Cybermedia Center Seminar: 21 Nov.2018

On solution of large linear system : iterative solver

Atsushi Suzuki[1]

[1] Cybermedia Center, Osaka University
atsushi.suzuki@cas.cmc.osaka-u.ac.jp

- 13:15 - 14:45 algorithm and theory of Krylov subspace methods
- 15:00 - 16:00 preconditioner and usage of GMRES in Intel MKL
- 16:00 - 17:00 hands-on tutorial

linear solvers

A : sparse matrix, nonsingular, $Ax = b$

- direct method LU-factorization
- Jacobi, Gauss-Seidel, SOR
- multigrid
- Krylov subspace method (CG, GMRES, BiCG, BiCGSTAB,...)

question : which solver of Krylov subspace method is useful to solve PDE problems?
answer : GMRES + additive Schwarz preconditioner (combination with domain decomposition + direct solver in subdomains)

- robustness with monotone convergence
- easily parallelizable preconditioner on cluster system of multicore CPUs

Rerefences

- Y. Saad, “Iterative Methods for Sparse Linear Systems, 2nd ed.”, 2003, SIAM
- F. Magoulès, F.-X. Roux, G. Houzeaux, “Parallel Scientific Computing”, 2015, Wiley
- A. Greenbaum, “Iterative Methods solving Sparse Linear Systems”, 1997, SIAM
- J. Málek, Z. Strakoš, “Preconditioningand the Conjugate Gradient Method in the Context of Solving PDEs”, 2015, SIAM

Hands-on tutorial to deal with iterative solver packages

GMRES, CG ∈ Intel Math Kernel Library (MKL) with
Reverse Communication Interface (RCI)

GMRES code from scratch

- `ILU(0)` ∈ Intel MKL
- additive Schwarz preconditioner using direct solver in submatrices
- `METIS` graph partitioner
- `Pardiso` sparse direct solver
- `SpMV` sparse matrix vector product, `mkl_dcsrgemv` ∈ Intel MKL

sparse matrces on structural mechanics/fluid dynamics generated by `FreeFem++`

matrix	N	nnz	characteristic
`Navier3D.32.P2`	750,141	31,214,610	symmetric positive definite
`Navier3DmeshP2`	677,163	28,853,844	symmetric postiive definite
`Stokes3Dsym`	187,218	8,691,446	symmetric indefinite
`Stokes3D`	187,218	17,195,674	unsymmetric coercive
`RayleighBenard3D`	536,567	63,903,043	general

- `Pardiso` works well for multicore environment
  other direct sovlers, `MUMPS`, `Dissection` also work for multicore
- `SpMV` is memory bounded operatoin, but short iteration by efficient preconditioner resolves this issue

Krylov subspace and solution of the linear system : 1/2

- $A \in \mathbb{R}^{N\times N}$, invertible, $\vec{b} \in \mathbb{R}^N$,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual.

Krylov subspace

$$K_n(A, \vec{r}_0) := \text{span}[\vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n-1}\vec{r}_0]$$

Lemma

$$A^n\vec{r}_0 \in K_n(A, \vec{r}_0) \ \Rightarrow \ A^{n+m}\vec{r}_0 \in K_n(A, \vec{r}_0) \quad \forall m > 0$$

proof by induction, suppose that $A^{n+m}\vec{r}_0 \in K_n(A, \vec{r}_0) \quad m \geq 0$

$$\begin{aligned} A^{n+m}\vec{r}_0 &= \textstyle\sum_{k=0}^{n-1}\alpha_k A^k\vec{r}_0 \\ A^{n+m+1}\vec{r}_0 &= \textstyle\sum_{k=0}^{n-2}\alpha_k A^{k+1}\vec{r}_0 + \alpha_{n-1}A^n\vec{r}_0 \\ &= \textstyle\sum_{k=0}^{n-2}\alpha_k A^{k+1}\vec{r}_0 + \alpha_{n-1}\sum_{k=0}^{n-1}\beta_k A^k\vec{r}_0 \in K_n(A, \vec{r}_0)\,. \end{aligned}$$

dimension of the largest Krylov subspace created by A and $\vec{r}_0$.

- $n_0 := \min_n\{K_n(A, \vec{r}_0) = K_{n+1}(A, \vec{r}_0)\}$

$K_1(A, \vec{r}_0) \subset K_2(A, \vec{r}_0) \subset \cdots \subset K_{n_0}(A, \vec{r}_0) = K_{n_0+1}(A, \vec{r}_0) = K_{n_0+2}(A, \vec{r}_0) = \cdots$

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

Krylov subspace and solution of the linear system : 2/2

Theorem

$\vec{x}$: solution of linear system $A\vec{x} = \vec{b} \Rightarrow \vec{x} \in \vec{x}_0 + K_{n_0}(A, \vec{r}_0)$.

proof

recalling that $\vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n_0-1}\vec{r}_0$: linearly independent.

$\alpha_0 \neq 0$ such that $A^{n_0}\vec{r}_0 = \sum_{k=0}^{n_0-1} \alpha_k A^k \vec{r}_0$.

$\alpha_0 = 0 \;\Rightarrow\; A^{n_0}\vec{r}_0 = \sum_{k=1}^{n_0-1} \alpha_k A^k \vec{r}_0,$

by applying A^{-1}

$$A^{n_0-1}\vec{r}_0 = \sum_{k=1}^{n_0-1} \alpha_k A^{k-1}\vec{r}_0,$$

$\Rightarrow \quad \vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n_0-1}\vec{r}_0$: linearly dependent $\Rightarrow\Leftarrow$

$$\alpha_0\vec{r}_0 + \sum_{k=1}^{n_0-1} \alpha_k A^k \vec{r}_0 - A^{n_0}\vec{r}_0 = \vec{0} \;\Leftrightarrow\; \vec{r}_0 + \sum_{k=1}^{n_0-1} \frac{\alpha_k}{\alpha_0} A^k \vec{r}_0 - \frac{1}{\alpha_0} A^{n_0}\vec{r}_0 = \vec{0}$$

$$\Leftrightarrow \left(\vec{b} - A\vec{x}_0\right) + \sum_{k=1}^{n_0} \gamma_k A^k \vec{r}_0 = \vec{0} \;\Leftrightarrow\; A\left(\vec{x}_0 - \sum_{k=1}^{n_0} \gamma_k A^k \vec{r}_0\right) = \vec{b}.$$

$\vec{x}_0 - \sum_{k=1}^{n_0} \gamma_k A^k \vec{r}_0 \in \vec{x}_0 + K_{n_0}(A, \vec{r}_0)$ + uniqueness of the solution : $A\vec{x} = \vec{b}$.

Krylov subspace and variational solution of the linear system

Theorem
problem (V) to find $\vec{x} \in \vec{x}_0 + K_{n_0}(A, \vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_{n_0}(A, \vec{r}_0)$
has a unique solution and is equivalent to the problem $A\vec{x} = \vec{b}$.
proof

- $\vec{x}_* \in \vec{x}_0 + K_{n_0}(A, \vec{r}_0)$: solution of (V).
- $\vec{x}_1$: solution of $(A\vec{x} - b, \vec{y}) = 0 \quad \forall \vec{y} \in \mathbb{R}^N \Rightarrow \vec{x}_1 \in \vec{x}_0 + K_{n_0}(A, \vec{r}_0)$.
  $(A\vec{x}_1 - b, \vec{y}) = 0 \quad \forall \vec{y} \in K_{n_0}(A, \vec{r}_0) \subset \mathbb{R}^N \Rightarrow \vec{x}_1$: solution of (V).

to verify uniqueness $(A(\vec{x}_0 - \vec{x}_*), \vec{y}) = 0 \quad \forall \vec{y} \in K_{n_0}(A, \vec{r}_0) \overset{?}{\Rightarrow} \vec{x}_0 - \vec{x}_* = \vec{0}$
$A : 1$ to 1 on $K_{n_0}(A, \vec{r}_0)$ is verifed as

$$\vec{z} \in K_{n_0}(A, \vec{r}_0) \text{ satisfying } (A\vec{z}, \vec{y}) = 0 \quad \forall \vec{y} \in K_{n_0}(A, \vec{r}_0)$$

$$\Rightarrow A\vec{z} \in (K_{n_0}(A, \vec{r}_0))^{\perp} \vee \vec{z} \in \ker A$$

$$\left.\begin{array}{rcl} \vec{z} \in K_{n_0}(A, \vec{r}_0) & \Rightarrow & A\vec{z} \in K_{n_0}(A, \vec{r}_0) \\ \exists A^{-1} & \Rightarrow & \ker A = \{\vec{0}\} \end{array}\right\} \Rightarrow \vec{z} = \vec{0}.$$

successive computation of variational problems
do $m = 1, 2, \cdots, n_0$
find $\vec{x} \in \vec{x}_0 + K_m(A, \vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(A, \vec{r}_0)$

- Conjugate Gradient (CG) method $\Leftarrow A$: symmetric positive definite
- Full Orthogonization Method (FOM) $\Leftarrow A$: coercive

A : coercive $\Leftrightarrow (A\vec{x}, \vec{x}) > 0 \; \forall \vec{x} \neq \vec{0} \Leftrightarrow (A + A^T)/2$: positive definite.

Arnoldi process

$\|\vec{v}_1\| = 1,$

$\{\vec{v}_1, A\vec{v}_1, A^2\vec{v}_1, \cdots, A^{m-1}\vec{v}_1\}$ $\rightarrow$ orthonormal basis $\{\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_m\}$
by Gram-Schmidt

Algorihm (Arnoldi process)

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

$h_{i,j} = (A\vec{v}_j, \vec{v}_i) \quad 1 \le i \le j$

$$\vec{w}_j = A\vec{v}_j - \sum_{i=1}^{j} h_{i,j}\vec{v}_i$$

$h_{j+1,j} = \|\vec{w}_j\|^2$

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

from the last line,

$$h_{j+1,j}\vec{v}_{j+1} = A\vec{v}_j - \sum_{i=1}^{j} h_{i,j}\vec{v}_i,$$

$$A\vec{v}_j = \sum_{i=1}^{j+1} h_{i,j}\vec{v}_i.$$

$$[A\vec{v}_1, \cdots, A\vec{v}_m] = [\vec{v}_1, \cdots, \vec{v}_m, \vec{v}_{m+1}] \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\ & & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ & & & & h_{m+1,m} \end{bmatrix}$$

$AV_m = V_{m+1}\overline{H}_m \qquad \overline{H}_m \in \mathbb{R}^{(m+1)\times m}$: Hessenberg matrix,

$V_m^T A V_m = H_m \quad \Leftarrow V_m^T V_m = I_m \Leftrightarrow (\vec{v}_j, \vec{v}_i) = \delta_{i\,j} \ 1 \le i, j \le m$

Full Orthogonizatien Method

- $A \in \mathbb{R}^{N\times N}$, invertible, $\vec{b} \in \mathbb{R}^N$,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual.
- $K_n(A, \vec{r}_0) :=$ span$[\vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n-1}\vec{r}_0]$: Krylov subspace

construction of basis of Krylov subspace by Arnoldi process
starting from $\vec{v}_1 = \vec{r}_0/\beta, \beta = \|\vec{r}_0\|$,

$$
\begin{aligned}
AV_m &= V_{m+1}\overline{H}_m \\
V_m^T AV_m &= H_m \\
V_m^T \vec{r}_0 &= V_m^T \beta \vec{v}_1 = \beta \vec{\epsilon}_m^{(1)}, \quad [\vec{\epsilon}_m^{(1)}]_i = \delta_{i\,1}\ 1 \leq i \leq m
\end{aligned}
$$

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

$$
\begin{aligned}
\vec{x}_m &= \vec{x}_0 + V_m \vec{\eta}_m \\
A\vec{x}_m - \vec{b} &= A\vec{x}_0 - \vec{b} + AV_m\vec{\eta}_m \\
&= -\vec{r}_0 + AV_m\vec{\eta}_m \\
V_m^T(A\vec{x}_m - \vec{b}) &= -V_m^T\vec{r}_0 + V_m^T AV_m\vec{\eta}_m \\
&= -\beta\vec{\epsilon}_m^{(1)} + H_m\vec{\eta}_m, \\
\vec{\eta}_m &= H_m^{-1}(\beta\vec{\epsilon}_m^{(1)})
\end{aligned}
$$

H_m : invertible? A is coercive $\Rightarrow$ yes

Generalized Minimal Residual (GMRES) Method : 1/3

- $A \in \mathbb{R}^{N\times N}$, invertible, $\vec{b} \in \mathbb{R}^N$,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual.
- $K_n(A, \vec{r}_0) :=$ span$[\vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n-1}\vec{r}_0]$: Krylov subspace

construction of basis of Krylov subspace by Arnoldi process
starting from $\vec{v}_1 = \vec{r}_0/\beta, \beta = \|\vec{r}_0\|$,

$$AV_m = V_{m+1}\overline{H}_m,$$

$$\vec{r}_0 = \beta\vec{v}_1 = \beta V_{m+1}\vec{\epsilon}^{\,(1)}_{m+1}, \quad [\vec{\epsilon}^{\,(1)}_{m+1}]_i = \delta_{i\,1}\ 1 \leq i \leq m+1$$

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

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

$$\vec{y} = \vec{x}_0 + V_m\vec{\eta}_m$$

$$\begin{aligned}\vec{b} - A\vec{y} &= \vec{b} - A\vec{x}_0 - AV_m\vec{\eta}_m \\ &= \vec{r}_0 - AV_m\vec{\eta}_m \\ &= V_{m+1}\left(\beta\vec{\epsilon}^{\,(1)}_{m+1} - \overline{H}_m\vec{\eta}_m\right)\end{aligned}$$

$$\|\vec{b} - A\vec{y}\| = \|\beta\vec{\epsilon}^{\,(1)}_{m+1} - \overline{H}_m\vec{\eta}_m\| \qquad \Leftarrow V^T_{m+1}V_{m+1} = I_{m+1}.$$

find $\vec{x}_m = \vec{x}_0 + V_m\vec{\eta}_m,\ \vec{\eta}_m = \underset{\vec{\eta}}{\mathrm{argmin}}\|\beta\vec{\epsilon}^{\,(1)}_{m+1} - \overline{H}_m\vec{\eta}_m\|$ works for any $\overline{H}_m$

Generalized Minimal Residual (GMRES) Method : 2/3

a way to solve minimization problem by QR-factorization with Givens rotation

Givens rotation matrices $\Omega_i \in \mathbb{R}^{(m+1)\times(m+1)}$

$$\Omega_1 := \begin{bmatrix} c_1 & s_1 & \\ -s_1 & c_1 & \\ & & I_{m-1} \end{bmatrix}, \quad c_1 := \frac{h_{1\,1}}{\sqrt{h_{1\,1}^2 + h_{2\,1}^2}}, \; s_1 := \frac{h_{2\,1}}{\sqrt{h_{1\,1}^2 + h_{2\,1}^2}}.$$

$$\Omega_1 \overline{H}_m = \begin{bmatrix} h_{1,1}^{(1)} & h_{1,2}^{(1)} & \cdots & h_{1,m-1}^{(1)} & h_{1,m}^{(1)} \\ 0 & h_{2,2}^{(1)} & \cdots & h_{2,m-1}^{(1)} & h_{2,m}^{(1)} \\ & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\ & & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ & & & & h_{m+1,m} \end{bmatrix}, \; \beta\Omega_1 \vec{\epsilon}_{m+1}^{\,(1)} = \beta\Omega_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \beta \begin{bmatrix} c_1 \\ -s_1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$Q_m := \Omega_m \Omega_{m-1} \cdots \Omega_1 \in \mathbb{R}^{(m+1)\times(m+1)},$$

$$Q_{m-1}\overline{H}_m = \begin{bmatrix} h_{1,1}^{(1)} & h_{1,2}^{(1)} & h_{1,3}^{(1)} & \cdots & h_{1,m-1}^{(1)} & h_{1,m}^{(1)} \\ 0 & h_{2,2}^{(2)} & h_{2,3}^{(2)} & \cdots & h_{2,m-1}^{(2)} & h_{2,m}^{(2)} \\ & 0 & h_{3,3}^{(3)} & \cdots & h_{3,m-1}^{(3)} & h_{3,m}^{(3)} \\ & & & \ddots & \vdots & \vdots \\ & & & \ddots & h_{m-1,m-1}^{(m-2)} & h_{m,m-1}^{(m-2)} \\ & & & & 0 & h_{m,m}^{(m-1)} \\ & & & & 0 & h_{m+1,m} \end{bmatrix} \quad \beta \begin{bmatrix} c_1 \\ -c_2 s_1 \\ c_3 s_2 s_1 \\ \vdots \\ \gamma_{m-2} \\ \gamma_{m-1} \\ -s_{m-1}\gamma_{m-1} \\ 0 \end{bmatrix}$$

Generalized Minimal Residual (GMRES) Method : 3/3

$$Q_m \overline{H}_m = \begin{bmatrix} h^{(1)}_{1,1} & h^{(1)}_{1,2} & h^{(1)}_{1,3} & \cdots & h^{(1)}_{1,m-1} & h^{(1)}_{1,m} \\ 0 & h^{(2)}_{2,2} & h^{(2)}_{2,3} & \cdots & h^{(2)}_{2,m-1} & h^{(2)}_{2,m} \\ & 0 & h^{(3)}_{3,3} & \cdots & h^{(3)}_{3,m-1} & h^{(3)}_{3,m} \\ & & 0 & \ddots & \vdots & \vdots \\ & & & \ddots & h^{(m-1)}_{m-1,m-1} & h^{(m-1)}_{m,m-1} \\ & & & & 0 & h^{(m)}_{m,m} \\ & & & & 0 & 0 \end{bmatrix}, \quad \beta Q_m \vec{\epsilon}^{\,(1)}_{m+1} = \beta \begin{bmatrix} c_1 \\ -c_2 s_1 \\ c_3 s_2 s_1 \\ \vdots \\ \gamma_{m-1} \\ \gamma_m \\ -s_m \gamma_m \end{bmatrix}$$

$\overline{R}_m := Q_m \overline{H}_m$: upper triangular,

$\vec{\gamma}_{m+1} := \beta Q_m \vec{\epsilon}^{\,(1)}_{m+1} = [\, \gamma_1,\ \gamma_2,\ \cdots,\ \gamma_{m+1} \,]^T = [\vec{\gamma}_m^T,\ \gamma_{m+1}]^T$,

$\min \|\beta \vec{\epsilon}^{\,(1)}_{m+1} - \overline{H}_m \vec{\eta}\| = \min \|Q_m(\vec{\gamma}_{m+1} - \overline{R}_m \vec{\eta})\| = |\gamma_{m+1}| = |s_1 s_2 \cdots s_m| \beta\,.$

$\vec{\eta}_m = R_m^{-1} \vec{\gamma}_m$ attains the minimum.

- $\exists R_m^{-1}$ $(1 \le m \le n_0)$ for invertible matrix $A \Leftarrow h_{j+1,j} > 0$ $(1 \le j < n_0)$
- residual $\|\vec{r}_m\| = \|\vec{b} - A\vec{x}_m\|$ decreases monotonically thanks to s_m.
- $h^{(m-1)}_{m,m} = 0 \;\Rightarrow\; Q_{m-1} H_m$: singular, FOM fails
  $\Rightarrow c_m = 0, s_m = 1$, GMERS stagnates at m-th step.
- $\vec{r}_m^{\,\text{GMRES}} = s_m^2 \vec{r}_{m-1}^{\,\text{GMRES}} + c_m^2 \vec{r}_m^{\,\text{FOM}}$ $\quad s_{n_0} = 0,\ c_{n_0} = 1 \Leftarrow h_{n_0+1,n_0} = 0.$

conjugate gradient method : 1/3

- $A \in \mathbb{R}^{N\times N}$, symmetric positive definite, $\vec{b} \in \mathbb{R}^N$,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual.
- $K_n(A, \vec{r}_0) :=$ span$[\vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n-1}\vec{r}_0]$: Krylov subspace

Algorithm(CG)
$\vec{p}_0 = \vec{r}_0$.
do $m = 0, 1, \ldots$
  $\alpha_n = \|\vec{r}_m\|^2/(A\vec{p}_m, \vec{p}_m)$,
  $\vec{x}_{m+1} = \vec{x}_m + \alpha_m\vec{p}_m$,
  $\vec{r}_{m+1} = \vec{r}_m - \alpha_m A\vec{p}_m$,
  if $\|\vec{r}_{m+1}\| < \epsilon$ exit loop.
  $\beta_m = \|\vec{r}_{m+1}\|^2/\|\vec{r}_m\|^2$,
  $\vec{p}_{m+1} = \vec{r}_{m+1} + \beta_m\vec{p}_m$.

Lemma for $1 \le m \le n_0$

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

successive computation of variational problems
do $m = 1, 2, \cdots, n_0$
  find $\vec{x} \in \vec{x}_0 + K_m(A, \vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(A, \vec{r}_0)$

conjugate gradient method : 2/3

proof of Lemma by induction

for $m = 1$

(1) $$(\vec{r}_1, \vec{r}_0) = (\vec{r}_0 - \alpha_0 A\vec{p}_0, \vec{r}_0) = (\vec{r}_0, \vec{r}_0) - \frac{\|\vec{r}_0\|}{(A\vec{p}_0, \vec{p}_0)}(A\vec{p}_0, \vec{p}_0) = 0$$

(2) $$(A\vec{p}_1, \vec{r}_0) = (\vec{r}_1 + \beta_0\vec{p}_0, A\vec{p}_0) \quad \text{by symmetry of } A$$
$$= (\vec{r}_1 + \beta_0\vec{p}_0, \frac{1}{\alpha_0}(\vec{r}_0 - \vec{r}_1)) = -\frac{1}{\alpha_0}(\vec{r}_1, \vec{r}_1) + \frac{\beta_0}{\alpha_0}(\vec{p}_0, \vec{r}_0) = 0$$

(3) $$\text{span}[\vec{r}_0, \vec{r}_1] = \text{span}[\vec{p}_0, \vec{p}_1] = K_2(A, \vec{r}_0) \ \Leftarrow \alpha_0 \neq 0$$

for $m = k$, $\vec{z} \in K_{k+1}(A, \vec{r}_0)$: decomposed as $\vec{z} = \vec{z}_0 + \gamma_k\vec{p}_k$, $\vec{z}_0 \in K_k(A, \vec{r}_0)$

(1) $$(\vec{r}_{k+1}, \vec{z}_0) = (\vec{r}_k - \alpha_0 A\vec{p}_k, \vec{z}_0) = (\vec{r}_k, \vec{z}_0) - \alpha_k(A\vec{p}_k, \vec{z}_0) = 0$$
$$(\vec{r}_{k+1}, \vec{p}_k) = (\vec{r}_k, \vec{p}_k) - \alpha_k(A\vec{p}_k, \vec{p}_k)$$
$$= (\vec{r}_k, \vec{r}_k + \beta_{k-1}\vec{p}_{k-1}) - \|r_k\|^2 = \beta_{k-1}(\vec{r}_k, \vec{p}_{k-1}) = 0$$

(2) $$(A\vec{p}_{k+1}, \vec{z}_0) = (A(\vec{r}_{k+1} + \beta_k\vec{p}_k), \vec{z}_0) = (\vec{r}_{k+1}, A\vec{z}_0) + \beta_k(A\vec{p}_k, \vec{z}_0) = 0$$
$$(A\vec{p}_{k+1}, \vec{p}_k) = (\vec{r}_{k+1}, A\vec{p}_k) + \beta_k(A\vec{p}_k, \vec{p}_k)$$
$$= (\vec{r}_{k+1}, \frac{1}{\alpha_k}(\vec{r}_k - \vec{r}_{k+1})) + \beta_k(A\vec{p}_k, \vec{p}_k)$$
$$= -\frac{1}{\alpha_{k+1}}\|\vec{r}_{k+1}\|^2 + \|\vec{r}_{k+1}\|^2\frac{(A\vec{p}_k, \vec{p}_k)}{\|\vec{r}_k\|^2} = 0$$

(3) $$\text{span}[\vec{r}_0, \cdots, \vec{r}_k, \vec{r}_{k+1}] = \text{span}[\vec{r}_0, \cdots, \vec{r}_k, \vec{r}_k - \alpha_k A\vec{p}_k] = K_{k+2}(A, \vec{r}_0)$$

conjugate gradient method : 3/3

relation with Lanczos process

$A = A^T$: symmetric

$$[A\vec{v}_1, \cdots, A\vec{v}_m] = [\vec{v}_1, \cdots, \vec{v}_m, \vec{v}_{m+1}] \begin{bmatrix} h_{1,1} & h_{1,2} & & & \\ h_{2,1} & h_{2,2} & \ddots & & \\ & h_{3,2} & \ddots & & \\ & & \ddots & \ddots & h_{m,m-1} \\ & & & h_{m,m-1} & h_{m,m} \\ & & & & h_{m+1,m} \end{bmatrix}$$

$AV_m = V_{m+1}\overline{T}_m$ $\quad \overline{T}_m \in \mathbb{R}^{(m+1)\times m}$: tri-diagonal matrix, T_m : symmetric

$V_m^T A V_m = T_m \quad \Leftarrow V_m^T V_m = I_m \Leftrightarrow (\vec{v}_j, \vec{v}_i) = 0 \; 1 \leq i, j \leq m$

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

$$\vec{x}_m = \vec{x}_0 + V_m \vec{\eta}_m$$

$$0 = V_m^T(A\vec{x}_m - \vec{b}) = -V_m^T \vec{r}_0 + V_m^T A V_m \vec{\eta}_m$$

$$= -\beta \vec{\epsilon}_m^{(1)} + T_m \vec{\eta}_m,$$

- A : symmetric positive definite $\Rightarrow T_m$ can be factorized without permutation
- Conjugate Gradient method computes $\vec{x}_m$ without explicit tridiagonal factorization

bi-conjugate gradient method : 1/3

- $A \in \mathbb{R}^{N\times N}$: invertible, $\vec{b} \in \mathbb{R}^N$,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual, $\vec{r}_0^{\,*}$: shadow residual
- $K_n(A, \vec{r}_0) :=$ span$[\vec{r}_0, A\vec{r}_0, A^2\vec{r}_0, \cdots, A^{n-1}\vec{r}_0]$, $\quad K_n(A^T, \vec{r}_0^{\,*})$

Algorithm(Bi-CG)

$\vec{p}_0 = \vec{r}_0, \quad \vec{p}_0 = \vec{r}_0.$
do $m = 0, 1, \ldots$
$\alpha_n = (\vec{r}_m, \vec{r}_m^{\,*})/(A\vec{p}_m, \vec{p}_m^{\,*}),$
$\vec{x}_{m+1} = \vec{x}_m + \alpha_m \vec{p}_m,$
$\vec{r}_{m+1} = \vec{r}_m - \alpha_m A\vec{p}_m, \quad \vec{r}_{m+1}^{\,*} = \vec{r}_m^{\,*} - \alpha_m A^T \vec{p}_m^{\,*},$
if $\|\vec{r}_{m+1}\| < \epsilon$ exit loop.
$\beta_m = (\vec{r}_{m+1}, \vec{r}_{m+1}^{\,*})/(\vec{r}_m, \vec{r}_m^{\,*}),$
$\vec{p}_{m+1} = \vec{r}_{m+1} + \beta_m \vec{p}_m, \quad \vec{p}_{m+1}^{\,*} = \vec{r}_{m+1}^{\,*} + \beta_m \vec{p}_m^{\,*},$

Lemma if without breakdown for $1 \le m \le n_0$

- $(\vec{r}_m, \vec{z}) = 0 \quad \forall \vec{z} \in K_m(A^T, \vec{r}_0^{\,*})$
- $(A\vec{p}_m, \vec{z}) = 0 \quad \forall \vec{z} \in K_m(A^T, \vec{r}_0^{\,*})$
- span$[\vec{r}_0, \vec{r}_1, \cdots, \vec{r}_m]$ = span$[\vec{p}_0, \vec{p}_1, \cdots, \vec{p}_m] = K_{m+1}(A, \vec{r}_0)$
- span$[\vec{r}_0^{\,*}, \vec{r}_1^{\,*}, \cdots, \vec{r}_m^{\,*}]$ = span$[\vec{p}_0^{\,*}, \vec{p}_1^{\,*}, \cdots, \vec{p}_m^{\,*}] = K_{m+1}(A^T, \vec{r}_0^{\,*})$

successive computation of variational problems

do $m = 1, 2, \cdots, n_0$
find $\vec{x} \in \vec{x}_0 + K_m(A, \vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(A^T, \vec{r}_0^{\,*})$

bi-conjugate gradient method : 2/3

Lanczos biorthogonalization process

$$[A\vec{v}_1, \cdots, A\vec{v}_m] = [\vec{v}_1, \cdots, \vec{v}_m, \vec{v}_{m+1}] \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \ddots & & \\ & \delta_3 & \ddots & & \\ & & \ddots & \ddots & \beta_m \\ & & & \delta_m & \alpha_m \\ & & & & \delta_{m+1} \end{bmatrix}$$

$$\begin{aligned} AV_m &= V_m T_m + \delta_{m+1} \vec{v}_{m+1} \vec{\epsilon}_m^{(m)\ T} \\ A^T W_m &= W_m T_m^T + \beta_{m+1} \vec{w}_{m+1} \vec{\epsilon}_m^{(m)\ T} \\ W_m^T A\, V_m &= T_m \qquad \Leftarrow W_m^T V_m = I_m : \text{bi-orthogonality} \end{aligned}$$

two-sided Lanczos algorithm
variational problem with Petrov-Galerkin type

find $\vec{x} \in \vec{x}_0 + K_m(A, \vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(A^T, \vec{r}_0{}^*)$

find $\vec{x}_m = \vec{x}_0 + V_m \vec{\eta}_m,$ by solving $T_m \vec{\eta}_m = \beta \vec{\epsilon}_m^{(1)}$

Two possibilities of break down

- $(A\vec{p}_m, \vec{p}_m^{\,*}) = 0 \Rightarrow T_m$ becomes singular
- $(\vec{r}_m, \vec{r}_m^{\,*}) = 0 \quad \Rightarrow$ breakdown of Lanczos biorthogonalization process

bi-conjugate gradient method : 3/3

Composite step biconjugate gradient method — Bank-Chan 1993
stable factorization of T_m with 2×2 block pivots

Quasi-Minimal Residual (QMR) method — Freund-Nachtigal 1991
V_m generated by look-ahead Lanczos process — Parlett-Taylor-Liu 1985

$$\vec{x}_m = \vec{x}_0 + V_m \vec{\eta}_m$$

$$\begin{aligned} \vec{b} - A\vec{x}_m &= \vec{r}_0 - AV_m\vec{\eta}_m \\ &= V_{m+1}(\beta\vec{\epsilon}^{\,(1)}_{m+1} - \overline{T}_m\vec{\eta}_m). \end{aligned}$$

$V_{m+1}^T V_{m+1} \neq I_{m+1}$ in general.
find $\vec{\eta}_m \in \mathbb{R}^m \quad \|\beta\vec{\epsilon}^{\,(1)}_{m+1} - \overline{T}_m\vec{\eta}_m\| \leq \|\beta\vec{\epsilon}^{\,(1)}_{m+1} - \overline{T}_m\vec{\eta}\| \quad \forall\vec{\eta} \in \mathbb{R}^m$

to avoid transposed matrix-vector operation
Conjugate Gradient Squared (CGS) method — Sonnenveld 1989
in BiCG with polynomial of degree m, $\vec{r}_m = \phi_m(A)\vec{r}_0$, $\vec{r}_m^{\,*} = \phi_m(A^T)\vec{r}_0^{\,*}$,

$$\alpha_m = \frac{(\phi_m(A)\vec{r}_0, \phi_m(A^T)\vec{r}_0^{\,*})}{(A\vec{p}_m, \vec{p}_m^{\,*})} = \frac{(\phi_m(A)^2\vec{r}_0, \vec{r}_0^{\,*})}{(A\vec{p}_m, \vec{p}_m^{\,*})}$$

new residual $\vec{r}_m^{\,\prime} = \phi_m(A)^2\vec{r}_0$ is computed without multiplication of A^T.
to stabilize / smooth convergence
Bi-Conjugate Gradient Stabilized (BiCGSTAB) — van der Vorst 1992
residual $\vec{r}_m^{\,\prime} = \psi_m(A)\phi_m(A)\vec{r}_0$ with smoothing polynomial of degree m,
$\psi_m(t) = (1 - \omega_m t)\psi_{m-1}(t)$: polynomial with variable t.

preconditioned conjugate gradient method

- $A, Q \in \mathbb{R}^{N\times N}$, symmetric positive definite, $Q \sim A^{-1}$ $\quad \vec{b} \in \mathbb{R}^N$,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual.
- $K_n(QA, Q\vec{r}_0) :=$ span$[Q\vec{r}_0, QAQ\vec{r}_0, (QA)^2Q\vec{r}_0, \cdots, (QA)^{n-1}Q\vec{r}_0]$

Algorithm(preconditioned CG)
$\vec{p}_0 = Q\vec{r}_0$.
do $m = 0, 1, \ldots$
  $\alpha_n = (Q\vec{r}_m, \vec{r}_m)/(A\vec{p}_m, \vec{p}_m)$,
  $\vec{x}_{m+1} = \vec{x}_m + \alpha_m\vec{p}_m$,
  $\vec{r}_{m+1} = \vec{r}_m - \alpha_m A\vec{p}_m$,
  if $\|\vec{r}_{m+1}\| < \epsilon$ exit loop.
  $\beta_m = (Q\vec{r}_{m+1}, \vec{r}_{m+1})/(Q\vec{r}_m, \vec{r}_m)$,
  $\vec{p}_{m+1} = Q\vec{r}_{m+1} + \beta_m\vec{p}_m$.

Lemma for $1 \leq m \leq n_0$

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

successive computation of variational problems
do $m = 1, 2, \cdots, n_0$
  find $\vec{x} \in \vec{x}_0 + K_m(QA, Q\vec{r}_0) \quad (A\vec{x} - \vec{b}, \vec{y}) = 0 \quad \forall \vec{y} \in K_m(QA, Q\vec{r}_0)$

preconditioned Kyrlov subspace method : 1/2

$Q \in \mathbb{R}^N$: preconditioner, $Q^{-1} \sim A$.

- left preconditioner $(Q\,A)\vec{x} = Q\vec{b}$
- right preconditioner $(A\,Q)\vec{z} = \vec{b}\,,\quad \vec{x} = Q\vec{z}$

preconditioned conjugate gradient method can be seen as following variational problem with $A = A^T$.

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

assumption : A : 1 to 1 on $K_{n_0}(QA, Q\vec{r}_0)$

Theorem

Variational problem $(V_Q^{(n_0)})$ in $K_{n_0}(QA, Q\vec{r}_0)$ has a unique solution and is equivalent to the problem $A\vec{x} = \vec{b}$.

- A, Q : symmetric positive definite $\Rightarrow$ assumption for CG is OK
- A, Q : coercive $\Rightarrow$ assumption for FOM is OK

left preconditioned GMRES

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

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

$Q^T Q$: symmetric positive definite $\Rightarrow \exists q_1 > 0, \exists q_2 > 0$

$$q_1\|Q\vec{x}\| \leq \|\vec{x}\| \leq q_2\|Q\vec{x}\| \quad \forall \vec{x} \in \mathbb{R}^N$$

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

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

flexible GMRESS

Flexible GMRES as an extention of right preconditioned GMRES

- $A \in \mathbb{R}^{N \times N}$: invertible $\vec{b} \in \mathbb{R}^N$,
- Q_m : right preconditioner at m-th step,
- $\vec{x}_0$: initial guess,
- $\vec{r}_0 := \vec{b} - A\vec{x}_0$: initial residual, $\beta = \|\vec{r}_0\|$, $\vec{v}_1 = \vec{r}_0/\beta$.

Arnoldi process with modified Gram-Schmidt is used

Algorithm(flexible GMRES)

do $j = 1, 2, \ldots, m$
  $\vec{z}_j = Q_j \vec{v}_j$
  $\vec{w} = A\vec{z}_j$
  do $i = 1, \cdots, j$
    $h_{i,j} := (\vec{w}, \vec{v}_j)$
    $\vec{w} := \vec{w} - h_{i,j}\vec{v}_i$
  $h_{j+1,j} := \|\vec{w}\|$
  $\vec{v}_{j+1} = \vec{w}/h_{j+1,j}$
$Z_m := [\vec{z}_1, \cdots, \vec{z}_m]$
$\vec{\eta}_m = \underset{\vec{\eta}}{\mathrm{argmin}} \|\beta\vec{\epsilon}^{(1)}_{(m+1)} - \overline{H}_m\vec{\eta}\|$,
$\vec{x}_m = \vec{x}_0 + Z_m\vec{\eta}_m$.

right preconditioned GMRES

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

flexible GMRES

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

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

$V^T_{m+1}V_{m+1} = I_{m+1}$ then

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

span$[Q_1\vec{v}_1, Q_2\vec{v}_2, \cdots, Q_m\vec{v}_m]$ is no longer a Krylov subspace except the case $Q_j = Q$ for $1 \leq j \leq m$

convergence analysis of CG

- A : symmetric positive definte, $\exists \alpha > 0$ $(A\vec{x}, \vec{x}) \geq \alpha \|\vec{x}\|^2$ $\forall \vec{x} \in \mathbb{R}^N$.
- $A = V\Lambda V^T$, Λ : eigenvalues, V : eigenvectors $V^T V = I_N$
- $\vec{x}_*$: solution of $A\vec{x} = \vec{b}$, $\vec{x}_m$: approximate solution by CG
- $\mathbb{P}_m$: polynomial of degree m.

$$\begin{aligned}
\vec{y}_m - \vec{x}_* &= \vec{x}_0 + q_{m-1}(A)\vec{r}_0 - \vec{x}_* \qquad q_{m-1} \in \mathbb{P}_{m-1} \\
&= \vec{x}_0 + q_{m-1}(A)(\vec{b} - A\vec{x}_0) - \vec{x}_* = (\vec{x}_0 - \vec{x}_*) + q_{m-1}(A)A(\vec{x}_* - \vec{x}_0) \\
&= (I - q_{m-1}(A)A)(\vec{x}_0 - \vec{x}_*) = r_m(A)(\vec{x}_0 - \vec{x}_*) \quad r_m \in \mathbb{P}_m, r(0) = 1.
\end{aligned}$$

Galerkin orthogonality $(\vec{b} - A\vec{x}_m, \vec{x}_m - \vec{y}_m) = 0 \quad \forall \vec{y}_m \in \vec{x}_0 + K_m(A, \vec{r}_0)$

$$\alpha \|\vec{x}_m - \vec{x}_*\|^2 \leq (A(\vec{x}_* - \vec{x}_m), \vec{x}_* - \vec{x}_m) \leq \|A\| \|\vec{x}_m - \vec{x}_*\| \|\vec{x}_* - \vec{y}_m\|$$

$$\|\vec{y}_m - \vec{x}_*\| = \|r_m(A)(\vec{x}_0 - \vec{x}_*)\| = \|V r_m(\Lambda) V^T (\vec{x}_0 - \vec{x}_*)\| \leq \|r_m(\Lambda)\| \|\vec{x}_0 - \vec{x}_*\|$$

$$\begin{aligned}
\min_{r_m \in \mathbb{P}_m, r_m(0)=1} \|r_m(A)(\vec{x}_0 - \vec{x}_*)\| &\leq \min_{r_m \in \mathbb{P}_m, r_m(0)=1} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |r_m(\lambda)| \|\vec{x}_0 - \vec{x}_*\| \\
&\leq C_m\left(\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right)^{-1} \|\vec{x}_0 - \vec{x}_*\|
\end{aligned}$$

$C_m(k) = \cosh(k \cosh^{-1}(t))$ $|t| \geq 1$: Chebyshev polynomial of the first kind
$\kappa = \lambda_{\max}/\lambda_{\min}$: condition number

$$\|\vec{x}_m - \vec{x}_*\| \leq 2\frac{\|A\|}{\alpha} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^m \|\vec{x}_0 - \vec{x}_*\|$$

short summary on Krylov subspace method

- CG, FOM, and GMRES are direct method ? yes and no
  if exact arithmetic is possible, CG and FOM for a positive matrix (symmetric positive definite or coercive) can find the exact solution after n_0 iterations
- Due to numerical round of error, orthogonality of Lanczos process is rapidly lost in practice
- Since we need approximate solution normally, Krylov subspace method is useful with termination of iteration by certain criteria before n_0 iterations
- FOM and GMRES require to store Arnoldi basis vector and computational complexity of Arnoldi process is large, but by short iterations realized by good preconditioner, these methods are robust and practical.
- residual of GMRES decreases monotonically but there is still no convergence estimate for indefinite matrices
- family of BiCG method has no monotonic decreasing in residual and in the worst case bi-orthogonal Lanczos process breaks, though look-ahead technique is employed

incomplete LU factorization as preconditioner

nonzero pattern $NZ(A) := \{(i,j)\,;\,a_{ij} \neq 0\}$

Algorihm (ILU(0))

do i=2,$\cdots,N$

  do k=$1\cdots,i-1\,;\;(i,k)\in NZ(A)$

    $a_{ik} = a_{ik}/a_{kk}$

    do j=$k+1,\cdots,N\,;\;(i,j)\in NZ(A)$

      $a_{ij} = a_{ij} - a_{ik}/a_{kj}$

subroutine in intel Math Kernel Library (MKL)

```
int nrow, ierr;
double *coefs, *ilu;  // non-zero values
int *ia, *ja;         // CSR non-zero indexes
int ipar[128];        // ipar[30]=1 to conitnue for 0 diagonal
double dpar[128];     // dpar[30]=1.0e-16, dpar[31]=1.0e-10

dcsrilu0(&nrow, coefs, ia, ja, ilu, ipar, dpar, &ierr);
```

Schwarz methods as preconditioner

overlapping decomposition of the matrix $\Lambda = \bigcup_{p=1}^{P} \Lambda_p, \ \Lambda_p \cap \Lambda_q \neq \emptyset$

- R_p: restriction from the total DOF to sub-matrix : $\Lambda \to \Lambda_p$
- D_p : discrete representation of partition of the unity

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

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

ASM preconditioner

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

ASM does not converge as fixed point iteration, but M_{ASM}^{-1} is symmetric and works well as a preconditioner for CG method.

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 does converge but M_{RAS}^{-1} is not symmetric and then works as a preconditioner for GMRES method.

non overlapping decomposition of the matrix by METIS : 1/2

```
int nrow, nnz;
int xadj[nrow];        // connectivity of the sparse matrix
int adjcy[nnz - nrow]; // excluding diagonal from CSR
int part[nrow];
int ncon = 1, objval;
idx_t options[METIS_NOPTIONS] = {0} ;
METIS_SetDefaultOptions(options);
options[METIS_OPTION_NUMBERING] = 0;
options[METIS_OPTION_DBGLVL] = METIS_DBG_INFO;
METIS_PartGraphRecursive(&nrow, &ncon, xadj, adjcy,
                         NULL, NULL, NULL,
                         &nparts,
                         NULL, NULL,
                         options, &objval, part);
```

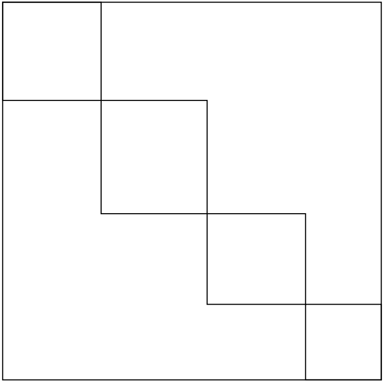
overlapping decomposition by extension with connected entries from non-overlapping

$$\Lambda = \bigcup_{p=1}^{P} \Lambda_p, \ \Lambda_p \cap \Lambda_q \neq \emptyset, \text{ : non-overlapping decomposition}$$

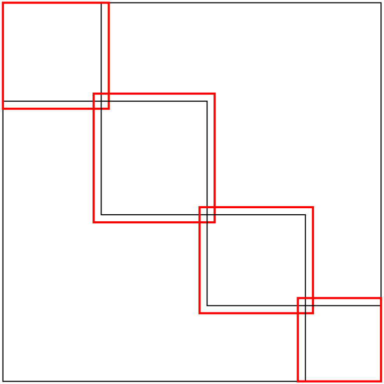
$$\Lambda_p^{(1)} = \{j \in \Lambda_p, \ a_{ij} \neq 0, i \in \Lambda_p\}$$

non overlapping decomposition of the matrix by METIS : 2/2

decomposition of the matrix into overlapping sub-matrices
only by information on connectivity of the sparse matrix



non-overlapping decomposition by METIS



one-layer extension

repeated extension

$$\{\Lambda_p\}_{p=1}^{P} \rightarrow \left\{\Lambda_p^{(1)}\right\}_{p=1}^{P} \rightarrow \left\{\Lambda_p^{(2)}\right\}_{p=1}^{P}$$

sparse matrix format : 1/3

n : # of rows

nnz : # of nonzeros

$[A]_{i\,j}$: nonzero entries at $(i,\ j)$

- COO (Coordinate) format `MUMPS`

```
int irow[nnz];
int jcol[nnz];
double coef[nnz];
```

- CSR (Compressed Sparse Row) /
  CRS (Compressed Row Storage) format `Pradiso`, `Dissection`

```
int ptrow[n+1];
int indcol[nnz];
double coef[nnz];
```

$[A]_{i\,j}$=coef[k]

j=indcol[k], ptrow[i] $\leq k <$ ptrow[$i+1$]

sparse matrix format, zero-based index : 2/3

an example, 5×5 unsymmetric matrix, $n = 5$, $nnz = 15$.

1.1	1.2		1.4	
2.1	2.2	2.3		2.5
	3.2	3.3		
4.1			0.0	4.5
	5.2		5.4	5.5

	0	1	2	3	4
0	0	1		2	
1	3	4	5		6
2		7	8		
3	9			10	11
4		12		13	14

i	0			1				2		3			4			5
ptrow[i]	0			3				7		9			12			15
indcol[k]	0	1	3	0	1	2	4	1	2	0	3	4	1	3	4	
coef[k]	1.1	1.2	1.4	2.1	2.2	2.3	2.5	3.2	3.3	4.1	0.0	4.5	5.2	5.4	5.5	

- diagonal entry should exist even if the value is 0
- `indcol[]` should be in ascending order in each row

sparse matrix format, zero-based index : 3/3

5×5 symmetric matrix, upper triangular, $n = 5$, $nnz = 10$.

1.1	1.2		1.4	
	2.2	2.3		2.5
		3.3		
			0.0	4.5
				5.5

	0	1	2	3	4
0	0	1		2	
1		3	4		5
2			6		
3				7	8
4					9

i	0			1			2	3		4	5
`ptrow[`i`]`	0			3			6	7		9	10
`indcol[`k`]`	0	1	3	1	2	4	2	3	4	4	
`coef[`k`]`	1.1	1.2	1.4	2.2	2.3	2.5	3.3	0.0	4.5	5.5	

- upper triangular matrix is accepted by `Pardiso`

usage of Paridso

```
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
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 instaed 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
```

Reverse Communication Interface

in `FGMRES` in Intel MKL, user needs to write `SpMV` (sparse-matrix vector product) and preconditioner
`RCI_request` shows stage of the GMRES operation

```
ipar[1] = 6; ipar[4] = max_iter;  ipar[14] = max_iter;
int size_tmp  =((2*ipar[14]+1)*nrow+
                (ipar[14]*(ipar[14]+9))/2 + 1);
double *tmp = new double[size_tmp];  // to keep Arnoldi basis
dfgmres_init(&nrow, sol, rhs, &RCI_request, ipar, dpar, tmp);

int m = 0;  // counter for GMRES iteration
while (1) {
  dfgmres (&nrow, sol, rhs, &RCI_request, ipar, dpar, tmp);
  if (RCI_request <= 0) break;  // stoping criteria satisfied
  if (RCI_request == 1) {
    fprintf(stderr, "%d %g\n", m, dpar[4]); // print resiudal
    mkl_cspblas_dcsrgemv(&cvar, &nrow, a.coefs, a.ia, a.ja,
                         &tmp[ipar[21] - 1],
                         &tmp[ipar[22] - 1]);        // SpMV
    // additive Schwarz preconditioner -> tmp[ipar[22] - 1]
  }
  m++;
}
```

Numerical example : 1/4

stoping criteria : relative residual $\leq 10^{-12}$
Navier equations : elasticity problem descretized by P2 finite element

$N = 750,141$, $nnz = 31,214,610$: strctured mesh generated from 32^3 cubes

`Navier3D.32.P2`

		elapsed time (sec.)			
overlap	# iteration	total	LU-factorization	iteration	error
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

$N = 677,163$, $nnz = 28,853,844$: unstrctured mesh (with mesh refinement)

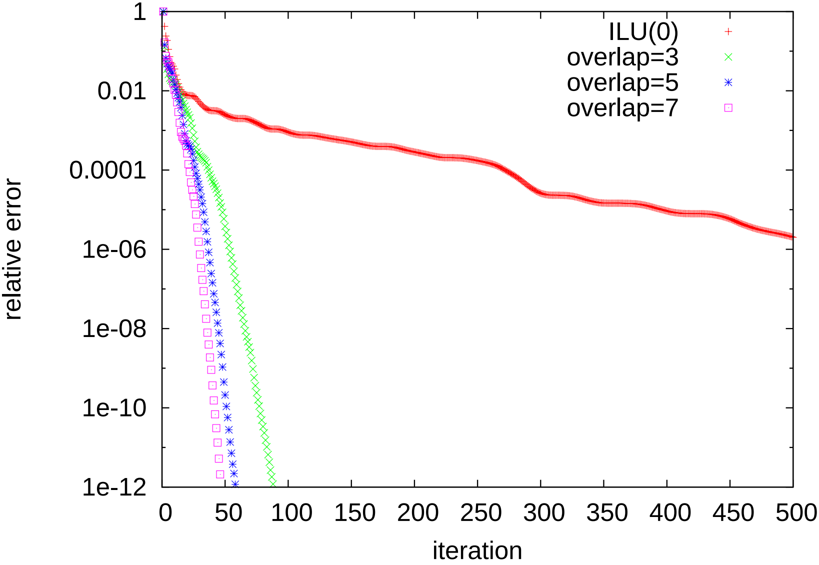
`Navier3DmeshP2`

		elapsed time (sec.)			
overlap	# iteration	total	LU-factorization	iteration	error
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) preconditioner is not strong enough for unstructured mesh problem

Numerical example : 2/4

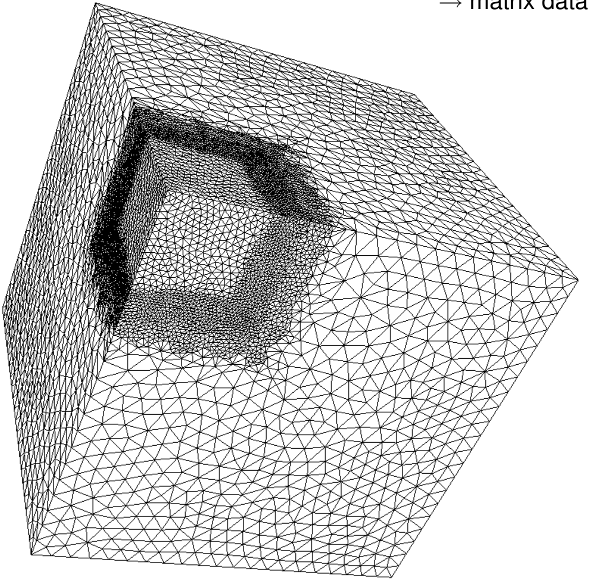
convergence history



Numerical example : 3/4

unstructured mesh generated by `tetgen` and `FreeFem++`
P2 finite element, $N = 677,163$, $nnz = 28,853,844$

$\rightarrow$ matrix data `Navier3DmeshP2`



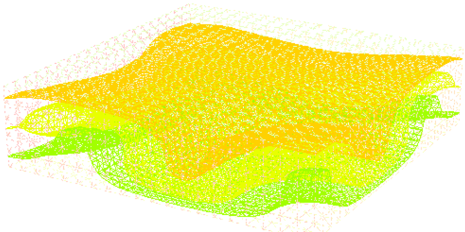
Numerical example : 3/4

non-dimensional Rayleigh-Bénard equations at stationary state

$$\frac{1}{Pr} u \cdot \nabla u - 2\nabla \cdot D(u) + \nabla p = Ra\theta \vec{e}_2 \text{ in } \Omega \,,$$
$$\nabla \cdot u = 0 \text{ in } \Omega \,,$$
$$u \cdot \nabla \theta - \triangle \theta = 0 \text{ in } \Omega$$
$$u \cdot n = 0 \text{ on } \partial\Omega \,,$$
$$\theta = 1 \text{ on } \Gamma_1 \,, \theta = 0 \text{ on } \Gamma_3 \,, \partial_n \theta = 0 \text{ on } \Gamma_2 \cup \Gamma_4 \,.$$

The stiffness matrix is unsymmetric and indefinite.

$$K = \begin{bmatrix} A_0 + A_1(\vec{u_n}) & B^T & D_{Ra} \\ B & -\epsilon M & 0 \\ C_2(\vec{\theta_n}) & 0 & C_0 + C_1(\vec{u}_n) \end{bmatrix}$$



temperature distribution of a stationary state → `RayleighBenard3D`

Hands-on tutorial

in the frontend of CMC, `octopus.hpc.cmc.osaka-u.ac.jp`
`/octfs/apl/kosyu/20181121/` contains materials

`-- matrix/` : matrix data in matrix-makert format generated by `FreeFem++`
`src/` : sources, rci-GMRES-RAS.cpp etc.

- `LECTURE` dedicated queue of `octopus` only for this seminar
- Intel Compiler ver. 17 is necssary for `mkl_dcsrcoo()` for conversion sparse matrix data from COO to CSR formats on Bash

```
# . /octfs/apl/Intel/psxece2017u5/bin/compilervars.sh intel64
```

- `/octfs/apl/METIS` : metis ver 5.1.0

Makefile for testing program

```
CXX = icpc
CC = icc
LD = $(CXX)
DEBUG = -g -O3
MKL_SHARED = -L/opt/intel/compilers_and_libraries_2017/linux/mkl/
             lib/intel64 \
             -lmkl_intel_lp64 -lmkl_core -lmkl_intel_thread -liomp5
METIS_INCLUDE = -I/octfs/apl/METIS/include
METIS_SHARED = -Xlinker -rpath=/octfs/apl/METIS/lib \
               -L/octfs/apl/METIS/lib -lmetis

rci-GMRES-RAS.o: rci-GMRES-RAS.cpp
        $(CXX) $(DEBUG) $(METIS_INCLUDE) -I. -c rci-GMRES-RAS.cpp
rci-GMRES-RAS: rci-GMRES-RAS.o
        $(LD) -o rci-GMRES-RAS rci-GMRES-RAS.o $(MKL_SHARED) \
        $(METIS_SHARED)
```

Rerefences

- M. R. Hestenes, E. Stiefel, "Methods of conjugate gradients for solving linear systems". Journal of Research of the National Bureau of Standards. 49 409-435 (1952)
- Y. Saad, "Krylov subspace methods for solving large unsymmetric linear systems", Mathematics of Computation, 37 105-126 (1981)
- Y. Saad, M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems", SIAM J. Sci. Stat. Comput., 7 856-869 (1986)
- R. E. Bank, T. F. Chan, "An anaylsis of the composite step biconjugate gradient method", Numer Math. 66, 295-320 (1993)
- R. W. Freund, N. M. Nachtigal, "QMR: a quasi-minimal residual method for non-Hermitian linear systems", Numer. Math. 60 315-339 (1991)
- B. N. Parlett, D. R. Taylor, Z. A. Liu, "A look-ahead Lanczos algorithm for unsymmetric matrices", Mathematics of Computation, 44 105-124 (1985)
- P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 10 36-52 (1989)
- H. A. van der Vorst, "Bi-CGSTAB : a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems", SIAM J. Sci. Stat. Comput., 12 631-644 (1992)
- G. Karypis, V. Kuma, "A fast and high quality multilevel scheme for partitioning irregular graphs",SIAM Journal on Scientific Computing, 20 359–392 (1998)
- O. Schenk, K. Gärtner, "Solving unsymmetric sparse systems of liner equations with PARDISO", Future Generation Computer Systems, 20 457-487 (2004)