Fast algorithms exploiting low-rank structure for graph clustering and integral equations

Victor Minden, Stanford ICME

Joint work with Anil Damle, Ken L. Ho, & Lexing Ying

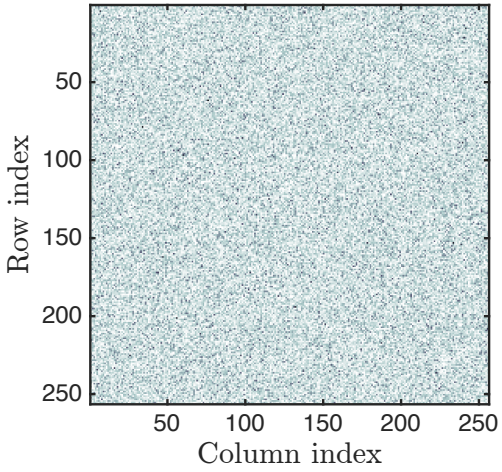
December, 2016

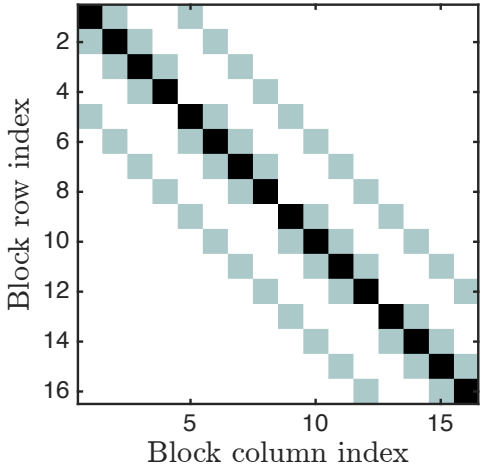
Introduction

- My work: fast algorithms exploiting sparsity/related structure

- This talk
  - Part 1: a simple linear algebraic algorithm for graph clustering

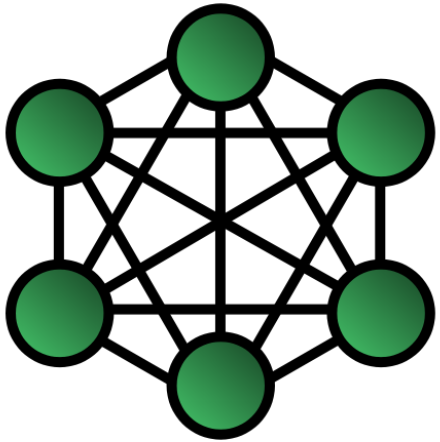
  - Part 2: a rank-structured factorization for the inverse operator of fast multipole method systems (discretizations of integral equations)





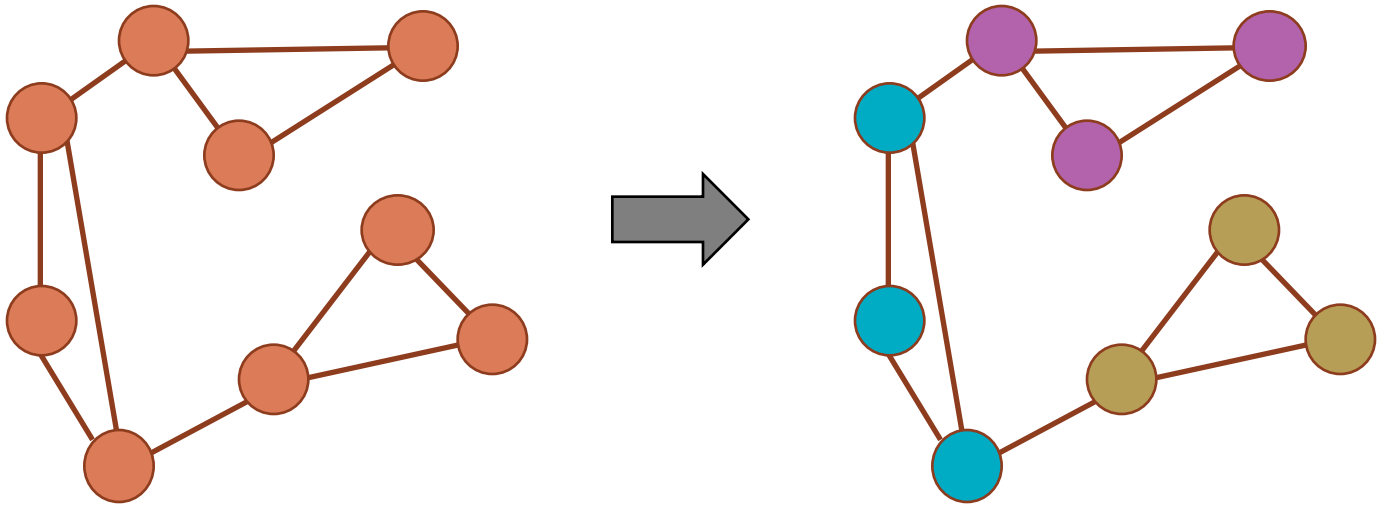
Robust and efficient multi-way spectral clustering

WITH DAMLE, YING



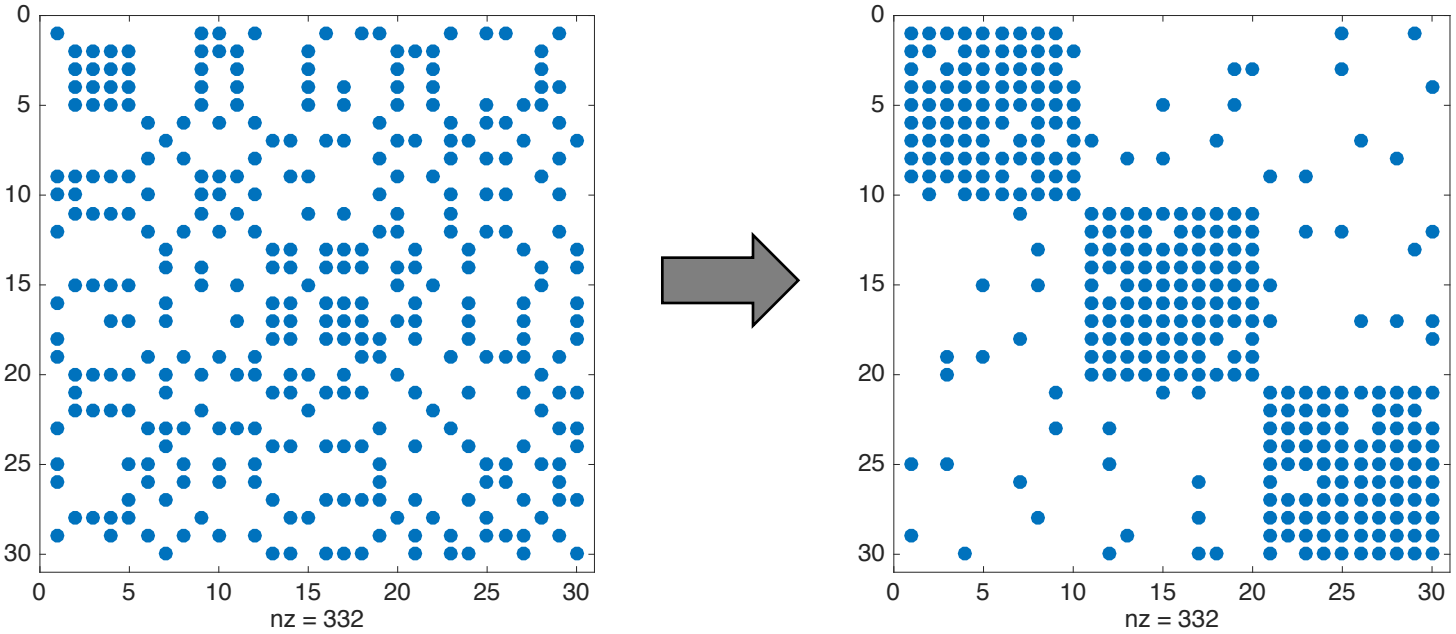
Graph clustering

Task: find *k* relatively well-connected communities of nodes in a simple graph



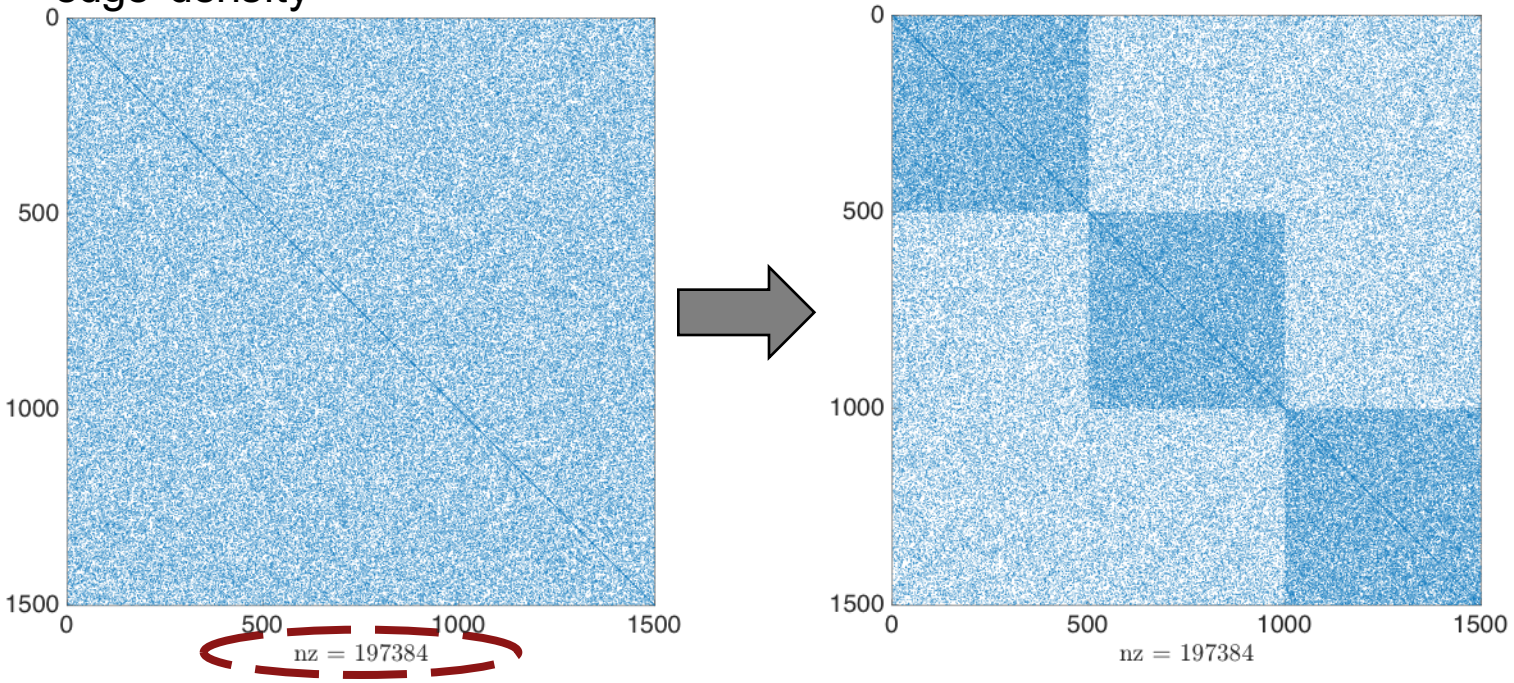
Graph clustering

Sometimes this is easy, and adjacency matrix is essentially block diagonal under some permutation



Graph clustering

Sometimes "within-cluster" edge density barely different from "between-cluster" edge density



8% sparsity

Basics

- For a graph G with n vertices, the (symmetric) adjacency matrix of G is $A \in \{0,1\}^{n \times n}$ with

$$A_{ij} = \begin{cases} 1 & \text{edge } (i,j) \text{ is in } G \\ 0 & \text{else.} \end{cases}$$

- The stochastic block model (SBM) [Holland et al., 1983]
  - Entries of the symmetric adjacency matrix A are drawn independently from a Bernoulli distribution, where (for our purposes)

$$\mathbb{P}[A_{ij} = 1] = \begin{cases} p & i \text{ and } j \text{ in same cluster,} \\ q & \text{else.} \end{cases}$$

The SBM mean adjacency matrix (k=3 clusters)

$$\mathbb{E}[A] = M = \Pi \begin{pmatrix} p & p & q & q & q & q \\ p & p & q & q & q & q \\ q & q & p & p & q & q \\ q & q & p & p & q & q \\ q & q & q & q & p & p \\ q & q & q & q & p & p \end{pmatrix} \Pi^T$$

$$= \Pi \begin{pmatrix} 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \\ & & 1 \end{pmatrix} \begin{pmatrix} p & q & q \\ q & p & q \\ q & q & p \end{pmatrix} \begin{pmatrix} 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \\ & & 1 \end{pmatrix}^{\mathrm{T}} \Pi^T = \Pi X \begin{pmatrix} p & q & q \\ q & p & q \\ q & q & p \end{pmatrix} X^T \Pi^T$$

Relation to eigendecomposition of M

Using $\begin{pmatrix} p & q & q \\ q & p & q \\ q & q & p \end{pmatrix} = Q\Lambda Q^T$, we see eigenvectors are a rotation of indicators:

$$\mathbb{E}[A] = M = \Pi X \begin{pmatrix} p & q & q \\ q & p & q \\ q & q & p \end{pmatrix} X^T \Pi^T = (\Pi X Q)\Lambda(Q^T X^T \Pi^T) = U_k \Lambda' U_k^T$$

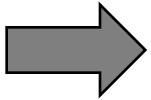
$$U_k = \Pi X Q$$

$\Pi X \in \mathbb{R}^{n \times k}$ is a collection of indicator vectors
$Q \in \mathbb{R}^{k \times k}$ is a rotation

Another way to think about it

- Relax the maximum likelihood estimator for clusters

$$\begin{aligned} &\text{maximize} && \text{Trace}(X^T A X) \\ &\text{subject to} && X^T X = \frac{n}{k} I_k \\ & && X_{ij} \text{ binary} \end{aligned}$$



$$\begin{aligned} &\text{maximize} && \text{Trace}(X^T A X) \\ &\text{subject to} && X^T X = \frac{n}{k} I_k \\ & && X_{ij} \text{ real-valued} \end{aligned}$$

- We hope the solution to the relaxed problem is close to the solution to the binary problem.

- But, relaxed solution is determined only *up to rotation*: $X^* = V_k Q$

Our clustering algorithm

```
[Q,R,P] = qr(Vk',0);
[~,c] = max(abs(R*P'));
```

- Given matrix of top-k eigenvectors V_k, basic form has two steps:
  1. Compute a column-pivoted QR factorization of V_k^T

$$V_k^T \Pi = QR$$

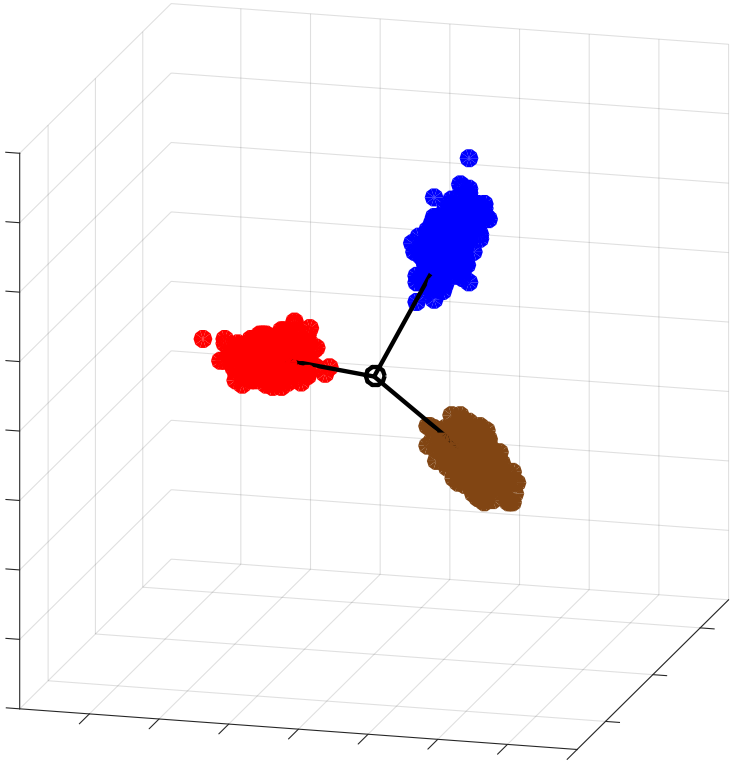
  2. For each j = 1, …, n, assign vertex j to cluster

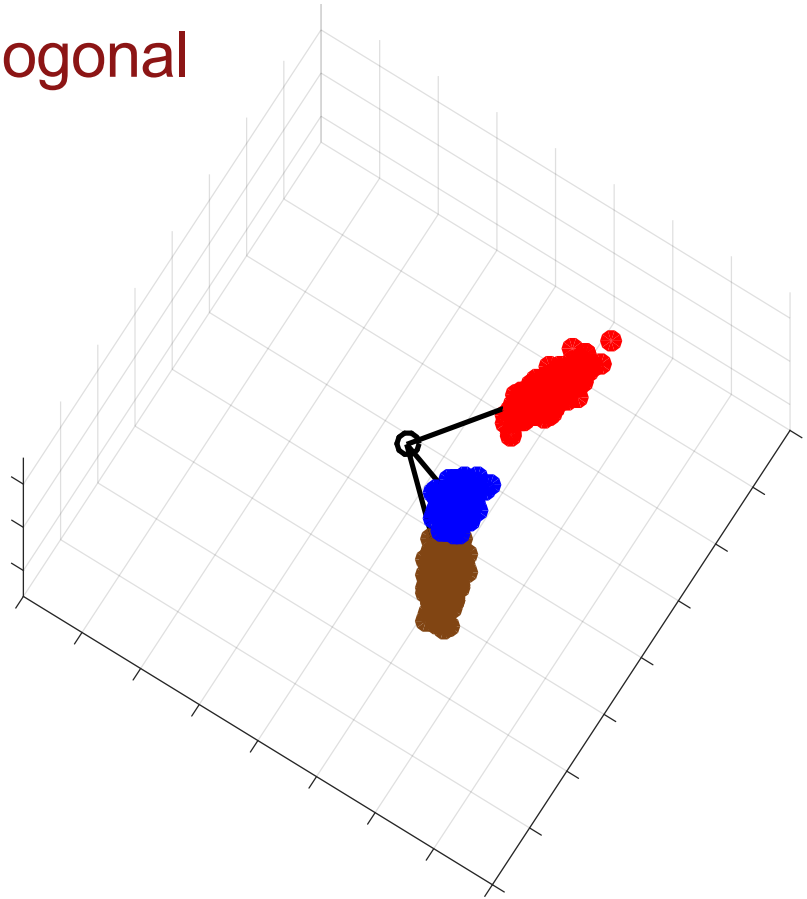
$$c_j = \arg\max_i |R\Pi^T|_{i,j}$$

- Complexity: $\mathcal{O}(nk^2)$

Distinct clusters are almost orthogonal

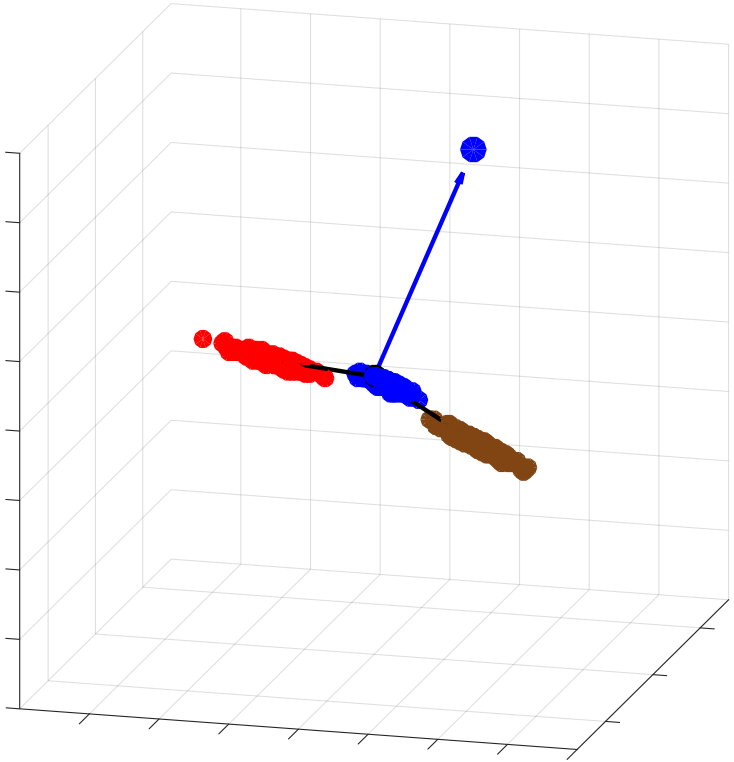
- Eigenvector embedding from A:

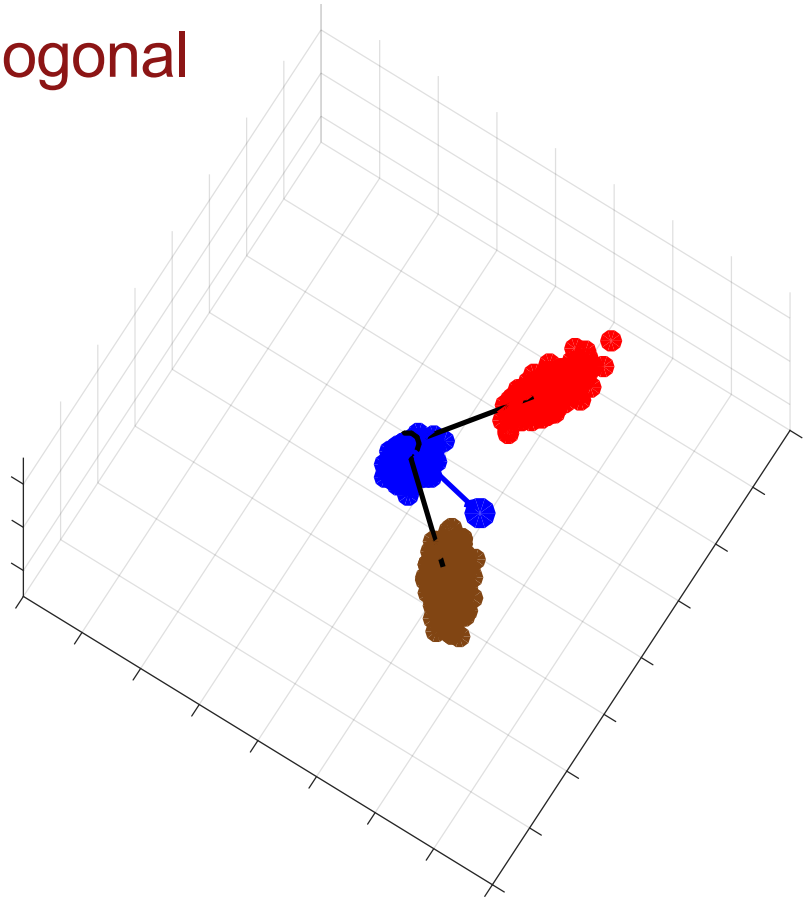




Distinct clusters are almost orthogonal

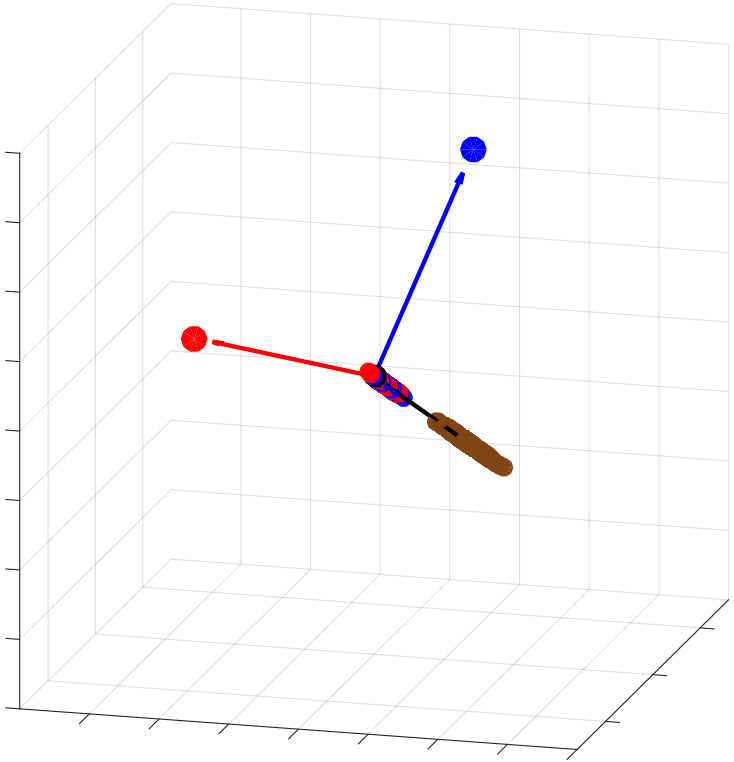
- After step 1 of pivoted QR

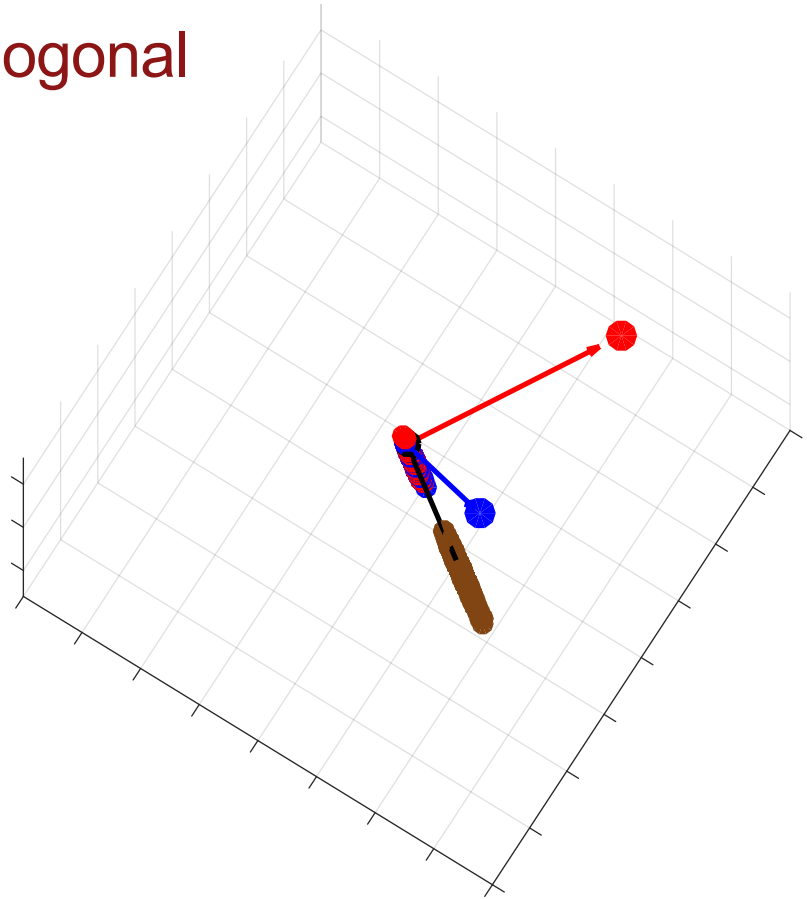




Distinct clusters are almost orthogonal

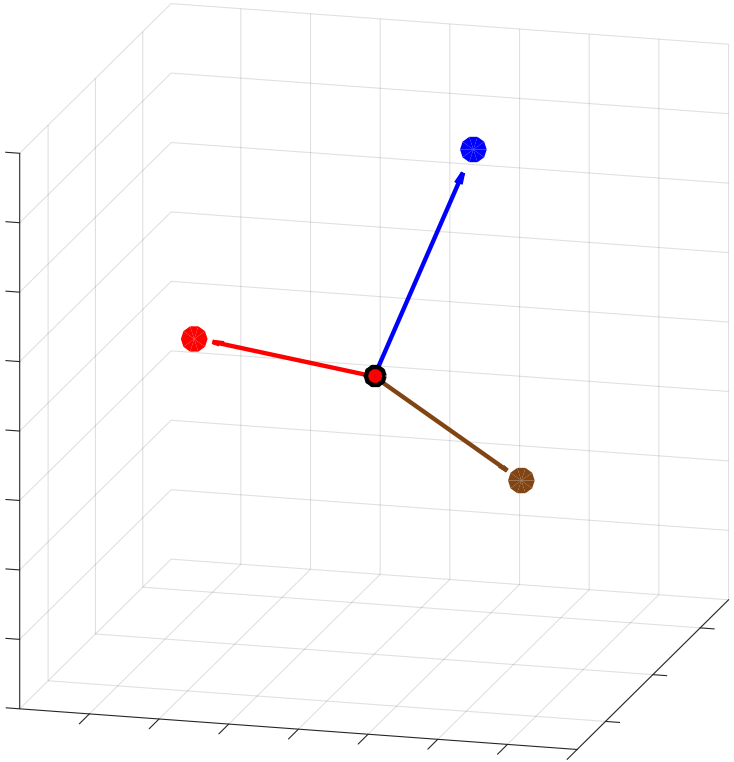
- After step 2 of pivoted QR

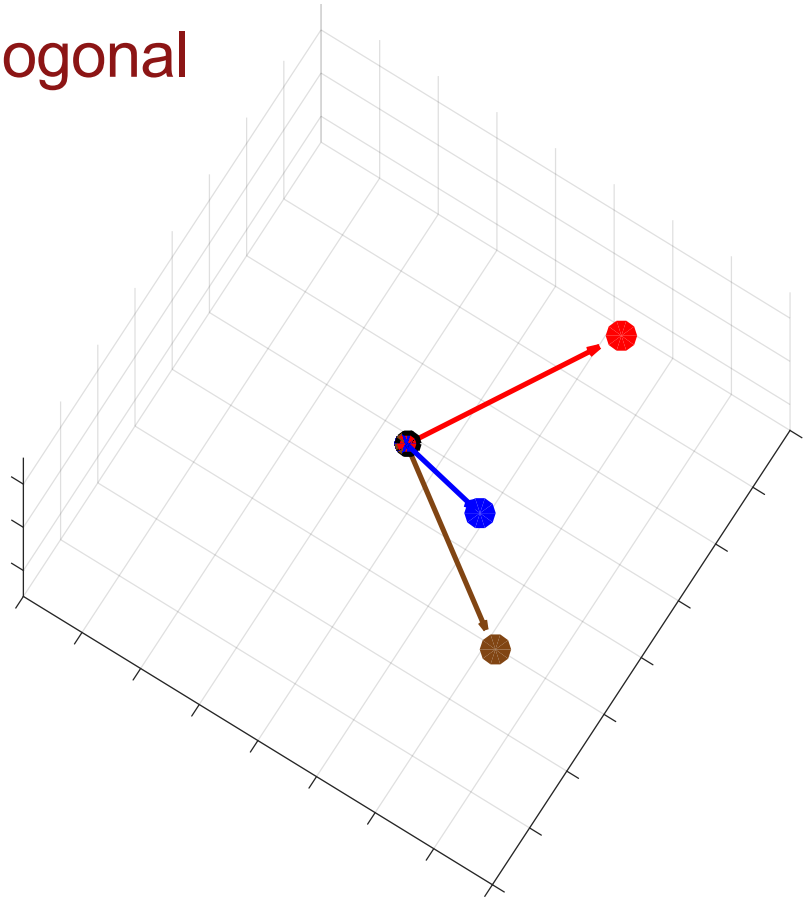




Distinct clusters are almost orthogonal

- After step 3 of pivoted QR





Randomized variant

- Observation: we just need to select a single “good” column from each cluster to get a basis of vectors that are almost indicators.
- Idea: use coupon-collecting argument to subsample the vertices prior to pivoted QR.
- Cost is still $\mathcal{O}(nk^2)$ overall, but a much smaller constant.

Our randomized algorithm

1. Sample $\mathcal{O}(k \log k)$ vertices and denote this set $\mathcal{J}$, letting $W_k = V_k(\mathcal{J}, :)$
2. Compute a column-pivoted QR factorization of W_k^T

$$W_k^T \Pi = QR$$

3. For each j = 1, …, n, assign vertex j to cluster

$$c_j = \arg\max_{\mathrm{i}} \left| Q^T V_k^T \right|_{i,j}$$

- Complexity: $\mathcal{O}(nk) + \mathcal{O}(k^3 \log k) + \mathcal{O}(nk^2)$

When does it work?

IF eigenvectors of A are close enough to eigenvectors of M in span

$$\|V_k V_k^T - U_k U_k^T\|_2 < \epsilon = o\left(\frac{1}{\sqrt{n}}\right)$$

THEN our (randomized) algorithm recovers the clusters for sufficiently large *n*

Theorem 1. *Let $W \in \mathbf{R}^{n\times k}$ be a normalized indicator matrix and suppose $n > 4$. Let $V \in \mathbf{R}^{n\times k}$ with orthonormal columns satisfy*

$$\left\|WW^T - VV^T\right\|_2 \leq \epsilon$$

with $\epsilon < 2^{-k}/\sqrt{n}$. If $V^T\Pi = QR$ is a CPQR factorization of V^T, then there exists a permutation matrix $\widehat{\Pi}$ and a diagonal matrix D with $D_{ii} = \pm 1$ such that

$$\|W\widehat{\Pi}D - VQ\|_F \leq \epsilon(1+\sqrt{2})k\sqrt{n} + \mathcal{O}\left(\epsilon^2\right).$$

Back to the stochastic block model

- Identifiability of clusters in SBM exhibits a threshold behavior
  - See [Abbe et al., 2016], [Agarwal et al., 2015], and [Hajek et al., 2016]

- Parameterize:

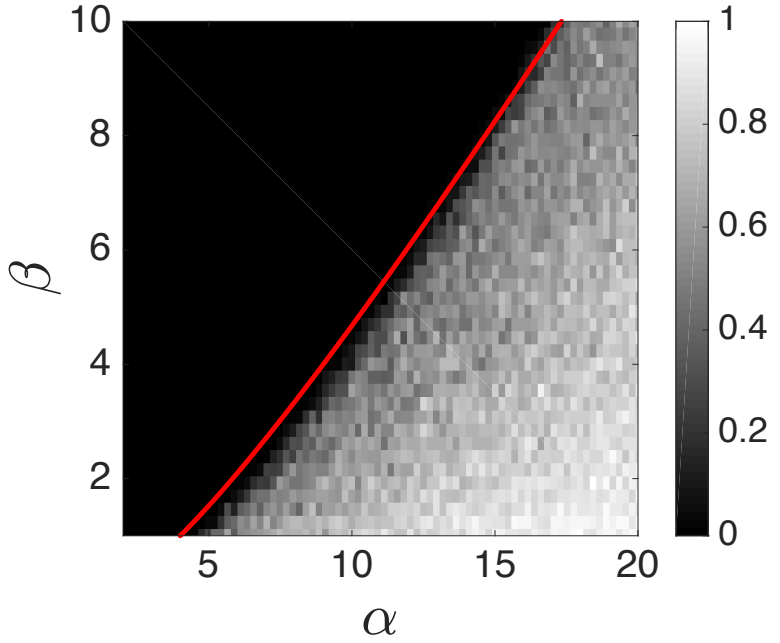
$$\mathbb{P}[A_{ij} = 1] = \begin{cases} \alpha \log m \,/m & i \text{ and } j \text{ in same cluster,} \\ \beta \log m \,/m & \text{else.} \end{cases}$$

- Result: asymptotically in number of nodes, recovery is possible with high probability if and only if

$$\sqrt{\alpha} - \sqrt{\beta} > 1$$

So, what's the problem?

- Spectral clustering with `k-means++` on eigenvector embedding
  - Bad recovery results (right)
  - Color is empirical recovery probability
  - Need a good initialization to avoid local minima!

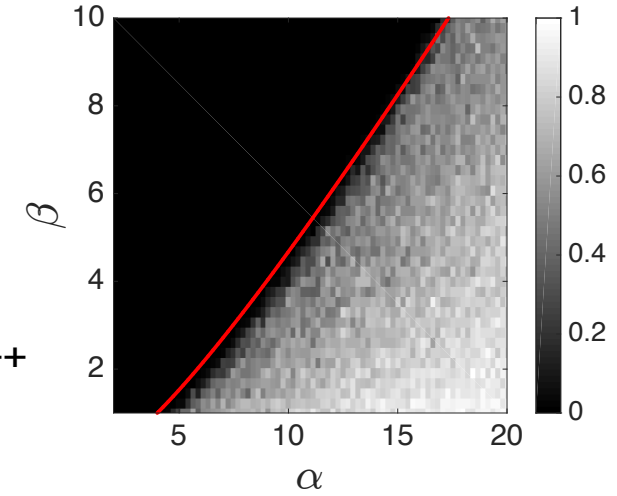


- A semidefinite program over *n*-by-*n* semidefinite matrices attains asymptotically optimal recovery
  - [Abbe et al., 2016], [Agarwal et al., 2015], and [Hajek et al., 2016]
  - Too expensive for large *n*

Results: equisized clusters

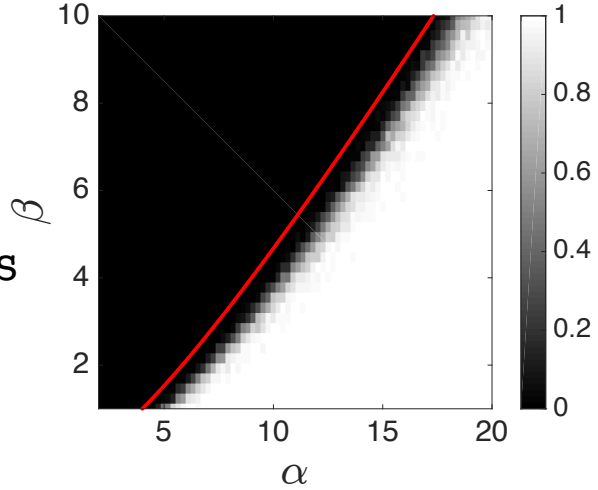
- $\mathrm{n} = 1350$ nodes
- $k = 9$ clusters

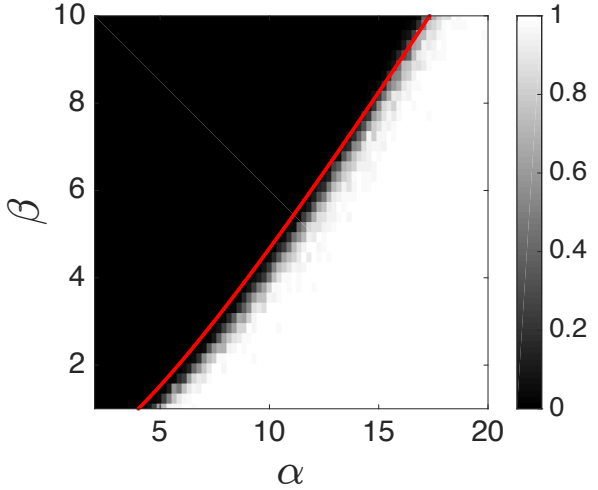
Right: k-means++



Left: QR

Right: QR + k-means

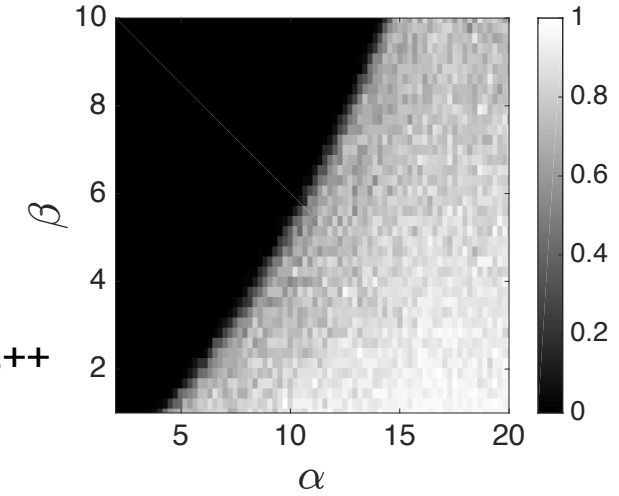




Results: unequisized clusters

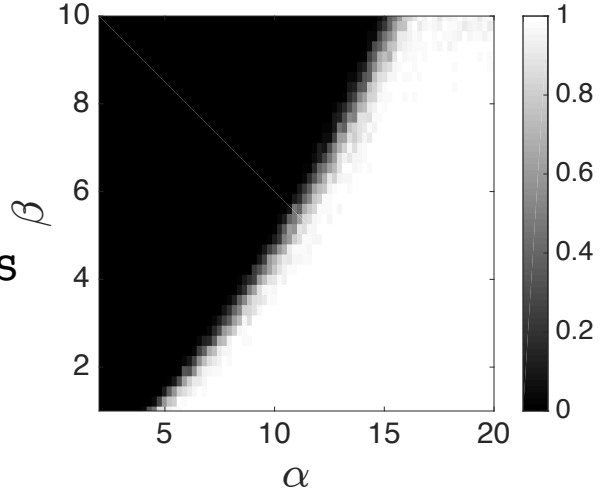
- $m \in [70,130]$ nodes/cluster
- $k = 7$ clusters

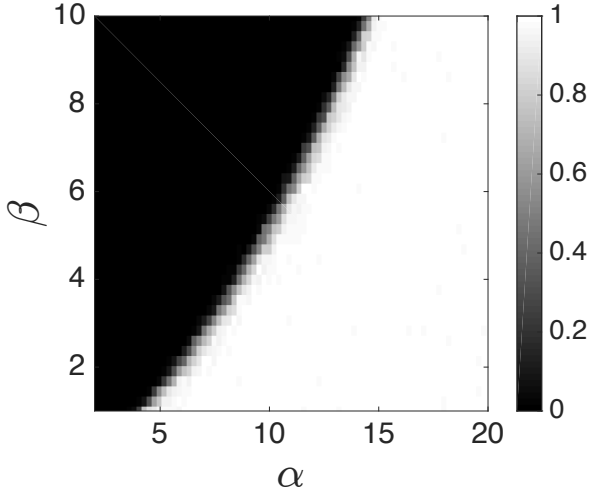
Recall: k-means++



Left: QR

Right: QR + k-means





Related work for cluster recovery of sparse SBM

- [Zha et al., 2001]: a similar pivoted QR based algorithm for general data
  - Different cluster assignment scheme
  - Not in the context of graphs
  - Deterministic only

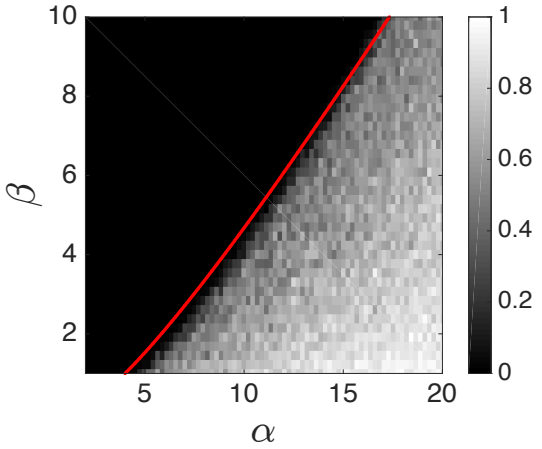
- [Bopanna, 1987] and [McSherry, 2001]: more complicated spectral-based algorithms, with provable recovery guarantees but not at the limit

- [Mossel et al., 2016]: spectral algorithm plus post-processing for $k=2$, with recovery guarantees at correct limit

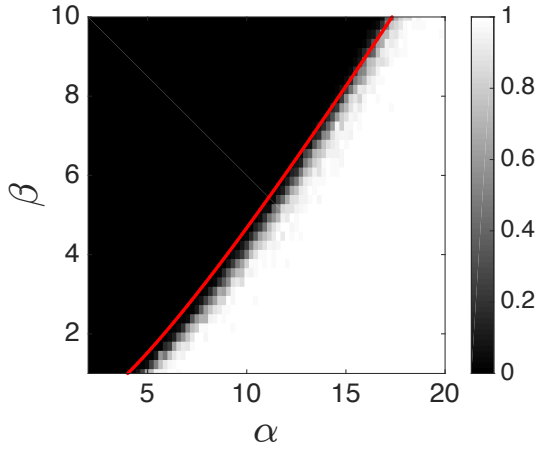
Ongoing and future work

- Can we prove asymptotically optimal recovery with this algorithm (or a variant)?
- Application to optimization problems in computer vision

- Take-away:

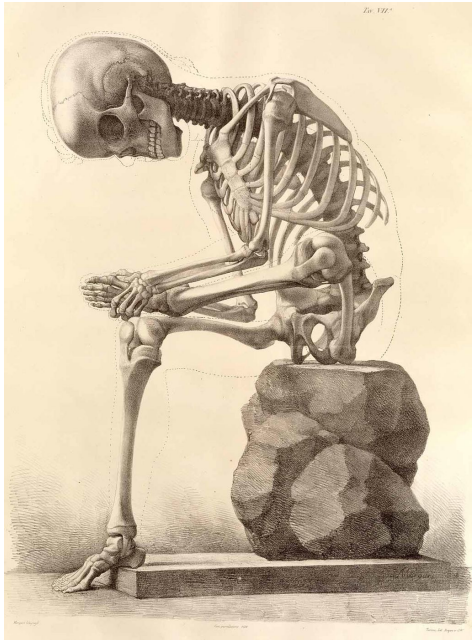



```
[Q,R,P] = qr(Vk',0);
[~,c] = max(abs(R*P'));
```



A recursive skeletonization factorization based on strong admissibility

WITH HO, DAMLE, YING

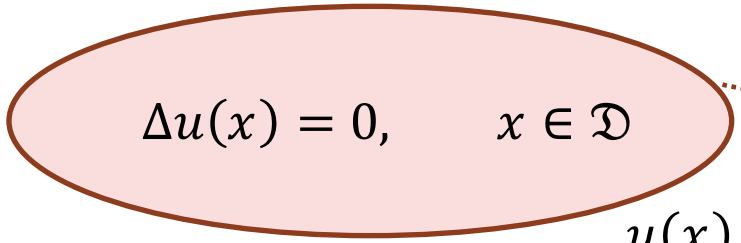


Introduction

- Many elliptic differential equations describing physical phenomena can be recast as integral equations of the form

$$a(x)u(x) + \int_{\Omega} b(x)K(x-y)c(y)u(y)\,dy = f(x), \qquad x \in \Omega$$

- Examples
  - Laplace equation
  - Helmholtz equation
  - Lippmann-Schwinger
  - Stokes equation (vector-valued)
  - Elasticity equations (vector-valued)



Introduction

- Let's simplify:

$$\int_{\Omega} K(x-y)u(y)\,dy = f(x), \qquad x \in \Omega$$

- $K(x-y) = K(r) = \ldots$
  - $1/|r|$
  - $\log(|r|)$
  - A derivative of one of these

$$\Delta u(x) = 0, \qquad x \in \mathfrak{D}$$

$$u(x) = f(x), \quad x \in \partial\mathfrak{D}$$

Discretization

- Discretization using standard methods leads to a many-body sum (assume uniform weights)

$$\sum_j K(x_i - x_j)u_j = f_i, \qquad i = 1, \ldots, N$$

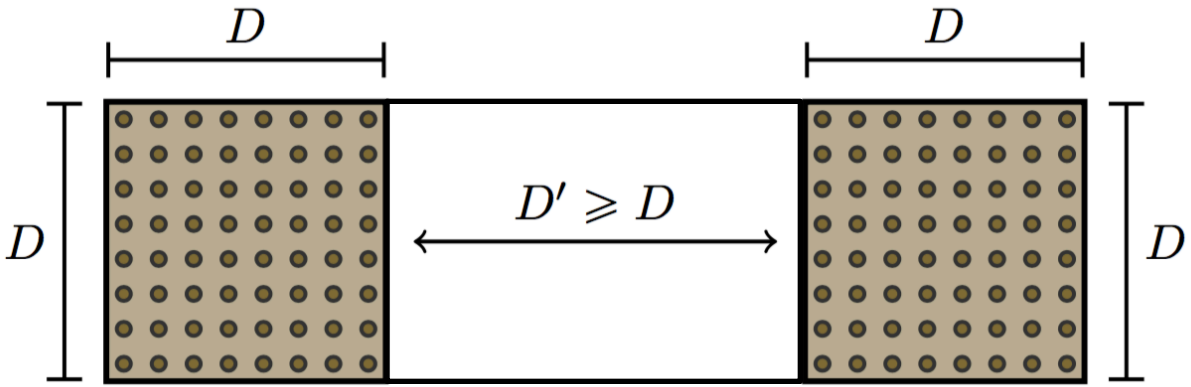
- Naïvely, evaluating the output f given the input u costs $\mathcal{O}(N^2)$

- Tree codes [Barnes & Hut, 1986] or fast multipole method (FMM) [Greengard & Rokhlin, 1987 &1997] use clever approximations to reduce this to $\mathcal{O}(N \log N)$ or $\mathcal{O}(N)$ with a constant depending on accuracy.

Tree codes and FMM

- Basic idea: well-separated interactions are compressible

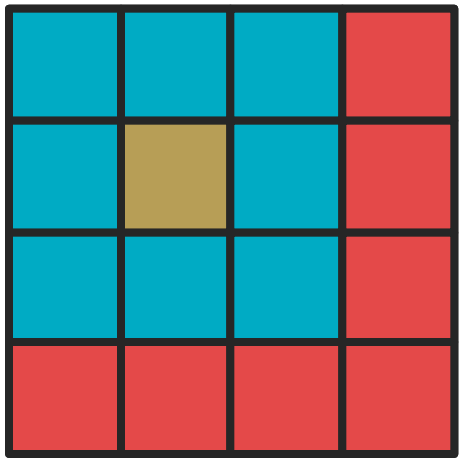
$$\sum_j K(x_i - x_j)u_j = f_i, \qquad i = 1, \ldots, N$$



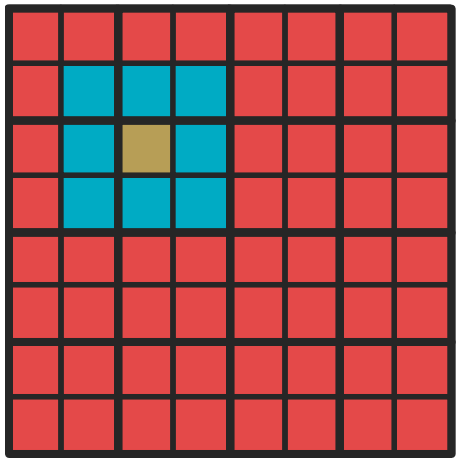
$$\mathbb{K}(\mathcal{B}_1, \mathcal{B}_2) \approx UV^T$$

Tree codes and FMM

- Exploit far-field compression idea in a hierarchical fashion to apply operator quickly
- This is as much as we need to understand tree codes for this talk



One level of tree



Next lower level of tree

Solving systems

- What about the opposite problem, solving $Ku = f$ for u?

$$\sum_j K(x_i - x_j)u_j = f_i, \qquad i = 1, \ldots, N$$

  - Iterative methods (conjugate gradient, etc.) are $\mathcal{O}(N)$ per iteration with FMM, but often require many iterations.

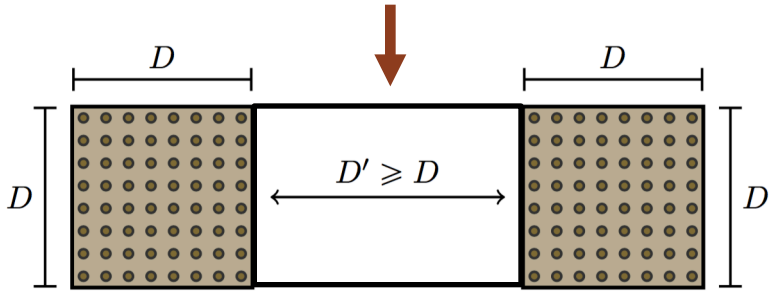
- Large body of work on approximate solvers or preconditioners using hierarchical low-rank structure of this problem, giving better complexity.

Hierarchical representations from skeletonization

- "Recursive skeletonization" and related literature has led to many nice hierarchical factorizations based on "weak admissibility"

- HSS / HBS matrices
  - [Martinsson & Rokhlin, 2005]
  - [Chandrasekaran et al., 2006 & 2007]
  - [Ho & Greengard, 2012]
  - [Xia et al., 2012]
  - [Gillman et al., 2012]
- HODLR matrices
  - [Martinsson, 2008]
  - [Ambikasaran & Darve, 2013]

Separation not enforced
by HSS/HBS/HODLR,
giving less compression



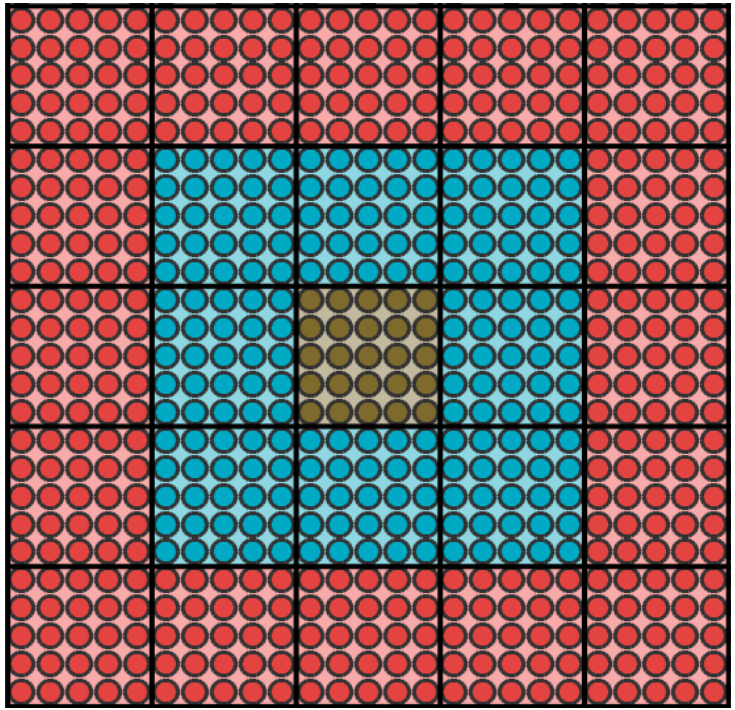
Our approach

- Adapt recursive skeletonization [Martinsson & Rokhlin, 2005] to compress only well-separated interactions using the multiplicative formulation of skeletonization in [Ho & Ying, 2016].

- Result: approximate factorization of K^{-1} as the product of
  - permutation matrices
  - block-unit-triangular matrices (with small blocks)
  - a block-diagonal matrix (with small blocks)

- Simple and fast way to apply K, K^{-1}, $K^{1/2}$, or $K^{-1/2}$, or compute log-det

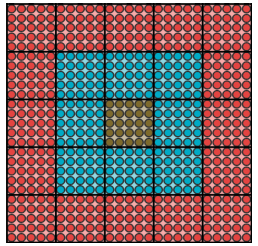
Quadtree decomposition and admissible neighbors



- **Brown**: box b
- **Blue**: near-field neighbors of b
- **Red**: far-field neighbors of b (lots of these)

$$\left[\begin{array}{c|c|c} A_{bb} & A_{bn} & A_{bf} \\ \hline A_{nb} & A_{nn} & A_{nf} \\ \hline A_{fb} & A_{fn} & A_{ff} \end{array}\right]$$

Interpolative decomposition



- By assumption, the interactions between **box b** and **far-field neighbors f** are low-rank.

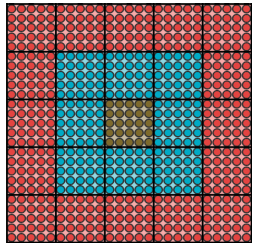
- Compress these with an *interpolative decomposition* [Cheng et al., 2005], where box **b** is partitioned into **"skeleton set" s** and **"redundant set" r**

$$\left[\begin{array}{c|c|c} A_{bb} & A_{bn} & A_{bf} \\ \hline A_{nb} & A_{nn} & A_{nf} \\ \hline A_{fb} & A_{fn} & A_{ff} \end{array}\right]$$

$$b = s \cup r$$

$$A_{fr} \approx A_{fs} T$$

Block sparse elimination

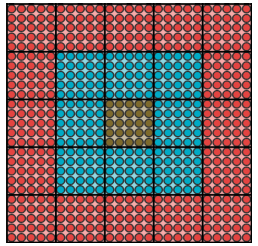


$$\left[\begin{array}{c|c|c} A_{bb} & A_{bn} & A_{bf} \\ \hline A_{nb} & A_{nn} & A_{nf} \\ \hline A_{fb} & A_{fn} & A_{ff} \end{array}\right] = \left[\begin{array}{cc|c|c} A_{rr} & A_{rs} & A_{rn} & A_{rf} \\ A_{sr} & A_{ss} & A_{sn} & A_{sf} \\ \hline A_{nr} & A_{ns} & A_{nn} & A_{nf} \\ \hline A_{fr}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

$$b = s \cup r$$

$$A_{fr} \approx A_{fs} T$$

Block sparse elimination

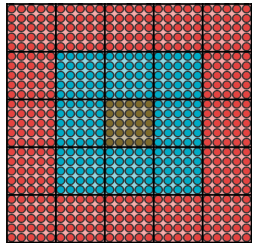


$$\begin{bmatrix} A_{bb} & A_{bn} & A_{bf} \\ A_{nb} & A_{nn} & A_{nf} \\ A_{fb} & A_{fn} & A_{ff} \end{bmatrix} \approx \begin{bmatrix} A_{rr} & A_{rs} & A_{rn} & T^T A_{sf} \\ A_{sr} & A_{ss} & A_{sn} & A_{sf} \\ A_{nr} & A_{ns} & A_{nn} & A_{nf} \\ A_{fs}T & A_{fs} & A_{fn} & A_{ff} \end{bmatrix}$$

$$b = s \cup r$$

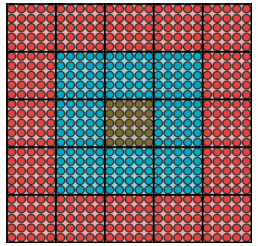
$$A_{fr} \approx A_{fs}T$$

Block sparse elimination



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

Block sparse elimination

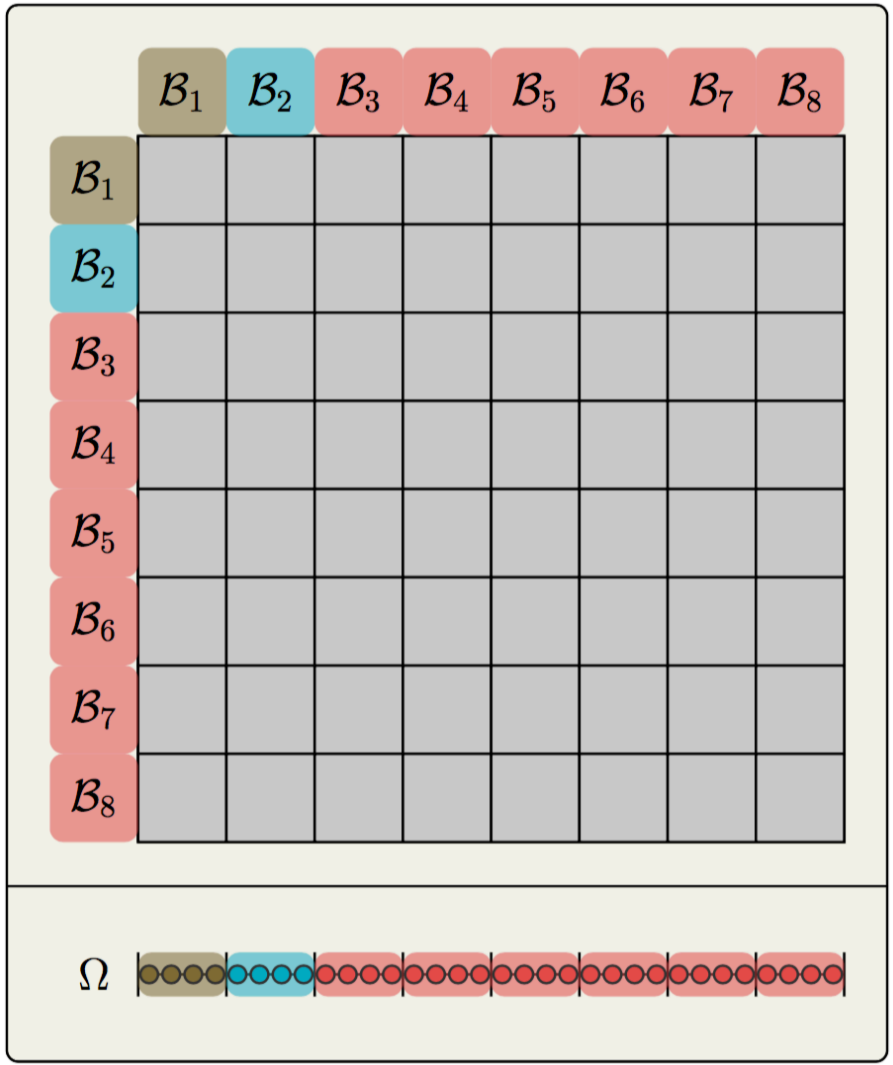


$$\left[\begin{array}{cc|c|c} X_{rr} & X_{rs} & X_{rn} & \\ X_{sr} & A_{ss} & A_{sn} & A_{sf} \\ \hline X_{nr} & A_{ns} & A_{nn} & A_{nf} \\ \hline & A_{fs} & A_{fn} & A_{ff} \end{array}\right] \longrightarrow \left[\begin{array}{c|cc|c} X_{rr} & & & \\ \hline & X_{ss} & X_{sn} & A_{sf} \\ \hline & X_{ns} & X_{nn} & A_{nf} \\ \hline & A_{fs} & A_{fn} & A_{ff} \end{array}\right]$$

After block Gaussian elimination, we have decoupled the redundant points and updated interactions between the skeleton points and near-field neighbors.

Note: every operation was cheap and is easy to undo

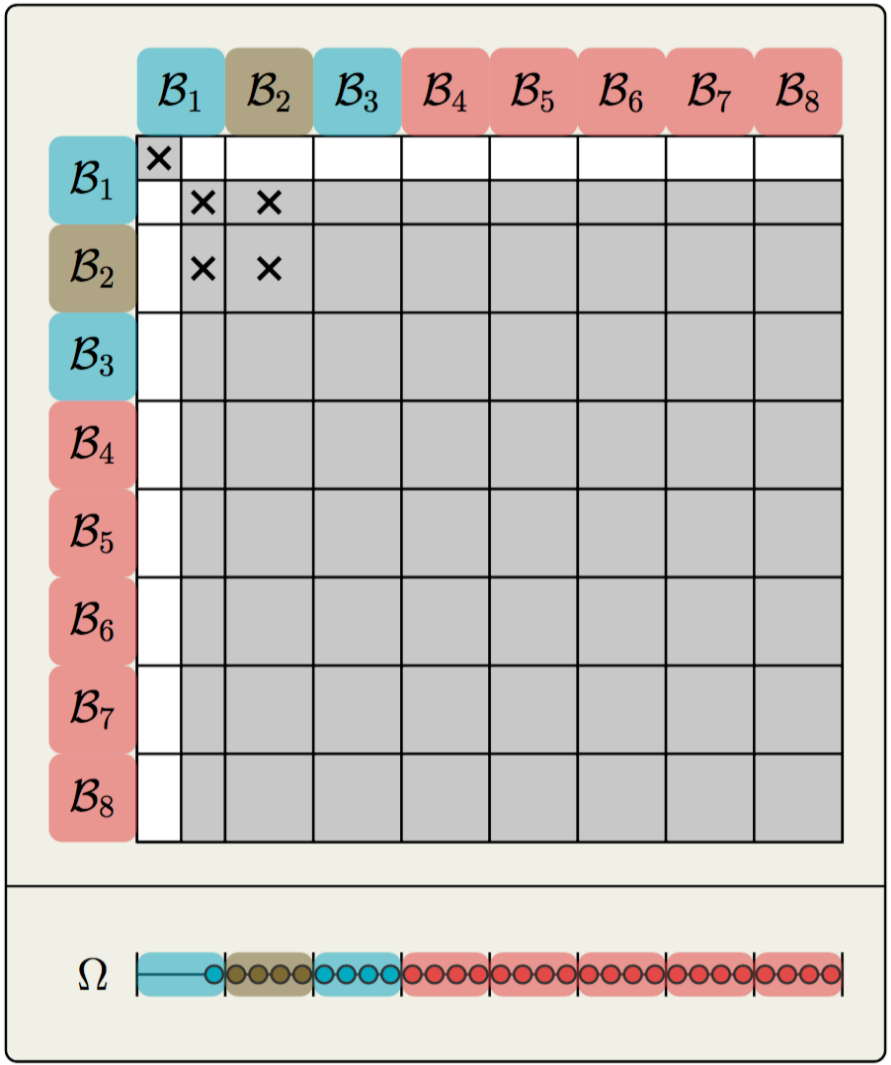
1D
Level 1
Before $\mathcal{B}_1$



The matrix

Points in the domain

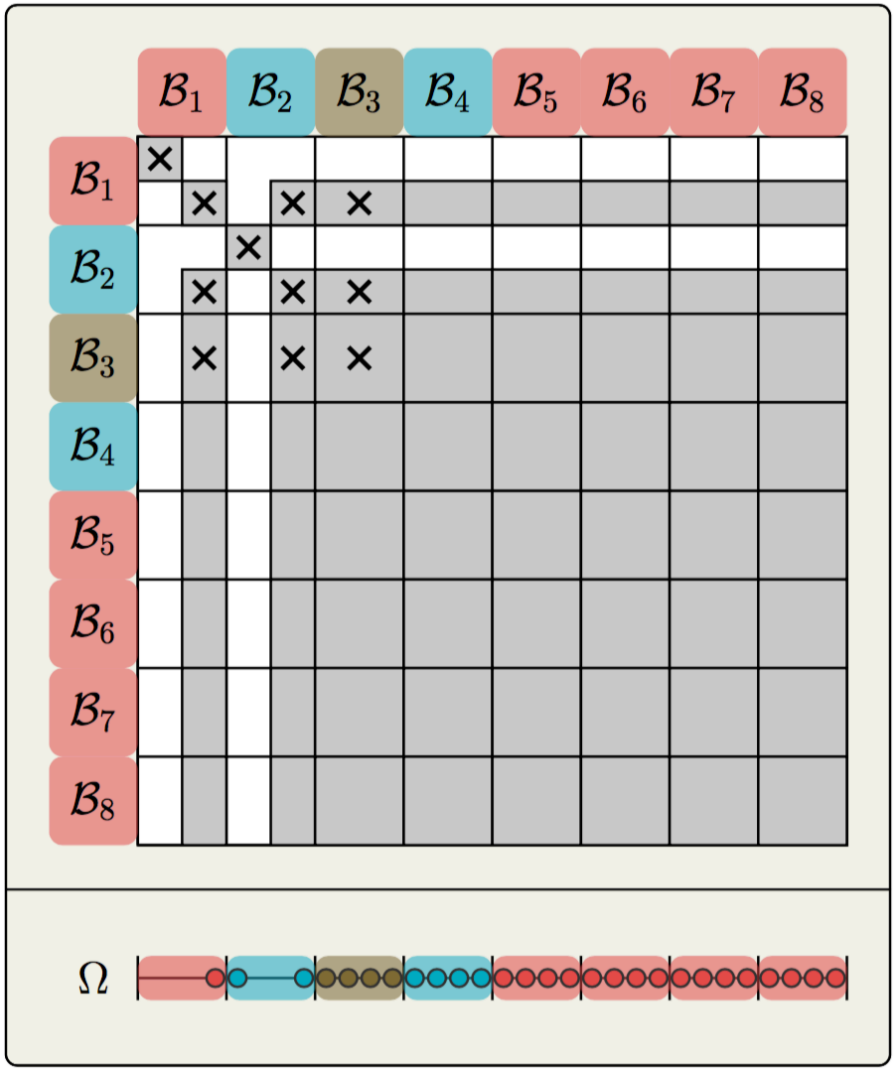
1D
Level 1
Before $\mathcal{B}_2$



The matrix

Points in the domain

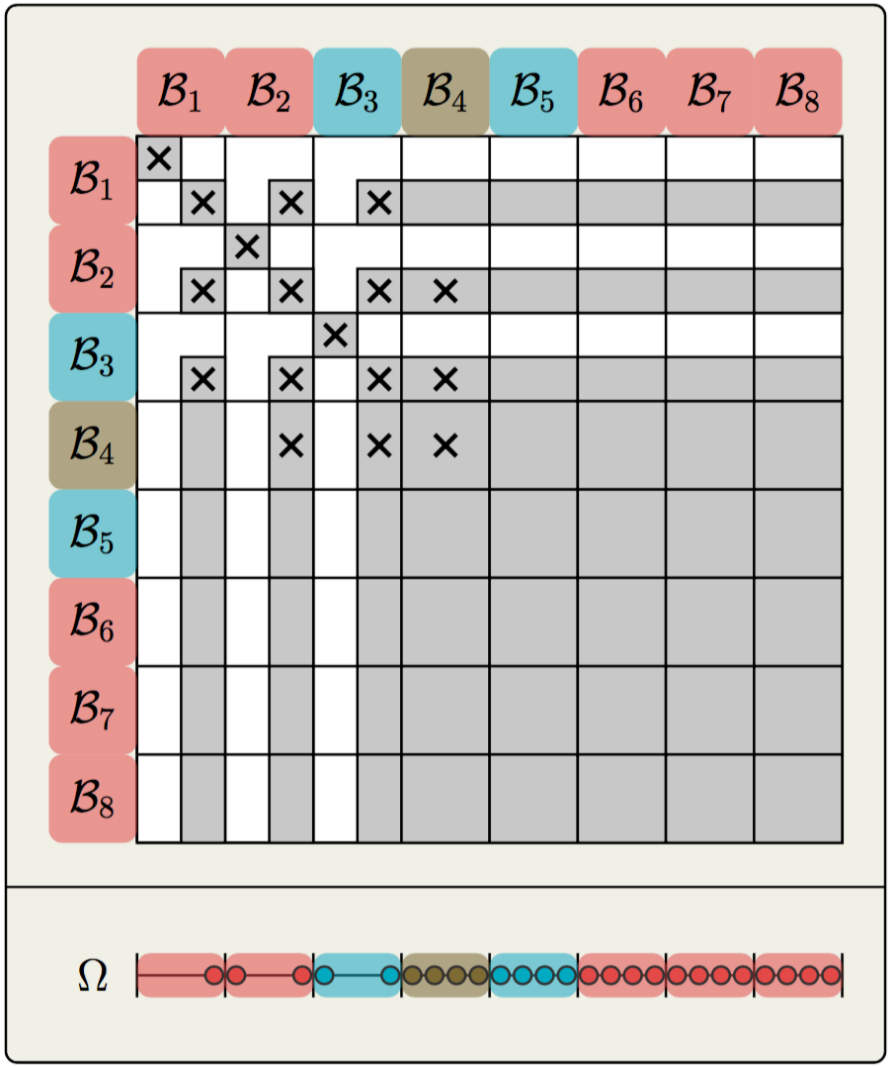
1D
Level 1
Before $\mathcal{B}_3$



The matrix

Points in the domain

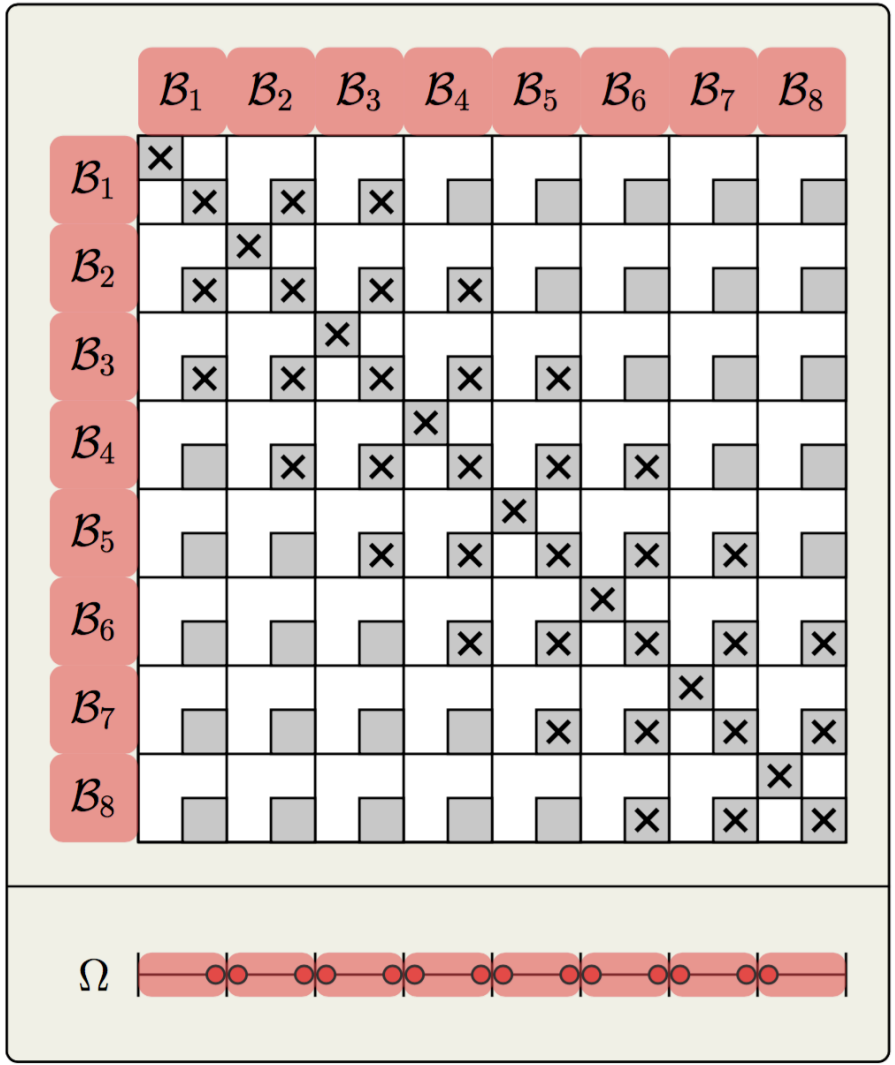
1D
Level 1
Before $\mathcal{B}_4$



The matrix

Points in the domain

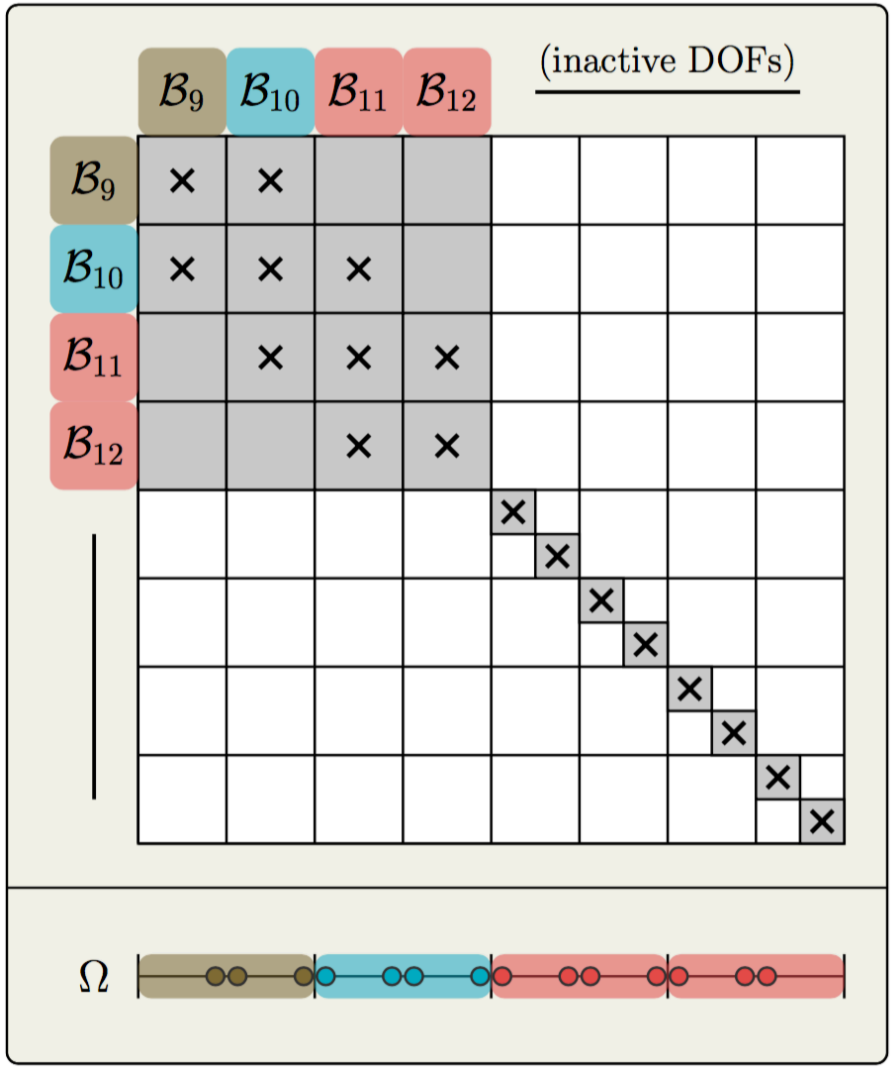
1D
Level 1
After $\mathcal{B}_8$



The matrix

Points in the domain

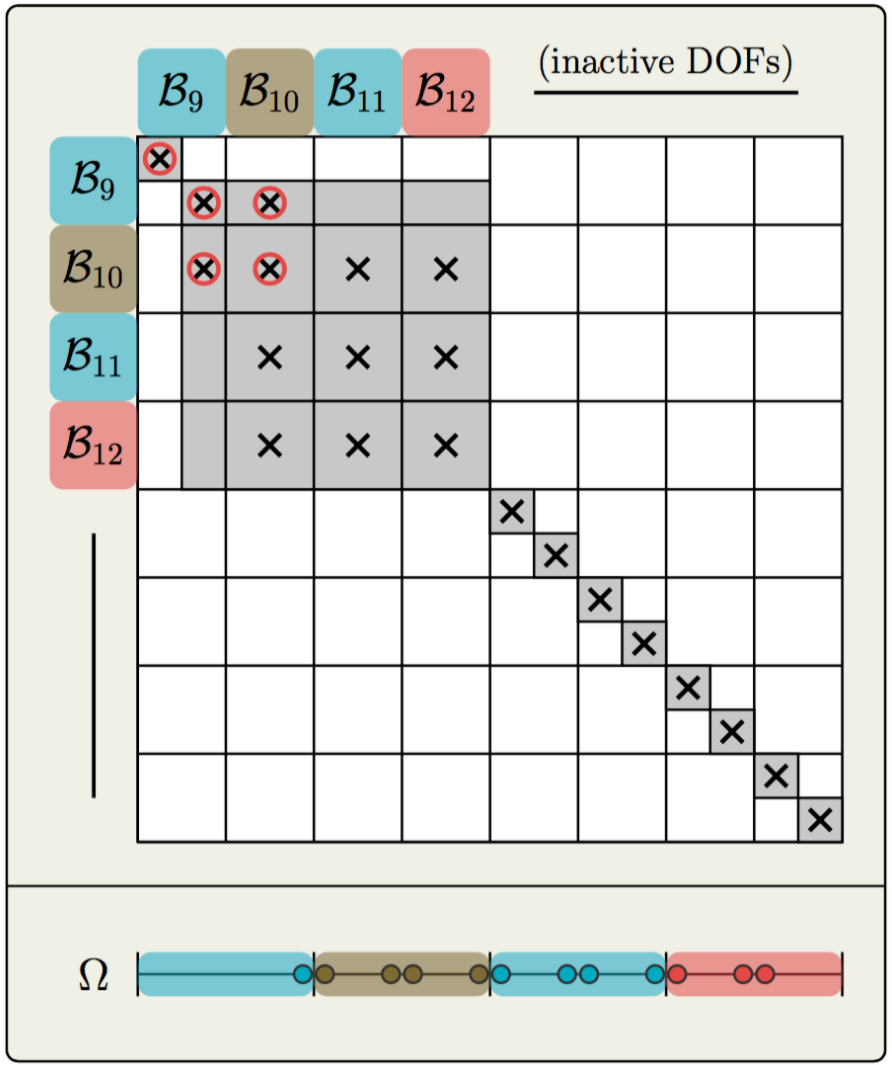
1D
Level 2
Before $\mathcal{B}_9$



← The matrix

Points in the domain →

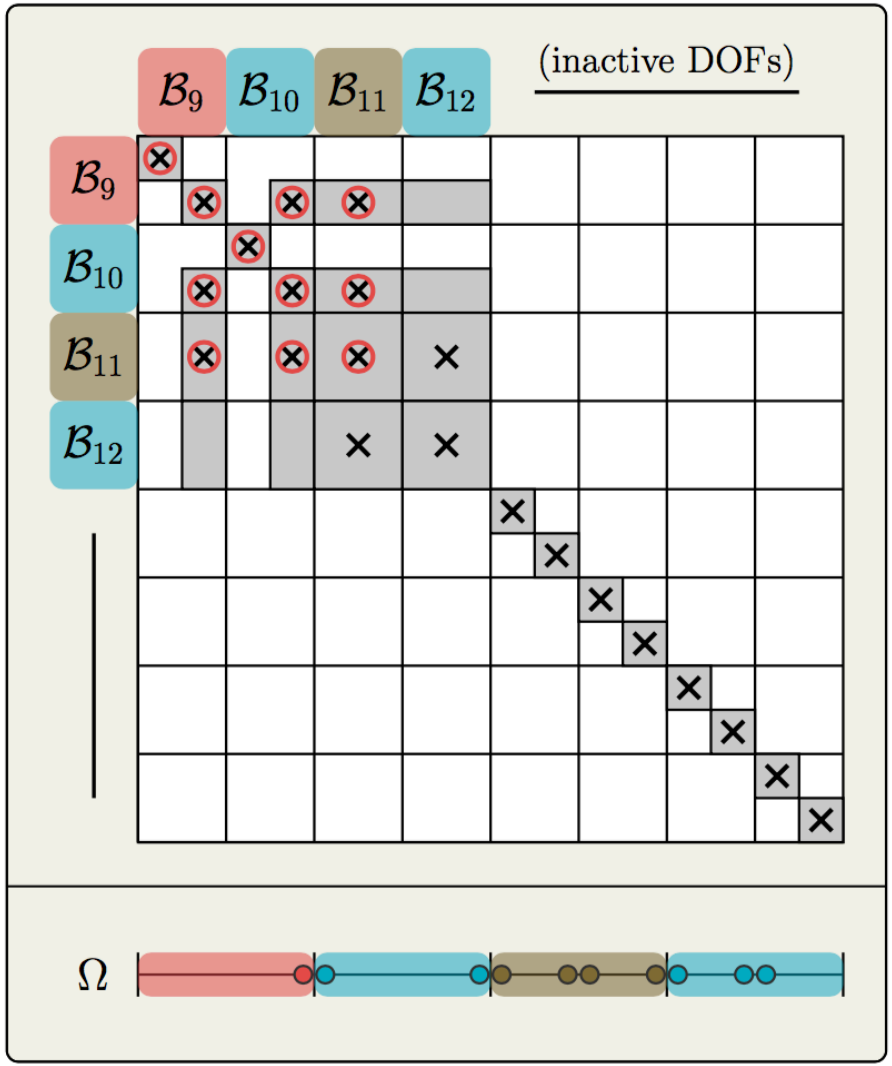
1D
Level 2
Before $\mathcal{B}_{10}$



The matrix

Points in the domain

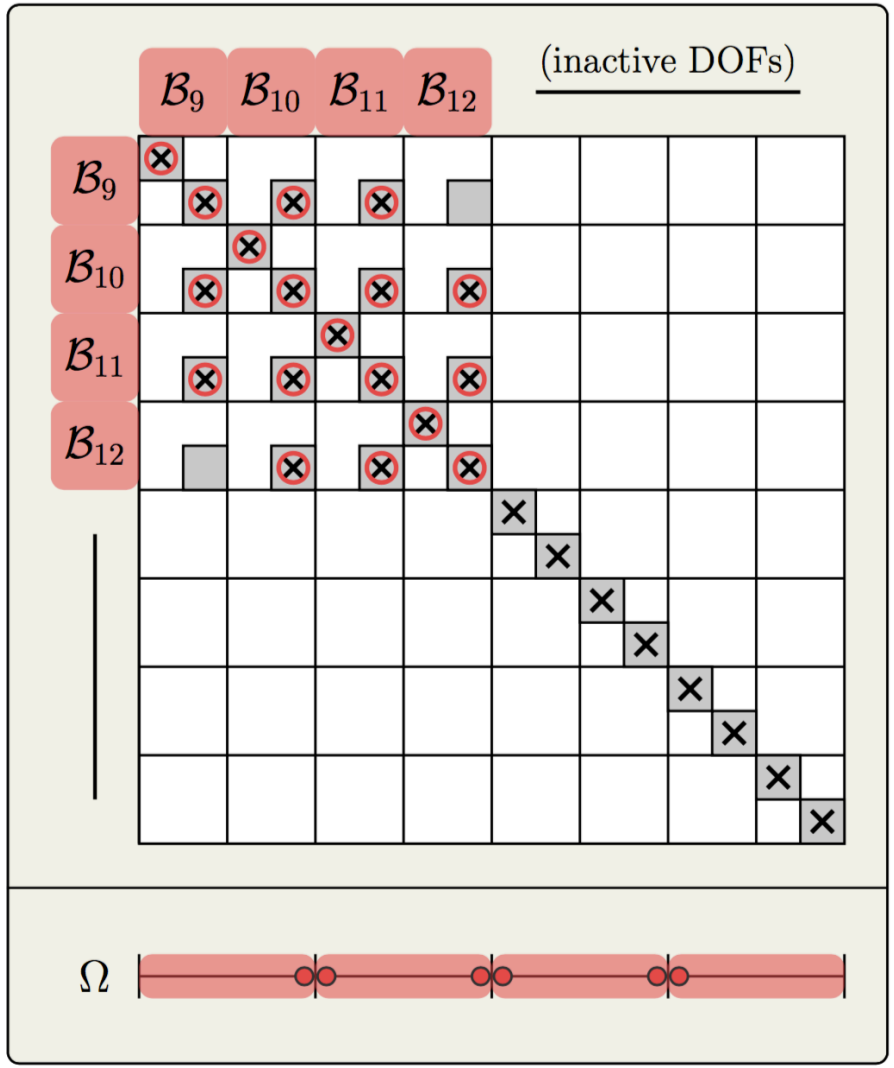
1D
Level 2
Before $\mathcal{B}_{11}$



The matrix

Points in the domain

1D
Level 2
After $\mathcal{B}_{12}$

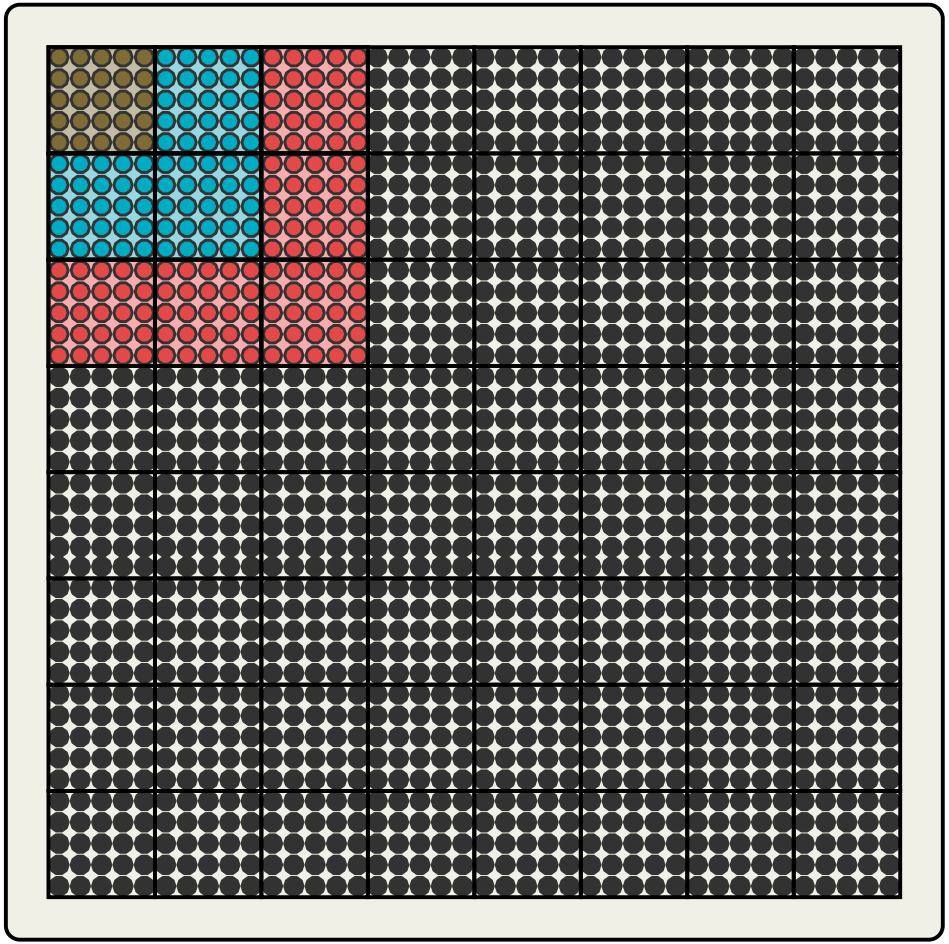


← The matrix

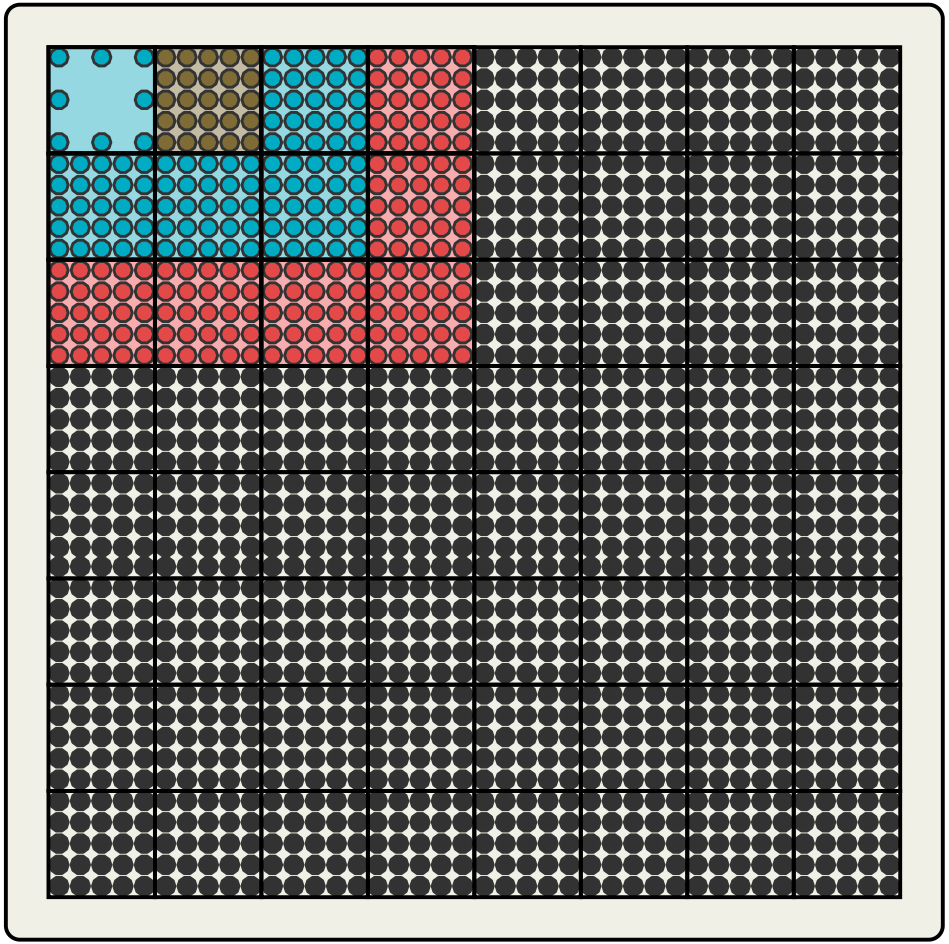
Next: finish $\mathcal{B}_{12}$ and invert (or factor) diagonal blocks

Points in the domain →

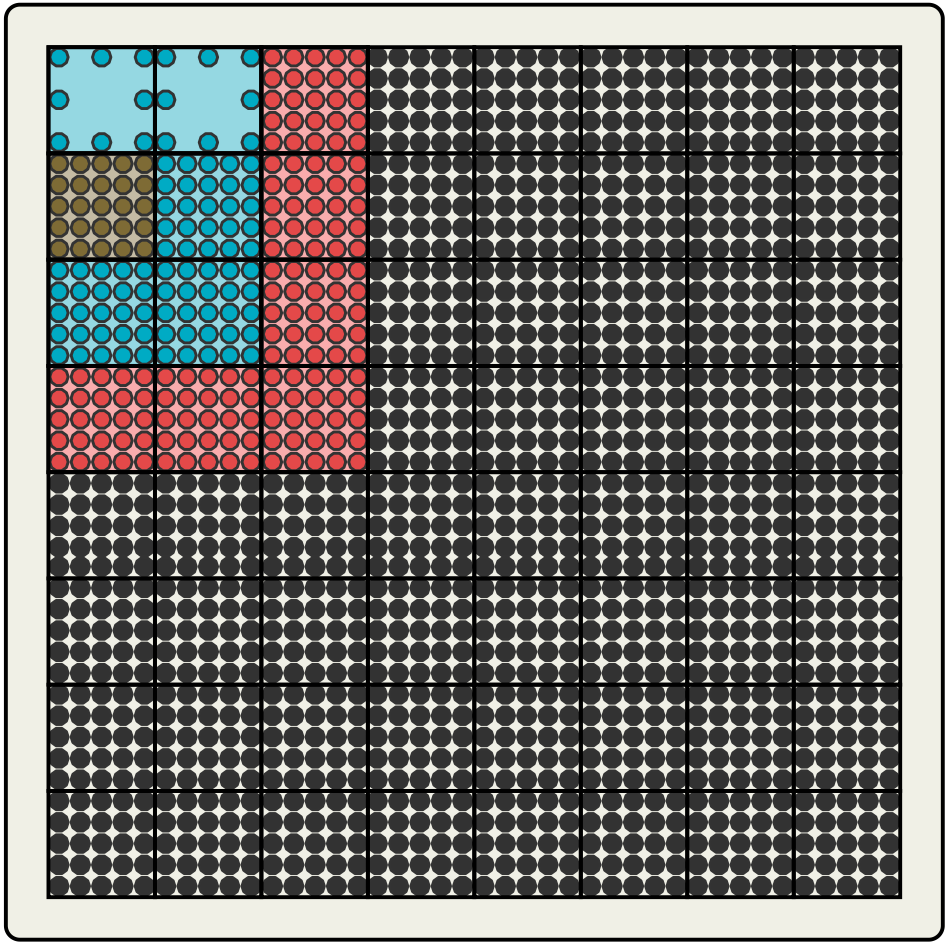
2D
Level 1
Before $\mathcal{B}_1$



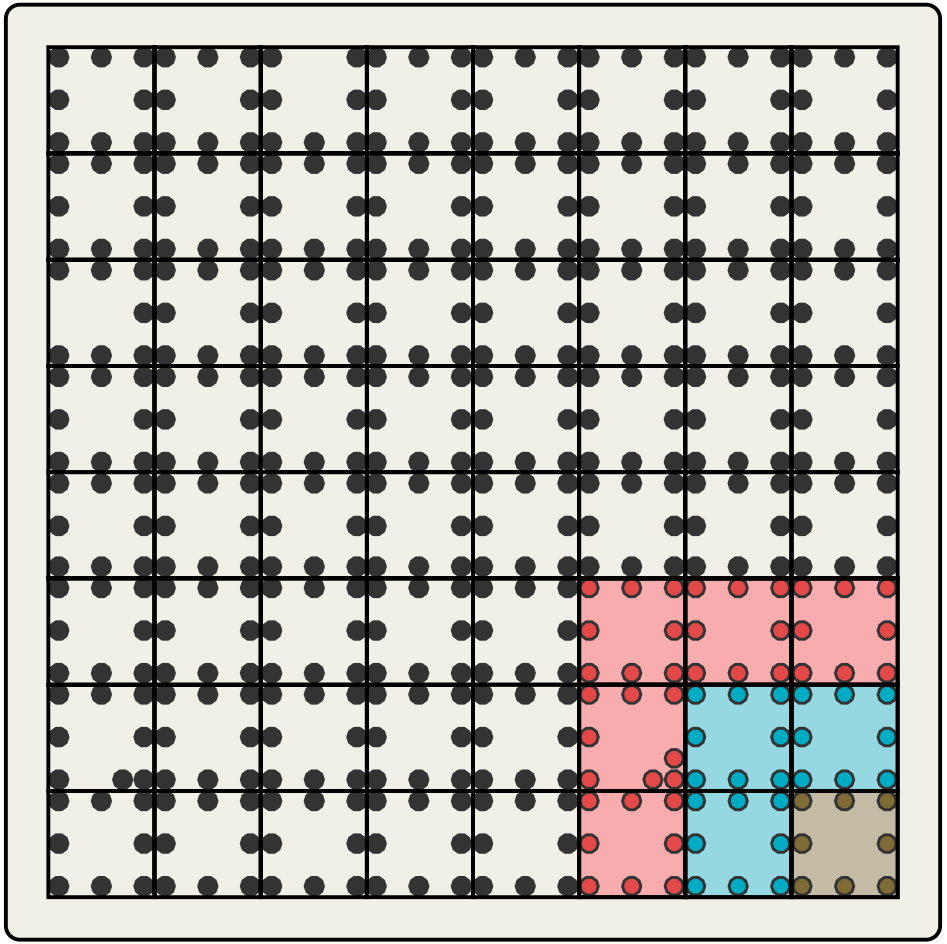
2D
Level 1
Before $\mathcal{B}_2$



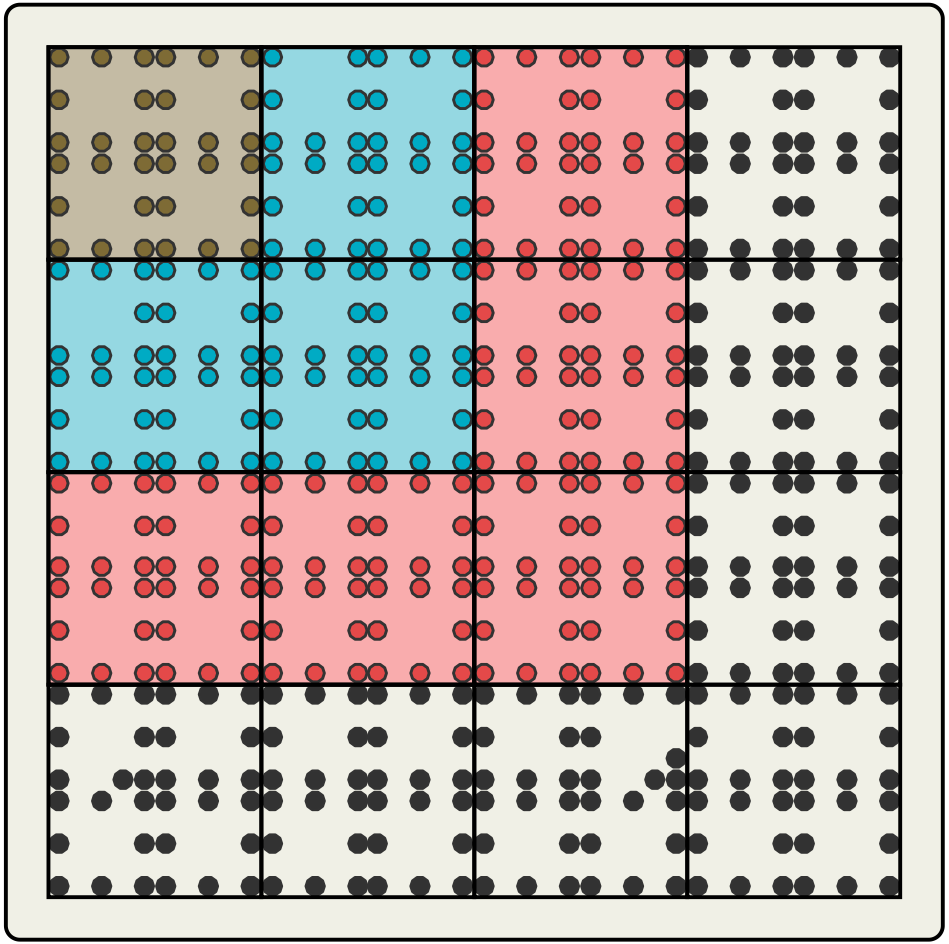
2D
Level 1
Before $\mathcal{B}_3$



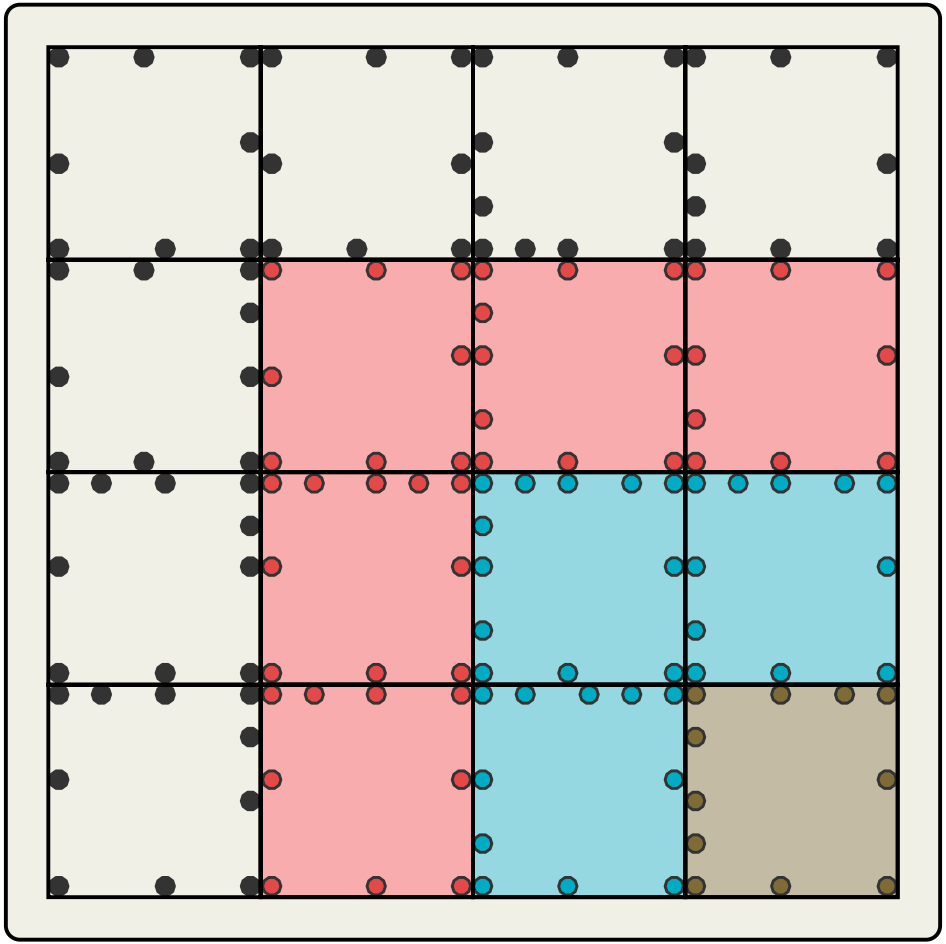
2D
Level 1
After $\mathcal{B}_{64}$



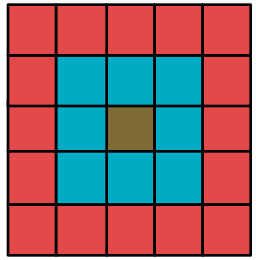
2D
Level 2
Before $\mathcal{B}_{65}$



2D
Level 2
After $\mathcal{B}_{80}$



Complexity sketch



- For each box:
  1. Compute interpolative decomposition
  2. Perform block elimination

$$b = s \cup r$$

$$A_{fr} \approx A_{fs} T$$

- If proxy surface used and ranks do not grow in N:

  $\mathcal{O}(N)$ (linear complexity)

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

- Note: hard part is constant factors

  (store/retrieve Schur updates efficiently)

Related approaches

- $\mathcal{H}$-matrices
  - [Hackbusch & collaborators, 1999-2002]
  - [Bebendorf, 2008]

- [Coulier et al., 2015] and [Ambikasaran & Darve, 2014] also look at applying an approximate inverse FMM using strong admissibility
  - Compared to our approach:
    - Multiplicative factorization versus additive decomposition
    - Skeletonization versus more general framework
    - We have extension to hybrid admissibility

Some scaling results (3D unit cube)

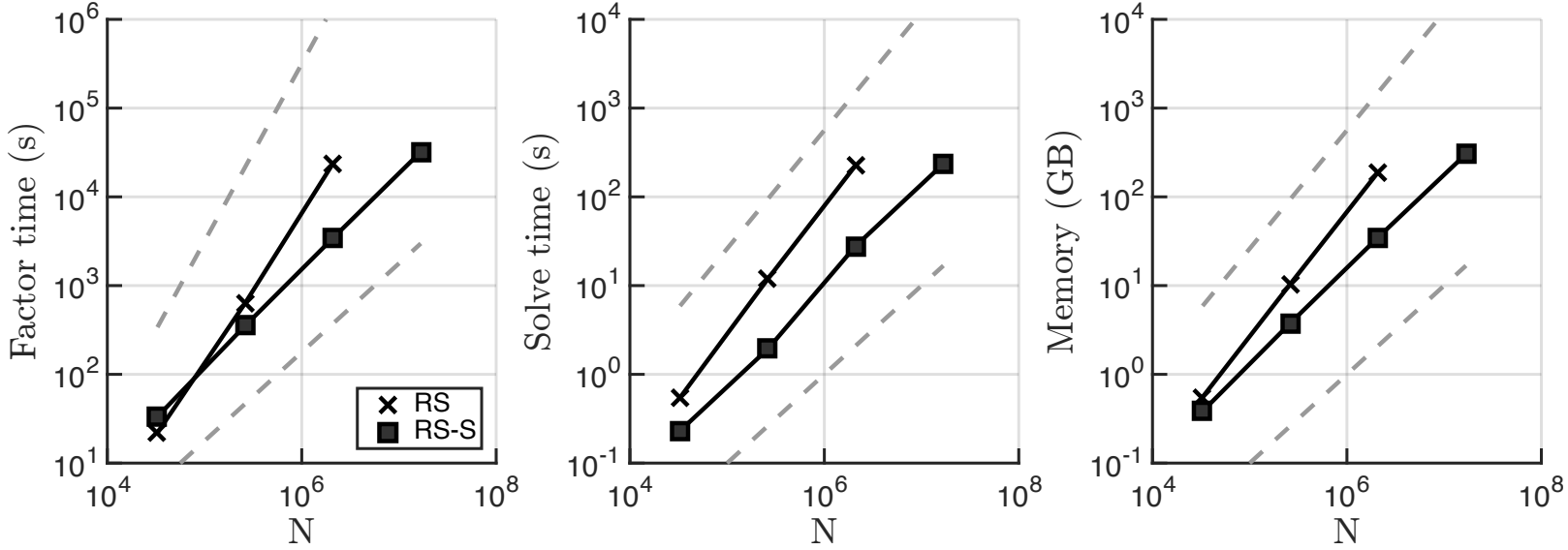
- Intel(R) Xeon(R) CPU E7-8890 @ 2.50GHz

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

- Methods
  - Recursive skeletonization (RS)
  - Strong recursive skeletonization (RS-S)

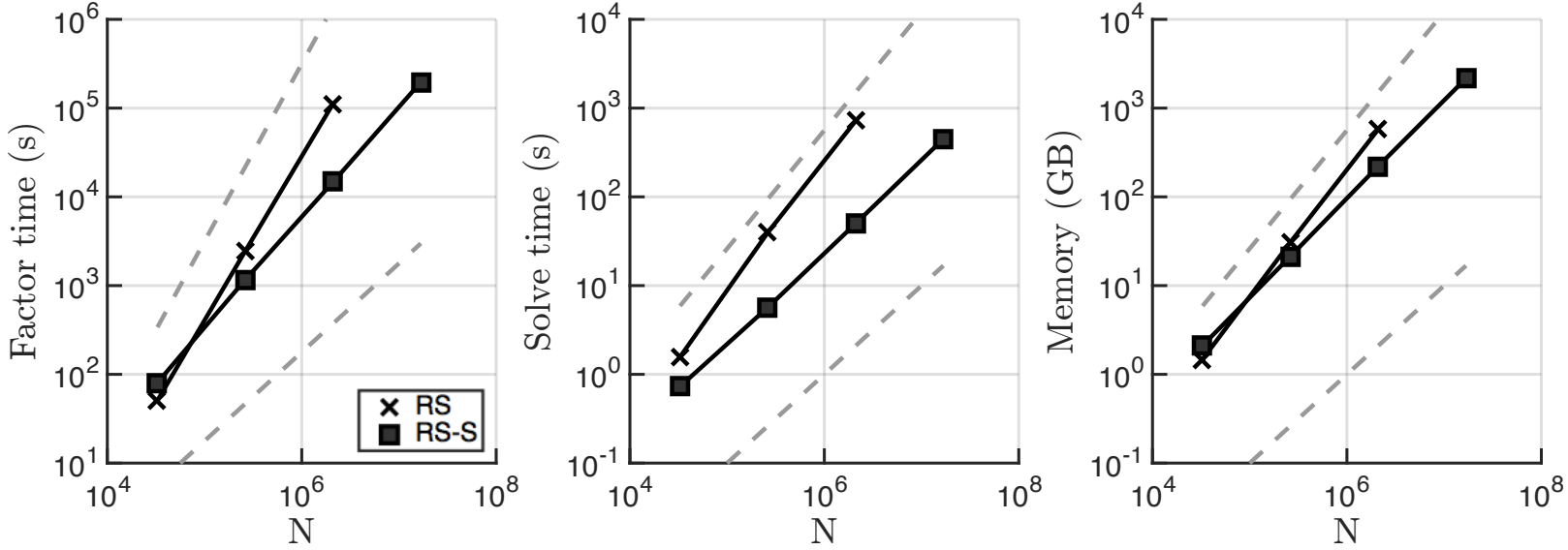
Accuracy parameter 10^{-3}

- Conjugate gradient converges to a relative residual norm of 10^{-12} in about 10 preconditioned iterations.



Accuracy parameter 10^{-6}

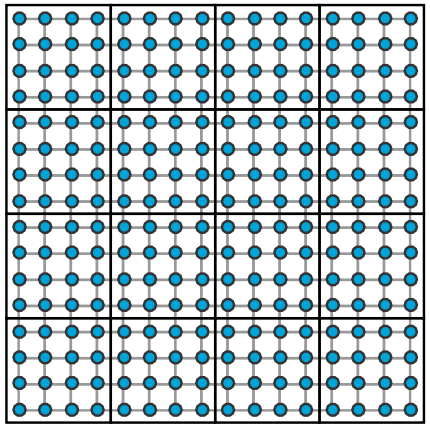
- Conjugate gradient converges to a relative residual norm of 10^{-12} in about 3 preconditioned iterations.



Comments on results

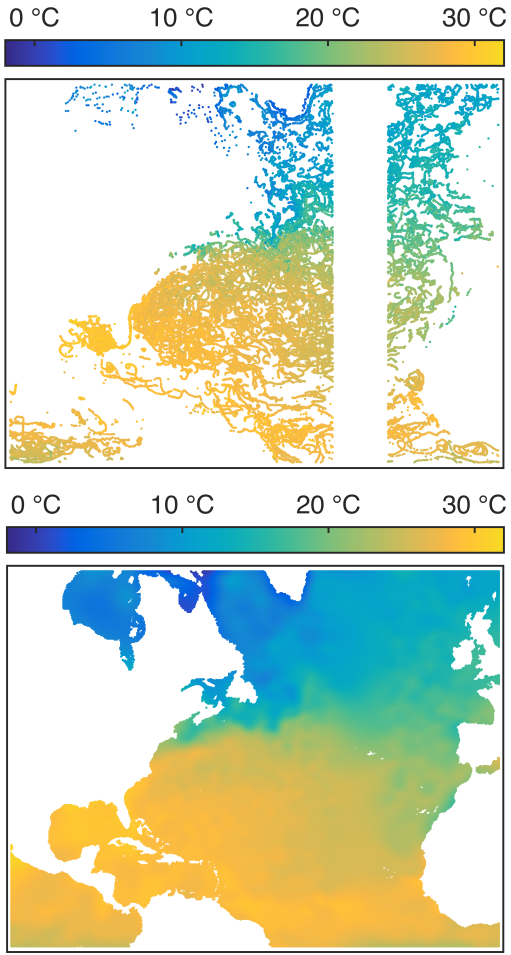
- Basic take-away
  - New factorization can be viewed as an “inverse” form of the FMM based on recursive skeletonization
  - Obtain an approximate multiplicative factorization of the inverse operator
  - Exhibits linear scaling at preconditioner accuracy levels, avoiding excess rank growth from compressing near-field
  - Simple to understand and to implement

Conclusions



Other work exploiting structure

- Updating hierarchical factorizations after local perturbations [Minden et al., 2016]
- Green's functions for time-stepping Maxwell's equations [Lo, Minden, & Colella, 2016]
- Maximum likelihood estimation for spatial Gaussian processes [Minden et al., 2016]
- Sparse canonical correlation analysis (ongoing, with Xiaotong Suo)



Papers referenced in this talk

- Graph clustering
  - References refer to bibliography section of

    Damle, Minden, & Ying, “Robust and efficient multi-way spectral clustering”, arXiv:1609.08251

- Hierarchical factorizations
  - References refer to bibliography section of

    Minden, Ho, Damle, & Ying, “A recursive skeletonization factorization based on strong admissibility”, arXiv:1609.08130

Acknowledgments

- V.M. is supported by a Stanford Graduate Fellowship and was previously supported by a U.S. Department of Energy Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308.

Thanks for listening!