第40回　計算数理工学フォーラム

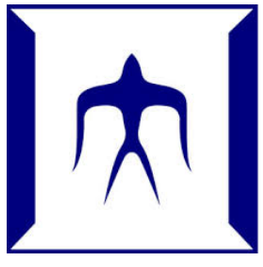
2021/9/24

階層的低ランク近似法に関するレビュー

東京工業大学 学術国際情報センター

横田 理央

rioyokota@gsic.titech.ac.jp



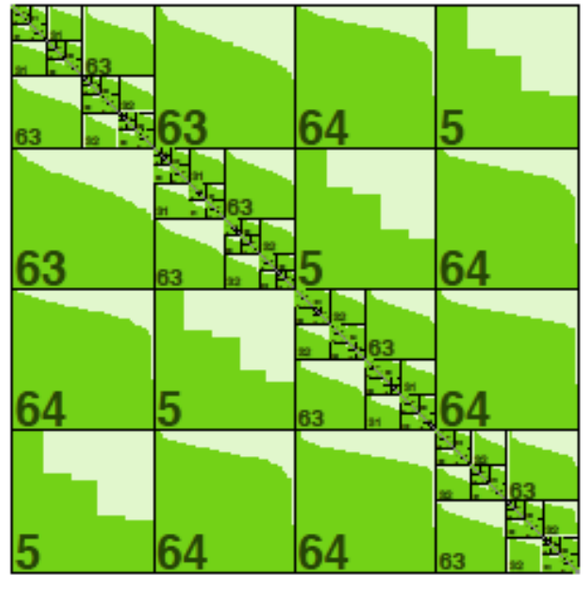
階層的低ランク近似法とは？

密行列の近似直接解法

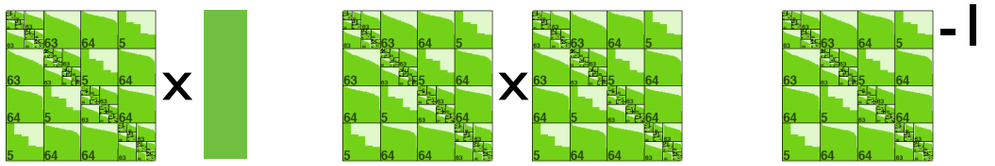
演算量 : $\mathcal{O}(N^3) \longrightarrow \mathcal{O}(N)$

メモリ : $\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$

非対角ブロックのランクは小さいという仮定

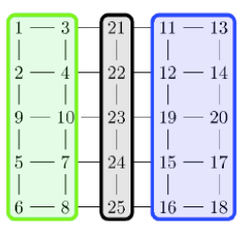


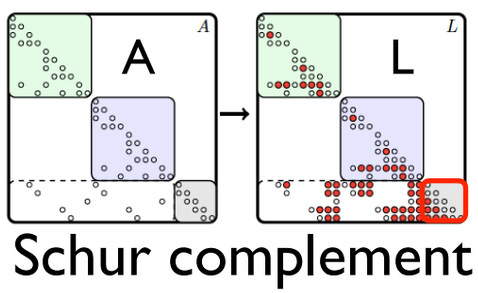
疎行列に使えるのか？



Fill-inさせないことの方が重要

Schur補元は密だが非対角ブロックのランクは小さい





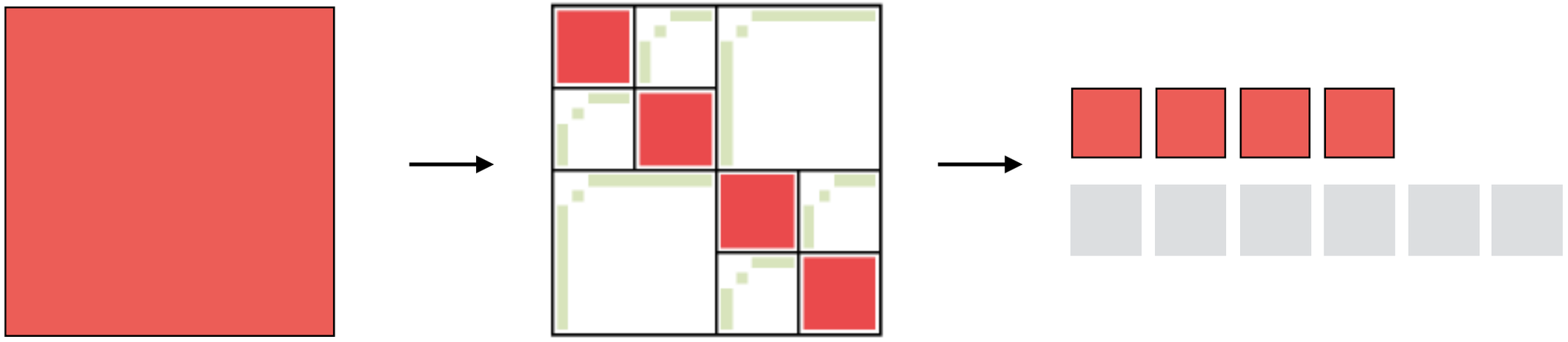
反復法では？

前処理として使えるがMultigridにはなかなか勝てない

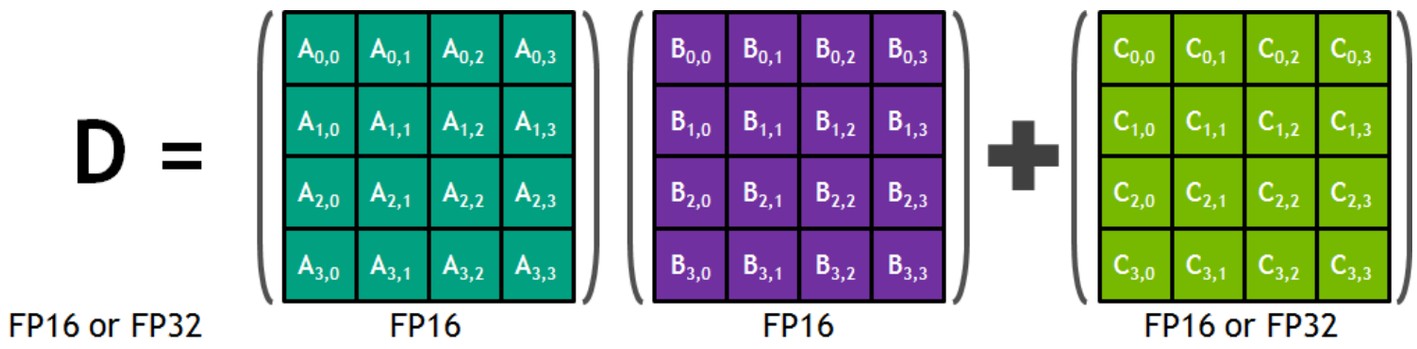
条件数が悪い問題でなら優位かも？

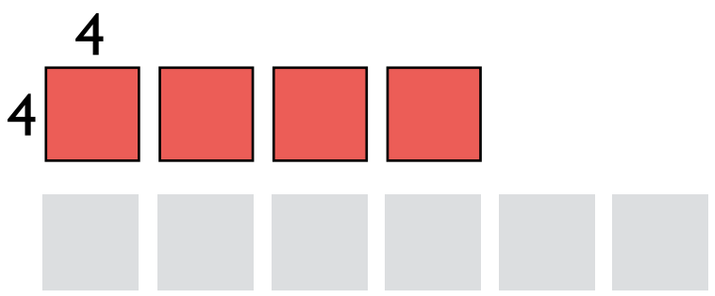
機械学習向けハードウェアと相性が良い

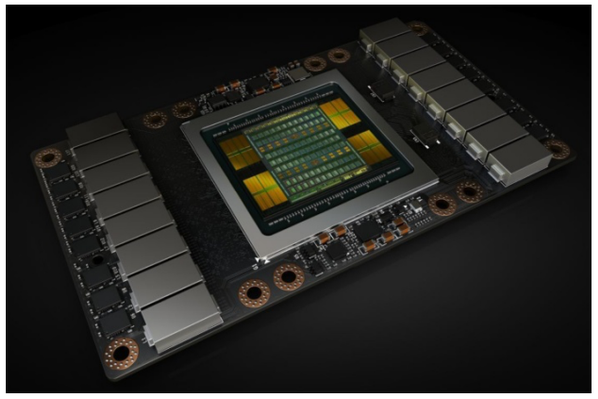
小さな密行列演算がたくさん生じる



低ランク近似なので低精度演算で十分

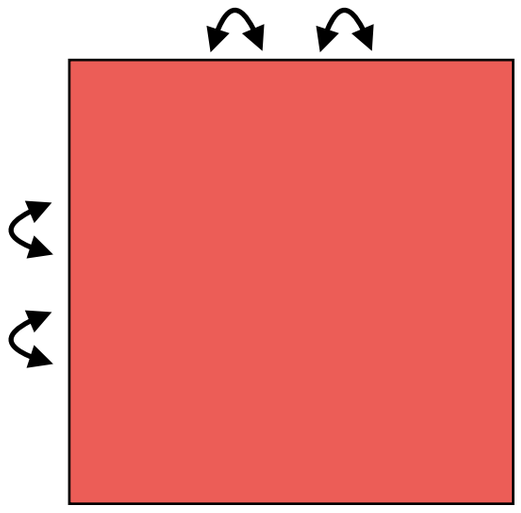






階層的低ランク近似法の3つのステップ

並べ替え



<u>最小化したいのは？</u>

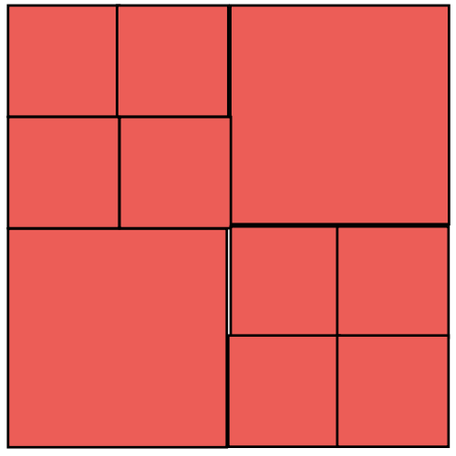
ランク (幾何学的距離)

通信 (データ局在性)

Fill-in (グラフの接続)

→普通は近いものと
繋がっているので
Fill-inとランクの
最小化は両立する

階層化



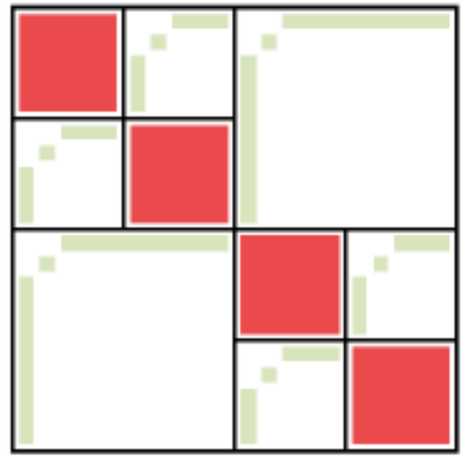
<u>どこまで分割するか？</u>

分割数を増やせば
それぞれのランクは
小さくなる

ランクは固定で分割に
よって精度を制御する
こともできる

→SIMD friendly

低ランク近似



<u>速さか，安定性か？</u>

ACAは速いが不安定

RSVDは安定だが遅い

どのブロックを近似
するかを選択するのに
Gram距離を使うこと
もできる

Replacing Exact Linear Algebra with Low-Rank

厳密解

$\mathcal{O}(N^3)$

近似解

$\mathcal{O}(N)$

Application

ScaLAPACK | cuSolverMG

LAPACK PLASMA | MKL | cuSolverDN MAGMA

BLAS | CUBLAS

CPU
FP64

GPU
FP32

分散

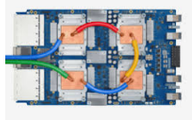
QR
LU
MatMul
Mat-vec

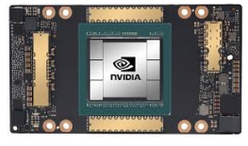
App.

HiCMA
STRUMPACK
GOFMM
LoRaSp

HBLAS

?PU
TF32, bfloat16





List of implementations

	Method	Developer	url
AHMED	H-matrix	M. Bebendorf	https://github.com/xantares/ahmed
ASKIT	FMM	C. D. Yu	http://padas.ices.utexas.edu/libaskit
DMHM	H-matrix	J. Poulson	https://bitbucket.org/poulson/dmhm/src/default/
GOFMM	H^2-matrix	C. D. Yu	https://github.com/ChenhanYu/hmlp
H2Lib	H^2-matrix	S. Börm	https://github.com/H2Lib/H2Lib
H2Tools	H^2-matrix	A. Mikhalev	https://bitbucket.org/muxas/h2tools
HACApK	H-matrix	A. Ida	https://github.com/HLRA-JHPCN/HACApK-MAGMA
HiCMA	H-matrix	H. Ltaief	https://github.com/ecrc/hicma
HLib	H-matrix	L. Grasydyck	http://www.hlib.org
HLibPro	H-matrix	R. Kriemann	http://www.hlibpro.com
hmglib	H-matrix	P. Zaspel	https://github.com/zaspel/hmglib
HODLR	HODLR	A. Aminfar	https://github.com/amiraa127/Dense_HODLR
HSS	HSS	J. Xia	http://www.math.purdue.edu/~xiaj/
LoRaSp	H^2-matrix	H. Pouransari	https://bitbucket.org/hadip/lorasp
MUMPS-BLR	BLR	P. R. Amestoy	http://mumps.enseeiht.fr
STURMPACK	HSS	P. Ghysels	http://portal.nersc.gov/project/sparse/strumpack

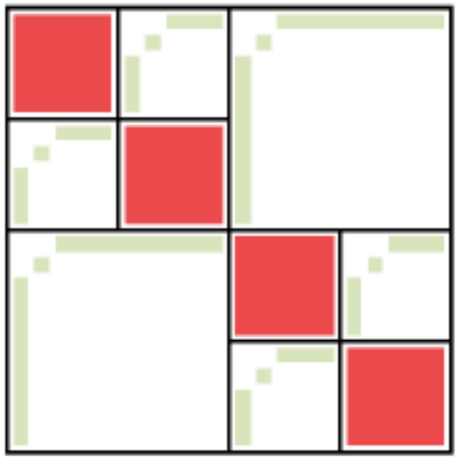
https://github.com/gchavez2/awesome_hierarchical_matrices

手法間の違い

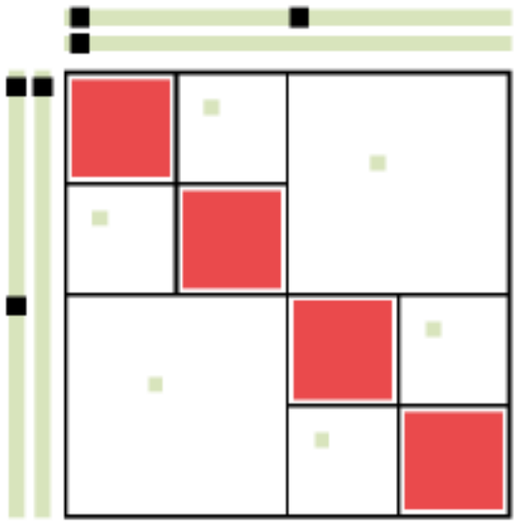
	基底の共有	許容条件
H行列	無	強
H^2行列	有	強
HODLR	無	弱
HSS	有	弱
BLR	無	非階層的
BLR^2	有	非階層的

基底

非共有

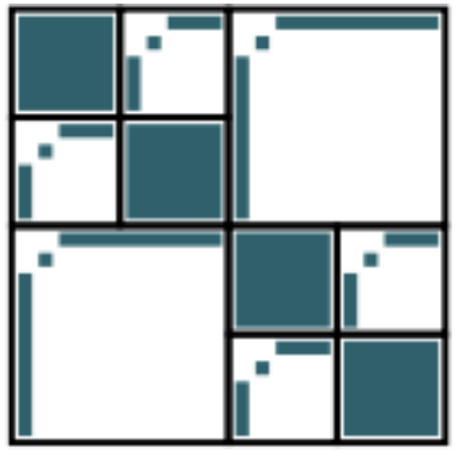


共有



許容条件

弱



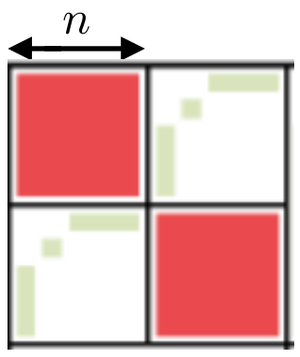
強

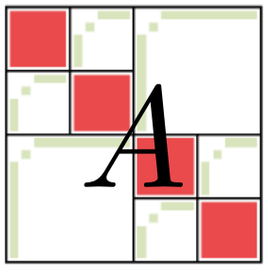
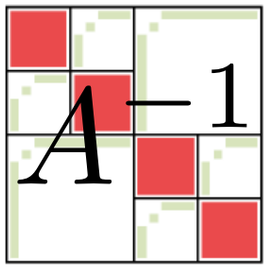
Nullity Theorem

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

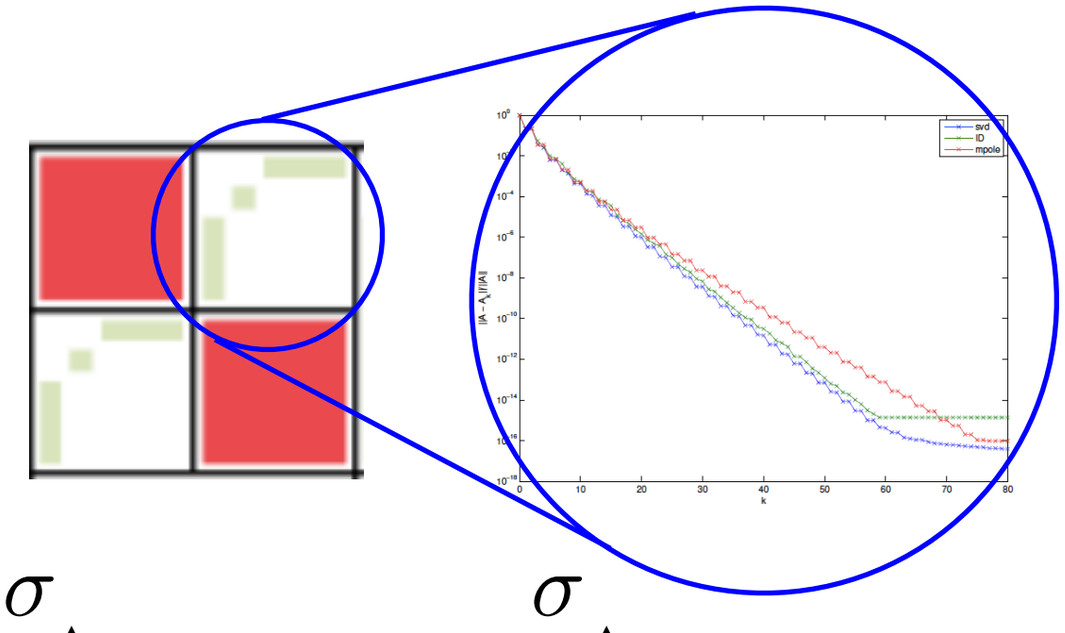
$$\begin{aligned} &\text{nullity } A = \text{nullity } H, \\ &\text{nullity } B = \text{nullity } F, \\ &\text{nullity } C = \text{nullity } G, \\ &\text{nullity } D = \text{nullity } E. \end{aligned}$$

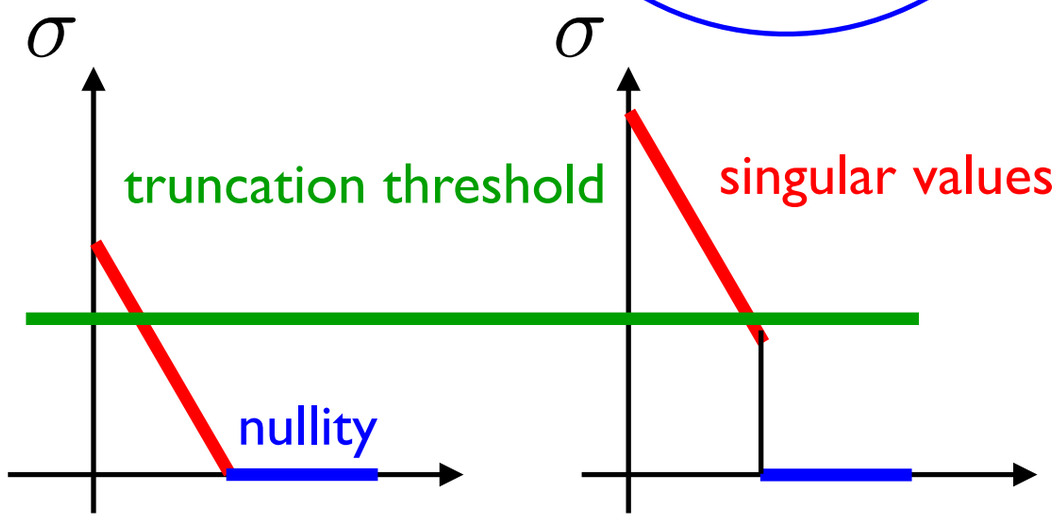
$$\text{rank}(A) + \text{nullity}(A) = n.$$



if  then 

特異値の減衰

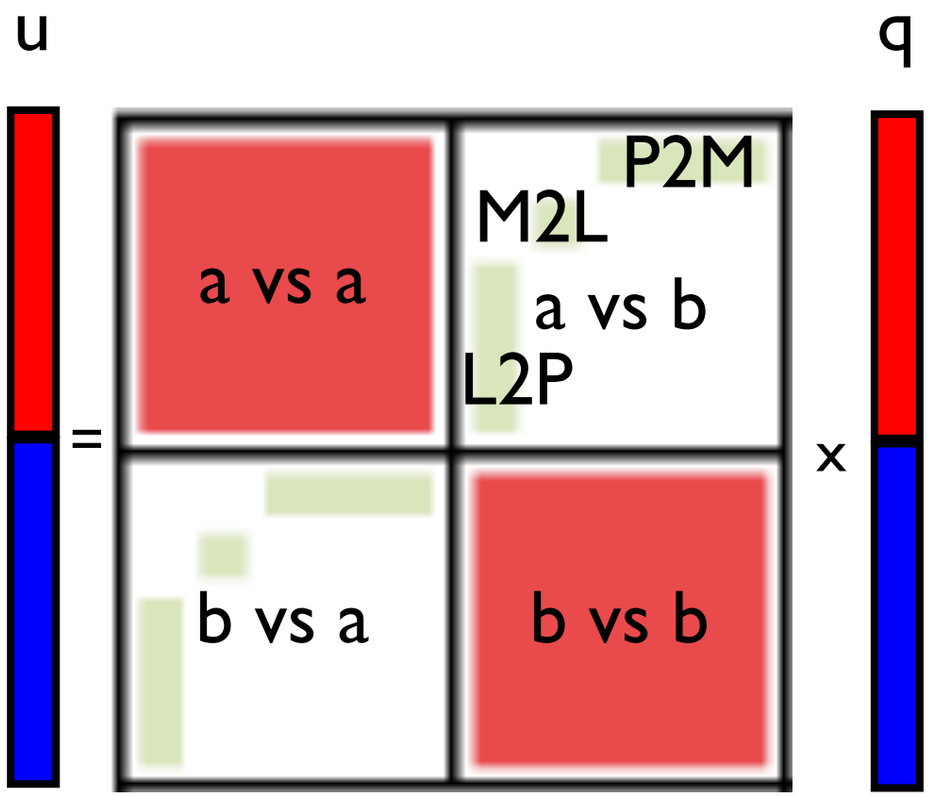


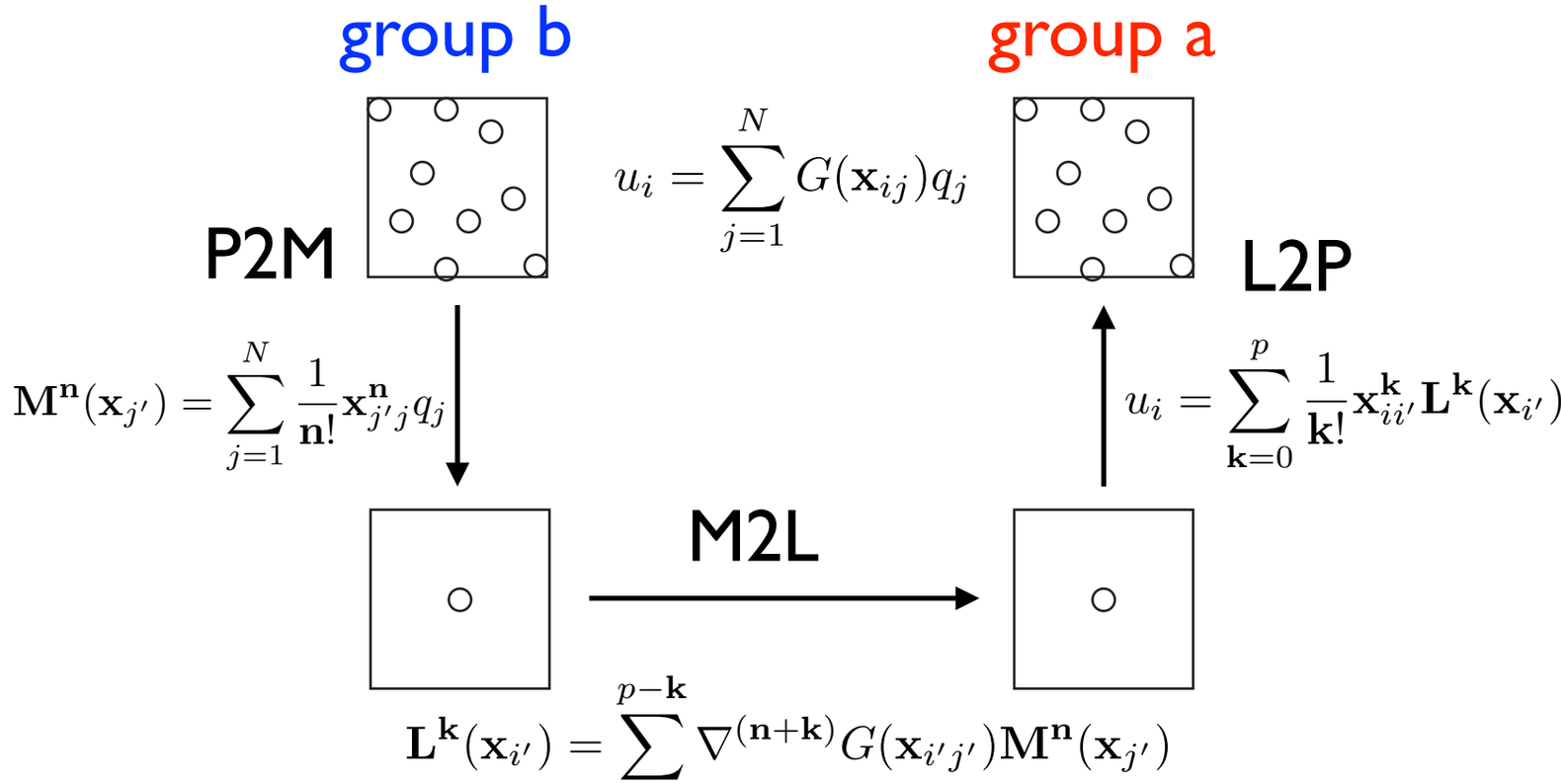


A $A * 100$

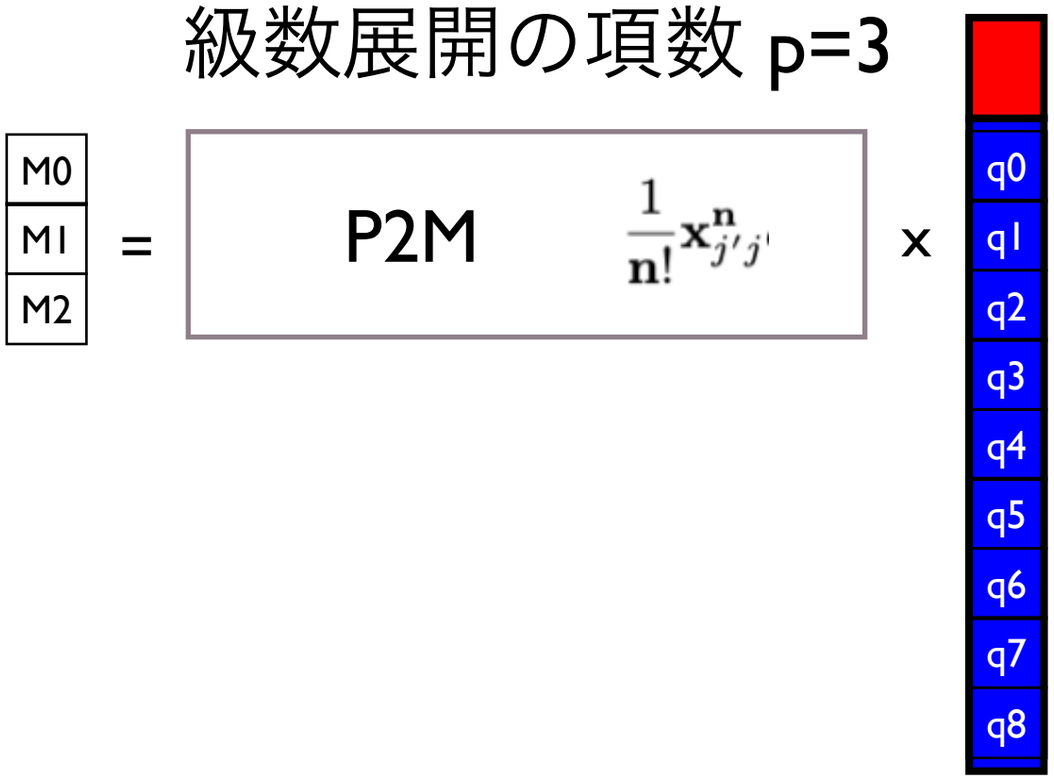
Gustafson, William H. (1984), "A note on matrix inversion", *Linear Algebra and Its Applications*, **57**: 71–73, doi:10.1016/0024-3795(84)90177-0
Strang, Gilbert; Nguyen, Tri (2004), "The interplay of ranks of submatrices" (PDF), *SIAM Review*, **46** (4): 637–646, doi:10.1137/S0036144503434381

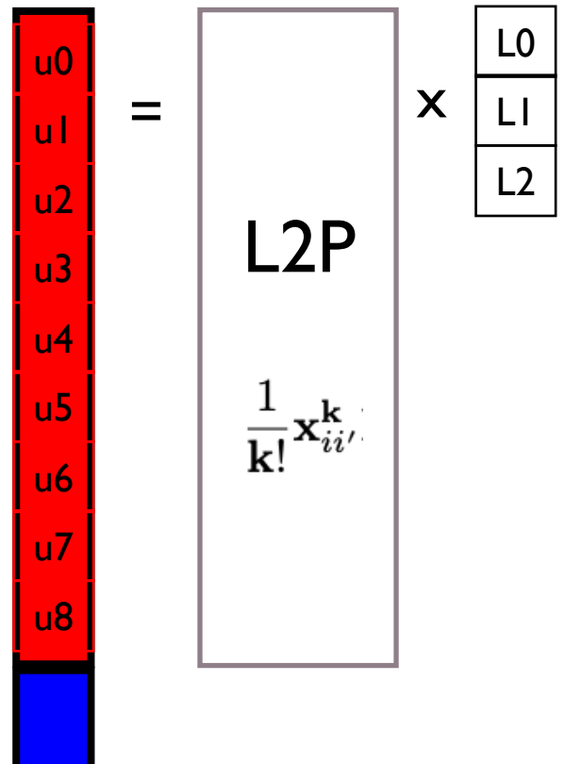
FMMはH²行列ーベクトル積





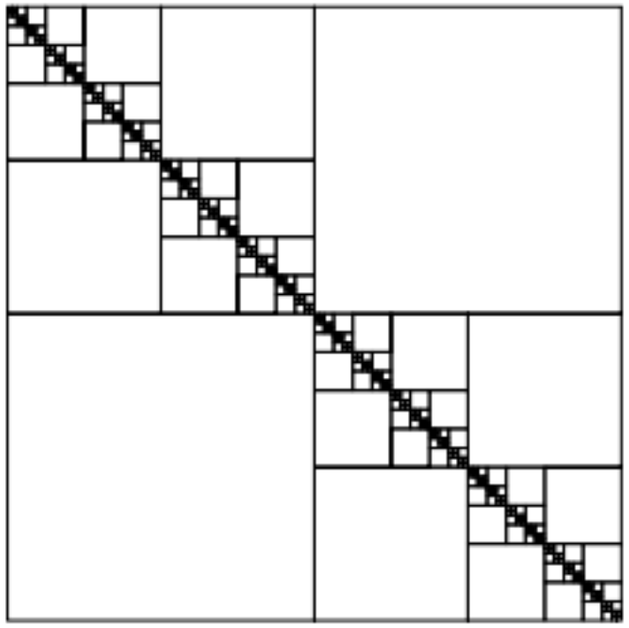
級数展開の項数 p=3

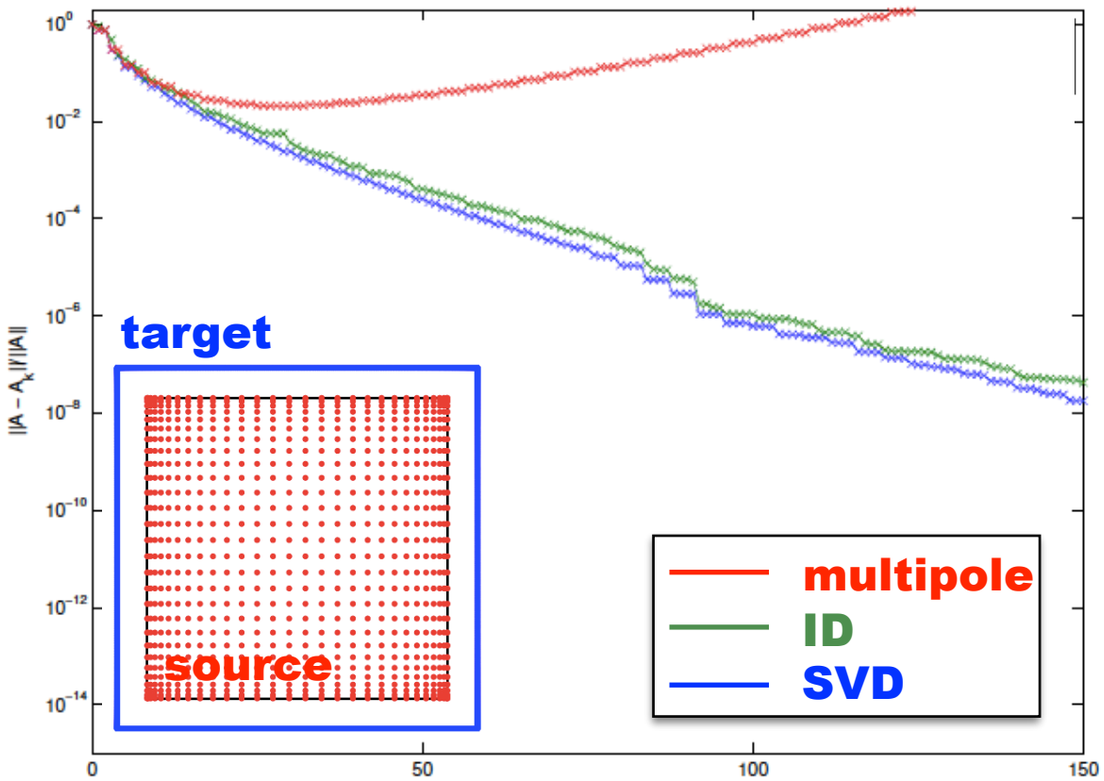




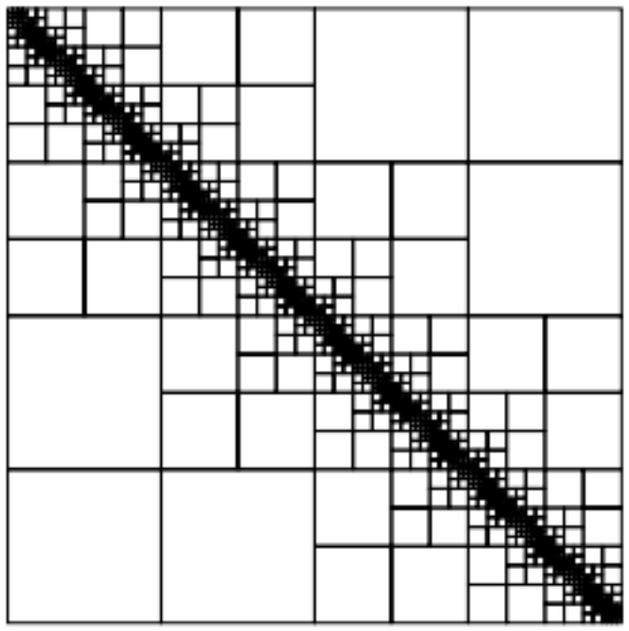
許容条件

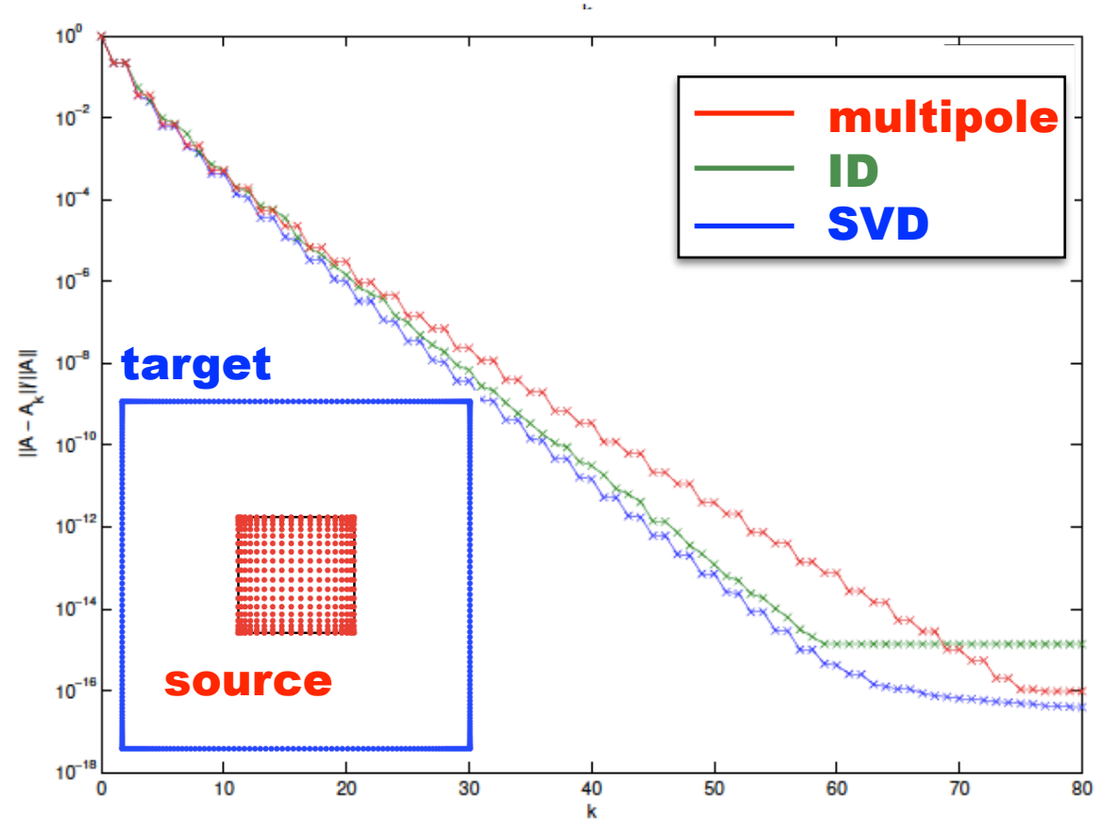
Weak admissibility





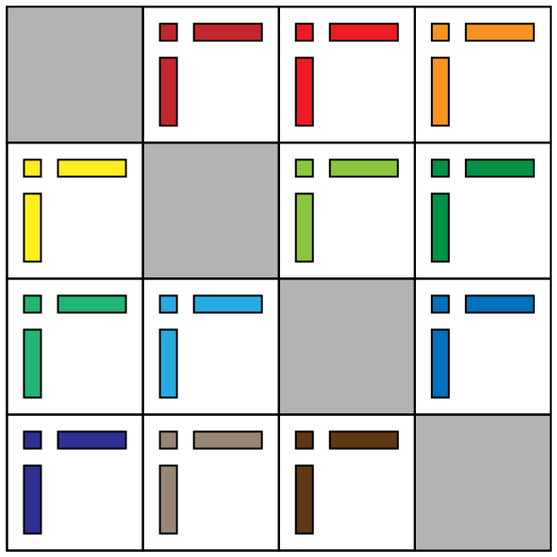
Standard admissibility



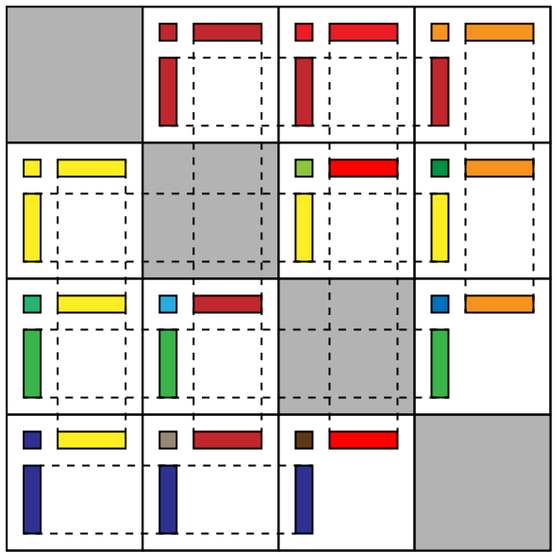


Uniform Basis

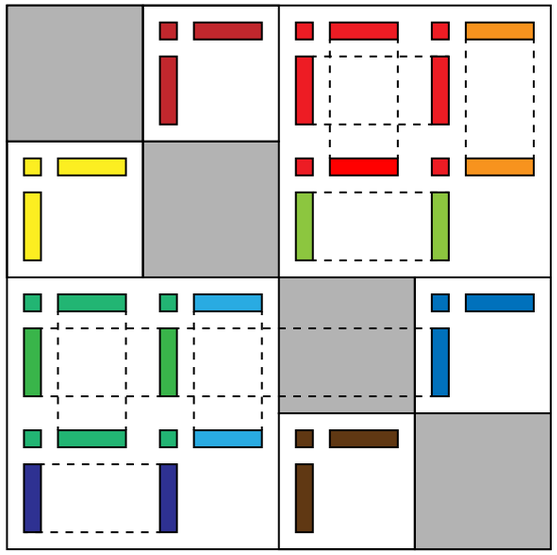
BLR



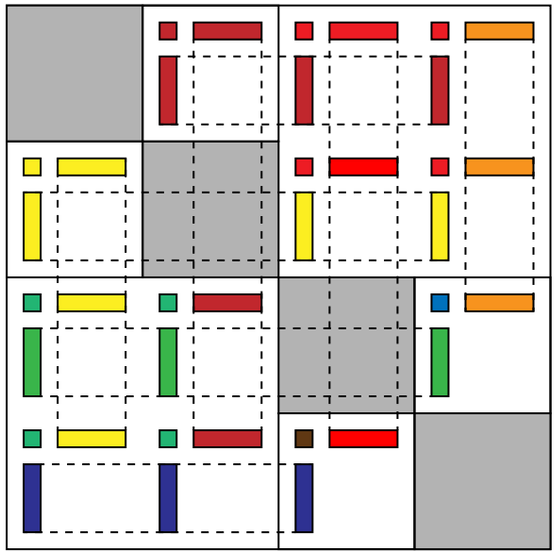
BLR^2



H-matrix (HODLR)

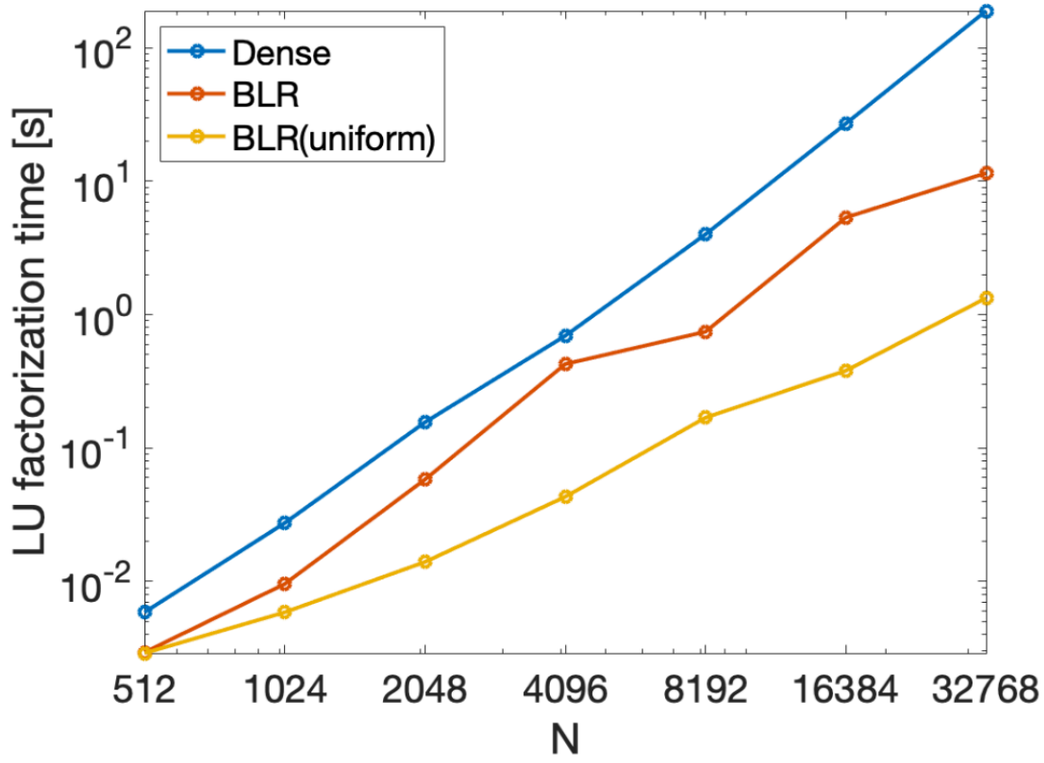


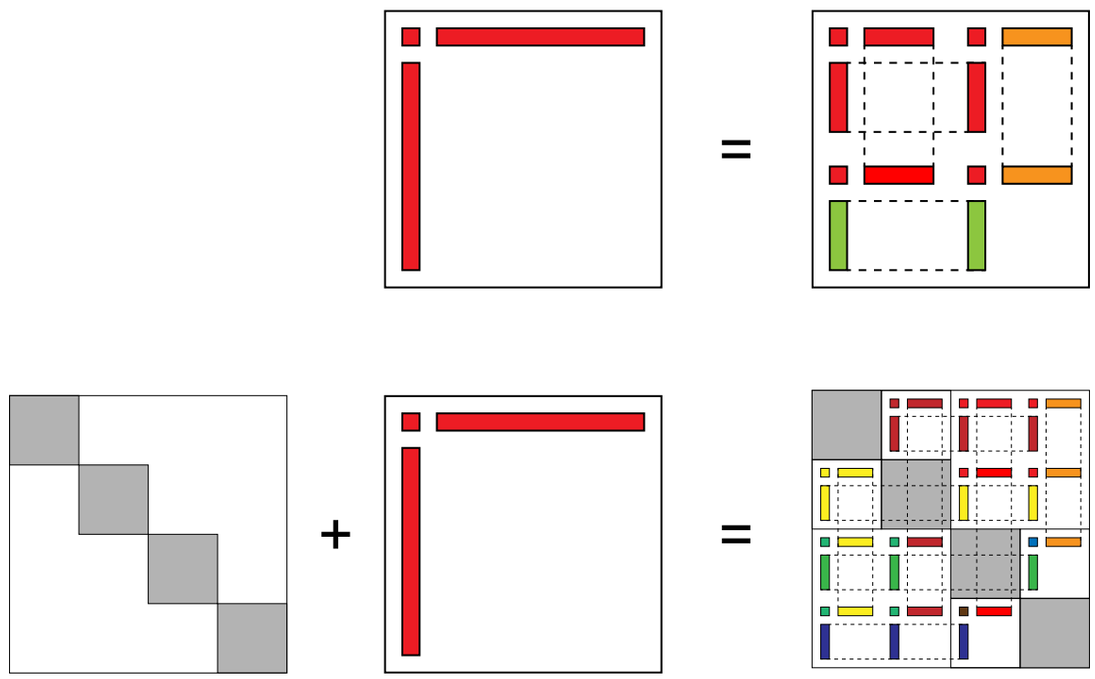
H^2-matrix (HSS)



Nonuniform basis

Uniform basis





C. Ashcraft, A. Buttari, T. Mary, "Block Low-Rank Matrices with Shared Bases: Potential and Limitations of the BLR^2 Format", https://hal.archives-ouvertes.fr/hal-03070416

勢力図

Germany

- Shared memory H-LU
  Kriemann (2014)
- Nested cross approximation
  Börm & Christophersen (2014)
- H^2-matrix for eigenvalues
  Berner et al. (2015)
- OmpSs H-LU
  Aliaga et al. (2017)
- GCA H^2-matrix
  Börm et al. (2018)

Berkeley

- HSS2D
  Xia (2014)
- HSS selected inversion
  Xia et al. (2015)
- Superfast DC eigenvalue
  Vogel, et al. (2016)
- Shared memory HSS MF
  Ghysels et al. (2016)
- Distributed HSS MF
  Rouet et al. (2016)

Japan

- Distributed H-matrix
  Ida et al. (2015)
- Distributed GPU H-matrix
  Yamazaki et al. (2018)
- Lattice H-matrix
  Ida (2018)
- GPU load-balance ACA
  Hoshino et al. (2018)
- Mixed precision H-matrix
  Ooi et al. (2020)

EPFL

- HODLR QR
  Kressner et al. (2018)

Minnesota

- Multilevel Low-Rank
  Li & Saad (2013)
- DD Low-Rank
  Li & Saad (2014)
- Multilevel Schur Low-Rank
  Xi et al. (2016)
- SMASH
  Cai et al. (2018)
- H^2-matrix + FMM
  Xing & Chow (2021)

Stanford(Ying)

- O(N) RS 2-D
  Corona (2015)
- HIF for PDEs
  Ho & Ying (2016)
- Distributed memory HIF
  Li & Ying (2016)
- RS for maximum likelihood
  Minden et al. (2016)
- RS with strong admissibility
  Minden et al. (2017)
- Quantized Tensor Train
  Corona et al. (2017)

Stanford(Darve)

- HODLR multifrontal
  Aminfar et al. (2016)
- IFMM precond. Stokes
  Coulier et al. (2017)
- IFMM precond. Helmholtz
  Takahashi et al. (2017)
- Non-extensive sparsification
  Sushnikova et al. (2017)
- Sparsified Nested Dissection
  Cambier et al. (2019)
- spaND QR
  Gnanasekaran et al. (2020)

INRIA

- BLR multifrontal
  Amestoy et al. (2015)
- BLR multicore
  Amestoy et al. (2017)
- Multilevel BLR
  Amestoy et al. (2019)

KAUST

- BLR Cholesky
  Akbudak et al. (2017)
- Batched QR, SVD
  Boukaram et al. (2018)
- GPU MatVec
  Boukaram et al. (2019)

Texas (Biros)

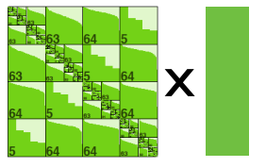
- inv-ASKIT
  Yu et al. (2016)
- Distributed inv-ASKIT
  Yu et al. (2017)
- GOFMM
  Yu et al. (2017)
- Distributed GOFMM
  Yu et al. (2018)

最近の論文の紹介

著者・年	行列	演算	構造	近似法	OpenMP	MPI	GPU
2016Vogel	密	$Q\Lambda Q^T$	HSS	?	×	×	×
2017Akbudak	密	Cholesky	BLR	RSVD	○	×	×
2017Fernando	密	LU	HSS	ID	×	○	○
2017Ghysels	疎	LU	HSS	RSVD	○	○	×
2017Li	疎	LU	HSS	RRQR	×	○	×
2017Minden	疎	LU	H2	RRQR	×	×	×
2018Amestoy	疎	LU	MBLR	?	×	×	×
2018Börm	密	O(N)圧縮	H2	GCA	×	×	×
2018Cai	密	O(N)圧縮	HSS	RRQR	×	×	×
2018Kressner	密	QR	HODLR	?	×	×	×
2018Yu	密	O(N)圧縮	HSS	ID	○	○	×
2019Amestoy	疎	LU	BLR	?	○	○	×
2019Boukaram	密	MV	H2	RSVD	×	×	○
2019Cambier	疎	LU	H2	RRQR	×	×	×
2019Zaspel	密	MV	H	ACA	×	×	○

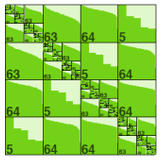
並列化

R. Kriemann (2005), Parallel {H}-Matrix Arithmetics on Shared Memory Systems
R. Kriemann (2015), H-LU factorization on many-core systems



batched MAGMA

LU
QR



StarPU / OmpSs

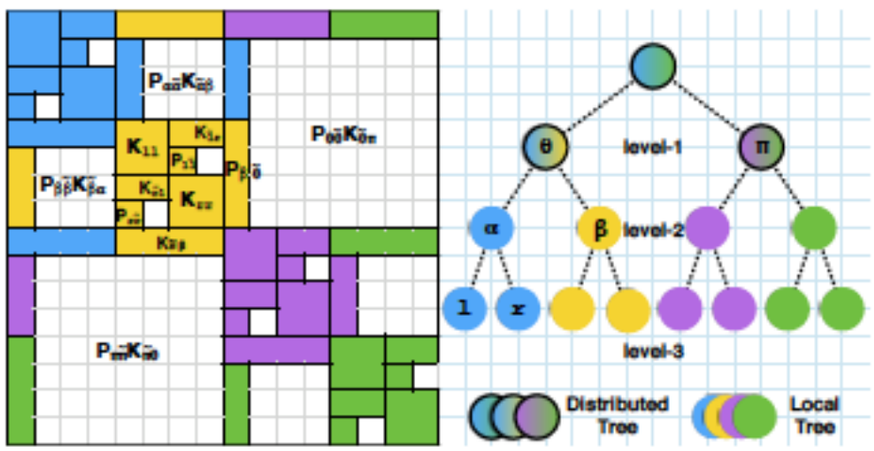
JHPCN

A. Ida
I. Yamazaki
S. Oshima
T. Hiraishi
T. Iwashita
K. Nakajima
T. Aoki
J. Dongarra

共有メモリ

分散メモリ

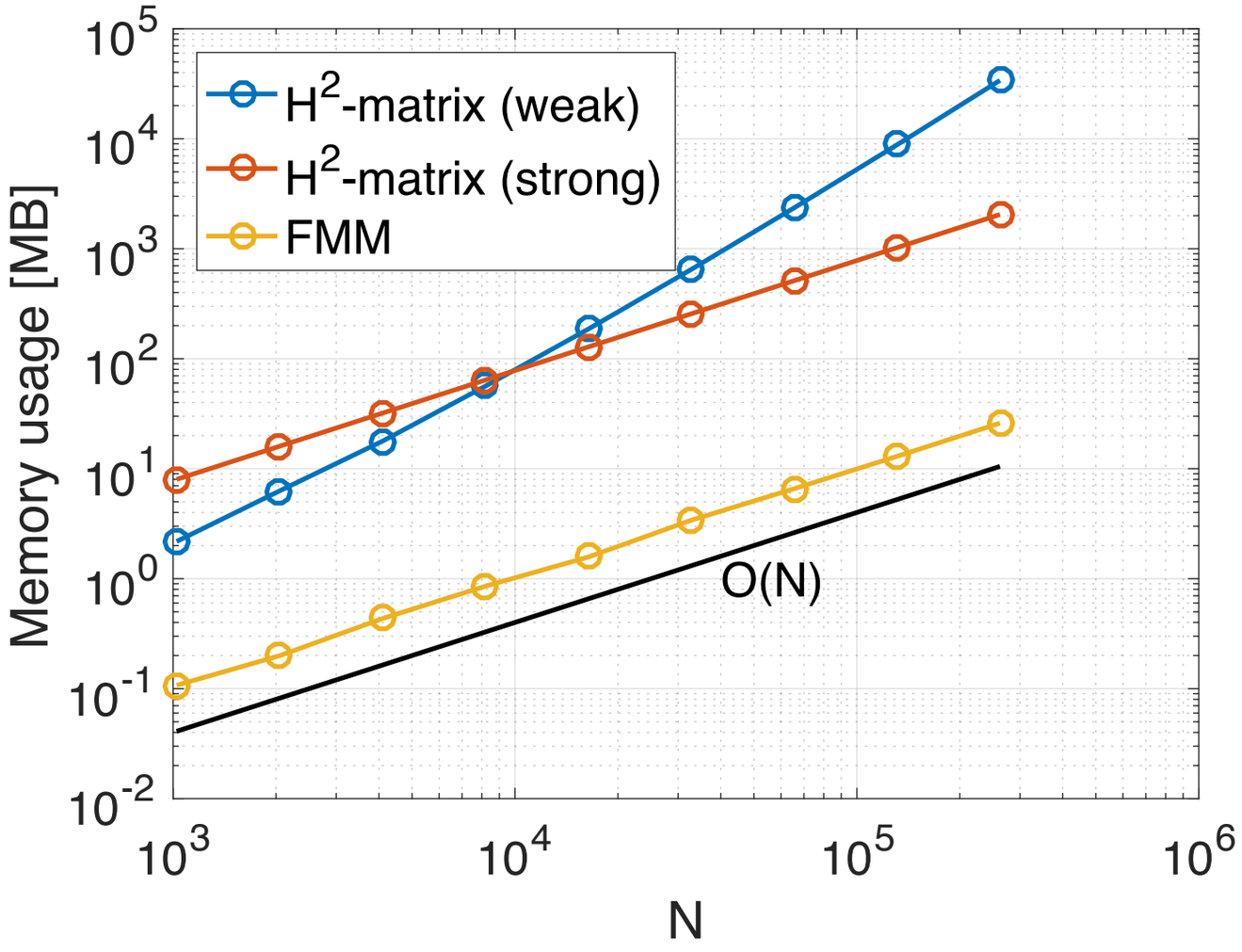
M. Izadi (2012), Hierarchical Matrix Techniques on Massively Parallel Computers
S. Wang (2013), Efficient Scalable Algorithms for Solving Dense Linear Systems with HSS
Y. Li (2016), Distributed-memory Hierarchical Interpolative Factorization
C. D. Yu (2016), INV-ASKIT: A Parallel Fast Direct Solver for Kernel Matrices



	Complexity	Concurrency
BLR	$\mathcal{O}(N^{4/3})$	High
H²(HSS)	$\mathcal{O}(N)$	Low

complexity-concurrency tradeoff

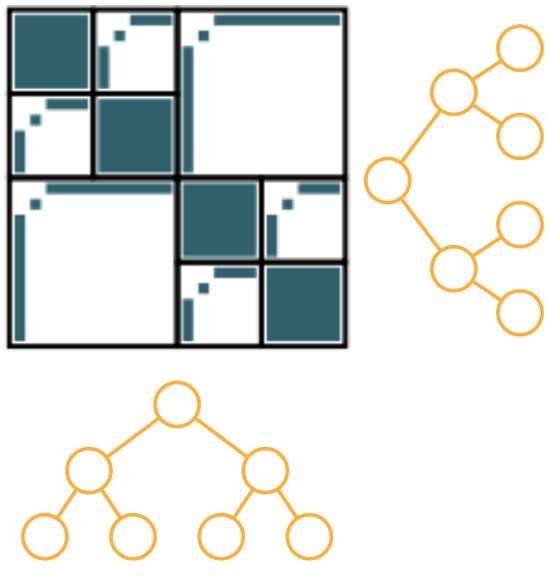
FMMとH²-matrixの関係



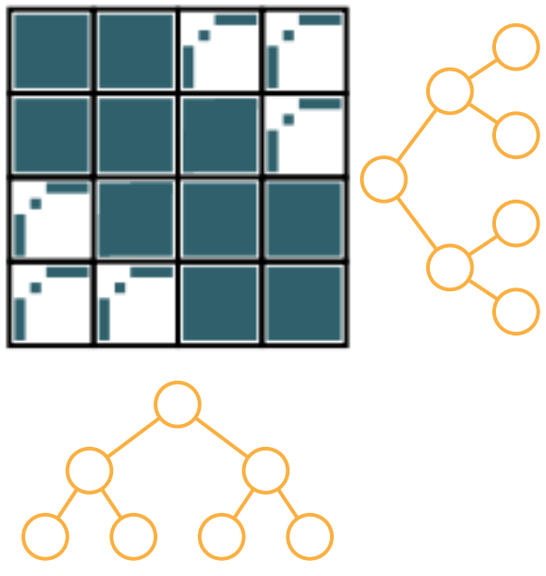
FMMはmatrix-freeのH²-matrix

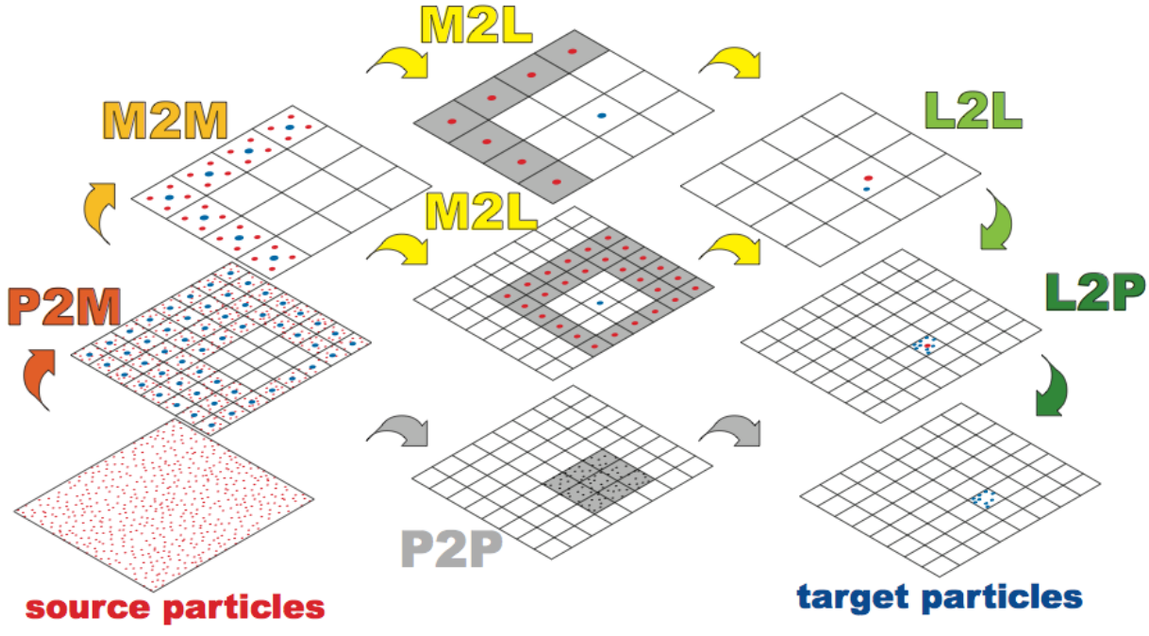
FMMは動径基底関数しか扱えない

weak admissibility



strong admissibility





My new C++ code

C++ Class

dense D

low-rank LR

hierarchical H

Operator overload

```
D=D+D
L=D*L
H=H+H
```

constructor destructor

```
Dense D(N,N)
LowRank LR(D,rank)
Hierarchical H(2,2)
```

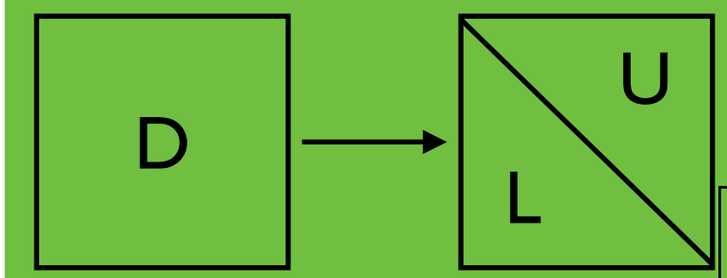
D	LR
LR	D

H	LR
LR	H

H	H
H	H

LU decomposition

D.getrf()



H.getrf()

U_{11} / L_{11} getrf()	U_{12}= $L_{11}^{-1}D_{12}$ trsm()
L_{21}= $D_{21}U_{11}^{-1}$ trsm()	D_{22}= D_{22}-$L_{21}U_{12}$ gemm()

D.trsm(L)

$L_{11}^{-1}D_{12}$

H_D.trsm(H_L)

$$\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1} \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} L_{11}^{-1}D_{11} & L_{22}^{-1}D_{12} \\ -L_{22}^{-1}L_{21}L_{11}^{-1}D_{11} + L_{22}^{-1}D_{21} & -L_{22}^{-1}L_{21}L_{11}^{-1}D_{12} + L_{22}^{-1}D_{22} \end{bmatrix}$$

D.gemm(L,U)

D_{22}-$L_{21}U_{12}$

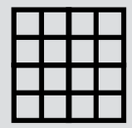
H_D.gemm(H_L,H_U)

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} -= \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}$$

Any H structure

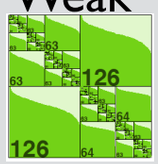
```
Hierarchical H(4,4)
```

BLR too

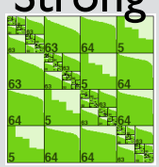


admissibility

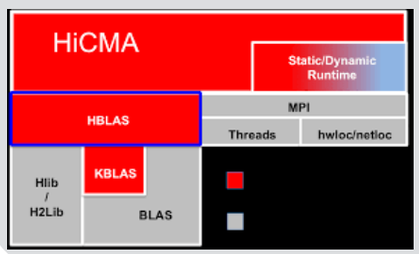
Weak



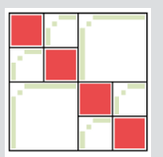
Strong

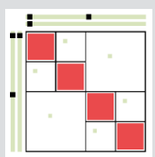


```
Hierarchical A(laplace1d, randx, N, N,
rank, nleaf, admis, nblocks, nblocks);
```



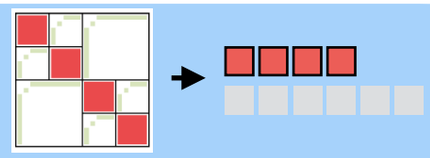
nested basis



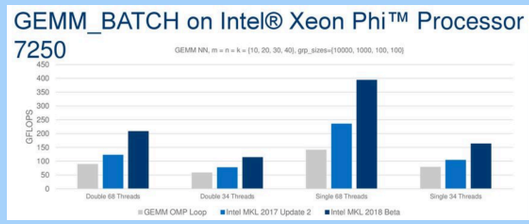


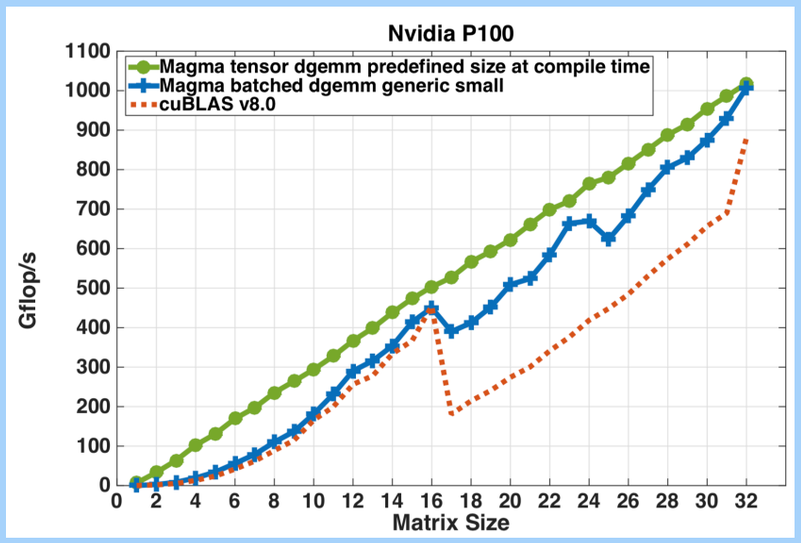
GPU implementation

Batch KBLAS



batch GEMM
batch GEMV





Contents lists available at ScienceDirect

Parallel Computing

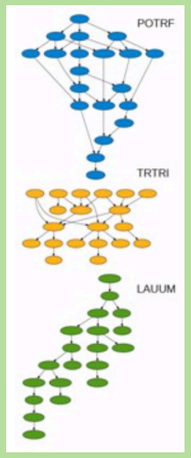
journal homepage: www.elsevier.com/locate/parco

Batched QR and SVD algorithms on GPUs with applications in hierarchical matrix compression

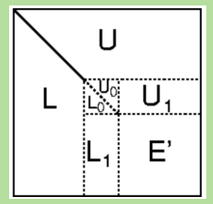
Wajih Halim Boukaram[a,*], George Turkiyyah[b], Hatem Ltaief[a], David E. Keyes[a]

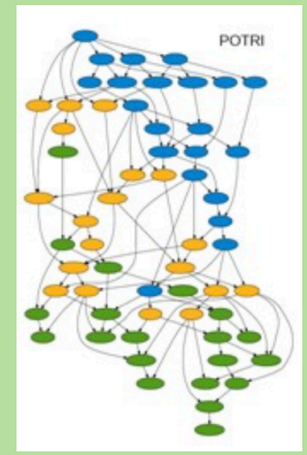
batch QR
batch SVD
batch RSVD
batch ACA (variable M,N,K)

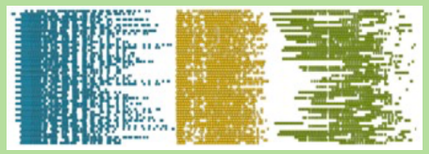
Runtime for LU

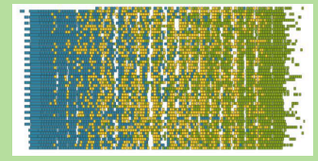


starPU
OmpSs

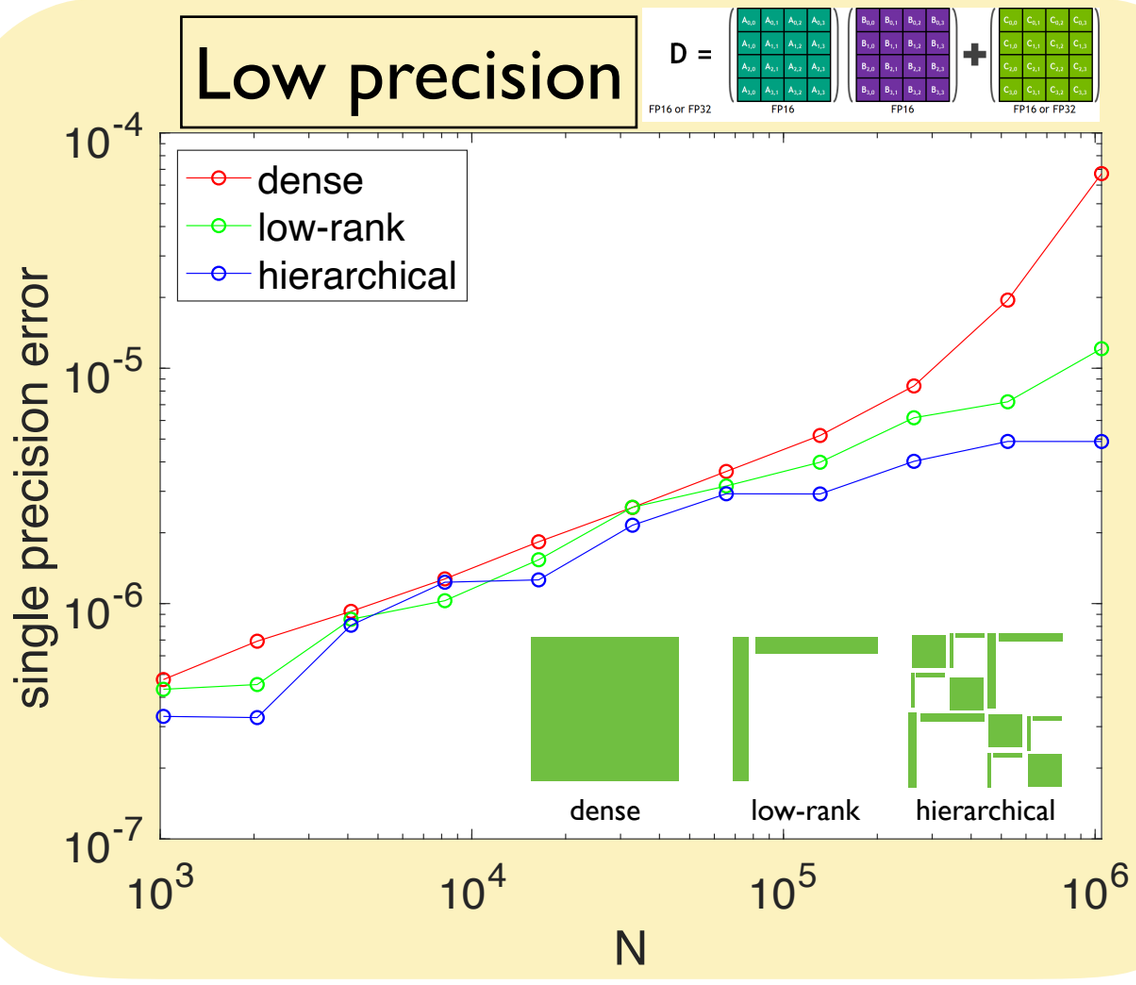






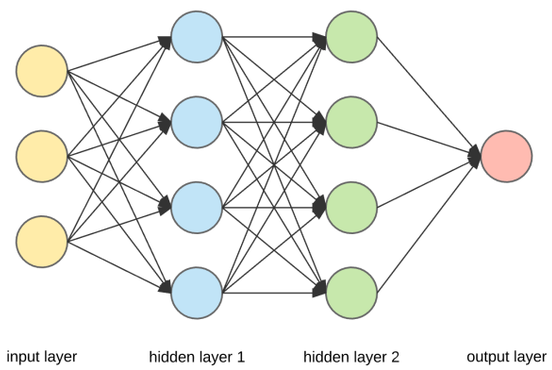


Low precision

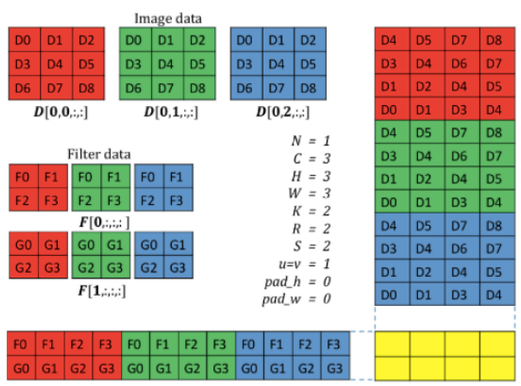


H行列は深層学習に適用できるか？

深層ニューラルネットは
密行列演算になる

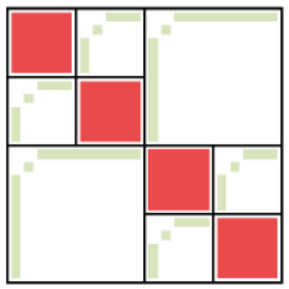


畳み込みNNは
密行列積になる



https://arxiv.org/abs/1410.0759

H行列は密行列の
高速近似解法

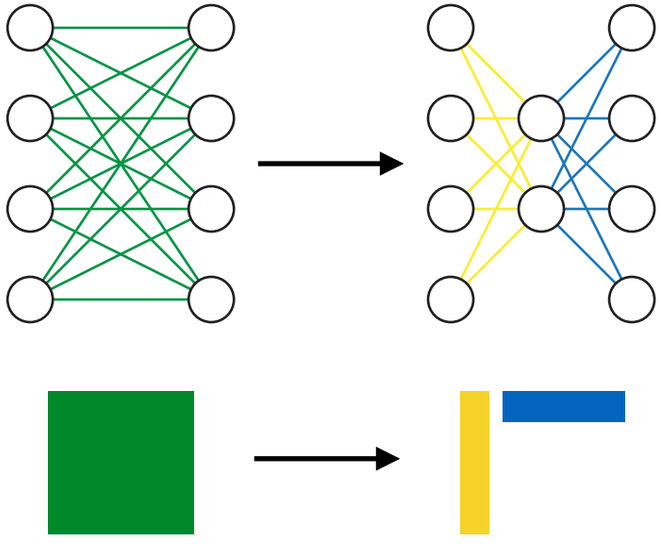


行列が非常に長細い

ブロックのランクが高い

圧縮した行列が１回しか使われない

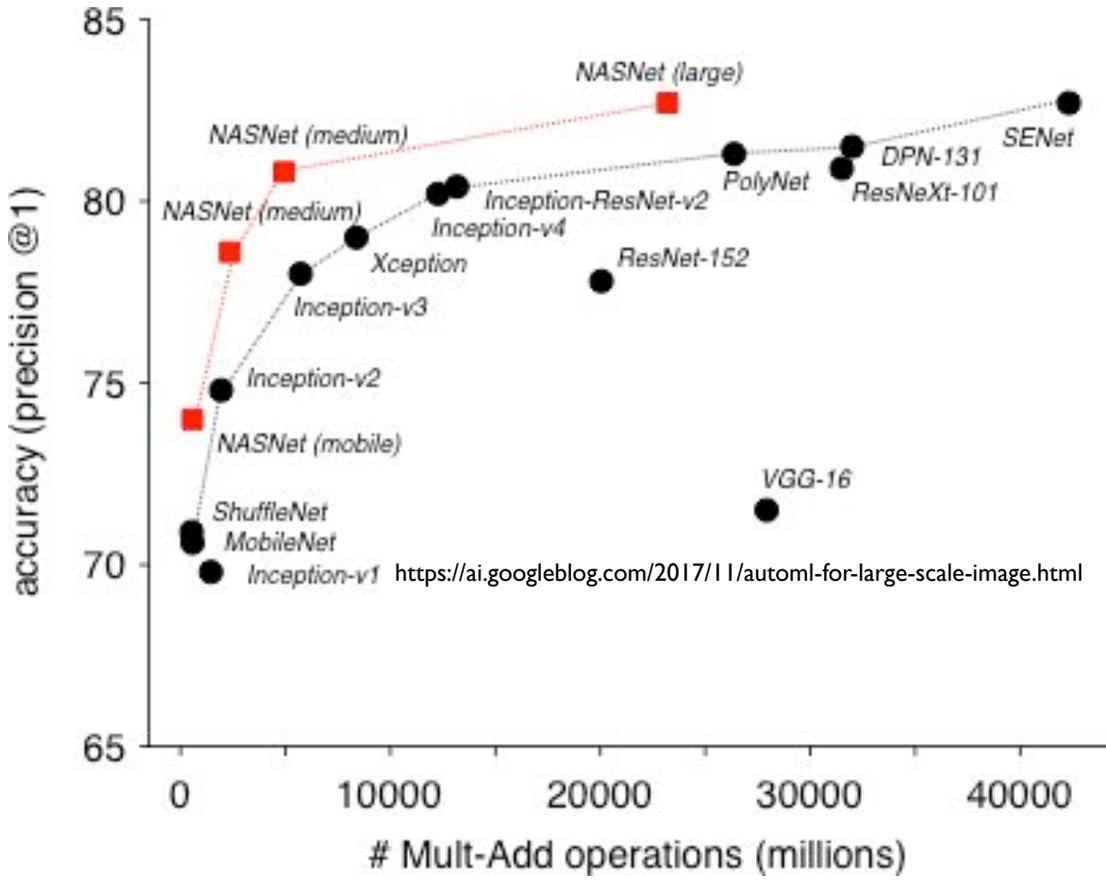
低ランク近似を用いてNN自体を圧縮



汎化性能と敵対的データに対する
堅牢性が維持できるのか？

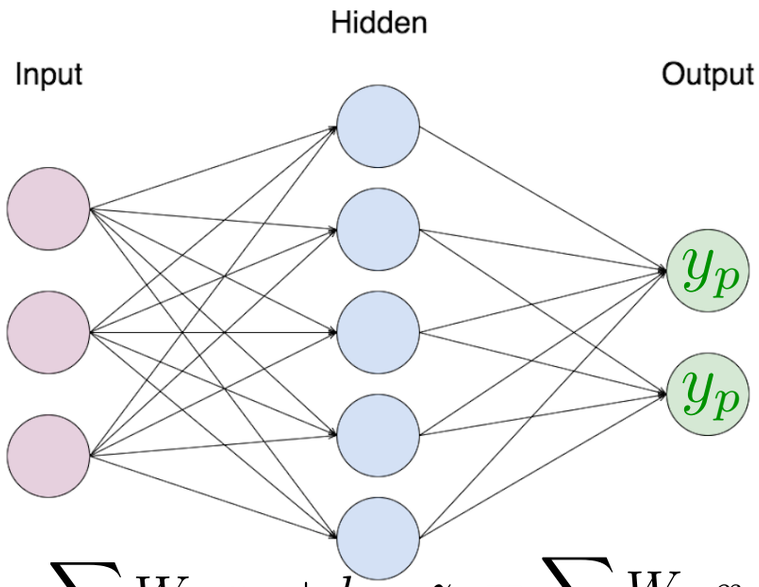
全結合層にしか有効でない

AutoMLの出現



https://ai.googleblog.com/2017/11/automl-for-large-scale-image.html

H行列とKronecker因子分解



y_q data (label)

y_p model (output)

1

0.9

0

0.1

$$z_i = \sum_j W_{ij} x_j + b \qquad z_i = \sum_j W_{ij} x_j + b$$

$$x_i = \max(0, z_i)$$

ReLU

$$y_{pi} = \frac{exp(z)}{\sum_i \exp(z_i)}$$

softmax

Cross entropy loss function

$$\begin{aligned} J(\theta) &= E_q(-\log p_\theta(y|\mathbf{x})) \\ &= \sum_{(\mathbf{x},y)} -q(y|\mathbf{x}) \log p_\theta(y|\mathbf{x}) \\ &= \sum_{\mathbf{x}} \{-y_q \log y_p - (1-y_q)\log(1-y_p)\} \\ &= \sum_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \theta) \end{aligned}$$

Back propagation

$$\frac{\partial J}{\partial W_{ij}} = \frac{\partial J}{\partial y_{pi}} \frac{\partial y_{pi}}{\partial z_i} \frac{\partial z_i}{\partial W_{ij}}$$

Kronecker Product

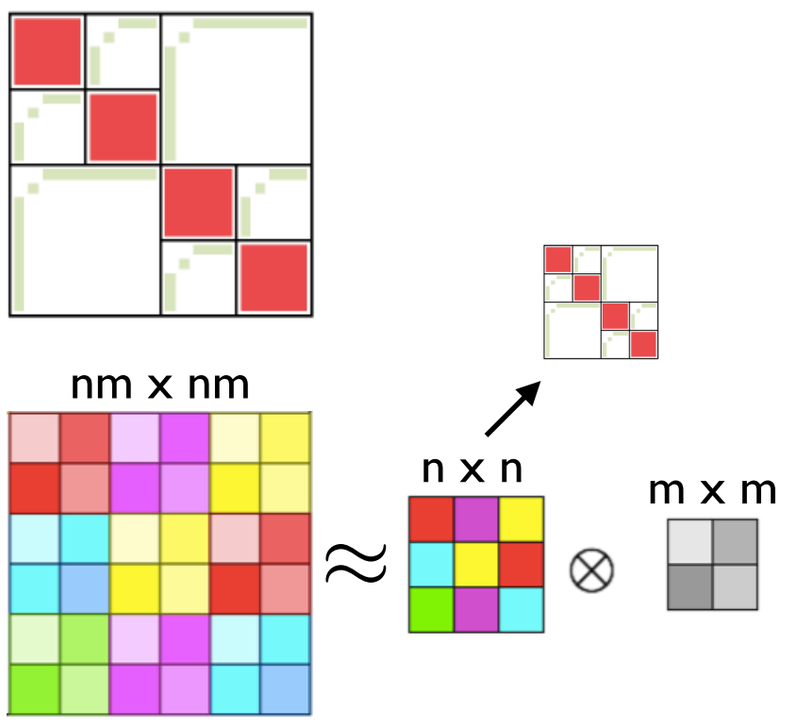
$$= \sum_{\mathbf{x}} \frac{\partial \mathcal{L}}{\partial W_{ij}} = \sum_{\mathbf{x}} \frac{\partial \mathcal{L}}{\partial z_i} \frac{z_i}{W_{ij}} = \sum_{\mathbf{x}} g_i a_j$$

Hierarchical Low-rank?

Fisher Matrix

100Mx100M

$$\frac{\partial J}{\partial \boldsymbol{\theta}}^T \frac{\partial J}{\partial \boldsymbol{\theta}}$$



Kronecker Factorization

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}}^T \frac{\partial J}{\partial \boldsymbol{\theta}}\right)^{-1} = \left\{\left(\sum_{\mathbf{x}} \mathbf{g} \otimes \mathbf{a}\right)^T \left(\sum_{\mathbf{x}} \mathbf{g} \otimes \mathbf{a}\right)\right\}^{-1}$$

$$\approx \left(\sum_{\mathbf{x}} \mathbf{g}^T \mathbf{g}\right)^{-1} \otimes \left(\sum_{\mathbf{x}} \mathbf{a}^T \mathbf{a}\right)^{-1}$$

$$= G^{-1} \otimes A^{-1}$$

まとめ

- 行列の低ランク構造は元となる計算点の幾何学的な配置に依存
- 疎行列は計算点の接続, ランクは計算点の距離に関係する
- FMM は matrix-free のH^2行列ーベクトル積
- 行列の形でFMMの演算を保存しておくことは左辺が多数ある場合に有効
- 階層的低ランク近似の手法間の違いは基底の共有と許容条件の違い
- Nullity theoremはAとA^{-1}の厳密なランク構造が同じことを保証するが数値的なランクは異なる
- 深層学習で扱う高次元の空間では低ランク近似よりもクロネッカー因子分解のほうが相性が良い