Fast multipole matrix factorization

Kenneth L. Ho (TSMC)

SIAM AN 2018

Minden/Ho/Damle/Ying, A recursive skeletonization factorization based on strong admissibility, MMS 2017

Fast multipole matrix factorization

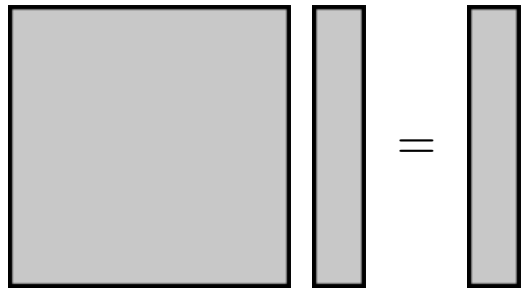
Kenneth L. Ho (TSMC)

SIAM AN 2018

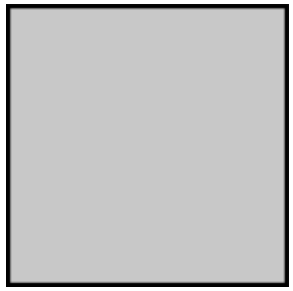
Minden/Ho/Damle/Ying, A recursive skeletonization factorization based on strong admissibility, MMS 2017

Structured matrix problems

$$Ax = b$$

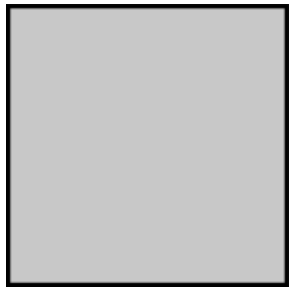


Structured matrix problems



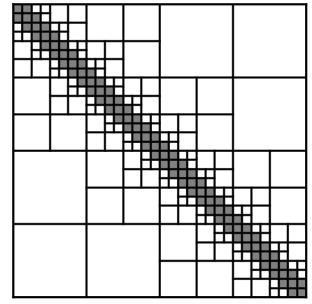
- Data: $O(N^2)$
- Matvec: $O(N^2)$
- Factor: $O(N^3)$

Structured matrix problems



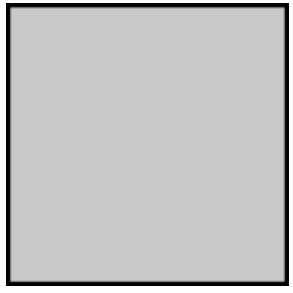
- Data: $O(N^2)$
- Matvec: $O(N^2)$
- Factor: $O(N^3)$

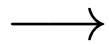
$\longrightarrow$

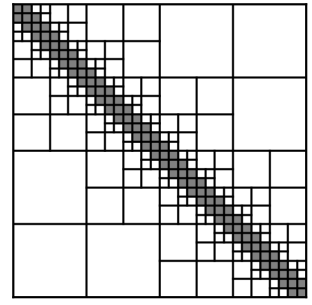


- Data: $O(N)$
- Matvec: $O(N)$
- Factor: $O(N)$

Structured matrix problems





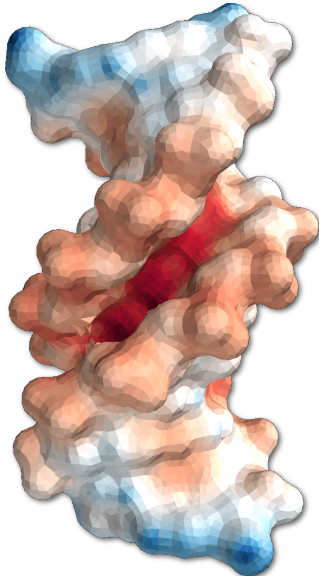
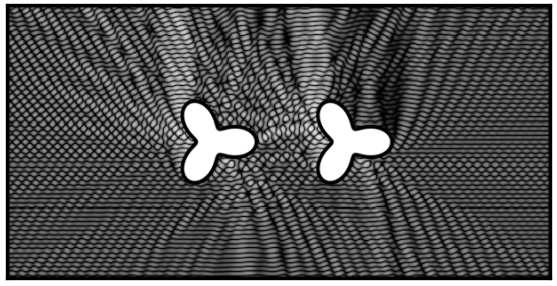
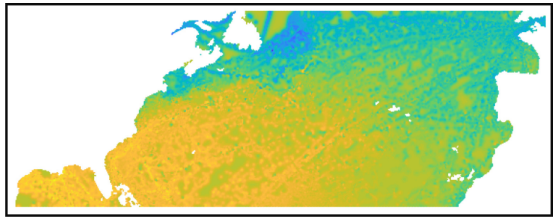
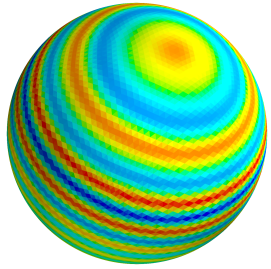
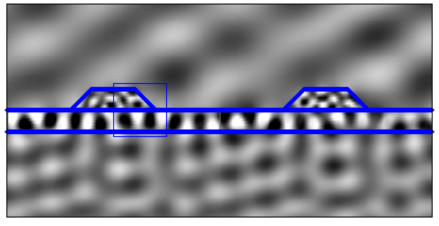
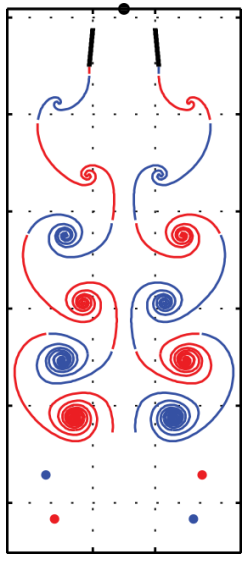


- Data: $O(N^2)$
- Matvec: $O(N^2)$
- Factor: $O(N^3)$

- Data: $O(N)$
- Matvec: $O(N)$
- Factor: $O(N)$

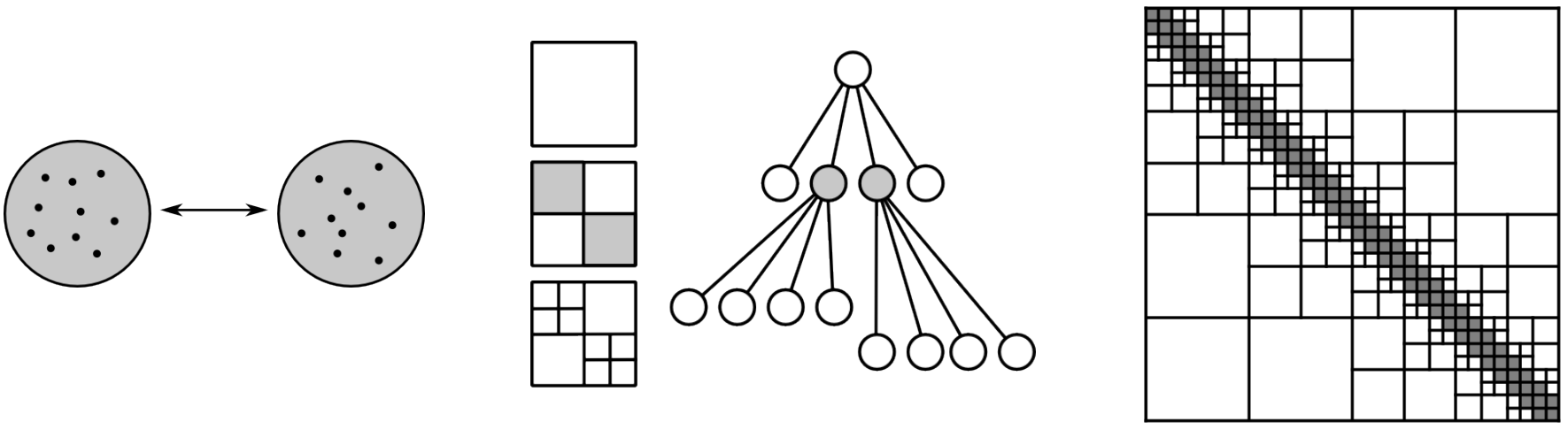
- **Integral equations:** $a(x)u(x) + \int K(x,y)u(y)\,dy = f(x)$
- **PDEs:** $-\nabla \cdot (a(x)\nabla u(x)) + b(x)u(x) = f(x)$
- **Stats/ML:** $f(\hat{x}) = K(\hat{x},x)K^{-1}(x,x)f(x)$

Structured matrix problems

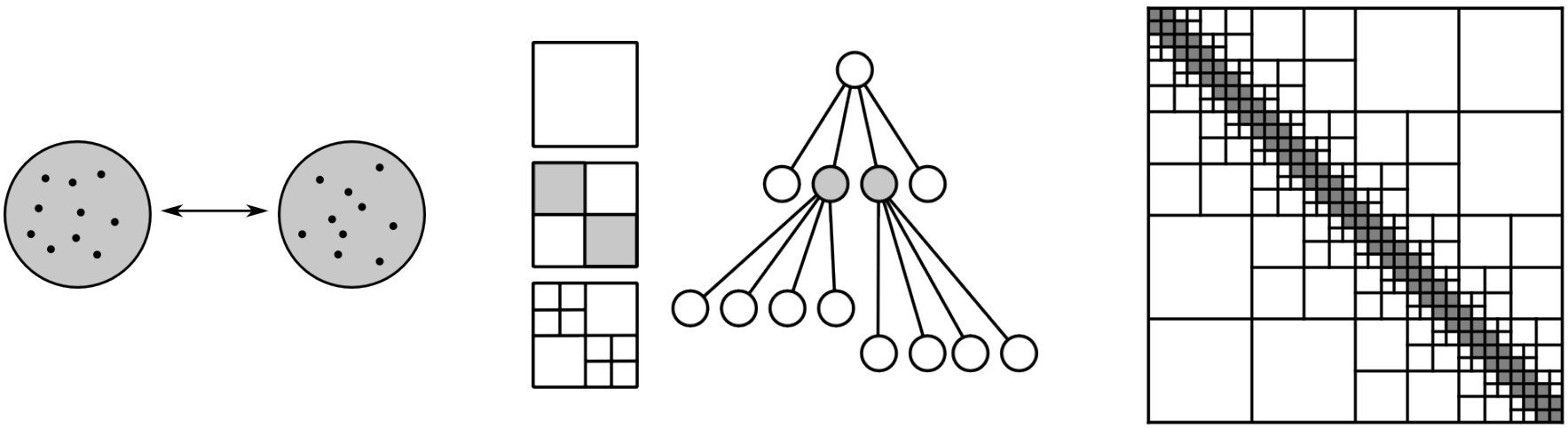
- **Integral equations:** $a(x)u(x) + \int K(x,y)u(y)\,dy = f(x)$
- **PDEs:** $-\nabla \cdot (a(x)\nabla u(x)) + b(x)u(x) = f(x)$
- **Stats/ML:** $f(\hat{x}) = K(\hat{x},x)K^{-1}(x,x)f(x)$

Brief history



- **Core idea:** geometry $\Longrightarrow$ low-rank, exploit hierarchically

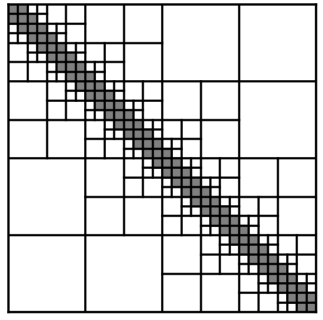
Brief history



- **Core idea:** geometry $\Longrightarrow$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices

[Barnes/Hut, Greengard/Rokhlin, Hackbusch et. al., ...]

Brief history



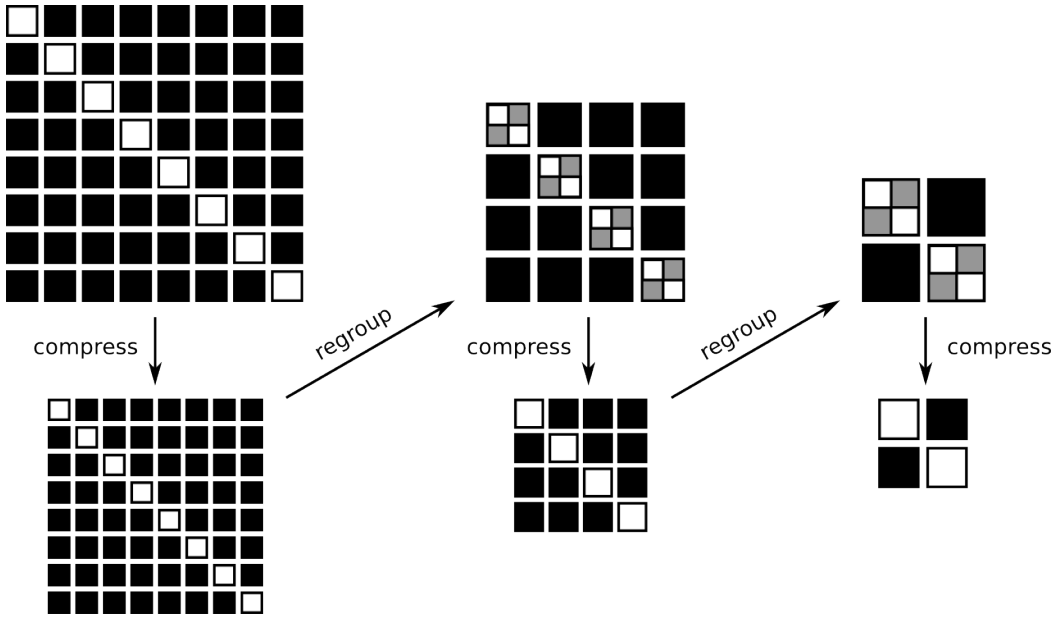
$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$\Longrightarrow M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix}, \quad S = D - CA^{-1}B$$

- **Core idea:** geometry $\Longrightarrow$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices
- 90s/00s: $\mathcal{H}$-matrix-based direct solvers

[Hackbusch, Börm, Bebendorf, Grasedyck, ...]

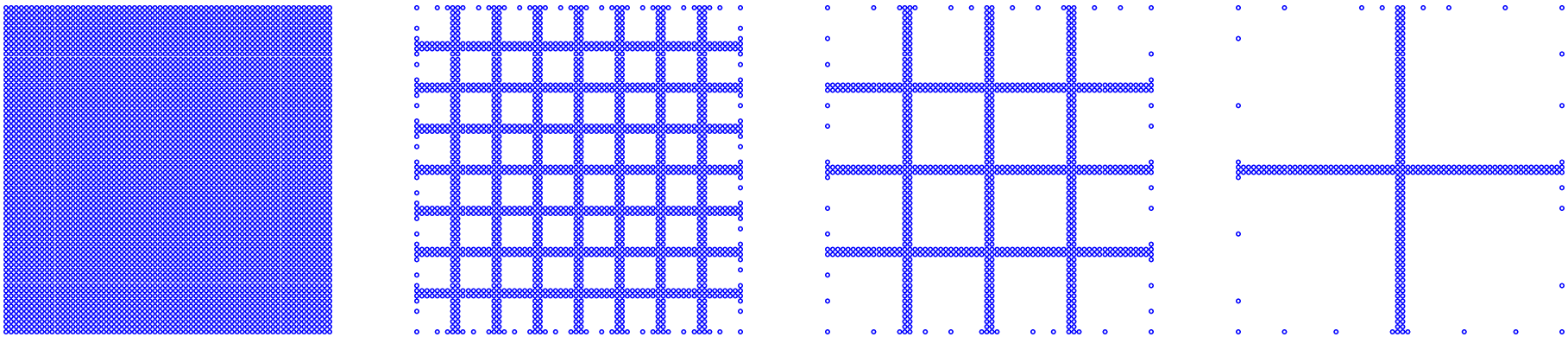
Brief history



- **Core idea:** geometry $\implies$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices
- 90s/00s: $\mathcal{H}$-matrix-based direct solvers
- 00s/10s: more efficient "1D" solvers; Martinsson/Rokhlin, HSS, HODLR, etc.

[Martinsson/Gillman/Rokhlin, Ho/Greengard, Chandrasekaran/Gu, Xia et. al., Ambikasaran/Darve, ...]

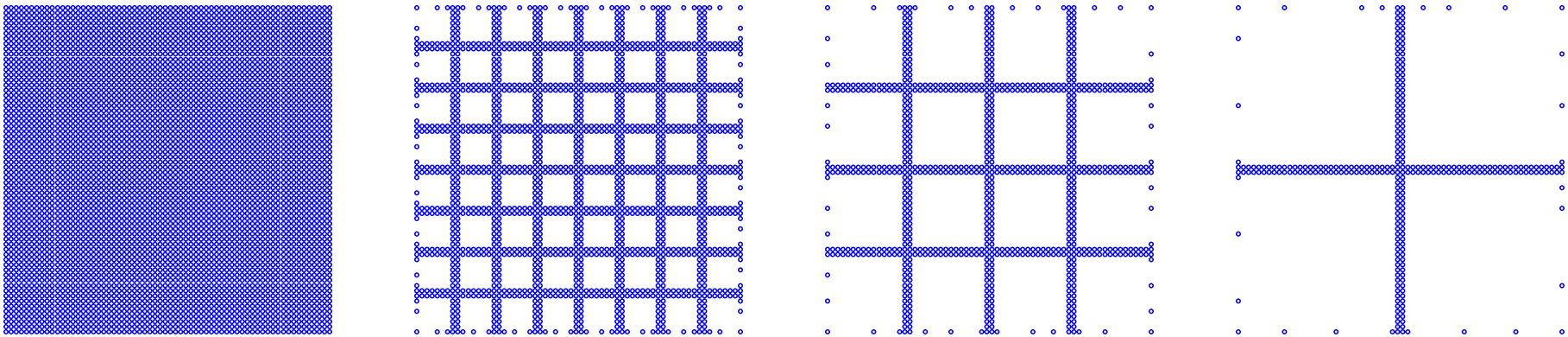
Brief history



- **Core idea:** geometry $\implies$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices
- 90s/00s: $\mathcal{H}$-matrix-based direct solvers
- 00s/10s: more efficient "1D" solvers; Martinsson/Rokhlin, HSS, HODLR, etc.

[Martinsson/Gillman/Rokhlin, Ho/Greengard, Chandrasekaran/Gu, Xia et. al., Ambikasaran/Darve, ...]

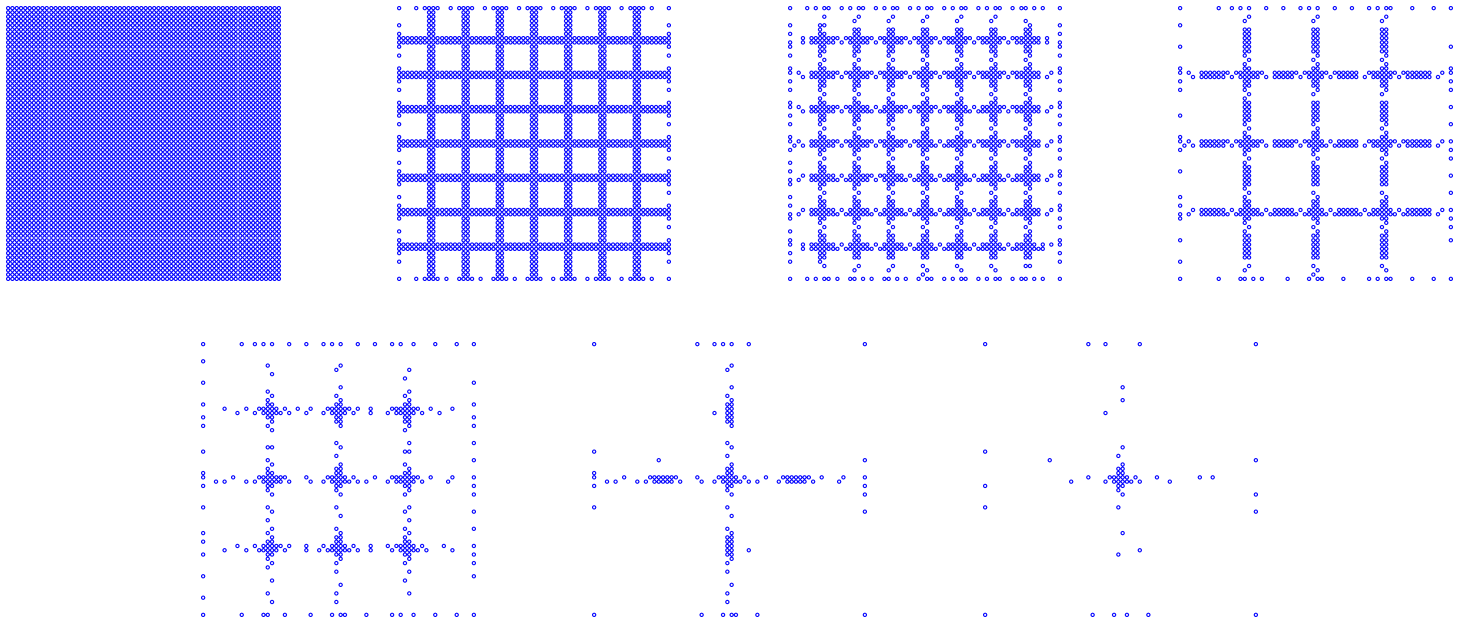
Brief history



- **Core idea:** geometry $\implies$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices
- 90s/00s: $\mathcal{H}$-matrix-based direct solvers
- 00s/10s: more efficient "1D" solvers; Martinsson/Rokhlin, HSS, HODLR, etc.
- 10s: HSS-C, HIF, MHS, IFMM, ...

[Corona/Martinsson/Zorin, Ho/Ying, Xia, Ambikasaran/Coulier/Pouransari/Darve, ...]

Brief history



- **Core idea:** geometry $\implies$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices
- 90s/00s: $\mathcal{H}$-matrix-based direct solvers
- 00s/10s: more efficient "1D" solvers; Martinsson/Rokhlin, HSS, HODLR, etc.
- 10s: HSS-C, HIF, MHS, IFMM, ...

[Corona/Martinsson/Zorin, Ho/Ying, Xia, Ambikasaran/Coulier/Pouransari/Darve, ...]

Brief history

$$A = B + USV^*$$

$$Ax = b \implies \underbrace{\begin{bmatrix} B & U & \\ V^* & & -I \\ & -I & S \end{bmatrix}}_{\text{eliminate and compress}} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

- **Core idea:** geometry $\implies$ low-rank, exploit hierarchically
- 80s/90s: fast matvec/iterative solvers with treecodes, FMM, $\mathcal{H}$-matrices
- 90s/00s: $\mathcal{H}$-matrix-based direct solvers
- 00s/10s: more efficient "1D" solvers; Martinsson/Rokhlin, HSS, HODLR, etc.
- 10s: HSS-C, HIF, MHS, IFMM, ...

[Corona/Martinsson/Zorin, Ho/Ying, Xia, Ambikasaran/Coulier/Pouransari/Darve, ...]

This talk

FMM factorization: direct sparsification/elimination via "far-field skeletons"

- Much simpler $O(N)$ with no tree breaking, nested hierarchy, auxiliary matrices, etc.
- Based on multiplicative recursive skeletonization factorization framework (RS-S)
- Generalized LU decomposition: matvec, solve, determinant, Cholesky, etc.
- Opinion: proper translation of FMM ideas to direct solver context
- Similar method for $\mathcal{H}^2$-matrices in [Sushnikova/Oseledets '17]

This talk

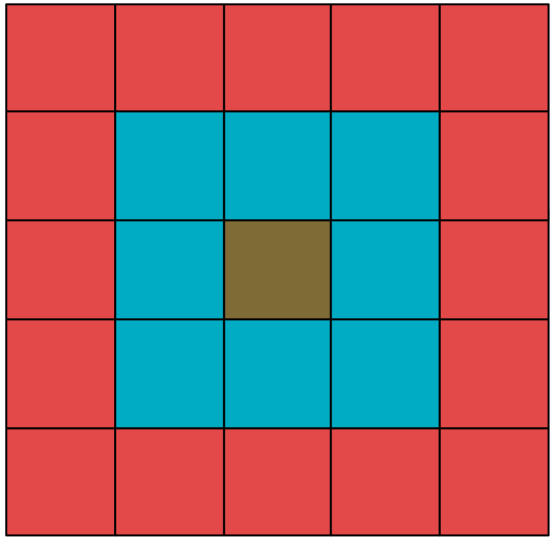
FMM factorization: direct sparsification/elimination via "far-field skeletons"

- Much simpler $O(N)$ with no tree breaking, nested hierarchy, auxiliary matrices, etc.
- Based on multiplicative recursive skeletonization factorization framework (RS-S)
- Generalized LU decomposition: matvec, solve, determinant, Cholesky, etc.
- Opinion: proper translation of FMM ideas to direct solver context
- Similar method for $\mathcal{H}^2$-matrices in [Sushnikova/Oseledets '17]

Extensions/directions/perspectives:

- What else can we do?
- Where do we go from here?

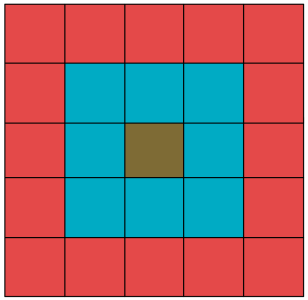
Domain partitioning



- **Brown:** current box
- **Blue:** near field (neighbors)
- **Red:** far field (everybody else)

$$A = \left[\begin{array}{c|c|c} A_{bb} & A_{bn} & A_{bf} \\ \hline A_{nb} & A_{nn} & A_{nf} \\ \hline A_{fb} & A_{fb} & A_{ff} \end{array}\right] \qquad \text{with} \qquad A_{bf} \text{ and } A_{fb} \text{ low-rank}$$

Interpolative decomposition

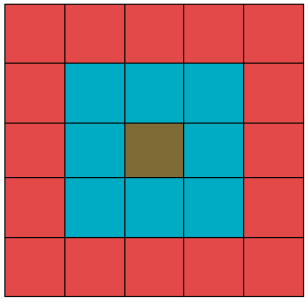


- Find partitioning $b = r \cup s$ and interpolation matrix T such that $\begin{bmatrix} A_{fr} \\ A_{rf}^* \end{bmatrix} \approx \begin{bmatrix} A_{fs} \\ A_{sf}^* \end{bmatrix} T$
- Interpolate **redundant** indices r from **skeleton** indices s
- Structure-preserving compression; basically an RRQR

$$\left[\begin{array}{c|c|c} A_{bb} & A_{bn} & A_{bf} \\ \hline A_{nb} & A_{nn} & A_{nf} \\ \hline A_{fb} & A_{fb} & A_{ff} \end{array}\right] = \left[\begin{array}{cc|c|c} A_{rr} & A_{rs} & A_{rn} & A_{rf} \\ A_{sr} & A_{ss} & A_{sn} & A_{sf} \\ \hline A_{nr} & A_{ns} & A_{nn} & A_{nf} \\ \hline A_{fr} & A_{fs} & A_{fn} & A_{ff} \end{array}\right] \approx \left[\begin{array}{cc|c|c} A_{rr} & A_{rs} & A_{rn} & T^* A_{sf} \\ A_{sr} & A_{ss} & A_{sn} & A_{sf} \\ \hline A_{nr} & A_{ns} & A_{nn} & A_{nf} \\ \hline A_{fs} T & A_{fs} & A_{fn} & A_{ff} \end{array}\right]$$

[Cheng/Gimbutas/Martinsson/Rokhlin '05, Gu/Eisenstat '96]

Far-field (strong) skeletonization



$$\left[\begin{array}{ccc|c} A_{rr} & A_{rs} & A_{rn} & T^* A_{sf} \\ A_{sr} & A_{ss} & A_{sn} & A_{sf} \\ \hline A_{nr} & A_{ns} & A_{nn} & A_{nf} \\ \hline A_{fs} T & A_{fs} & A_{fn} & A_{ff} \end{array}\right] \xrightarrow{\text{sparsify}} \left[\begin{array}{ccc|c} X_{rr} & X_{rs} & X_{rn} & \\ X_{sr} & A_{ss} & A_{sn} & A_{sf} \\ \hline X_{nr} & A_{ns} & A_{nn} & A_{nf} \\ \hline & A_{fs} & A_{fn} & A_{ff} \end{array}\right]$$

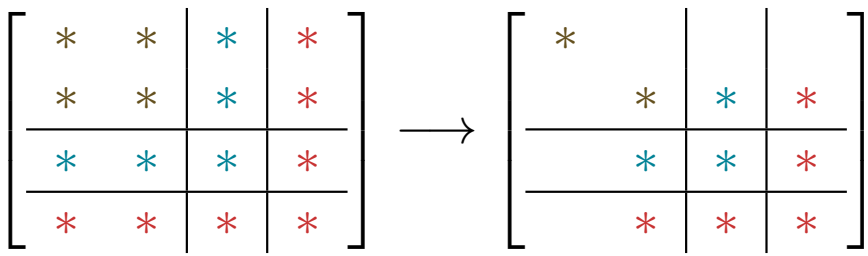
$$\xrightarrow{\text{elim}} \left[\begin{array}{ccc|c} X_{rr} & & & \\ & X_{ss} & X_{sn} & A_{sf} \\ \hline & X_{ns} & X_{nn} & A_{nf} \\ \hline & A_{fs} & A_{ff} & A_{ff} \end{array}\right]$$

- Redundant DOFs eliminated, neighbor interactions updated, far field unchanged
- Sequence of block triangular transformations

[Ho/Ying '16]

Algorithm

Build (adaptive) quadtree/octree.
for each level $\ell = 1, 2, \ldots, L$ from finest to coarsest **do**
    for each box on level ℓ **do**
        Far-field skeletonize remaining DOFs in box.
    end for
end for



Algorithm

Build (adaptive) quadtree/octree.
for each level $\ell = 1, 2, \ldots, L$ from finest to coarsest **do**
    for each box on level ℓ **do**
        Far-field skeletonize remaining DOFs in box.
    end for
end for

- Block diagonalization:

$$D \approx U_L^* \cdots U_1^* A V_1 \cdots V_L$$

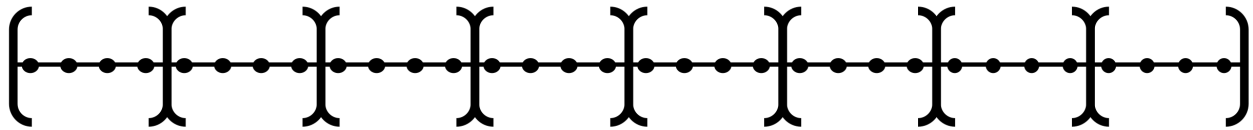
- Generalized LU decomposition:

$$\begin{aligned} A &\approx U_1^{-*} \cdots U_L^{-*} D \quad V_L^{-1} \cdots V_1^{-1} \\ A^{-1} &\approx V_1 \quad \cdots V_L \quad D^{-1} U_L^* \quad \cdots U_1^* \end{aligned}$$

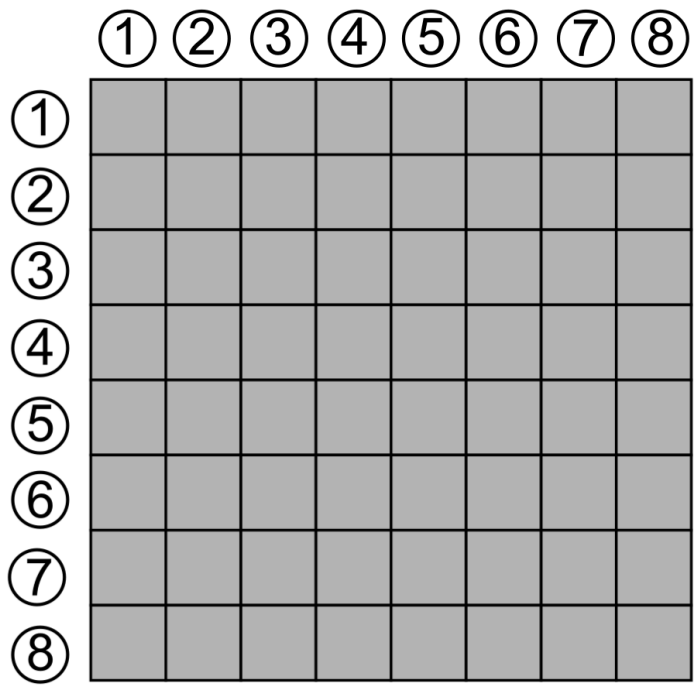
1D cartoon

Level 1

Domain:



Matrix:

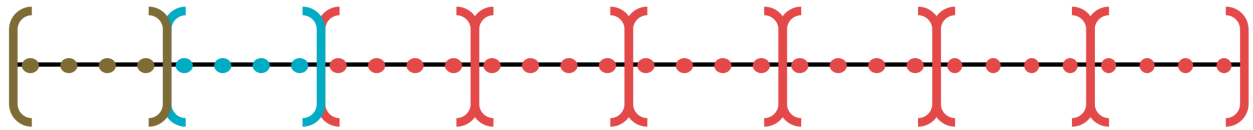


(but, of course, should just use RS)

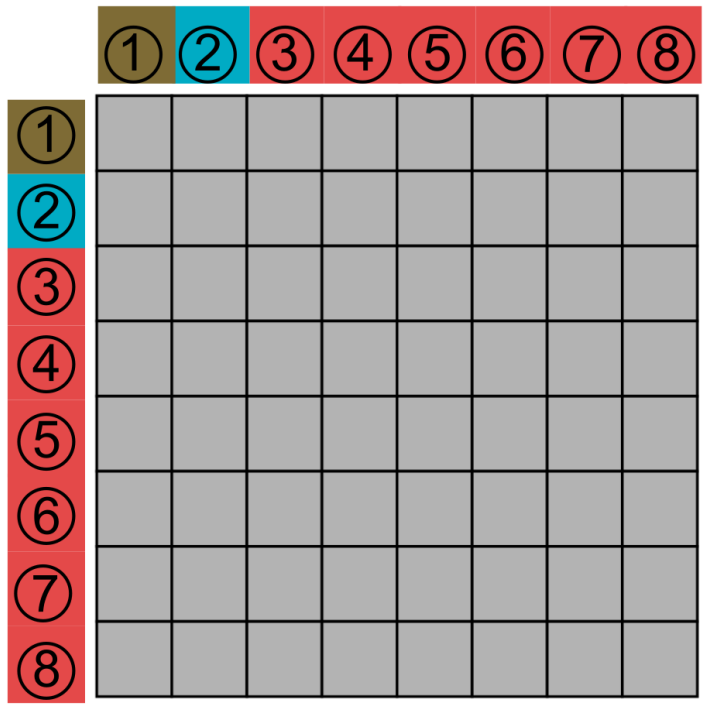
1D cartoon

Level 1: box 1 (before)

Domain:



Matrix:

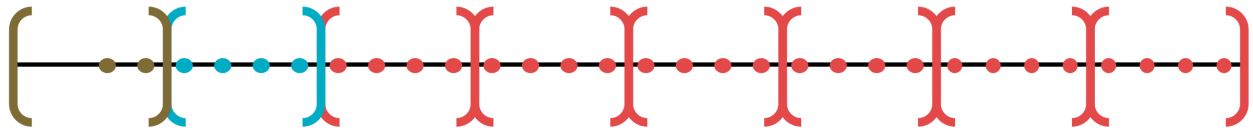


(but, of course, should just use RS)

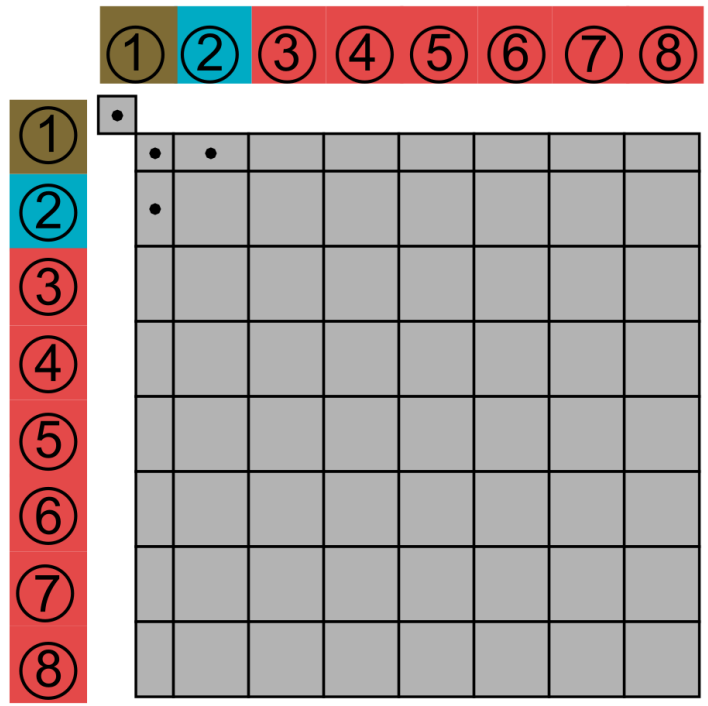
1D cartoon

Level 1: box 1 (after)

Domain:



Matrix:

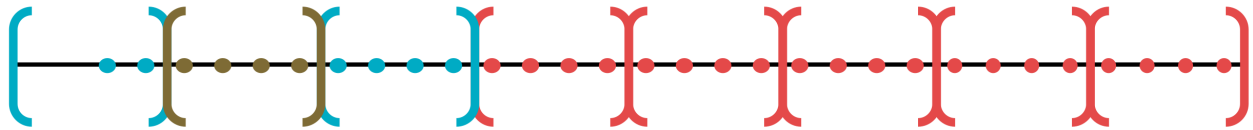


(but, of course, should just use RS)

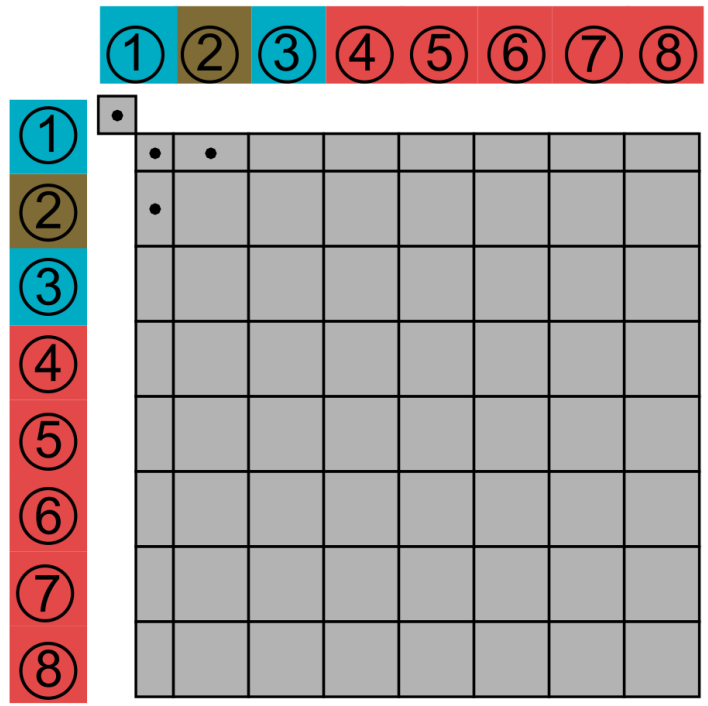
1D cartoon

Level 1: box 2 (before)

Domain:



Matrix:

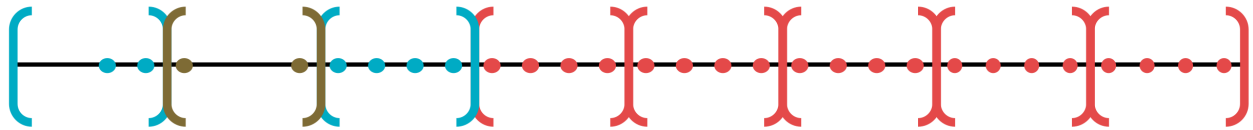


(but, of course, should just use RS)

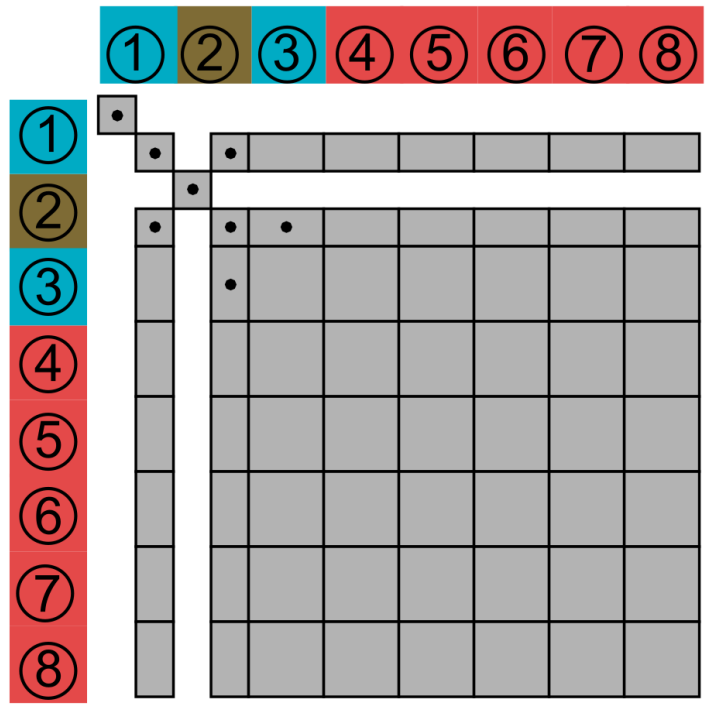
1D cartoon

Level 1: box 2 (after)

Domain:



Matrix:



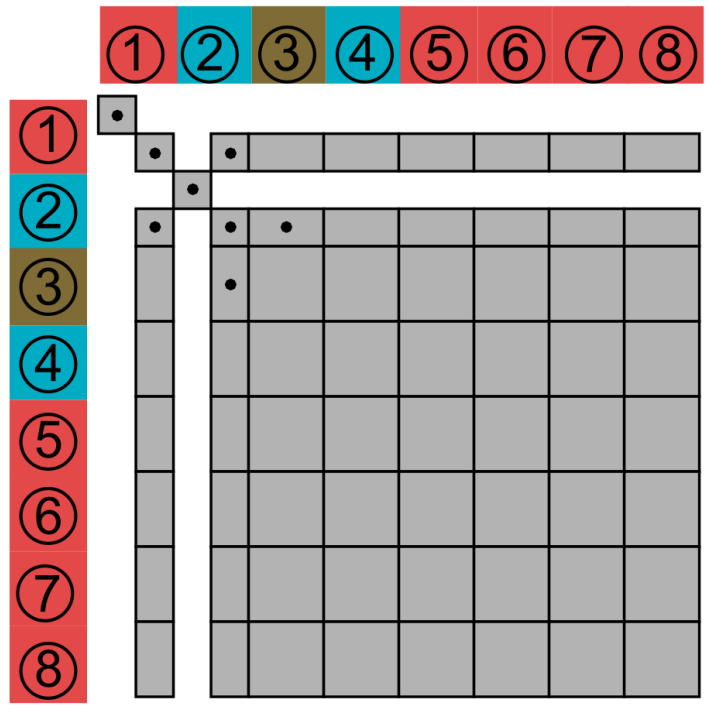
(but, of course, should just use RS)

1D cartoon

Level 1: box 3 (before)

Domain:

Matrix:



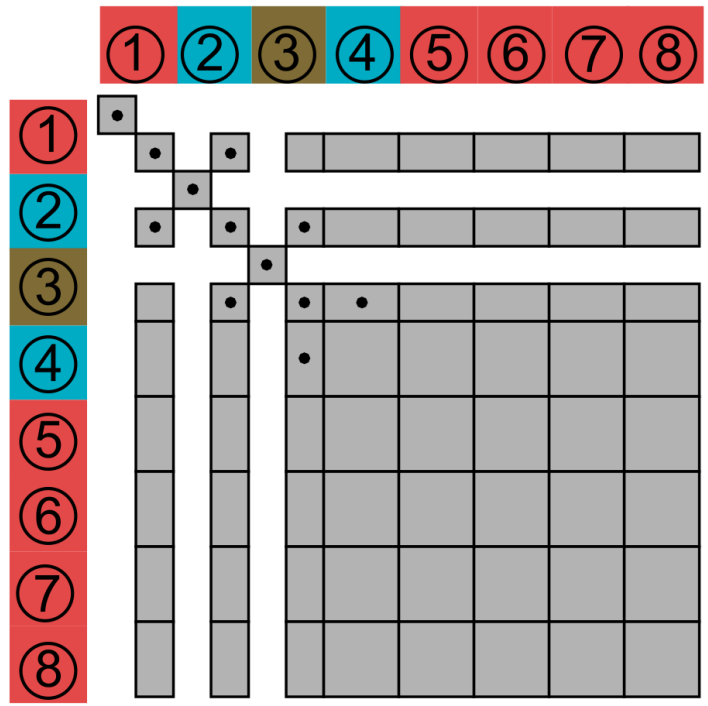
(but, of course, should just use RS)

1D cartoon

Level 1: box 3 (after)

Domain:

Matrix:

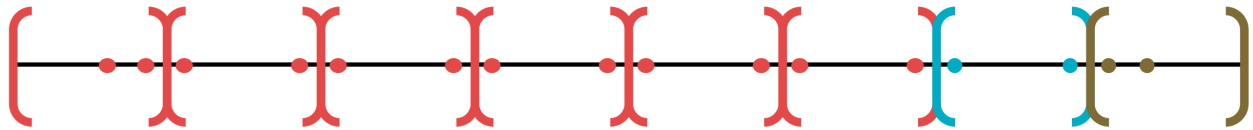


(but, of course, should just use RS)

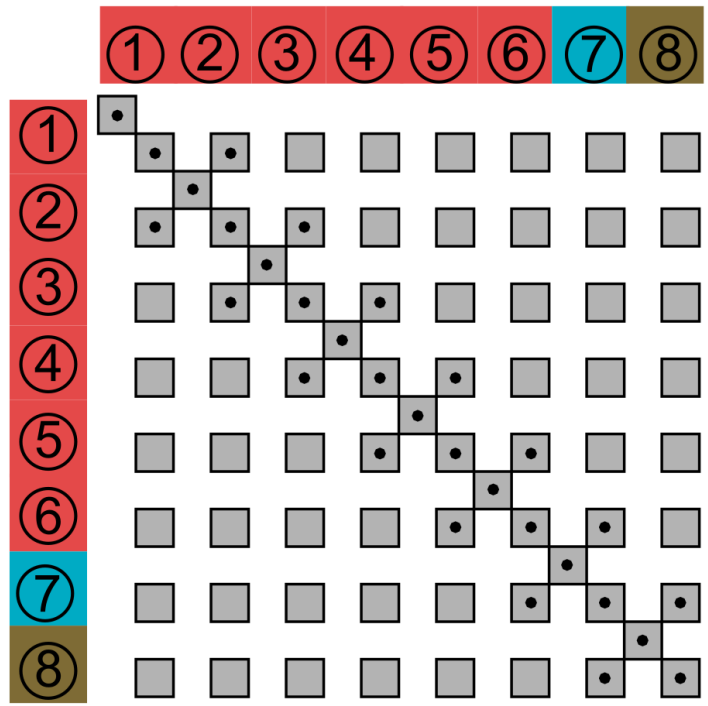
1D cartoon

Level 1: ... box 8 (after)

Domain:



Matrix:

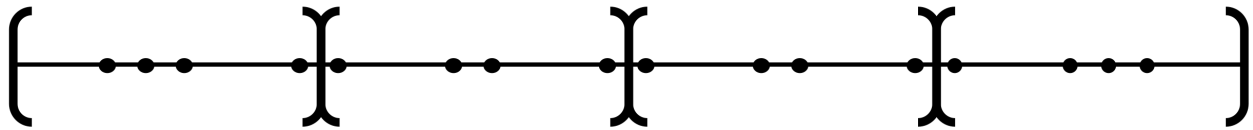


(but, of course, should just use RS)

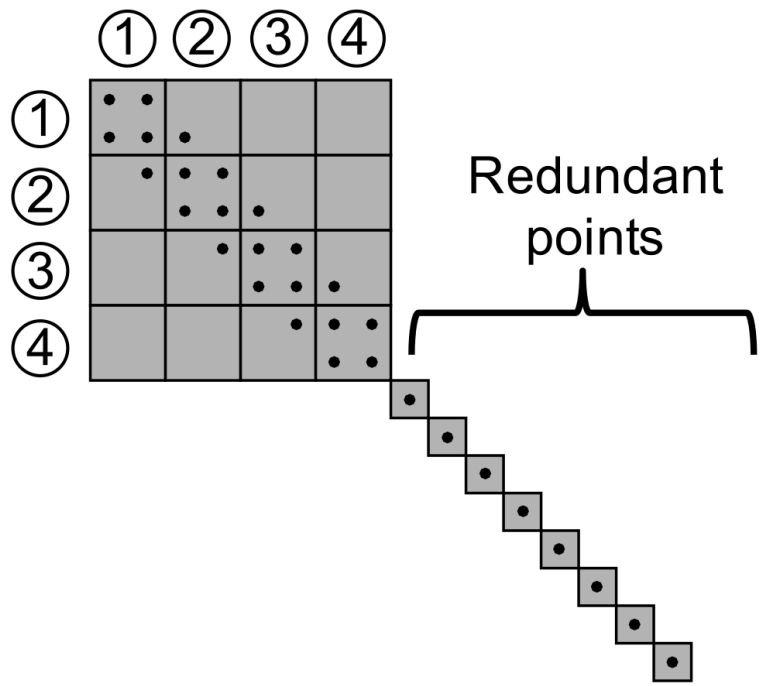
1D cartoon

Level 2: permute and combine boxes

Domain:



Matrix:



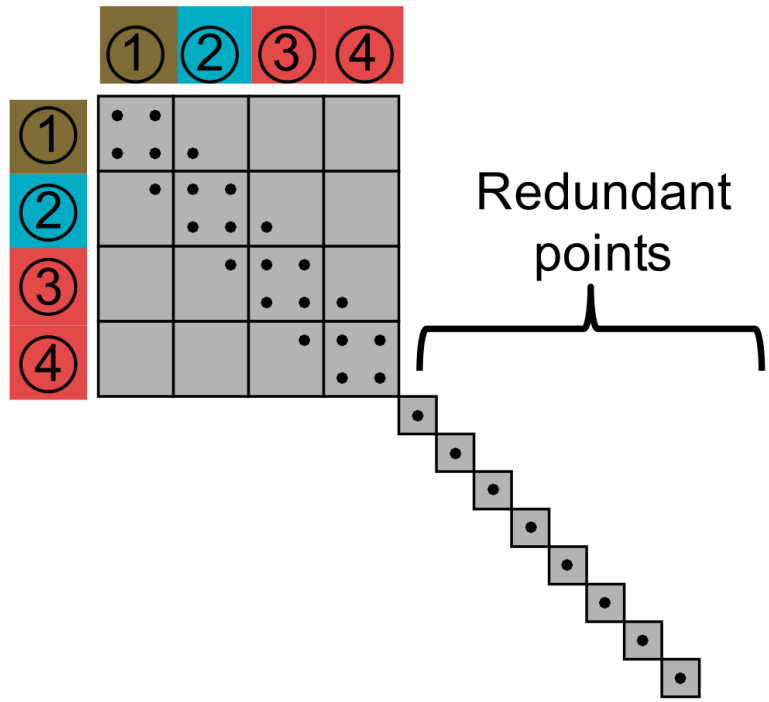
(but, of course, should just use RS)

1D cartoon

Level 2: ... and so on

Domain:

Matrix:

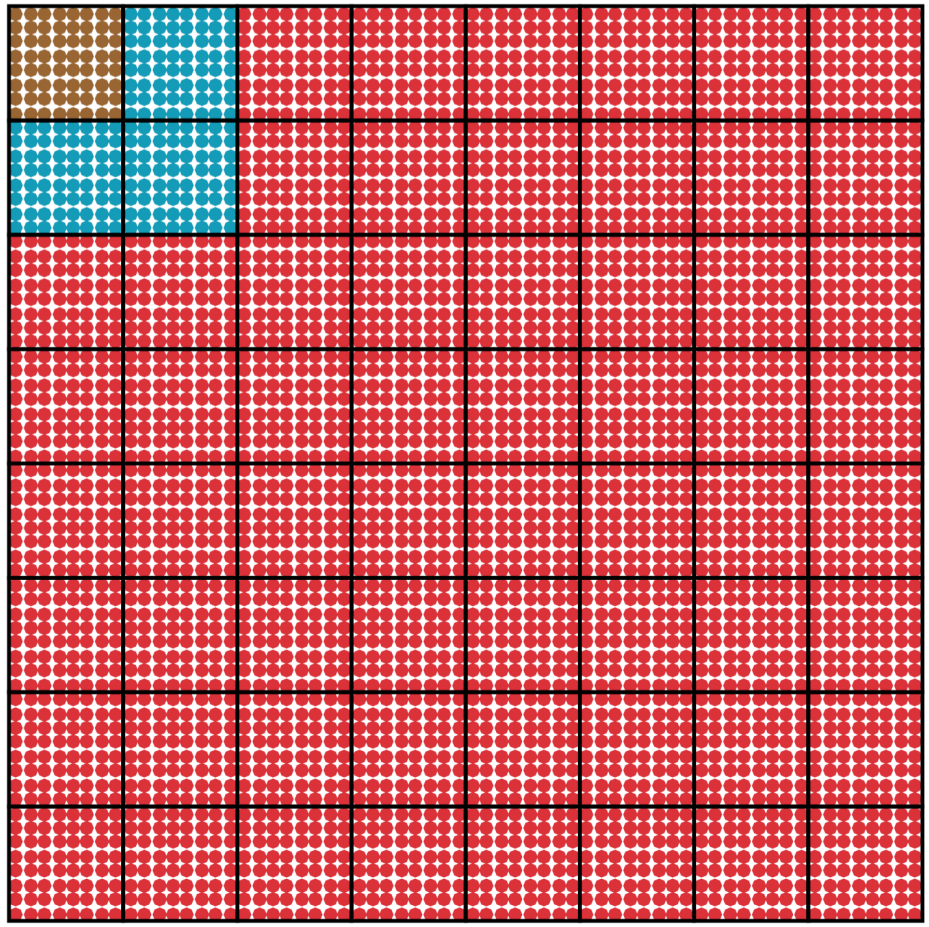


(but, of course, should just use RS)

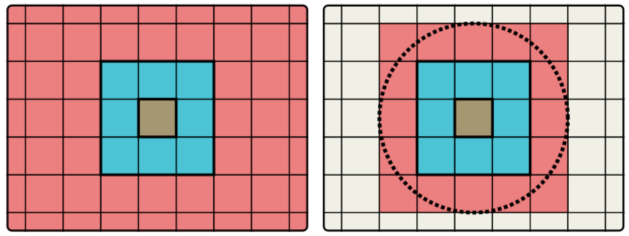
2D cartoon

Level 1: box 1 (before)

Domain:

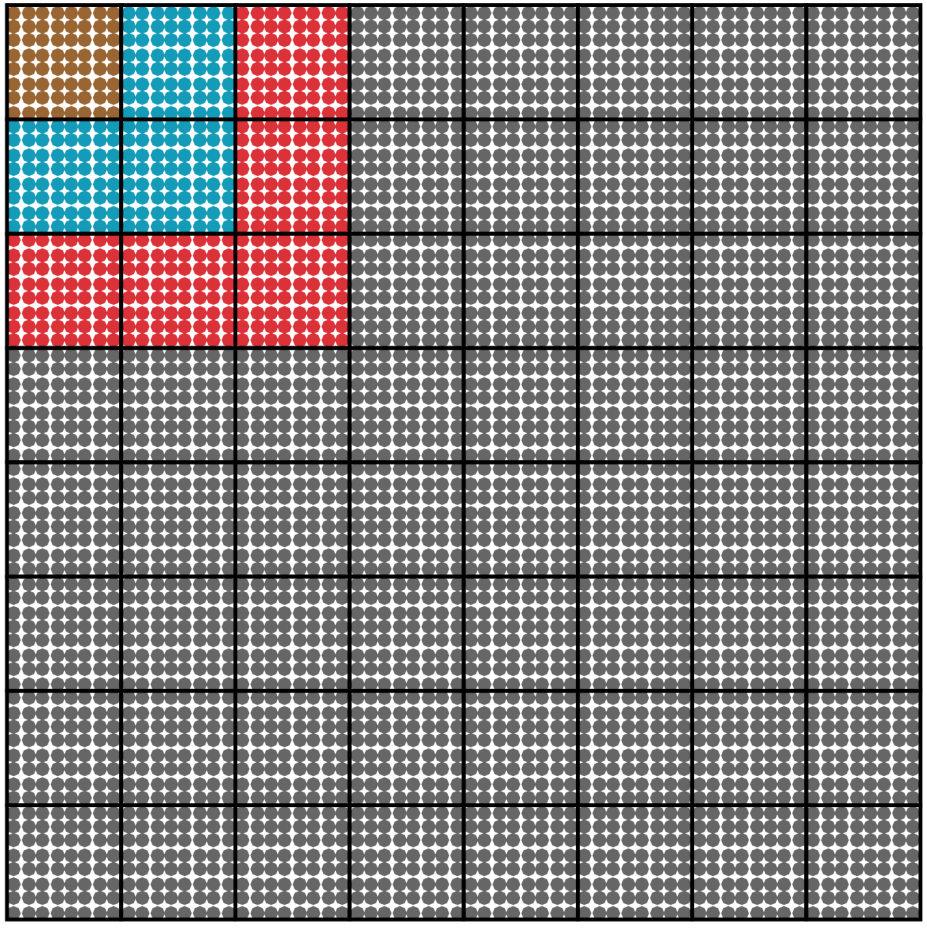


2D cartoon



Level 1: box 1 (before), use proxy trick

Domain:

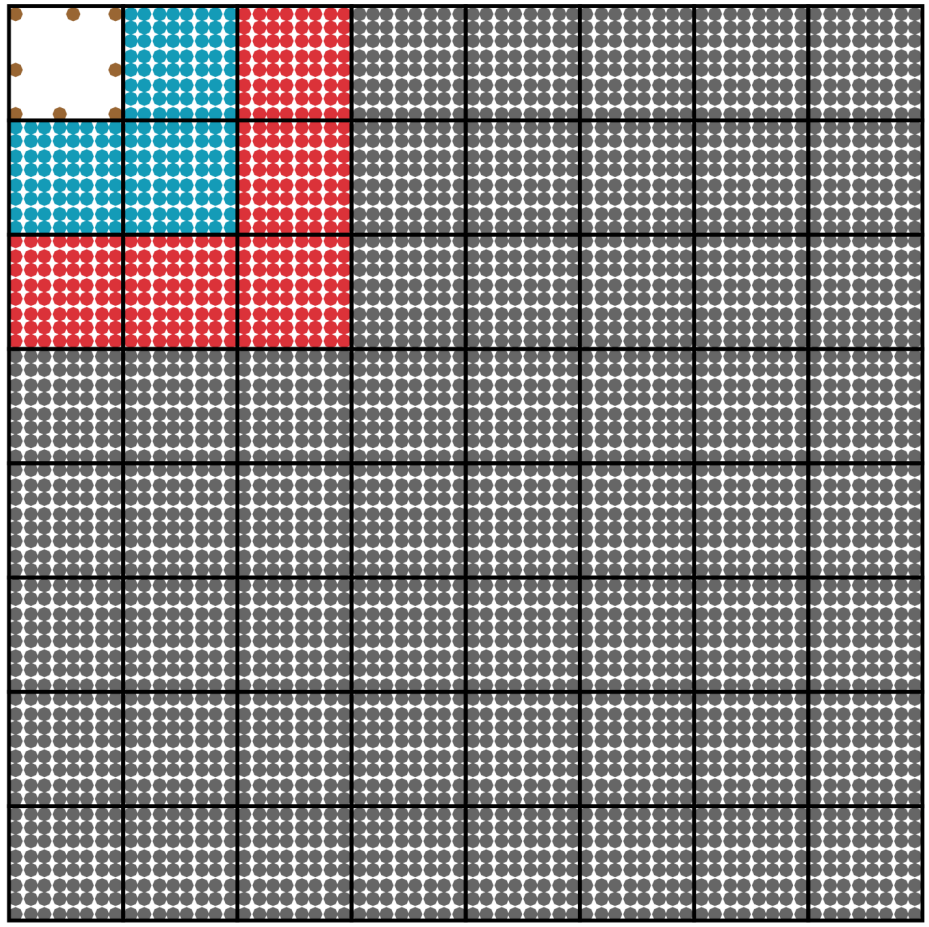


[Martinsson/Rokhlin '05]

2D cartoon

Level 1: box 1 (after)

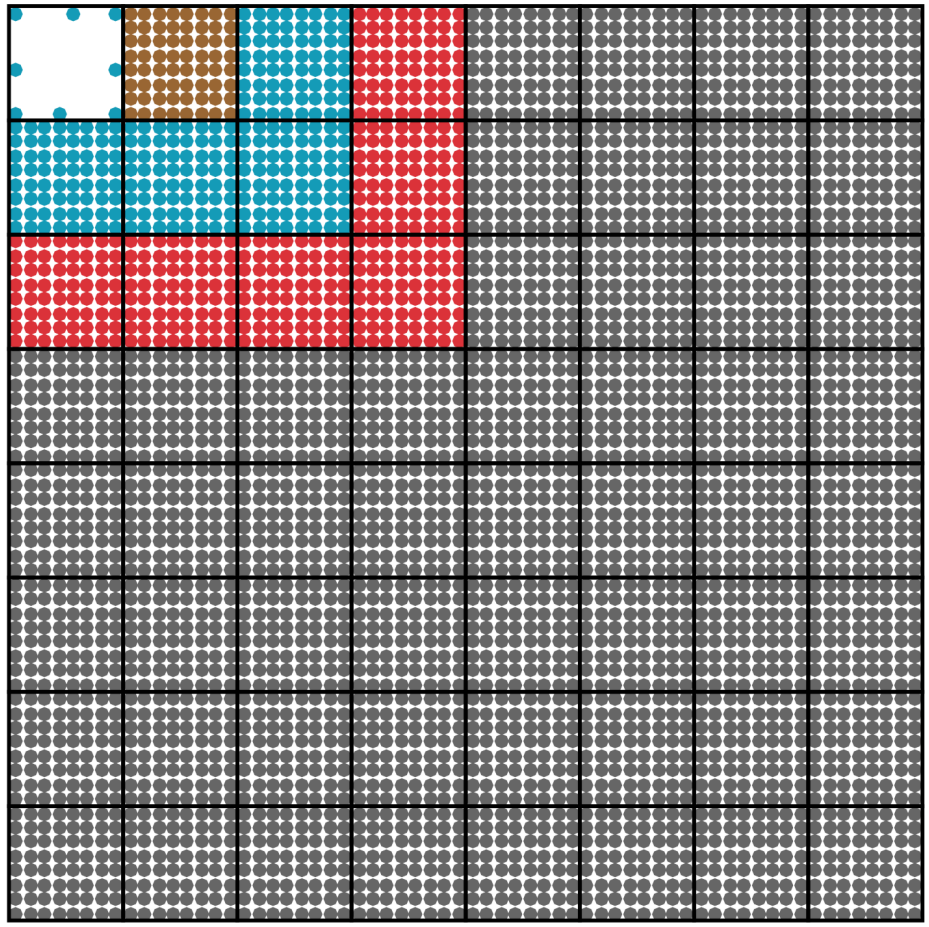
Domain:



2D cartoon

Level 1: box 2 (before)

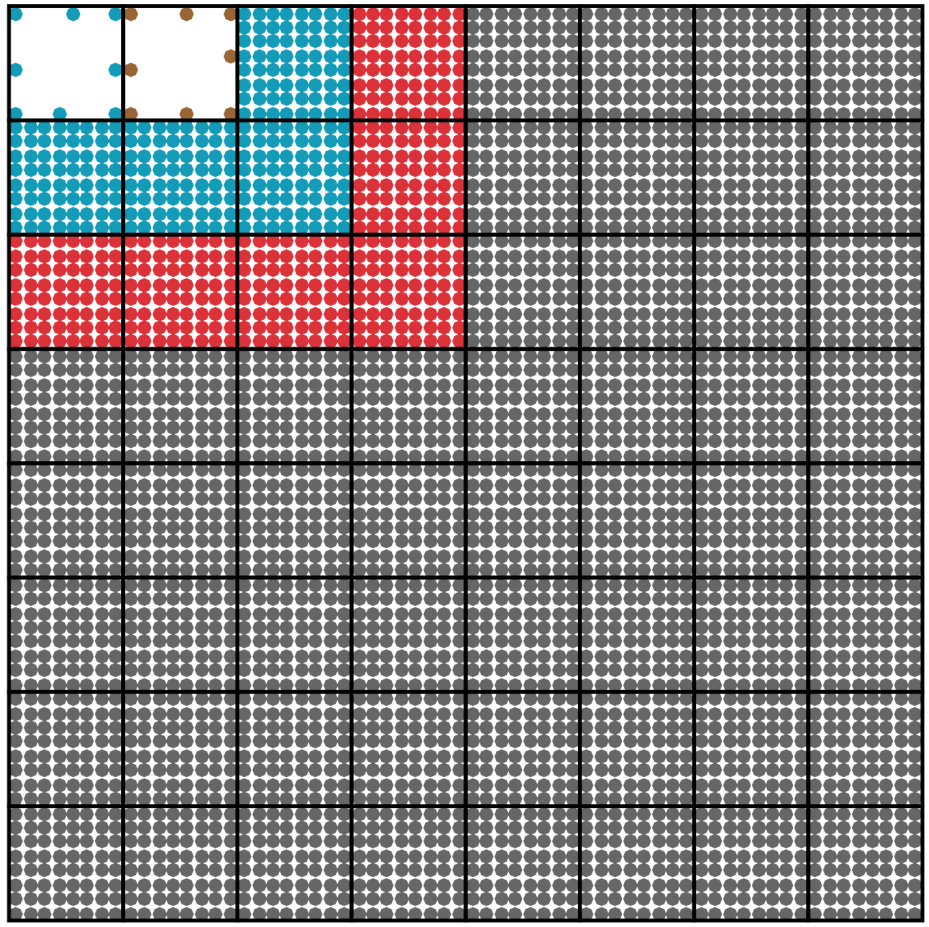
Domain:



2D cartoon

Level 1: box 2 (after)

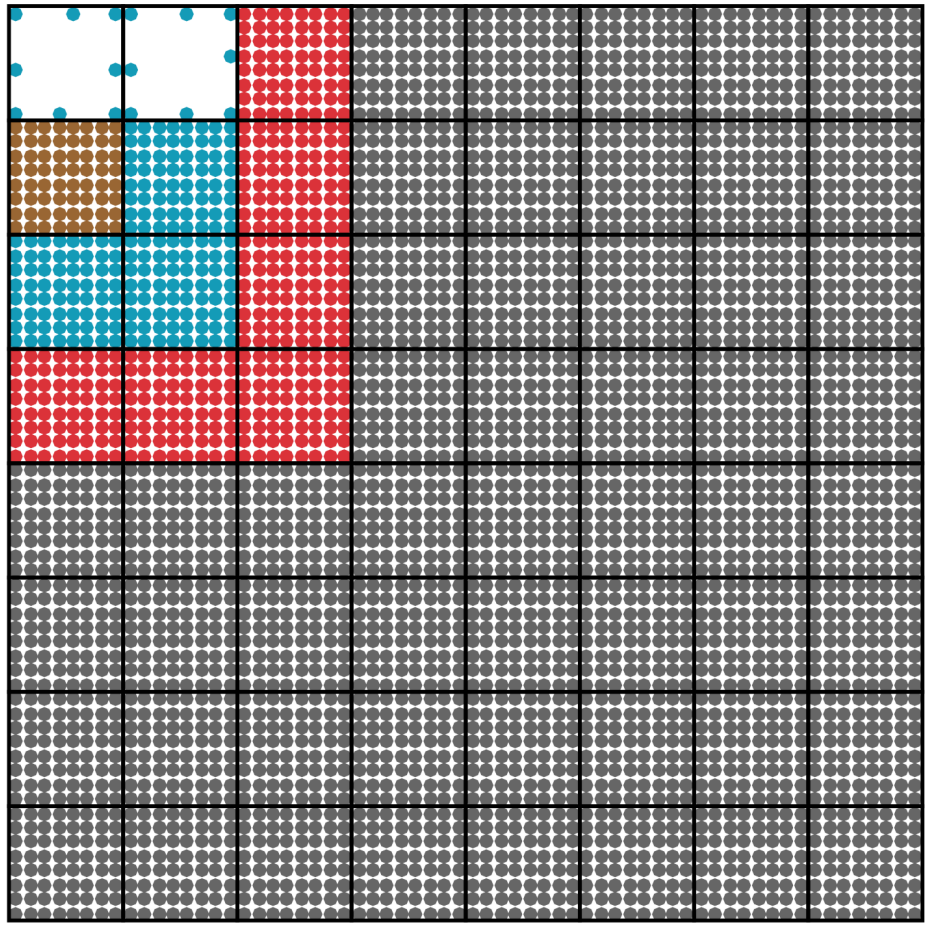
Domain:



2D cartoon

Level 1: box 3 (before)

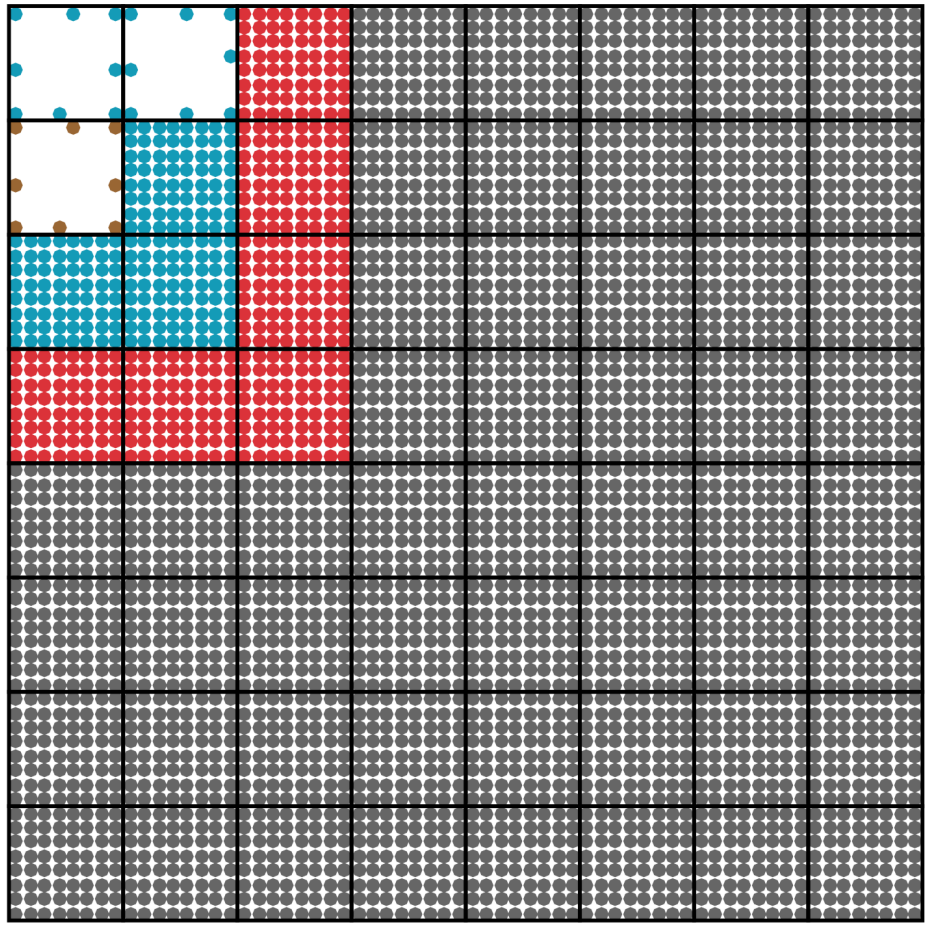
Domain:



2D cartoon

Level 1: box 3 (after)

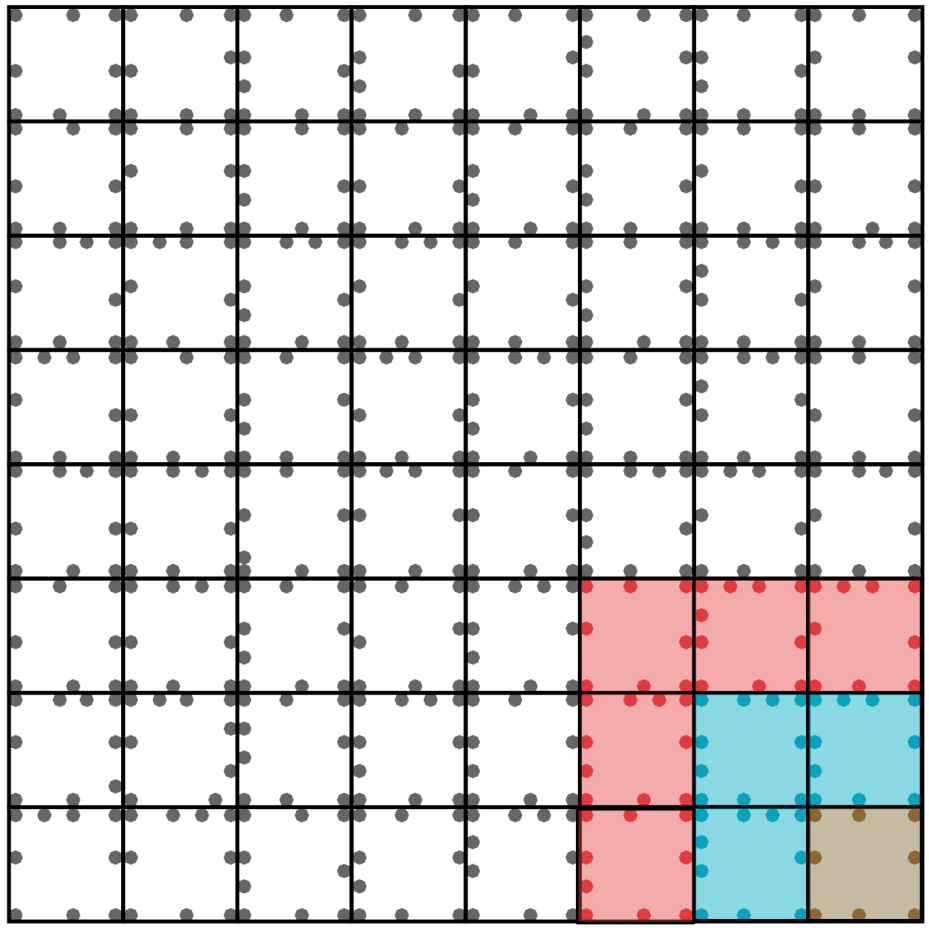
Domain:



2D cartoon

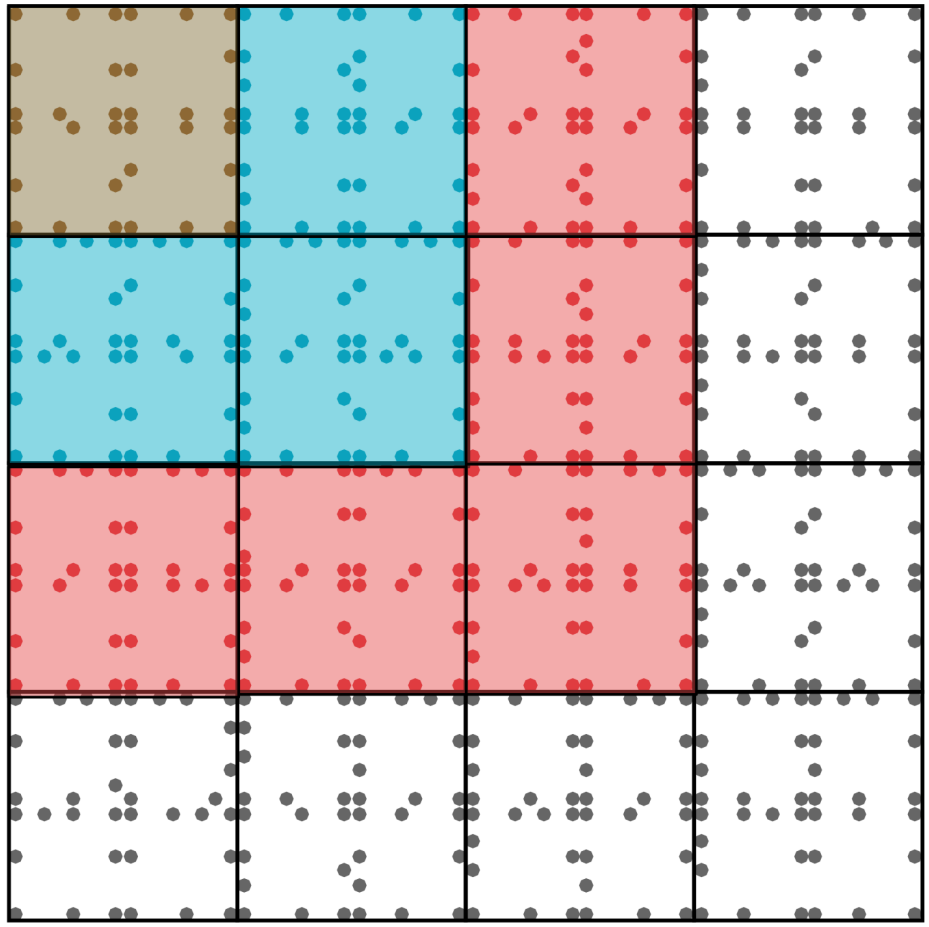
Level 1: ... box 64 (after)

Domain:



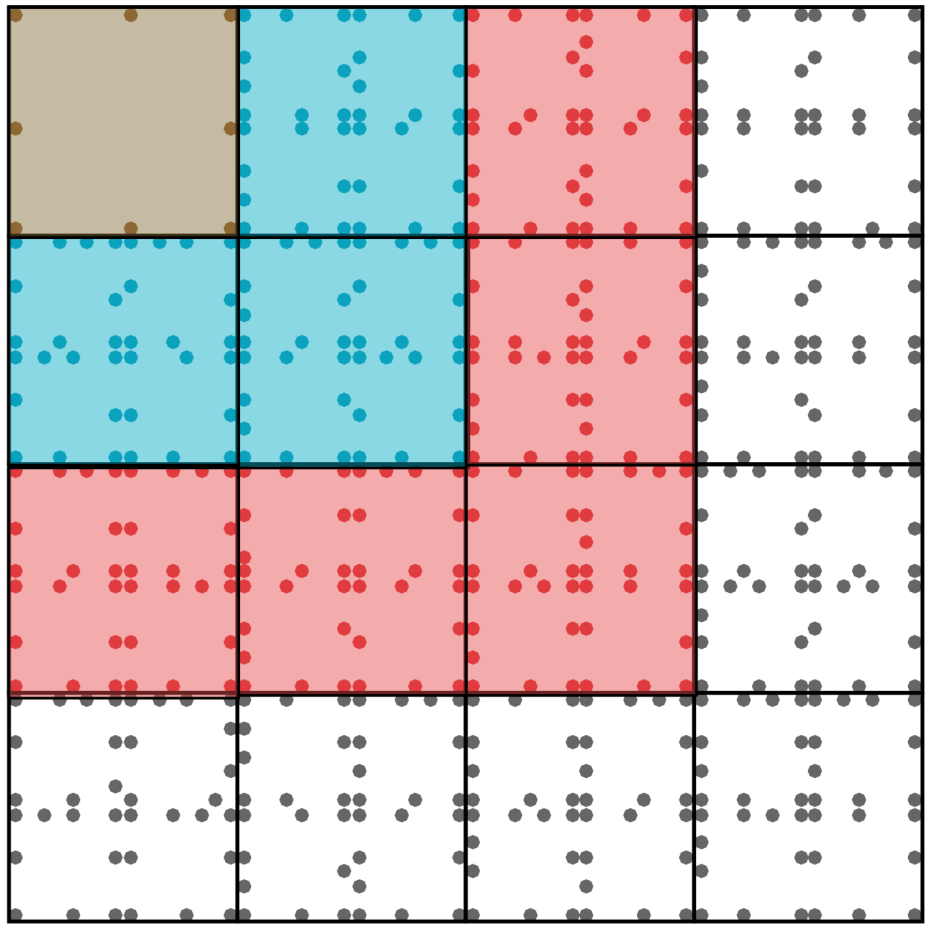
Level 2: box 1 (before)

Domain:



Level 2: box 1 (after)

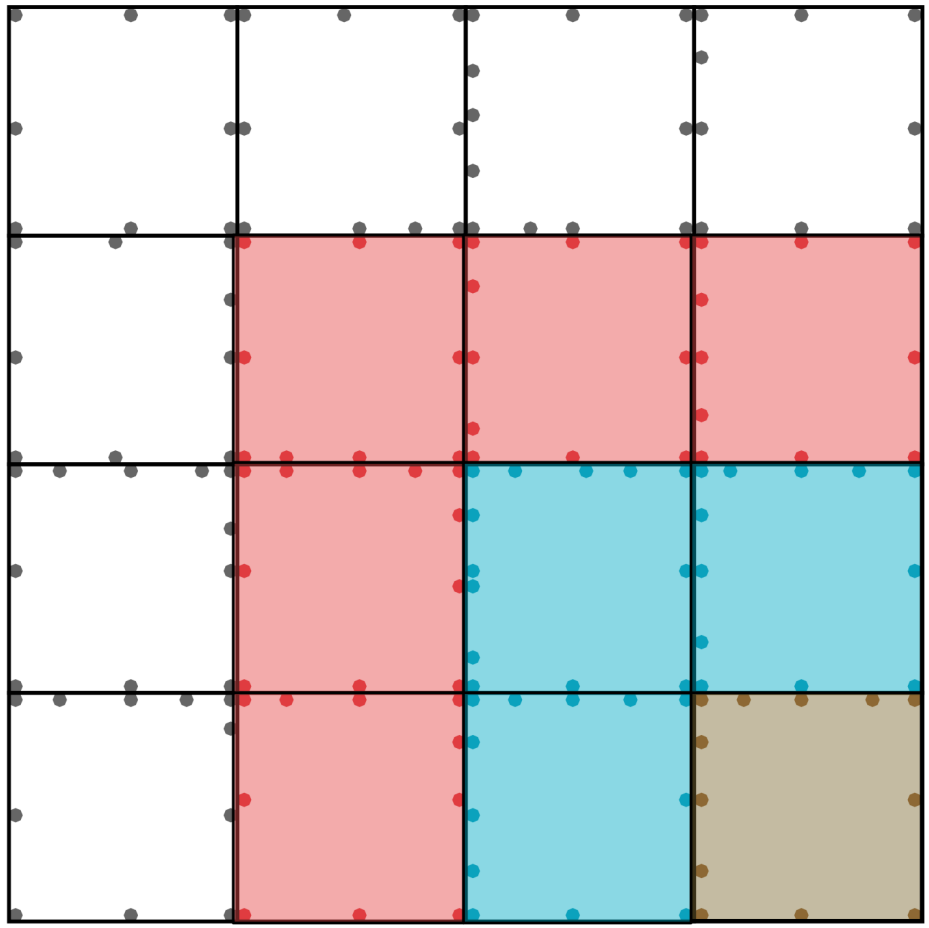
Domain:



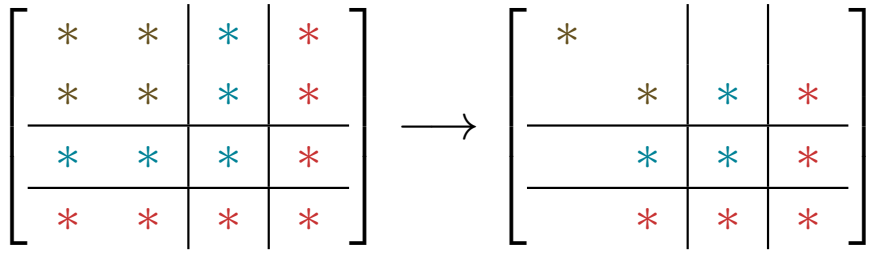
2D cartoon

Level 2: ... and so on

Domain:



Remarks



Complexity sketch:

- Only local operations for each box using proxy trick
- Well-separated rank $|s| \simeq O(1) \implies O(N)$ total cost

Uses: generalized FMM, fast direct solver, preconditioner, etc.

Comparison with other methods:

- RS: superlinear cost, e.g., $O(N^{3/2})$ in 2D, $O(N^2)$ in 3D
- HIF: $O(N)$ but slightly awkward "tree breaking"
- IFMM: $O(N)$ but works on extended sparse matrix

2D example

Laplace kernel in 2D with $N = 2048^2$ at $\epsilon = 10^{-6}$:

$$-\frac{1}{2\pi}\int_{[0,1]^2} \log\|x-y\|\, u(y)\, dy = f(x)$$

level	#boxes	$\sum\lvert b\rvert \rightarrow \sum\lvert s\rvert$		avg. $\lvert b\rvert \rightarrow$ avg. $\lvert s\rvert$	
1	65536	4194304	1378912	64.00	21.04
2	16384	1378912	423076	84.16	25.82
3	4096	423076	120792	103.29	29.49
4	1024	120792	32509	117.96	31.75
5	256	32509	8413	126.99	32.86
6	64	8413	2046	131.45	31.97
7	16	2046	446	127.88	27.88
8	4	446	232	111.50	58.00
9	1	232	0	232.00	0.00

2D example

Laplace kernel in 2D with $N = 2048^2$ at $\epsilon = 10^{-6}$:

$$-\frac{1}{2\pi}\int_{[0,1]^2} \log\|x-y\|\, u(y)\, dy = f(x)$$

level	#boxes	$\sum\|b\|$	$\rightarrow$	$\sum\|s\|$	avg. $\|b\|$	$\rightarrow$	avg. $\|s\|$	$\simeq O(1)$
1	65536	4194304		1378912	64.00		21.04	
2	16384	1378912		423076	84.16		25.82	
3	4096	423076		120792	103.29		29.49	
4	1024	120792		32509	117.96		31.75	
5	256	32509		8413	126.99		32.86	
6	64	8413		2046	131.45		31.97	
7	16	2046		446	127.88		27.88	
8	4	446		232	111.50		58.00	
9	1	232		0	232.00		0.00	

2D example

Laplace kernel in 2D with $N = 2048^2$ at $\epsilon = 10^{-6}$:

$$-\frac{1}{2\pi}\int_{[0,1]^2} \log\|x-y\|\, u(y)\, dy = f(x)$$

level	RSF	HIF	FMF
1	47.92	47.92	21.04
2	99.51	41.27	25.82
3	187.87	62.70	29.49
4	345.63	85.81	31.75
5	663.88	108.35	32.86
6	1235.80	129.21	31.97
7	2118.81	139.79	27.88
8	2829.25	113.75	58.00
top level $\lvert b\rvert \longrightarrow$ final	11317	455	232

[Ho/Ying '16, Minden/Ho/Damle/Ying '17]

2D example

Laplace kernel in 2D with $N = 2048^2$ at $\epsilon = 10^{-6}$:

$$-\frac{1}{2\pi}\int_{[0,1]^2} \log\|x-y\|\, u(y)\, dy = f(x)$$

	level	RSF	HIF	FMF
	1	47.92	47.92	21.04
	2	99.51	41.27	25.82
	3	187.87	62.70	29.49
	4	345.63	85.81	31.75
	5	663.88	108.35	32.86
	6	1235.80	129.21	31.97
	7	2118.81	139.79	27.88
	8	2829.25	113.75	58.00
top level $\lvert b\rvert \longrightarrow$	final	11317	455	232

$$O(2^\ell) \to O(N^{3/2})$$

[Ho/Ying '16, Minden/Ho/Damle/Ying '17]

2D example

Laplace kernel in 2D with $N = 2048^2$ at $\epsilon = 10^{-6}$:

$$-\frac{1}{2\pi}\int_{[0,1]^2} \log\|x-y\|\, u(y)\, dy = f(x)$$

level	RSF	HIF	FMF
1	47.92	47.92	21.04
2	99.51	41.27	25.82
3	187.87	62.70	29.49
4	345.63	85.81	31.75
5	663.88	108.35	32.86
6	1235.80	129.21	31.97
7	2118.81	139.79	27.88
8	2829.25	113.75	58.00
top level $\lvert b\rvert \longrightarrow$ final	11317	455	232
		$O(\ell) \to O(N)$	

[Ho/Ying '16, Minden/Ho/Damle/Ying '17]

3D example

Laplace kernel in 3D with $N = 128^3$ at $\epsilon = 10^{-3}$:

$$\frac{1}{4\pi} \int_{[0,1]^3} \frac{1}{\|x - y\|} u(y)\, dy = f(x)$$

level	#boxes	$\sum \lvert b \rvert$	$\rightarrow$	$\sum \lvert s \rvert$	avg. $\lvert b \rvert$	$\rightarrow$ avg. $\lvert s \rvert$
1	32768	2097152		902380	64.00	27.54
2	4096	902380		175249	220.31	42.79
3	512	175249		29513	342.28	57.64
4	64	29513		3642	461.14	56.91
5	8	3642		2133	455.25	266.62
6	1	2133		0	2133.00	0.00

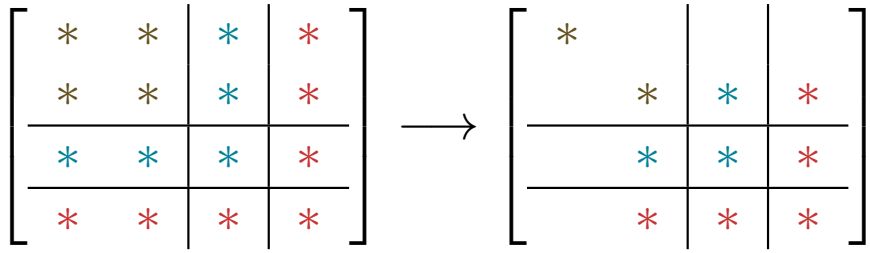
3D example

Laplace kernel in 3D with $N = 64^3$ at $\epsilon = 10^{-3}$:

$$\frac{1}{4\pi} \int_{[0,1]^3} \frac{1}{\|x - y\|} u(y)\, dy = f(x)$$

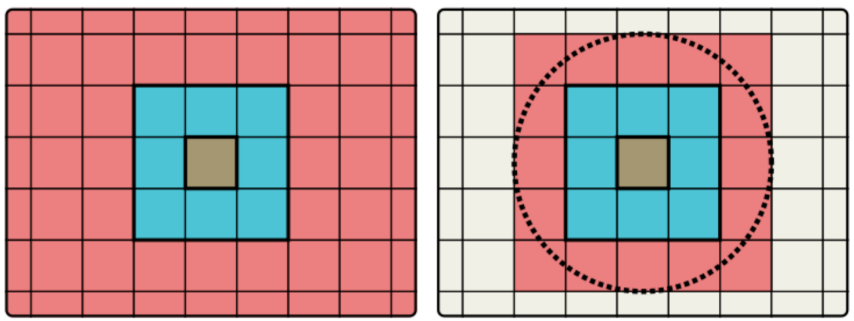
level	RSF	HIF	FMF
1	64.00	64.00	27.24
2	273.26	75.05	40.31
3	1061.45	157.31	47.05
4	3000.62	328.33	228.25
final	24005	1970	1826
	$O(N^2)$	$O(N \log N)$	$O(N)$

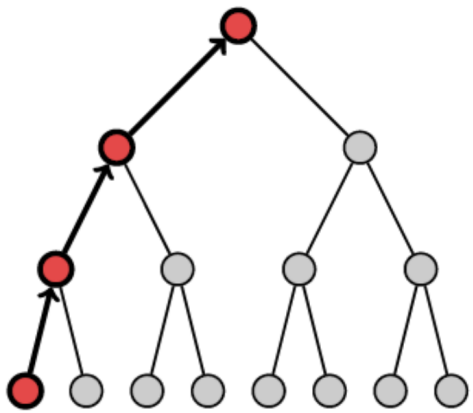
Extensions



Generalized LU decomposition $A \approx U_1^{-*} \cdots U_L^{-*} D V_L^{-1} \cdots V_1^{-1}$ like all RSF-type methods

- Fast matvec, solve, preconditioner
- Cholesky square root, determinant
- Local updating, sparse multiply/solve, trace, selected inversion
- Parallelization





[Ho/Ying '16; Minden/Damle/Ho/Ying '16, '17; Minden '17; Xia/Xi/Cauley/Balakrishnan '15; Li/Ying '17]

FMM factorization/RS-S:

- $O(N)$ factorization of structured matrices following FMM
- Simple multiplicative formulation, extremely flexible

Opinion: general framework done, now engineering/problem-specific optimizations

- Sparse matrices (PDEs) [Sushnikova/Oseledets '16]
- Analytic/pre-select skeletons, symmetries [Corona/Martinsson/Zorin '15]
- Partial factorization with iteration [Yu/March/Biros '17]

Future directions:

- Preconditioning: 1D [Darve/Xia/Chow, ...], HIF-DE [Feliu-Fabà/Ho/Ying]
- Least squares [Ho/Greengard '14, Liu/Barnett/Ho]
- High dimensions, no explicit geometry [March/Yu/Biros et. al. '16, '17]
- Global updating [Xi/Xia/Chan '14, Greengard/Ho/Lee '14], SVD, matrix functions
- High-frequency: multidirectional FMM, butterfly

Some references from us

- Minden/Ho/Damle/Ying, A recursive skeletonization factorization based on strong admissibility, MMS 2017
- Ho/Greengard, A fast direct solver for structured linear systems by recursive skeletonization, SISC 2012
- Ho/Greengard, A fast semidirect least squares algorithm for hierarchically block separable matrices, SIMAX 2014
- Greengard/Ho/Lee, A fast direct solver for scattering from periodic structures with multiple material interfaces in two dimensions, JCP 2014
- Ho/Ying, Hierarchical interpolative factorization for elliptic operators: integral equations, CPAM 2016
- Ho/Ying, Hierarchical interpolative factorization for elliptic operators: differential equations, CPAM 2016
- Minden/Damle/Ho/Ying, A technique for updating hierarchical skeletonization-based factorizations of integral operators, MMS 2016
- Minden/Damle/Ho/Ying, Fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations, MMS 2017
- Li/Ying, Distributed-memory hierarchical interpolative factorization, Res. Math. Sci., 2017
- Ho, Fast direct methods for molecular electrostatics, Ph.D. thesis, NYU, 2012
- Minden, Data-sparse algorithms for structured matrices, Ph.D. thesis, Stanford, 2017s
- Li/Yang/Martin/Ho/Ying, Butterfly factorization, MMS 2015
- Ho, FLAM: fast linear algebra in MATLAB, doi:10.5281/zenodo.1253581