

# Towards an Optimal-Order Approximate Sparse Factorization Exploiting Data-Sparseness in Separators

**X. Sherry Li**, Lawrence Berkeley National Laboratory

**Artem Napov**, Université Libre de Bruxelles

**Francois-Henry Rouet**, Lawrence Berkeley National Laboratory

**Shen Wang**, ConocoPhillips

**Jianlin Xia**, Purdue University

Workshop Celebrating 40 Years of Nested Dissection, Waterloo, July 22-23, 2013

# Introduction

Consider solving large sparse linear system  $A\mathbf{u} = \mathbf{b}$  with Gaussian elimination:  $A = LU$

- Deliver reliable solution, error bounds, condition estimation, efficient for many RHS, . . .
- **Complexity wall** ... not linear time

[George '73] For model problems, (exact) sparse LU with best ordering **Nested Dissection** gives optimal complexity:

- ▶ 2D ( $k \times k = n$  grids):  $\mathcal{O}(n \log n)$  Fill,  $\mathcal{O}(n^{3/2})$  Flops

**Fill:** adding up the dense submatrices of all the “+” separators:

$$k^2 + 4 \left(\frac{k}{2}\right)^2 + 4^2 \left(\frac{k}{4}\right)^2 + \dots = \sum_{i=0} \left(\frac{k}{2^i}\right)^2 = O(k^2 \log k)$$

**Flops:** dominated by cubic term of factorizing top-level separator:  $O(k^3)$

# Approximation, sparsification

## Exploit “data-sparseness” structure in separators

- **data-sparse**: matrix may be dense, but has a compressed representation smaller than  $N^2$

## Low-rank matrices as basic building blocks

- If  $B$  has exact rank at most  $k$ :
  - ▶ Outer-product form:  $B_{m \times n} = U_{m \times k} V_{k \times n}^T, k \leq n$
  - ▶ Orthonormal outer-product form:  
$$B_{m \times n} = U_{m \times k} X_{k \times k} V_{k \times n}^T, \quad U^T U = V^T V = I_k$$
- If  $A$  has numerical low rank  $k$  (called  **$\varepsilon$ -rank**):  
$$A = U \Sigma V^T \approx A_k := U \Sigma_k V^T, \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_k, \sigma_{k+1}, \dots, \sigma_n)$$
  
$$\Sigma_k = \text{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0), \text{ with } \sigma_k > \varepsilon$$

## Algorithms:

- truncated SVD
- rank-revealing QR
- randomized sampling, ...

# Various low-rank approaches

Have been widely used in structured matrices:

- Hierarchical matrices:  $\mathcal{H}$ ,  $\mathcal{H}^2$ , (Bebendorf, Börm, Grasedyck, Hackbusch, ...) (Eric Darve's talk, Monday, 11:15am-12:00pm)
- Hierarchically/Sequential Semiseparable (**HSS / SSS**) representations (Chandrasekaran, Dewilde, Gu, Olshevsky, Vandebril, Xia, ...) (this talk)
- Low-rank sparse Cholesky (David Bindle' talk, Monday, 1:30-2:00pm)
- BLR (Block LR) (Jean-Yves L'Excellent's talk, Tuesday, 2:00-2:45pm)
- MLR (Multilevel LR) (Saad et al.)

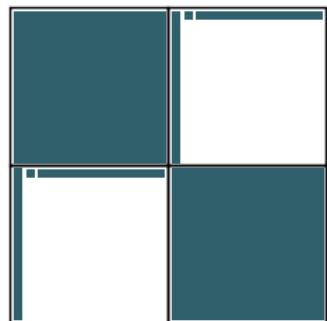
# Outline

- How HSS + multifrontal works operationally?
- Theory
- Practice

# Hierarchically Semi-Separable matrices

An HSS matrix  $A$  is a dense matrix whose off-diagonal blocks are low-rank.  
High-level structure:  $2 \times 2$  blocks

$$A = \left[ \begin{array}{c|c} D_1 & U_1 B_1 V_1^T \\ \hline U_2 B_2 V_1^T & D_2 \end{array} \right]$$



Fundamental property required for efficiency: nested bases

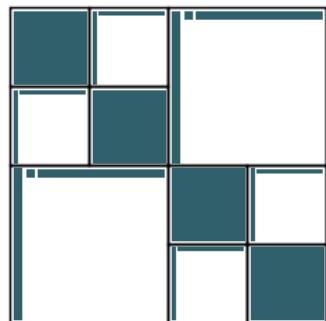
$$U_3 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} U_3^{small}, U_3^{small} : 2k \times k$$

Same for  $U_3, U_6, V_6$  and recursively at subsequent levels.

# Hierarchically Semi-Separable matrices

An HSS matrix  $A$  is a dense matrix whose off-diagonal blocks are low-rank.  
Recursion

$$A = \left[ \begin{array}{c|c} D_1 & U_1 B_1 V_2^T \\ \hline U_2 B_2 V_1^T & D_2 \\ \hline \end{array} \quad \begin{array}{c} U_3 B_3 V_6^T \\ \hline \end{array} \right] \\ \left[ \begin{array}{c|c} U_6 B_6 V_3^T & \begin{array}{c|c} D_4 & U_4 B_4 V_5^T \\ \hline U_5 B_5 V_4^T & D_5 \end{array} \end{array} \right]$$



Fundamental property required for efficiency: nested bases

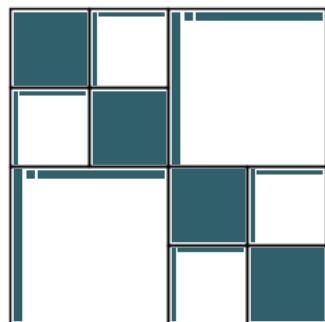
$$U_3 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} U_3^{small}, U_3^{small} : 2k \times k$$

Same for  $U_3, U_6, V_6$  and recursively at subsequent levels.

# Hierarchically Semi-Separable matrices

An HSS matrix  $A$  is a dense matrix whose off-diagonal blocks are low-rank.  
Recursion

$$A = \left[ \begin{array}{c|c} D_1 & U_1 B_1 V_2^T \\ \hline U_2 B_2 V_1^T & D_2 \\ \hline \end{array} \quad \begin{array}{c} U_3 B_3 V_6^T \\ \hline \end{array} \right] \\ \left[ \begin{array}{c|c} U_6 B_6 V_3^T & \begin{array}{c|c} D_4 & U_4 B_4 V_5^T \\ \hline U_5 B_5 V_4^T & D_5 \end{array} \end{array} \right]$$

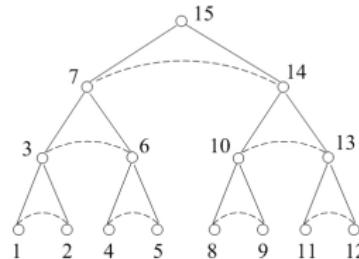
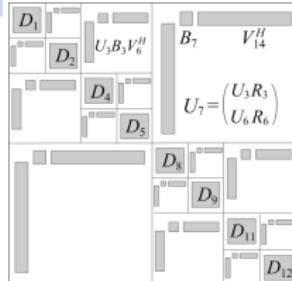


Fundamental property required for efficiency: nested bases

$$U_3 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} U_3^{small}, U_3^{small} : 2k \times k$$

Same for  $U_3, U_6, V_6$  and recursively at subsequent levels.

# Hierarchical bases, HSS tree



For efficiency, require:

$$U_3 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} U_3^{small}, \quad U_3^{small} : 2k \times k, \quad U_6 = \begin{bmatrix} U_4 & 0 \\ 0 & U_5 \end{bmatrix} U_6^{small}, \quad U_6^{small} : 2k \times k$$

$$U_7 = \begin{bmatrix} U_3 & 0 \\ 0 & U_6 \end{bmatrix} U_7^{small}, \quad U_7^{small} : 2k \times k$$

Each basis is a product of descendants' bases:

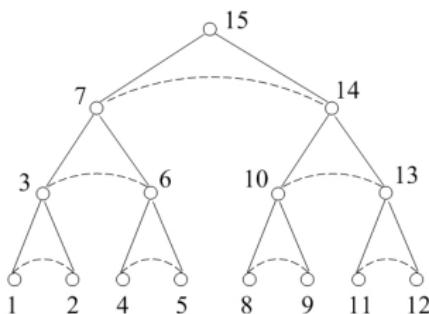
$$U_7 = \begin{bmatrix} U_1 & 0 & 0 & 0 \\ 0 & U_2 & 0 & 0 \\ 0 & 0 & U_4 & 0 \\ 0 & 0 & 0 & U_5 \end{bmatrix} \begin{bmatrix} U_3^{small} & 0 \\ 0 & U_6^{small} \end{bmatrix} U_7^{small},$$

Not to multiply out!

# HSS explicit representation (construction)

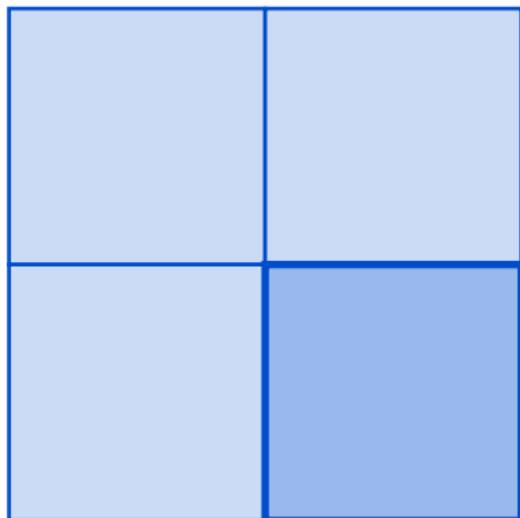
[Martinsson]

$$U^{(3)} \left( U^{(2)} \left( U^{(1)} B^{(0)} V^{(1)*} + B^{(1)} \right) V^{(2)*} + B^{(2)} \right) V^{(3)*} + D^{(3)}$$



- keep it as an unevaluated product & sum
- operations going up / down the HSS tree

# Structured factorization



HSS node :

$$() \approx V(\tilde{T}_r, \tilde{T}_c^H)$$

2. Compl. basis with  $V_L$ :  
 $(V_L, V)$  is unitary

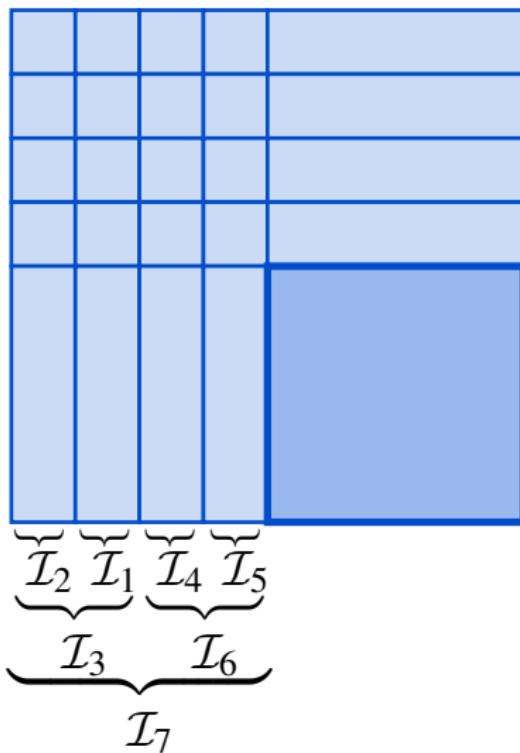
3. Change basis:  
 $\tilde{D} = (V_L, V)^H (V_L, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_R & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization

HSS node :



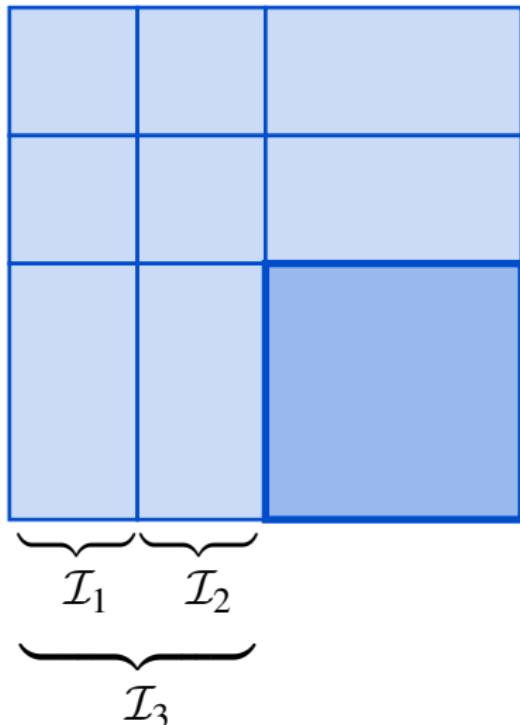
$$(\cdot) \approx V(\tilde{T}_r, \tilde{T}_c^H)$$

2. Compl. basis with  $V_L$ :  
 $(V_L, V)$  is unitary

3. Change basis:  
 $\tilde{D} = (V_L, V)^H (V_L, V)$

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_B \end{pmatrix} \begin{pmatrix} D_B & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node :

$$() \approx V(\tilde{T}_r, \tilde{T}_c^H)$$

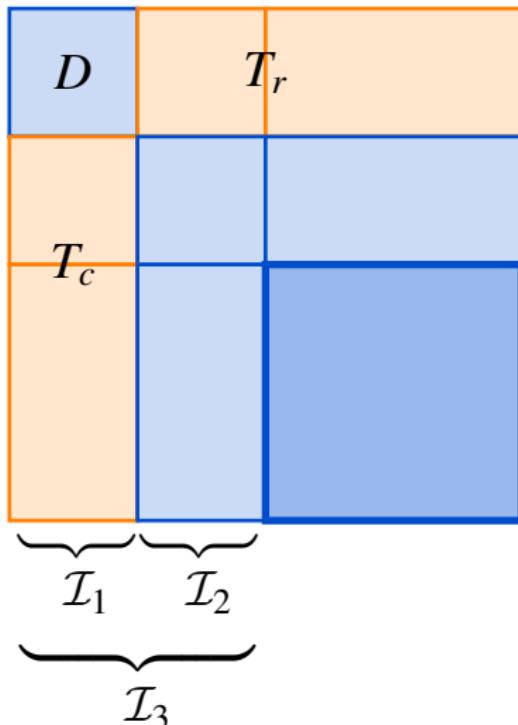
2. Compl. basis with  $V_L$ :  
 $(V_L, V)$  is unitary

3. Change basis:  
 $\tilde{D} = (V_L, V)^H (V_L, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_B \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node 1:

1. Approximate:  
 $(\mathbf{T}_r, \mathbf{T}_c^H) \approx V(\tilde{\mathbf{T}}_r, \tilde{\mathbf{T}}_c^H)$

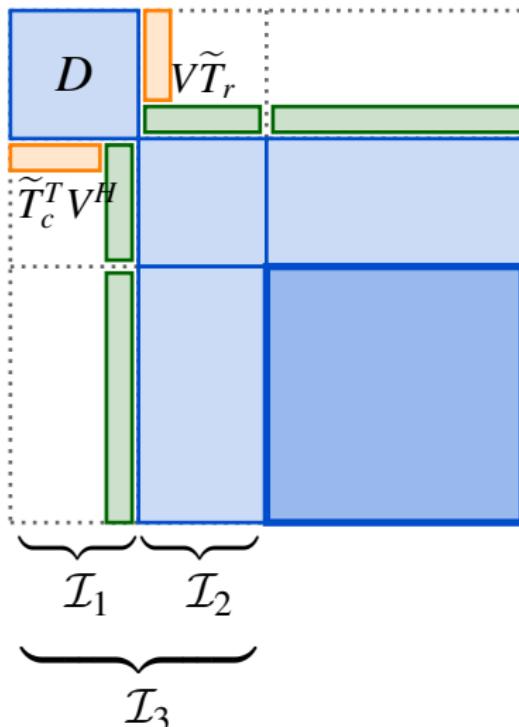
2. Compl. basis with  $V_\perp$ :  
 $(V_\perp, V)$  is unitary

3. Change basis:  
 $\tilde{D} = (V_\perp, V)^H (V_\perp, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_T \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node 1:

1. Approximate:  
 $(T_r, T_c^H) \approx V(\tilde{T}_r, \tilde{T}_c^H)$

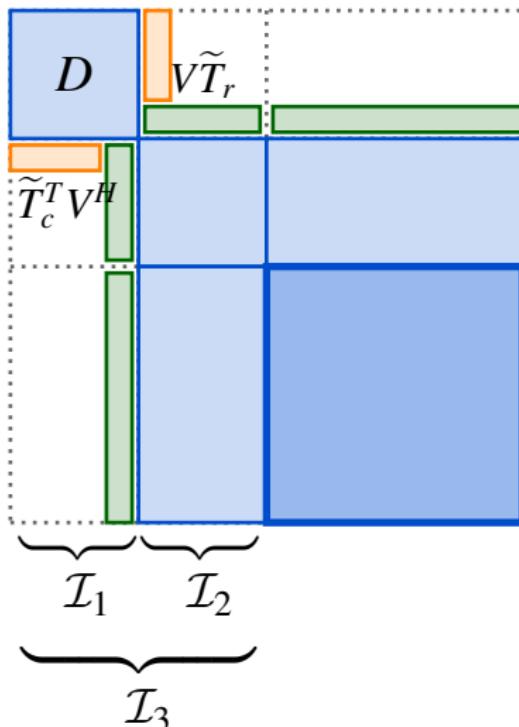
2. Compl. basis with  $V_\perp$ :  
 $(V_\perp, V)$  is unitary

3. Change basis:  
 $\tilde{D} = (V_\perp, V)^H (V_\perp, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_T \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node 1:

1. Approximate:  
 $(T_r, T_c^H) \approx V(\tilde{T}_r, \tilde{T}_c^H)$

2. Compl. basis with  $V_\perp$ :  
 $(V_\perp, V)$  is unitary

3. Change basis:  
 $\tilde{D} = (V_\perp, V)^H (V_\perp, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_T \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization

$$\begin{pmatrix} V_{\perp} & V \end{pmatrix} \times \begin{pmatrix} V^H \\ V^H \end{pmatrix} \times \begin{array}{c} \boxed{D} \\ \boxed{V\tilde{T}_r} \\ \boxed{\tilde{T}_c^T V^H} \end{array} = \boxed{\text{Matrix}} \underbrace{\quad}_{\mathcal{I}_1} \underbrace{\quad}_{\mathcal{I}_2} \underbrace{\quad}_{\mathcal{I}_3}$$

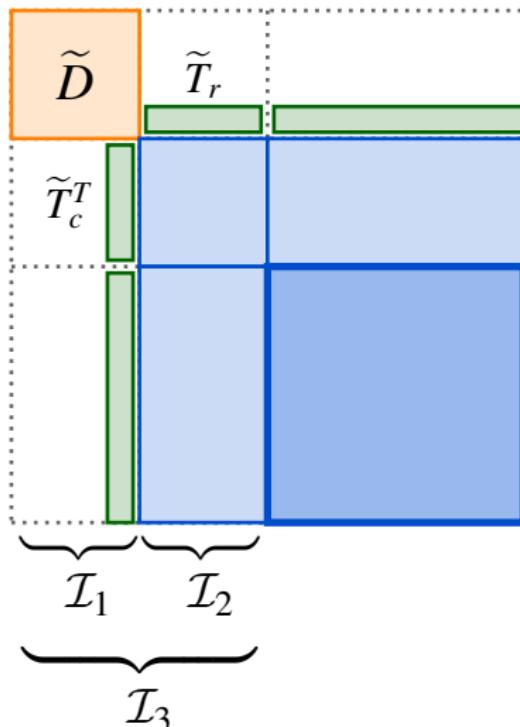
HSS node 1:

1. Approximate:  
 $( T_r, T_c^H ) \approx V( \tilde{T}_r, \tilde{T}_c^H )$
2. Compl. basis with  $V_{\perp}$ :  
 $( V_{\perp}, V )$  is unitary
3. Change basis:  
 $\tilde{D} = ( V_{\perp}, V )^H D ( V_{\perp}, V )$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_R & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node 1:

1. Approximate:  
 $(T_r, T_c^H) \approx V(\tilde{T}_r, \tilde{T}_c^H)$

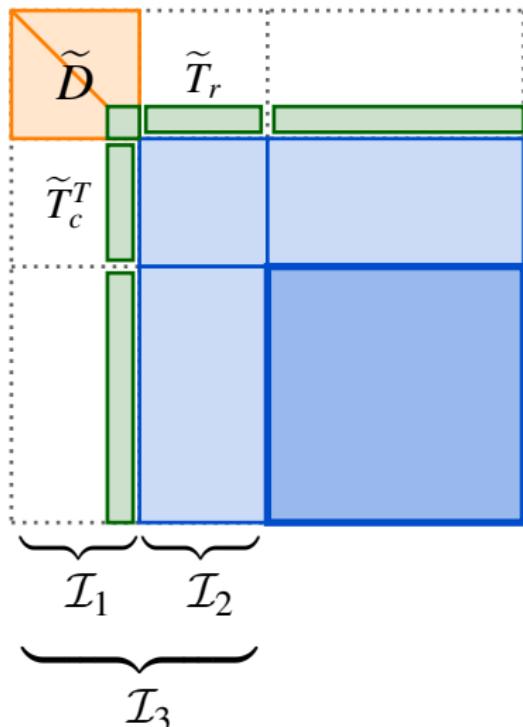
2. Compl. basis with  $V_\perp$ :  
 $(V_\perp, V)$  is unitary

3. Change basis:  
 $\tilde{D} = (V_\perp, V)^H D (V_\perp, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L \\ D_U & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_U \\ D_U & I \end{pmatrix} = \tilde{D}$$

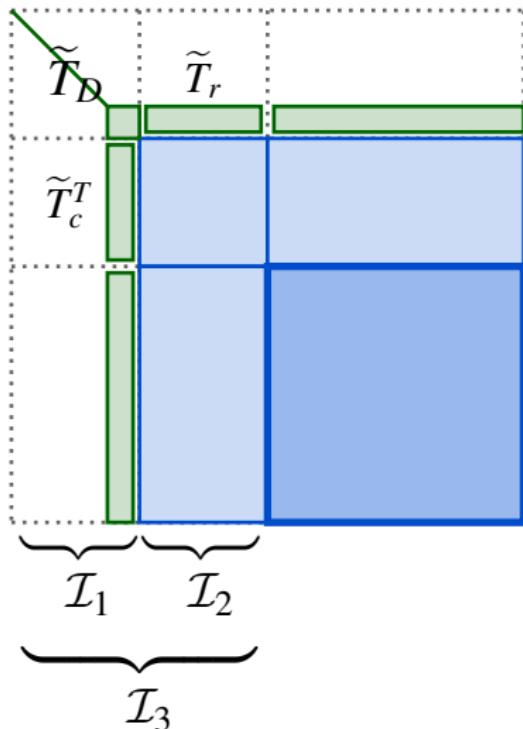
# Structured factorization



HSS node 1:

1. Approximate:  
 $( T_r , T_c^H ) \approx V( \tilde{T}_r , \tilde{T}_c^H )$
2. Compl. basis with  $V_\perp$ :  
 $( V_\perp , V )$  is unitary
3. Change basis:  
 $\tilde{D} = ( V_\perp , V )^H D ( V_\perp , V )$
4. Partially factorize  $\tilde{D}$ :  
$$\begin{pmatrix} D_L & \tilde{T}_D \\ D_c & \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



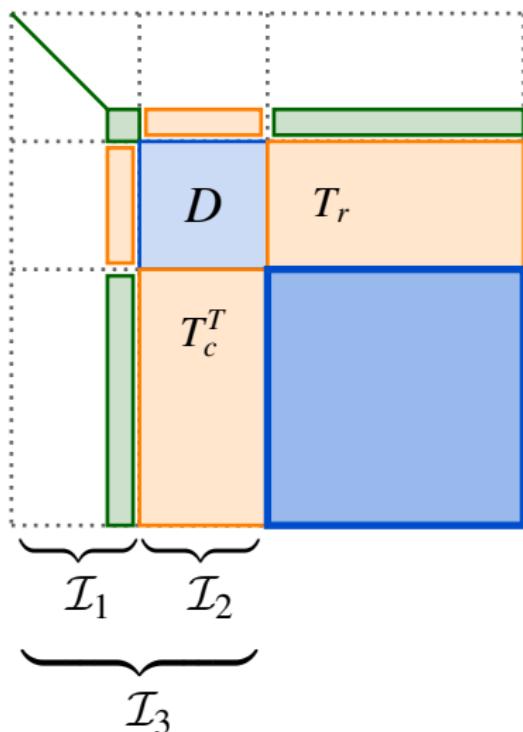
HSS node 1:

1. Approximate:  
 $(\tilde{T}_r, \tilde{T}_c^H) \approx V(\tilde{T}_r, \tilde{T}_c^H)$
2. Compl. basis with  $V_\perp$ :  
 $(V_\perp, V)$  is unitary
3. Change basis:  
 $\tilde{D} = (V_\perp, V)^H D (V_\perp, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node 2:

1. Approximate:  
 $( T_r , T_c^H ) \approx V( \tilde{T}_r , \tilde{T}_c^H )$

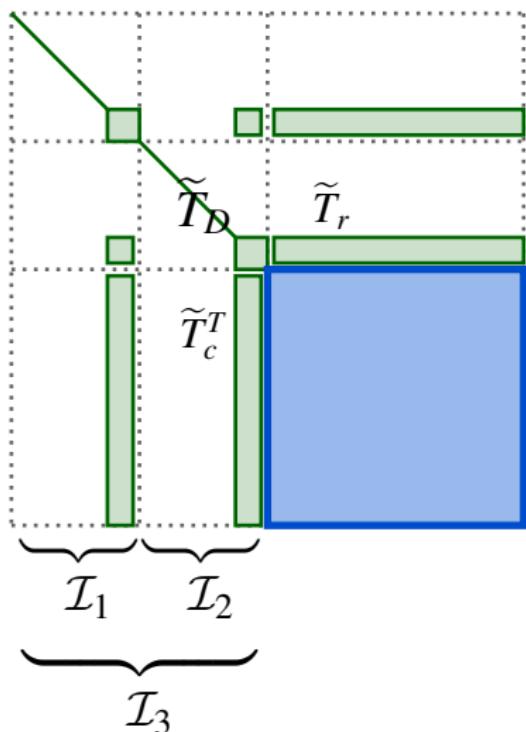
2. Compl. basis with  $V_\perp$ :  
 $( V_\perp , V )$  is unitary

3. Change basis:  
 $\tilde{D} = ( V_\perp , V )^H D ( V_\perp , V )$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



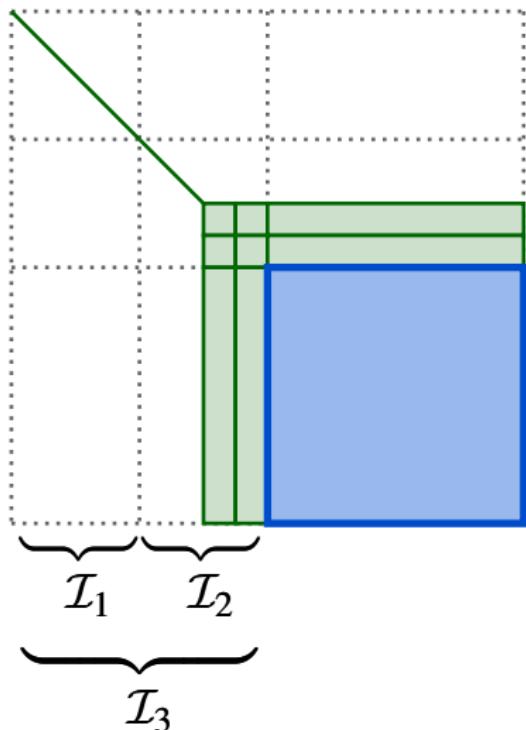
HSS node 2:

1. Approximate:  
 $( T_r , T_c^H ) \approx V( \tilde{T}_r , \tilde{T}_c^H )$
2. Compl. basis with  $V_\perp$ :  
 $( V_\perp , V )$  is unitary
3. Change basis:  
 $\tilde{D} = ( V_\perp , V )^H D ( V_\perp , V )$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



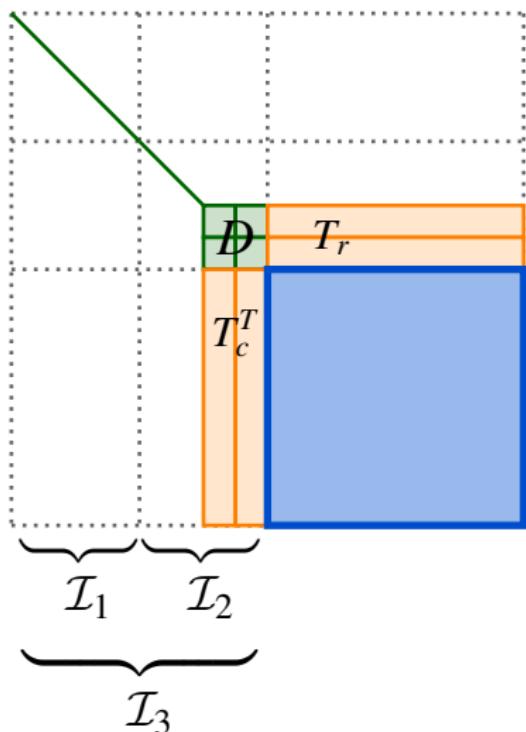
HSS node 2:

1. Approximate:  
 $(T_r, T_c^H) \approx V(\tilde{T}_r, \tilde{T}_c^H)$
2. Compl. basis with  $V_\perp$ :  
 $(V_\perp, V)$  is unitary
3. Change basis:  
 $\tilde{D} = (V_\perp, V)^H D (V_\perp, V)$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



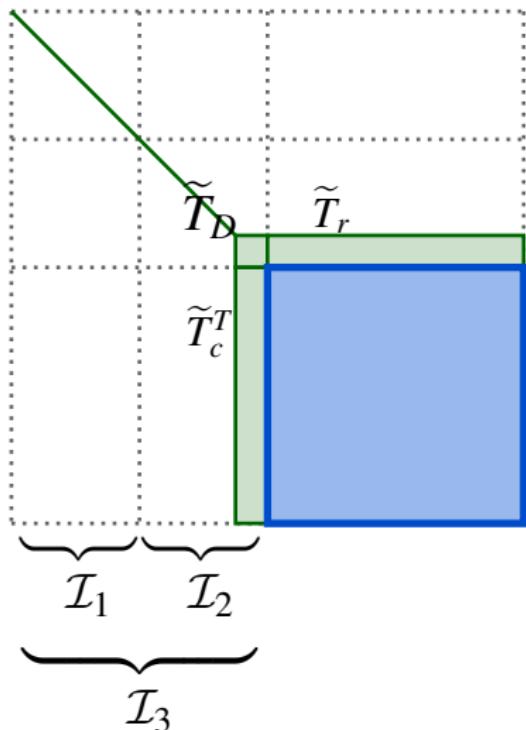
HSS node 3:

1. Approximate:  
 $( T_r , T_c^H ) \approx V( \tilde{T}_r , \tilde{T}_c^H )$
2. Compl. basis with  $V_\perp$ :  
 $( V_\perp , V )$  is unitary
3. Change basis:  
 $\tilde{D} = ( V_\perp , V )^H D ( V_\perp , V )$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & D_r \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_r \\ I & \end{pmatrix} = \tilde{D}$$

# Structured factorization



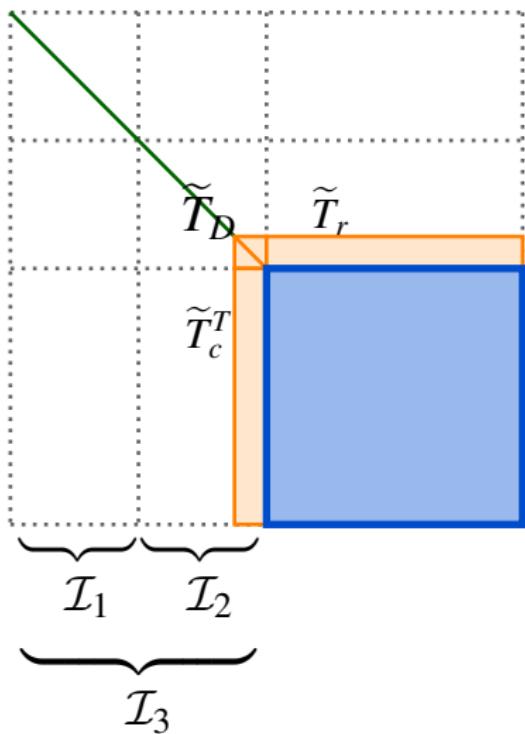
HSS node 3:

1. Approximate:  
 $( T_r , T_c^H ) \approx V( \tilde{T}_r , \tilde{T}_c^H )$
2. Compl. basis with  $V_\perp$ :  
 $( V_\perp , V )$  is unitary
3. Change basis:  
 $\tilde{D} = ( V_\perp , V )^H D ( V_\perp , V )$

4. Partially factorize  $\tilde{D}$ :

$$\begin{pmatrix} D_L & \\ D_c & \tilde{T}_D \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \tilde{D}$$

# Structured factorization



HSS node 3:

5. Partially factorize:

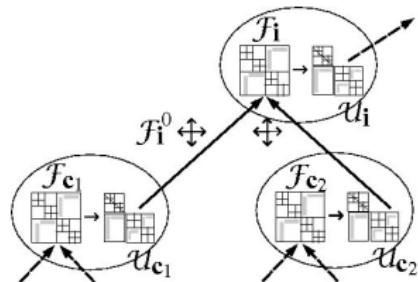
$$\begin{pmatrix} D_L & \\ D_c & \mathcal{U} \end{pmatrix} \begin{pmatrix} D_U & D_r \\ & I \end{pmatrix} = \begin{pmatrix} \tilde{T}_D & \tilde{T}_r \\ \tilde{T}_c & \tilde{T}_U \end{pmatrix}$$

# Embedding HSS in multifrontal

Approximate **Frontal & Update** matrices by HSS

Need following operations:

- **frontal HSS factorization of  $F_i$**
- **extend-add of two HSS update matrices  $U_i$  and  $U_j$**



Final Cholesky factor: Classical vs HSS-embedded



# Theory

- Schur monotonicity for approximate Cholesky factorization
- conditioning analysis
- rank analysis for discretized PDEs

# Schur monotonicity for approximate Cholesky $A = R^T R$

$$R = \begin{pmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,n} \\ & R_{2,2} & \cdots & R_{2,n} \\ & & \ddots & \vdots \\ & & & R_{n,n} \end{pmatrix}, \quad R \approx S = \begin{pmatrix} R_1 & \tilde{R}_{1,2} & \cdots & \tilde{R}_{1,n} \\ & R_2 & \cdots & \tilde{R}_{2,n} \\ & & \ddots & \vdots \\ & & & R_n \end{pmatrix}$$

First approximation step:  $R_1^T R_1 = A_{11}$  and  $H_1 = R_1^{-T} A_{1,2:n}$

$$H_1 = \begin{pmatrix} U_1 & \widehat{U}_1 \end{pmatrix} \begin{pmatrix} \mathcal{Q}_1 & \widehat{\mathcal{Q}}_1 \end{pmatrix}^T, \quad H_1^T H_1 = \mathcal{Q}_1 \mathcal{Q}_1^T + \widehat{\mathcal{Q}}_1 \widehat{\mathcal{Q}}_1 e^T, \quad \|\widehat{\mathcal{Q}}_1\|_2 \leq \tau$$

Orthogonal dropping:  $\tilde{H}_1 = U_1 \mathcal{Q}_1 \longrightarrow \tilde{H}_1^T (H_1 - \tilde{H}_1) = 0$

Schur complement:  $\mathcal{A}_1 = A_{2:n,2:n} - H_1^T H_1 = A_{2:n,2:n} - \mathcal{Q}_1 \mathcal{Q}_1^T - \widehat{\mathcal{Q}}_1 \widehat{\mathcal{Q}}_1^T$ .

Approximate  $\mathcal{A}_1$  by  $\tilde{\mathcal{A}}_1 = A_{2:n,2:n} - \mathcal{Q}_1 \mathcal{Q}_1^T = \mathcal{A}_1 + \widehat{\mathcal{Q}}_1 \widehat{\mathcal{Q}}_1^T = \mathcal{A}_1 + o(\tau^2)$

Nice Property: Successive Schur complements do not decrease in SPD sense  
⇒ factorization is breakdown free

1. M. Gu, X.S. Li, P. Vassilevski, “Direction-Preserving and Schur-Monotonic Semiseparable Approximations of Symmetric Positive Definite Matrices”, SIMAX, 31 (5), 2650-2664, 2010.
2. J. Xia, M. Gu, “Robust approximate Cholesky factorization of rank-structured symmetric positive definite matrices”, SIMAX, 31 (5), 2899-2920, 2010.

# Conditioning analysis when $R^T R$ as preconditioner (Napov)

- Goal: analyze spectral condition number  $\kappa(R^{-T} A R^{-1})$
- Sketch: look at approximation after each step  $k$  of total  $l$  steps; capture different approximation order:

$$B_k = \begin{pmatrix} \mathcal{R}_{11}^{(k)T} & \\ \widetilde{\mathcal{R}}_{12}^{(k)T} & \widetilde{S}_B^{(k)} \end{pmatrix} \begin{pmatrix} \mathcal{R}_{11}^{(k)} & \widetilde{\mathcal{R}}_{12}^{(k)} \\ & I \end{pmatrix}, \quad \widetilde{S}_B^{(k)} = A_{i_k+1:n, i_k+1:n} - \widetilde{\mathcal{R}}_{12}^{(k)T} \widetilde{\mathcal{R}}_{12}^{(k)}$$

- SSS bound (sequential order):

$$\kappa(R^{-T} A R^{-1}) \leq \prod_{k=1}^l \frac{1 + \gamma_k}{1 - \gamma_k}, \quad \text{where } \gamma_k = \|(\mathcal{R}_{12}^{(k)} - \widetilde{\mathcal{R}}_{12}^{(k)}) \widetilde{S}_B^{(k)}\|^{-1/2} < 1$$

- HSS bound: can be computed numerically using good estimates
- $\gamma_k$  estimate:  $\gamma_k \leq \|(\mathcal{R}_{12}^{(k)} - \widetilde{\mathcal{R}}_{12}^{(k)})\| \|A^{-1}\|^{1/2}$ 
  - ▶ can estimate  $\|A^{-1}\|$  with a few iterations of Conjugate Gradient
- Adaptive threshold strategy based on  $\gamma_k$  estimate

A. Napov, “Conditioning analysis of incomplete Cholesky factorizations with orthogonal dropping”, to appear in SIMAX.

# Rank analysis for some PDEs

Consider  $n \times m$  grid, lexicographical (layer-by-layer ordering) order gives rise to block tridiagonal matrix:

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ \ddots & \ddots & \ddots & \\ & A_{m,m-1} & A_{m-1,m} & A_{m,m} \end{pmatrix}, \text{ each } n \times n \text{ Schur compl. } S_{i+1} = A_{i,i} - A_{i,i-1}S_i^{-1}A_{i,i-1}$$

Model problem (Chandrasekaran et al.):

$$A_{i,i} = A_{j,j}, \quad A_{i-1,i} = A_{j-1,j}, \quad A_{i,i-1} = A_{j,j-1}$$

- In 2D,  $\varepsilon$ -rank of the off-diagonal Hankel block is constant, for  $n \rightarrow \infty$
- In 3D ( $k^3$ ),  $\varepsilon$ -rank of the strip Hankel block is bounded by  $O(k)$

Helmholtz equations with constant velocity (Engquist, Ying):

look at the Green's function of the Helmholtz operator.

- In 2D ( $k^2$ ),  $\varepsilon$ -rank of the off-diagonal block  $O(\log k)$
- In 3D ( $k^3$ ),  $\varepsilon$ -rank of the off-diagonal block  $O(k)$

1. S. Chandrasekaran, P. Dewilde, M. Gu, and N. Somasunderam, "On the Numerical Rank of the Off-Diagonal Blocks of Schur Complements of Discretized Elliptic PDEs", SIMAX, 31 (5), 2261-2290, 2010.
2. B. Engquist, L. Ying, "Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation", Communications in Pure and Applied Mathematics 64 (2011).

# Complexity of HSS-embedded multifrontal factorization

- With ND order, the intermediate Schur complements may have slightly higher ranks, but **no more than twice**:

$$A = \begin{pmatrix} A_{11} & & A_{33} \\ & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}, \quad \text{Schur compl. } S = A_{33} - A_{31}A_{11}^{-1}A_{13} - A_{32}A_{22}^{-1}A_{32}$$

Each contribution  $-A_{31}A_{11}^{-1}A_{13}$  (or  $-A_{32}A_{22}^{-1}A_{32}$ ) satisfies the off-diagonal rank bound, together, the off-diagonal rank bound of  $S$  is at most twice as that of layer-by-layer order.

- Given the rank bounds, can show the following cost of the HSS-MF factorization algorithm:

Problem		$r$	MF		HSS-MF	
			flops	fill	flops	fill
2D ( $k^2$ )	Elliptic	$O(1)$	$O(n^{3/2})$	$O(n \log n)$	$O(n \log n)$	$O(n \log \log n)$
	Helmholtz	$O(\log k)$				
3D ( $k^3$ )	Elliptic	$O(k)$	$O(n^2)$	$O(n^{4/3})$	$O(n^{4/3} \log n)$	$O(n \log n)$
	Helmholtz	$O(k)$				

J. Xia, “Efficient structured multifrontal factorization for general large sparse matrices”, SISC, 35 (2), A832-A860, 2012.

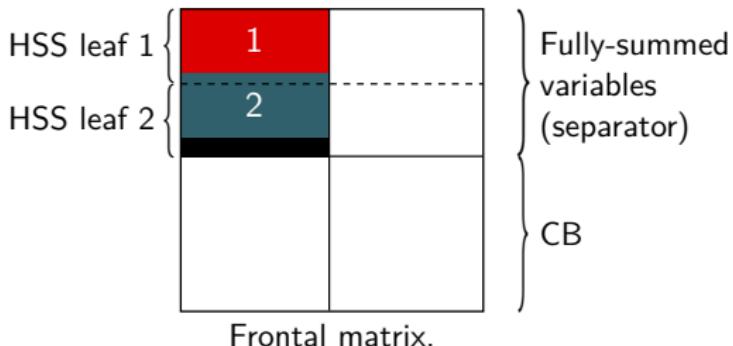
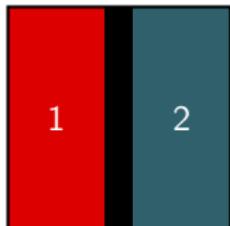
**Warning:** The constant prefactor may be large:  $\sim O(100s)$  ( $\sim 10$  for classical)

# Practice

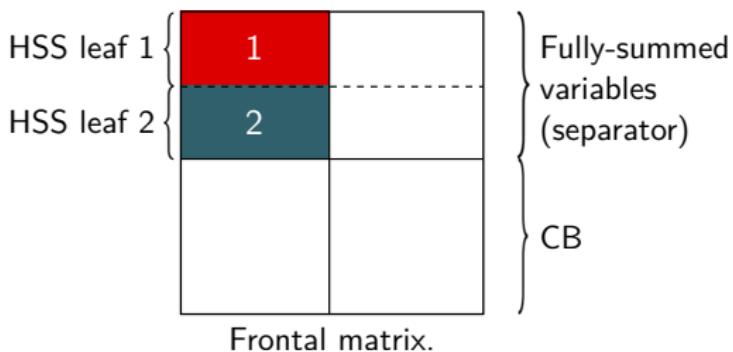
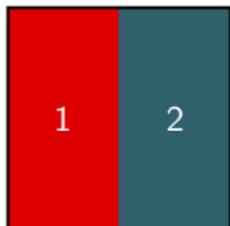
- ordering within separator
- parallelization
- preconditioning

# Separator ordering: vertex-based vs edge-based

Vertex-based approach:



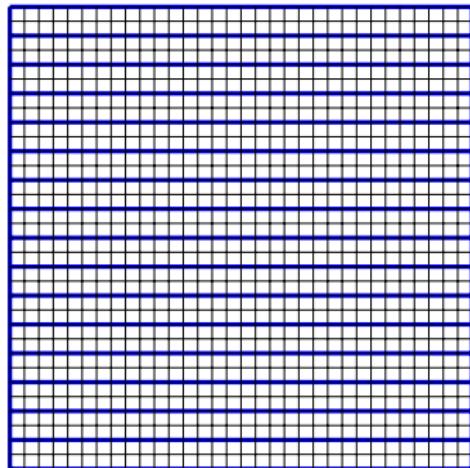
Edge-based approach:



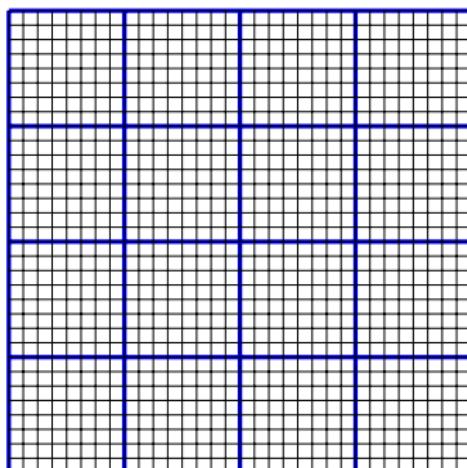
An edge-based ordering allows us to simply match parts of the separators with nodes of the HSS tree.

## Ordering of separators: shape

In order to ensure some kind of admissibility condition, parts should have a **small diameter**.



Large diameters.



Small diameters.

For simplicity, we divide the separator into **square blocks** (chessboard).

Cf. also talk by **J.-Y. L'Excellent** for the influence of diameter vs rank.

## Ordering of separators: ordering of blocks

In the **HSS compression** stage, blocks are merged two-by-two following a tree flow. **Merged blocks should also have small diameter**, thus the partitioning should have some recursive property.

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

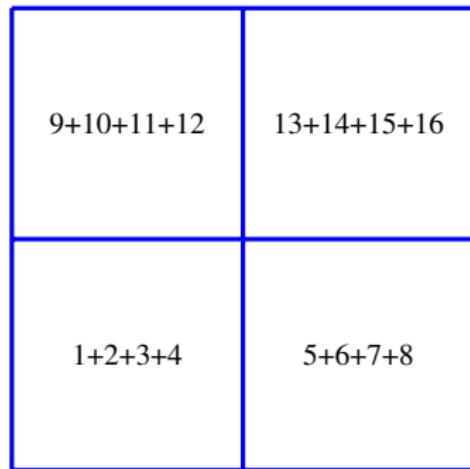
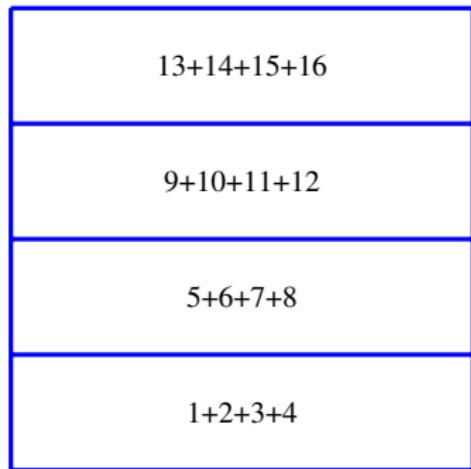
10	11	15	16
9	10	13	14
3	4	7	8
1	2	5	6

Leaf level.

We use an **edge-based Nested Dissection** (we cut the domain into squares, and order these squares using ND/Morton ordering).

## Ordering of separators: ordering of blocks

In the **HSS compression** stage, blocks are merged two-by-two following a tree flow. **Merged blocks should also have small diameter**, thus the partitioning should have some recursive property.



Two levels above leaves: blocks are merged four-by-four.

We use an **edge-based Nested Dissection** (we cut the domain into squares, and order these squares using ND/Morton ordering).

# Results - topmost separator

Topmost separator of a  $200^3$  domain ( $200 \times 200$  plane). We compare:

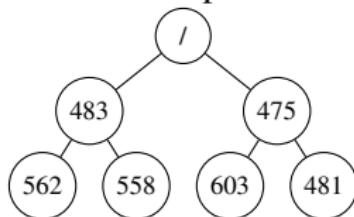
**VND:** vertex-based ND.

**Nat:** square blocks ordering in natural/lexicographic order.

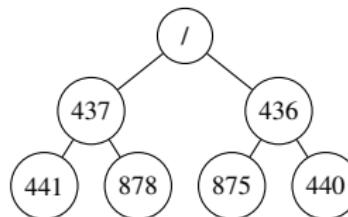
**END:** edge-based ND: square blocks ordered in ND.

	VND	Nat	END
Total HSS time (s)	55.0	51.8	32.3
Max rank	731	893	646
Min time in RRQR (s)	15.2	20.3	11.0
Max time in RRQR (s)	53.0	50.2	30.7

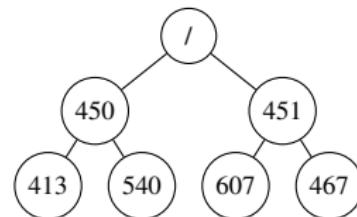
Ranks at the top of HSS trees:



VND.



Nat.



END.

END yields lower rank and better balance of ranks.

## Results - complete problem

Helmholtz equations with PML boundary

$$\left( -\Delta - \frac{\omega}{v(x)^2} \right) u(x, \omega) = s(x, \omega)$$

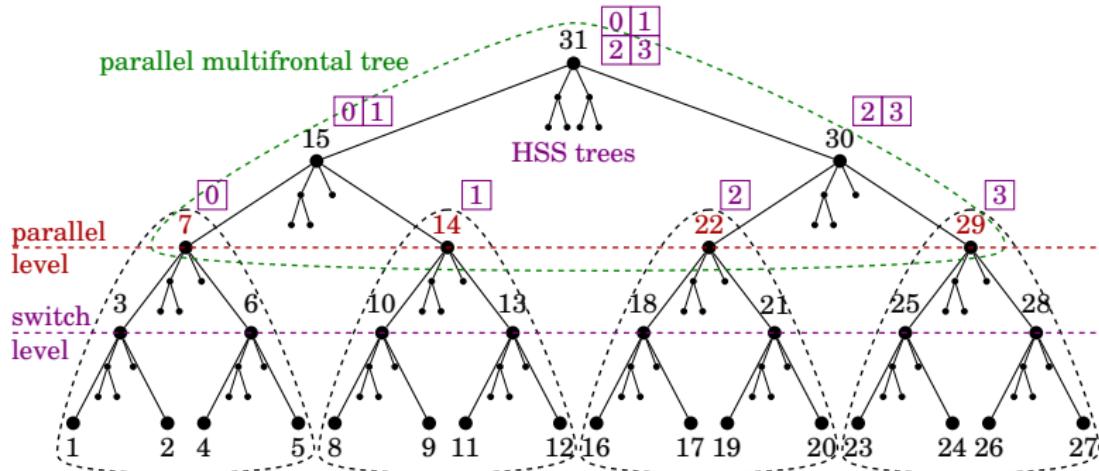
On the complete problem, with 256 cores and HSS compression at the 8 topmost levels of the tree:

	VND	Nat	END
Total factorization time (s)	984.8	978.5	938.0
Max rank	865	893	868
Min time in RRQR (s)	304.8	322.1	310.9
Max time in RRQR (s)	674.9	683.8	654.7

END: marginal ( 5%) improvement in run time but better workload balance, so hopefully more potential for strong scaling.

# Parallelization: two types of tree-based parallelism

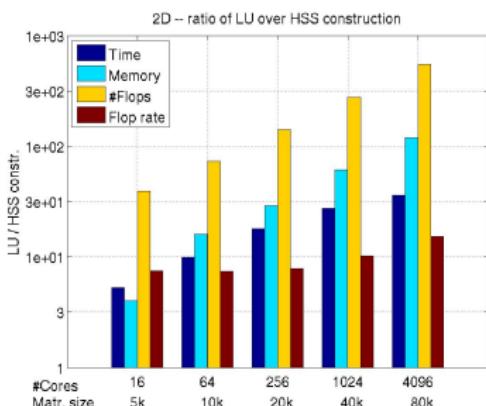
- **Outer tree:** separator tree for multifrontal factorization
- **Inner tree:** HSS tree at each internal separator node



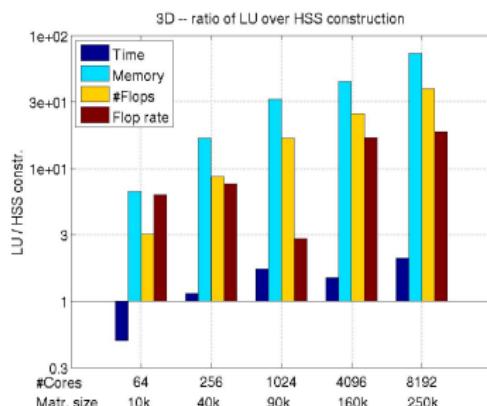
# Parallel HSS performance

- HSS construction on the last Schur complement corresp. to the top separator.

Performance ratio of LU over HSS:



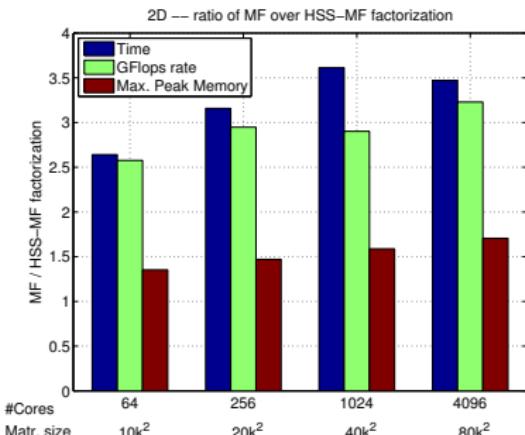
(a) 2D, max\_rank=7



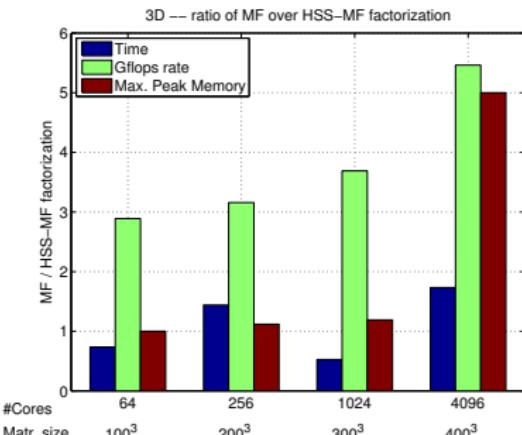
(b) 3D, max\_rank=848

S. Wang, X.S. Li, J. Xia, and M.V. de Hoop, “Efficient scalable algorithms for solving linear systems with hierarchically semiseparable matrices”, SISC, Nov. 2012. (revised)

# Parallel HSS-MF performance



(c) 2D Helmholtz, 10Hz



(d) 3D Helmholtz, 5Hz

HSS-MF succeeded with  $600^3$  on 16,384 cores, while MF failed.

S. Wang, X.S. Li, F.-H. Rouet, J. Xia, and M. de Hoop, “A Parallel Geometric Multifrontal Solver Using Hierarchically Semiseparable Structure”, ACM TOMS, June 2013. (in submission)

# Sparse results - 2D problems

2D Helmholtz problems on square grids (mesh size  $k$ ,  $N = k^2$ ), 10 Hz.

	$k$	10,000	20,000	40,000	80,000
	$P$	64	256	1,024	4,096
MF	Factorization (s)	258.6	544.8	1175.8	2288.5
	Gflops/s	507.3	2109.3	8185.6	31706.9
	Solution+refinement (s)	10.4	10.8	11.5	11.6
	Factors size (GB)	120.1	526.7	2291.2	9903.7
	Max. peak (GB)	2.3	2.5	2.7	2.9
	Communication volume (GB)	136.2	1202.5	9908.1	79648.4
HSS	HSS+ULV (s)	97.9	172.5	325.3	659.3
	Gflops/s	196.9	715.6	2820.7	9820.6
	Solution+refinement (s)	20.2	55.4	61.4	115.8
	Steps	3	3	9	9
	Factors size (GB)	66.2	267.7	1333.2	4572.3
	Max. peak (GB)	1.7	1.7	1.7	1.7
	Communication volume (GB)	74.2	573.8	4393.4	41955.8
	HSS rank	258	503	1013	2015
	$\ x - x_{\text{MF}}\  / \ x_{\text{MF}}\ $	$1.5 \times 10^{-5}$	$2.2 \times 10^{-5}$	$3.1 \times 10^{-5}$	$3.5 \times 10^{-6}$
	$\max_i \frac{ Ax-b _i}{( A  x + b )_i}$	$7.1 \times 10^{-6}$	$1.0 \times 10^{-5}$	$2.0 \times 10^{-6}$	$3.5 \times 10^{-6}$

# Results - 3D problems

3D Helmholtz problems on cubic grids (mesh size  $k$ ,  $N = k^3$ ), 5 Hz.

	$k$	100	200	300	400
	$P$	64	256	1,024	4,096
MF	Factorization (s)	88.4	1528.0	1175.8	6371.6
	Gflops/s	600.6	2275.7	9505.6	35477.3
	Solution+refinement (s)	0.6	2.2	3.5	4.8
	Factors size (GB)	16.6	280.0	1450.1	4636.1
	Max. peak (GB)	0.5	1.9	2.5	2.0
	Communication volume (GB)	83.1	2724.7	26867.8	165299.3
HSS	HSS+ULV (s)	120.4	1061.3	2233.8	3676.5
	Gflops/s	207.8	720.4	2576.6	6494.8
	Solution+refinement (s)	2.3	8.2	31.5	182.8
	Steps	4	5	10	6
	Factors size (GB)	10.7	112.9	434.3	845.3
	Max. peak (GB)	0.5	1.7	2.1	0.4
	Communication volume (GB)	93.6	2241.2	18621.1	143300.0
	HSS rank	481	925	1391	1860
	$\ x - x_{\text{MF}}\  / \ x_{\text{MF}}\ $	$6.2 \times 10^{-6}$	$9.4 \times 10^{-7}$	$1.1 \times 10^{-6}$	$1.7 \times 10^{-6}$
	$\max_i \frac{ Ax-b _i}{( A  x + b )_i}$	$1.5 \times 10^{-7}$	$5.7 \times 10^{-7}$	$9.7 \times 10^{-7}$	$3.7 \times 10^{-6}$

# Preconditioning results

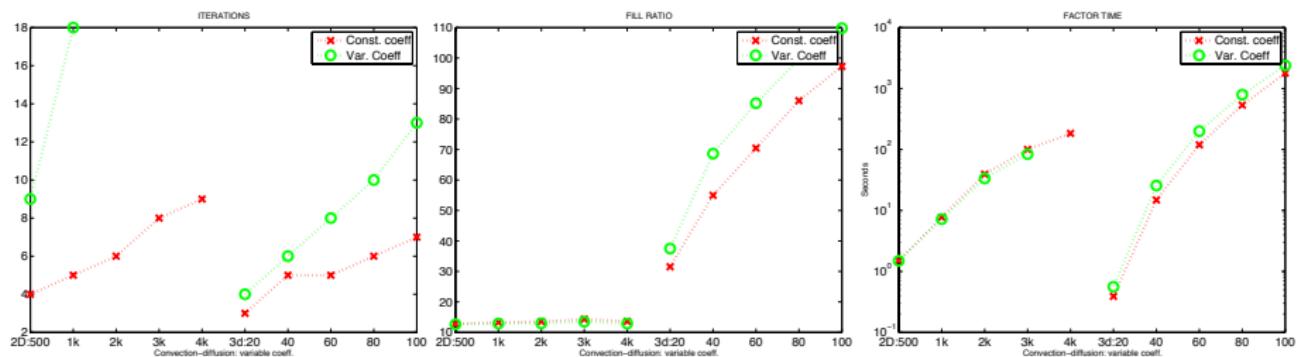
- Test matrices: 2D & 3D
  - ▶ model problems
  - ▶ convection-diffusion: constant coefficient, variable coefficient
  - ▶ curl-curl edge elements (Nedelec elements)
  - ▶ Helmholtz
  - ▶ general matrices
- $\text{RHS} = (1, 1, \dots)^T$
- GMRES(30)
  - ▶ right preconditioner
  - ▶ initial  $x_0 = (0, 0, \dots)^T$
  - ▶ stopping criterion:  $\frac{\|r_k\|_2}{\|b\|_2} \leq 10^{-6}$

# Convection-diffusion

$$-\nu \Delta u + \mathbf{v} \cdot \nabla u = \mathbf{b} \quad \text{on } \Omega.$$

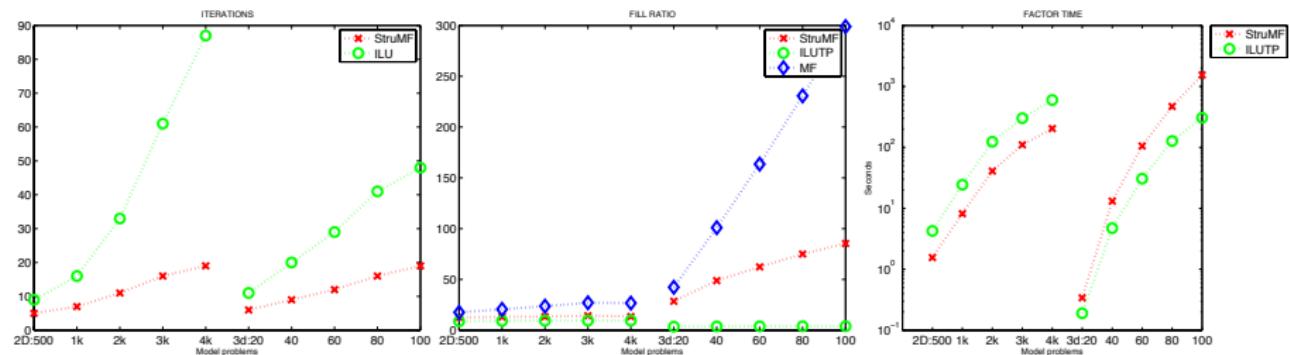
$\mathbf{v} = \dots$

	constant coeff.	variable coeff.
2D	$(1/\sqrt{2} \quad 1/\sqrt{2})$	$(x(1-x)(2y-1) \quad y(1-y)(2x-1))$
3D	$(1/2 \quad 1/2 \quad 1/\sqrt{2})$	$(x(1-x)(2y-1)z \quad y(y-1)(2x-1) \quad (2x-1)(2y-1)z(z-1))$



# Model problems: $\Delta u = f$

- Compare to ILU in SuperLU (Li/Shao 10)
  - ▶ supernode-based ILUTP, threshold, partial pivoting
  - ▶  $10^{-4}$  for HSS truncation, and ILU threshold



# General matrices

Matr.	Descr.	N	HSS-rank	Fill-ratio		Factor (s)		Its	
				HSS	ILU	HSS	ILU	HSS	ILU
add32	circuit	4,690	0	2.1	1.3	0.01	0.01	1	2
mchln85ks17	car tire	84,180	948	13.5	12.3	133.8	216.1	4	39
mhd500	plasma	250,000	100	11.6	15.6	2.5	7.9	2	8
poli_large	economic	15,575	64	4.8	1.6	0.04	0.02	1	2
stomach	bio eng.	213,360	92	12.1	2.9	13.8	18.7	2	2
tdr190k	accelerator	1,100242	596	14.1	–	629.2	–	7	–
torso3	bio eng.	259,156	136	22.6	2.4	86.7	63.7	2	2
utm5940	TOKAMAK	5,940	123	6.7	8.0	0.1	0.16	3	15
wang4	device	26,068	385	45.3	23.1	4.4	6.4	3	4

# Summary

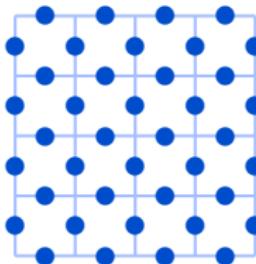
- More theory has been developed
- In practice: very promising for large problems, large machines
  - ▶ demonstrated that it is implementable in parallel, with reduced communication
- Compare to ILU preconditioner
  - ▶ breakdown free
  - ▶ More parallel
  - ▶ Dropping operation may be more expensive (row/col vs. entry-wise in ILU)

# Extra Slides

# Curl-curl edge element (Nedelec element)

$$\nabla \times \nabla \times u + \beta = \mathbf{b} \text{ on } \Omega$$

$$\tau = 10^{-8}$$



mesh size	HSS-rank	fill-ratio	factor (s)	Its	GMRES (s)
$500^2$	59	14.8e	5.8	2	0.5
$1000^2$	52	14.9	25.0	3	3.3
$2000^2$	60	14.9	106.1	3	13.3
$3000^2$	50	14.4	114.6	5	48.7
$20^3$	388	78.9	6.7	2	0.1
$40^3$	824	125.6	261.5	3	1.7
$60^3$	804	156.1	2055.8	3	7.4

# Helmholtz

$$\left( -\Delta - \frac{\omega}{v(x)^2} \right) u(x, \omega) = s(x, \omega)$$
$$\tau = 10^{-4}$$

mesh size	HSS-rank	fill-ratio	factor (s)	Its	GMRES (s)
$500^2$	85	8.8	8.6	4	1.6
$1000^2$	210	9.5	53.1	4	6.5
$2000^2$	229	9.7	307.1	71	500.1
$3000^2$	380	10.0	950.2	139	2464.1
$20^3$	275	13.3	2.4	3	0.1
$40^3$	533	27.0	151.9	3	1.1
$60^3$	1039	38.8	1434.6	3	5.3
$80^3$	1167	47.3	7708.1	3	16.8

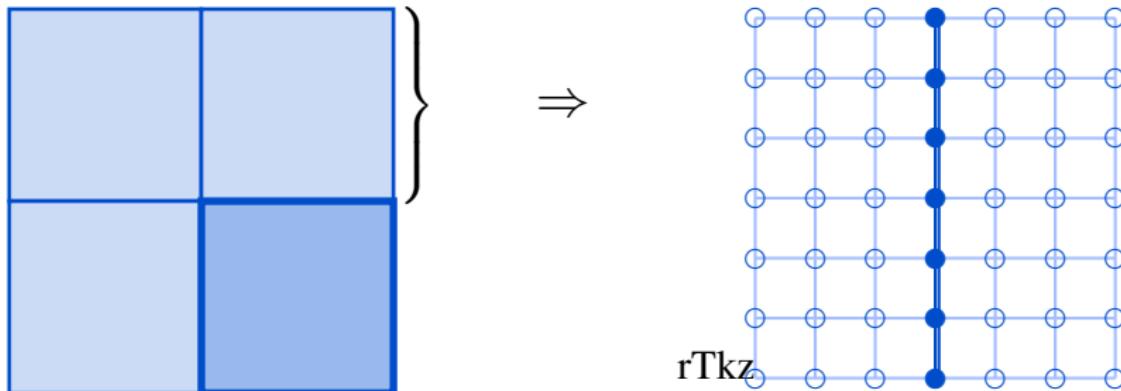
# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

2D model separators:



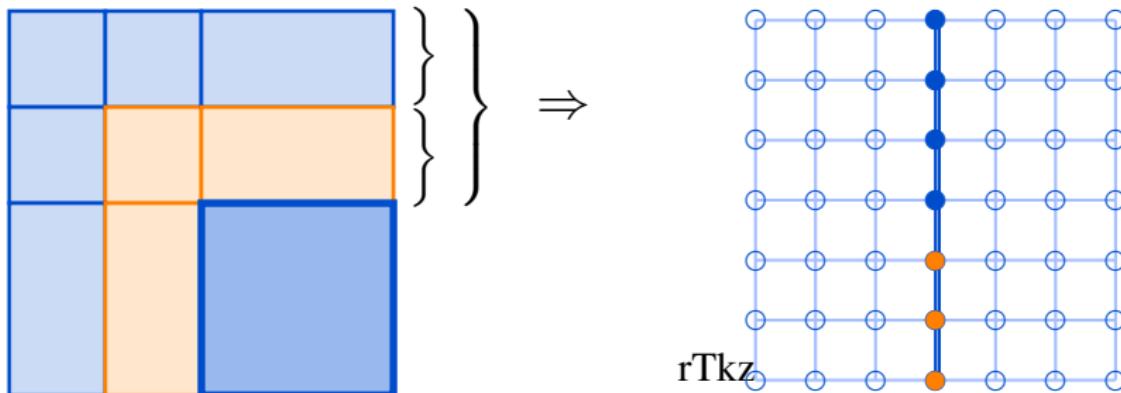
# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

2D model separators:



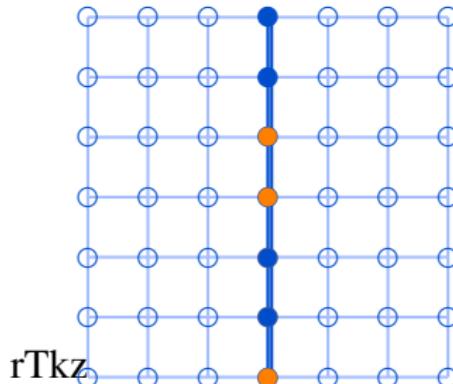
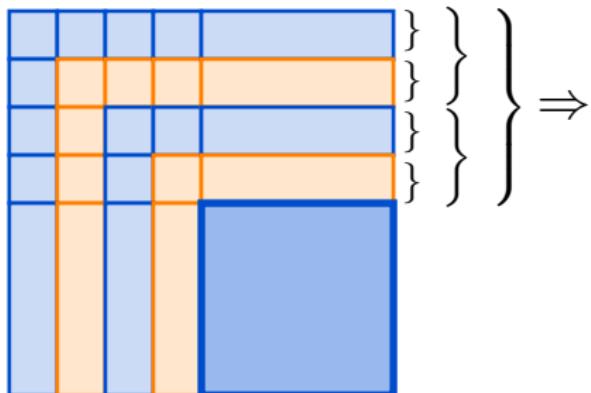
# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

2D model separators:



rTkz

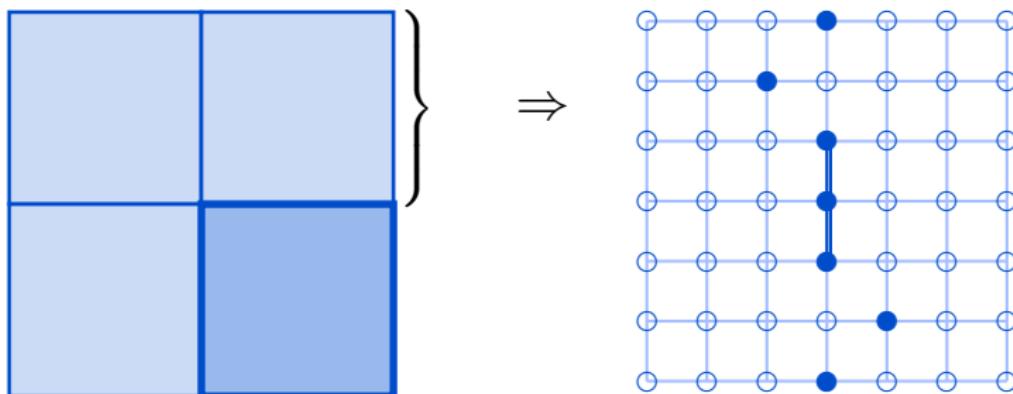
# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

2D separators from Scotch:



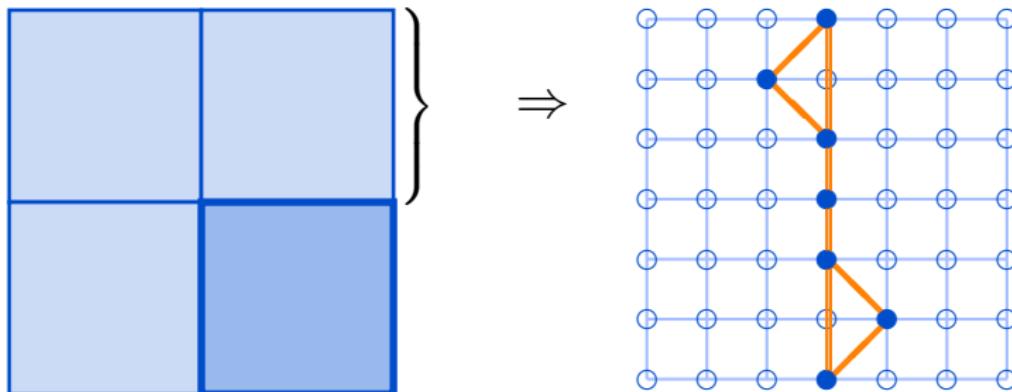
# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

2D separators from Scotch: connections of length 2 outside separator



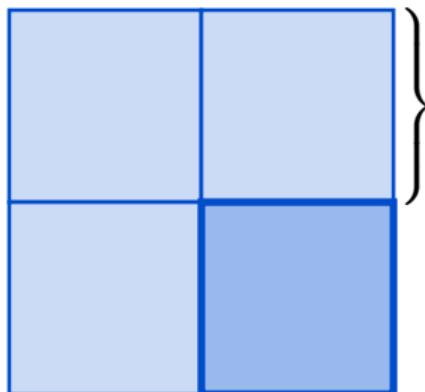
# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

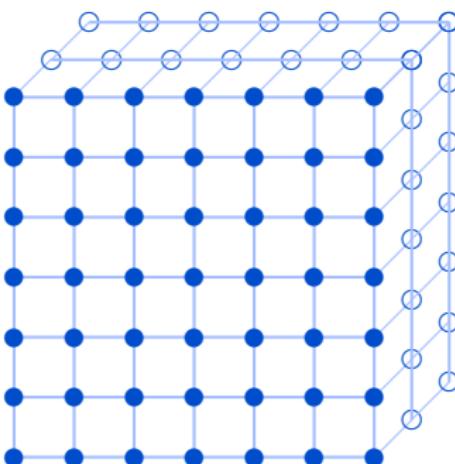
rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

3D model separators:



⇒



# Separator ordering

Model problem analysis [Chandrasekaran et al., 10]:

rank is bounded proportionally to the number of concoctions between the corresp. subset and the remaining nodes of separator

Implemented: Recursive edge bisection ordering (via METIS)

3D model separators:

