

Hierarchical low-rank approximation methods on distributed memory and GPUs

背景

H行列などの階層的低ランク近似法では、密行列を階層的に分割し非対角ブロックを低ランク近似することで行列積や行列分解のコストを $O(N^3)$ から $O(N)$ に低減できる

境界要素法の反復解法における前処理行列、疎行列のマルチフロントル法による解法におけるフロントル行列、統計学における分散共分散行列の行列式などの計算に現れる密行列はH行列で表すことができる

H行列はこれまで応用数学の分野の研究者が中心となって開発しており高性能実装、並列化、GPU化などがほとんど行われていない

H行列の内部では小さな行列積の計算が無数に行われるため、TensorCoreなどを用いたBatched GEMMを活かすことができる

低精度演算を用いるとき、密行列を厳密に計算することは無駄であり、H行列を用いれば低精度な行列演算は全て $O(N)$ で行うことができる

継続元課題の成果

- 2016年度：H行列一ベクトル積のGPU実装、動的負荷分散
- 2017年度：H行列のLU分解、batched MAGMAによる高速化
- 2018年度：格子H行列、LU分解の分散並列実装
- 2019年度：H行列のQR分解、Tensor Coreによる高速化
- 2020年度：格子H行列のGPU実装、H²行列への拡張
- 2021年度：H²行列のLU分解、H行列のランタイムによる高速化
- 2022年度：BLR行列の固有値分解、H行列のLDL分解

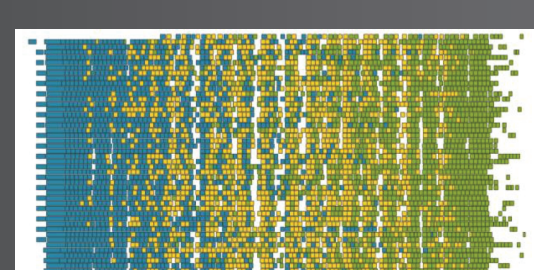
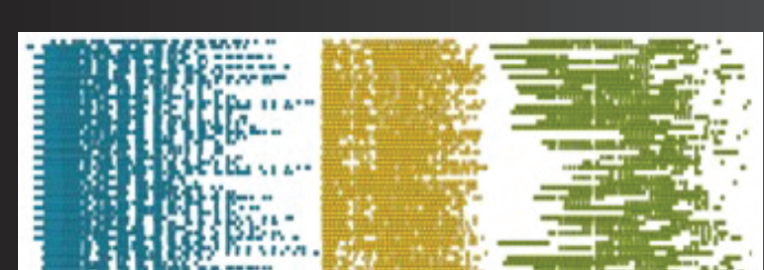
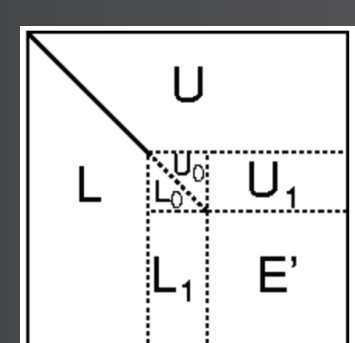
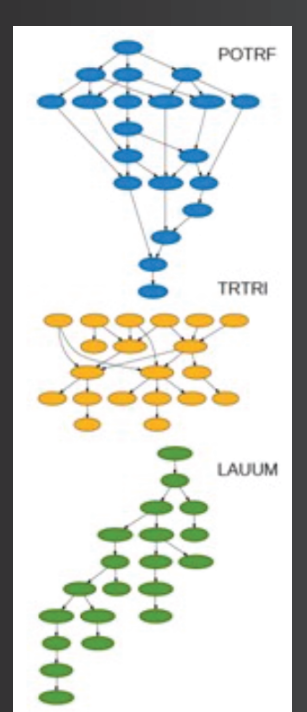
2023年度の目標

- 1) ランタイムによるH²行列のLU分解の高速化
- 2) H²行列のLDL分解を用いたk番目の固有値の計算
- 3) H²行列のLU分解のマルチGPU実装
- 4) BLR行列のQR分解のGPU実装
- 5) BLR行列の三重対角化

ランタイム (PaRSEC)

Fork-join

Runtime



低ランク行列のLU分解

$$A = [U^R \ U^S] \begin{bmatrix} 0 & 0 \\ 0 & SSS \end{bmatrix} \begin{bmatrix} V^R \\ V^S \end{bmatrix}$$

Low-Rank = * * * *

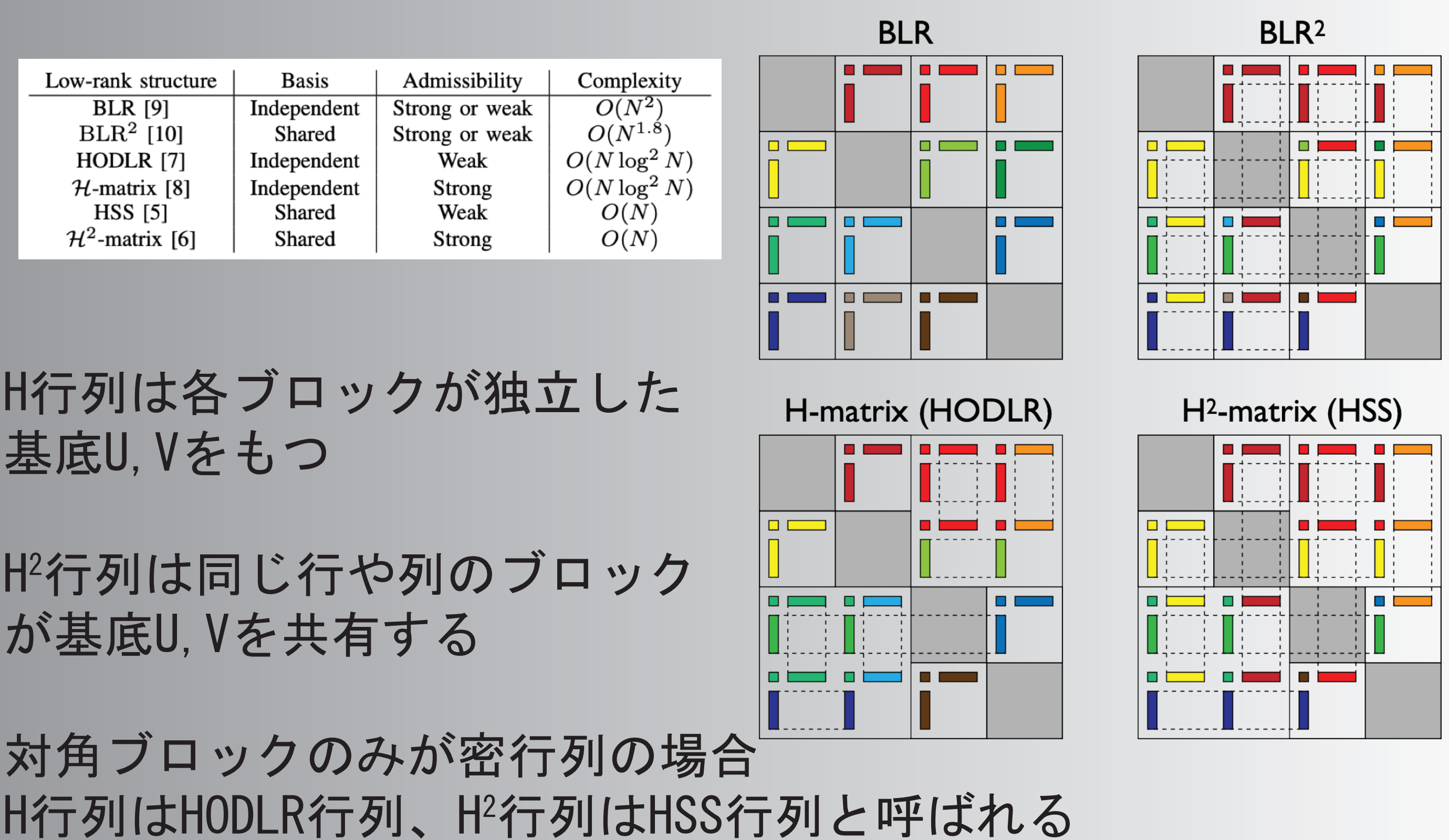
大きい特異値を右下に持ってくる
→この部分を除いてLU分解
→Fill-inが起きない

$$A = [U^R \ U^S] \begin{bmatrix} S^{RR} & S^{RS} \\ S^{SR} & S^{SS} \end{bmatrix} \begin{bmatrix} V^R \\ V^S \end{bmatrix}$$

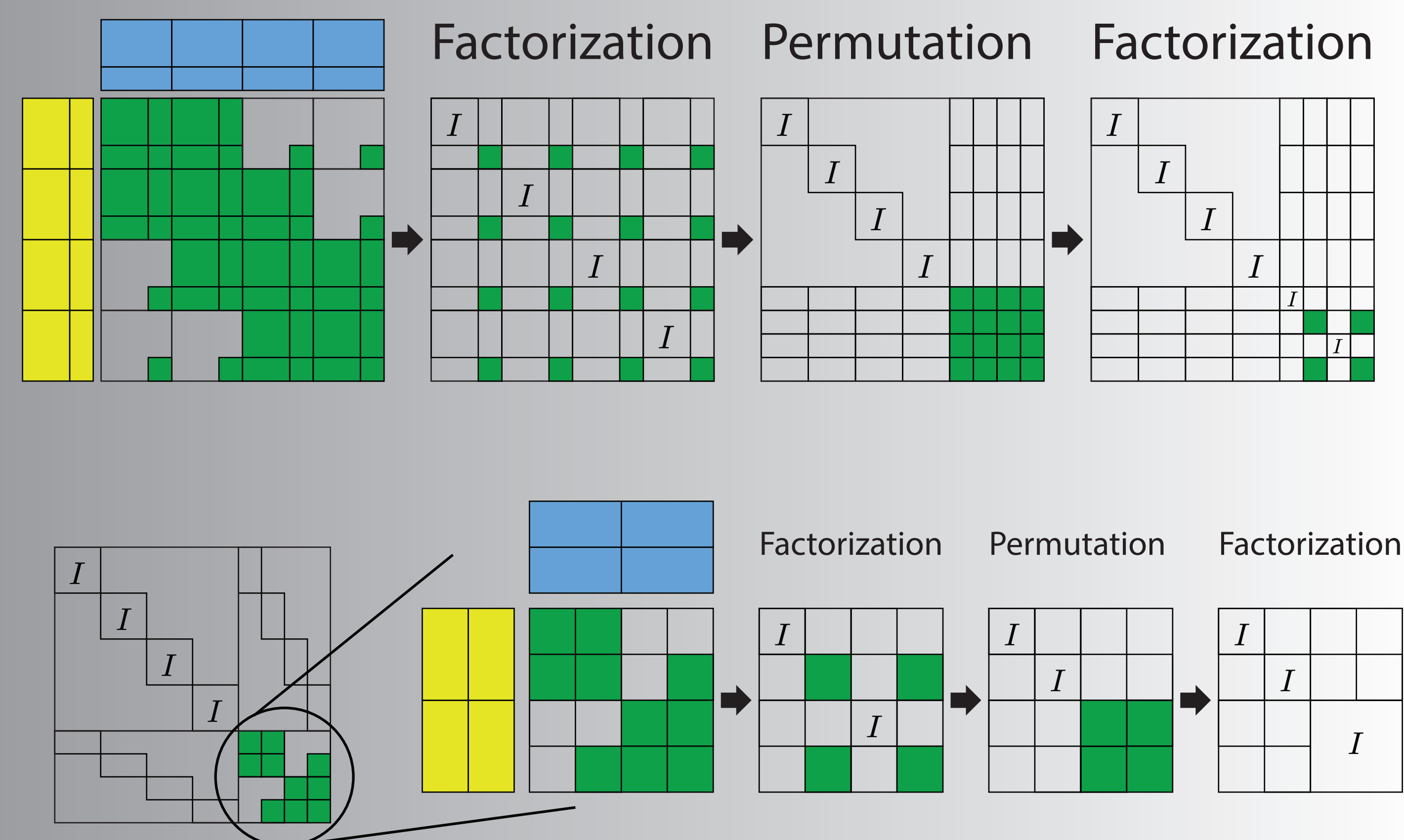
Dense = * * * *

密ブロックも同じように分解
→各行・列を共通のU, Vで特異値分解
→行列全体をU S Vに分解できる
→SのみをLU分解

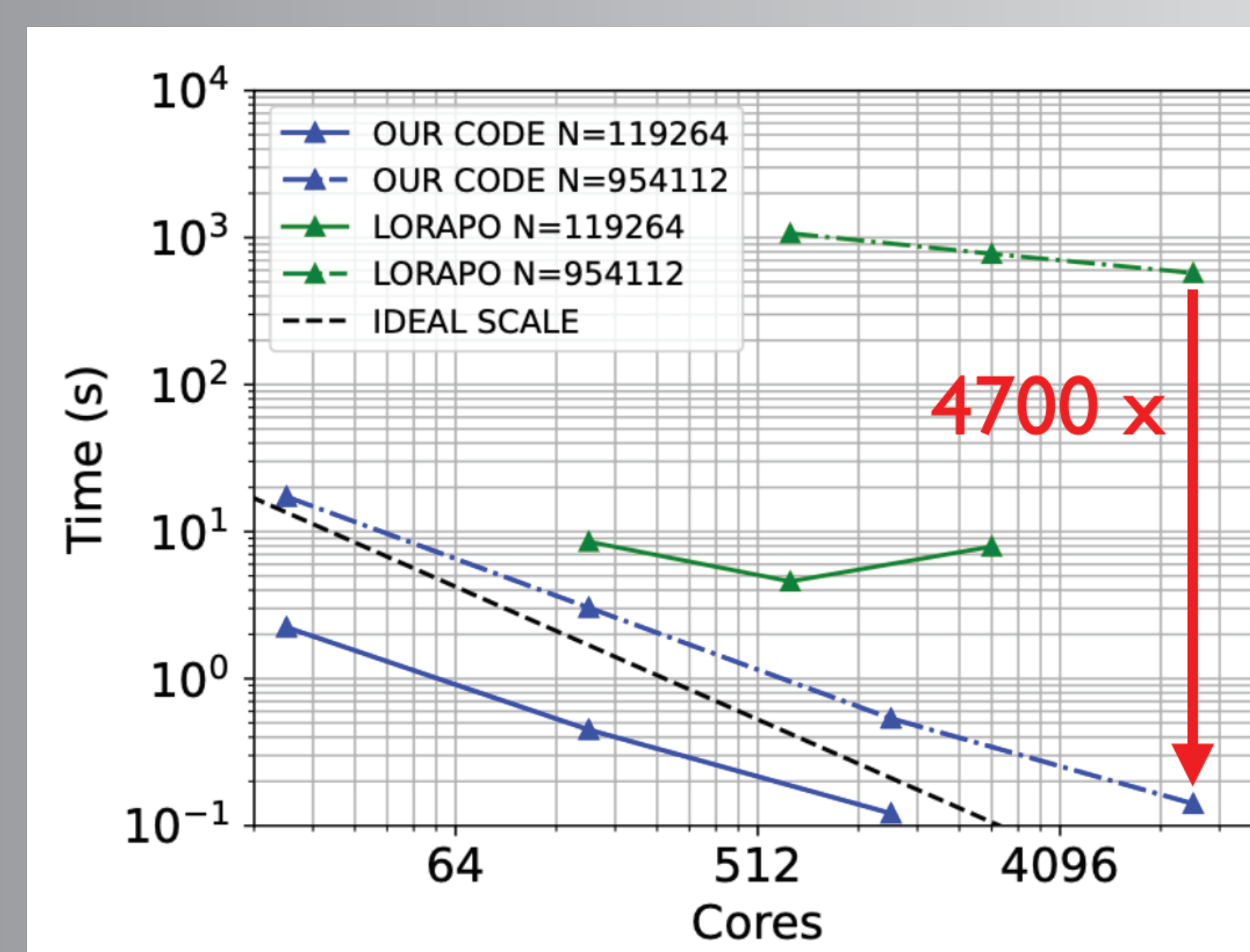
H行列とH²行列の違い



H²行列のLU分解を $O(N)$ で行うには



H²行列のLU分解の並列化効率



Gordon Bell Finalistのコード LORAP0よりも4700倍高速

$O(N)$ と $O(N^2)$ の違い

並列化効率が圧倒的に高い

