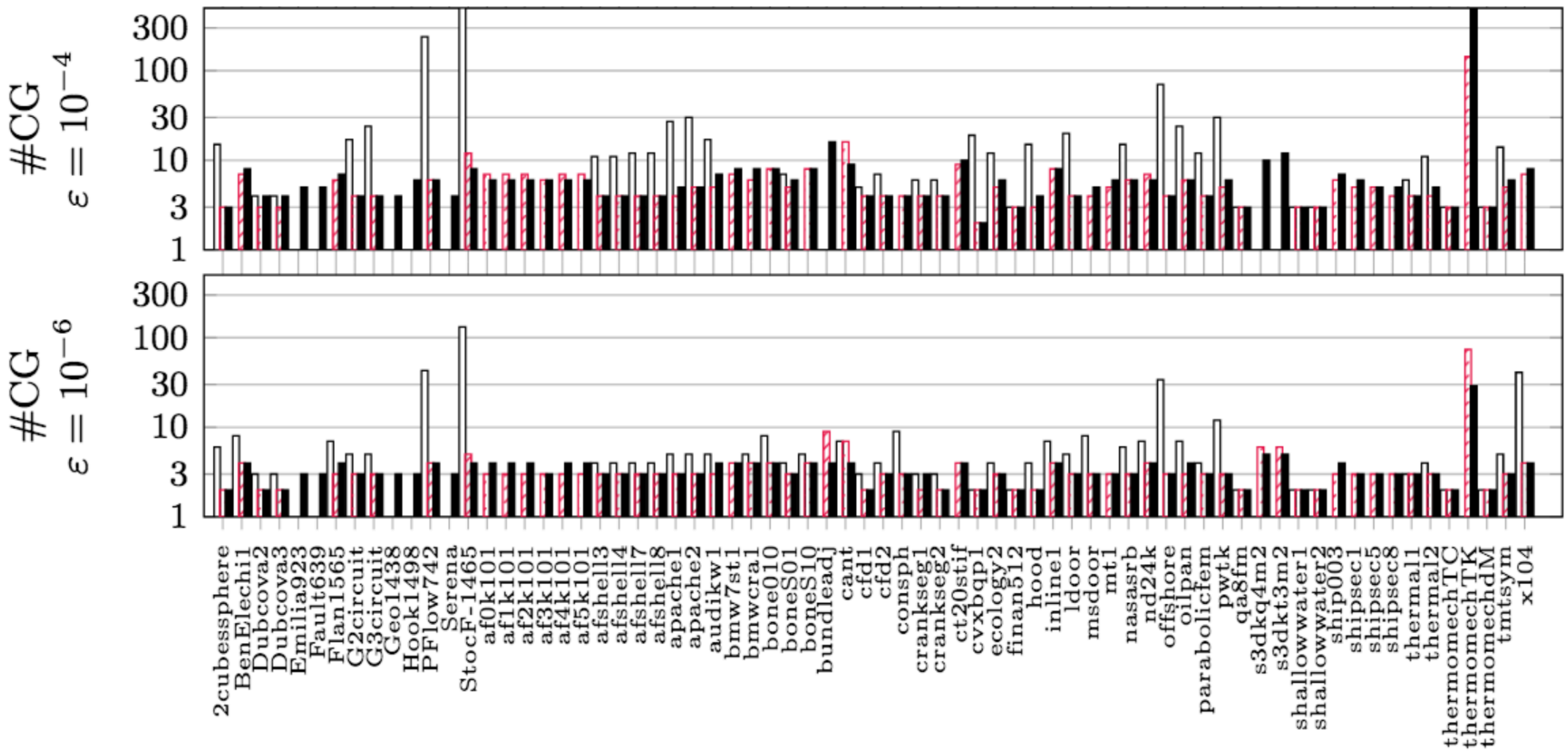
(3D $\nabla \cdot (a(x)\nabla u)$ Poisson-like, SPD)





Stability

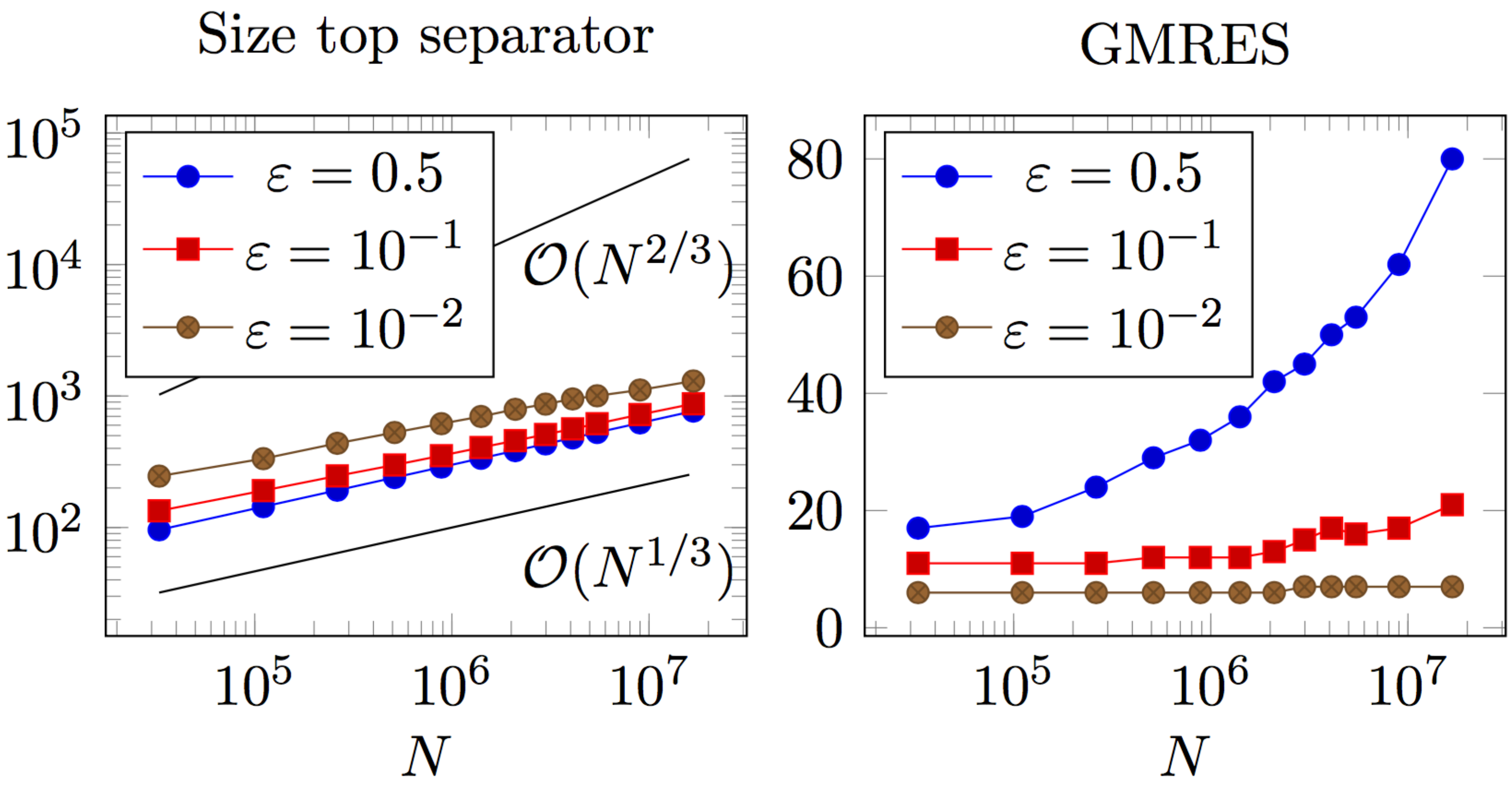
Black = spaND
Red, white = spaND variants



All >50k SPD problems from SuiteSparse, except 2 that didn't converge in <500

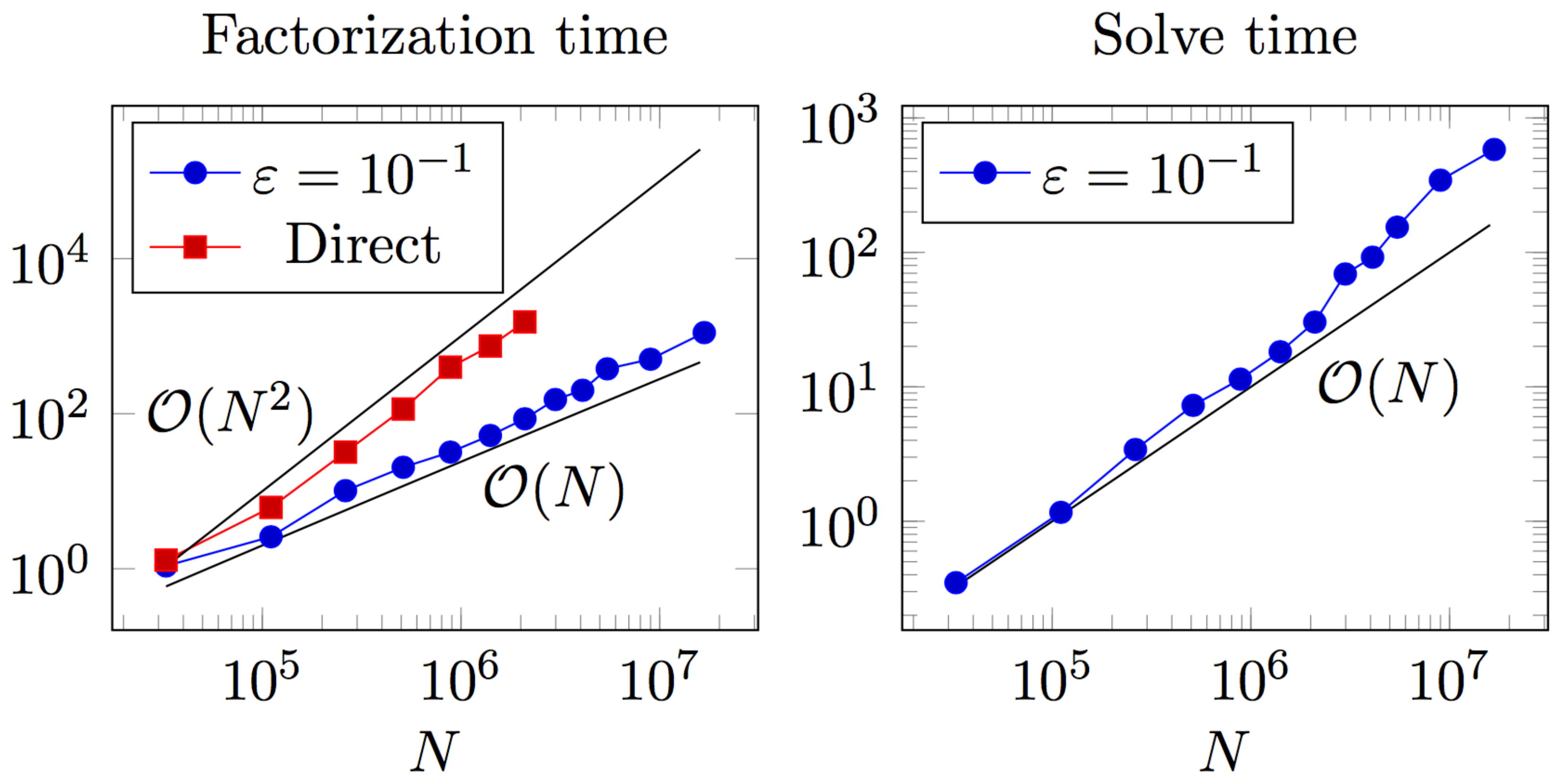
Advection Diffusion

(3D FD $a\Delta u + b\nabla u = f$, Dirichlet, $a = 10^{-2}, b = 1$, Unsymmetric)



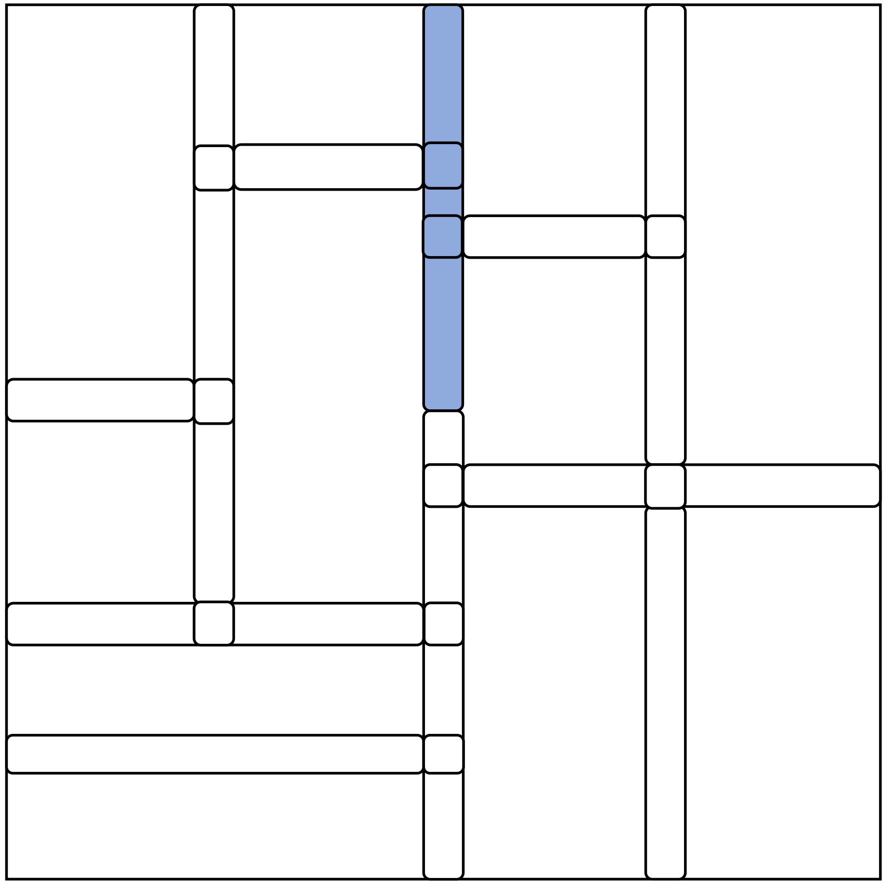
Advection Diffusion

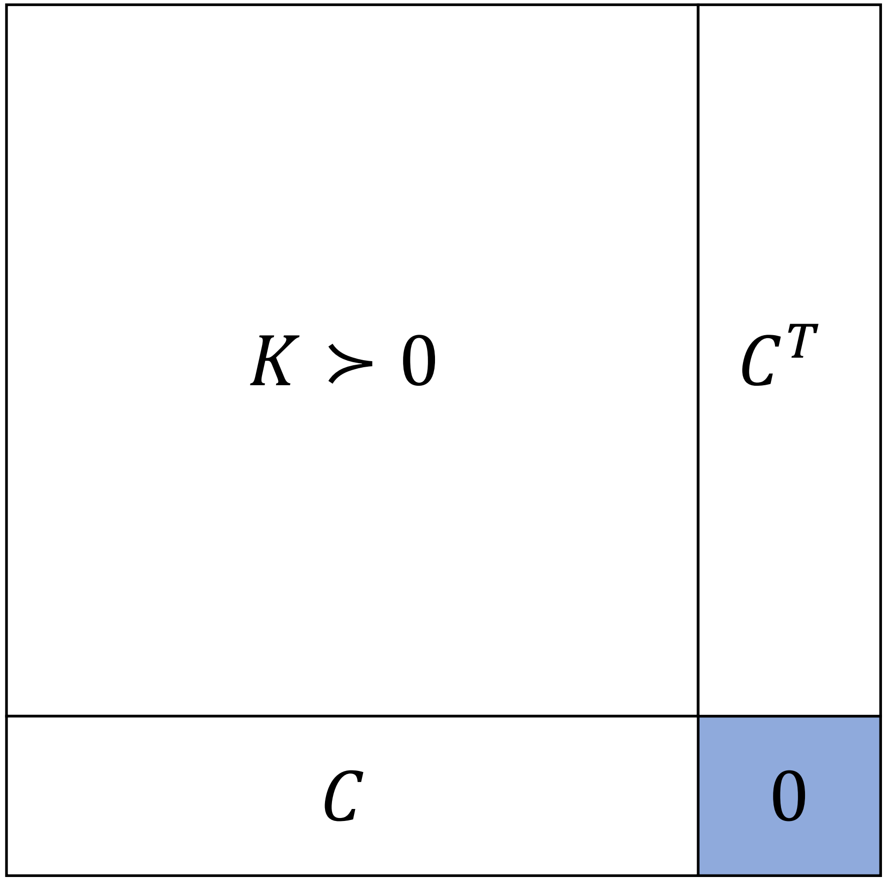
(3D FD $a\Delta u + b\nabla u = f$, Dirichlet, $a = 10^{-2}, b = 1$, Unsymmetric)



Fault Problems

- Fault sparsified when neighbors eliminated
- Fault eliminated last

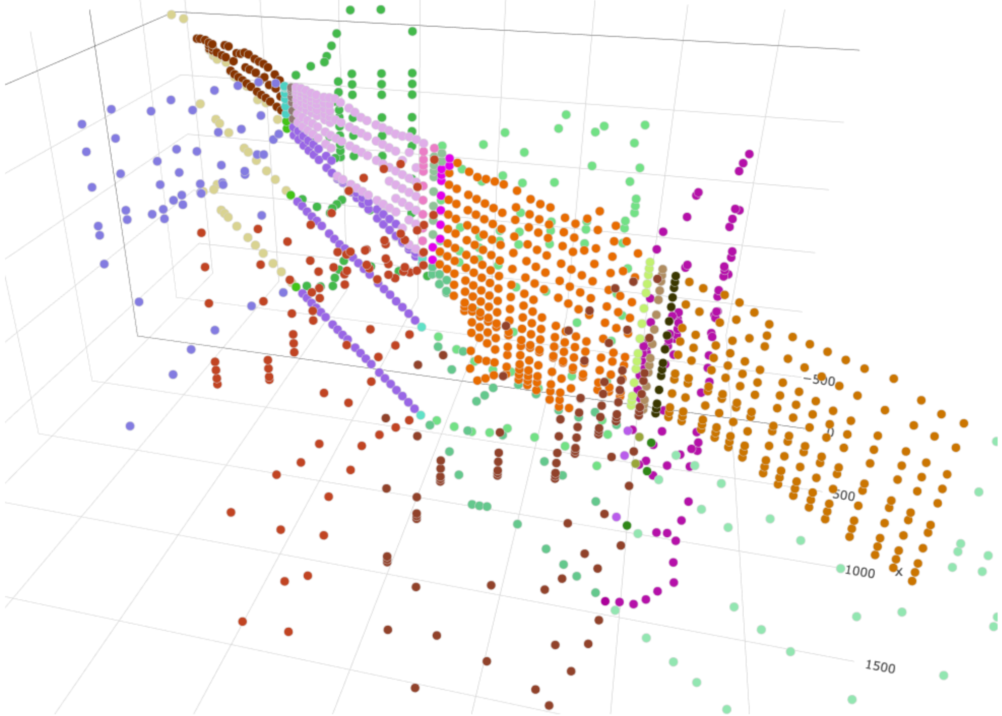




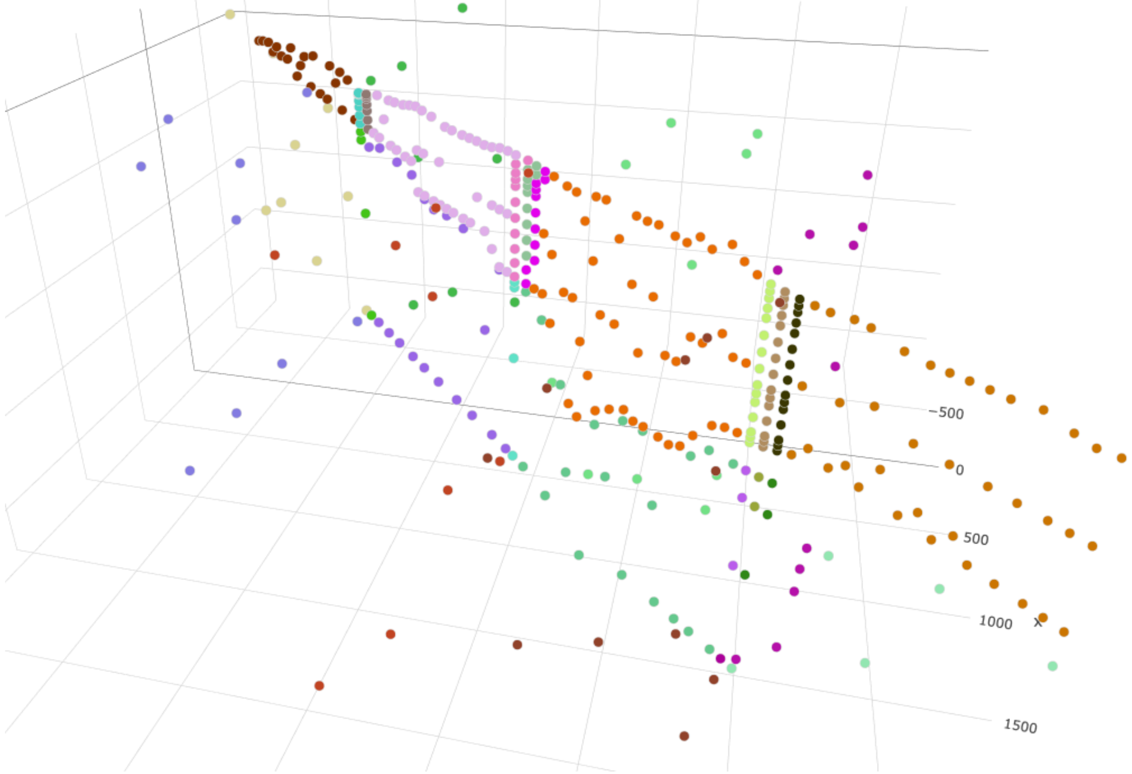
With Bazyli Klockiewicz

Fault Sparsification

Before sparsification



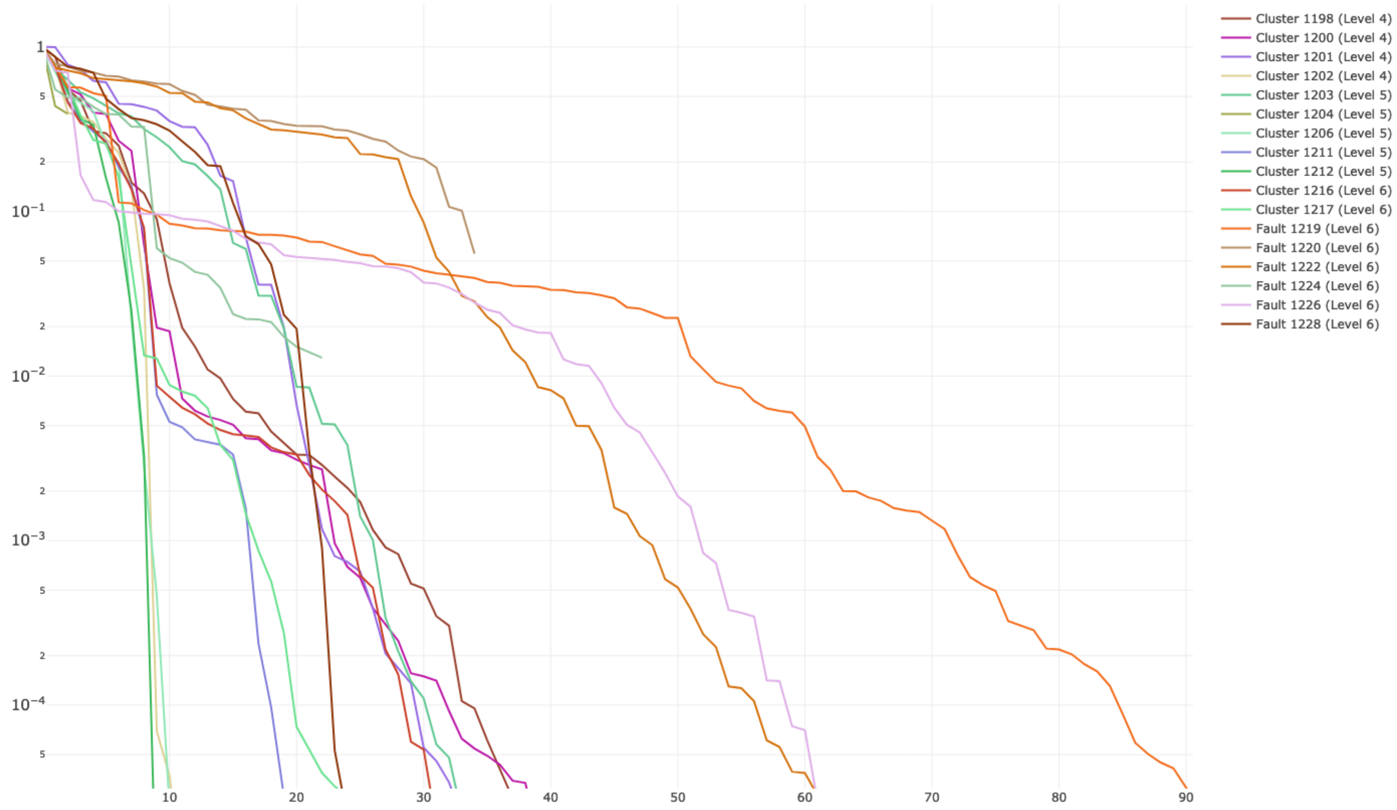
After sparsification



Visualization from Tanay Sonthalia

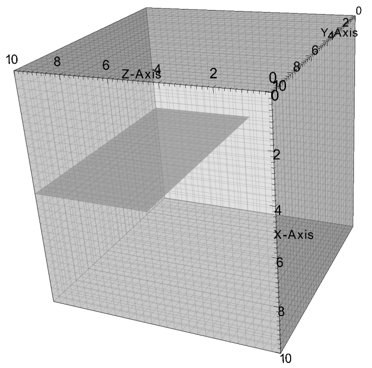
Fault Sparsification

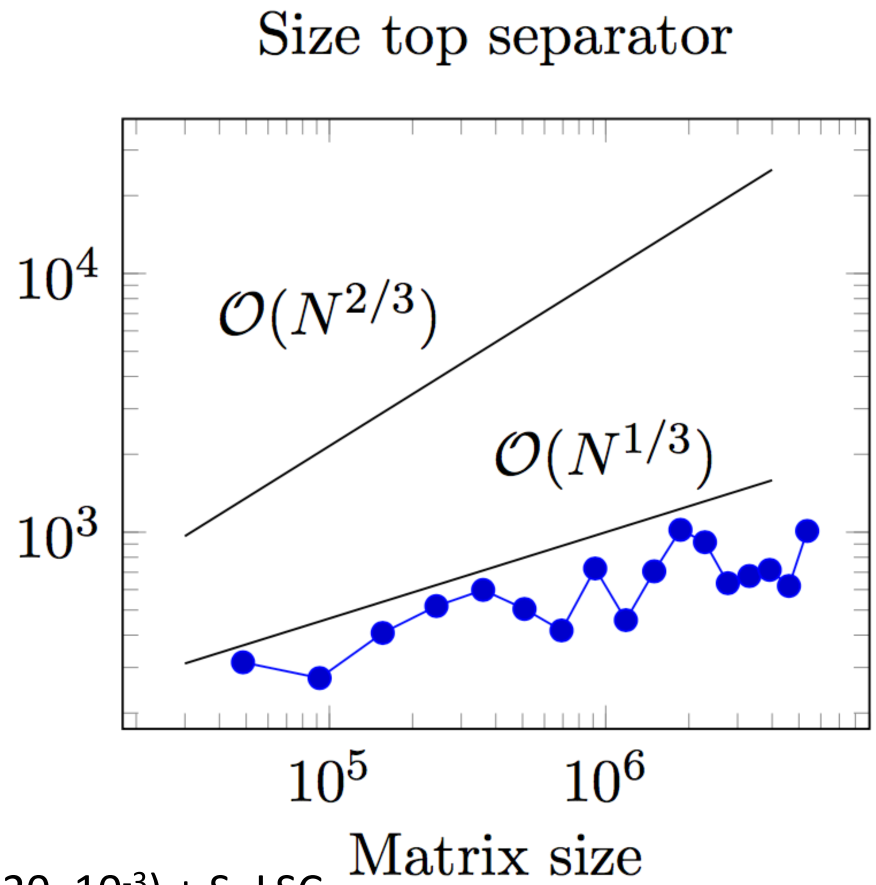
Diagonal of RRQR's R for all blocks

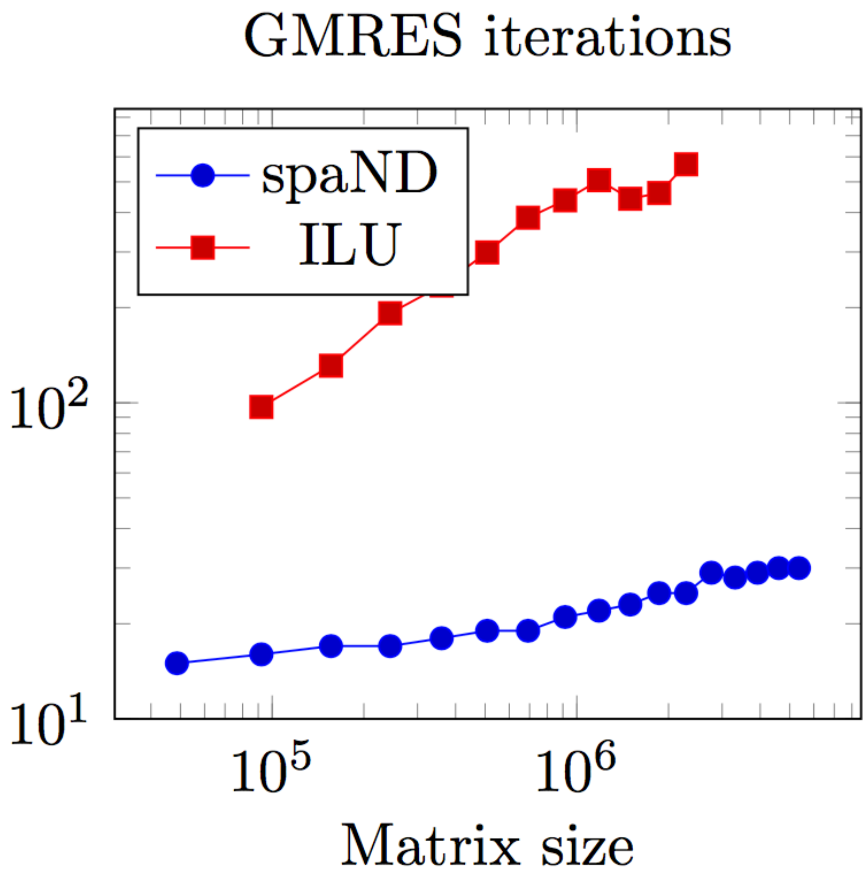


Isotropic Fracture Problems

(3D FE, 1 fault, symmetric, not SPD, many singular blocks)







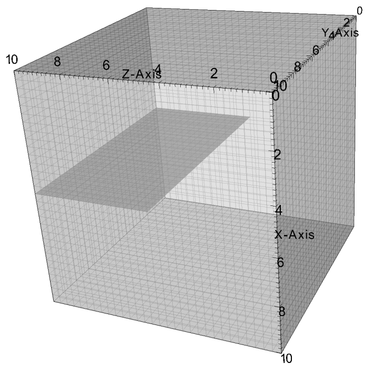
ILU = ILU(20, 10^{-3}) + S_LSC

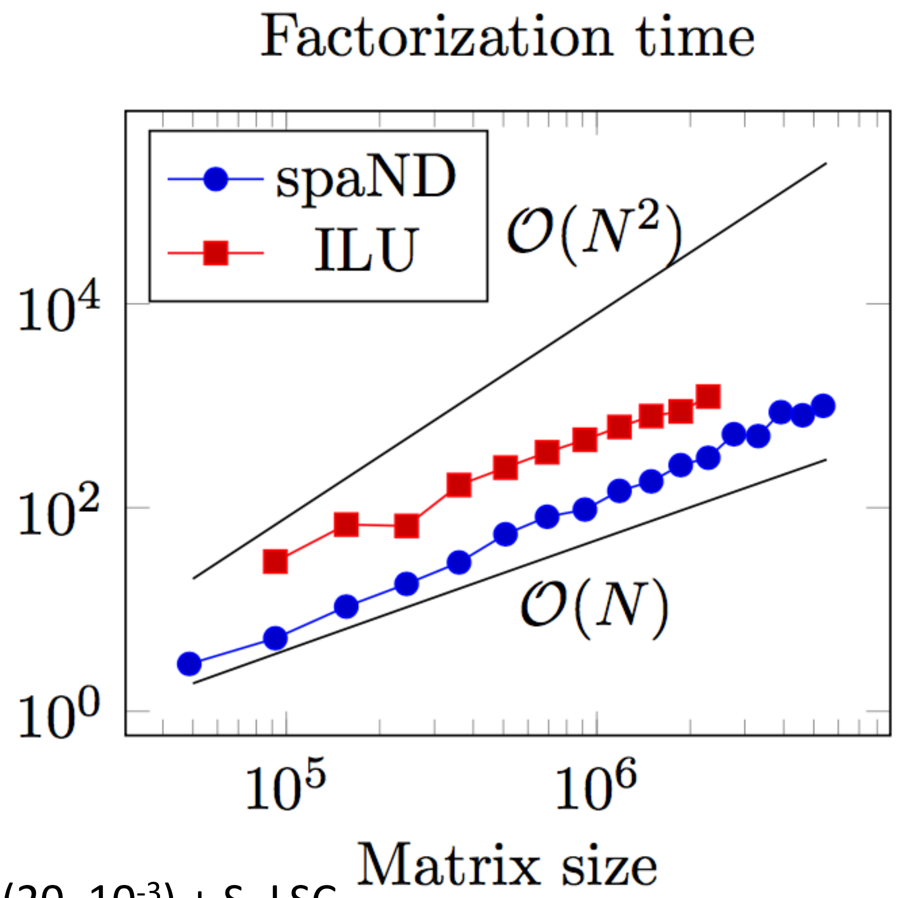
spaND = 0.05 RRQR + near kernel approx.

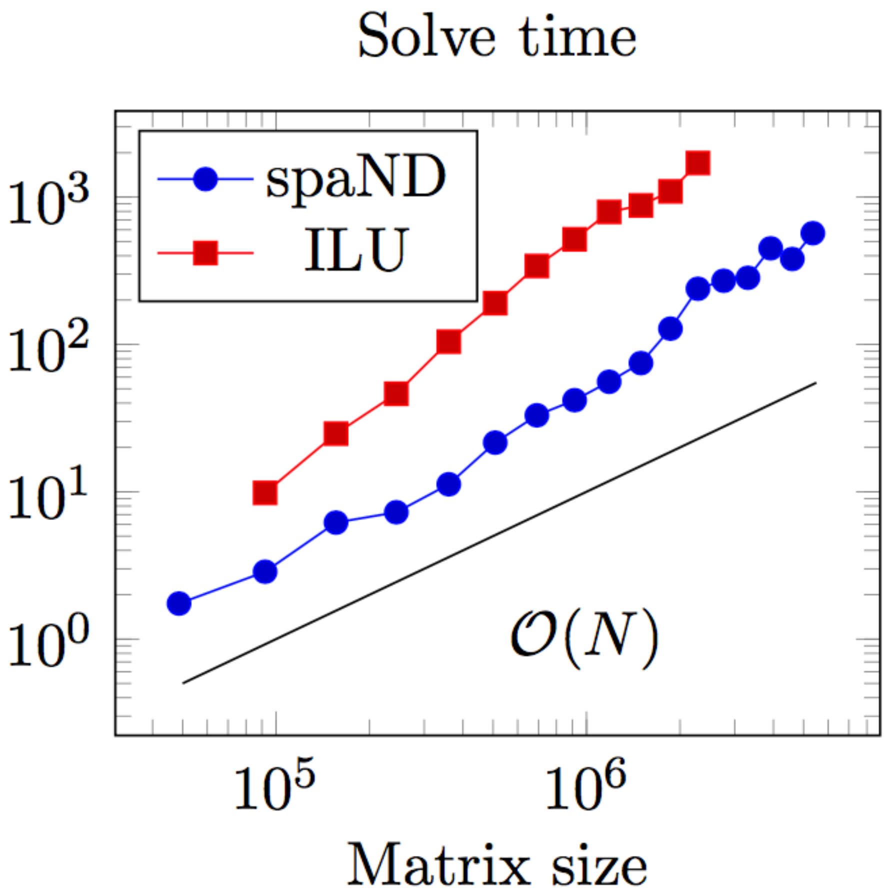
With Bazyli Klockiewicz

Isotropic Fracture Problems

(3D FE, 1 fault, symmetric, not SPD, many singular blocks)







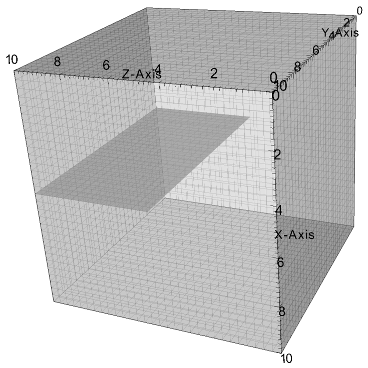
ILU = ILU(20, 10^{-3}) + S_LSC

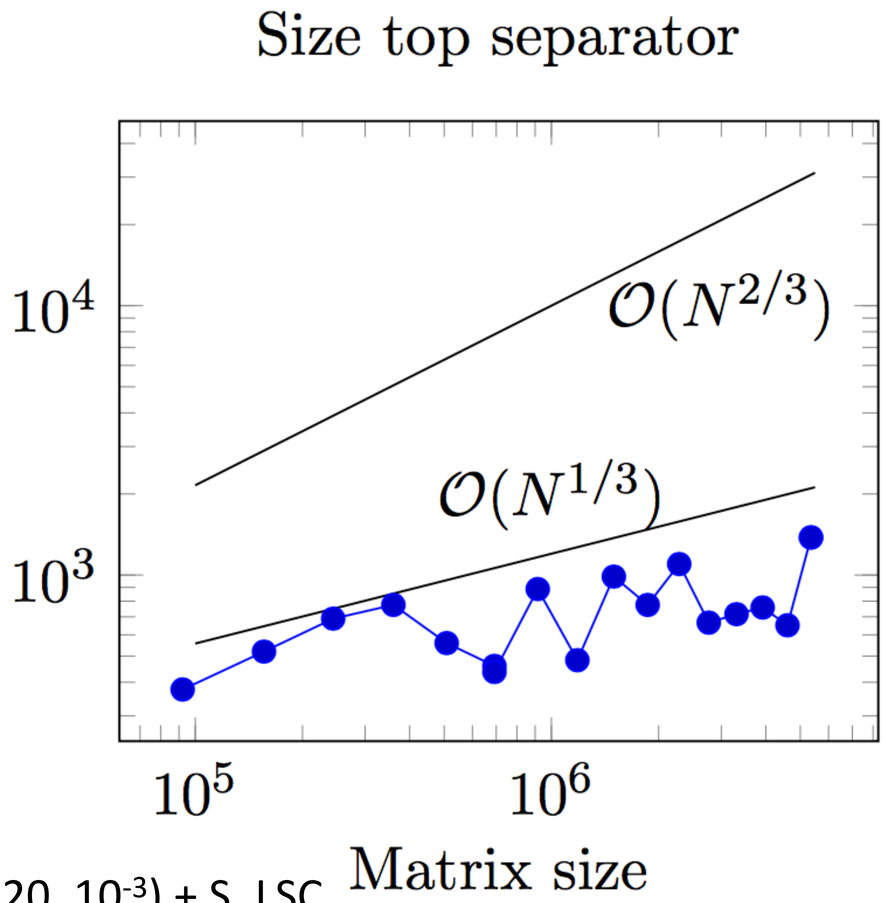
spaND = 0.05 RRQR + near kernel approx

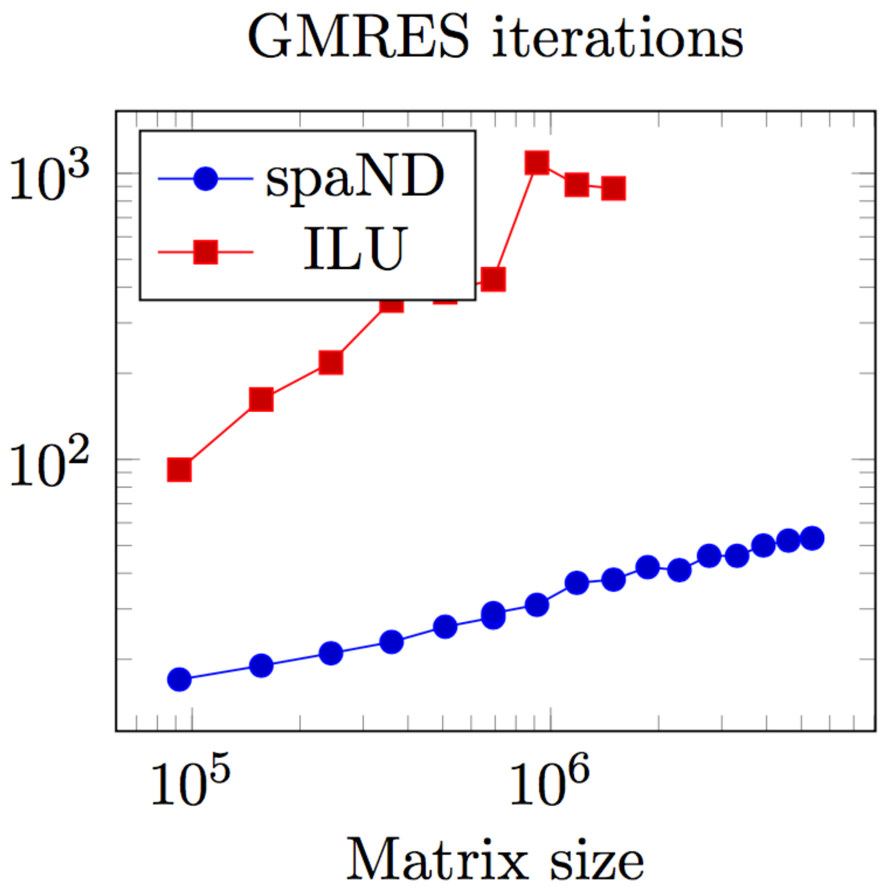
With Bazyli Klockiewicz

Anisotropic Fracture Problems

(3D FE, 1 fault, symmetric, anisotropic 1/5/10, not SPD, many singular blocks







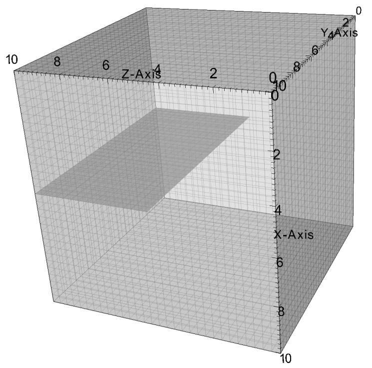
ILU = ILU(20, 10^{-3}) + S_LSC

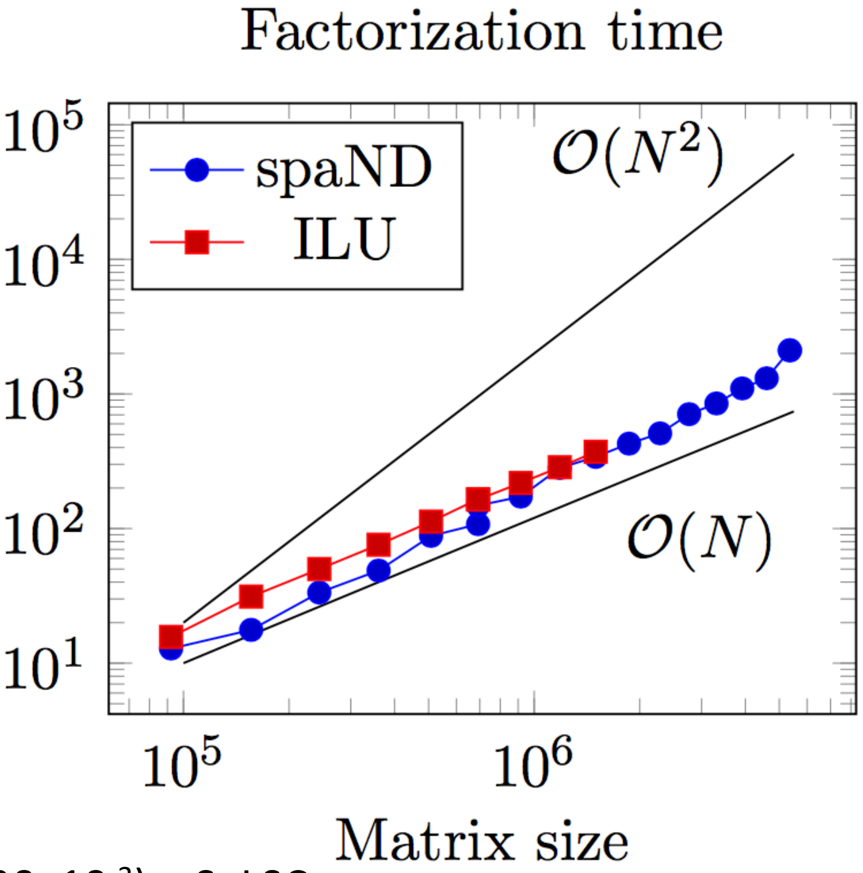
spaND = 0.05 RRQR + near kernel approx

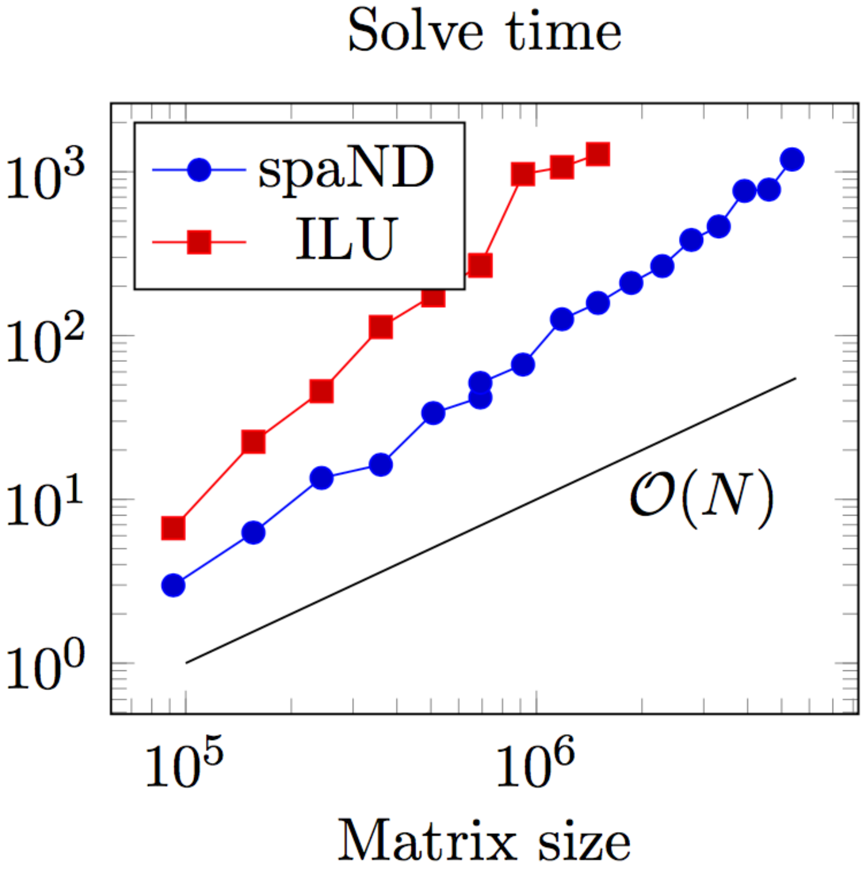
With Bazyli Klockiewicz

Anisotropic Fracture Problems

(3D FE, 1 fault, symmetric, anisotropic 1/5/10, not SPD, many singular blocks







ILU = ILU(20, 10^{-3}) + S_LSC

spaND = 0.05 RRQR + near kernel approx

With Bazyli Klockiewicz

Conclusions and Future Work

- General algebraic algorithm, works on large class of problems
- Future work
  - O(N log N) down to O(N) algorithm
    - Potentially faster
    - More scalable
  - Parallel task-based version with TaskTorrent (https://github.com/leopoldcambier/tasktorrent) runtime

Acknowledgements

- Initial spaND work with Chao Chen, Eric Darve, Erik Boman, Ray Tuminaro, Siva Rajamanickam: https://arxiv.org/abs/1901.02971
- Fault + spaND work with Bazyli Klockiewicz
- Visualization from Tanay Sonthalia
- Support from Sandia National Lab & Total