

Risoluzione di sistemi lineari con metodi iterativi.

Calcolo Numerico a.a. 2021-22

Elena Loli Piccolomini

Metodi iterativi

- I **metodi iterativi**. A partire da uno o più dati iniziali, calcolano dei valori x_k attraverso un procedimento che si ripete (itera) sempre uguale ad ogni passo k :

$$x_k = G(x_{k-1})$$

- Sotto opportune condizioni gli iterati x_k convergono alla soluzione x^* (tale che $x^* = A^{-1}b$) per $k \rightarrow \infty$.

Generalità

Dato il sistema lineare $\mathbf{Ax} = \mathbf{b}$ i metodi iterativi ricercano la soluzione mediante una opportuna successione \mathbf{x}_{k+1} che può avere una delle seguenti forme:

$$\mathbf{x}_{k+1} = \mathbf{H}\mathbf{x}_k + \mathbf{d} \quad \left\{ \begin{array}{l} \text{Metodo di Jacobi} \\ \text{Metodo di Gauss Seidel} \\ \text{Metodi di Rilassamento (SOR, SSOR)} \end{array} \right.$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad \left\{ \begin{array}{l} \text{Metodo Gradienti Coniugati} \\ \text{Metodo GMRES} \\ \text{Metodi di Krylov} \end{array} \right.$$

dove $\mathbf{H} \in \mathbb{R}^{n \times n}$, $\mathbf{d}, \mathbf{p}_k \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$.

Metodi iterativi

Schema algoritmo iterativo:

1. Dati: x_0
2. $k=1$
3. Ripeti finchè convergenza
 - 3.1 $x_k = G(x_{k-1})$
 - 3.2 $k = k + 1$

end

Sono da specificare nel singolo metodo le condizioni di convergenza che comunque contengono sempre la seguente:

$$k \leq \text{maxit}$$

Metodi iterativi

Convergenza metodi iterativi.

Si dice che la successione x_k generata da un metodo iterativo converge ad α con ordine $p \geq 1$ se:

$$\exists C > 0 : \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} \leq C, \forall k \geq k_0$$

dove k_0 è un intero opportuno. In tal caso si dirà che il **metodo è di ordine p** .

Osservazione. nel caso $p = 1$ per avere convergenza deve essere $C < 1$. In questo caso C prende il nome di *fattore di convergenza*.

Introduzione ai metodi iterativi per sistemi lineari

- Consentono di mantenere la struttura della matrice
- Si applicano a matrici di grandi dimensioni e sparse, quali quelle che si incontrano nella risoluzione di equazioni differenziali con condizioni ai limiti, risolte con metodi alle differenze finite o agli elementi finiti.
- La complessità computazionale è di un prodotto matrice vettore per iterazione.

Introduzione ai metodi iterativi per sistemi lineari

- Consentono di mantenere la struttura della matrice
- Si applicano a matrici di grandi dimensioni e sparse, quali quelle che si incontrano nella risoluzione di equazioni differenziali con condizioni ai limiti, risolte con metodi alle differenze finite o agli elementi finiti.
- La complessità computazionale è di un prodotto matrice vettore per iterazione.

Introduzione ai metodi iterativi per sistemi lineari

- Consentono di mantenere la struttura della matrice
- Si applicano a matrici di grandi dimensioni e sparse, quali quelle che si incontrano nella risoluzione di equazioni differenziali con condizioni ai limiti, risolte con metodi alle differenze finite o agli elementi finiti.
- La complessità computazionale è di un prodotto matrice vettore per iterazione.

Metodi iterativi per sistemi lineari

Dato il sistema lineare $\mathbf{Ax} = \mathbf{b}$ i metodi iterativi ricercano la soluzione mediante una opportuna successione \mathbf{x}_{k+1} che può avere una delle seguenti forme:

$$\mathbf{x}_{k+1} = \mathbf{H}\mathbf{x}_k + \mathbf{d} \quad \left\{ \begin{array}{l} \text{Metodo di Jacobi} \\ \text{Metodo di Gauss Seidel} \\ \text{Metodi di Rilassamento (SOR, SSOR)} \end{array} \right.$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad \left\{ \begin{array}{l} \text{Metodo Gradienti Coniugati} \\ \text{Metodo GMRES} \\ \text{Metodi di Krylov} \end{array} \right.$$

dove $\mathbf{H} \in \mathbb{C}^{n \times n}$, $\mathbf{d}, \mathbf{p}_k \in \mathbb{C}^n$, $\alpha \in \mathbb{R}$.

Introduzione ai metodi iterativi

$A \in \mathbb{R}^{n \times n}$ una matrice non singolare,

$$A = M - N$$

dove M è una matrice non singolare e caratterizzata dal fatto che $Mz = h$ sia “facilmente” risolubile.

Allora il sistema

$$Ax = b \tag{1}$$

si può scrivere come

$$Mx - Nx = b$$

ovvero

$$x = Tx + c \tag{2}$$

dove $T = M^{-1}N$ e $c = M^{-1}b$.

Il sistema (5) è equivalente al (2).

Introduzione ai metodi iterativi

Assegnato un vettore iniziale x_0 si considera la successione x_1, x_2, \dots definita da

$$x_k = Tx_{k-1} + c \quad k = 1, 2, \dots \quad (3)$$

Se

- la successione $\{x_k\}$ è convergente e
- $x^* = \lim_{k \rightarrow \infty} x_k$

passando al limite

$$x^* = Tx^* + c. \quad (4)$$

Quindi x^* è la soluzione del sistema.

La relazione (3) individua un **metodo iterativo** per determinare la soluzione x^* di (5); la matrice T si chiama *matrice di iterazione* del metodo.

Introduzione ai metodi iterativi

Al variare del vettore iniziale x_0 si ottengono da (3) diverse successioni $\{x_k\}$, alcune delle quali sono convergenti ed altre no.

Un metodo iterativo è detto **convergente** se, qualunque sia il vettore iniziale x_0 , la successione $\{x_k\}$ è convergente.

Esempio.

$$T = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 2 \end{pmatrix} \quad c = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

per cui $x^* = (0, 0, 0)^T$, si ha

$$T^k = \begin{pmatrix} \frac{1}{2^k} & 0 & 0 \\ 0 & \frac{1}{2^k} & 0 \\ 0 & 0 & 2^k \end{pmatrix}.$$

Metodi iterativi

Se $x_0 = (1, 0, 0)^T$ si ottiene la successione

$$x_k = (1/2^k, 0, 0) \quad k = 1, 2, \dots$$

che converge alla soluzione x^* del sistema. Se invece si pone $x_0 = (0, 1, 1)^T$ si ottiene la successione

$$x_k = (0, 1/2^k, 2^k) \quad k = 1, 2, \dots$$

che non converge a x^* .

Questo è un esempio di metodo **non convergente**.

Teorema

Il metodo iterativo è convergente se e solo se $\rho(T) < 1$.

Velocità convergenza metodi iterativi

Fissata una norma di vettori $\|\cdot\|$ e la corrispondente norma di matrici indotta

$$\|e_k\| \leq \|T^k\| \|e_0\|$$

Esempio Siano

$$T = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.6 \end{pmatrix} \quad S = \begin{pmatrix} 0.5 & 0.25 \\ 0 & 0.5 \end{pmatrix}.$$

Si ha

$$T^k = \begin{pmatrix} 0.5^k & 0 \\ 0 & 0.6^k \end{pmatrix} \quad S^k = \begin{pmatrix} 0.5^k & k0.5^{k+1} \\ 0 & 0.5^k \end{pmatrix}.$$

Utilizzando la norma infinito risulta

$$\|T^k\|_\infty = 0.6^k \quad \|S^k\|_\infty = (2+k)0.5^{k+1}.$$

- per $k \leq 9$ si ha che $\|T^k\|_\infty < \|S^k\|_\infty$ e per $k \geq 10$ si ha che $\|T^k\|_\infty > \|S^k\|_\infty$

Costruzione dei metodi iterativi

$$A = D - E - F$$

dove $D = \text{diag}\{a_{11}, a_{22}, \dots, a_{nn}\}$, $-E$ è la parte strettamente triangolare inferiore di A e $-F$ è la parte strettamente triangolare superiore.

Si definisce il procedimento iterativo

$$Mx_k = Nx_{k-1} + b$$

Metodo di Jacobi

metodo di Jacobi (o delle sostituzioni simultanee):

$$M = D \quad N = E + F.$$

La matrice di iterazione \mathcal{J} del metodo di Jacobi è

$$\mathcal{J} = D^{-1}(E + F) = I - D^{-1}A.$$

Il metodo di Jacobi è definito se D è non singolare ($a_{ii} \neq 0$ per ogni $i = 1, \dots, n$).

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) / a_{ii}, \quad i = 1, \dots, n$$

dove il superindice k indica il numero di iterazione e $x^{(0)}$ è un vettore arbitrario.

Metodo di Jacobi

metodo di Jacobi (o delle sostituzioni simultanee):

$$M = D \quad N = E + F.$$

La matrice di iterazione \mathcal{J} del metodo di Jacobi è

$$\mathcal{J} = D^{-1}(E + F) = I - D^{-1}A.$$

Il metodo di Jacobi è definito se D è non singolare ($a_{ii} \neq 0$ per ogni $i = 1, \dots, n$).

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) / a_{ii}, \quad i = 1, \dots, n$$

dove il superindice k indica il numero di iterazione e $x^{(0)}$ è un vettore arbitrario.

Metodo di Gauss-Sidel

metodo di Gauss-Sidel (o delle sostituzioni successive)

$$M = D - E \quad N = F.$$

La matrice di iterazione è

$$\mathcal{L}_1 = (D - E)^{-1}F.$$

Il metodo di Jacobi è definito se $D - E$ è non singolare ($a_{ii} \neq 0$ per ogni $i = 1, \dots, n$) ed è espresso dal procedimento iterativo

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) / a_{ii}, \quad i = 1, \dots, n$$

dove il superindice k indica il numero di iterazione e $x^{(0)}$ è un vettore arbitrario.

Metodo di Gauss-Sidel

metodo di Gauss-Sidel (o delle sostituzioni successive)

$$M = D - E \quad N = F.$$

La matrice di iterazione è

$$\mathcal{L}_1 = (D - E)^{-1}F.$$

Il metodo di Jacobi è definito se $D - E$ è non singolare ($a_{ii} \neq 0$ per ogni $i = 1, \dots, n$) ed è espresso dal procedimento iterativo

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) / a_{ii}, \quad i = 1, \dots, n$$

dove il superindice k indica il numero di iterazione e $x^{(0)}$ è un vettore arbitrario.

Metodi di Jacobi e Gauss-Sidel

$$A = \begin{pmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & -1 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 7 \\ 13 \\ -4 \end{pmatrix}$$

Matrice di iterazione di Jacobi:

$$J = \begin{pmatrix} 0 & 0 & -4/3 \\ -7/4 & 0 & -1/2 \\ -1/2 & -1/2 & 0 \end{pmatrix} \quad \rho(J) = 1.337510$$

Matrice di iterazione di Gauss-Sidel

$$G = \begin{pmatrix} 0 & 0 & -4/3 \\ 0 & 0 & 11/6 \\ 0 & 0 & -1/4 \end{pmatrix} \quad \rho(G) = 0.25$$

quindi Gauss-Sidel è convergente, Jacobi no.

Metodi di Jacobi e Gauss-Sidel

$$A = \begin{pmatrix} -3 & 3 & 6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{pmatrix} \quad b = \begin{pmatrix} -6 \\ -5 \\ 3 \end{pmatrix}$$

Matrice di iterazione di Jacobi:

$$J = \begin{pmatrix} 0 & 1 & -2 \\ 4/7 & 0 & 8/7 \\ 5/9 & 7/9 & 0 \end{pmatrix} \quad \rho(J) = 0.8133091$$

Matrice di iterazione di Gauss-Sidel

$$G = \begin{pmatrix} 0 & 1 & -2 \\ 0 & 4/7 & 0 \\ 0 & 1 & -10/9 \end{pmatrix} \quad \rho(G) = 1.11111$$

quindi Jacobi è convergente, Gauss-Sidel no.

Criteri di arresto



$$\|x_k - x_{k-1}\| \leq \varepsilon \|x_k\|$$

dove ε è una tolleranza relativa prefissata.

Matrici particolari

- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è con **diagonale dominante in senso stretto** se $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$ per ogni $i = 1, 2, \dots, n$.
- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è **irriducibile** se non esiste una matrice di permutazione P per cui

$$PAP^{-1} = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}$$

con B_{11}, B_{22} matrici quadrate.

- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ si dice **irriducibile con diagonale dominante** se A è irriducibile e $|a_{ii}| \geq \sum_{i \neq j} |a_{ij}|$, $i = 1, 2, \dots, n$, con almeno un indice i per cui la disuguaglianza vale in senso stretto.

Matrici particolari

- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è con **diagonale dominante in senso stretto** se $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$ per ogni $i = 1, 2, \dots, n$.
- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è **irriducibile** se non esiste una matrice di permutazione P per cui

$$PAP^{-1} = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}$$

con B_{11}, B_{22} matrici quadrate.

- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ si dice **irriducibile con diagonale dominante** se A è irriducibile e $|a_{ii}| \geq \sum_{i \neq j} |a_{ij}|$, $i = 1, 2, \dots, n$, con almeno un indice i per cui la disuguaglianza vale in senso stretto.

Matrici particolari

- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è con **diagonale dominante in senso stretto** se $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$ per ogni $i = 1, 2, \dots, n$.
- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è **irriducibile** se non esiste una matrice di permutazione P per cui

$$PAP^{-1} = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}$$

con B_{11}, B_{22} matrici quadrate.

- Una matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ si dice **irriducibile con diagonale dominante** se A è irriducibile e $|a_{ii}| \geq \sum_{i \neq j} |a_{ij}|$, $i = 1, 2, \dots, n$, con almeno un indice i per cui la disuguaglianza vale in senso stretto.

Criteri di convergenza

- Se A è una matrice di ordine n con **diagonale dominante in senso stretto**, oppure **irriducibile con diagonale dominante** allora il **metodo di Jacobi** è convergente.
- Se A è una matrice di ordine n con diagonale dominante in senso stretto, oppure irriducibile con diagonale dominante allora il **metodo di Gauss-Seidel** è convergente.
- Sia A una matrice hermitiana non singolare con elementi sulla diagonale reale e positivi. Allora il **metodo di Gauss-Seidel** è convergente se e solo se A è definita positiva.

Criteri di convergenza

- Se A è una matrice di ordine n con **diagonale dominante in senso stretto, oppure irriducibile con diagonale dominante** allora il **metodo di Jacobi** è convergente.
- Se A è una matrice di ordine n con **diagonale dominante in senso stretto, oppure irriducibile con diagonale dominante** allora il **metodo di Gauss-Seidel** è convergente.
- Sia A una matrice **hermitiana non singolare con elementi sulla diagonale reale e positivi**. Allora il **metodo di Gauss-Seidel è convergente se e solo se A è definita positiva**.

Criteri di convergenza

- Se A è una matrice di ordine n con **diagonale dominante in senso stretto, oppure irriducibile con diagonale dominante** allora il **metodo di Jacobi** è convergente.
- Se A è una matrice di ordine n con **diagonale dominante in senso stretto, oppure irriducibile con diagonale dominante** allora il **metodo di Gauss-Seidel** è convergente.
- Sia A una matrice **hermitiana non singolare con elementi sulla diagonale reale e positivi**. Allora il **metodo di Gauss-Seidel è convergente se e solo se A è definita positiva**.

Criteri di convergenza

Matrici tridiagonali

Sia $A \in \mathbb{C}^{n \times n}$ matrice tridiagonale:

$$A = \begin{pmatrix} a_1 & c_1 & & & \\ b_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & c_{n-1} & \\ & & & b_{n-1} & a_n \end{pmatrix}$$

in cui $a_i \neq 0, i = 1, \dots, n$. Valgono:

- 1 se μ autovalore di J , allora μ^2 autovalore di G ;
- 2 se λ autovalore non nullo di G allora $\sqrt{(\lambda)}$ autovalore di J .

Quindi, per le matrici tridiagonali, il metodo di Gauss-Sidel è convergente se e solo se lo è il metodo di Jacobi e vale:

$$\rho(G) = \rho^2(J)$$

Metodi di rilassamento

Il metodo di Gauss-Sidel può essere visto nella forma:

$$x_k = x_{k-1} + r_k$$

dove:

$$r_k = x_k - x_{k-1} = D^{-1}(Ex_k + Fx_{k-1} + b) - x_{k-1}$$

Quindi il punto $x^{(k)}$ si ottiene a partire da x_{k-1} effettuando un passo nella direzione r_k di lunghezza $\|r_k\|_2$. Non sempre questa è la scelta migliore per avere una convergenza veloce. Si modifica allora la lunghezza del passo introducendo un parametro ω :

$$x_k = x_{k-1} + \omega r_k$$

metodi di *rilassamento*:

- $\omega < 0$ *sottorilassamento*
- $\omega > 0$ *sovrarilassamento*

Metodi di rilassamento

Il metodo di Gauss-Sidel può essere visto nella forma:

$$x_k = x_{k-1} + r_k$$

dove:

$$r_k = x_k - x_{k-1} = D^{-1}(Ex_k + Fx_{k-1} + b) - x_{k-1}$$

Quindi il punto $x^{(k)}$ si ottiene a partire da x_{k-1} effettuando un passo nella direzione r_k di lunghezza $\|r_k\|_2$. Non sempre questa è la scelta migliore per avere una convergenza veloce. Si modifica allora la lunghezza del passo introducendo un parametro ω :

$$x_k = x_{k-1} + \omega r_k$$

metodi di *rilassamento*:

- $\omega < 0$ *sottorilassamento*
- $\omega > 0$ *sovrarilassamento*

Metodo SOR

metodo di Gauss-Seidel rilassato (o SOR):

$$x_k = (D - \omega E)^{-1}((1 - \omega)D + \omega F)x_{k-1} + \omega(D - \omega E)^{-1}b,$$

in cui la matrice di iterazione è

$$\mathcal{L}_\omega = (D - \omega E)^{-1}((1 - \omega)D + \omega F).$$

Metodi di rilassamento

Per ogni matrice A di ordine n una **condizione necessaria di convergenza per il metodo SOR** è (Teorema di Kahan)

$$0 < \omega < 2.$$

La condizione $0 < \omega < 2$ risulta anche sufficiente se la matrice A è definita positiva.

Teorema (Ostrowski-Reich).

Se A è definita positiva e ω è un numero reale $0 < \omega < 2$, allora il metodo di rilassamento è convergente.

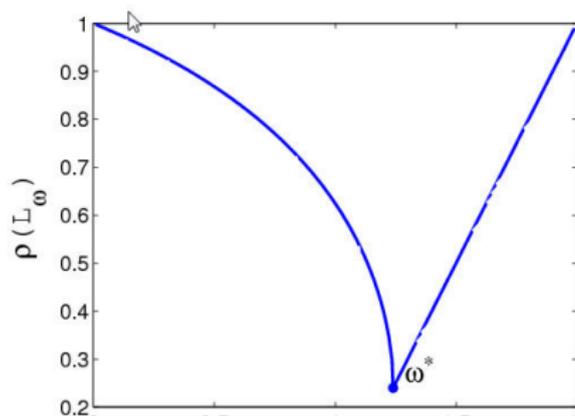
Convergenza metodi SOR

Il valore ottimale di ω è

$$\omega^* = \frac{2}{1 + \sqrt{1 - \rho(\mathcal{J})^2}}$$

e

$$\rho(\mathcal{L}_{\omega^*}) = \omega^* - 1 = \left(\frac{\rho(\mathcal{J})}{1 + \sqrt{1 - \rho(\mathcal{J})^2}} \right)^2. \quad (5)$$



Metodi di Krylov

Metodi iterativi negli spazi di Krylov per la risoluzione di

$$Ax = b$$

- Non hanno, al contrario dei metodi iterativi stazionari, una matrice di iterazione.
- minimizzano, alla k -esima iterazione, una misura dell'errore nello spazio:

$$x_0 + \mathcal{K}_k$$

dove x_0 è l'iterato iniziale e \mathcal{K}_k è il k -esimo sottospazio di Krylov:

$$\mathcal{K}_k = \text{span}(r_0, Ar_0, \dots, A^{k-1}r_0)$$

dove

$$r_k = b - Ax_k, \quad k = 1, \dots, n$$

Gradienti Coniugati (I)

A simmetrica e definita positiva ($\forall \mathbf{x} \in \mathbb{R} \mathbf{x}^t \mathbf{A} \mathbf{x} > 0$).

$$\min_{\mathbf{x} \in \mathbb{R}} \Phi(\mathbf{x}) \text{ dove } \Phi(\mathbf{x}) = \frac{1}{2} \mathbf{x}^t \mathbf{A} \mathbf{x} - \mathbf{x}^t \mathbf{b}$$



$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

Vettore **residuo**: $\mathbf{r} = \mathbf{b} - \mathbf{A} \mathbf{x}$ Si approssima la soluzione mediante la successione:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

dove \mathbf{p}_k è detta direzione di discesa, $\alpha_k \geq 0$ è il passo.

Gradienti Coniugati: convergenza

Teorema. Sia A spd. Allora l'algoritmo dei Gradienti Coniugati (CG) calcola la soluzione in n iterazioni.

Teorema Sia A spd. Se ci sono esattamente $k \leq N$ autovalori distinti di A , allora le iterazioni di CG terminano in al più k iterazioni

- Questo è vero in aritmetica esatta, in aritmetica finita in generale le iterazioni per la convergenza sono in numero maggiore di n .
- Per questo il CG viene generalmente visto come metodo iterativo e non esatto.
- Quindi le iterazioni terminano quando una certa tolleranza di errore è raggiunta

Gradienti Coniugati: convergenza

Teorema. Sia A spd. Allora l'algoritmo dei Gradienti Coniugati (CG) calcola la soluzione in n iterazioni.

Teorema Sia A spd. Se ci sono esattamente $k \leq N$ autovalori distinti di A , allora le iterazioni di CG terminano in al più k iterazioni

- Questo è vero in aritmetica esatta, in aritmetica finita in generale le iterazioni per la convergenza sono in numero maggiore di n .
- Per questo il CG viene generalmente visto come metodo iterativo e non esatto.
- Quindi le iterazioni terminano quando una certa tolleranza di errore è raggiunta

Gradienti Coniugati: convergenza

Teorema. Sia A spd. Allora l'algoritmo dei Gradienti Coniugati (CG) calcola la soluzione in n iterazioni.

Teorema Sia A spd. Se ci sono esattamente $k \leq N$ autovalori distinti di A , allora le iterazioni di CG terminano in al più k iterazioni

- Questo è vero in aritmetica esatta, in aritmetica finita in generale le iterazioni per la convergenza sono in numero maggiore di n .
- Per questo il CG viene generalmente visto come metodo iterativo e non esatto.
- Quindi le iterazioni terminano quando una certa tolleranza di errore è raggiunta

Gradienti Coniugati:criterio di arresto

Relazione di decrescita dell'errore in norma A:

$$\|x_k - x^*\|_A \leq 2\|x_0 - x^*\|_A \left(\frac{K_2(A) - 1}{K_2(A) + 1} \right)^2$$

dove $\sqrt{K_2(A)} = \lambda_1/\lambda_n$.

- Questa stima può essere anche molto pessimistica. Per esempio quando gli autovalori di A sono accumulati in pochi intervalli, il numero di condizione può essere molto grande ma il CG converge velocemente.
- La trasformazione del problema in uno i cui autovalori sono tutti accumulati verso 1 si chiama **precondizionamento**

Gradienti Coniugati:criterio di arresto

Relazione di decrescita dell'errore in norma A:

$$\|x_k - x^*\|_A \leq 2\|x_0 - x^*\|_A \left(\frac{K_2(A) - 1}{K_2(A) + 1} \right)^2$$

dove $\sqrt{K_2(A)} = \lambda_1/\lambda_n$.

- Questa stima può essere anche molto pessimistica. Per esempio quando gli autovalori di A sono accumulati in pochi intervalli, il numero di condizione può essere molto grande ma il CG converge velocemente.
- La trasformazione del problema in uno i cui autovalori sono tutti accumulati verso 1 si chiama **precondizionamento**

Gradienti Coniugati: implementazione

- Criterio per terminare le iterazioni:

$$\|b - Ax_k\|_2 = \|r_k\|_2 \leq \eta \|b\|_2.$$

- La matrice A non deve necessariamente essere memorizzata; solo una funzione per calcolare il prodotto matrice-vettore è necessaria. Per questo i metodi negli spazi di Krylov sono di solito detti **matrix-free**.
- Il costo per iterazione è il costo della funzione che calcola il prodotto Av .

Gradienti Coniugati: implementazione

- Criterio per terminare le iterazioni:

$$\|b - Ax_k\|_2 = \|r_k\|_2 \leq \eta \|b\|_2.$$

- La matrice A non deve necessariamente essere memorizzata; solo una funzione per calcolare il prodotto matrice-vettore è necessaria. Per questo i metodi negli spazi di Krylov sono di solito detti **matrix-free**.
- Il costo per iterazione è il costo della funzione che calcola il prodotto Av .

Gradienti Coniugati: implementazione

- Criterio per terminare le iterazioni:

$$\|b - Ax_k\|_2 = \|r_k\|_2 \leq \eta \|b\|_2.$$

- La matrice A non deve necessariamente essere memorizzata; solo una funzione per calcolare il prodotto matrice-vettore è necessaria. Per questo i metodi negli spazi di Krylov sono di solito detti matrix-free.
- Il costo per iterazione è il costo della funzione che calcola il prodotto Av .

Gradienti Coniugati: implementazione

- Criterio per terminare le iterazioni:

$$\|b - Ax_k\|_2 = \|r_k\|_2 \leq \eta \|b\|_2.$$

- La matrice A non deve necessariamente essere memorizzata; solo una funzione per calcolare il prodotto matrice-vettore è necessaria. Per questo i metodi negli spazi di Krylov sono di solito detti **matrix-free**.
- Il costo per iterazione è il costo della funzione che calcola il prodotto Av .

Gradienti Coniugati: matlab

Nella function Matlab `cg` si implementa tale metodo arrestando le iterazioni se $\|\mathbf{r}_k\| < \epsilon \|\mathbf{b}\|$

```
function [x, iter] = cg(A, b);
x = zeros(size(b));
r = b; nb = sqrt(b' * b);
num = r' * r; p = r;
err = sqrt(num); maxit = n; iter = 0;
while(err > eps * nb & iter < maxit)
    v = A * p; den = p' * v; alfa = num/den; den = num;
    x = x + alfa * p; r = r - alfa * v;
    num = r' * r; err = sqrt(num);
    beta = num/den; p = r + beta * p; iter = iter + 1;
end
```

Precondizionamento

- Se il numero di condizione $\mathcal{K}_2(A)$ è grande il metodo può essere lento.
- Per ridurre il numero di condizione, si può sostituire il sistema $Ax=b$ con un altro sistema con matrice spd e stessa soluzione.
- Se M è una matrice spd che approssima A^{-1} allora gli autovalori di MA sono vicini ad 1.
- Tuttavia MA in generale non è spd e quindi non è possibile applicare i CG al sistema: $MAx = Mb$.

Precondizionamento

- Se il numero di condizione $\mathcal{K}_2(A)$ è grande il metodo può essere lento.
- Per ridurre il numero di condizione, si può sostituire il sistema $Ax=b$ con un altro sistema con matrice spd e stessa soluzione.
- Se M è una matrice spd che approssima A^{-1} allora gli autovalori di MA sono vicini ad 1.
- Tuttavia MA in generale non è spd e quindi non è possibile applicare i CG al sistema: $MAx = Mb$.

Precondizionamento

- Se il numero di condizione $\mathcal{K}_2(A)$ è grande il metodo può essere lento.
- Per ridurre il numero di condizione, si può sostituire il sistema $Ax=b$ con un altro sistema con matrice spd e stessa soluzione.
- Se M è una matrice spd che approssima A^{-1} allora gli autovalori di MA sono vicini ad 1.
- Tuttavia MA in generale non è spd e quindi non è possibile applicare i CG al sistema: $MAx = Mb$.

Precondizionamento

- Se il numero di condizione $\mathcal{K}_2(A)$ è grande il metodo può essere lento.
- Per ridurre il numero di condizione, si può sostituire il sistema $Ax=b$ con un altro sistema con matrice spd e stessa soluzione.
- Se M è una matrice spd che approssima A^{-1} allora gli autovalori di MA sono vicini ad 1.
- Tuttavia MA in generale non è spd e quindi non è possibile applicare i CG al sistema: $MAx = Mb$.

Precondizionamento

- Se S approssima B^{-1} dove $B^2 = A$, posso scrivere il sistema con un preconditionamento two-sides:

$$SASz = Sy$$

la cui soluzione è $z = S^{-1}x$.

- Questo procedimento si può fare senza ricorrere a due moltiplicazioni per la matrice S ad ogni iterazione.

Precondizionamento

- Se S approssima B^{-1} dove $B^2 = A$, posso scrivere il sistema con un precondizionamento two-sides:

$$SASz = Sy$$

la cui soluzione è $z = S^{-1}x$.

- Questo procedimento si può fare senza ricorrere a due moltiplicazioni per la matrice S ad ogni iterazione.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

Precondizionamento

- Il costo del PCG è uguale a quello del CG con in più:
 - 1 l'applicazione del preconditionatore M
 - 2 il prodotto scalare per calcolare τ_k
- Precondizionatori efficienti tengono conto della struttura della matrice A
- esempi di preconditionatori:
 - ▶ preconditionatore per il metodo di Jacobi: M è l'inverso della diagonale di A
 - ▶ Cholsky incompleto: se $A = LL^T + E$ con E piccola, $M = (LL^T)^{-1}$.

CG applicato alle equazioni normali (CGLS o CGNR)

Se A è nonsingolare e non simmetrica, si può pensare di risolvere $Ax = b$ applicando il CG alle equazioni normali:

$$A^T Ax = A^T b.$$

Questo metodo è detto **CGLS** o **CGNR**. Tutta la teoria del CG e PCG si può ora applicare.