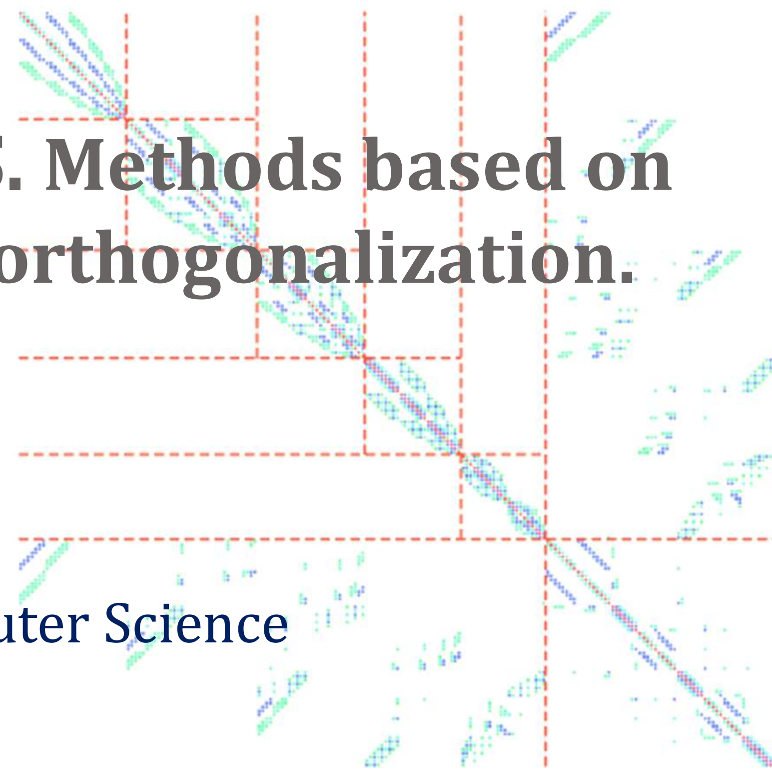
Numerical Methods of Linear Algebra for Sparse Matrices

Lecture 15

Comparison of FOM and GMRES. Methods based on Lacnzoc orthogonalization and biorthogonalization. Preconditioning techniques



Anna Nasedkina

Department of Mathematical Modeling

Institute of Mathematics, Mechanics and Computer Science

Southern Federal University

Krylov Subspace methods based on Arnoldi's orthogonalization

Comparison of FOM and GMRES

Optimal and efficient methods

Full Orthogonalization method (FOM)

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$, calculate residual norm $\beta = \|r_0\|_2$ and the first vector $v_1 = \dfrac{r_0}{\beta}$.
2. Define $H_m = \{h_{ij}\} \in \mathbb{C}^{m\times m}$, $V_m \in \mathbb{C}^{n\times m}$. Set $H_m = 0$
3. Build the basis $\{v_1, v_2, ..., v_m\}$ for Krylov subspace $K_m(A, r_0)$ using Arnoldi process of orthogonalization. Obtain the matrix $V_m = \{v_1, v_2, ..., v_m\}$ and upper-Hessenberg matrix H_m.
4. Find y_m from solving the system $H_m y = \beta e_1$
5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.
6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

In step 4 we find y from the system $H_m y = \beta e_1$ with m equations and m unknowns

$$y = y_m = \begin{pmatrix} y_m^1, & y_m^2, & \dots & y_m^m \end{pmatrix}$$

$$\begin{cases} h_{11}y_m^1 + h_{12}y_m^2 + \quad \dots \quad + h_{1m}y_m^m = \beta \\ h_{21}y_m^1 + h_{22}y_m^2 + \quad \dots \quad + h_{2m}y_m^m = 0 \\ \dots \\ \qquad h_{m,m-1}y_m^{m-1} + h_{mm}y_m^m = 0 \end{cases}$$

- The residual in FOM is

$$r_m = -h_{m+1,m} e_m^T y_m v_{m+1}$$

- The residual norm is

$$\|r_m\|_2 = h_{m+1,m} |e_m^T y_m|$$

Generalized Minimal Residual method (GMRES)

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$, calculate residual norm $\beta = \|r_0\|_2$ and the first vector $v_1 = \dfrac{r_0}{\beta}$.

2. Define $\overline{H}_m = \{h_{ij}\} \in \mathbb{C}^{(m+1)\times m}$, $V_m \in \mathbb{C}^{n\times m}$. Set $\overline{H}_m = 0$

3. Build the basis $\{v_1, v_2, ..., v_m\}$ for Krylov subspace $K_m(A, r_0)$ using Arnoldi process of orthogonalization. Obtain the matrix $V_m = \{v_1, v_2, ..., v_m\}$ and upper-Hessenberg matrix $\overline{H}_m$.

4. Find y_m as a minimizer of the functional $J(y) = \left\| \beta e_1 - \overline{H}_m y \right\|_2$

5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.

6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

In step 4 we find y from the system $\overline{H}_m y = \beta e_1$
with $m+1$ equations and m unknowns (overdefined system)

$$y = y_m = \left(y_m^1, \quad y_m^2, \quad \dots \quad y_m^m \right)$$

$$\begin{cases} h_{11} y_m^1 + h_{12} y_m^2 + \quad \dots \quad + h_{1m} y_m^m = \beta \\ h_{21} y_m^1 + h_{22} y_m^2 + \quad \dots \quad + h_{2m} y_m^m = 0 \\ \dots \\ \qquad h_{m,m-1} y_m^{m-1} + h_{mm} y_m^m = 0 \\ \qquad h_{m+1,m} y_m^m = 0 \end{cases}$$

- The residual in GMRES is
  $$r_m = V_{m+1}\left(\beta e_1 - \overline{H}_m y_m\right) = $$
  $$= V_{m+1}\, Q_m^T (\gamma_{m+1} e_{m+1})$$
- The residual norm is
  $$\|r_m\|_2 = |\gamma_{m+1}|$$

Summary: FOM and GMRES based on Arnoldi orthogonalization process

FOM and GRMRES are Krylov subspace methods: $x_m = x_0 + \delta_m$, $\delta_m \in K_m(A, r_0)$ is the correction vector, $K_m(A, r_0) = span\{r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0\}$ is Krylov subspace

Orthogonal basis of $K_m(A, r_0)$ is produced by Arnoldi process, the basis vectors are the columns of V_m.

Correction vector $\delta_m \in K_m(A, r_0) \Leftrightarrow \exists y \in C^m : \delta_m = V_m y$

- Arnoldi relation for FOM: $AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T$, $V_m^H A V_m = H_m$
- Arnoldi relation for GMRES: $AV_m = V_{m+1} \overline{H}_m$, $V_{m+1}^T A V_m = \overline{H}_m$

- **FOM**

We choose $y = y_{FOM}$ such, that

$r_m \perp K_m(A, r_0)$

Equivalent condition:

$\beta e_1 - H_m y_{FOM} = 0, \beta = \|r_0\|_2$

We find $y = y_{FOM}$ from the linear system: $H_m y_{FOM} = \beta e_1$

- **GMRES**

We choose $y = y_{GMRES}$ such, that

$$\|r_m\|_2 = \min_{\delta \in K_m(A, r_0)} \|b - A(x_0 + \delta)\|_2$$

The norm of the residual:

$\|r_m\|_2 = \left\|\beta e_1 - \overline{H}_m y_{GMRES}\right\|_2$

We find $y = y_{GMRES}$ from the least squares problem: $\left\|\beta e_1 - \overline{H}_m y_{GMRES}\right\|_2 \to \min$

Optimal and efficient methods

- FOM and GMRES are not efficient methods, because as m grows, we have to store more and more basis vectors
- FOM and GMRES are optimal methods. In absence of round-off errors they have the same solution
- For general matrices it is not possible to derive an iterative method that would be efficient and yet optimal
- It is possible to derive an <u>efficient and optimal</u> method for Hermitian matrices, where the solution is obtained by a <u>short-term recurrence</u>
- *Examples*: CG (derived from FOM) and MINRES (derived from GMRES)

Lanczos method of orthogonalization for symmetric (Hermitian) matrices

- Algorithm of Lanczos orthogonalization
- Lanczos method for symmetric (Hermitian) systems
- Direct Lanczos method
- Conjugate Gradient (CG)
- Conjugate Residual (CR)
- Generalized Conjugate Residual (GCR)

Lanzsoc process of orthogonalization for symmetric (Hermitian) systems

Consider the case when matrix A is symmetric, then Hessenberg matrix H_m will be tridiagonal.

Theorem. When Arnoldi algorithm is applied to a real symmetric matrix A ($A = A^T$), then $h_{i,j} = 0,\ 1 \le i < j-1;\ h_{j+1,j} = h_{j,j+1},\ j = \overline{1,m}$. I. e. H_m is tridiagonal.

$$H_m = \begin{pmatrix} h_{11} & h_{12} & \dots & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & \dots & h_{2m} \\ 0 & h_{32} & \dots & \dots & h_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & h_{m,m-1} & h_{mm} \end{pmatrix}.$$ For symmetric (Hermitian) $A \Rightarrow$

$$H_m = T_m = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & 0 & \dots & \dots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & 0 & \dots & \dots & 0 \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ 0 & \dots & \dots & 0 & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & \dots & \dots & 0 & 0 & \beta_m & \alpha_m \end{pmatrix};$$

$T_m = tridiag(\beta_i, \alpha_i, \beta_{i+1})$ is symmetric (Hermitian)

Lanzsoc algorithm of orthogonalization for symmetric systems

1) Choose v_1 : $\|v_1\|_2 = 1$. Matrix $A = A^T$ and dimension of subspace m are given.

$\beta_1 := 0,\ v_0 := 0$

2) Loop for j from 1 to m:

2.1) $w_j := Av_j - \beta_j v_{j-1}$

2.2) $\alpha_j := (w_j, v_j)$

2.3) $w_j := w_j - \alpha_j v_j$

2.4) $\beta_{j+1} := \|w_j\|_2$. If $\beta_{j+1} = 0 \Rightarrow$ stop, else

2.5) $v_{j+1} := \dfrac{w_j}{\beta_{j+1}}$

Lanczos relation

$$AV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T$$

$$H_m = V_m^T A\, V_m = T_m$$

- Matrix T_m is symmetric tridiagonal

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & & \dots & \dots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}$$

Lanzsoc method for symmetric (Hermitian) linear systems

- Similar to FOM based on Arnoldi orthogonalization, for symmetric systems we can derive Lanczos method based on Lanczos orthogonalization
- This can be regarded as a special case of FOM for symmetric systems

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$, calculate residual norm $\beta = \|r_0\|_2$, and the first vector $v_1 = \frac{r_0}{\beta}$
2. Define $T_m = tridiag(\beta_j, \alpha_j, \beta_{j+1})$.
3. Build the basis for Krylov subspace using Lanczos process:
   Loop for j from 1 to m:
   3.1) $w_j := Av_j - \beta_j v_{j-1}$ (If $j = 1$, set $\beta_1 v_0 = 0$)
   3.2) $\alpha_j := (w_j, v_j)$
   3.3) $w_j := w_j - \alpha_j v_j$
   3.4) $\beta_{j+1} := \|w_j\|_2$. If $\beta_{j+1}=0 \Rightarrow m := j$, go to step 4, else
   3.5) $v_{j+1} := \frac{w_j}{\beta_{j+1}}$
4. Find y_m from solving the system $T_m y = \beta e_1$
5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.
6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

- The residual is

$$r_m = -\beta_{m+1} e_m^T y_m v_{m+1}$$

- The residual norm is

$$\|r_m\|_2 = \beta_{m+1} |e_m^T y_m|$$

Derivation of Direct Lanczos method

Now we'll derive the method which solves the system $T_m y = \beta e_1$
with tridiagonal matrix T_m directly.
Use LU-factorization of $T_m = L_m U_m$

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & & \dots & \dots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ \mu_2 & 1 & & & \\ & & \dots & & \\ & & \mu_{m-1} & 1 & \\ & & & \mu_m & 1 \end{pmatrix} \begin{pmatrix} \eta_1 & \beta_2 & & & \\ & \eta_2 & \beta_3 & & \\ & & \dots & \dots & \\ & & & \eta_{m-1} & \beta_m \\ & & & & \eta_m \end{pmatrix}$$

Here $\beta_2, \dots, \beta_m$ and $\alpha_1, \dots, \alpha_m$ are the coefficients from Lanczos algorithm.
We need to find coefficients of LU-factorization $\mu_2, \dots, \mu_m$ and $\eta_1, \dots, \eta_m$
Direct Lanczos method will use recursions.

From matrix multiplication in LU-factorization above:

$$\alpha_m = \mu_m \beta_m + \eta_m \cdot 1 \Rightarrow \eta_m = \alpha_m - \mu_m \beta_m \qquad (1)$$

$$\beta_m = \mu_m \eta_{m-1} + 1 \cdot 0 \Rightarrow \mu_m = \frac{\beta_m}{\eta_{m-1}} \qquad (2)$$

Derivation of Direct Lanczos method

Formally $y_m = T_m^{-1}\beta e_1 = (L_m U_m)^{-1}\beta e_1 = U_m^{-1}L_m^{-1}\beta e_1$

Approximate solution is given by

$$x_m = x_0 + V_m y_m = x_0 + V_m T_m^{-1}\beta e_1 = x_0 + \underline{V_m U_m^{-1}}\,\underline{\underline{L_m^{-1}\beta e_1}}$$

Denote $P_m = V_m U_m^{-1}$ and $z_m = L_m^{-1}\beta e_1$. Hence $x_m = x_0 + P_m z_m$

Consider the columns of $P_m = \begin{pmatrix} | & | & & | \\ p_1 & p_2 & \cdots & p_m \\ | & | & & | \end{pmatrix}$

For the formula $x_m = x_0 + P_m z_m$ Direct Lanczos derives the way to compute x_m using vectors p_i (the columns of P_m).

$$P_m = V_m U_m^{-1} \Rightarrow V_m = P_m U_m$$

Last column of V_m : $v_m = \beta_m p_{m-1} + \eta_m p_m \Rightarrow$

recursion for p_m : $p_m = \dfrac{1}{\eta_m}(v_m - \beta_m p_{m-1}) \qquad (3)$

Derivation of Direct Lanczos method

Consider the vector and $z_m = L_m^{-1}\beta e_1$ with unknown components $\xi_1, \xi_2, \ldots, \xi_m$ to find.

$$z_m = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \ldots \\ \ldots \\ \xi_m \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ \mu_2 & 1 & & & \\ & & \ldots & & \\ & & \mu_{m-1} & 1 & \\ & & & \mu_m & 1 \end{pmatrix}^{-1} \begin{pmatrix} \beta \\ 0 \\ \ldots \\ \ldots \\ 0 \end{pmatrix}; \; L_m^{-1} = \begin{pmatrix} 1 & & & & \\ -\mu_2 & 1 & & & \\ & & \ldots & & \\ -\mu_{m-1}\ldots\mu_2 & & -\mu_{m-1} & 1 & \\ -\mu_m\mu_{m-1}\ldots\mu_2 & \ldots & -\mu_m\mu_{m-1} & -\mu_m & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & & & & \\ -\mu_2 & 1 & & & \\ & & \ldots & & \\ -\mu_{m-1}\ldots\mu_2 & & -\mu_{m-1} & 1 & \\ -\mu_m\mu_{m-1}\ldots\mu_2 & \ldots & -\mu_m\mu_{m-1} & -\mu_m & 1 \end{pmatrix} \begin{pmatrix} \beta \\ 0 \\ \ldots \\ \ldots \\ 0 \end{pmatrix} = \begin{pmatrix} \beta \\ -\mu_2\beta \\ \ldots \\ \ldots \\ -\mu_m\mu_{m-1}\ldots\mu_2\beta \end{pmatrix}.$$

$$\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \ldots \\ \xi_m \end{pmatrix} = \begin{pmatrix} \beta \\ -\mu_2\beta \\ -\mu_3\mu_2\beta \\ \ldots \\ -\mu_m\mu_{m-1}\ldots\mu_2\beta \end{pmatrix} \Rightarrow \begin{array}{c} \xi_1 = \beta \\ \xi_2 = -\mu_2\beta = -\mu_2\xi_1 \\ \xi_3 = -\mu_3\mu_2\beta = -\mu_3\xi_2 \\ \ldots \\ \xi_m = -\mu_m\mu_{m-1}\ldots\mu_2\beta = -\mu_m\xi_{m-1} \end{array}$$

From this formula we get recursion for ξ_m: $\xi_m = -\mu_m\xi_{m-1}$ (4)

Derivation of Direct Lanczos method

$$\text{As } z_m = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \dots \\ \xi_m \end{pmatrix} = \begin{pmatrix} \beta \\ -\mu_2\xi_1 \\ -\mu_3\xi_2 \\ \dots \\ -\mu_m\xi_{m-1} \end{pmatrix} ;\ z_{m-1} = \begin{pmatrix} \beta \\ -\mu_2\xi_1 \\ -\mu_3\xi_2 \\ \dots \\ -\mu_{m-1}\xi_{m-2} \end{pmatrix} \Rightarrow z_m = \begin{pmatrix} | \\ z_{m-1} \\ | \\ \xi_m \end{pmatrix}$$

Lastly, we derive recursions for x_m

$$x_m = x_0 + P_m z_m$$

$$x_1 = x_0 + P_1 z_1 = x_0 + p_1 z_1 = x_0 + p_1\xi_1$$

$$x_2 = x_0 + P_2 z_2 = x_0 + p_1\xi_1 + p_2\xi_2 = x_1 + p_2\xi_2 \ \Rightarrow$$

$$x_m = x_{m-1} + p_m\xi_m \qquad (5)$$

These five recursions for the search directions p_m, coefficients μ_m, η_m, ξ_m and for approximate solution x_m give an algorithm that solves the system $T_m y = \beta e_1$ progressively by using Gaussian elimination without partial pivoting

Direct Lanczos method (D-Lanczos)

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$,

calculate residual norm $\xi_1 = \beta = \|r_0\|_2$, and the first vector $v_1 = \dfrac{r_0}{\beta}$.

Set $\mu_1 := \beta_1 := 0$; $p_0 = 0$

2. Start iterative process for $m = 1, 2, \ldots.$ until convergence

2.1) $w := Av_m - \beta_m v_{m-1}$

2.2) $\alpha_m = (w, v_m)$

2.3) If $m > 1$, then $\mu_m = \dfrac{\beta_m}{\eta_{m-1}}$; $\xi_m = -\mu_m \xi_{m-1}$

2.4) $\eta_m = \alpha_m - \mu_m \beta_m$

2.5) $p_m = \dfrac{1}{\eta_m}(v_m - \beta_m p_{m-1})$

2.6) $x_m = x_{m-1} + p_m \xi_m$ Check convergence of x_m. If not converged, then

2.7) $w := w - \alpha_m v_m$

2.8) $\beta_{m+1} = \|w\|_2$, $v_{m+1} := \dfrac{w}{\beta_{m+1}}$

Residuals and search directions in D-Lanczos

In Lanczos method for symmetric systems $r_m = -\beta_{m+1} e_m^T y_m v_{m+1} = \tau_m v_{m+1}, P_m = V_m U_m^{-1}$ where τ_m is a scalar

Since vectors v_m, $m = 1, 2, ...$ are orthonormal: $(v_i, v_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$ and

$r_m = \tau_m v_{m+1} \Rightarrow$ residuals r_m are orthogonal: $(r_i, r_j) = \begin{cases} 0, & i \neq j \\ \neq 0, & i = j \end{cases}$

Recall that $P_m = V_m U_m^{-1}$. Consider $P_m^T A P_m$ ($P_m^H A P_m$ in complex space)

$P_m^T A P_m = (V_m U_m^{-1})^T A V_m U_m^{-1} = U_m^{-T} \underline{V_m^T A V_m} U_m^{-1} = U_m^{-T} \underline{\underline{T_m U_m^{-1}}} = U_m^{-T} L_m,$

since $V_m^T A V_m = T_m$ and $T_m = L_m U_m \Rightarrow T_m U_m^{-1} = L_m$

Hence $P_m^T A P_m = U_m^{-T} L_m$ is symmetric lower triangular matrix, hence it is diagonal:

$P_m^T A P_m = diag(d_1, d_2, ...d_m) \Rightarrow$

$(Ap_i, p_j) = \begin{cases} 0, & i \neq j \\ d_i, & i = j \end{cases}$

This property is called *A-conjugacy*. Thus search directions p_m are A-conjugate.

Residuals and search directions in D-Lanczos

Theorem. Let r_m are the residual vectors, p_m are the search direction vectors from Lanczos method and D-Lanczos , $m = 0,1,2,\ldots.$ Then

1) residuals r_m are orthogonal: $(r_i, r_j) = 0,\ i \neq j$

2) search directions p_m are A-conjugate: $(Ap_i, p_j) = 0,\ i \neq j$

What about (Ap_i, p_i)?

$(Ap_i, p_i) = d_i$ diagonal entry of diagonal matrix $P_m^T A P_m$

$(Ap_i, p_i) \neq 0$ when A is positive definite

In this case $(Ap_i, p_i) > 0$

The method, which uses the condition for matrix A to be symmetric (Hermitian) positive definite is called *conjugate gradient method*.
See the derivation of this method in Yosef Saad book, p. 199 and lecture notes.

Conjugate Gradient (CG) method

- A is a symmetric positive definite matrix.
- We impose conditions of orthogonality and conjugacy:

Residuals r_j are orthogonal: $(r_i, r_j) = 0, i \neq j$

Search directions p_j are A-conjugate: $(Ap_i, p_j) = 0, i \neq j$

Note that $(Ap_j, p_j) \neq 0$ for a positive definite matrix.

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$; $p_0 := r_0$
2. Start iterative process for $j = 0, 1, 2, ...$ until convergence

2.1) $\alpha_j = \dfrac{(r_j, r_j)}{(Ap_j, p_j)}$

2.2) $x_{j+1} = x_j + \alpha_j p_j$

2.3) $r_{j+1} = r_j - \alpha_j Ap_j$

2.4) $\beta_j = \dfrac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}$

2.5) $p_{j+1} = r_{j+1} + \beta_j p_j$

Relation between Lanczos method and CG

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & & \cdots & \cdots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_0} & \frac{\sqrt{\beta_0}}{\alpha_0} & & & \\ \frac{\sqrt{\beta_0}}{\alpha_0} & \frac{1}{\alpha_1}+\frac{\beta_0}{\alpha_0} & \frac{\sqrt{\beta_1}}{\alpha_1} & & \\ & & \cdots & \cdots & \\ & & \frac{\sqrt{\beta_{m-3}}}{\alpha_{m-3}} & \frac{1}{\alpha_{m-2}}+\frac{\beta_{m-3}}{\alpha_{m-3}} & \frac{\sqrt{\beta_{m-2}}}{\alpha_{m-2}} \\ & & & \frac{\sqrt{\beta_{m-2}}}{\alpha_{m-2}} & \frac{1}{\alpha_{m-1}}+\frac{\beta_{m-2}}{\alpha_{m-2}} \end{pmatrix}$$

Coefficients from Lanczos method

Coefficients from CG

Conjugate Residual (CR) method

Similar to GMRES based on Arnoldi orthogonalization, for symmetric (Hermitian) systems we can derive Conjugate residual method based on Lanczos orthogonalization

- A is a Hermitian matrix

Residuals r_j are A-conjugate: $(Ar_i, r_j) = 0,\ i \neq j$

Vectors Ap_j are orthogonal : $(Ap_i, Ap_j) = 0,\ i \neq j$

Note that for $A = A^H$ $(Ar_i, r_j) = (r_i, A^H r_j) = (r_i, Ar_j)$

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0;\ p_0 := r_0$
2. Start iterative process for $j = 0, 1, 2, \ldots.$ until convergence

2.1) $\alpha_j = \dfrac{(r_j, Ar_j)}{(Ap_j, Ap_j)}$

2.2) $x_{j+1} = x_j + \alpha_j p_j$

2.3) $r_{j+1} = r_j - \alpha_j Ap_j$

2.4) $\beta_j = \dfrac{(r_{j+1}, Ar_{j+1})}{(r_j, Ar_j)}$

2.5) $p_{j+1} = r_{j+1} + \beta_j p_j$

2.6) Compute Ap_{j+1} for the next iteration: $Ap_{j+1} = Ar_{j+1} + \beta_j Ap_j$

Generalized Conjugate Residual (GCR) method

In the case of nonsymmetric (non-Hermitian) matrix the recursions cannot be short

- A is a nonsymmetric matrix

Residuals r_j are A-conjugate: $(Ar_i, r_j) = 0, i \neq j$

Vectors Ap_j are orthogonal : $(Ap_i, Ap_j) = 0, i \neq j$

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$; $p_0 := r_0$
2. Start iterative process for $j = 0, 1, 2, \ldots.$ until convergence

2.1) $\alpha_j = \dfrac{(r_j, Ap_j)}{(Ap_j, Ap_j)}$

2.2) $x_{j+1} = x_j + \alpha_j p_j$

2.3) $r_{j+1} = r_j - \alpha_j Ap_j$

2.4) $\beta_{ij} = \dfrac{-(Ar_{j+1}, Ap_i)}{(Ap_i, Ap_i)}, \ i = \overline{0, j}$

2.5) $p_{j+1} = r_{j+1} + \sum_{i=0}^{j} \beta_{ij} p_j$

Krylov Subspace methods based on Lanczos biorthogonalization

Lanczos method of biorthogonalization for nonsymmetric matrices

- Lanzos method for nonsymmetric systems
- Biconjugate Gradient (BiCG)

The idea of Lanzsoc biorthogonalization

The idea of short recursions is possible whenever the projected matrix H_m is tridiagonal. In fact, tridiagonal matrix can be also obtained for nonsymmetric systems.

Consider the case when matrix A is nonsymmetric.

Search subspace: $K = K_m(A, v_1) = span\{v_1 \ , Av_1, A^2 v_1, ..., A^{m-1} v_1\}$

Subspace of constraints: $L = K_m(A^H, w_1) = span\{w_1, A^H w_1, \left(A^H\right)^2 w_1, ..., \left(A^H\right)^{m-1} w_1\}$

The sets of vectors $\{v_1, v_2, ..., v_m\}$ and $\{w_1, w_2, ..., w_m\}$ are called biorthonormal, if

$$(v_i, w_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

The algorithm which constructs two biorthonormal bases for Krylov subspaces $K_m(A, v_1)$ and $K_m(A^H, w_1)$ is called *Lanczos method of biorthogonalization*

Lanczos method of biorthogonalization

Consider real space. The idea is to build two biorthonormal bases for the two subspaces: search subspace $K_m = K_m(A, v_1) = span\{v_1, Av_1, A^2v_1, \dots, A^m v_1\}$ and subspace of constraints $L_m = K_m(A^T, w_1) = span\{w_1, A^T w_1, (A^T)^2 w_1, \dots, (A^T)^{m-1} w_1\}$

1) Choose v_1 and w_1 : $(v_1, w_1) = 1$. Matrix A and dimension of subspace m are given.

Set $\beta_1 = \delta_1 = 0,\ w_0 = v_0 = 0$

2) Loop for j from 1 to m :

2.1) $\alpha_j := (Av_j, w_j)$

2.2) $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$

2.3) $\hat{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}$

2.4) $\delta_{j+1} = \sqrt{\left|(\hat{v}_{j+1}, \hat{w}_{j+1})\right|}$. If $\delta_{j+1} = 0 \Rightarrow$ stop, else

2.5) $\beta_{j+1} = \dfrac{(\hat{v}_{j+1}, \hat{w}_{j+1})}{\delta_{j+1}}$

2.6) $w_{j+1} = \dfrac{\hat{w}_{j+1}}{\beta_{j+1}}$

2.7) $v_{j+1} = \dfrac{\hat{v}_{j+1}}{\delta_{j+1}}$

Properties of Lanczos algorithm of biorthogonalization

The algorithm insures that $(v_{j+1}, w_{j+1}) = 1$

$$\beta_{j+1} = \frac{(\hat{v}_{j+1}, \hat{w}_{j+1})}{\delta_{j+1}} = \frac{(\hat{v}_{j+1}, \hat{w}_{j+1})}{\sqrt{\left|(\hat{v}_{j+1}, \hat{w}_{j+1})\right|}} = \sqrt{\left|(\hat{v}_{j+1}, \hat{w}_{j+1})\right|}$$

As $\delta_{j+1} = \sqrt{\left|(\hat{v}_{j+1}, \hat{w}_{j+1})\right|}$, hence $\beta_{j+1} = \pm\delta_{j+1}$

Lanczos relation for biorthogonalization

1) $AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^T$

2) $A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T$

3) $W_m{}^T A V_m = T_m$

- Matrix T_m is nonsymmetric tridiagonal

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & & \dots & \dots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix}$$

Lanczos method for nonsymmetric systems

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$,

calculate residual norm $\beta = \|r_0\|_2$, and the first vector $v_1 = \frac{r_0}{\beta}$

2. Define $T_m = tridiag(\delta_j, \alpha_j, \beta_{j+1})$.

3. Build the bases for Krylov subspaces $K_m(A, v_1)$ and $K_m(A^H, w_1)$ using Lanczos biorthogonalization process. Generate V_m, W_m, T_m

4. Find y_m from solving the system $T_m y = \beta e_1$

5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.

6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

The residual norm is $\|r_m\|_2 = |\delta_{m+1} e_m^T y_m| \|v_{m+1}\|_2$

Note that the vectors v_j are not orthonormal in this method

Derivation of Biconjugate gradient method

Use LU-factorization of $T_m = L_m U_m$

Formally $y_m = T_m^{-1}\beta e_1 = (L_m U_m)^{-1}\beta e_1 = U_m^{-1}L_m^{-1}\beta e_1$

Approximate solution is given by

$$x_m = x_0 + V_m y_m = x_0 + V_m T_m^{-1}\beta e_1 = x_0 + \underline{V_m U_m^{-1}} L_m^{-1}\beta e_1$$

Denote $P_m = V_m U_m^{-1}$ and $\widetilde{P}_m = W_m L_m^{-T}$.

Consider the columns of $P_m = \begin{pmatrix} | & | & & | \\ p_1 & p_2 & \dots & p_m \\ | & | & & | \end{pmatrix}$, $\widetilde{P}_m = \begin{pmatrix} | & | & & | \\ \widetilde{p}_1 & \widetilde{p}_2 & \dots & \widetilde{p}_m \\ | & | & & | \end{pmatrix}$

Consider $\widetilde{P}_m^T A P_m$ ($\widetilde{P}_m^H A P_m$ in complex space)

$$P_m^T A P_m = (W_m L_m^{-T})^T A V_m U_m^{-1} = L_m^{-1}\underline{W_m^T A V_m} U_m^{-1} = L_m^{-1}\underline{\underline{T_m U_m^{-1}}} = L_m^{-1}L_m = I_m,$$

since $V_m^T A V_m = T_m$ and $T_m = L_m U_m \Rightarrow T_m U_m^{-1} = L_m$

Hence $(Ap_i, \widetilde{p}_j) = \begin{cases} 0, & i \neq j \\ d_i, & i = j \end{cases}$

Biconjugate gradient (BiCG) method

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$.

Choose $\tilde{r}_0$: $(r_0, \tilde{r}_0) \neq 0$. Set $p_0 := r_0$, $\tilde{p}_0 := \tilde{r}_0$

2. Start iterative process for $j = 0,1,2,....$ until convergence

2.1) $\alpha_j = \dfrac{(r_j, \tilde{r}_j)}{(Ap_j, \tilde{p}_j)}$

2.2) $x_{j+1} = x_j + \alpha_j p_j$

2.3) $r_{j+1} = r_j - \alpha_j A p_j$

2.4) $\tilde{r}_{j+1} = \tilde{r}_j - \alpha_j A^T \tilde{p}_j$

2.5) $\beta_j = \dfrac{(r_{j+1}, \tilde{r}_{j+1})}{(r_j, \tilde{r}_j)}$

2.6) $p_{j+1} = r_{j+1} + \beta_j p_j$

2.7) $\tilde{p}_{j+1} = \tilde{r}_{j+1} + \beta_j \tilde{p}_j$

Residuals $\boldsymbol{r_j}$ and $\boldsymbol{\tilde{r}_i}$ are biorthogonal: $(r_j, \tilde{r}_i) = 0, i \neq j$

Search directions $\boldsymbol{p_j}$ and $\boldsymbol{\tilde{p}_i}$ are biconjugate: $(Ap_j, \tilde{p}_i) = 0, i \neq j$

Overview of efficient and optimal methods

optimal not efficient work for $A \neq A^*$	FOM $z_m: \quad r_m \perp \mathcal{K}_m(A, r_0)$	GMRES $z_m: \quad \|r_m\|_2 \to \min_{\mathcal{K}_m(A,r_0)}$
optimal efficient work for $A = A^*$	CG $z_m: \quad r_m \perp \mathcal{K}_m(A, r_0)$ (A must be positive definite)	MINRES $z_m: \quad \|r_m\|_2 \to \min_{\mathcal{K}_m(A,r_0)}$
suboptimal efficient work for $A \neq A^*$	BiCG $z_m: \quad r_m \perp \mathcal{K}_m(A^*, \tilde{r}_0)$	QMR $z_m: \quad \|r_m\|_2$ is "quasi" minimized
suboptimal efficient work for $A \neq A^*$	CGS	TFQMR
suboptimal efficient work for $A \neq A^*$	Hybrid methods: BiCGSTAB and BiCGstab(ℓ)	

In this table z_m is the correction vector for approximate solution

- FOM: Full Orthogonalization Method
- GMRES: Generalized Minimal Residual Method
- CG: Conjugate Gradient method
- MINRES: Minimal Residual Method
- BiCG: BiConjuagate Gradient method
- QMR: Quasi Minimal Residual method
- CGS: Conjugate Gradient Squared method
- TFQMR: Transpose Free QMR method
- BiCGSTAB: BiConjuagate Gradient method Stabilzed with GMRES
- BiCGSTAB(l): BiConjuagate Gradient method Stabilzed with GMRES(l)

Table from M.A. Botchev Lecture notes on "Scientific Computing", 2015

How to select an iterative method?

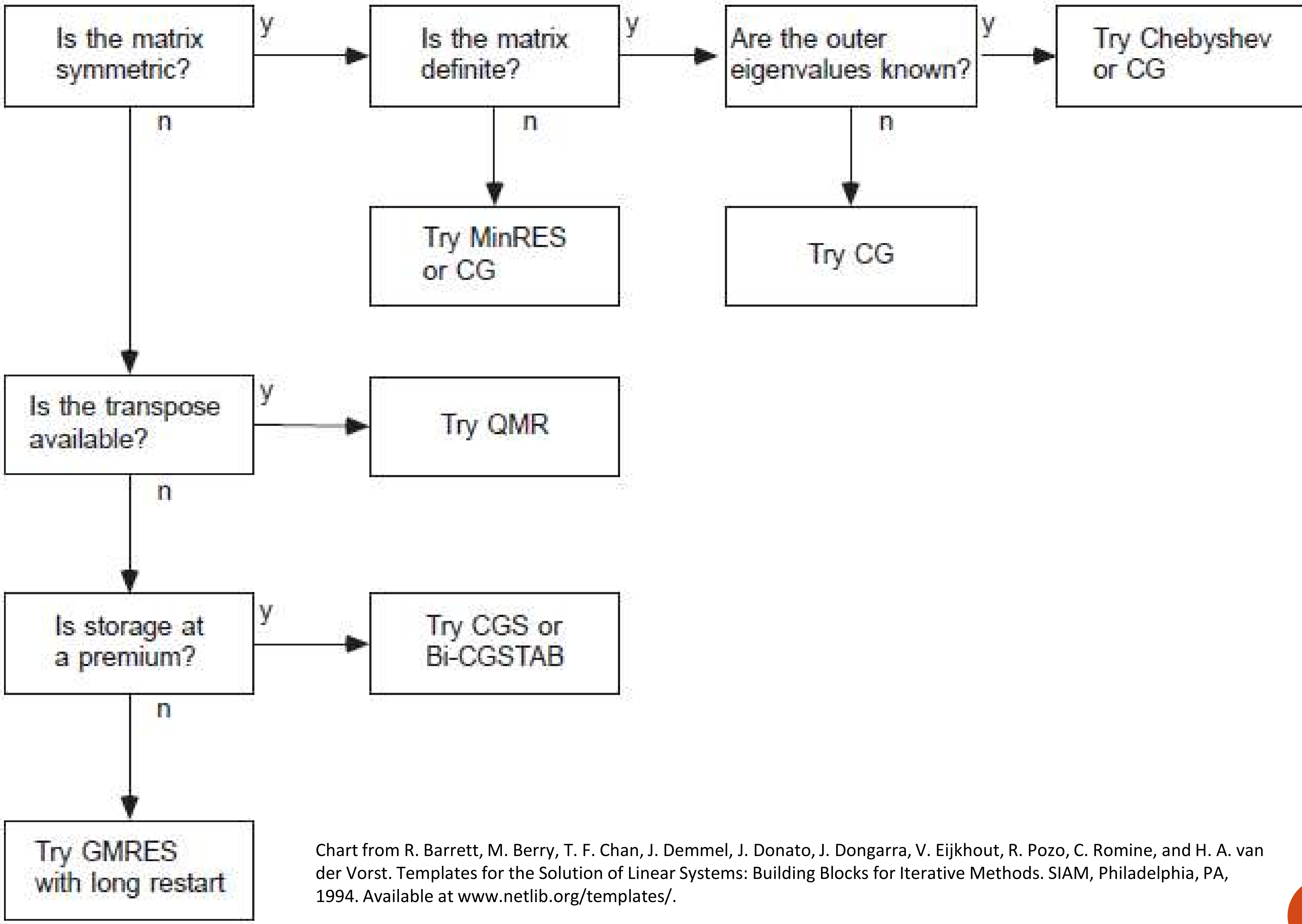


Chart from R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. A. van der Vorst. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM, Philadelphia, PA, 1994. Available at www.netlib.org/templates/.

Introduction to preconditioning

Preconditioning technique

Examples of preconditioners

Preconditioning technique and types of preconditioners

The idea of preconditioning is to apply an iterative solver to a system $\tilde{A}\tilde{x} = \tilde{b}$ equivalent to the initial system $Ax = b$, for which the solver would converge faster.

We have linear system $Ax = b$

1) **Left preconditioning**: multiply by M^{-1} on the left

$$M^{-1}Ax = M^{-1}b$$

$$M^{-1}A = \tilde{A},\ x = \tilde{x},\ M^{-1}b = \tilde{b}$$

Find $x = \tilde{x}$ from solving the system $\tilde{A}\tilde{x} = \tilde{b}$

Preconditioning technique and types of preconditioners

2) **Right preconditioning**: multiply by M^{-1} on the right

$AM^{-1}Mx = b$

$AM^{-1} = \tilde{A},\; Mx = \tilde{x},\; b = \tilde{b}$

2.1. Find $\tilde{x}$ from solving the system $\tilde{A}\tilde{x} = \tilde{b}$

2.2. Find x from solving the system $Mx = \tilde{x}$

3) **Two-sided preconditioning**: multiply by M_1^{-1} on the left and M_2^{-1} on the right

$M_1^{-1}AM_2^{-1}M_2x = M_1^{-1}b$

$M_1^{-1}AM_2^{-1} = \tilde{A},\; M_2x = \tilde{x},\; M_1^{-1}b = \tilde{b}$

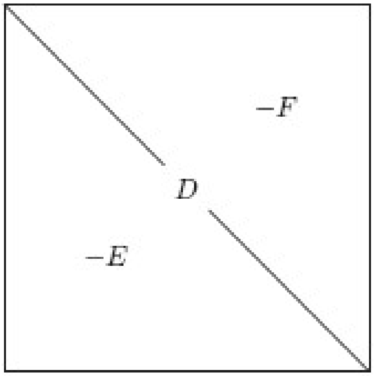
3.1. Find $\tilde{x}$ from solving the system $\tilde{A}\tilde{x} = \tilde{b}$

3.2. Find x from solving the system $M_2x = \tilde{x}$

Examples of preconditioners

Consider splitting of A into lower triangular matrix $-E$, diagonal matrix D and upper triangular matrix $-F$:

$A = D - E - F$



Jacobi method:

$M_J = D$

Gauss-Seidel method:

$M_{GS} = D - E$

SOR method:

$$M_{SOR} = \frac{1}{\omega} D - E$$

SSOR method:

$$M_{SSOR} = \frac{1}{\omega(2-\omega)}(D - \omega E)D^{-1}(D - \omega F)$$

Incomplete LU: $A = \widehat{L}\widehat{U} + R$

Neglect part of fill-ins $M_1 = \widehat{L},\ M_2 = \widehat{U}$

$\widehat{L}$ and $\widehat{U}$ have the same patterns as lower and upper triangular parts of A

Preconditioned methods

- Preconditioned conjugate gradient method (PCG)
- Split Preconditioned Conjugate Gradient method (Split PCG)
- Left preconditioned Generalized Minimal Residual method (LP GMRES)
- Right preconditioned Generalized Minimal Residual method (RP GMRES)

Deriving preconditioned method for symmetric positive definite matrix

When matrix A is symmetric positive definite, the preconditioner M should also be symmetric positive definite.

The first way to preserve symmetry is to use M-inner product instead of Eucledian inner product:

$$(x, y)_M \overset{def}{=} (Mx, y) = (x, M^T y) = (x, My) \Rightarrow (Mx, y) = (x, My)$$

Note that left and right preconditioning techniques are in general symmetric.

Here the symmetry of A and its preconditioner M is preserved, using the fact that $M^{-1}A$ is self-adjoint for the M-inner product:

$$(M^{-1}Ax, y)_M = (MM^{-1}Ax, y) = (Ax, y) = (x, A^T y) = (x, Ay) =$$
$$= (x, MM^{-1}Ay) = (M^T x, M^{-1}Ay) = (Mx, M^{-1}Ay) = (x, M^{-1}Ay)_M$$

Derivation of preconditioned conjugate gradient method (PCG) for symmetric positive definite matrix

Original residual: $r_j = b - Ax_j$

Residual for preconditioned system: $z_j = M^{-1} r_j$

- **CG without preconditioning**
- **Preconditioned CG**

$$\alpha_j = \frac{(r_j, r_j)}{(Ap_j, p_j)} \qquad \alpha_j = \frac{(z_j, z_j)_M}{(M^{-1}Ap_j, p_j)_M}$$

$$x_{j+1} = x_j + \alpha_j p_j$$

$$r_{j+1} = r_j - \alpha_j A p_j$$

$$\beta_j = \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)} \qquad \beta_j = \frac{(z_{j+1}, z_{j+1})_M}{(z_j, z_j)_M}$$

$$p_{j+1} = r_{j+1} + \beta_j p_j \qquad p_{j+1} = z_{j+1} + \beta_j p_j$$

Algorithm of preconditioned conjugate gradient method (PCG) for symmetric positive definite matrix

Since $(z_j, z_j)_M = (M^{-1}r_j, z_j)_M = (MM^{-1}r_j, z_j) = (r_j, z_j) \Rightarrow$

$(M^{-1}Ap_j, p_j)_M = (MM^{-1}Ap_j, p_j) = (Ap_j, p_j)$

Thus we can move back to Eucledian inner product.

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$, preconditioned residual $z_0 = M^{-1}r_0$; $p_0 := r_0$

2. Start iterative process for $j = 0, 1, 2, ...$ until convergence

2.1) $\alpha_j = \dfrac{(r_j, z_j)}{(Ap_j, p_j)}$

2.2) $x_{j+1} = x_j + \alpha_j p_j$

2.3) $r_{j+1} = r_j - \alpha_j Ap_j$

Steps 2.2 and 2.3 are the same as in CG

2.4) $z_{j+1} = M^{-1}r_{j+1}$

2.5) $\beta_j = \dfrac{(r_{j+1}, z_{j+1})}{(r_j, z_j)}$

2.6) $p_{j+1} = z_{j+1} + \beta_j p_j$

Derivation of Split Preconditioned Conjugate Gradient method (Split PCG) for symmetric positive definite matrix

The second way to preserve the symmetry of A and its preconditioner M is to use Cholessky factorization $M = LL^T$ and two-sided preconditioning.

For $Ax = b$ use the preconditioner $M = M_L M_R$,

where $M_L = L,\ M_R = L^T$

$$M_L^{-1} A M_R^{-1} M_R x = M_L^{-1} b$$

$$M_L^{-1} A M_R^{-1} = \tilde{A},\ M_R x = \tilde{x},\ M_L^{-1} b = \tilde{b} \ \Rightarrow\ L^{-1} A L^{-T} = \tilde{A},\ \ L^{-T} b = \tilde{b}$$

1. Find $\tilde{x}$ from solving the system $\tilde{A}\tilde{x} = \tilde{b}$

2. Find x from solving the system $L^T x = \tilde{x}$

Split Preconditioned Conjugate Gradient method (Split PCG) for symmetric positive definite matrix

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$, preconditioned initial residual $\hat{r}_0 = L^{-1} r_0$; $p_0 := L^{-T} \hat{r}_0$

2. Start iterative process for $j = 0, 1, 2, \ldots.$ until convergence

2.1) $\alpha_j = \dfrac{(\hat{r}_j, \hat{r}_j)}{(Ap_j, p_j)}$

2.2) $x_{j+1} = x_j + \alpha_j p_j$

2.3) $\hat{r}_{j+1} = \hat{r}_j - \alpha_j L^{-1} A p_j$

2.4) $z_{j+1} = M^{-1} r_{j+1}$

2.5) $\beta_j = \dfrac{(\hat{r}_{j+1}, \hat{r}_{j+1})}{(\hat{r}_j, \hat{r}_j)}$

2.6) $p_{j+1} = L^{-T} \hat{r}_{j+1} + \beta_j p_j$

In Split PCG only preconditioned residual is used

Left and right preconditioned GMRES

Left preconditioned Generalized Minimal Residual method (LP GMRES)

In LP GMRES algorithm is applied to the left preconditioned system

$M^{-1}Ax = M^{-1}b$

Initial residual is taken preconditioned: $z_0 = M^{-1}(b - Ax_0) = M^{-1}r_0$

Basis is constructed for the left preconditioned Krylov subspace

$K_m(M^{-1}A, z_0) = span\{z_0, M^{-1}Az_0, (M^{-1}A)^2 z_0, \ldots, (M^{-1}A)^{m-1} z_0\}$

Right preconditioned Generalized Minimal Residual method (RP GMRES)

In RP GMRES algorithm is applied to the right preconditioned system

$AM^{-1}Mx = b$

$AM^{-1} = \tilde{A},\ Mx = \tilde{x},\ b = \tilde{b} \Rightarrow \tilde{x} = Mx$ is preconditioned solution,

$x = M^{-1}\tilde{x}$ is original solution

Basis is constructed for the right preconditioned Krylov subspace

$K_m(AM^{-1}, r_0) = span\{r_0, AM^{-1}r_0, (AM^{-1})^2 r_0, \ldots, (AM^{-1})^{m-1} r_0\}$

Left preconditioned GMRES (LP GMRES)

1. Take initial guess x_0, calculate preconditioned initial residual $z_0 = M^{-1}(b - Ax_0)$,

calculate preconditioned initial residual norm $\beta = \|z_0\|_2$, and the first vector $v_1 = \dfrac{z_0}{\beta}$

2. Define $\overline{H}_m = \{h_{ij}\} \in \mathbb{C}^{(m+1)\times m}$, $V_m \in \mathbb{C}^{n\times m}$, $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$. Set $\overline{H}_m = 0$

3. Build the basis for Krylov subspace using Arnoldi process:

Loop for j from 1 to m.

3.1) $w_j = M^{-1}Av_j$

3.2) Loop for i from 1 to j $h_{ij} := (w_j, v_i)$, $w_j := w_j - h_{ij}v_i$

3.3) $h_{j+1,j} := \|w_j\|_2$. If $h_{j+1,j}$=0 $\Rightarrow$ $m := j$, go to step 4, else

3.4) $v_{j+1} := \dfrac{w_j}{h_{j+1,j}}$

4. Find y_m as a minimizer of the functional $J(y) = \left\|\beta e_1 - \overline{H}_m y\right\|_2$

(solve overdefined system $\overline{H}_m y = \beta e_1$)

5. Calculate preconditioned residual z_m, estimate residual norm: $\|z_m\|_2 < \varepsilon$.

6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

If satisfied, then stop. Else

7. Set $x_0 := x_m$ and go to step 1.

The residual can be calculated in the same way, as in GMRES, using Givens rotations

Right preconditioned GMRES (RP GMRES)

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$,

calculate residual norm $\beta = \|r_0\|_2$, and the first vector $v_1 = \dfrac{r_0}{\beta}$

2. Define $\overline{H}_m = \{h_{ij}\} \in \mathbb{C}^{(m+1)\times m}$, $V_m \in \mathbb{C}^{n\times m}$, $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$. Set $\overline{H}_m = 0$

3. Build the basis for Krylov subspace using Arnoldi process:

Loop for j from 1 to m.

3.1) $w_j = AM^{-1}v_j$

3.2) Loop for i from 1 to j $h_{ij} := (w_j, v_i)$, $w_j := w_j - h_{ij}v_i$

3.3) $h_{j+1,j} := \|w_j\|_2$. If $h_{j+1,j}=0 \Rightarrow m := j$, go to step 4, else

3.4) $v_{j+1} := \dfrac{w_j}{h_{j+1,j}}$

4. Find y_m as a minimizer of the functional $J(y) = \|\beta e_1 - \overline{H}_m y\|_2$

(solve overdefined system $\overline{H}_m y = \beta e_1$)

5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.

6. Calculate new approximate solution $x_m = x_0 + M^{-1}V_m y_m$

If satisfied, then stop. Else

7. Set $x_0 := x_m$ and go to step 1.

The residual can be calculated in the same way, as in GMRES, using Givens rotations