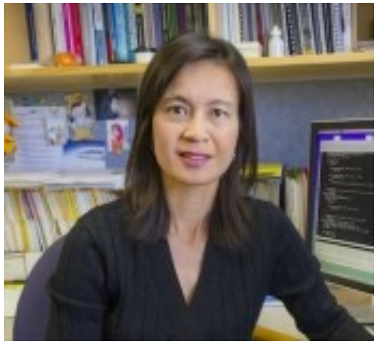
Scalable and Robust Hierarchical Matrix Factorizations via Randomization

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

Conference on Fast Direct Solvers (Online)
October 23-24, 2021
Purdue, Center for Computational & Applied Mathematics

People

Sherry Li



Lisa Claus

Pieter Ghysels

Yang Liu

Past members:

- Gustavo Chavez, Machinify
- Chris Gorman, UC Santa Barbara
- Liza Rebrova, Princeton
- Francois-Henry Rouet, LSTC

Acknowledgement

- This research is supported in part by the Scientific Discovery through Advanced Computing (SciDAC) program through the FASTMath Institute under Contract No. DE-AC02-05CH11231.

- This research is supported in part by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

Fast direct solvers, preconditioners using ideas from structured matrices

Two types of structured matrices (for the purpose of this talk …)

- "Dense" → data-sparse

  e.g., low-rank, butterfly representation

- "Sparse" → structurally-sparse

  becomes sparser when combining with data-sparse

Software

- ButterflyPACK https://github.com/liuyangzhuan/ButterflyPACK
  (**Yang Liu** et al.)
  - Distributed-memory, OpenMP, Fortran2008
  - Support H, HODLR with LR and Butterfly
  - Dense
- STRUMPACK https://portal.nersc.gov/project/sparse/strumpack/
  (**Pieter Ghysels** et al.)
  - Distributed-memory, OpenMP, C++
  - Support HSS, BLR with LR, interface with most ButterflyPACK functionalities
  - Dense and sparse (multifrontal + data-sparse fronts)

	H (LR/BF)	HODLR (LR/BF)	HSS	BLR	H^2
ButterflyPACK	☺	☺			
STRUMPACK		☺	☺	☺	*WIP*

Outline

- Adaptive random sketching, error estimate, stopping criteria

  C. Gorman, G. Chavez, P. Ghysels, T. Mary, F.-H. Rouet, X.S. Li, “Robust and Accurate Stopping Criteria for Adaptive Randomized Sampling in Matrix-free Hierarchically Semiseparable Construction", SIAM J. Sci. Comput., 2019

- Butterfly compression for high frequency wave equations

  Y. Liu, X. Xing, H. Guo, E. Michielssen, P. Ghysels, X.S. Li, “Butterfly factorization via randomized matrix-vector multiplications’’, SIAM J. Sci. Comput., 2021

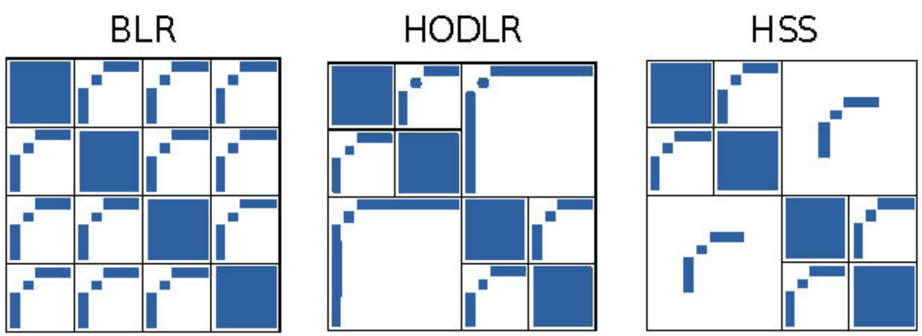
- Sparse direct solver enhanced with data-sparse

  Y. Liu, P. Ghysels, L. Claus, X.S. Li, “Sparse Approximate Multifrontal Factorization with Butterfly Compression for High Frequency Wave Equations", SIAM J. Sci. Comput., 2021

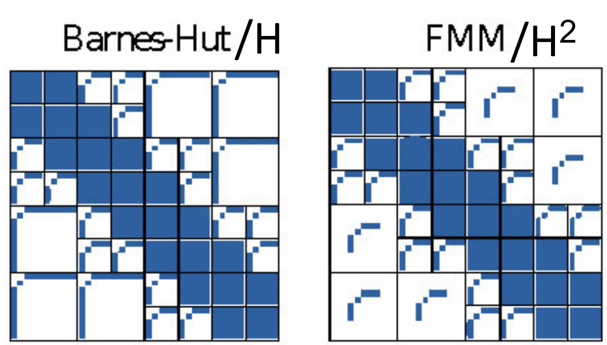
Classes of LR structured matrices

Lower complexity usually requires multilevel / hierarchical approach

	Strong admissibility	Weak admissibility
Independent bases	H matrix	(Hierarchically off-diagonal low-rank) HODLR matrix
Nested bases	H^2 matrix Inverse FMM	HSS matrix Recursive Skeletonization HIF



Weak admissibility



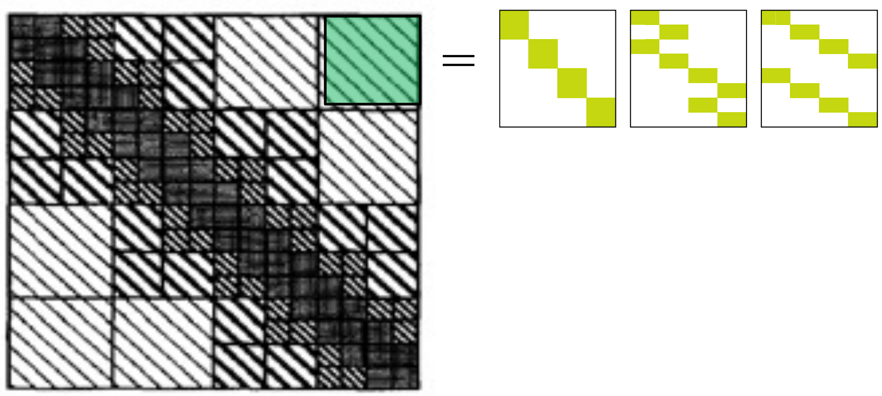
Strong admissibility

Butterfly structured matrices

- Butterfly factorization: generalized FFT

  Michielssen/Boag (1996), Li, Liu, Poulson, Yang, Ying, …

	Strong admissibility	Weak admissibility
Independent bases	H-BF	HOD-BF
Nested bases	Directional H^2 ?	HSS-BF



Stages of operations

- **Data clustering, matrix reordering**
  - minimize off-diag ranks

- **Compression** – usually dominating cost
  complexity depends on two situations:
  1. only black-box matvec K*g and K'*g are available
  2. only black-box entry evaluation K(i,j) is available

  Goal: $O(N \log^{\alpha} N)$

- **Matrix operations with compressed format**
  - Matrix-vector multiplication
  - Factorization / solve, inversion, …

Sweeping through **"trees"** upward / downward:

- HSS tree, butterfly tree
- Principal tools for parallelization

LR compression mechanisms

- SVD
- RRQR

- Randomized projection (sampling, sketching) : $O(n^2r)$

- ACA, block ACA → hierarchical blocked ACA : $O(nr^2)$
  (Liu, Sid-Lakhdar, Rebrova, Ghysels, Li, 2020)
- Interpolative Decomposition (ID) (skeletonization)

$B \approx B(:,J)\,X$, $B(:,J)$ has k columns, X is called interpolation matrix

- Nearest neighbors: relies on geometry

Compression via randomized sketching (RS)

… more flexible and faster than traditional rank-revealing QR

Approximate range of A:

1. Pick random matrix $\Omega_{nx(k+p)}$, k target rank, p small, e.g. 10
2. Sample matrix $S = A\,\Omega$ (tall-skinny)
3. Compute Q = ON-basis(S) via rank-revealing QR

   Can show: $\|(I - QQ^*)A\|$ is small with high probability

$$\mathrm{E}[\| A - QQ^* A \|] = \left(1 + \frac{4\sqrt{k+p}}{p-1}\sqrt{\min\{m,n\}}\right)\sigma_{k+1}$$

(Halko, Martinsson, Tropp, SIAM Rev. 2011)

Benefits:

- Matrix-free, only need matvec
- When embedded in sparse frontal solver, simplifies “extend-add”

Compression via randomized sketching (RS)

… more flexible and faster than traditional rank-revealing QR

Approximate range of A:

1. Pick random matrix $\Omega_{nx(k+p)}$, k target rank, p small, e.g. 10
2. Sample matrix $S = A\ \Omega$ (tall-skinny)
3. Compute Q = ON-basis(S) via rank-revealing QR

   Can show: $\|(I - QQ^*)A\|$ is small with high probability

$$\mathrm{E}[\| A - QQ^* A \|] = \left(1 + \frac{4\sqrt{k+p}}{p-1}\sqrt{\min\{m,n\}}\right)\sigma_{k+1}$$

(Halko, Martinsson, Tropp, SIAM Rev. 2011)

Practical issue: do not know target rank!

Earlier adaptive strategy

- Sample based posterior error estimation
  (Halko, Martinsson, Tropp, SIAM Rev. 2011)

LEMMA 4.1. *Let* $\boldsymbol{B}$ *be a real* $m \times n$ *matrix. Fix a positive integer* r *and a real number* $\alpha > 1$. *Draw an independent family* $\{\boldsymbol{\omega}^{(i)} : i = 1, 2, \ldots, r\}$ *of standard Gaussian vectors. Then*

$$\|\boldsymbol{B}\| \le \alpha \sqrt{\frac{2}{\pi}} \max_{i=1,\ldots,r} \|\boldsymbol{B}\boldsymbol{\omega}^{(i)}\|$$

except with probability α^{-r}.

- Drawback:
  - Only provide absolute bound, not relative
    (relative is particularly desirable when B are submatrices in hierarchical matrices)

New: Incremental RS for robustness and performance

Increase sample size d, build Q incrementally (block Gram-Schmidt)
S = [S_1 , S_2 , S_3 , …]

- Projection:

$$\hat{S} = (I - QQ^*)S_{i+1} \quad \text{(one step block Gram−Schmidt)}$$

$$\hat{S} = (I - QQ^*)^2 S_{i+1} \quad \text{(in practice, “twice” to ensure orthogonality)}$$

- $\hat{S}$ is used to expand Q, only need internal orthogonalizing $\hat{S}$

$$[\bar{Q}, \mathrm{R}] = \mathrm{QR}(\hat{S}), \quad \text{augment } Q \leftarrow [Q \ \ \overline{Q}]$$

Need good error estimation to bound error for A: ||(I – QQ$^{*)}$A||

- only have S (only matvec available)
- want relative error estimate

Stochastic norm estimation

Let $A \in \Re^{mxn}$, and $x \in \Re^{n}$ with $x_i \sim N(0,1)$. Consider SVD:

$$A = U\Sigma V^* = [U_1 \;\; U_2]\begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix}$$

Define $\zeta = V^* x$, ζ is also a Gaussian random vector.

$$\|Ax\|_2^2 = \|\Sigma\zeta\|_2^2 = \zeta_1^2\sigma_1^2 + \cdots + \zeta_r^2\sigma_r^2$$

here, $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > 0$ are positive singular values. Then:

$$\mathrm{E}\left[\|Ax\|_2^2\right] = \sigma_1^2 + \cdots + \sigma_r^2 = \|A\|_F^2$$

For d sample vectors: $\mathrm{E}\left[\|S\|_F^2\right] = d\,\|A\|_F^2$

Probabilistic tail bounds: Gorman et al., SISC 2019

Adaptive sampling: stopping criteria

Let $[S_1 \ \ S_2] = A\,[R_1 \ R_2\,]$, $Q = QR(S_1)$, block size d

Absolute stopping criterion:

$$\|(I \ - QQ^*)A\|_F \approx \frac{1}{\sqrt{d}} \|(I \ - QQ^*)S_2\|_F \leq \varepsilon_a$$

Relative stopping criterion:

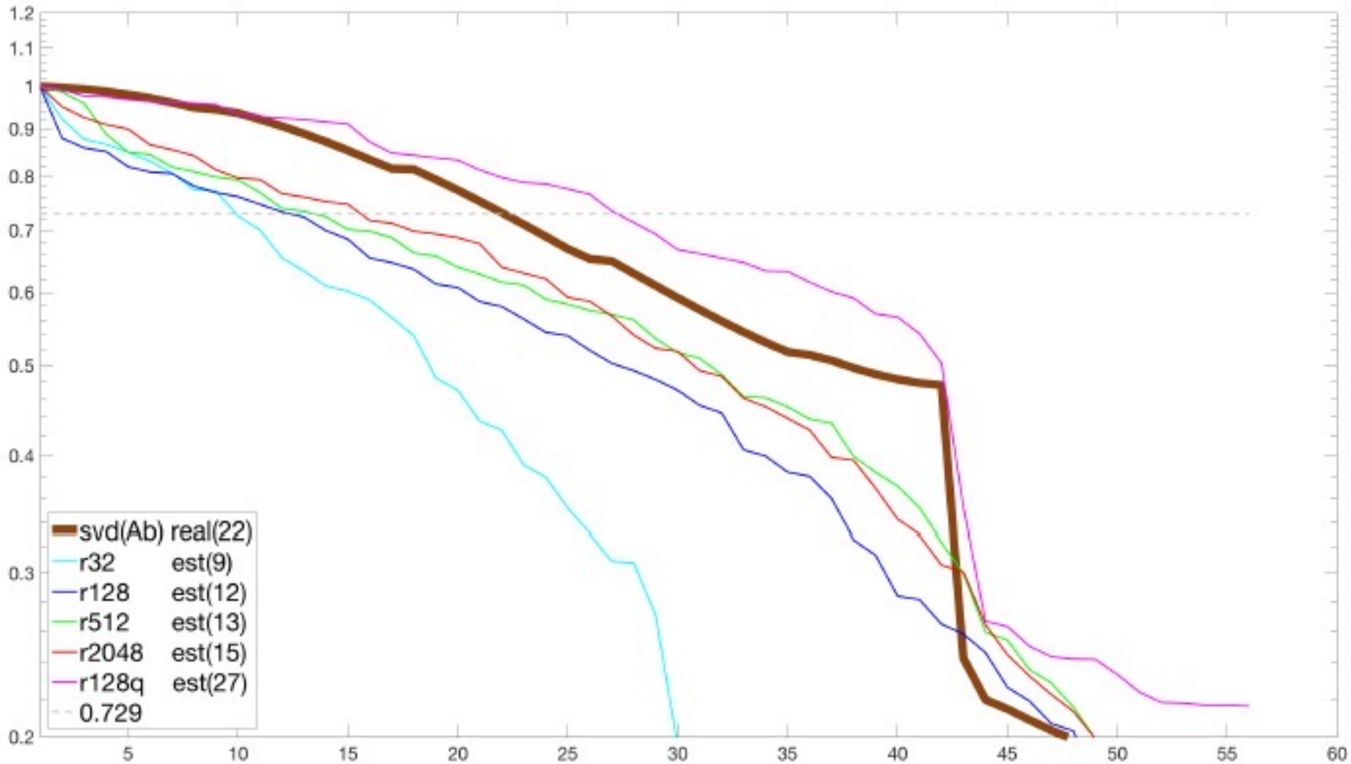
$$\frac{\|(I - QQ^*)A\|_F}{\|A\|_F} \approx \frac{\|(I - QQ^*)S_2\|}{\|S_2\|} \leq \varepsilon_r$$

- Cost: one reduction to compute norms of the sample vectors
- Give enough samples (robustness), but not too many (performance)

Example from HSS compression

- One internal rectangular Hankel block decay (56x3544) (from the 3600x3600 dense Poisson front)

- Older strategy stops at rank 15. New strategy stops at 27 (real 22)



Outline

- Adaptive randomized sampling, error estimate, stopping criteria

  C. Gorman, G. Chavez, P. Ghysels, T. Mary, F.-H. Rouet, X.S. Li, "Robust and Accurate Stopping Criteria for Adaptive Randomized Sampling in Matrix-free Hierarchically Semiseparable Construction", SIAM J. Sci. Comput., 2019

- Butterfly compression for high frequency wave equations

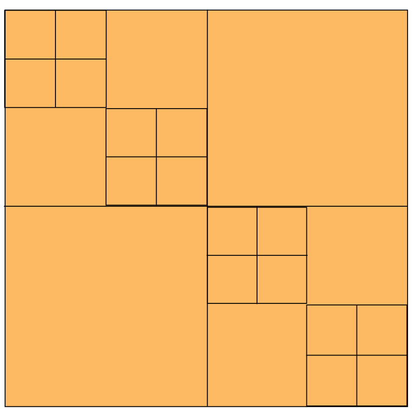
  Y. Liu, X. Xing, H. Guo, E. Michielssen, P. Ghysels, X.S. Li, "Butterfly factorization via randomized matrix-vector multiplications'', SIAM J. Sci. Comput., 2021

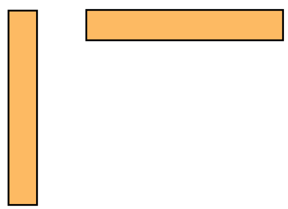
- Sparse direct solver enhanced with data-sparse

  Y. Liu, P. Ghysels, L. Claus, X.S. Li, "Sparse Approximate Multifrontal Factorization with Butterfly Compression for High Frequency Wave Equations", SIAM J. Sci. Comput., 2021

What about wave equations ?

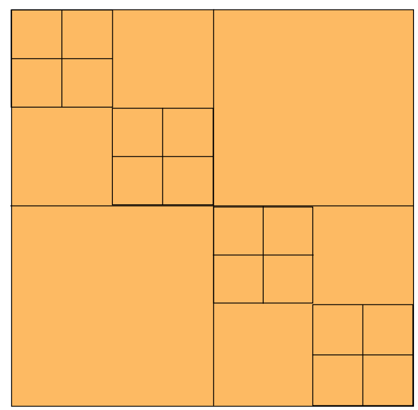
- Low/medium frequency: inversion cost $O(N \log^{\alpha} N)$
- High frequency: LR is not effective, inversion cost $O(N^3)$
- Remedies:
  - High-frequency FMM (Rokhlin 1993)
  - L-Sweeps (Taus), approximate-inverse (Ying), Xi, Vuik, etc.
  - Butterfly factorization (this talk)

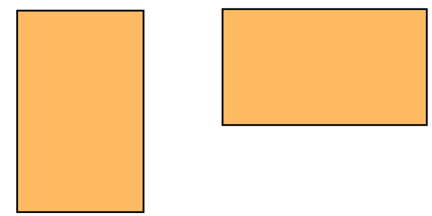




Z_{12} rank const
Memory $O(N)$

Z: low-freq

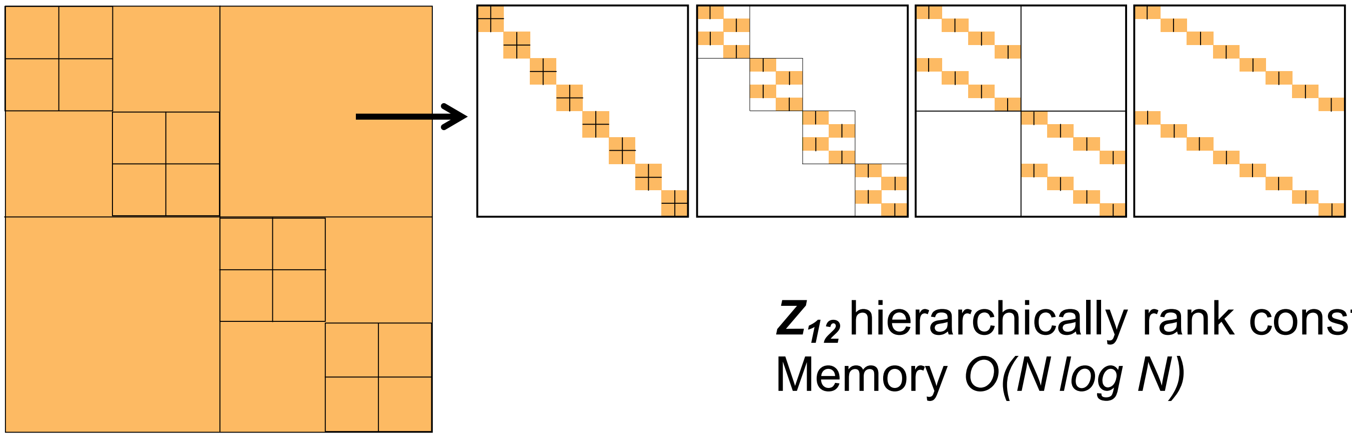




Z_{12} rank $O(N)$
Memory $O(N^2)$

Z: high-freq

Butterfly factorization



$\boldsymbol{Z}$: high-freq

$\boldsymbol{Z_{12}}$ hierarchically rank const
Memory $O(N \log N)$

- Butterfly is a FFT-inspired compression tool for highly oscillatory kernels
- Applications
  - Special function transforms (Radon, Fourier, spherical harmonic)
  - Fast direct and iterative integral (IE) equation solvers for high-frequency Helmholtz
  - Fast direct differential equation solvers for high-frequency Helmholtz
  - Scalable machine learning algorithms

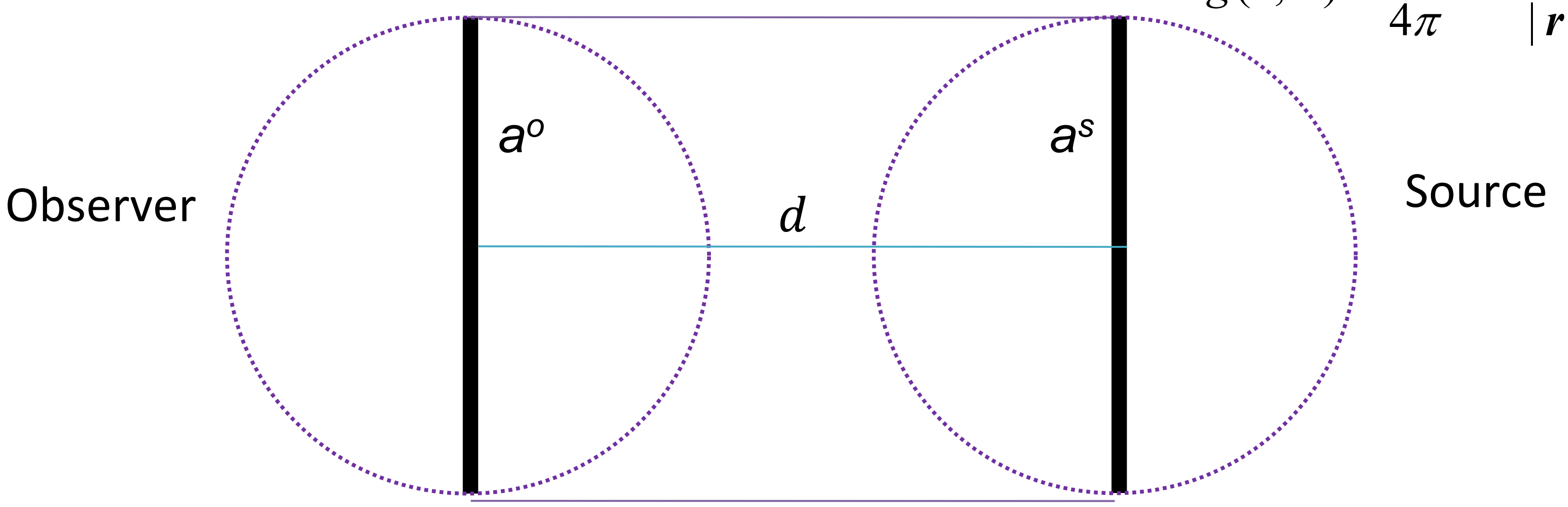
Physical interpretation of Butterfly

Degree of freedom (interaction rank): 2D $\sim \dfrac{k\,a^s a^o}{d}$, $\quad 3D \sim \left(\dfrac{k\,a^s a^o}{d}\right)^2$

a^o : diameter of sub-observer. a^s : diameter of sub-source.

d : distance between source observer centers. k : wave number.

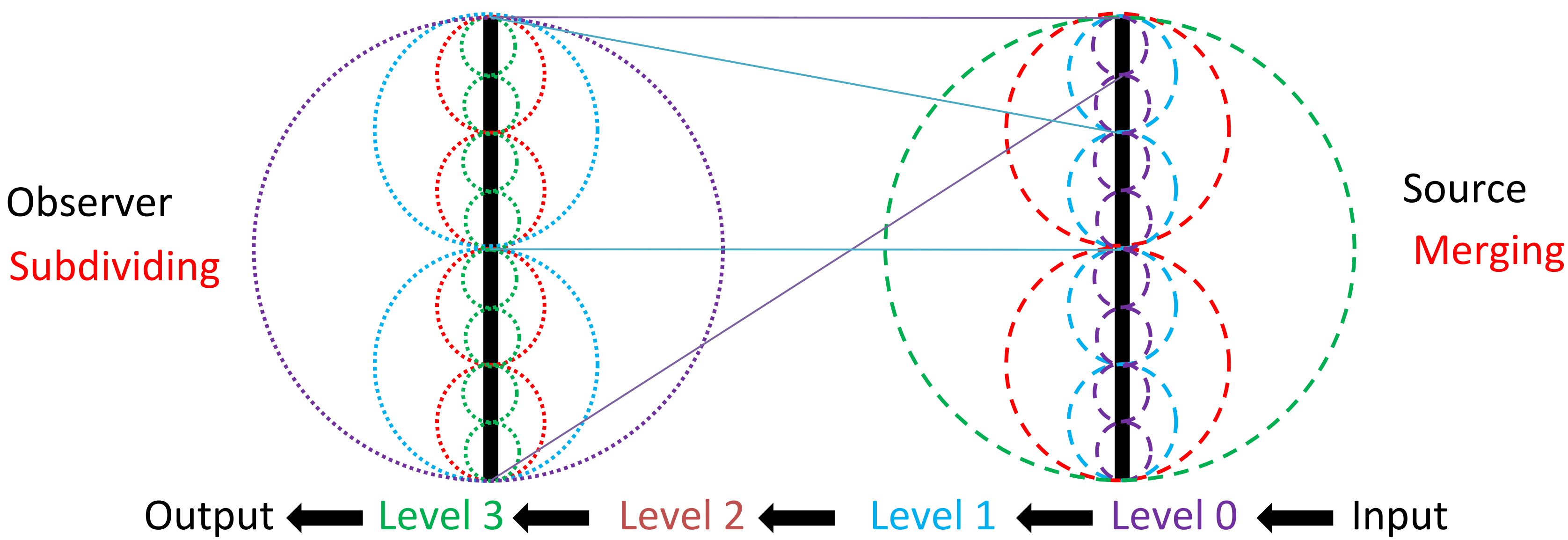
$$g(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{4\pi}\frac{\exp(ik\,|\boldsymbol{r}-\boldsymbol{r}'|)}{|\boldsymbol{r}-\boldsymbol{r}'|}$$

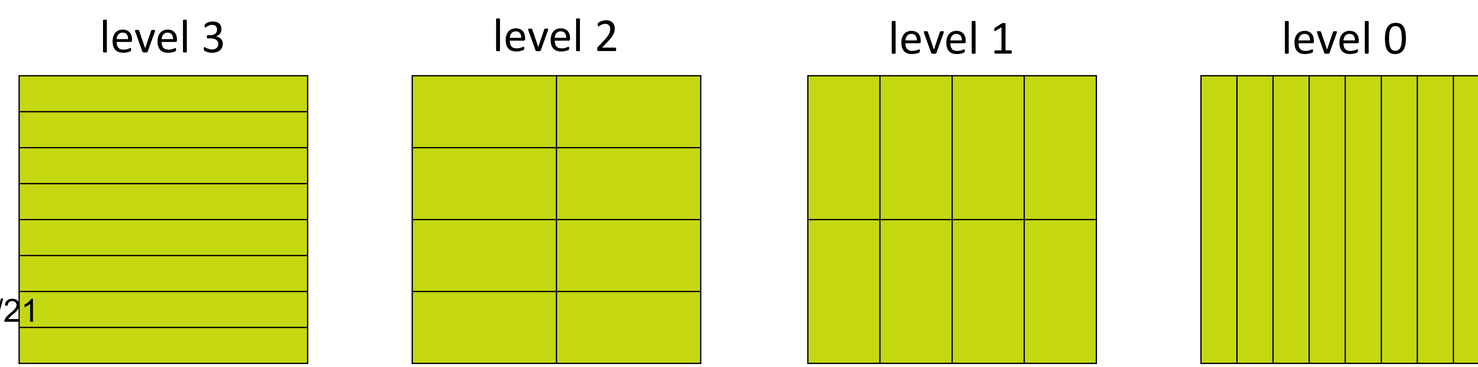


Physical interpretation of Butterfly

Degree of freedom (interaction rank): $2\text{D} \sim \dfrac{k\, a^s a^o}{d}, \quad 3D \sim \left(\dfrac{k\, a^s a^o}{d}\right)^2$

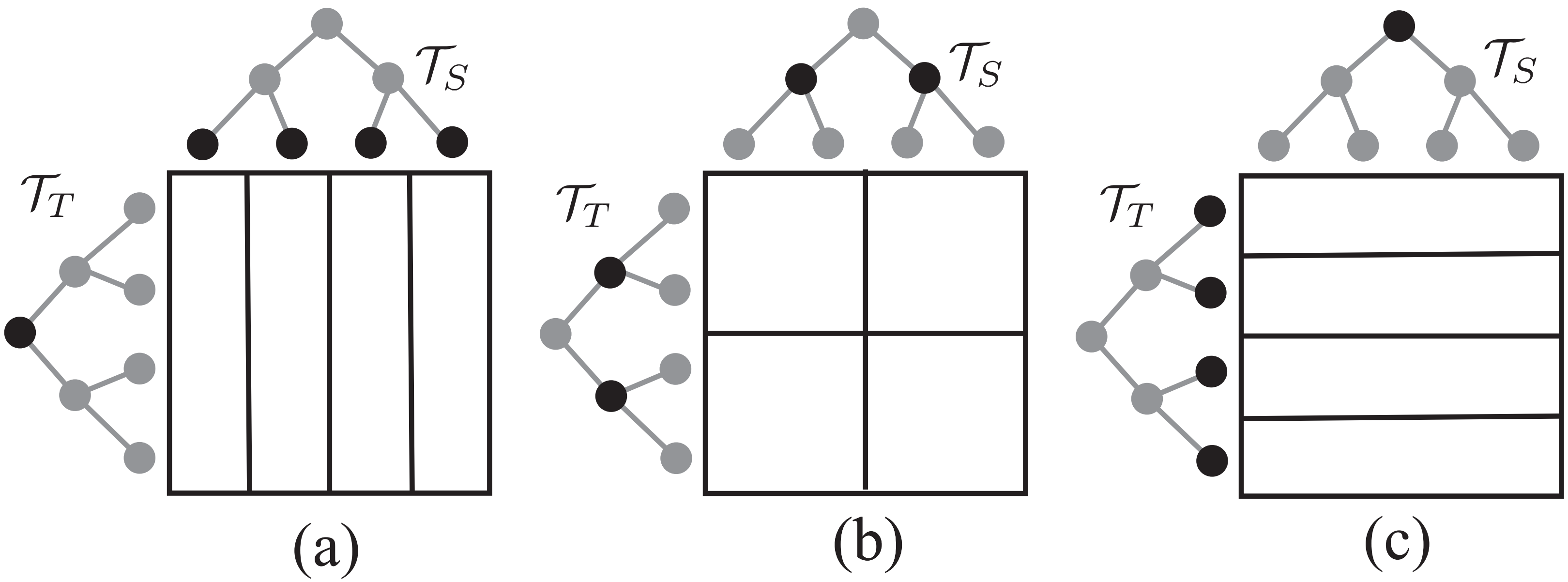
The ranks of all submatrices are r = constant (butterfly rank)





Hierarchical partitioning: Butterfly trees

Complementary low-rank property



Butterfly construction

Depending on how operator A is available

- Each entry can be evaluated in O(1) time

- Fast matvec available, e.g. in O(n logn) time

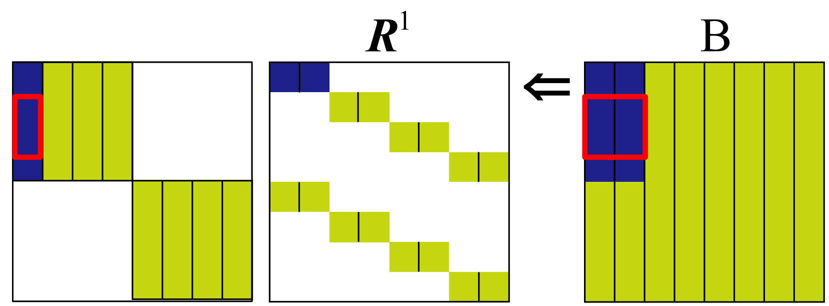
  Examples:
  - Compression of frontal matrices in sparse multifrontal solver
  - Conversion to butterfly from other FMM-like formats
  - Recompressing the composition of Fourier integrals
  - …

Butterfly Construction: entry-evaluation based

Let ***B*** has butterfly rank *r*, its *L*-level butterfly factorization is $\boldsymbol{B} = \boldsymbol{R}^L \ldots \boldsymbol{R}^1$

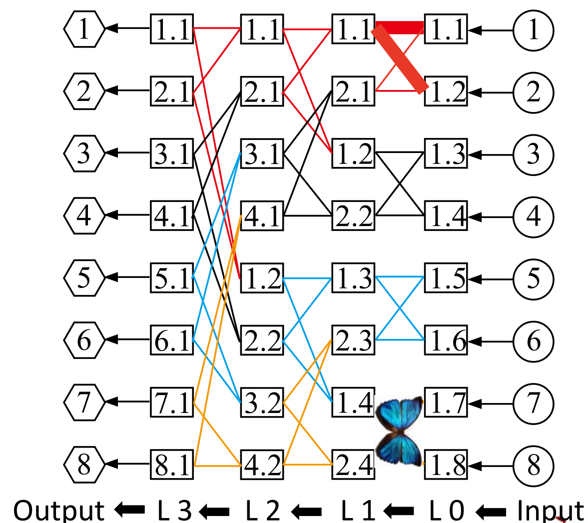
Compute LR compression for all judiciously selected submatrices

$R^{\ell}_{i,j}$ has size at most $r \times r$, Memory $O(n \log n)$, Flops $O(n \log n)$



$$\boldsymbol{R}^l = \mathrm{diag}(\boldsymbol{R}^l_1, \ldots, \boldsymbol{R}^l_{2^{l-1}})$$

$$\boldsymbol{R}^l_i = \begin{pmatrix} \boldsymbol{R}^l_{2i-1,1} \mid \boldsymbol{R}^l_{2i-1,2} & & \\ & \ddots & \\ & & \boldsymbol{R}^l_{2i-1,2^{L-l+1}-1} \mid \boldsymbol{R}^l_{2i-1,2^{L-l+1}} \\ \boldsymbol{R}^l_{2i,1} \mid \boldsymbol{R}^l_{2i,2} & & \\ & \ddots & \\ & & \boldsymbol{R}^l_{2i,2^{L-l+1}-1} \mid \boldsymbol{R}^l_{2i,2^{L-l+1}} \end{pmatrix}$$



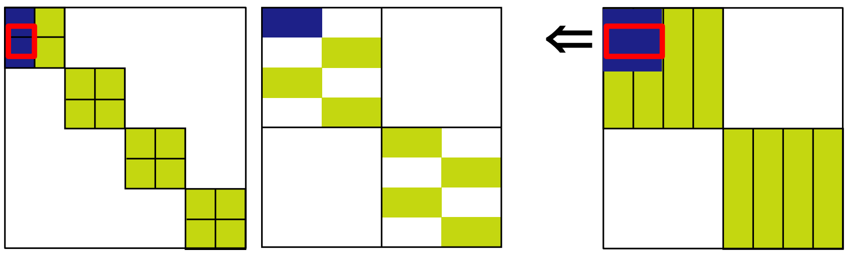
Butterfly Construction: entry-evaluation based

Let $\boldsymbol{B}$ has butterfly rank r, its L-level butterfly factorization is $\boldsymbol{B} = \boldsymbol{R}^L \ldots \boldsymbol{R}^1$

Compute LR compression for all judiciously selected submatrices

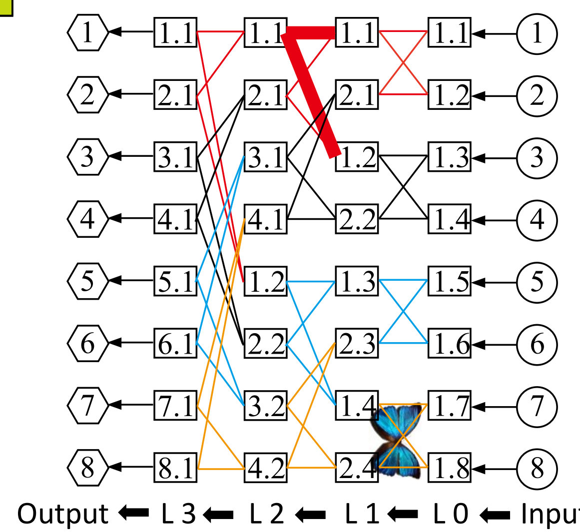
$R^{\ell}_{i,j}$ has size at most $r \times r$, Memory $O(n \log n)$, Flops $O(n \log n)$

$\boldsymbol{R}^3$ $\quad$ $\boldsymbol{R}^2$



$$\boldsymbol{R}^l = \mathrm{diag}(\boldsymbol{R}^l_1, \ldots, \boldsymbol{R}^l_{2^{l-1}})$$

$$\boldsymbol{R}^l_i = \begin{pmatrix} \boldsymbol{R}^l_{2i-1,1} \mid \boldsymbol{R}^l_{2i-1,2} & & \\ & \ddots & \\ & & \boldsymbol{R}^l_{2i-1,2^{L-l+1}-1} \mid \boldsymbol{R}^l_{2i-1,2^{L-l+1}} \\ \boldsymbol{R}^l_{2i,1} \mid \boldsymbol{R}^l_{2i,2} & & \\ & \ddots & \\ & & \boldsymbol{R}^l_{2i,2^{L-l+1}-1} \mid \boldsymbol{R}^l_{2i,2^{L-l+1}} \end{pmatrix}$$



Butterfly Construction: entry-evaluation based

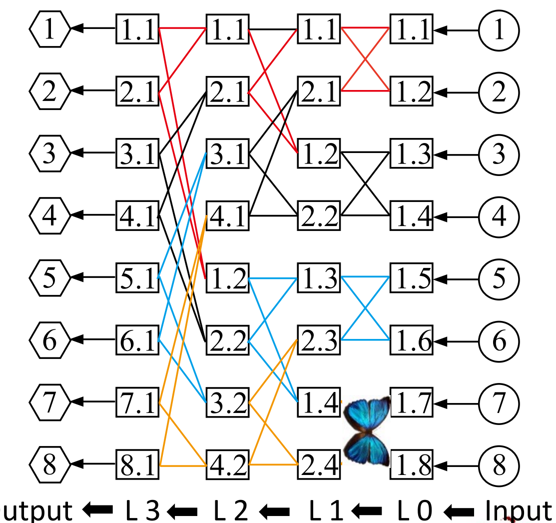
Let ***B*** has butterfly rank *r*, its *L*-level butterfly factorization is ***B* = *R*^{L} … *R*^{1}**

Compute LR compression for all judiciously selected submatrices

$R^{\ell}_{i,j}$ has size at most $r \times r$, Memory $O(n \log n)$, Flops $O(n \log n)$

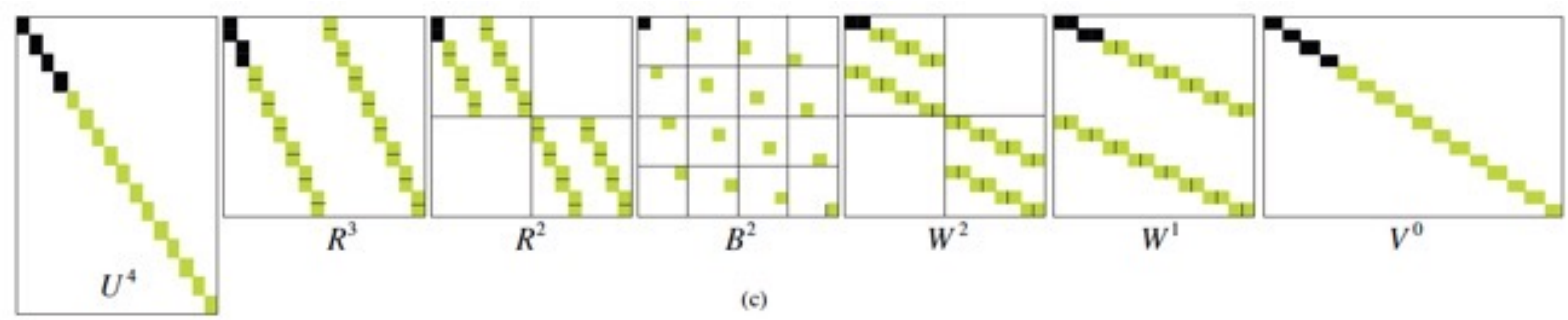
$$\mathrm{R}^l = \mathrm{diag}(\mathrm{R}^l_1, \ldots, \mathrm{R}^l_{2^{l-1}})$$

$$\boldsymbol{R}^l_i = \begin{pmatrix} \boldsymbol{R}^l_{2i-1,1} \mid \boldsymbol{R}^l_{2i-1,2} & & \\ & \ddots & \\ & & \boldsymbol{R}^l_{2i-1,2^{L-l+1}-1} \mid \boldsymbol{R}^l_{2i-1,2^{L-l+1}} \\ \boldsymbol{R}^l_{2i,1} \mid \boldsymbol{R}^l_{2i,2} & & \\ & \ddots & \\ & & \boldsymbol{R}^l_{2i,2^{L-l+1}-1} \mid \boldsymbol{R}^l_{2i,2^{L-l+1}} \end{pmatrix}$$

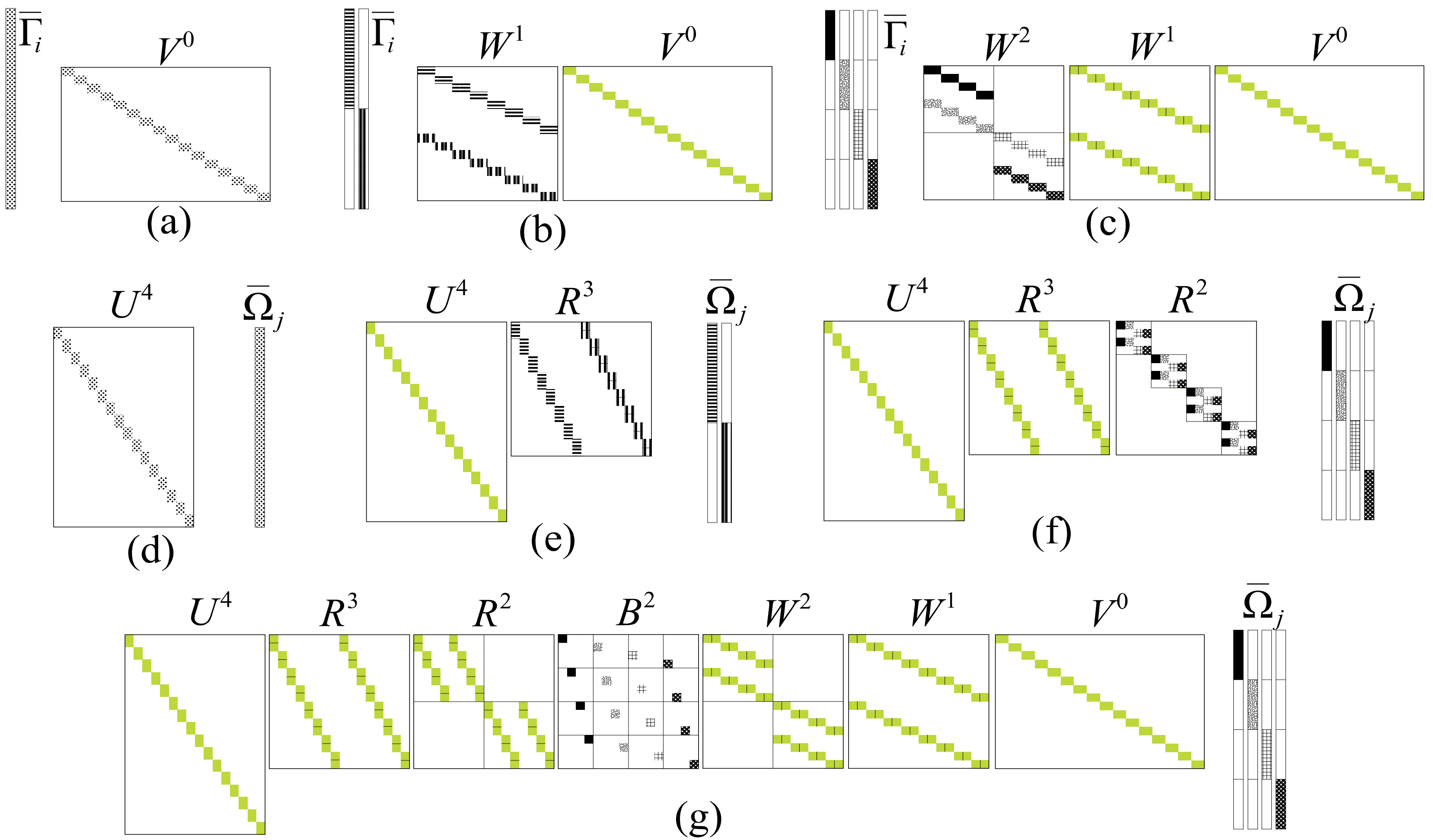


Hybrid form

- Column basis on the left, row basis on the right
- Meet at middle layer $l = L/2$



Butterfly Construction: random projection based

- BF_random_matvec : $O(n^{1.5}\log n)$ time, $O(n\log n)$ memory
- 

- For a $n\times n$ block A, $O(n^{0.5})$ vectors are needed
- Its L-level butterfly B satisfies: $||A-B||_F^2 \leq (L+2)\epsilon^2||A||_F^2$

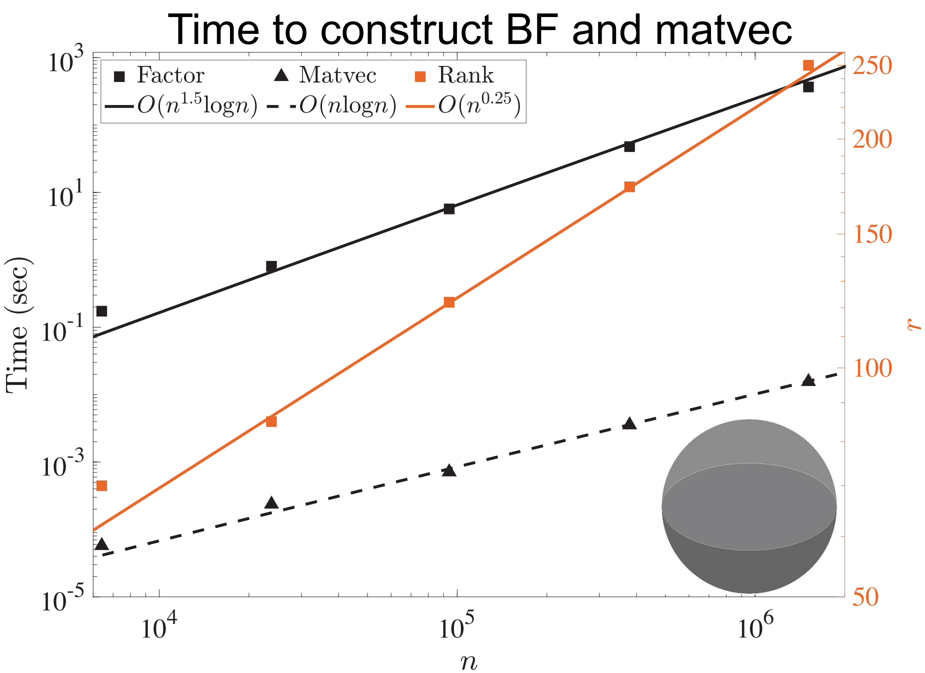
Butterfly Construction: randomized sketching based

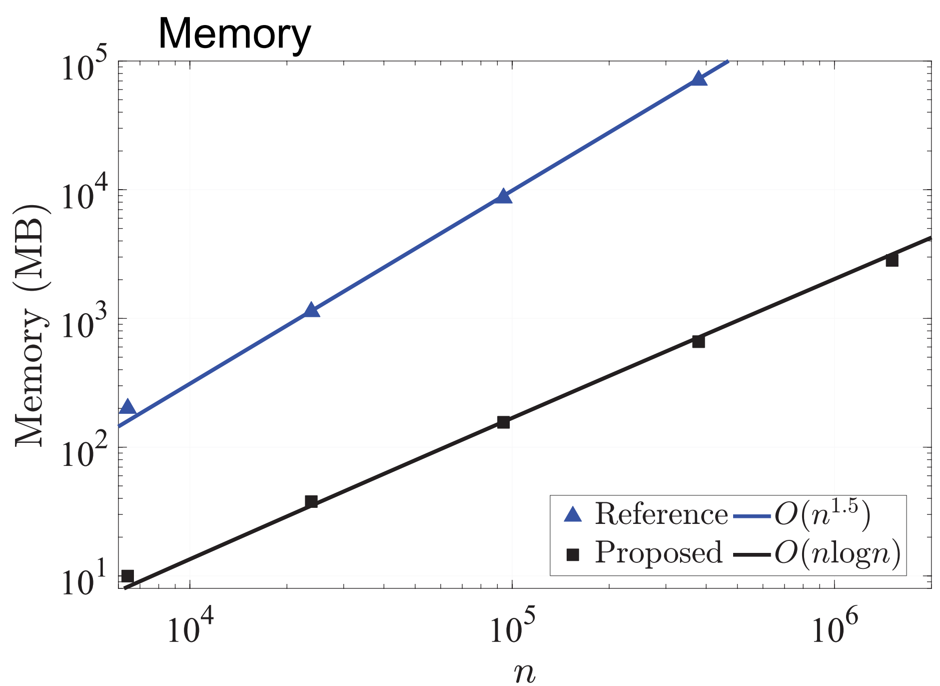
Liu, Xing, Guo, Michielssen, Ghysels, Li, 2021 (with parallelization)

3D Helmholtz kernel: interaction between two semi-sphere surfaces

BF to approximate:

$$A_{i,j} = \frac{\exp(i2\pi\kappa|\rho_i - \rho_j|)}{|\rho_i - \rho_j|}$$



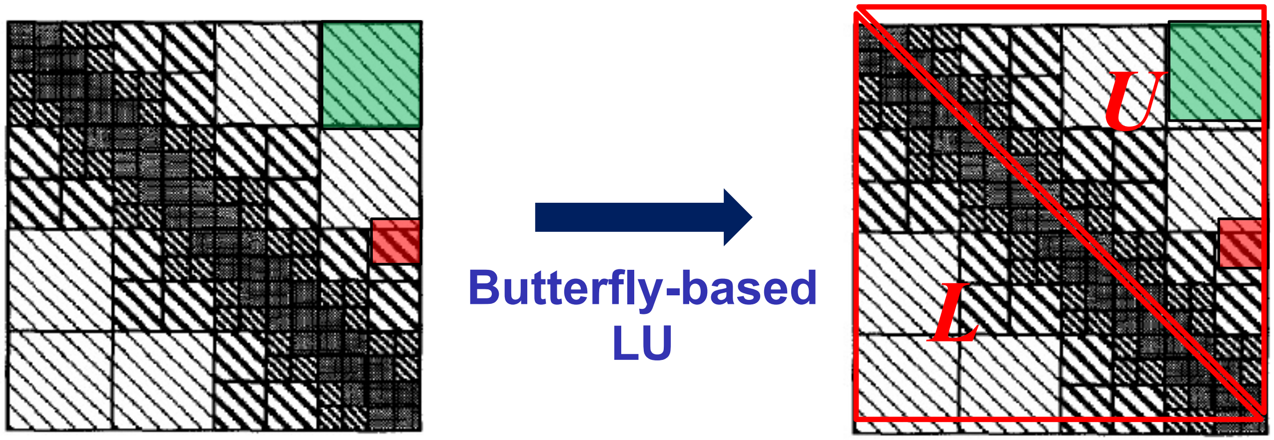


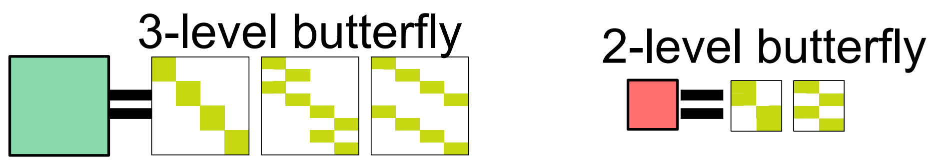
After construction, can do several operations

- Matrix-vector multiplication
  - Michielssen, Boag, 1996
  - Li, Yang, Martin, Ho, Ying, 2015
  - Poulson, Demanet , Maxwell, Ying, 2014 – parallelization

- Factorization / inversion
  - Liu, Guo, Michielssen, 2017
  - Liu, Xing, Guo, Michielssen, Ghysels, Li, 2020

Butterfly Solver: LU with H-BF





$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & U_{22} \end{bmatrix}$$

(1) L_{11} and U_{11}: LU decompose $Z_{11} = L_{11}U_{11}$

(2) U_{12}: Triangular solve $Z_{12} = L_{11}U_{12}$

(3) L_{21}: Triangular solve $Z_{21} = L_{21}U_{11}$

(4) Update Z_{22}: $Z_{22} = Z_{22} - L_{21}U_{12}$

(5) L_{22} and U_{22}: LU decompose: $Z_{22} = L_{22}U_{22}$

Randomized butterfly operations:

Butterfly

$B = B_1 + B_2$

butterfly

$B = B_1 A$ or $B = AB_1$

Butterfly

$B = L^{-1}B_1$ or $B = B_1 U^{-1}$

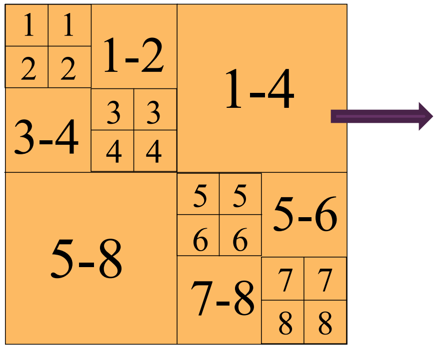
Lower triangular

Upper triangular

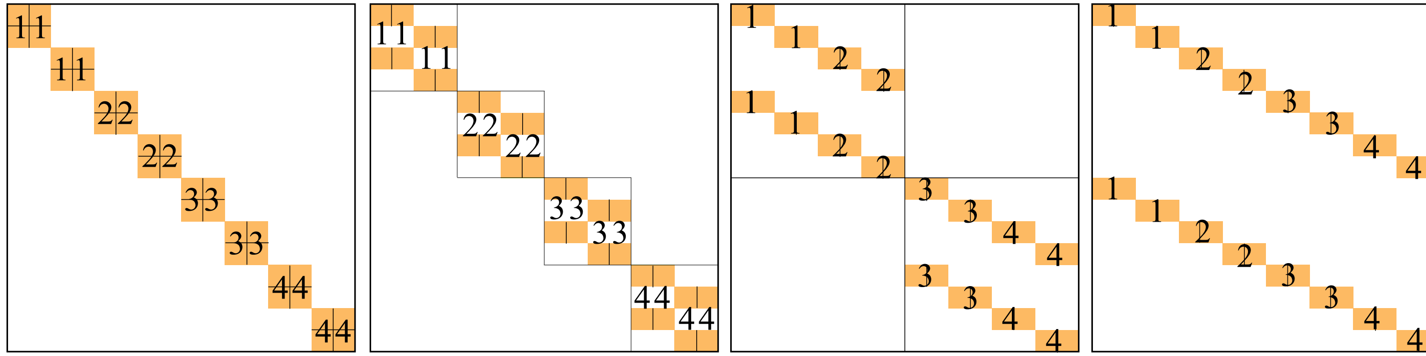
H. Guo, Y. Liu, J. Hu, and E. Michielssen, "A butterfly-based direct integral equation solver using hierarchical LU factorization for analyzing scattering from electrically large conducting objects", IEEE Trans. Antennas Propag., 2017

Distributed-memory Parallelization: BF construction via RS

- HODLR with Butterfly, symmetric bit-reversal ordering for each butterfly



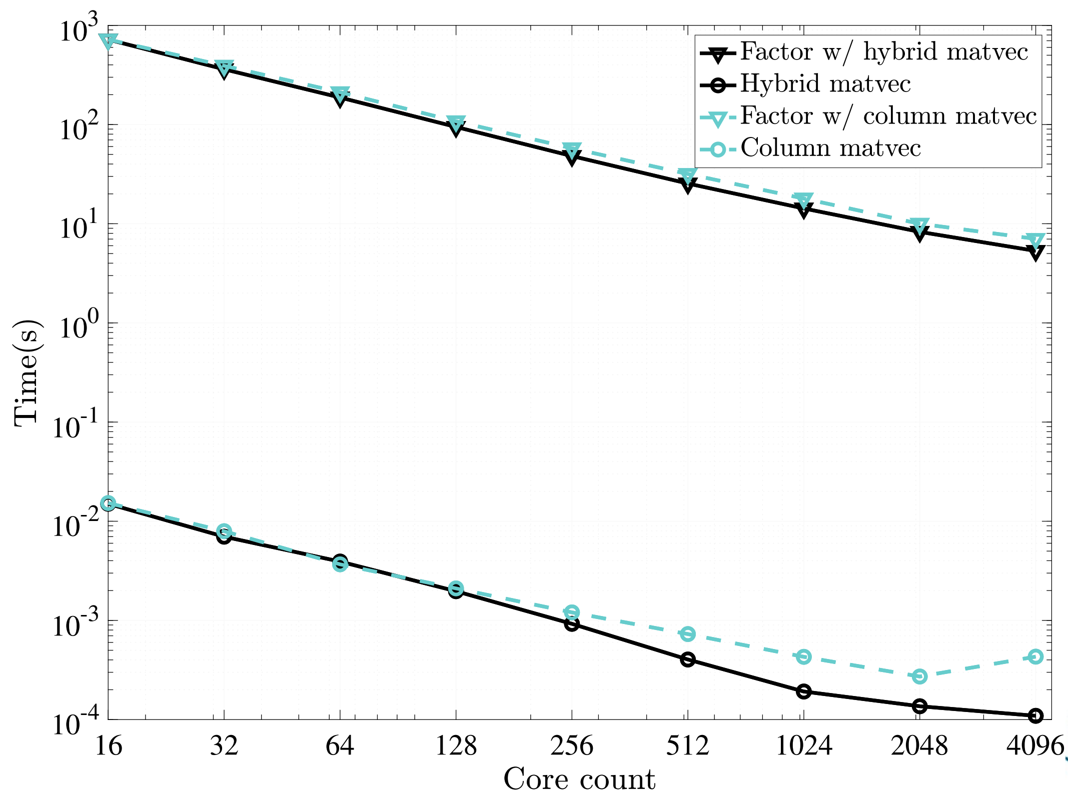
distributed



Distributed butterfly of $\boldsymbol{Z_{12}}$

$n = 2.56 \times 10^6$

core count: 16 - 4096



Outline

- Adaptive randomized sampling, error estimate, stopping criteria

  C. Gorman, G. Chavez, P. Ghysels, T. Mary, F.-H. Rouet, X.S. Li, "Robust and Accurate Stopping Criteria for Adaptive Randomized Sampling in Matrix-free Hierarchically Semiseparable Construction", SIAM J. Sci. Comput., 2019

- Butterfly compression for high frequency wave equations

  Y. Liu, X. Xing, H. Guo, E. Michielssen, P. Ghysels, X.S. Li, "Butterfly factorization via randomized matrix-vector multiplications'', SIAM J. Sci. Comput., 2021

- Sparse direct solver enhanced with data-sparse

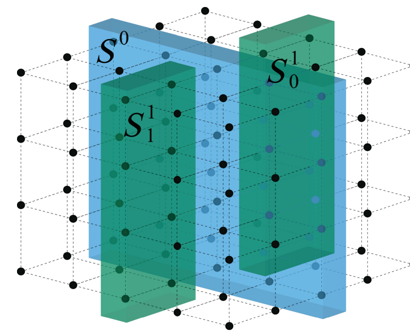
  Y. Liu, P. Ghysels, L. Claus, X.S. Li, "Sparse Approximate Multifrontal Factorization with Butterfly Compression for High Frequency Wave Equations", SIAM J. Sci. Comput., 2021

Sparse direct solver: STRUMPACK
Embedding LR data-sparse in multifrontal sparse factorization

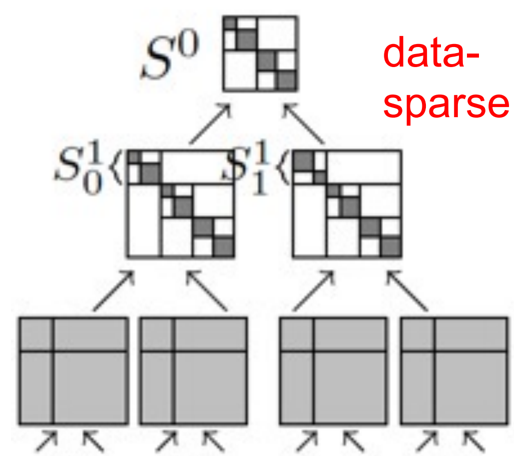
- Globally sparse, locally dense
  - Embed LR data-sparse in sparse multifrontal algorithm
- Baseline is a sparse multifrontal direct solver
- Nested Dissection ordering → separator tree
- In addition to structural sparsity, further apply LR data-sparsity to dense frontal matrices
- **Nested bases + randomized sampling** to achieve linear scaling in sparse case
  - **O(N logN) flops, O(N) memory for 3D elliptic PDEs**

    (as opposed to O(N^2) flops with exact factorization)

Nested dissection ordering

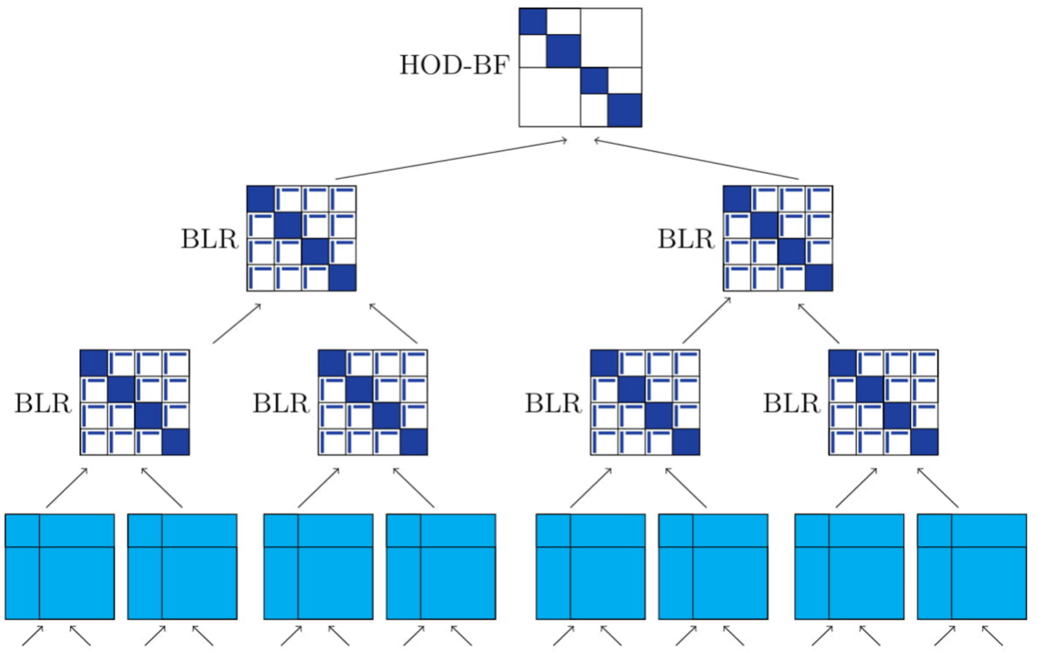


Multifrontal Separator tree



Sparse direct solver: combine multiple data-sparse

- Combining BLR and HOD-
  - Large fronts: HOD-BF
  - Medium fronts: BLR
  - Small fronts: no compr

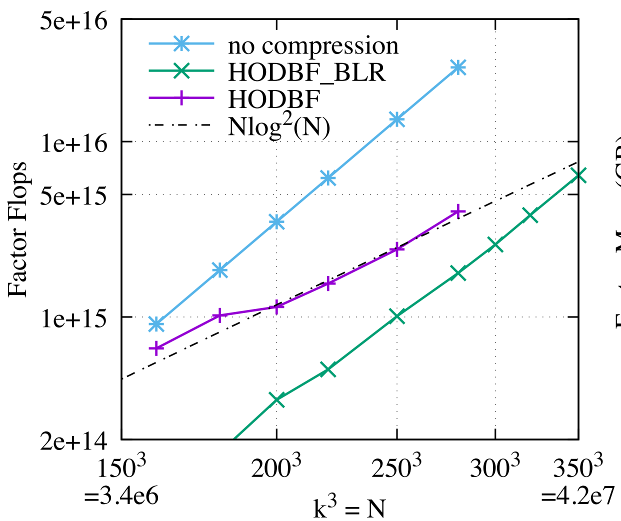


separator size(HOD-BF): 30k
separator size(BLR): 300

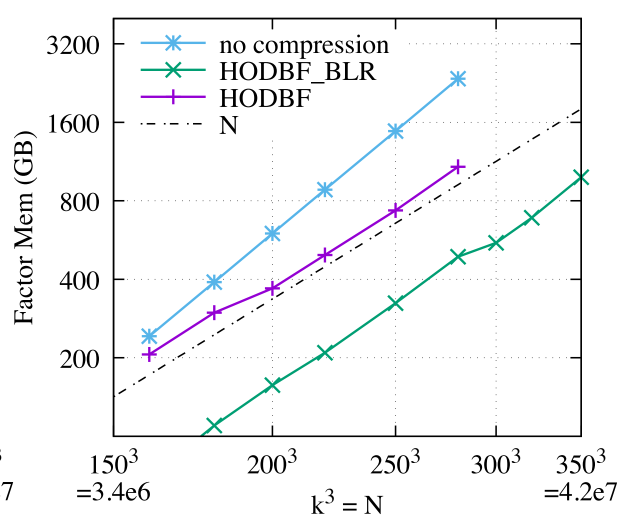
Sparse MF: HOD-BF/BLR: Finite Difference for 3D Helmholtz

$$\left(\sum_i \rho(\mathbf{x}) \frac{\partial}{\partial x_i} \frac{1}{\rho(\mathbf{x})} \frac{\partial}{\partial x_i}\right) p(\mathbf{x}) + \frac{\omega^2}{\kappa^2(\mathbf{x})} p(\mathbf{x}) = -f(\mathbf{x})$$

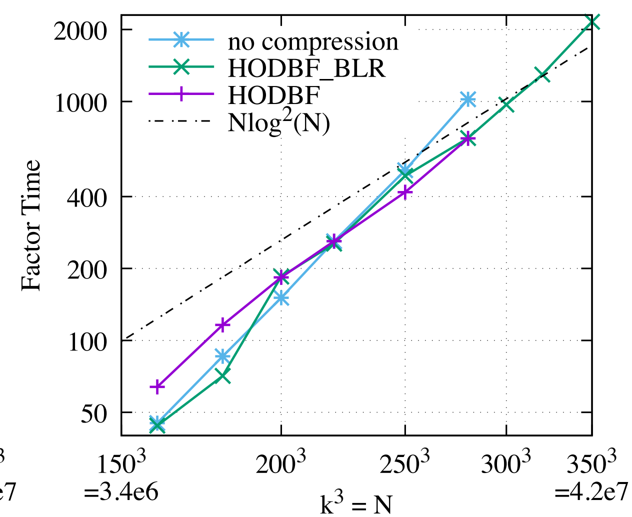
- 27-point finite difference stencil for 3D visco-acoustic propagation
- $v(\mathbf{x}) = 4000\mathrm{m/s}$, $\rho(\mathbf{x}) = 1\mathrm{kg/m}^3$, 15 points per wavelength
- Up to 64 Cori Haswell nodes (2048 cores)
- compression tolerance: 10^{-4} usually ~10 steps of GMRES



factor flops



factor size



Factor time

Sparse MF: HOD-BF vs. HSS

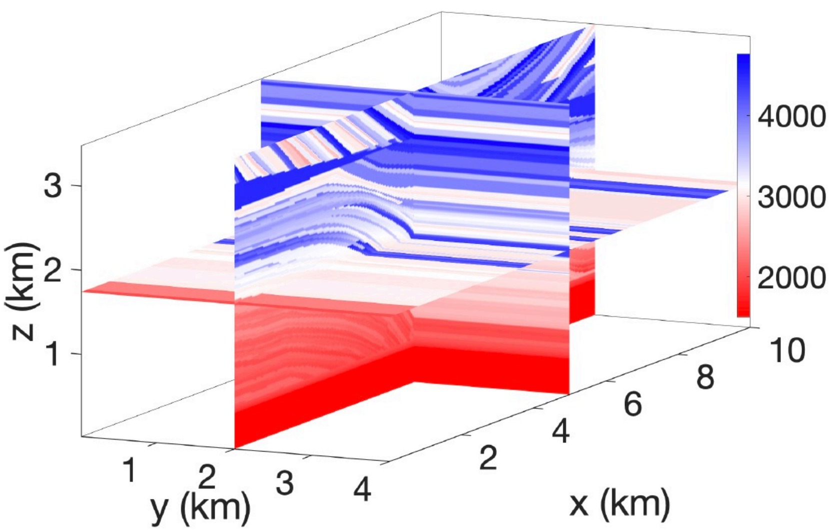
N = 250^3 , constant coefficients
32 Cori nodes (1024 cores)

Solver	Exact	HSS	HOD-BF	HOD-BF	HOD-BF
ε	-	10^{-3}	10^{-3}	10^{-2}	10^{-3}
$n_{\min}$	-	10K	10K	10K	7K
Compressed fronts	0	39	39	39	197
Dense fronts	1,869,841	1,869,802	1,869,802	1,869,802	1,869,644
$\boxed{\text{Factor time (sec)}}$	513	$\boxed{947}$	$\boxed{433}$	354	556
Factor flops (10^{15})	13.4	4.98	2.44	2.24	1.21
Flop Compression (%)	100	37.1	18.2	16.7	9.0
Factor mem (10^3 GB)	1.48	0.84	0.73	0.72	0.47
Mem Compression (%)	100	56.8	49.6	48.8	32.2
$\boxed{\text{Max. rank}}$	-	$\boxed{4698}$	$\boxed{364}$	153	389
Top 2 fronts					
Mem Compression (%)	-	21.9/14.6	7.29/3.54	4.4/1.89	6.3/3.6
Rank	-	4538/4698	154/242	121/213	177/255
Front time (sec)	37/108	172/195	52/88	42/60	46/70.6
GMRES its.	1	18	6	56	23
Solve flops (10^{12})	0.46	8.01	2.84	23.4	7.18
Solve time (sec)	0.72	19.2	3.5	30.1	13.2

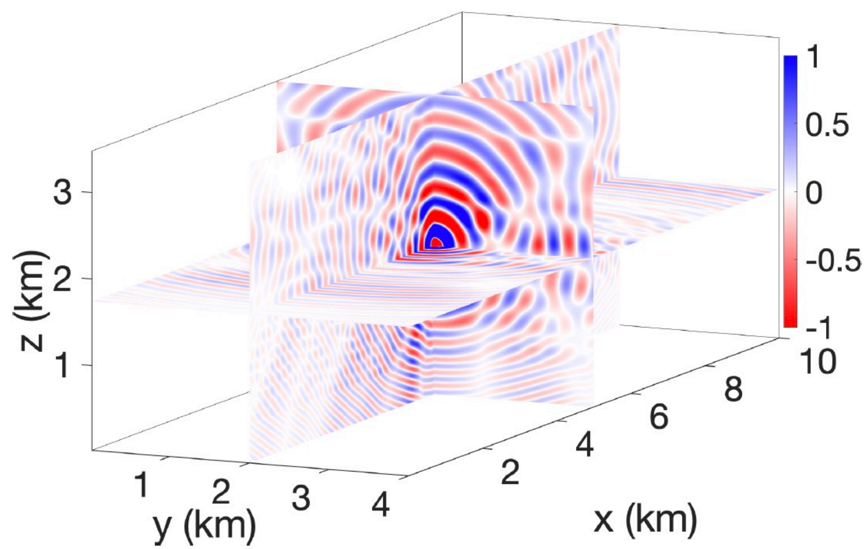
Sparse MF: HOD-BF: 3D Helmholtz, heterogeneous media

$$\left(\sum_i \rho(\mathbf{x})\frac{\partial}{\partial x_i}\frac{1}{\rho(\mathbf{x})}\frac{\partial}{\partial x_i}\right)p(\mathbf{x}) + \frac{\omega^2}{\kappa^2(\mathbf{x})}p(\mathbf{x}) = -f(\mathbf{x})$$

- 27-point finite difference stencil for 3D visco-acoustic propagation
- Marmousi2 P-wave velocity model $190\times216\times516, N = 21{,}176{,}640$, $\omega = 20\pi\mathrm{Hz}$, 7.5 points per wavelength
- 32 Cori Haswell nodes (1024 cores)



velocity model



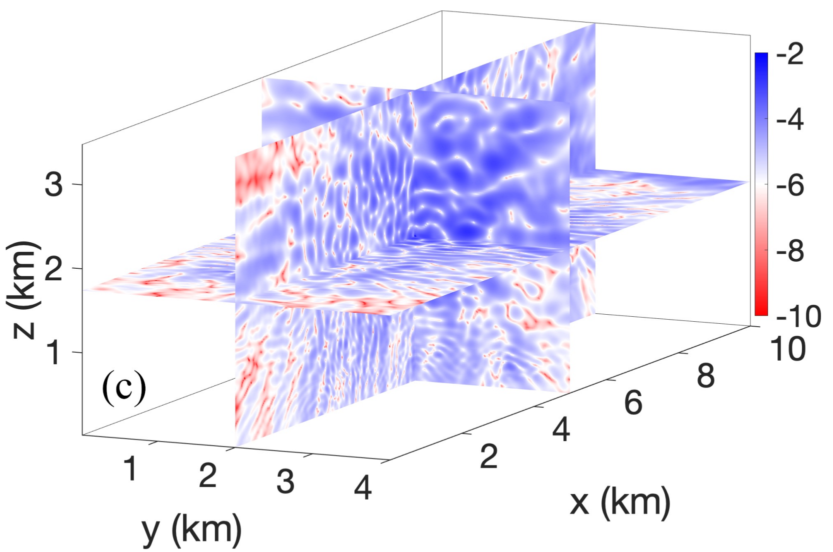
$\mathrm{Re}(p(\mathbf{x}))$

Sparse MF: HOD-BF: 3D Helmholtz, heterogeneous media

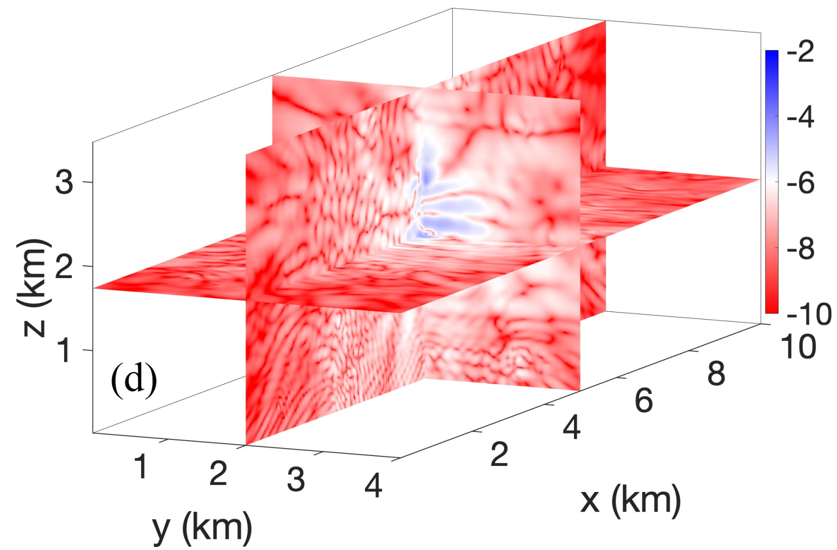
$$\left(\sum_i \rho(\mathbf{x})\frac{\partial}{\partial x_i}\frac{1}{\rho(\mathbf{x})}\frac{\partial}{\partial x_i}\right)p(\mathbf{x}) + \frac{\omega^2}{\kappa^2(\mathbf{x})}p(\mathbf{x}) = -f(\mathbf{x})$$

- Differences from truth in Re($p(\mathbf{x})$) with 2 tolerances

$\varepsilon = 10^{-4}$



$\varepsilon = 10^{-6}$



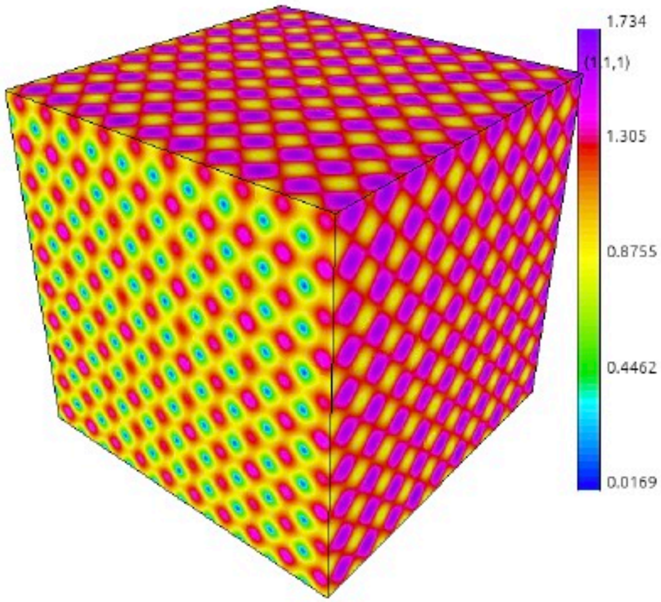
Sparse MF: HOD-BF: Finite Element for 3D Maxwell

$$\nabla \times \nabla \times \mathbf{E} - \Omega^2 \mathbf{E} = \mathbf{f}$$

- First-order Nedelec element discretization for 3D indefinite Maxwell in a homogenous cube
- $N = 14{,}827{,}904$, $\Omega = 16$, 24 points per wavelength
- 32 Cori Haswell nodes

ε	10^{-5}
HOD-BF fronts	6
Dense fronts	3,773,215
Factor time (sec)	581.1
Factor flops	$1.30 \cdot 10^{15}$
Flops fraction of direct (%)	61.3
Memory (GB)	426
Compression (%)	78.8
Maximum rank / front size	955 / 78203
GMRES its.	17
Solve flops	$5.73 \cdot 10^{12}$
Solve time (sec)	23.8

Solver breakdowns



$|\mathbf{E}(\mathbf{x})|$

Algorithm complexity (in bigO sense)

- Dense LU: $O(N^3)$
- Model PDEs with regular mesh, Nested Dissection ordering

	2D problems **$N = k^2$**			**3D problems** **$N = k^3$**		
	Factor flops	Solve flops	Memory	Factor flops	Solve flops	Memory
Exact sparse LU	$N^{3/2}$	$N \log(N)$	$N \log(N)$	N^2	$N^{4/3}$	$N^{4/3}$
STRUMPACK With LR compression	N	N	N	N^{α} polylog(N) ($\alpha < 2$)	$N \log(N)$	$N \log(N)$

Perspectives, future directions

- Techniques from structured matrices are very promising for preconditioning
  - LA tools: ID, randomization
  - Parallelism: coarse-grain (trees), fine-grain (dense submatrices)

- Caveat: wide spectrum of algorithms, not (yet) possible to have a decision tree of algorithm choices (e.g., iterative solution template book)
  - Problem-specific (esp. clustering)
  - Implementations are still being worked on for larger scale problems and machines

- Randomized LA is a very useful tool, rigorous error analyses are needed to understand approximation quality

References

- J. Xia, S. Chandrasekaran, M. Gu, X. S. Li, ``Fast Algorithms for Hierarchically Semiseparable Matrices'', Numerical Linear Algebra with Applications, Vol 17, Issue 6, 953-976, 2010.
- A. Napov and X.S. Li, ``An algebraic multifrontal preconditioner that exploits the low-rank property'', Numerical Linear Algebra with Applications, 2016, 23:61-82.
- C. Gorman, G. Chavez, P. Ghysels, T. Mary, F.-H. Rouet, X.S. Li, ``Robust and Accurate Stopping Criteria for Adaptive Randomized Sampling in Matrix-free Hierarchically Semiseparable Construction", SIAM J. Sci. Comput., Vol. 41, No. 5, pp. S61-S85, 2019.
- Y. Liu, W. Sid-Lakhdar, E. Rebrova, P. Ghysels, and X.S. Li, ``A Parallel Hierarchical Blocked Adaptive Cross Approximation Algorithm", Int. Journal of High Performance Computing, September 2019.
- G. Chavez, Y. Liu, P. Ghysels, X.S. Li, ``Scalable and Memory-Efficient Kernel Ridge Regression", IPDPS 2020, May 18-22, New Orleans.
- Y. Liu, X. Xing, H. Guo, E. Michielssen, P. Ghysels, X.S. Li, ``Butterfly factorization via randomized matrix-vector multiplications'', SIAM J. Sci. Comput., 2021
- Y. Liu, P. Ghysels, L. Claus, X.S. Li, ``Sparse Approximate Multifrontal Factorization with Butterfly Compression for High Frequency Wave Equations", SIAM J. Sci. Comput., 2021.

- P. Sao, R. Vuduc, X.S. Li, ``A communication-avoiding 3D algorithm for sparse LU factorization on heterogeneous systems'', J. Parallel and Distributed Computing (JPDC), September 2019.
- P. Sao, R. Kannan, X.S. Li, R. Vuduc, ``A communication-avoiding 3D sparse triangular solver", ICS 2019, June 26-28, Phoenix, AZ. Proceedings, pp. 127-137.

Questions?