本講習会の目的

- 本講習会では，有限要素法による熱伝導解析プログラムをMPIを使用して並列化するための手順，特に並列分散データ構造に関する考え方を中心に説明します。
- お手元のプログラムをMPIによって「並列化」したいと考えている方は是非受講をお勧めいたします。

- https://www.cc.u-tokyo.ac.jp/events/lectures/144/
- http://nkl.cc.u-tokyo.ac.jp/pFEM/

背 景

- 大規模並列計算機
- ポストペタスケール，エクサスケール時代の到来
- 並列プログラミングの重要性
  - 体系的に学習する機会は少ない
  - アプリケーションの観点から教えているマテリアルは少ない（ほとんど無い）
- 計算機アーキテクチャの多様化，複雑化
  - Multicore，Manycore，GPU
  - MPI+X(+Y)
    - X：OpenMP, CUDA, OpenACC, OpenCL
    - Y：Vectorization
- Co-Designの重要性

本講習会の位置づけ

- 有限要素法，MPI+OpenMP
  - 疎行列，共役勾配法
- 計算機科学の人がアプリケーションの勉強をするのにも適している
- 本来，まる5日かける内容を1日で実施
  - アカウントは一ヶ月有効なのでよく復習してください。
- 前処理付き反復法
  - 実は非常に重要な研究開発分野なのだが，本講習会では簡単なもののみ教える，興味のある人は「マルチコアプログラミング入門」を受講してください。

- 0900-0930　有限要素法プログラムの概要
- 0930-1015　並列有限要素法への道
- 1015-1045　ログイン・東大センターのスパコン
- 1045-1200　一般化された通信テーブル
- 1300-1430　一次元有限要素法の並列化
- 1430-1600　三次元並列有限要素法（1/2）
- 1600-1700　三次元並列有限要素法（2/2）
- 1710-1800　OpenMP・ハイブリッド並列

キーワード

- 熱伝導問題
- ガラーキン法
- 低次要素（六面体一次要素）
- 前処理付共役勾配法

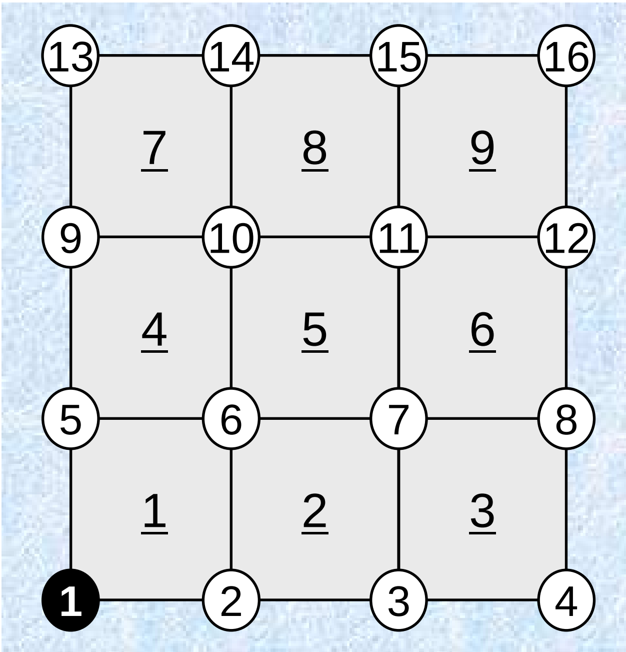
3分でわかる有限要素法

Finite-Element Method (FEM)

- 偏微分方程式の解法として広く知られている
  - elements (meshes，要素) & nodes (vertices，節点)
- 以下の二次元熱伝導問題を考える:

$$\lambda\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$$

  - 16節点，9要素（四角形）
  - 一様な熱伝導率 (λ=1)
  - 一様な体積発熱 (Q=1)
  - 節点1で温度固定：T=0
  - 周囲断熱



3分でわかる有限要素法

Galerkin FEM procedures

- 各要素にガラーキン法を適用:

$$\int_V [N]^T \left\{ \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q \right\} dV = 0$$

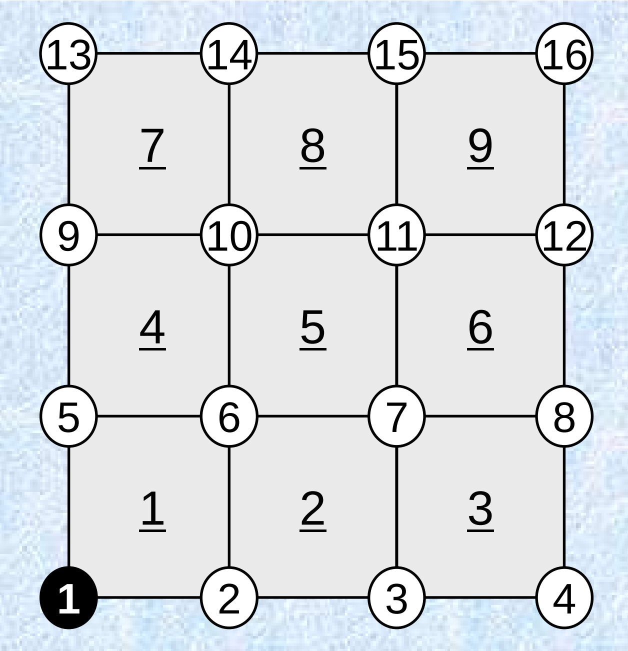
各要素で: $T = [N]\{\phi\}$

$[N]$：形状関数（内挿関数）

- 偏微分方程式に対して，ガウス・グリーンの定理を適用し，以下の「弱形式」を導く

$$-\int_V \lambda \left(\frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} \right) dV \cdot \{\phi\}$$

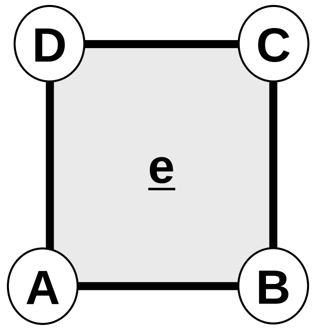
$$+\int_V Q[N]^T dV = 0$$



3分でわかる有限要素法

Element Matrix：要素マトリクス

- 各要素において積分を実行し，要素マトリクスを得る



$$
-\int_V \lambda \left(\frac{\partial [N]^T}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial [N]^T}{\partial y} \frac{\partial [N]}{\partial y} \right) dV \cdot \{\phi\}
$$

$$
+\int_V Q[N]^T dV = 0
$$

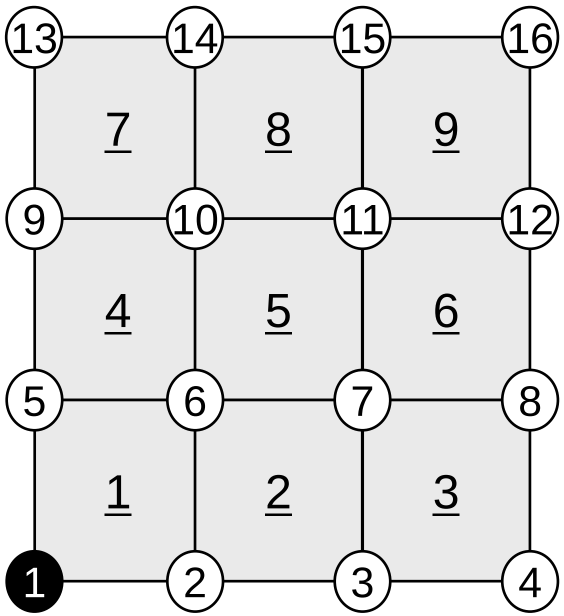
$$
[k^{(e)}]\{\phi^{(e)}\} = \{f^{(e)}\}
$$

$$
\begin{bmatrix} k_{AA}^{(e)} & k_{AB}^{(e)} & k_{AC}^{(e)} & k_{AD}^{(e)} \\ k_{BA}^{(e)} & k_{BB}^{(e)} & k_{BC}^{(e)} & k_{BD}^{(e)} \\ k_{CA}^{(e)} & k_{CB}^{(e)} & k_{CC}^{(e)} & k_{CD}^{(e)} \\ k_{DA}^{(e)} & k_{DB}^{(e)} & k_{DC}^{(e)} & k_{DD}^{(e)} \end{bmatrix} \begin{Bmatrix} \phi_A^{(e)} \\ \phi_B^{(e)} \\ \phi_C^{(e)} \\ \phi_D^{(e)} \end{Bmatrix} = \begin{Bmatrix} f_A^{(e)} \\ f_B^{(e)} \\ f_C^{(e)} \\ f_D^{(e)} \end{Bmatrix}
$$

3分でわかる有限要素法

Global/overall Matrix：全体マトリクス

各要素マトリクスを全体マトリクスに足しこむ



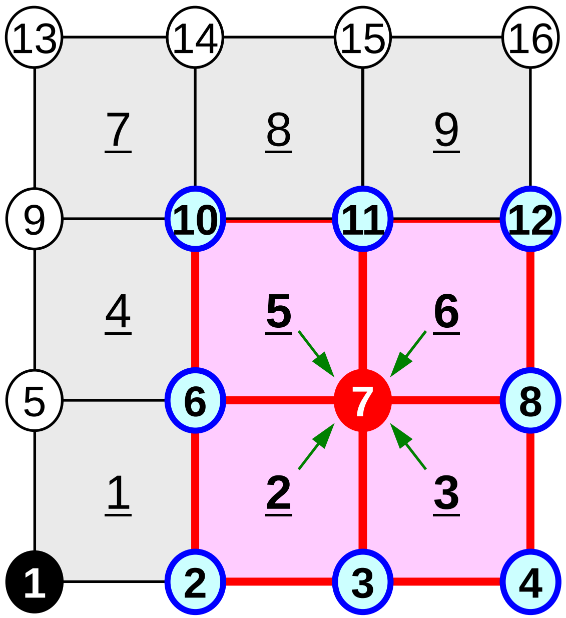
$$[K]\{\Phi\}=\{F\}$$

$$\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}$$

3分でわかる有限要素法

Global/overall Matrix：全体マトリクス

各要素マトリクスを全体マトリクスに足しこむ



$$[K]\{\Phi\}=\{F\}$$

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}
$$

3分でわかる有限要素法

得られた大規模連立一次方程式を解く

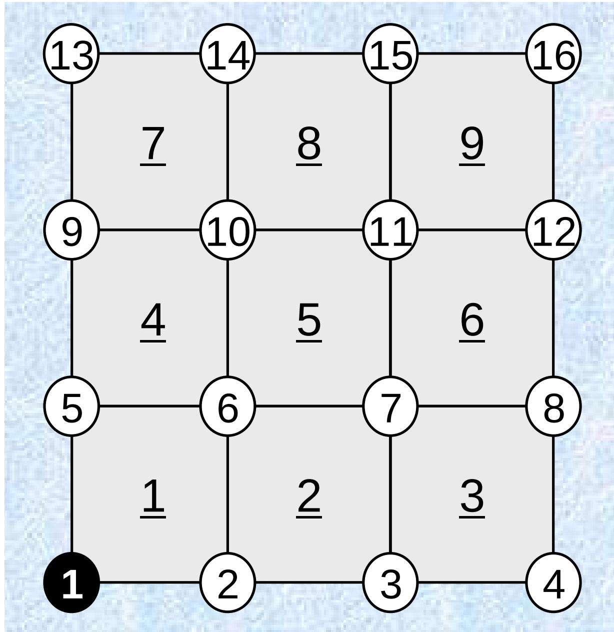
ある適切な境界条件 (ここではΦ_1=0)を適用

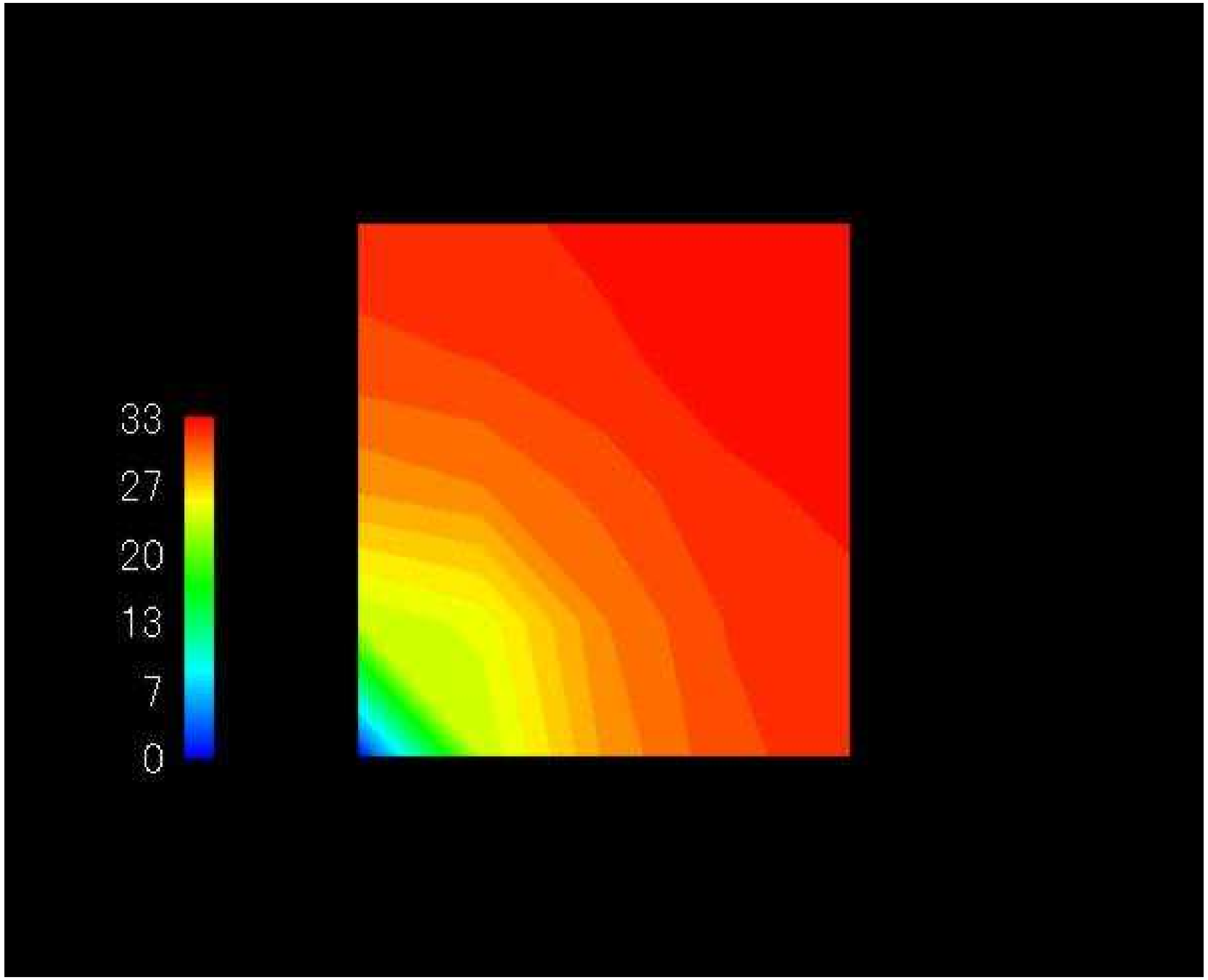
「疎（ゼロが多い）」な行列

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}
$$

計算結果

$$\lambda\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$$





代表的な反復法：共役勾配法

- Conjugate Gradient法，略して「CG」法
  - 最も代表的な「非定常」反復法
- 対称正定値行列（Symmetric Positive Definite：SPD）
  - 任意のベクトル$\{x\}$に対して$\{x\}^T[A]\{x\}>0$
  - 全対角成分>0，全固有値>0，全部分行列式>0と同値
  - （ガラーキン法）熱伝導，弾性，ねじり：本コードの場合もSPD

$$
\det\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & a_{24} & \cdots & a_{2n} \\
a_{31} & a_{32} & a_{33} & a_{34} & \cdots & a_{3n} \\
a_{41} & a_{42} & a_{43} & a_{44} & \cdots & a_{4n} \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
a_{n1} & a_{n2} & a_{n3} & a_{n4} & \cdots & a_{nn}
\end{bmatrix}
$$

- アルゴリズム
  - 最急降下法（Steepest Descent Method）
  - $\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha_i \mathbf{p}^{(i)}$
    - $x^{(i)}$：反復解，$p^{(i)}$：探索方向，α_i：定数）
  - 厳密解をyとするとき $\{x-y\}^T[A]\{x-y\}$を最小とするような$\{x\}$を求める。
  - 詳細は参考文献参照
    - 例えば：森正武「数値解析（第2版）」（共立出版）

共役勾配法のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減
  - DAXPY (Double Precision: a{X} + {Y})

$x^{(i)}$: ベクトル

α_i : スカラー

前処理（preconditioning）とは?

- 反復法の収束は係数行列の固有値分布に依存
  - 固有値分布が少なく，1に近いほど収束が早い（単位行列）
  - 条件数（condition number）（対称正定＝最大最小固有値比
    - 条件数が1に近いほど収束しやすい
- もとの係数行列**[A]**に良く似た前処理行列**[M]**を適用することによって固有値分布を改善する。
  - 前処理行列**[M]**によって元の方程式[A]{x}={b}を[A′]{x}={b′}へと変換する。ここで$[A']=[M]^{-1}[A]$，$\{b'\}=[M]^{-1}\{b\}$ である。
  - $[A']=[M]^{-1}[A]$が単位行列に近ければ良い
  - より一般的には[A′]{x′}={b′}（$[A']=[M_L]^{-1}[A][M_R]^{-1}$，$\{b'\}=[M_L]^{-1}\{b\}$，$\{x'\}=[M_R]\{x\}$
  - $[M_L]$／$[M_R]$：左／右前処理（left/right preconditioning）

前処理付き共役勾配法のアルゴリズム

Compute $r^{(0)} = b - [A]x^{(0)}$
for $i = 1, 2, \ldots$
  solve $[M]z^{(i-1)} = r^{(i-1)}$
  $\rho_{i-1} = r^{(i-1)}\ z^{(i-1)}$
  if $i=1$
    $p^{(1)} = z^{(0)}$
  else
    $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
    $p^{(i)} = z^{(i-1)} + \beta_{i-1}\ p^{(i-1)}$
  endif
  $q^{(i)} = [A]p^{(i)}$
  $\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)} = r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence $|r|$
end

$$[\mathbf{M}] = [\mathbf{M}_1][\mathbf{M}_2]$$

$$[\mathbf{A}']\mathbf{x}' = \mathbf{b}'$$
$$[\mathbf{A}'] = [\mathbf{M}_1]^{-1}[\mathbf{A}][\mathbf{M}_2]^{-1}$$
$$\mathbf{x}' = [\mathbf{M}_2]\mathbf{x},\quad \mathbf{b}' = [\mathbf{M}_1]^{-1}\mathbf{b}$$

$$\mathbf{p}' \Rightarrow [\mathbf{M}_2]\mathbf{p},\quad \mathbf{r}' \Rightarrow [\mathbf{M}_1]^{-1}\mathbf{r}$$

$$\mathbf{p}'^{(i)} = \mathbf{r}'^{(i-1)} + \beta'_{i-1}\ \mathbf{p}'^{(i-1)}$$

$$[\mathbf{M}_2]\mathbf{p}^{(i)} = [\mathbf{M}_1]^{-1}\mathbf{r}^{(i-1)} + \beta'_{i-1}\ [\mathbf{M}_2]\mathbf{p}^{(i-1)}$$

$$\mathbf{p}^{(i)} = [\mathbf{M}_2]^{-1}[\mathbf{M}_1]^{-1}\mathbf{r}^{(i-1)} + \beta'_{i-1}\ \mathbf{p}^{(i-1)}$$
$$\mathbf{p}^{(i)} = [\mathbf{M}]^{-1}\mathbf{r}^{(i-1)} + \beta'_{i-1}\ \mathbf{p}^{(i-1)}$$

$$\beta'_{i-1} = ([\mathbf{M}]^{-1}\mathbf{r}^{(i-1)}, \mathbf{r}^{(i-1)}) / ([\mathbf{M}]^{-1}\mathbf{r}^{(i-2)}, \mathbf{r}^{(i-2)})$$

$$\alpha'_{i-1} = ([\mathbf{M}]^{-1}\mathbf{r}^{(i-1)}, \mathbf{r}^{(i-1)}) / (\mathbf{p}^{(i-1)}, [\mathbf{A}]\mathbf{p}^{(i-1)})$$

ILU(0), IC(0)

- 最もよく使用されている前処理（疎行列用）
  - 不完全LU分解
    - Incomplete LU Factorization
  - 不完全コレスキー分解
    - Incomplete Cholesky Factorization（対称行列）

- 不完全な直接法
  - もとの行列が疎でも，逆行列は疎とは限らない。
  - fill-in
  - もとの行列と同じ非ゼロパターン（fill-in無し）を持っているのがILU（0），IC（0）

対角スケーリング，点ヤコビ前処理

- 前処理行列として，もとの行列の対角成分のみを取り出した行列を前処理行列 [M] とする。
  - 対角スケーリング，点ヤコビ（point-Jacobi）前処理

$$
[M] = \begin{bmatrix} D_1 & 0 & \dots & 0 & 0 \\ 0 & D_2 & & 0 & 0 \\ \dots & & \dots & & \dots \\ 0 & 0 & & D_{N-1} & 0 \\ 0 & 0 & \dots & 0 & D_N \end{bmatrix}
$$

- **solve [M]z$^{(i-1)}$= r$^{(i-1)}$**という場合に逆行列を簡単に求めることができる。
- 簡単な問題では収束する。

有限要素法で得られるマトリクス

- 疎行列
  - 0が多い
- $A(i,j)$のように正方行列の全成分を記憶することは疎行列では非効率的
  - 「密」行列向け

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}
$$

- 有限要素法：非零非対角成分の数は高々「数百」規模
  - 例えば未知数が10^8個あるとすると記憶容量（ワード数）は
    - 正方行列：$O(10^{16})$
    - 非零非対角成分数：$O(10^{10})$
- 非零成分のみ記憶するのが効率的

行列ベクトル積への適用

（非零）非対角成分のみを格納，疎行列向け方法 Compressed Row Storage (CRS)

Diag [i]	対角成分（実数，[N]）
Index[i]	非対角成分数に関する一次元配列（通し番号）（整数，[N+1]）
Item[k]	非対角成分の要素（列）番号（整数，[Index[N]]）
AMat[k]	非対角成分（実数，[Index[N]]）

```
{Y}=[A]{X}

for(i=0;i<N;i++){
        Y[i] = Diag[i] * X[i];
        for(k=Index[i];k<Index[i+1];k++){
                Y[i] += AMat[k]*X[Item[k]];
        }
}
```

$$
\begin{bmatrix}
D & X & & & X & X & & & & & & & & & & \\
X & D & X & & X & X & X & & & & & & & & & \\
& X & D & X & & X & X & X & & & & & & & & \\
& & X & D & & & X & X & & & & & & & & \\
X & X & & & D & X & & & X & X & & & & & & \\
X & X & X & & X & D & X & & X & X & X & & & & & \\
& X & X & X & & X & D & X & & X & X & X & & & & \\
& & X & X & & & X & D & & & X & X & & & & \\
& & & & X & X & & & D & X & & & X & X & & \\
& & & & X & X & X & & X & D & X & & X & X & X & \\
& & & & & X & X & X & & X & D & X & & X & X & X \\
& & & & & & X & X & & & X & D & & & X & X \\
& & & & & & & & X & X & & & D & X & & \\
& & & & & & & & X & X & X & & X & D & X & \\
& & & & & & & & & X & X & X & & X & D & X \\
& & & & & & & & & & X & X & & & X & D
\end{bmatrix}
\begin{Bmatrix}
\Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16}
\end{Bmatrix}
=
\begin{Bmatrix}
F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \\ F_{15} \\ F_{16}
\end{Bmatrix}
$$

行列ベクトル積：密行列⇒とても簡単

$$
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1,N-1} & a_{1,N} \\
a_{21} & a_{22} & & a_{2,N-1} & a_{2,N} \\
\cdots & & \cdots & & \cdots \\
a_{N-1,1} & a_{N-1,2} & & a_{N-1,N-1} & a_{N-1,N} \\
a_{N,1} & a_{N,2} & \cdots & a_{N,N-1} & a_{N,N}
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{Bmatrix}
=
\begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{Bmatrix}
$$

```
{Y}= [A]{X}

for (i=0;j<N;j++){
  Y[i]= 0.d0;
  for (j=0;j<N;j++){
    Y[i]= Y[i] + A[i][j]*X[j]
  }
}
```

Compressed Row Storage (CRS)

$$
\begin{array}{c}
\\
1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{bmatrix}
1.1 & 2.4 & 0 & 0 & 3.2 & 0 & 0 & 0 \\
4.3 & 3.6 & 0 & 2.5 & 0 & 3.7 & 0 & 9.1 \\
0 & 0 & 5.7 & 0 & 1.5 & 0 & 3.1 & 0 \\
0 & 4.1 & 0 & 9.8 & 2.5 & 2.7 & 0 & 0 \\
3.1 & 9.5 & 10.4 & 0 & 11.5 & 0 & 4.3 & 0 \\
0 & 0 & 6.5 & 0 & 0 & 12.4 & 9.5 & 0 \\
0 & 6.4 & 2.5 & 0 & 0 & 1.4 & 23.1 & 13.1 \\
0 & 9.5 & 1.3 & 9.6 & 0 & 3.1 & 0 & 51.3
\end{bmatrix}
\end{array}
$$

Compressed Row Storage (CRS)
Cでは0番から番号付け

	0	1	2	3	4	5	6	7
0	**1.1** ◎	2.4 ①			3.2 ④			
1	4.3 ◎	**3.6** ①		2.5 ③		3.7 ⑤		9.1 ⑦
2			**5.7** ②		1.5 ④		3.1 ⑥	
3		4.1 ①		**9.8** ③	2.5 ④	2.7 ⑤		
4	3.1 ◎	9.5 ①	10.4 ②		**11.5** ④		4.3 ⑥	
5			6.5 ②			**12.4** ⑤	9.5 ⑥	
6		6.4 ①	2.5 ②			1.4 ⑤	**23.1** ⑥	13.1 ⑦
7		9.5 ①	1.3 ②	9.6 ③		3.1 ⑤		**51.3** ⑦

N= 8

対角成分

```
Diag[0]=  1.1
Diag[1]=  3.6
Diag[2]=  5.7
Diag[3]=  9.8
Diag[4]= 11.5
Diag[5]= 12.4
Diag[6]= 23.1
Diag[7]= 51.3
```

Compressed Row Storage (CRS)

	0	1	2	3	4	5	6	7
0	1.1 ◎		2.4 ①			3.2 ④		
1	3.6 ①	4.3 ◎			2.5 ③		3.7 ⑤		9.1 ⑦
2	5.7 ②					1.5 ④		3.1 ⑥	
3	9.8 ③		4.1 ①			2.5 ④	2.7 ⑤		
4	11.5 ④	3.1 ◎	9.5 ①	10.4 ②				4.3 ⑥	
5	12.4 ⑤			6.5 ②				9.5 ⑥	
6	23.1 ⑥		6.4 ①	2.5 ②			1.4 ⑤		13.1 ⑦
7	51.3 ⑦		9.5 ①	1.3 ②	9.6 ③		3.1 ⑤		

Compressed Row Storage (CRS)

						非対角成分数	
							Index[0]= 0
0	1.1 ◎	2.4 ①	3.2 ④			2	Index[1]= 2
1	3.6 ①	4.3 ◎	2.5 ③	3.7 ⑤	9.1 ⑦	4	Index[2]= 6
2	5.7 ②	1.5 ④	3.1 ⑥			2	Index[3]= 8
3	9.8 ③	4.1 ①	2.5 ④	2.7 ⑤		3	Index[4]= 11
4	11.5 ④	3.1 ◎	9.5 ①	10.4 ②	4.3 ⑥	4	Index[5]= 15
5	12.4 ⑤	6.5 ②	9.5 ⑥			2	Index[6]= 17
6	23.1 ⑥	6.4 ①	2.5 ②	1.4 ⑤	13.1 ⑦	4	Index[7]= 21
7	51.3 ⑦	9.5 ①	1.3 ②	9.6 ③	3.1 ⑤	4	Index[8]= 25

NPLU= 25
(=Index[N])

Index[i]~Index[i+1]-1番目がi行目の非対角成分

Compressed Row Storage (CRS)

						非対角成分数	
							Index[0]= 0
0	1.1 ◎	2.4 ①,0	3.2 ④,1			2	Index[1]= 2
1	3.6 ①	4.3 ◎,2	2.5 ③,3	3.7 ⑤,4	9.1 ⑦,5	4	Index[2]= 6
2	5.7 ②	1.5 ④,6	3.1 ⑥,7			2	**Index[3]= 8**
3	9.8 ③	**4.1 ①,8**	**2.5 ④,9**	**2.7 ⑤,10**		**3**	**Index[4]= 11**
4	11.5 ④	3.1 ◎,11	9.5 ①,12	10.4 ②,13	4.3 ⑥,14	4	Index[5]= 15
5	12.4 ⑤	6.5 ②,15	9.5 ⑥,16			2	Index[6]= 17
6	23.1 ⑥	6.4 ①,17	2.5 ②,18	1.4 ⑤,19	13.1 ⑦,20	4	Index[7]= 21
7	51.3 ⑦	9.5 ①,21	1.3 ②,22	9.6 ③,23	3.1 ⑤,24	4	Index[8]= 25

NPLU= 25 (=Index[N])

Index[i]~Index[i+1]-1番目がi行目の非対角成分

Compressed Row Storage (CRS)

0	1.1 ◎	2.4 ①,0	3.2 ④,1		
1	3.6 ①	4.3 ◎,2	2.5 ③,3	3.7 ⑤,4	9.1 ⑦,5
2	5.7 ②	**1.5** **④,6**	3.1 ⑥,7		
3	9.8 ③	4.1 ①,8	2.5 ④,9	2.7 ⑤,10	
4	11.5 ④	3.1 ◎,11	9.5 ①,12	10.4 ②,13	4.3 ⑥,14
5	12.4 ⑤	6.5 ②,15	9.5 ⑥,16		
6	23.1 ⑥	6.4 ①,17	**2.5** **②,18**	1.4 ⑤,19	13.1 ⑦,20
7	51.3 ⑦	9.5 ①,21	1.3 ②,22	9.6 ③,23	3.1 ⑤,24

例:

```
Item[ 6]= 4, AMat[ 6]= 1.5
Item[18]= 2, AMat[18]= 2.5
```

Compressed Row Storage (CRS)

0	1.1 ◎	2.4 ①,0	3.2 ④,1		
1	3.6 ①	4.3 ◎,2	2.5 ③,3	3.7 ⑤,4	9.1 ⑦,5
2	5.7 ②	1.5 ④,6	3.1 ⑥,7		
3	9.8 ③	4.1 ①,8	2.5 ④,9	2.7 ⑤,10	
4	11.5 ④	3.1 ◎,11	9.5 ①,12	10.4 ②,13	4.3 ⑥,14
5	12.4 ⑤	6.5 ②,15	9.5 ⑥,16		
6	23.1 ⑥	6.4 ①,17	2.5 ②,18	1.4 ⑤,19	13.1 ⑦,20
7	51.3 ⑦	9.5 ①,21	1.3 ②,22	9.6 ③,23	3.1 ⑤,24

Diag [i]	対角成分（実数，[N]）
Index[i]	非対角成分数に関する一次元配列（通し番号）（整数，[N+1]）
Item[k]	非対角成分の要素（列）番号（整数，[Index[N]]）
AMat[k]	非対角成分（実数，[Index[N]]）

```
{Y}=[A]{X}

for(i=0;i<N;i++){
    Y[i] = Diag[i] * X[i];
    for(k=Index[i];k<Index[i+1];k++){
        Y[i] += AMat[k]*X[Item[k]];
      }
}
```

Compressed Row Storage (CRS)：C

0	1.1 ◎	2.4 ①,0	3.2 ④,1		
1	3.6 ①	4.3 ◎,2	2.5 ③,3	3.7 ⑤,4	9.1 ⑦,5
2	5.7 ②	1.5 ④,6	3.1 ⑥,7		
3	9.8 ③	4.1 ①,8	2.5 ④,9	2.7 ⑤,10	
4	11.5 ④	3.1 ◎,11	9.5 ①,12	10.4 ②,13	4.3 ⑥,14
5	12.4 ⑤	6.5 ②,15	9.5 ⑥,16		
6	23.1 ⑥	6.4 ①,17	2.5 ②,18	1.4 ⑤,19	13.1 ⑦,20
7	51.3 ⑦	9.5 ①,21	1.3 ②,22	9.6 ③,23	3.1 ⑤,24

Diag [i]	対角成分（実数，[N]）
Index[i]	非対角成分数に関する一次元配列（通し番号）（整数，[N+1]）
Item[k]	非対角成分の要素（列）番号（整数，[Index[N]]）
AMat[k]	非対角成分（実数，[Index[N]]）

```
{Y}=[A]{X}

for(i=0;i<N;i++){
    Y[i] = Diag[i] * X[i];
    for(k=Index[i];k<Index[i+1];k++){
        Y[i] += AMat[k]*X[Item[k]];
      }
}
```