Towards an optimal-order approximate sparse factorization exploiting data-sparseness in separators

X. Sherry Li
xsli@lbl.gov
Lawrence Berkeley National Laboratory

Artem Napov (Univ. of Brussels), Francois-Henry Rouet (LBNL), Shen Wang (ConocoPhillips), Jianlin Xia (Purdue Univ.)

Sparse Days, June 17-18, CERFACS

Motivation: sparse factorization

✓ Numerically robust, efficient for many RHS.

✗ Expensive. For model problems on regular grids (mesh size k):

- 2D ($N = k^2$): flops $= \Theta(N \log N)$, memory $= \Theta(N^{3/2})$.
- 3D ($N = k^3$): flops $= \Theta(N^{4/3})$, memory $= \Theta(N^2)$.

Example: finite difference, $1000 \times 1000 \times 1000$ grid, 27pt stencil:
flops=50 exaflops, mem>200 TB (the whole Cray XE6 system at NERSC!).

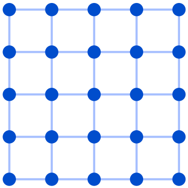
Idea

Resort to approximation.

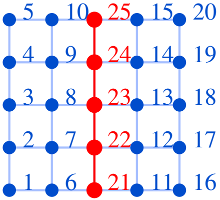
Outline

- Review of multifrontal LU
- Structured partial factorization: HSS-embedded multifrontal
- Parallelization, performance
- Remarks

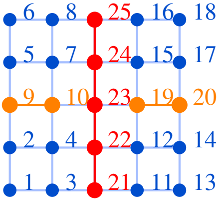
Nested dissection ordering



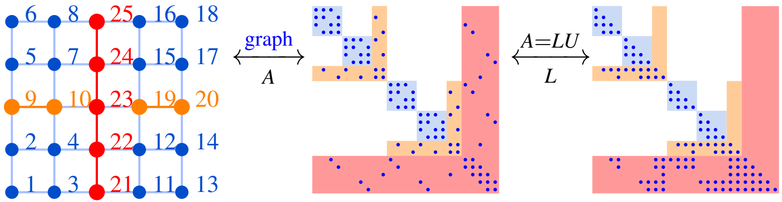
Nested dissection ordering



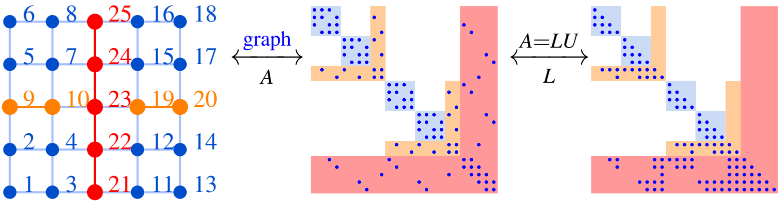
Nested dissection ordering



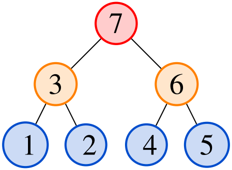
Nested dissection ordering

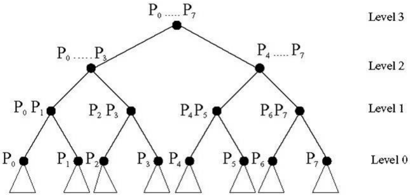


Nested dissection ordering

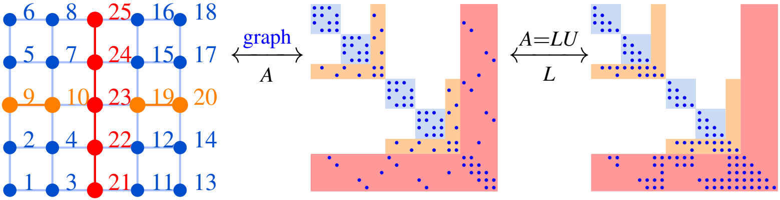


- Better parallel properties



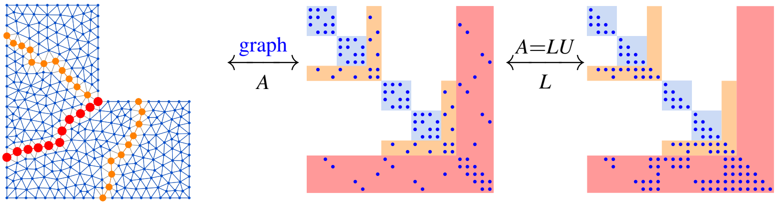


Nested dissection ordering



- Better parallel properties
- Fewer entries in factors (here, 129 vs. 116)
- Implementation: using Scotch partitioner, since separator info needed

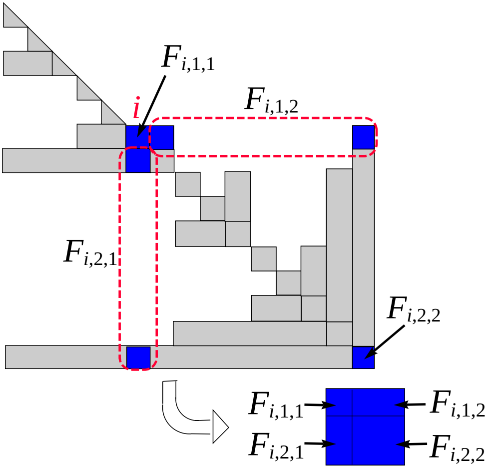
Nested dissection ordering



- Better parallel properties
- Fewer entries in factors (here, 129 vs. 116)
- Implementation: using Scotch partitioner, since separator info needed

Frontal and Update matrices

- Each separator corresp. to a dense submatrix (frontal & update)
  - Dominant cost: $F_{1,1} = LU$, $F_{2,1}$, $F_{1,2}$, Update matrix



Observation : presence of data-sparseness

Several approches are available in the literature that exploit low-rank in dense / structured matrices:

- Hierarchical matrices: $\mathcal{H}$, $\mathcal{H}^2$, (Bebendorf, Börm, Grasedyck, Hackbusch, . . .)
- Hierarchically/Sequential Semiseparable (HSS / SSS) representations (Chandrasekaran, Dewilde, Gu, Li, Olshevsky, Vandebril, Xia, . . .)
- BLR (Amestoy et al.)

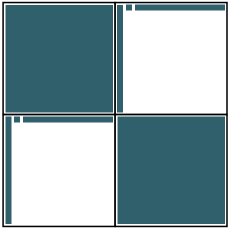
Some representations are simpler and apply to broader classes of problems but provide less gain in memory/operations;
Some others are more complex but allow for further gains in complexity.

We focus on Hierarchically Semi-Separable matrices (HSS) ...

Hierarchically Semi-Separable matrices

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

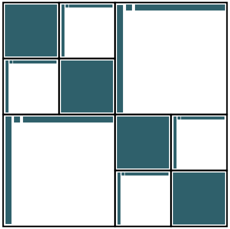
$$U_3 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} U_3^{small}, \; U_3^{small} : 2k \times k$$

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

$$A = \left[\begin{array}{cc|cc} D_1 & U_1B_1V_2^T & \multicolumn{2}{c}{\multirow{2}{*}{$U_3B_3V_6^T$}} \\ U_2B_2V_1^T & D_2 & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$U_6B_6V_3^T$}} & D_4 & U_4B_4V_5^T \\ & & U_5B_5V_4^T & D_5 \end{array}\right]$$



Fundamental property required for efficiency: nested bases

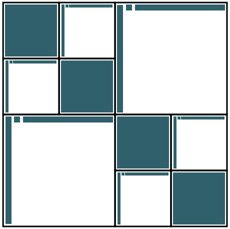
$$U_3 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} U_3^{small}, \; U_3^{small} : 2k \times k$$

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

$$A = \left[\begin{array}{cc|c} D_1 & U_1 B_1 V_2^T & \\ U_2 B_2 V_1^T & D_2 & U_3 B_3 V_6^T \\ \hline & & \begin{array}{c|c} D_4 & U_4 B_4 V_5^T \\ \hline U_5 B_5 V_4^T & D_5 \end{array} \\ \multicolumn{2}{c|}{U_6 B_6 V_3^T} & \end{array}\right]$$

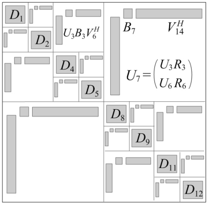


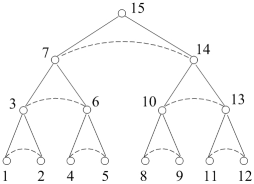
Fundamental property required for efficiency: nested bases

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

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

Hierarchical bases, HSS tree





For efficiency, require:

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

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

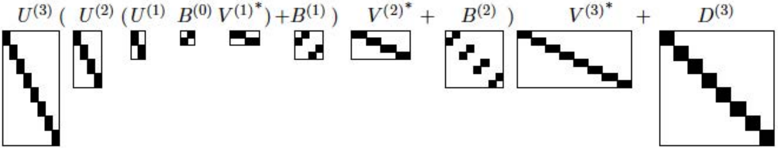
Each basis is a product of descendents' bases:

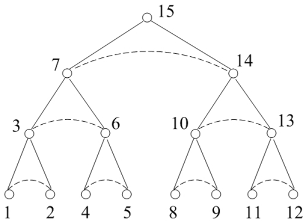
$$U_7 = \begin{bmatrix} U_1 & 0 & 0 & 0 \\ 0 & U_2 & 0 & 0 \\ 0 & 0 & U_4 & 0 \\ 0 & 0 & 0 & U_5 \end{bmatrix} \begin{bmatrix} U_3^{small} & 0 \\ 0 & U_6^{small} \end{bmatrix} U_7^{small},$$

Not to multiply out!

HSS explicit representation (construction)

[Martinsson]





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

HSS operations

With r the maximum rank of a block:

- HSS construction (compression of low-rank blocks): $\Theta(rN^2)$.
- ULV factorization: $\Theta(r^2N)$ [Chandrasekaran et al.].
- HSS solution $\Theta(rN)$.

Provable r for some discretized PDEs: [Chandrasekaran et al.'10, Enquist-Ying'11]

2D Poisson	$\Theta(1)$
2D Helmholtz	$\Theta(\log k)$
3D Poisson	$\Theta(k)$
3D Helmholtz	$\Theta(k)$

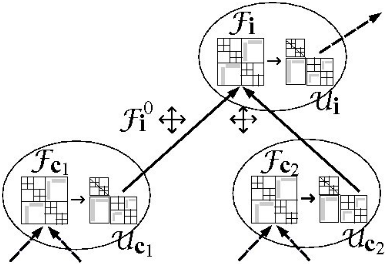
Constructing the HSS structure is the dominant operation.

Embedding HSS in multifrontal

Approximate **Frontal** & **Update** matrices by HSS

Operations:

- HSS construction of frontal $\mathcal{F}_i$, ULV factorize $\mathcal{F}_i(1,1)$
- HSS approximation of $\mathcal{U}_i, \mathcal{U}_j$
- extend-add of HSS matrices $\mathcal{U}_i$ and $\mathcal{U}_j$ to parent



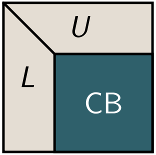
Difficulty: extend-add of two non-matching HSS structures

Embedding HSS in multifrontal

Several ways to deal with frontal matrices:

More complicated ↑ | More memory ↓

- Fully structured: HSS on the whole frontal matrix. No dense matrix.
- Partial+: HSS on the whole frontal matrix. Dense frontal matrix.
- Partially structured: HSS on the L, U part only. Dense frontal matrix, dense CB in stack after partial factorization.



Once a frontal matrix is in partial or complete HSS form, an *ULV* factorization is applied instead of usual LU factorization.

Complexity (for fully-structured) [Xia '11, Chandrasekaran et al. '11]:

Problem	Classical MF		MF with HSS	
	Mem	Flops	Mem	Flops
2D Poisson	$\Theta(N \log N)$	$\Theta(N^{3/2})$	$\Theta(N \log \log N)$	$\Theta(N \log N)$
2D Helmholtz	$\Theta(N \log N)$	$\Theta(N^{3/2})$	$\Theta(N \log \log N)$	$\Theta(N \log N)$
3D Poisson	$\Theta(N^{4/3})$	$\Theta(N^2)$	$\Theta(N \log N)$	$\Theta(N^{4/3})$
3D Helmholtz	$\Theta(N^{4/3})$	$\Theta(N^2)$	$\Theta(N \log N)$	$\Theta(N^{4/3} \log N)$

Experiments

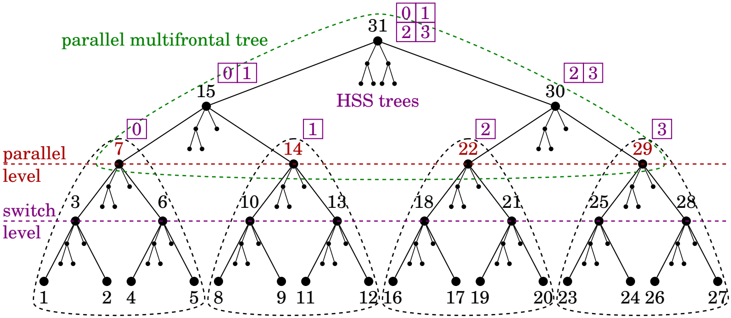
- Hopper, Cray XE6 at NERSC.
- Hsolver (for Helmholtz), geometric HSS-embedded multifrontal solver
- Helmholtz equations with PML boundary

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

  Δ: Laplacian
  ω: angular frequency
  $v(x)$: seismic velocity field
  $u(x, \omega)$: time-harmonic wavefield equation
- Finite Difference discretized system: complex, pattern-symmetric, non-Hermitian. Indefinite, ill-conditioned
- single precision

MF + HSS: two types of tree-based parallelism

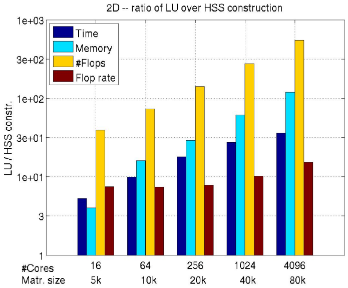
- **Outer tree**: separator tree for multifrontal factorization
- **Inner tree**: HSS tree at each internal separator node



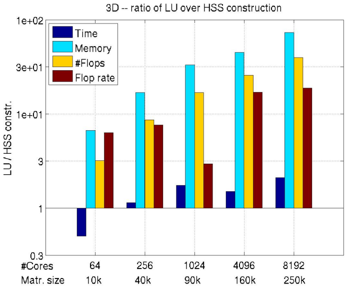
HSS paerformance

- HSS constrcution 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

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	
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	
	HSS rank	258	503	1013	2015
	$\|\|x - x_{MF}\|\|/\|\|x_{MF}\|\|$	1.5×10^{-5}	2.2×10^{-5}	3.1×10^{-5}	3.5×10^{-6}
	$\max_i \frac{\|Ax-b\|_i}{(\|A\|\|x\|+\|b\|)_i}$	7.1×10^{-6}	1.0×10^{-5}	2.0×10^{-6}	3.5×10^{-6}

Results - 3D problems

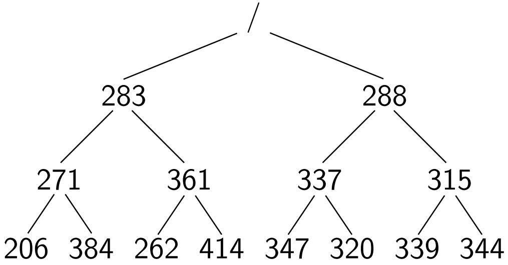
3D Helmholtz problems on cubic grids (mesh size k, $N = k^3$), 5 Hz.

	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
	$\|\|x - x_{\mathrm{MF}}\|\| / \|\|x_{\mathrm{MF}}\|\|$	6.2×10^{-6}	9.4×10^{-7}	1.1×10^{-6}	1.7×10^{-6}
	$\max_i \frac{\|Ax-b\|_i}{(\|A\|\|x\|+\|b\|)_i}$	1.5×10^{-7}	5.7×10^{-7}	9.7×10^{-7}	3.7×10^{-6}

Performance analysis – rank imbalance

- Rank revealing *QR* is the dominant operation (2/3 of the total time).
- Observed some load imbalance (factors up to 2.5) due to some imbalance in the ranks found in frontal matrices (run time of RRQR is proportional to the rank).

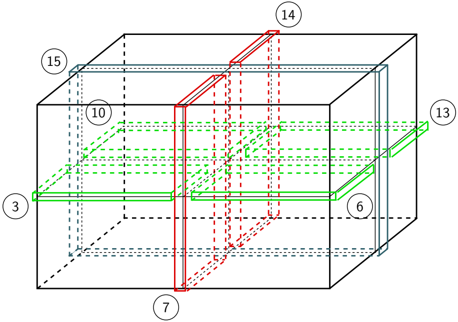
Example: ranks in the row compression of the root of a 100^3 problem

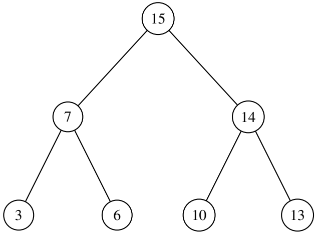


HSS tree; 8 leaves/8 processes.

Ordering of separators/fully-summed variables

In a regular 3D mesh with nested dissection, each frontal matrix corresponds to a plane:



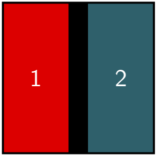


Top of the elimination tree.

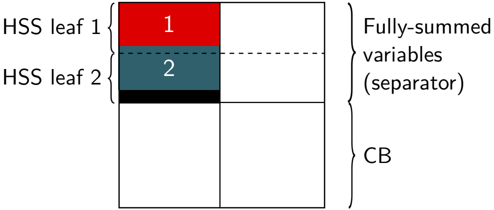
Within a frontal matrix/separator, the ordering of the variables is crucial to get low rank blocks (cf. admissibility condition for PDEs [Börm, Grasedyck, Hackbush]).

Ordering of separators/fully-summed variables – 2

Vertex-based approach:

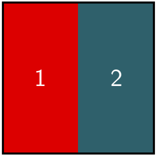


2D separator from a 3D domain.

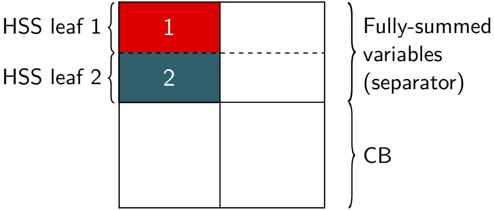


Frontal matrix.

Edge-based approach:



2D separator from a 3D domain.



Frontal matrix.

- An edge-based Nested Dissection / Morton ordering better preserves geometry and should provide better balance in ranks.
- On some medium-sized problems we observed $\sim 10\%$ gain in run time. Work in progress.

Summary, future

- Parallel geometric HSS-embedded MF solver for Helmholtz equations is faster than a regular MF solver. Gains increase with problem size.
- Using iterative refinement, it delivers accurate solutions.

- Explore ways to reduce further the memory footprint, in particular the stack of contribution blocks. Structured Schur complement? Randomized sampling?
- Move towards a parallel algebraic solver
- Analyze communication bound
- Black-box preconditioner? (compare to ILU, etc.)
- Compare to sparse solvers using other low-rank forms [Saad et al., Weisbecker et al., Ying et al.]
- Resilience at extreme scale