## Today

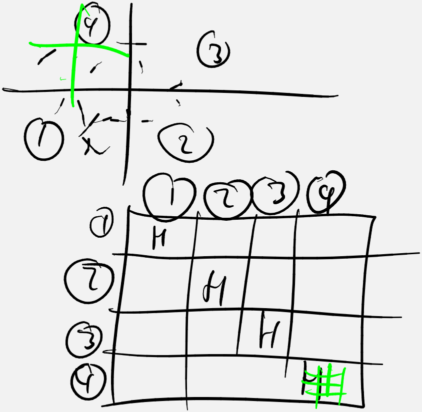
- Direct solve
- Butterfly

## Announcements

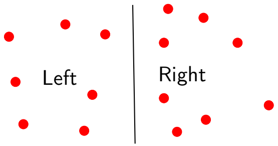
- Project
- HW3

# A Matrix View of Low-Rank Interaction

Only *parts of the matrix are low-rank*! What does this look like from a matrix perspective?

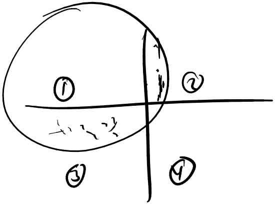


# (Recursive) Coordinate Bisection (RCB)



# Block-separable matrices

HODLR – Stanford




$$A = \begin{bmatrix} D_1 & A_{12} & A_{13} & A_{14} \\ A_{21} & D_2 & A_{23} & A_{24} \\ A_{31} & A_{32} & D_3 & A_{34} \\ A_{41} & A_{42} & A_{43} & D_4 \end{bmatrix}$$

where $A_{ij}$ has low rank: How to capture rank structure?

$$A_{ij} \approx [A_{ij}]_{(:,J_j)} \Pi_j$$

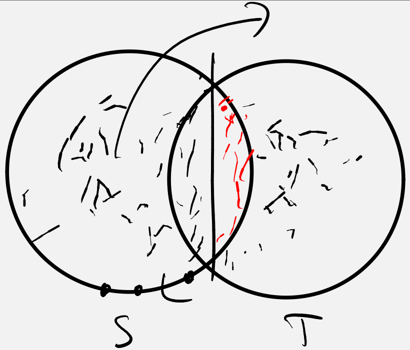
$$A_{ij} \approx P_i \, [A_{ij}]_{(I_i,J_j)} \Pi_j$$

# Proxy Recap

*Saw:* If $A$ comes from a kernel for which Green's formula holds, then the same skeleton will work for all of space, for a given set of sources/targets. What would the resulting matrix look like?

# Rank and Proxies

Unlike FMMs, partitions here do not include "buffer" zones of near elements. What are the consequences?



# Block-Separable Matrices (H)BSS

A *block-separable matrix* looks like this:

$$
A = \begin{bmatrix}
D_1 & P_1\tilde{A}_{12}\Pi_2 & P_1\tilde{A}_{13}\Pi_3 & P_1\tilde{A}_{14}\Pi_4 \\
P_2\tilde{A}_{21}\Pi_1 & D_2 & P_2\tilde{A}_{23}\Pi_3 & P_2\tilde{A}_{24}\Pi_4 \\
P_3\tilde{A}_{31}\Pi_1 & P_3\tilde{A}_{32}\Pi_2 & D_3 & P_3\tilde{A}_{34}\Pi_4 \\
P_4\tilde{A}_{41}\Pi_1 & P_4\tilde{A}_{42}\Pi_2 & P_4\tilde{A}_{43}\Pi_3 & D_4
\end{bmatrix}
$$

$r$

Here:

- $\tilde{A}_{ij}$ smaller than $A_{ij}$
- $D_i$ has full rank (not necessarily diagonal)
- $P_i$ shared for entire row
- $\Pi_i$ shared for entire column

$$\tilde{A}_{41} = [A_{41}]_{(I_i, J_i)}$$

Q: Why is it called that?

# Block-Separable Matrix: Questions

Q: Why is it called that?

sep. of variable

Q: How expensive is a matvec?

$O(N) + O\left(r^2 \frac{N}{r}\right)$ Using a strategy like BH $\to O(N^{3/2})$

Q: How about a solve?

# BSS Solve (I)

Use the following notation:

$$B = \begin{bmatrix} 0 & P_1\tilde{A}_{12} & P_1\tilde{A}_{13} & P_1\tilde{A}_{14} \\ P_2\tilde{A}_{21} & 0 & P_2\tilde{A}_{23} & P_2\tilde{A}_{24} \\ P_3\tilde{A}_{31} & P_3\tilde{A}_{32} & 0 & P_3\tilde{A}_{34} \\ P_4\tilde{A}_{41} & P_4\tilde{A}_{42} & P_4\tilde{A}_{43} & 0 \end{bmatrix}$$

and

$$D = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & D_3 & \\ & & & D_4 \end{bmatrix}, \quad \Pi = \begin{bmatrix} \Pi_1 & & & \\ & \Pi_2 & & \\ & & \Pi_3 & \\ & & & \Pi_4 \end{bmatrix}.$$

$Dx + B\Pi x = b$

Then $A = D + B\Pi$ and

$Dx + B\tilde{x} = b$

$$+\Pi D^{-1} \cdot \quad \begin{bmatrix} D & B \\ \cancel{-\Pi} & \mathrm{Id} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \tilde{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \longrightarrow \Pi x = \tilde{x}$$

$\hookrightarrow \boxed{\phantom{xxxxxx}}$

is equivalent to $A\mathbf{x} = \mathbf{b}$.

$$(\mathrm{Id} + \underbrace{\Pi D^{-1} B}_{})\tilde{x} = \Pi D^{-1} b$$

$$\begin{pmatrix} \Pi_1 & & \\ & \ddots & \\ & & \Pi_4 \end{pmatrix} \begin{pmatrix} D_1^{-1} & & & \\ & D_2^{-1} & & \\ & & \ddots & \\ & & & D_4^{-1} \end{pmatrix} \begin{pmatrix} 0 & & \ddots & \\ P\tilde{A}_{21} & 0 & & \ddots \\ & & 0 & \\ & \ddots & & 0 \end{pmatrix}$$

$$\Pi_i D_i^{-1} P_i \tilde{A}_{ij}$$

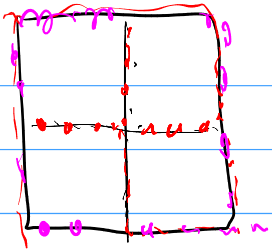
$$\tilde{A}_{ii} = (\Pi_i D_i^{-1} P_i)^{-1}$$

$$\underbrace{\mathrm{diag}(\tilde{\hat{A}}_{i,i}) \quad (\mathrm{Id} + \Pi D^{-1} B)}\tilde{x} = \mathrm{diag}(\hat{A}_{i,i})\, \Pi D^{-1} b$$

$$\begin{pmatrix} \tilde{\hat{A}}_{11} & \tilde{\hat{A}}_{12} & \\ \tilde{\hat{A}}_{21} & & \\ & & \ddots \end{pmatrix} \tilde{x} = \mathrm{diag}(\hat{A}_{i,i})\, \Pi D^{-1} b$$

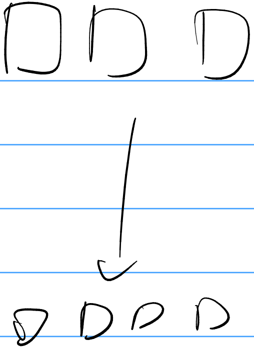
To complete the solve:

$$Dx + B\tilde{x} = b \iff Dx = b - B\tilde{x}$$



Finer level skeleton

Coarser level solve



# BSS Solve (II)

Q: What are the matrix sizes? The vector lengths of $\mathbf{x}$ and $\tilde{\mathbf{x}}$?

Now work towards doing *just* a 'coarse' solve on $\tilde{\mathbf{x}}$, using, effectively, the \ Schur complement. Multiply first row by $\Pi D^{-1}$, add to second:

# BSS Solve (III)

Focus in on the second row:

$$(\mathrm{Id} + \Pi D^{-1} B)\widetilde{\mathbf{x}} = \Pi D^{-1}\mathbf{b}$$

Every non-zero entry in $\Pi D^{-1} B$ looks like

$$\Pi_i D_i^{-1} P_i \tilde{A}_{ij}.$$

So set

$$\tilde{A}_{ii} = (\Pi_i D_i^{-1} P_i)^{-1}$$

The nomenclature makes (some) sense, because $\widetilde{A}_{ii}$ is a 'downsampled' version of $D_i$ (with two inverses thrown in for good measure).

# BSS Solve (IV)

Next, left-multiply $(\mathrm{Id} + \Pi D^{-1} B)$ by $\mathrm{diag}(\tilde{A}_{ii})$:

# BSS Solve: Summary

What have we achieved?

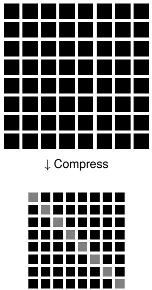
- Instead of solving a linear system of size

  $$(N_{L0\,\text{boxes}} \cdot m) \times (N_{L0\,\text{boxes}} \cdot m)$$

  we solve a linear system of size

  $$(N_{L0\,\text{boxes}} \cdot K) \times (N_{L0\,\text{boxes}} \cdot K),$$

  which is cheaper by a factor of $(K/m)^3$.
- We are now only solving on the skeletons.

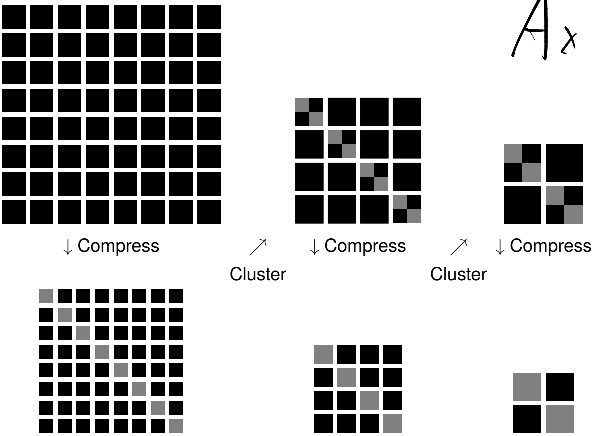



(Figure credit: G. Martinsson)

# Hierarchically Block-Separable

In order to get $O(N)$ complexity, could we apply this procedure recursively?

$Ax \to b$

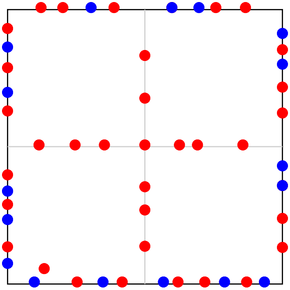



(Figure credit: G. Martinsson)

# Hierarchically Block-Separable

To get to $O(N)$, realize we can *recursively*

- group skeletons
- eliminate more variables.



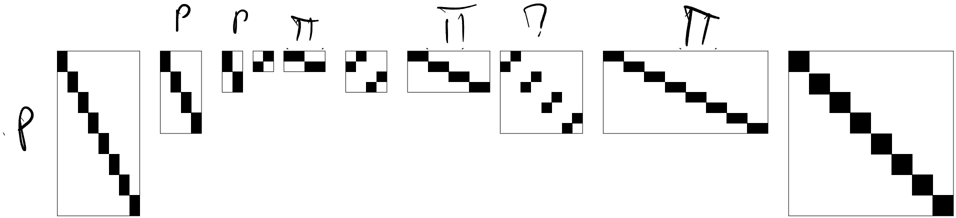
Level 1 skeletons · Level 0 skeletons

# Hierarchically Block-Separable

HODLR

- Using this hierarchical grouping gives us '/Hierarchically Block-Separable/' ('/HBS/') matrices.
- If you have heard the word '/$\mathcal{H}$-matrix/' and '/$\mathcal{H}^2$-matrix/', the ideas are very similar. Differences:
  - $\mathcal{H}$-family matrices don't typically use the ID (instead often use 'Adaptive Cross Approximation'–'ACA')
  - $\mathcal{H}^2$ does target clustering (like FMM), $\mathcal{H}$ does not (like Barnes-Hut)

# Telescoping Factorization



(Figure credit: G. Martinsson)

- The most decrease in ‘volume’ happens in the off-diagonal part of the matrix. $\rightarrow$ Rightfully so!
- All matrices are block-diagonal, except for the highest-level matrix–but that is small!

# Recap: Fast Fourier Transform

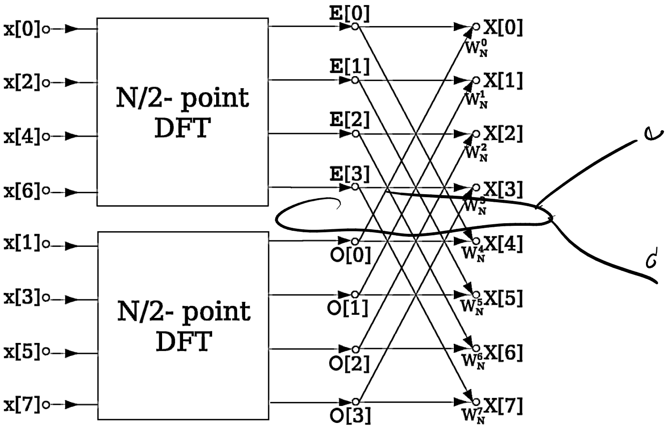
The *Discrete Fourier Transform (DFT)* is given by:

$$X_k = \sum_{n=0}^{N-1} x_n \underbrace{e^{-\frac{2\pi i}{N} nk}} \quad (k = 0, \dots, N-1)$$

The foundation of the *Fast Fourier Transform (FFT)* is the factorization:

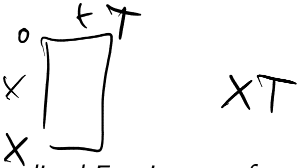
$$X_k = \underbrace{\sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2} mk}}_{\text{DFT of even}-\text{indexed part of } x_n} + e^{-\frac{2\pi i}{N} k} \underbrace{\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2} mk}}_{\text{DFT of odd}-\text{indexed part of } x_n} .$$

# FFT: Data Flow




Perhaps a little bit like a butterfly?

# Fourier Transforms: A Different View




Claim:

> *The [numerical] rank of the normalized Fourier transform with kernel $e^{i\gamma xt}$ is bounded by a constant times $\gamma$, at any fixed precision $\epsilon$.*

(i.e. rank is bounded by the area of the rectangle swept out by $x$ and $t$)
[O'Neil et al. '10]

**Demo:** Conditioning of Derivative Matrices (Part I)