

Krylov 部分空間法のための 加法的 Schwarz 前処理

鈴木 厚¹

¹ 理研計算科学研究センター 大規模並列数値計算技術研究チーム
atsushi.suzuki.aj@a.riken.jp

Krylov 部分空間法のための前処理

前処理 $Q \simeq A^{-1}$, 残差ベクトル $\vec{r}$ に作用させる:

経済的 (軽い) あるいは 演算集約的 (重い)

右前処理の方程式

$$AQ(Q^{-1}\vec{x}) = \vec{b} \Leftrightarrow A\vec{x} = \vec{b} \quad \text{初期残差 } \vec{r}_0 = \vec{b} - A\vec{x}_0$$

前処理付き GMRES

次を満たす $\vec{x}'_m \in Q^{-1}x_0 + K_m(AQ, \vec{r}_0)$ を見付けよ

$$\|\vec{b} - AQ\vec{x}'_m\| \leq \|\vec{b} - AQ\vec{y}\| \quad \forall \vec{y} \in Q^{-1}x_0 + K_m(AQ, \vec{r}_0)$$

- ▶ 対角前処理 $[Q]_{ii} = |[A]_{ii}|^{-1}$
- ▶ SOR 前処理 $A = L + D + U$ パラメータ $0 < \omega < 2$,
 $Q = (\frac{D}{\omega} + U)^{-1} \frac{2-\omega}{\omega} D (\frac{D}{\omega} + L)^{-1}$
- ▶ 不完全分解, ILU 前処理
- ▶ multigrid 前処理 / 代数的 multigrid : $\Rightarrow$ HYPRE, gamg in PETSc
- ▶ 重なりのある領域分割を用いた部分問題での直接法による加法的 Schwarz 前処理 GenEO + 2 段階法 $\Rightarrow$ HPDDM

前処理付共役勾配法

- ▶ $A, Q \in \mathbb{R}^{N \times N}$, 対称正定値, $Q \sim A^{-1}$ $\vec{b} \in \mathbb{R}^N$,
- ▶ $\vec{x}_0$: 初期推定,
- ▶ $\vec{r}_0 := \vec{b} - A\vec{x}_0$: 初期残差.
- ▶ $K_n(QA, Q\vec{r}_0) := \text{span}[Q\vec{r}_0, QAQ\vec{r}_0, (QA)^2Q\vec{r}_0, \dots, (QA)^{n-1}Q\vec{r}_0]$

アルゴリズム (前処理付き CG 法)

$\vec{p}_0 = Q\vec{r}_0$.

do $m = 0, 1, \dots$

$$\alpha_m = (Q\vec{r}_m, \vec{r}_m) / (A\vec{p}_m, \vec{p}_m),$$

$$\vec{x}_{m+1} = \vec{x}_m + \alpha_m \vec{p}_m,$$

$$\vec{r}_{m+1} = \vec{r}_m - \alpha_m A\vec{p}_m,$$

if $\|\vec{r}_{m+1}\| < \epsilon$ exit loop.

$$\beta_m = (Q\vec{r}_{m+1}, \vec{r}_{m+1}) / (Q\vec{r}_m, \vec{r}_m),$$

$$\vec{p}_{m+1} = Q\vec{r}_{m+1} + \beta_m \vec{p}_m.$$

命題 $1 \leq m \leq n_0$ に対して

- ▶ $(\vec{r}_m, \vec{z}) = 0 \quad \forall \vec{z} \in K_m(QA, Q\vec{r}_0)$
- ▶ $(A\vec{p}_m, \vec{z}) = 0 \quad \forall \vec{z} \in K_m(QA, Q\vec{r}_0)$
- ▶ $\text{span}[Q\vec{r}_0, Q\vec{r}_1, \dots, Q\vec{r}_m] = \text{span}[\vec{p}_0, \vec{p}_1, \dots, \vec{p}_m] = K_{m+1}(QA, Q\vec{r}_0)$

変分問題の解を続けて計算する

do $m = 1, 2, \dots, n_0$

$$\text{find } \vec{x} \in \vec{x}_0 + K_m(QA, Q\vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(QA, Q\vec{r}_0)$$

前処理付き Krylov 部分空間法: 1/2

- ▶ $A \in \mathbb{R}^{N \times N}$: 正則 (逆を持つ), $\vec{b} \in \mathbb{R}^N$,
- ▶ $Q \in \mathbb{R}^{N \times N}$: 前処理行列, $Q^{-1} \sim A$,
- ▶ $\vec{x}_0$: 初期推定,
- ▶ $\vec{r}_0 := \vec{b} - A\vec{x}_0$: 初期残差.

Krylov 部分空間

$$K_n(QA, Q\vec{r}_0) := \text{span}[Q\vec{r}_0, QAQ\vec{r}_0, (QA)^2Q\vec{r}_0, \dots, (QA)^{n-1}Q\vec{r}_0]$$

n_0 : 不変 Krylov 部分空間の最小の次元

$$\text{▶ } n_0 := \min_n \{K_n(QA, Q\vec{r}_0) = K_{n+1}(QA, Q\vec{r}_0)\}$$

$$K_1(QA, Q\vec{r}_0) \subset K_2(QA, Q\vec{r}_0) \subset \dots \subset K_{n_0}(QA, Q\vec{r}_0) = K_{n_0+1}(QA, Q\vec{r}_0) = \dots$$

$$\dim K_m(A, \vec{r}_0) = m \quad 1 \leq m \leq n_0$$

定理

$\vec{x}$: を線形方程式 $A\vec{x} = \vec{b}$ の解とする $\Rightarrow \vec{x} \in \vec{x}_0 + K_{n_0}(QA, Q\vec{r}_0)$.

定理

次の変分問題 (vP) を考える

$$\vec{x} \in \vec{x}_0 + K_{n_0}(QA, Q\vec{r}_0) \text{ は次を満たす } (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_{n_0}(QA, Q\vec{r}_0)$$

この問題は一意的な解を持ち, 連立一次方程式の解 $A\vec{x} = \vec{b}$ に等しい

変分問題の解を続けて計算する

do $m = 1, 2, \dots, n_0$

$$\text{find } \vec{x} \in \vec{x}_0 + K_m(QA, Q\vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(QA, Q\vec{r}_0)$$

- ▶ Conjugate Gradient (CG) 法 $\Leftarrow A$: 正定値対称
- ▶ Full Orthogonalization Method (FOM) $\Leftarrow A$: 強圧的

前処理付き Kyrlov 部分空間法: 2/2

GMRES に対して, 元の連立一次方程式を変形する.

- ▶ 左前処理 $(QA)\vec{x} = Q\vec{b}$
- ▶ 右前処理 $(AQ)\vec{z} = \vec{b}, \quad \vec{x} = Q\vec{z}$

左前処理付 GMRES

find $\vec{x}_m \in \vec{x}_0 + K_m(QA, Q\vec{r}_0)$

$$\|Q\vec{b} - (QA)\vec{x}_m\| \leq \|Q\vec{b} - (QA)\vec{y}\| \quad \forall \vec{y} \in \vec{x}_0 + K_m(QA, Q\vec{r}_0)$$

$$\begin{aligned} K_m(QA, Q\vec{r}_0) &= \text{span}[Q\vec{r}_0, (QA)Q\vec{r}_0, \dots, (QA)^{m-1}Q\vec{r}_0] \\ &= Q\text{span}[\vec{r}_0, (AQ)\vec{r}_0, \dots, (AQ)^{m-1}\vec{r}_0] = QK_m(AQ, \vec{r}_0) \end{aligned}$$

右前処理付 GMRES

find $\vec{x}_m \in \vec{x}_0 + QK_m(AQ, \vec{r}_0) = \vec{x}_0 + K_m(QA, Q\vec{r}_0)$

$$\|\vec{b} - (AQ)Q^{-1}\vec{x}_m\| \leq \|\vec{b} - (AQ)Q^{-1}\vec{y}\| \quad \forall \vec{y} \in \vec{x}_0 + QK_m(AQ, \vec{r}_0)$$

$$\Leftrightarrow \|\vec{b} - A\vec{x}_m\| \leq \|\vec{b} - A\vec{y}\| \quad \forall \vec{y} \in \vec{x}_0 + K_m(QA, Q\vec{r}_0)$$

flexible GMRES

Flexible GMRES は右前処理付き GMRES の拡張

- ▶ $A \in \mathbb{R}^{N \times N}$: 正則 (逆を持つ) $\vec{b} \in \mathbb{R}^N$,
- ▶ Q_m : m -ステップでの右前処理
- ▶ $\vec{x}_0$: 初期推定,
- ▶ $\vec{r}_0 := \vec{b} - A\vec{x}_0$: 初期残差, $\beta = \|\vec{r}_0\|$, $\vec{v}_1 = \vec{r}_0/\beta$.

Arnoldi プロセスに修正を加える

アルゴリズム (flexible GMRES) 右前処理付 GMRES

do $j = 1, 2, \dots, m$

$$\vec{z}_j = Q_j \vec{v}_j$$

$$\vec{w} = A\vec{z}_j$$

do $i = 1, \dots, j$

$$h_{i,j} := (\vec{w}, \vec{v}_i)$$

$$\vec{w} := \vec{w} - h_{i,j} \vec{v}_i$$

$$h_{j+1,j} := \|\vec{w}\|$$

$$\vec{v}_{j+1} = \vec{w}/h_{j+1,j}$$

$$Z_m := [\vec{z}_1, \dots, \vec{z}_m]$$

$$\vec{\eta}_m = \operatorname{argmin}_{\vec{\eta}} \|\beta \vec{e}_{(m+1)}^{(1)} - \overline{H}_m \vec{\eta}\|,$$

$$\vec{x}_m = \vec{x}_0 + Z_m \vec{\eta}_m.$$

$$AQ[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m] = V_{m+1} \overline{H}_m$$

flexible GMRES

$$A[Q_1 \vec{v}_1, Q_2 \vec{v}_2, \dots, Q_m \vec{v}_m] = V_{m+1} \overline{H}_m$$

$$\begin{aligned} \vec{b} - A(\vec{x}_0 + Z_m \vec{\eta}) &= \vec{r}_0 - AZ_m \vec{\eta} \\ &= V_{m+1} (\beta \vec{e}_{m+1}^{(1)} - \overline{H}_m \vec{\eta}) \end{aligned}$$

$V_{m+1}^T V_{m+1} = I_{m+1}$ より残差ベクトルの長さ
の計算を V_m で変換して

$$\|\vec{b} - A(\vec{x}_0 + Z_m \vec{\eta}_m)\| \leq \|\beta \vec{e}_{m+1}^{(1)} - \overline{H}_m \vec{\eta}\| \quad \forall \vec{\eta} \in \mathbb{R}^m$$

$\operatorname{span}[Q_1 \vec{v}_1, Q_2 \vec{v}_2, \dots, Q_m \vec{v}_m]$ は Q_j が全て共通して Q である場合を除き,
Krylov 部分空間ではない.

Schwarz による前処理

添字による行列の重なりのある分解 $\Lambda = \bigcup_{p=1}^P \Lambda_p$, $\Lambda_p \cap \Lambda_q \neq \emptyset$

- ▶ R_p : 全自由度から部分行列への制限: $\Lambda \rightarrow \Lambda_p$
- ▶ D_p : 離散的な単位の分解

$$\sum_{p=1}^P R_p^T D_p R_p = I_N,$$

$$[D_p]_{kk} = \begin{cases} 1 & k \in \Lambda_p, k \notin \Lambda_q, \forall q \neq p, \\ 1/\#\{p; k \in \Lambda_p\} & \text{それ以外} \end{cases}$$

ASM 前処理

$$M_{\text{ASM}}^{-1} = \sum_{p=1}^P R_p^T (R_p A R_p^T)^{-1} R_p$$

ASM は不動点反復法としては収束しない, しかしながら, M_{ASM}^{-1} は対称で CG 法の前処理として機能する.

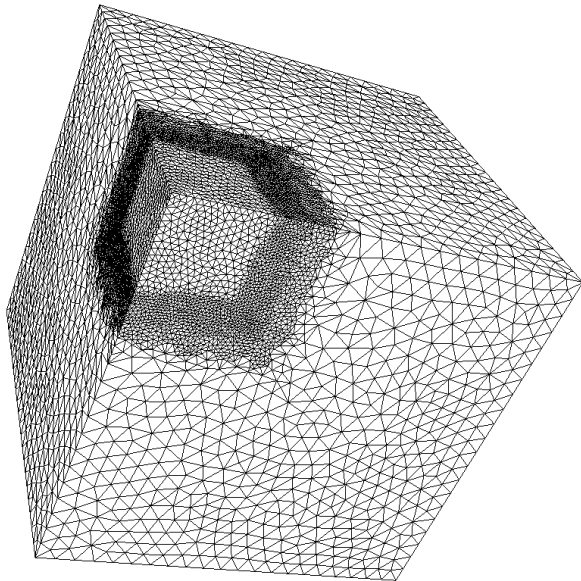
RAS preconditioner

$$M_{\text{RAS}}^{-1} = \sum_{p=1}^P R_p^T D_p (R_p A R_p^T)^{-1} R_p$$

RAS は収束するが M_{RAS}^{-1} は非対称のため GMRES 法の前処理として機能する.

数値例 : 1/3

tetgen と FreeFEM によって生成された非構造メッシュP2 有限要素,
 $N = 677,163$, $nnz = 28,853,844$



数値例 : 2/3

停止条件 : $\leq 10^{-12}$ を相対誤差

Navier 方程式 : P2 有限要素により離散化された弾性体問題

$N = 750,141$, $nnz = 31,214,610$: 立方体の構造メッシュ (32^3 節点)

Navier3D.32.P2

重なり幅	反復回数	経過時間 (秒)			
		全体	LU-分解	反復計算	誤差
1	59	81.397	44.749	36.647	4.41361e-12
2	41	87.871	55.339	32.532	1.70397e-12
3	33	106.13	72.975	33.158	1.45462e-12
ILU(0)	186	58.347	16.146	42.201	4.66513e-11

$N = 677,163$, $nnz = 28,853,844$: 要素細分による非構造メッシュ

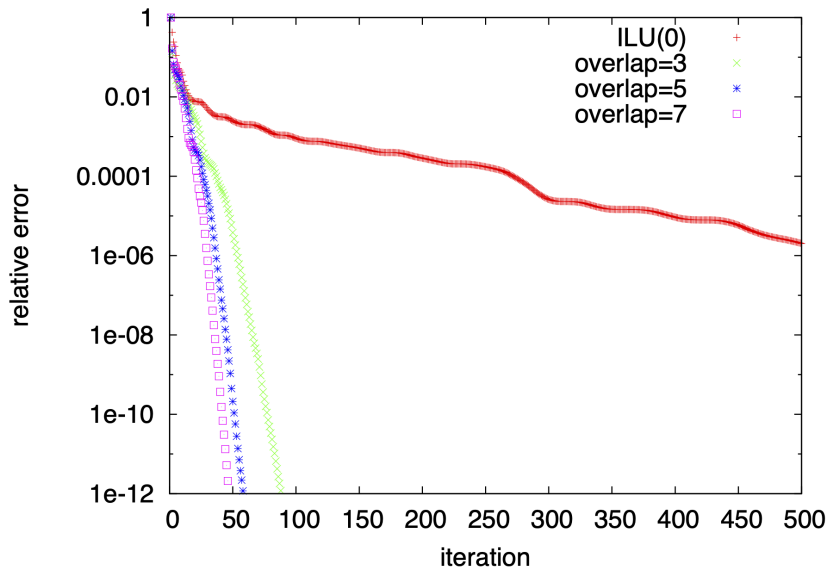
Navier3DmeshP2

重なり幅	反復回数	経過時間 (秒)			
		全体	LU-分解	反復計算	誤差
1	88	77.978	29.413	48.565	4.23959e-12
2	58	87.975	45.055	42.920	4.62264e-12
3	46	111.68	69.063	47.112	8.12039e-12
ILU(0)	500	209.91	30.791	179.12	1.12751e-2

ILU(0) 前処理は非構造メッシュの問題に対しては十分ではない

数值例 : 3/3

収束履歴



直接法の概要

係数行列 A は $\Pi_L^T L D U \Pi_R$ と分解される

- ▶ Π_L, Π_R : 全軸選択を実現する左/右置換行列
- ▶ 対称軸選択 $\Pi_L = \Pi_R = \Pi$ に ε -擾乱を追加するか 2×2 軸選択を更に加えるか, あるいはそのみ追加するか

置換行列 Π は次の二つから成る

- ▶ シンボリック分解によりフィルインを最小化する添字並べ替え
- ▶ 分解途中に対角成分の絶対値最大を見付けて動的に行と列を並べ替え

直接法の 3 つのフェーズ

- ▶ シンボリック分解
フィルインを最小化し並列化を実現するためのマルチフロンタル法, 置換行列は Π_S
- ▶ 数値分解
置換行列を Π_N とする動的な軸選択
- ▶ 前進消去/後退代入

$$\Pi^T L D U \Pi \vec{x} = \vec{b}$$

$$\vec{z} = L^{-1} \Pi \vec{b} \quad \text{前進消去 } L \vec{z} = \Pi \vec{b}$$

$$\vec{y} = D^{-1} \vec{z}$$

$$\vec{x} = \Pi^T U^{-1} \vec{y} \quad \text{後退代入 } U \Pi \vec{x} = \vec{y}$$

疎行列直接法ソフトウェアパッケージ

ソフトウェア	並列環境	消去方法	データ管理	軸選択	核判定
UMFPACK	—	multi-frontal	static	yes	no
SuperLU_MT	shared	super-nodal	dynamic	yes	no
Pardiso	shared/dist.	super-nodal	dynamic	yes + $\sqrt{\epsilon}$ -p.	no
SuperLU_DIST	distributed	super-nodal	static	no, $\sqrt{\epsilon}$ -p.	no
MUMPS	dist./shared	multi-frontal	dynamic	yes	yes
Dissection	shared	multi-frontal	static	yes	yes

T. A. Davis, I. S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices,
ACM Trans. Math. Software, 25 (1999), 1–20.

J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, J. W. H. Liu.
 A supernodal approach to sparse partial pivoting,
SIAM J. Matrix Anal. Appl., 20 (1999), 720–755.

O. Schenk, K. Gärtner. Solving unsymmetric sparse systems of liner equations with PARDISO,
Future Generation of Computer Systems, 20 (2004), 475–487.

X. S. Li, J. W. Demmel. SuperLU_DIST : A scalable distributed-memory sparse direct solver for unsymmetric linear systems,
ACM Trans. Math. Software, 29 (2003), 110–140.

P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent. Mutlifrontal parallel distributed symmetric and unsymmetric solvers,
Comput. Methods Appl. Mech. and Engrg., 184 (2000) 501–520.

A. Suzuki, F.-X. Roux, A dissection solver with kernel detection for symmetric finite element matrices on shared memory computers,
Int. J. Numer. Meth. in Engng, 100 (2014) 136–164.

疎行列の変数添字の並べ替え

疎行列の変数添字は並べ替える必要がある

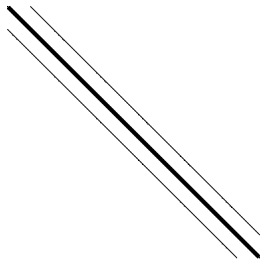
- ▶ フィルインを最小化するため
- ▶ 分解の並列度を高めるため
- ▶ ブロック構造のサイズの粒度を大きくするため

→ マルチフロント

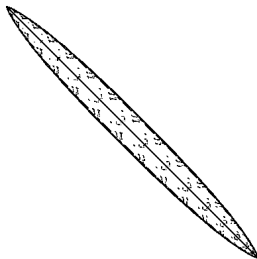
→ スーパーノード

例:

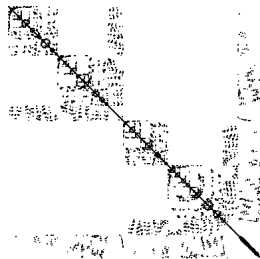
3次元 Poisson 方程式の 7 重対角行列, 合計 11^3 節点



もとの行列
帯行列

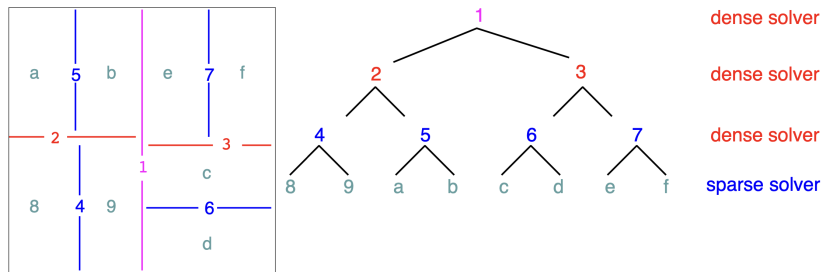


reverse Cuthill-McKee
逐次計算



nested-dissection
並列計算

グラフ分割による nested dissection 法



A. George. Numerical experiments using dissection methods to solve n by n grid problems. SIAM J. Num. Anal. 14 (1977),161–179.

software package:

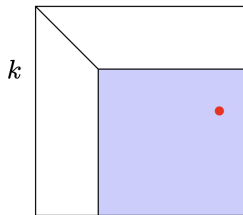
METIS : **V. Kumar, G. Karypis**, A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20 (1998) 359–392.

SCOTCH : **F. Pellegrini J. Roman J, P. Amestoy**, Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering. Concurrency: Pract. Exper. 12 (2000) 69–84.

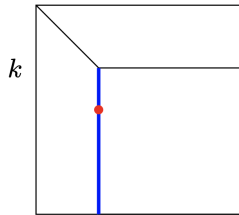
► 二分木の各々のノードは並列に計算できる ⇐ マルチフロント

軸選択

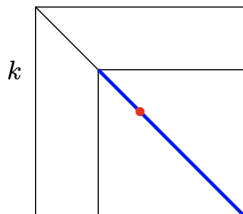
full pivoting : $A = \Pi_L^T L U \Pi_R$
 find $\max_{k < i, j \leq n} |A(i, j)|$



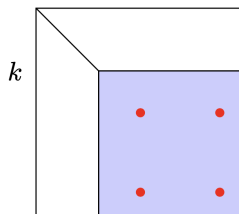
partial pivoting : $A = \Pi L U$
 find $\max_{k < i \leq n} |A(i, k)|$



symmetric pivoting : $A = \Pi^T L D U \Pi$
 find $\max_{k < i \leq n} |A(k, k)|$



2×2 pivoting : $A = \Pi^T L \tilde{D} U \Pi$
 find $\max_{k < i, j \leq n} \det \begin{vmatrix} A(i, i) & A(i, j) \\ A(j, i) & A(j, j) \end{vmatrix}$



sym. pivoting is mathematically not always possible

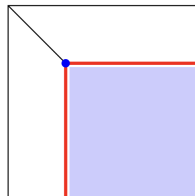
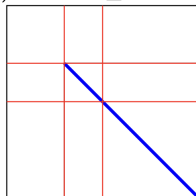
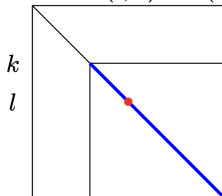
rank-1 更新による LDU 分解

$$\begin{aligned}\begin{bmatrix} a_{11} & \beta_1^T \\ \alpha_1 & A_{22} \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ \alpha_1 a_{11}^{-1} & I_2 \end{bmatrix} \begin{bmatrix} a_{11} & \beta_1^T \\ 0 & S_{22} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ \alpha_1 a_{11}^{-1} & I_2 \end{bmatrix} \begin{bmatrix} a_{11} & 0 \\ 0 & S_{22} \end{bmatrix} \begin{bmatrix} 1 & a_{11}^{-1} \beta_1^T \\ 0 & I_2 \end{bmatrix}\end{aligned}$$

Schur 補行列 $S_{22} = A_{22} - \alpha_1 a_{11}^{-1} \beta_1^T$ は rank-1 更新 によって計算でき, BLAS ライブラリーの **dger** を用いる

対称軸選択による LDU -分解アルゴリズム

```
do  $k = 1, \dots, N$   
  find  $\max |A(l, l)|$  with  $k < l \leq n$ ,  
  exchange rows and columns :  $A(k, *) \leftrightarrow A(l, *)$ ,  $A(*, k) \leftrightarrow A(*, l)$ .  
  dscal  $A(k, j) /= A(k, k)$      $k < j \leq N$ ,  
  dger  $A(i, j) -= A(i, k) A(k, j)$      $k < i, j \leq N$ ,  
  dscal  $A(i, k) /= A(k, k)$      $k < i \leq N$ .
```



不定値行列向けの 2×2 軸選択

対称行列

$$\begin{bmatrix} \frac{1}{4} & \frac{5}{4} & \frac{1}{2} \\ \frac{5}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 5 & \frac{1}{3} & \\ 2 & & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & & \\ & -6 & \\ & & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 & 5 & 2 \\ & 1 & \frac{1}{3} \\ & & 1 \end{bmatrix}$$

対称軸選択を作用すると

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{4} & \frac{5}{4} \\ \frac{1}{2} & \frac{5}{4} & \frac{1}{4} \end{bmatrix} = \begin{bmatrix} 1 & & \\ \frac{1}{2} & 1 & \\ \frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 0 & 1 \\ & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ & 1 & 0 \\ & & 1 \end{bmatrix}$$

対角の絶対値最大の要素を見付ける軸選択は不定値行列では失敗することがある。

解決方法として 2×2 軸選択を用いる。

対称な不定値行列に対して 1×1 と 2×2 軸選択を組み合わせる手法が提案されている。

J. R. Bunch, L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems, *Math. Comput.*, 31 (1977) 163–179.

正則化手法の一つ

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & \begin{matrix} 0 & \alpha \\ \beta & 0 \end{matrix} \end{bmatrix} \rightarrow \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & \begin{matrix} \sqrt{\varepsilon} & \alpha \\ \beta & 0 \end{matrix} \end{bmatrix}$$

- ▶ 方程式を変形してしまうので異なる問題を解いていることになるため反復改良手法により解の精度を向上させる
- ▶ ユーザーは非対称行列に対しては適切なパラメータの大きさを選択できる/しなければならない Pardiso のデフォルト値は 10^{-13}

C/C++ からの Pardiso の利用

```
MKL_INT *ptrow = new MKL_INT[n + 1]; // CSR data
MKL_INT *indcol = new MKL_INT[nnz];
double *coef = new double[nnz];
double *x = new double[n]; // solution
double *y = new double[n]; // RHS
void *pt[64]; // to keep internal pointers
for (int i = 0; i < 64; i++)
    pt[i] = (void *)0; // zero clear
MKL_INT *iparm = new MKL_INT[64]; // parameters!
MKL_INT mtype = 11; // structurally symmetric
MKL_INT nrhs = 1;
MKL_INT phase;
MKL_INT maxfct = 1, mnum = 1, msglvl = 1, error;
MKL_INT idum; // dummy pointer instead of user
                // providing permutation
phase = 11; // symbolic factorization
pardiso(pt, &maxfct, &mnum, &mtype, &phase, &n,
        (void *)coef, ptrow, indcol, &idum, &nrhs,
        iparm, &msglvl, (void *)y, (void *)x,
        &error);
phase = 22; // numeric factorization
phase = 33; // Fw/Bw substitution
phase = -1; // free working data
```

C/C++ からの MUMPS の利用

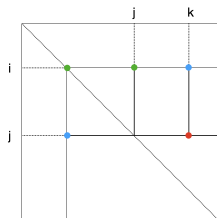
```
MUMPS_INT *irow = new MUMPS_INT[nnz];
MUMPS_INT *jcol = new MUMPS_INT[nnz];
double *coef = new double[nnz];
double *x = new double[n]; // solution&RHS
DMUMPS_STRUC_C id;
id.job = (-1); // job init
id.par = 1;
id.sym = isSym ? 2 : 0;
id.comm_fortran = USE_COMM_WORLD; // dummy MPI communicator
dmumps_c(&id);
id.job = 1; // symbolic factorization
id.n = n; id.nz = nnz; id.irn = irow; id.jcn = jcol;
// id.icntl[] set parameters
dmumps_c(&id);
id.job = 2; // numeric factorization
id.a = coef;
dmumps_c(&id);
id.job = 3; // Fw/Bw substitution
id.nrhs = 1;
id.rhs = x;
dmumps_c(&id);
id.job = -2; // free working data
```

不完全 LU 分解による前処理

right-looking 版

A と同じ非零パターンでの rank-1 更新 $ILU(0)$

A^{k+1} と同じ非零パターンでの rank-1 更新 $ILU(k)$



```
for (i = 0; i < nrow; i++) {  
    ia = diag[i];  
    vdiag = 1.0 / coef[ia]; coef[ia] = vdiag;  
    for (ia = diag[i] + 1; ia < ptrow[i + 1]; ++ia) {  
        j = indcol[ia];  
        for (j = ptrow[j]; j < ptrow[j + 1]; ++j) {  
            k = indcol[j];  
            if (k < i) continue;  
            if (k == i) { coef[j] *= vdiag; val = coef[j]; }  
            else {  
                for (ic = diag[i] + 1; ic < ptrow[i + 1]; ++ic) {  
                    if (k == indcol[ic]) coef[j] -= val * coef[ic];  
                }  
            }  
        }  
    }  
}
```

Multigrid (多重格子) 法 : 1/2

Finite Elements: Theory, Fast Solvers, and Applications in Elasticity Theory,
3rd ed. D. Braess Cambridge University Press, 2007, Chap. 5

平滑化 : $\omega > 0$ をパラメータとする Richardson 緩和

$$x \mapsto Sx = x - \omega(A_h x - b_h)$$

二つの階層を構成する入れ子の格子での解 S_h と S_{2h}
 k -ステップ目での **二つの格子での反復** : $u_h^k \mapsto u_h^{k+1}$

1. 平滑化 : ν 回の反復操作

$$\hat{u}_h^k = S^\nu u^k$$

2. 粗空間での解の補正

$$\text{find } v \in S_{2h} \quad (A_h \hat{u}_h^k + v, \varphi) = (b_h, \varphi) \quad \forall \varphi \in S_{2h}$$

$$u_h^{k+1} := \hat{u}_h^k + v$$

- ▶ 平滑化は解の高周波成分をダンプして残差が低周波成分で表現できるようにする

Jacobi, Richardson, SOR, 対称 SOR など.

Multigrid (多重格子) 法 : 2/2

入れ子の構造を持つ部分空間列 $V_0 \subset V_1 \subset \cdots \subset V_{l_{\max}}$

k -ステップでの**多重格子反復** $\text{MGM}_l : u^{l,k} \mapsto u^{l,k+1}$

1. 事前平滑化 : ν_1 反復の平滑化を実行する

$$\hat{u}^{l,k} = S^{\nu_1} u^{l,k+1}$$

2. 粗空間での解の補正

$$\text{find } v \in V_{l-1} \quad (A_l \hat{u}^{l,k} + v, \varphi) = (b_h, \varphi) \quad \forall \varphi \in V_{l-1}$$

$l = 1$ の時 $v^0 \in V_0$ は厳密に求める (直接法解法などで)

$l > 1$ の時 v^{l-1} は初期値を $u^{l-1,0} = 0$ とする μ -ステップの MGM_{l-1} 操作で求める

$$\tilde{u}^{l,k} := \hat{u}^{l,k} + v^{l-1}$$

3. 事後平滑化 : ν_2 反復の平滑化を実行する

$$\hat{u}^{l,k+1} = S^{\nu_2} \tilde{u}^{l,k}$$

$\mu = 1 \Rightarrow \text{V-サイクル}$

$\mu = 2 \Rightarrow \text{W-サイクル}$

- ▶ 階層を持つ格子から構成された入れ子の部分空間
→ geometrical multigrid (幾何的多重格子法)
- ▶ 疎行列構造から構成された入れ子でない部分空間
→ algebraic multigrid (代数的多重格子法)