



*Metodi iterativi  
per sistemi lineari sparsi  
non simmetrici*

---

Zanello Francesca  
[zanello@dmsa.unipd.it](mailto:zanello@dmsa.unipd.it)

11 Dicembre 2008



# Introduzione

---

- **Problema:** risoluzione sistema lineare

$$Ax = b$$

- **GCP converge solo per matrici SDP: un motivo è che per matrici non SDP non è possibile avere al contempo:**

1. ortogonalità
2. proprietà di minimizzazione
3. short term recurrence

- **Problemi SDP sono molto comuni, ma per esempio la matrice del problema convezione-diffusione non è SDP**



# Estensioni

---

- **Equazioni normali: GCP applicato al sistema SPD**

$$A^T A x = A^T b$$

- **Metodi che forniscono **ortogonalità** + **minimizzazione** con una **long-term recurrence** (GMRES)**
- **Metodi che forniscono la (bi)**ortogonalizzazione** (BiCG, BiCGStab)**
- **Metodi che forniscono qualche proprietà di minimizzazione (QMR)**

# Metodi proiettivi

---

□ **IDEA:** estrarre una soluzione approssimata del problema  $Ax = b$  da un sottospazio  $\mathcal{K} \subset R^n$  di dimensione  $m$ .

□ Tale soluzione approssimata può essere calcolata imponendo  $m$  vincoli, tipicamente vengono imposte  $m$  **condizioni indipendenti di ortogonalità sul vettore residuo**



definire implicitamente un altro sottospazio  $\mathcal{L} \subset R^n$  generato dagli  $m$  vettori indipendenti utilizzati per l'ortogonalità

□ In generale nei metodi proiettivi si ricerca una soluzione approssimata in  $\mathcal{K}$  tale che il vettore residuo è ortogonale a  $\mathcal{L}$  (**condizioni di Petrov - Galerkin**)

# Metodi proiettivi

□ **Soluzione approssimata**  $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{y}_m$  Con  $\mathbf{y}_m \in \mathcal{K}$

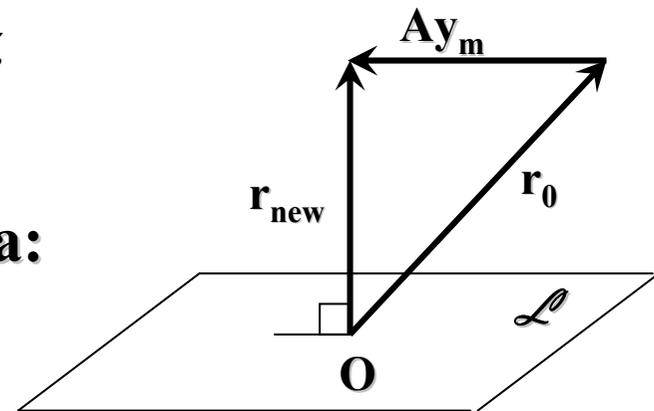
□ **Residuo  $m$ -simo**  $\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m = \mathbf{r}_0 - \mathbf{A}\mathbf{y}_m \perp \mathcal{L}_m$

□ **Se definisco:**  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ : base di  $\mathcal{K}$  e  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m$ : base di  $\mathcal{L}$

□ **Esprimo  $\mathbf{y}$  come:**  $\mathbf{y}_m = \sum_{i=1}^m z_i \mathbf{v}_i = \mathbf{V}_m \mathbf{z}$

□ **La relazione di ortogonalità diventa:**

$$\mathbf{w}_i^T (\mathbf{r}_0 - \mathbf{A}\mathbf{y}_m) = 0 \quad i=1, \dots, m$$



# Metodi proiettivi

---

- In forma matriciale:

$$W_m^T (r_0 - AV_m z) = 0$$

- Posto che  $W^T AV_m$  sia invertibile posso esprimere il vettore dei coefficienti come:

$$z = (W_m^T AV_m)^{-1} W_m^T r_0$$

- La soluzione approssimata sarà:

$$x_m = x_0 + V_m (W_m^T AV_m)^{-1} W_m^T r_0$$



# Metodi proiettivi sugli spazi di Krylov

---

- Una delle scelte più adottate per  $\mathcal{K}$  sono gli spazi di Krylov di dimensione  $m$  associato ad  $A$

$$\mathcal{K}_m(v_1) = \text{span}\{v_1, Av_1, A^2v_1, A^3v_1, \dots, A^{m-1}v_1\}$$

- Possono essere fatte diverse scelte per  $\mathcal{L}$ . uno degli approcci più validi è basato sull'utilizzo del sottospazio di Krylov associato a  $A^T$

$$\mathcal{L}_m(w_1) = \text{span}\{w_1, A^T w_1, (A^T)^2 w_1, (A^T)^3 w_1, \dots, (A^T)^{m-1} w_1\}$$

- Un metodo proiettivo è detto **ortogonale** se  $\mathcal{K} = \mathcal{L}$ , **obliquo** altrimenti
- Tutti i metodi proiettivi sono in origine metodi di ortogonalizzazione del residuo costruito nello spazio di Krylov  $\mathcal{K}_m$  per rispetto allo spazio ausiliario  $\mathcal{L}_m$ . Alcuni sono anche metodi di minimo del residuo  $r_m$  in una norma opportuna



# GMRES

---

- Determina la soluzione del sistema lineare

$$Ax = b$$

minimizzando la norma del residuo

$$r_m = b - Ax_m$$

su tutti i vettori  $x_m$  scritti come

$$x_0 + K_m = x_0 + \sum_{i=1}^m z_i v_i$$

dove  $x_0$  è un vettore arbitrario iniziale e  $\mathcal{K}_m$  è il sottospazio di Krylov generato dal residuo iniziale

$$K_m(r_0) = \text{span}\{r_0, Ar_0, A^2r_0, A^3r_0, \dots, A^{m-1}r_0\}$$

# GMRES

---

1. Ortogonalizzazione della base
2. Minimizzazione della norma del residuo

$$\beta = \|\mathbf{r}_0\|, \mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$$

DO  $k = 1, m$

1.  $\mathbf{w}_{k+1} = A\mathbf{v}_k$

2. DO  $j = 1, k - 1$

3.  $h_{jk} = \mathbf{w}_{k+1}^T \mathbf{v}_j$

4.  $\mathbf{w}_{k+1} := \mathbf{w}_{k+1} - h_{jk} \mathbf{v}_j$

5. END DO

6.  $h_{k+1,k} = \|\mathbf{w}_{k+1}\|; \mathbf{v}_{k+1} = \mathbf{w}_{k+1}/h_{k+1,k}$

END DO

**Gram Schmidt**

- 
- I vettori generati con questo procedimento verificano la seguente relazione

$$A\mathbf{v}_k = \sum_{j=1}^{k+1} h_{jk} \mathbf{v}_j$$

che può essere riscritta in forma matriciale:

$$AV_m = V_{m+1}H_m$$

$A$ [n x n]	•	$V_m$ [n x m]	=	$V_{m+1}$ [n x (m+1)]	•	$H_m$ [(m+1) x m]
----------------	---	------------------	---	--------------------------	---	----------------------

- La matrice H è una matrice di Hessenberg ovvero ha la struttura seguente:

$$\begin{bmatrix}
 h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} & \cdots & h_{1,m-2} & h_{1,m-1} & h_{1,m} \\
 h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} & \cdots & h_{2,m-2} & h_{2,m-1} & h_{2,m} \\
 \mathbf{0} & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} & \cdots & h_{3,m-2} & h_{3,m-1} & h_{3,m} \\
 \mathbf{0} & \mathbf{0} & h_{4,3} & h_{4,4} & h_{4,5} & \cdots & h_{4,m-2} & h_{4,m-1} & h_{4,m} \\
 \mathbf{0} & \mathbf{0} & \mathbf{0} & h_{5,4} & h_{5,5} & \cdots & h_{5,m-2} & h_{5,m-1} & h_{5,m} \\
 & & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
 \vdots & & & & \mathbf{0} & h_{m-4,m-1} & h_{m-3,m-1} & h_{m-2,m-1} & h_{m-2,m} \\
 & & & & & \mathbf{0} & h_{m-1,m-2} & h_{m-1,m-1} & h_{m-1,m} \\
 & & & & & & \mathbf{0} & h_{m,m-1} & h_{m,m} \\
 \mathbf{0} & & & \cdots & & & & \mathbf{0} & h_{m+1,m}
 \end{bmatrix}$$

# Minimizzazione della norma del residuo

---

$$\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m = \mathbf{V}_{m+1} [\beta \mathbf{e}_1 - \mathbf{H}_m \mathbf{z}]$$

con  $\beta = \|\mathbf{v}_1\|$  e  $\mathbf{e}_1 = [1; 0; \dots; 0]^T$

$$\|\mathbf{r}_m\| = \sqrt{\mathbf{r}_m^T \mathbf{r}_m} \quad \longrightarrow \quad \|\mathbf{r}_m\| = \|\beta \mathbf{e}_1 - \mathbf{H}_m \mathbf{z}\|$$

## FATTORIZZAZIONE QR:

Qualunque matrice  $\mathbf{H}$  può essere fattorizzata nel prodotto di una matrice ortogonale  $\mathbf{Q}$  per una matrice triangolare alta  $\mathbf{R}$

$$\begin{array}{|c|} \hline \mathbf{H}_m \\ \hline [(m+1) \times m] \\ \hline \end{array} = \begin{array}{|c|} \hline \mathbf{Q} \\ \hline [(m+1) \times (m+1)] \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \mathbf{R} \\ \hline [(m+1) \times m] \\ \hline 0 \ 0 \ \dots \ 0 \ 0 \\ \hline \end{array}$$

# Minimizzazione della norma del residuo

## FATTORIZZAZIONE QR:

Qualunque matrice  $H$  può essere fattorizzata nel prodotto di una matrice ortogonale  $Q$  per una matrice triangolare alta  $R$

$$\begin{array}{|c|} \hline H_m \\ \hline [(m+1) \times m] \\ \hline \end{array} = \begin{array}{|c|} \hline Q \\ \hline [(m+1) \times (m+1)] \\ \hline \end{array} \cdot \begin{array}{|c|} \hline R \\ \hline [(m+1) \times m] \\ 0 \ 0 \ \dots \ 0 \ 0 \\ \hline \end{array}$$

$$\min \| \beta e_1 - H_m z \| = \min \| \beta e_1 - QRz \| = \min \| g - Rz \|^2$$

$$\text{con } g = \beta Q^T e_1$$

# Minimizzazione della norma del residuo

---

- La matrice  $R$  assume la forma

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,m-1} & r_{1,m} \\ \mathbf{0} & r_{2,2} & \dots & r_{2,m-1} & r_{2,m} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & r_{m-1,m-1} & r_{m-1,m} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & r_{m,m} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}$$

- La soluzione a  $\min \|g - Rz\|$  si ottiene risolvendo  $\tilde{R}z = \tilde{g}$  dove  $\tilde{R}$  è ottenuta da  $R$  eliminando l'ultima riga e  $\tilde{g}$  sono le prime  $m$  componenti di  $g$



# Minimizzazione della norma del residuo

---

- Il sistema  $\tilde{R}z = \tilde{g}$  è quadrato di piccole dimensioni e la matrice del sistema è triangolare alta: quindi è di semplice risoluzione utilizzando il metodo della sostituzione all'indietro
- Infine si noti che  $J(z) = \|b - Ax_m\|$  assume come valore minimo  $|g_{m+1}|$

# Algoritmo

ALGORITHM: GMRES

Input:  $x_0, A, b, k_{\max}, \text{toll}$

■  $r_0 = b - Ax_0, k = 0, \rho_0 = \|r_0\|, \beta = \rho_0, v_1 = \frac{r_0}{\beta}$

■ WHILE  $\rho_k > \text{toll} \|b\|$  AND  $k < k_{\max}$  DO

1.  $k = k + 1$

2.  $v_{k+1} = Av_k$

3. DO  $j = 1, k$

$$h_{jk} = v_{k+1}^T v_j$$

$$v_{k+1} = v_{k+1} - h_{jk} v_j$$

END DO

4.  $h_{k+1,k} = \|v_{k+1}\|$

5.  $v_{k+1} = v_{k+1}/h_{k+1,k}$

6.  $z_k = \text{argmin} \|\beta e_1 - H_k z\|$

7.  $\rho_k = \|\beta e_1 - H_k z_k\|$

■ END WHILE

■  $x_k = x_0 + V_k z_k$

ortogonalizzazione



# Osservazioni finali

---

- GMRES è ottimo nel senso che minimizza il residuo su un sottospazio di dimensione crescente (**proprietà di terminazione finita**)
- GMRES è un metodo MOLTO costoso:
  1. **Memorizzazione:** è necessario mantenere in memoria tutti i vettori della base di Krylov e tutta la matrice  $H_m$  (**long-term recurrence**)
  2. **Costo computazionale:** il costo della ortogonalizzazione di Gram-Schmidt cresce all'aumentare del numero di iterazioni
- Nelle applicazioni si definisce a priori un numero massimo  $p$  di vettori che possono essere memorizzati. Dopo  $p$  iterazioni si calcola l'approssimazione  $x_p$  si cancellano tutti i vettori dello spazio di Krylov e si riparte con un nuovo spazio di Krylov avente come vettore generatore  $r_p = b - Ax_p$
- In questo modo però si perde la proprietà di ottimo con conseguente rallentamento del metodo
- **Lucky breakdown**

# Convergenza

---

- Supponiamo per semplicità che

$$A = V\Lambda V^{-1}$$

dove  $\Lambda$  è la matrice diagonale degli autovalori e le colonne di  $V$  sono i vettori normalizzati di  $A$ . Allora

$$\|r_k\| = \min_{P_k} \|V p_k(\Lambda) V^{-1}\| \cdot \|r_0\| \leq \kappa(V) \min_{P_k} \|p_k(\Lambda)\| \cdot \|r_0\|$$

oppure

$$\frac{\|r_k\|}{\|r_0\|} \leq \kappa(V) \min_{P_k} \max |\rho_k(\lambda_i)|$$

- La matrice  $V$  può essere mal condizionata
- Non è possibile porre in relazione la convergenza del GMRES con la distribuzione degli autovalori di  $A$  in modo semplice

# Precondizionamento

---

- Il precondizionamento è fondamentale per il successo dei metodi proiettivi basati sugli spazi di Krylov in applicazioni di fluidodinamica
- E' possibile usufruire di 3 diverse modalità di precondizionamento:

**1. Left-preconditioning:**  $M^{-1}Ax = M^{-1}b$

$$K_m = \text{span} \left\{ r_0, M^{-1}Ar_0, (M^{-1}A)^2 r_0, \dots, (M^{-1}A)^{m-1} r_0 \right\}$$

residuo precondizionato:  $y_m = M^{-1}(b - Ax_m) \neq r_m$

**2. Right-preconditioning:**  $AM^{-1}u = b$        $u = Mx$

$$K_m = \text{span} \left\{ r_0, AM^{-1}r_0, (AM^{-1})^2 r_0, \dots, (AM^{-1})^{m-1} r_0 \right\}$$

residuo precondizionato:  $y_m = (b - Ax_m) = r_m$

**3. Split-preconditioning:**  $L^{-1}AU^{-1}u = L^{-1}b$        $x = U^{-1}u$



# Precondizionamento

---

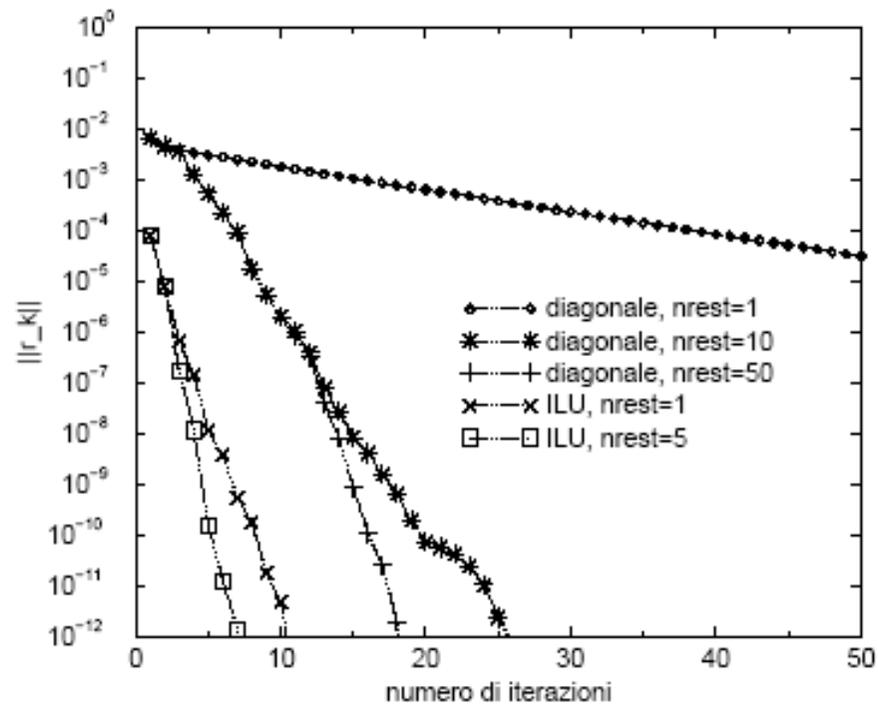
- ❑ Il precondizionamento più che ridurre il numero di condizionamento di  $A$  serve a raggruppare gli autovalori lontano dall'origine
- ❑ Precondizionatori più comuni:
  1. Diagonale:  $M = \text{diag}(\|A^i\|)$  con  $A^i$  la  $i$ -sima riga di  $A$
  2. Decomposizione ILU
  3. Approssimate dell'inversa
- ❑ Costo computazionale: invece di applicare  $A$  è necessario applicare a un vettore (step 2) come

$$t = Av_k$$

$$Mv_{k+1} = t$$

# Precondizionamento

- **Profilo di convergenza del GMRES applicato ad una matrice molto piccola ( $n=27$ ) con due diversi preconditionatori e diverse scelte per il parametro di restart  $n_{rest}$**



# Metodi proiettivi non ortogonali per sistemi non simmetrici

- Non consentono una previsione teorica a priori dell'errore
- Sono basati su uno schema con costo computazionale per singola iterazione fissato (short term recurrence)
- La soluzione approssimata:

$$x_m = x_0 + V_m z$$

con:

$$z = (W_m^T A V_m)^{-1} W_m^T r_0$$

- **IDEA:** scegliere  $V_m$  e  $W_m$  in modo che  $W_m^T A V_m$  sia tridiagonale e quindi il sistema

$$W_m^T A V_m z = W_m^T r_0$$

sia risolvibile agevolmente mediante l'algoritmo di Thomas

# Metodo di Lanczos

---

$$\mathcal{K}_m(v_1) = \text{span}\{v_1, Av_1, A^2v_1, A^3v_1, \dots, A^{m-1}v_1\}$$

$$\mathcal{L}_m(w_1) = \text{span}\{w_1, A^T w_1, (A^T)^2 w_1, (A^T)^3 w_1, \dots, (A^T)^{m-1} w_1\}$$

- Generiamo due basi per  $\mathcal{K}$  e  $\mathcal{L}$  tali che

$$w_i^T v_j = \delta_{ij}$$

Due successioni che soddisfano tale relazione sono dette ***biortogonali***

- Le basi per un sottospazio di Krylov associato rispettivamente ad  $A$  e ad  $A^T$  possono essere calcolate mediante

$$\tilde{v}_{k+1} = Av_k - \sum_{j=1}^k h_{jk} v_j$$

$$\tilde{w}_{k+1} = A^T w_k - \sum_{j=1}^k \tilde{h}_{jk} w_j$$

# Metodo di Lanczos

$$\left. \begin{array}{l} \mathbf{v}_j^T \tilde{\mathbf{w}}_{k+1} = 0 \quad \text{con } j < k+1 \\ \mathbf{w}_j^T \tilde{\mathbf{v}}_{k+1} = 0 \quad \text{con } j < k+1 \end{array} \right\} \begin{array}{l} \mathbf{h}_{jk} = \mathbf{w}_j^T \mathbf{A} \mathbf{v}_k \\ \tilde{\mathbf{h}}_{jk} = \mathbf{v}_j^T \mathbf{A}^T \mathbf{w}_k \end{array} \xrightarrow{\mathbf{v}_j^T \mathbf{A}^T \mathbf{w}_k = \mathbf{w}_k^T \mathbf{A} \mathbf{v}_j} \boxed{\mathbf{h}_{jk} = \tilde{\mathbf{h}}_{kj}}$$

$$\left. \begin{array}{l} \boxed{H = \tilde{H}^T} \\ \text{Due matrici di Hessenberg sono uguali} \\ \text{solo se sono tridiagonali} \end{array} \right\} \xrightarrow{} \boxed{\mathbf{h}_{jk} = \tilde{\mathbf{h}}_{kj} = 0 \quad \forall j < k - 1}$$

$$\left. \begin{array}{l} \alpha_k = \mathbf{h}_{k,k} = \tilde{\mathbf{h}}_{k,k} \\ \beta_k = \mathbf{h}_{k-1,k} \\ \delta_k = \tilde{\mathbf{h}}_{k-1,k} \end{array} \right\} \xrightarrow{} \begin{array}{l} \tilde{\mathbf{v}}_{k+1} = \mathbf{A} \mathbf{v}_k - \alpha_k \mathbf{v}_k - \beta_k \mathbf{v}_{k-1} \\ \tilde{\mathbf{w}}_{k+1} = \mathbf{A}^T \mathbf{w}_k - \alpha_k \mathbf{w}_k - \delta_k \mathbf{w}_{k-1} \end{array} \quad \text{Short term recurrence}$$

Affinché  $\mathbf{w}_{k+1}^T \mathbf{v}_{k+1} = 1$  scaliamo opportunamente

$$\mathbf{v}_{k+1} = \frac{\tilde{\mathbf{v}}_{k+1}}{\gamma_v} = \frac{\tilde{\mathbf{v}}_{k+1}}{\delta_{k+1}} \qquad \mathbf{w}_{k+1} = \frac{\tilde{\mathbf{w}}_{k+1}}{\gamma_w} = \frac{\tilde{\mathbf{w}}_{k+1}}{\beta_{k+1}}$$



# Metodo di Lanczos

---

1. Choose  $v_1$  and  $w_1$  such that  $w_1^T v_1 = 1$
  2. Set  $\beta_1 = \delta_1 = 0$  and  $v_0 = w_0 = 0$
  3. Do  $k = 1, \dots, m$
  4.  $\alpha_k = w_k^T A v_k$
  5.  $\hat{v}_{k+1} = A v_k - \alpha_k v_k - \beta_k v_{k-1}$
  6.  $\hat{w}_{k+1} = A^T w_k - \alpha_k w_k - \delta_k w_{k-1}$
  7. Choose  $\beta_{k+1}$  and  $\delta_{k+1}$  such that  $\beta_{k+1} \delta_{k+1} = \hat{w}_{k+1}^T \hat{v}_{k+1}$
  8.  $v_{k+1} = \hat{v}_{k+1} / \delta_{k+1}$
  9.  $w_{k+1} = \hat{w}_{k+1} / \beta_{k+1}$
  10. END DO
- 

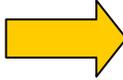
$$\delta_{k+1} = \left| \tilde{w}_{k+1}^T \tilde{v}_{k+1} \right|^{1/2}$$
$$\beta_{k+1} = \frac{\tilde{w}_{k+1}^T \tilde{v}_{k+1}}{\delta_{k+1}}$$

# Metodo di Lanczos

- L'aspetto più interessante è che consente di aumentare la dimensione delle basi di  $\mathcal{K}$  e  $\mathcal{L}$  usando solo gli ultimi 2 vettori della base;
- Vengono memorizzati solo  $\alpha_k$ ,  $\beta_k$  e  $\delta_k$  e invece di tutti gli  $h_{jk}$  come avveniva nel GMRES ovvero vengono memorizzati solo 3 vettori invece di una matrice di Hessenberg;
- ci sono più probabilità di **Breakdown** rispetto al GMRES: l'algoritmo si blocca quando  $\tilde{w}_{k+1}^T \tilde{v}_{k+1} = 0$  ovvero se:

✓  $\tilde{v}_{k+1} = 0$       **Lucky breakdown**       $x_m = x$

✓  $\tilde{w}_{k+1} = 0$        $x_m^* = x^*$

✓  $\tilde{w}_{k+1}^T \tilde{v}_{k+1} = 0 \quad \wedge \quad \tilde{v}_{k+1} \neq 0 \quad \tilde{w}_{k+1} \neq 0$  **Serious breakdown**  algoritmi  
Look-ahead  
Lanczos





# Algoritmo Two-sided Lanczos

---

- 
1. Compute  $r_0 = b - Ax_0$
  2.  $\beta = \|r_0\|_2$
  3. Set  $v_1 = r_0/\beta$  and  $w_1 = v_1$
  4. DO  $m = 1, \dots$  until convergence
  5.       Compute  $v_1, v_2, \dots, v_m$  and  $w_1, w_2, \dots, w_m$  with Algorithm 2
  6.        $z = \beta T_m^{-1} e_1$
  7.        $x_m = x_0 + V_m z$
  8. END DO
- 

- **L'adozione dell'algoritmo two-sided di Lanczos per costruire le basi biortogonali per  $\mathcal{K}$  e  $\mathcal{L}$  in un metodo proiettivo porta alla soluzione di un sistema tridiagonale di dimensione crescente ad ogni step dell'algoritmo**

# Il metodo del gradiente biconiugato (BiCG)

□ **E' basato sul processo two-sided di Lanczos:**

1. Dato un vettore arbitrario  $\hat{r}_0$  non ortogonale a  $r_0$ , costruisce 2 sets di vettori  $v_1, \dots, v_n \in \mathcal{K}(A, r_0)$  e  $w_1, \dots, w_n \in \mathcal{L}(A^T, \hat{r}_0)$  tali che

$$w_i^T v_j = \delta_{ij}$$

2. In forma matriciale:

$$AV_k = V_{k+1}T_{k+1,k} \quad A^T W_k = W_{k+1}\hat{T}_{k+1,k} \quad V_k^T W_k = I_k$$

□ **La relazione per la soluzione approssimata è**

$$x_m = x_0 + v_k z_k$$

□ **Tra le diverse scelte che si possono effettuare per il vettore  $z_k$ , BiCG impone**

$$r_k = r_0 - AV_k z_k \perp W_k$$

$$W_k^T r_0 - W_k^T AV_k z_k = 0$$

# Il metodo del gradiente biconiugato (BiCG)

$$\left. \begin{aligned} W_k^T A V_k &= T_k \\ W_k^T r_0 &= \|r_0\| e_1 = \beta e_1 \end{aligned} \right\} \longrightarrow T_k z_k = \beta e_1 \longrightarrow x_m = x_0 + \beta V_m T_m^{-1} e_1$$

- **BiCG utilizza la fattorizzazione LDU per  $T_k$  in modo da rendere l'algoritmo two sided di Lanczos ancora più efficiente dal punto di vista computazionale**

$$\left. \begin{aligned} T_m^{-1} &= U_m^{-1} L_m^{-1} \\ P_m &= V_m U_m^{-1} \\ a_m &= L_m^{-1} \beta e_1 \end{aligned} \right\} \longrightarrow \boxed{x_m = x_0 + P_m a_m}$$



# Il metodo del gradiente biconiugato (BiCG)

---

- Visto il ruolo duale delle basi  $v_i$  e  $w_i$  è possibile definire il sistema duale  $A^T x^* = b^*$  e il corrispondente residuo tale che  $r_i^* \parallel w_i$  e posto  $P_m^* = W_m L_m^{-T}$ , si può dimostrare che i vettori  $p_i^*$  e  $p_i^*$  sono mutuamente A-ortogonali
- Per tale motivo il metodo illustrato è detto **gradiente biconiugato (Bi-CG)**

# Algoritmo BiCG

---

1. Compute  $r_0 = b - Ax_0$  and choose  $r_0^*$  such that  $r_0^{*T} r_0^* \neq 0$
2. Set  $p_0 = r_0$  and  $p_0^* = r_0^*$
3. Do  $k = 0, \dots$  until convergence
4.  $\alpha_k = r_k^{*T} r_k / p_k^{*T} A p_k$
5.  $x_{k+1} = x_k + \alpha_k p_k$
6.  $r_{k+1} = r_k - \alpha_k A p_k$
7.  $r_{k+1}^* = r_k^* - \alpha_k A^T p_k^*$
8.  $\beta_k = r_{k+1}^{*T} r_{k+1} / r_k^{*T} r_k$
9.  $p_k = r_{k+1} + \beta_k p_k$
10.  $p_k^* = r_{k+1}^* + \beta_k p_k^*$
11. END DO

- **Ad ogni step è richiesta l'implementazione del prodotto per  $A^T$**
- **La convergenza è tipicamente oscillante con ampie variazioni della norma del residuo**

# Algoritmo BiCG stab

□ Il motivo è legato a come sono calcolati  $\alpha_k$  e  $\beta_k$ :

$$\rho_k = r_k^{*T} r_k = r_0^{*T} \phi_k^2(A) r_0 \quad \longrightarrow \quad \tilde{\rho}_k = r_0^{*T} \psi_k(A) \phi_k(A) r_0 = r_0^{*T} r_k$$

1. Compute  $r_0 = b - Ax_0$  and choose  $r_0^*$  arbitrary
2. Set  $p_0 = r_0$
3. Do  $k = 0, \dots$  until convergence
4.  $\alpha_k = \tilde{\rho}_k / r_0^{*T} A p_k$
5.  $s_k = r_k - \alpha_k A p_k$
6.  $\omega_k = s_k^T A s_k / (A s_k)^T A s_k$
7.  $x_{k+1} = x_k + \alpha_k p_k + \omega_k s_k$
8.  $r_{k+1} = s_k - \omega_k A s_k$
9.  $\tilde{\rho}_{k+1} = r_0^{*T} r_{k+1}$
10.  $\beta_k = \tilde{\rho}_{k+1} \alpha_k / \tilde{\rho}_k \omega_k$
11.  $p_{k+1} = r_{k+1} + \beta_k p_k - \omega_k \beta_k A p_k$
12. END DO

**1. La convergenza è stabilizzata**

**2. Non è necessario il prodotto per  $A^T$**