

# Serie 2b

## Sistemi Lineari – Metodi Iterativi

---

©2024 - Questo testo (compresi i quesiti ed il loro svolgimento) è coperto da diritto d'autore. Non può essere sfruttato a fini commerciali o di pubblicazione editoriale. Non possono essere ricavati lavori derivati. Ogni abuso sarà punito a termine di legge dal titolare del diritto. This text is licensed to the public under the Creative Commons Attribution-NonCommercial-NoDerivs2.5 License (<http://creativecommons.org/licenses/by-nc-nd/2.5/>)

---

### 1 Metodi iterativi preconditionati

Una strategia per implementare un metodo iterativo per la risoluzione del sistema lineare  $A\mathbf{x} = \mathbf{b}$  consiste nell'effettuare lo *splitting* della matrice  $A$ , dalla forma

$$A = P - (P - A)$$

dove  $P$  è una opportuna matrice non singolare detta *precondizionatore*. Con questa notazione, risolvere  $A\mathbf{x} = \mathbf{b}$  è equivalente a risolvere

$$P\mathbf{x} = (P - A)\mathbf{x} + \mathbf{b},$$

identità che suggerisce la definizione del metodo iterativo: assegnato  $\mathbf{x}^{(0)}$ , ottenere  $\mathbf{x}^{(k+1)}$  per  $k \geq 0$  risolvendo i sistemi

$$P\mathbf{x}^{(k+1)} = (P - A)\mathbf{x}^{(k)} + \mathbf{b}, \quad k \geq 0.$$

La matrice di iterazione per tali metodi risulta quindi essere  $B = P^{-1}(P - A) = I - P^{-1}A$ . Il metodo iterativo può anche essere posto nella forma

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \mathbf{r}^{(k)}, \quad k \geq 0,$$

dove  $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$  è il residuo alla  $k$ -esima iterazione. Questo metodo richiede ad ogni passo di trovare il cosiddetto *residuo preconditionato*  $\mathbf{z}^{(k)}$  dato dalla soluzione del sistema lineare

$$P\mathbf{z}^{(k)} = \mathbf{r}^{(k)}, \tag{1}$$

di conseguenza, la nuova iterata è definita da  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{z}^{(k)}$ , e la matrice  $P$ , oltre ad essere non singolare, deve essere scelta in modo tale che il costo computazionale richiesto dalla risoluzione del sistema (1) sia basso.

#### 1.1 Metodo di Jacobi

Consiste nello scegliere come *precondizionatore*  $P$  la matrice diagonale  $D$  rappresentata dagli elementi diagonali di  $A$ . Tale matrice è facilmente invertibile, se gli  $a_{ii} \neq 0$ ,  $i = 1, \dots, n$ , in quanto:

$$D = \begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & a_{nn} \end{bmatrix} \implies D^{-1} = \begin{bmatrix} 1/a_{11} & 0 & 0 & \cdots & 0 \\ 0 & 1/a_{22} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1/a_{nn} \end{bmatrix}$$

e il metodo può essere scritto direttamente in forma matriciale:

$$\begin{aligned}
& \mathbf{x}^{(0)} \text{ assegnato} \\
& \mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}, \quad k \geq 0 \\
& \mathbf{z}^{(k)} = D^{-1}\mathbf{r}^{(k)}, \\
& \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{z}^{(k)}.
\end{aligned} \tag{2}$$

EsPLICITANDO il calcolo di ogni singola componente della soluzione si ottiene la seguente forma algebrica:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n, \quad k = 0, 1, \dots \tag{3}$$

## 1.2 Metodo di Gauss-Seidel

Questo metodo si differenzia dal metodo di Jacobi per il fatto che nel calcolo di  $x_i^{(k+1)}$  vengono utilizzate le nuove componenti già disponibili  $x_j^{(k+1)}$ ,  $j = 1, \dots, i-1$ . Si modifica allora il metodo di Jacobi come segue:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n, \quad k = 0, 1, \dots \tag{4}$$

che equivale ad aver scelto come preconditionatore  $P = T$ , cioè la parte triangolare inferiore della matrice  $A$ .

$$T = \begin{bmatrix} a_{1,1} & 0 & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & 0 & \cdots & 0 \\ \vdots & a_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

La corrispondente forma matriciale

$$\begin{aligned}
& \mathbf{x}^{(0)} \text{ assegnato} \\
& \mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}, \quad k \geq 0 \\
& T\mathbf{z}^{(k)} = \mathbf{r}^{(k)}, \\
& \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{z}^{(k)},
\end{aligned} \tag{5}$$

evidenzia il fatto che ad ogni iterazione del metodo si risolve un sistema triangolare inferiore.

### Esercizio 1.1

Si consideri la seguente matrice  $A \in \mathbb{R}^{n \times n}$ :

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ R_1 & -R_2 & 0 & 0 & \cdots & 0 \\ 0 & R_1 & -R_2 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_1 & -R_2 \end{bmatrix}.$$

Si ponga  $n = 100$ ,  $R_1 = 1$  e  $R_2 = 2$ .

1. Memorizzare la matrice in formato pieno e determinare il numero di elementi non nulli tramite il comando `nnz` e visualizzare il pattern di sparsità della matrice con il comando `spy`. Verificare la diversa occupazione di memoria nel caso la matrice fosse convertita in formato sparso (*Suggerimento*: utilizzare il comando `sparse`<sup>1</sup>).
2. Si calcoli la fattorizzazione LU della matrice  $A$  (funzione `lu` di Matlab®); cosa si può osservare confrontando il pattern di sparsità delle matrici  $L$  e  $U$  con quello di  $A$ ? Che fenomeno si è verificato?
3. Si costruiscano esplicitamente le matrici di iterazione  $B_J = D^{-1}(D - A)$  e  $B_{GS} = T^{-1}(T - A)$  associate rispettivamente ai metodi di Jacobi e Gauss-Seidel, a partire dallo *splitting* della matrice  $A$ . Si calcolino i relativi raggi spettrali  $\rho(B_J)$  e  $\rho(B_{GS})$ , utilizzando il comando `eig`. La condizione necessaria e sufficiente per la convergenza del metodo iterativo è soddisfatta in entrambi i casi?
4. Scrivere una funzione Matlab® che implementi il metodo di Jacobi per un generico sistema lineare  $A\mathbf{x} = \mathbf{b}$ . Formulare tale funzione in modo da implementare la versione *matriciale* (2) del metodo, dopo aver esplicitamente costruito la matrice  $D^{-1}$ . I parametri in ingresso richiesti dalla funzione sono la matrice  $A$ , il termine noto  $\mathbf{b}$ , la guess iniziale  $\mathbf{x}^{(0)}$ , la tolleranza per il criterio d'arresto `tol` e il numero massimo di iterazioni ammesse `nmax`. La funzione restituisce la soluzione calcolata  $\mathbf{x}$  e il numero di iterazioni effettuate `k`. L'intestazione della funzione sarà, ad esempio la seguente:

```
[x,k] = jacobi(A,b,x0,tol,nmax)
```

Si arresti l'iterazione quando  $\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|} \leq \text{tol}$  (criterio d'arresto del residuo normalizzato).

5. Utilizzare la funzione scritta al punto precedente per determinare la soluzione del sistema lineare  $A\mathbf{x} = \mathbf{b}$ , con  $\mathbf{b}$  vettore di dimensione  $n$ :

$$\mathbf{b} = [2, 1, \dots, 1]^T.$$

Si ponga  $\mathbf{x}^{(0)} = [0, 0, \dots, 0]^T$ , `tol` =  $10^{-6}$  e `nmax` = 1000.

---

<sup>1</sup>Il comando inverso di `sparse` è il comando `full`, che converte una matrice nel formato pieno

## Esercizio 1.2

Si consideri il seguente sistema di equazioni lineari:

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

con:

$$A = \begin{bmatrix} 9 & -3 & 1 & & & & \\ -3 & 9 & -3 & 1 & & & \\ 1 & -3 & 9 & -3 & 1 & & \\ & 1 & -3 & 9 & -3 & 1 & \\ & & 1 & -3 & 9 & -3 & 1 \\ & & & 1 & -3 & 9 & -3 \\ & & & & 1 & -3 & 9 \end{bmatrix} \quad (6)$$

e termine noto:

$$\mathbf{b} = [7 \ 4 \ 5 \ 5 \ 5 \ 4 \ 7]^T.$$

1. Costruire la matrice  $A$  e il termine noto  $\mathbf{b}$ .
2. La matrice  $A$  è a dominanza diagonale stretta per righe?
3. La matrice  $A$  è simmetrica definita positiva?
4. Scrivere la funzione Matlab<sup>®</sup> che implementa il metodo di Gauss-Seidel nella forma matriciale (5). A partire dall'implementazione di `jacobi.m` dell'esercizio precedente, apportare le modifiche opportune per ottenere `gs.m`.

*Suggerimento:* Utilizzare la funzione `fwsb.m` implementata nella Serie 02a per la risoluzione del sistema triangolare inferiore. I parametri in ingresso richiesti dalla funzione sono la matrice  $A$ , il termine noto  $\mathbf{b}$ , la guess iniziale  $\mathbf{x}^{(0)}$ , la tolleranza per il criterio d'arresto `toll` e il numero massimo di iterazioni ammesse `nmax`. La funzione restituisce la soluzione calcolata  $\mathbf{x}$  e il numero di iterazioni effettuate  $k$ . L'intestazione della funzione sarà, ad esempio la seguente:

$$[\mathbf{x}, k] = \text{gs}(A, \mathbf{b}, \mathbf{x}_0, \text{toll}, \text{nmax}).$$

Si arresti l'iterazione quando  $\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|} \leq \text{toll}$ .

5. Si ponga  $\mathbf{x}^{(0)} = [0, 0, \dots, 0]^T$ , `toll` =  $10^{-6}$  e `nmax` = 1000. Si calcoli la soluzione tramite la funzione `gs`.
6. Si calcoli la soluzione del sistema anche con il metodo di Jacobi, e si confrontino il numero di iterazioni necessarie per arrivare a convergenza per i due metodi. Commentare i risultati.

## 2 I metodi di Richardson

Nella sezione precedente sono stati considerati i metodi di Jacobi e Gauss-Seidel, che possono essere ricondotti alla seguente definizione generale di metodo iterativi:

$$P \left( \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right) = \mathbf{r}^{(k)}, \quad k > 0,$$

dove la matrice  $P$  non singolare è detta *precondizionatore*. È possibile generalizzare questo metodo tramite l'introduzione di un parametro di accelerazione  $\alpha_k$ :

$$P \left( \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right) = \alpha_k \mathbf{r}^{(k)}, \quad k \geq 0, \quad \alpha_k \neq 0.$$

Lo scopo di  $\alpha_k$  è migliorare le proprietà di convergenza della successione  $\{x^{(k)}\}$ . Il metodo si definisce *stazionario* nel caso in cui  $\alpha_k = \alpha$ , con  $\alpha$  costante assegnata, *dinamico* nel caso in cui  $\alpha_k$  possa cambiare ad ogni iterazione.

A partire dagli algoritmi di Sezione 1, è possibile ottenere quello che implementa il metodo di Richardson tramite l'aggiunta del parametro  $\alpha_k$ : dato  $\mathbf{x}^{(0)}$  assegnato, si ponga  $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$  e per  $k = 0, 1, \dots$

$$\begin{aligned} P\mathbf{z}^{(k)} &= \mathbf{r}^{(k)} \\ \text{calcolare il parametro di accelerazione } \alpha_k \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)} \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k A\mathbf{z}^{(k)}. \end{aligned}$$

Se  $A$  e  $P$  sono entrambe simmetriche e definite positive, il metodo di Richardson *stazionario* converge per ogni possibile scelta di  $\mathbf{x}^{(0)}$  se solo se

$$0 < \alpha < 2/\lambda_{\max},$$

dove  $\lambda_{\max}(> 0)$  è l'autovalore massimo di  $P^{-1}A$ . Inoltre il raggio spettrale  $\rho(B_\alpha)$  della matrice di iterazione  $B_\alpha = I - \alpha P^{-1}A$  è minimo quando

$$\alpha = \alpha_{\text{opt}} = \frac{2}{\lambda_{\min} + \lambda_{\max}}$$

essendo  $\lambda_{\min}$  l'autovalore minimo di  $P^{-1}A$ .

Se  $A$  e  $P$  sono entrambe simmetriche e definite positive, il metodo di Richardson *dinamico* converge se, ad esempio,  $\alpha_k$  è scelto nel modo seguente:

$$\alpha_k = \frac{(\mathbf{z}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{z}^{(k)})^T A \mathbf{z}^{(k)}}, \quad \forall k \geq 0$$

dove  $\mathbf{z}^{(k)} = P^{-1}\mathbf{r}^{(k)}$  è il residuo preconditionato. Questo metodo è detto del *gradiente preconditionato* o semplicemente del *gradiente* se  $P = I$ .

## Esercizio 2.1

Si consideri il problema lineare  $A\mathbf{x} = \mathbf{b}$ , dove la matrice  $A \in \mathbb{R}^{n \times n}$  è pentadiagonale:

$$A = \begin{bmatrix} 4 & -1 & -1 & & & & \\ -1 & 4 & -1 & -1 & & & \\ -1 & -1 & 4 & -1 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & -1 & 4 & -1 & -1 \\ & & & -1 & -1 & 4 & -1 \\ & & & & -1 & -1 & 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ \vdots \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}. \quad (7)$$

Si vuole risolvere tale problema con il metodo di Richardson, soddisfacendo una tolleranza di  $10^{-5}$ , a partire dal vettore soluzione iniziale  $\mathbf{x}_0 = [0, \dots, 0]^T$ .

1. Utilizzando uno script si creino la matrice  $A$  (con  $n = 50$ ), il termine noto  $\mathbf{b}$  ed il vettore soluzione iniziale  $\mathbf{x}_0$ .
2. Si verifichi che la matrice  $A$  sia simmetrica e definita positiva e se ne calcoli il numero di condizionamento *senza utilizzare il comando* `cond`.
3. Si implementi la funzione `richardson.m` in grado di applicare il metodo di *Richardson preconditionato* sia per il caso stazionario, sia per quello dinamico (gradiente preconditionato) ad un generico sistema lineare. A partire dall'implementazione di `jacobi.m` dell'Esercizio 1.1, apportare le modifiche opportune per ottenere `richardson.m`.

La funzione deve avere la seguente intestazione:

$$[\mathbf{x}, k] = \text{richardson}(A, \mathbf{b}, P, \mathbf{x}_0, \text{tol}, \text{nmax}, \text{alpha}),$$

dove  $A$  è la matrice del sistema lineare,  $\mathbf{b}$  è il termine noto,  $P$  è il preconditionatore,  $\mathbf{x}_0$  è il vettore iniziale, `tol` è la tolleranza (per il criterio di arresto del residuo normalizzato), `nmax` è il numero massimo di iterazioni ed `alpha` è il valore del parametro di accelerazione (richiesto solo per il caso statico: nel caso dinamico deve essere omesso); in uscita la funzione restituisce la soluzione ottenuta  $\mathbf{x}$  ed il numero di iterazioni svolte  $k$ . *Suggerimento:* all'interno della funzione si usi il comando `nargin` per contare il numero di parametri in ingresso e decidere se `alpha` sia presente o se sia stato omesso.

4. Si risolva il sistema lineare con il metodo di Richardson stazionario non preconditionato ( $P = I$ ), utilizzando i seguenti parametri di accelerazione:  $\alpha = 0.2$ ,  $\alpha = 0.33$  ed  $\alpha = \alpha_{\text{opt}} = 2/(\lambda_{\min} + \lambda_{\max})$  (parametro di accelerazione ottimale) dove  $\lambda_{\min}$  e  $\lambda_{\max}$  sono rispettivamente l'autovalore minimo e l'autovalore massimo della matrice  $P^{-1}A$ . Per ciascun valore di  $\alpha$  si calcoli il raggio spettrale della matrice di iterazione  $(I - \alpha P^{-1}A)$  e si risolva il sistema riportando a video il numero di iterazioni eseguite (si suggerisce di utilizzare `nmax = 10000`).

5. Si risolva il sistema lineare con il metodo di Richardson stazionario utilizzando come preconditionatore la matrice triangolare inferiore di  $A$  e come parametro di accelerazione  $\alpha = 1$  (ovvero il metodo di Gauss–Seidel), riportando a video il numero di iterazioni eseguite. Calcolare il raggio spettrale della matrice di iterazione corrispondente. Si verifichi infine che il risultato ottenuto è identico a quello che si otterrebbe utilizzando la funzione `gs.m`, che implementa il metodo di Gauss–Seidel.
6. Si consideri il seguente preconditionatore tridiagonale:

$$P = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}. \quad (8)$$

Si risolva il sistema lineare con il metodo di Richardson stazionario utilizzando come preconditionatore la matrice  $P$ , dopo aver verificato che sia simmetrica e definita positiva, e come parametro di accelerazione  $\alpha_{opt}$ , riportando a video il numero di iterazioni eseguite. Si calcolino il raggio spettrale della matrice di iterazione ed il numero di condizionamento di  $P^{-1}A$ . Si risolva infine il sistema utilizzando il metodo del gradiente preconditionato.

### 3 Metodi del gradiente e gradiente coniugato: interpretazione geometrica

Ad ogni sistema lineare  $A\mathbf{x} = \mathbf{b}$  con  $A$  simmetrica e definita positiva si può associare una forma quadratica

$$\varphi(\mathbf{x}) = \frac{1}{2} (\mathbf{x}^T A \mathbf{x}) - \mathbf{x}^T \mathbf{b},$$

il cui punto di minimo  $\mathbf{x}^*$  è proprio la soluzione del sistema lineare, cioè  $A\mathbf{x}^* = \mathbf{b}$  (che esprime la condizione  $\nabla\varphi = \mathbf{0}$ ).

Ad esempio, se  $A \in \mathbb{R}^{2 \times 2}$ , la forma quadratica associata  $\varphi$  rappresenta un paraboloide in  $\mathbb{R}^3$ . Le linee di livello di  $\varphi$  sono quindi delle ellissi nel piano  $(x, y)$ , i cui assi sono orientati nelle direzioni degli autovettori di  $A$ ,  $\mathbf{v}_1, \mathbf{v}_2$ . La lunghezza degli assi è inversamente proporzionale alle radici degli autovalori di  $A$ ,  $\lambda_1, \lambda_2$ .

Se si vuole arrivare al punto di minimo di  $\varphi$  un'idea può essere quella di partire da un punto qualsiasi  $\mathbf{x}_k$  e muoversi nella direzione di massima discesa, cioè in direzione  $\mathbf{z}_k = -\nabla\varphi(\mathbf{x}_k)$ , che, nel caso di una forma quadratica, è proprio  $\mathbf{z}_k = -A\mathbf{x}_k + \mathbf{b} = \mathbf{r}_k$ .

Di quanto ci si deve muovere in direzione  $\mathbf{z}_k$ ? Se fissiamo a priori un valore  $\alpha$  si ottiene:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{z}_k,$$

ovvero il metodo di Richardson stazionario. Il passo ottimale è quello che colloca  $\mathbf{x}_{k+1}$  nel punto di minimo della forma quadratica lungo la direzione  $\mathbf{z}_k$ ; tale valore coincide con il valore di  $\alpha_k$  utilizzato dal metodo del gradiente.

### Esercizio 3.1

1. Costruire la matrice

$$A_1 = \begin{pmatrix} 6.8 & -2.4 \\ -2.4 & 8.2 \end{pmatrix}$$

e verificare che ha come autovettori

$$\mathbf{v}_1 = \frac{1}{5} \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \quad \mathbf{v}_2 = \frac{1}{5} \begin{pmatrix} -3 \\ 4 \end{pmatrix}$$

e come autovalori

$$\lambda_1 = 5, \quad \lambda_2 = 10.$$

Costruire il vettore

$$\mathbf{b} = \begin{pmatrix} 4 \\ 8 \end{pmatrix}$$

e visualizzare la forma quadratica  $\varphi_1(\mathbf{x}) = \frac{1}{2}(\mathbf{x}^T A_1 \mathbf{x}) - \mathbf{x}^T \mathbf{b}$  associata e le corrispondenti linee di livello. Cosa succede alla forma quadratica se si cambia il primo autovalore in  $\lambda_1 = 1$ ?

In particolare, si definisca  $A_2 = V D V^T$ , dove  $V = [\mathbf{v}_1, \mathbf{v}_2]$  e  $D = \text{diag}(1, \lambda_2)$  e si consideri la forma quadratica

$$\varphi_2(\mathbf{x}) = \frac{1}{2}(\mathbf{x}^T A_2 \mathbf{x}) - \mathbf{x}^T \mathbf{b}.$$

2. Utilizzare il metodo di Richardson stazionario e il metodo del gradiente per risolvere il problema di minimizzazione della forma quadratica  $\varphi_2$ . Per il metodo di Richardson utilizzare come valori  $\alpha = 0.05$  e  $\alpha = 0.24$ . Visualizzare la successione dei punti che si ottiene in ciascuno dei tre casi. (Suggerimento: *modificare la `function richardson.m`, creando una nuova `function richardson_it.m` che implementi lo stesso metodo, ma restituisca in uscita la matrice che ha per colonne la successione delle iterate  $\mathbf{x}_k$ .*)

All'iterazione  $k$  il metodo del gradiente avanza in direzione  $\mathbf{z}_k$  perpendicolare alla linea di livello passante per il punto  $\mathbf{x}_k$ , e sceglie il nuovo punto  $\mathbf{x}_{k+1}$  sulla linea di livello alla quale la direzione  $\mathbf{z}_k$  è tangente.

Questo si traduce in una discesa in cui due direzioni consecutive sono sempre ortogonali, ed il metodo risulta particolarmente inefficiente se le linee di livello sono molto eccentriche (cioè quando gli autovalori di  $A$  sono molto diversi fra loro); in tal caso infatti la direzione  $\mathbf{z}_k$  non punta verso il minimo.

Una possibile soluzione è quella di utilizzare un preconditionatore  $P$  per il sistema lineare. Ci sono due alternative possibili:

- a) utilizzare  $P$  per passare ad una forma quadratica con lo stesso minimo ma con linee di livello meno eccentriche, ed applicare a questa nuova forma quadratica il metodo del gradiente.  $P$  deve essere simmetrica e definita positiva, in modo che  $P^{-1}A$  abbia autovalori reali e positivi, e quindi sia possibile associare al sistema lineare preconditionato  $P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}$  una nuova forma quadratica

$$\varphi_{prec}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}^T P^{-1} A \mathbf{x}) - \mathbf{x}^T P^{-1} \mathbf{b}.$$



Inoltre,  $P$  deve essere un'approssimazione di  $A$ , (in modo che il numero di condizionamento  $K_2(P^{-1}A)$  sia vicino a 1), ma più facile da "invertire" di  $A$ ;

- b) utilizzare  $P$  per scegliere delle direzioni non più ortogonali fra di loro, ma che tengano conto della forma delle linee di livello, privilegiando passi più ampi nella direzione in cui le linee di livello sono più allungate. Queste operazioni sono quelle eseguite dal metodo del *gradiente preconditionato*. In questo caso  $P$  deve essere definita positiva (ma non necessariamente simmetrica) e deve essere nuovamente un'approssimazione di  $A$  (al fine di scegliere le direzioni in modo accurato) ma più facile da invertire di  $A$  (ad ogni iterazione il gradiente preconditionato risolve  $P\mathbf{z} = \mathbf{r}$ ).

### Esercizio 3.1 (cont.)

3. Utilizzare i due approcci appena illustrati per risolvere il sistema lineare associato alla forma quadratica  $\varphi_2$ , utilizzando come preconditionatore la matrice

$$P = \begin{pmatrix} 1.0912 & -0.8587 \\ -0.8587 & 1.5921 \end{pmatrix}.$$

Visualizzare le linee di livello della forma quadratica  $\varphi_{prec}$  associata al sistema  $P^{-1}A_2\mathbf{x} = P^{-1}\mathbf{b}$ :  $\varphi_{prec}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}^T P^{-1}A_2\mathbf{x}) - \mathbf{x}^T P^{-1}\mathbf{b}$  e il percorso seguito dai due approcci nella discesa verso il punto di minimo.

Si può dimostrare che le direzioni utilizzate dal metodo del gradiente preconditionato sono tali che  $\mathbf{z}_{k+1}^T P\mathbf{z}_k = 0$  (direzioni *P-ortogonali*). Questa proprietà in generale non è valida per due direzioni non consecutive (cioè, ad esempio,  $\mathbf{z}_3^T P\mathbf{z}_1 \neq 0$ ), e quindi il metodo del gradiente preconditionato non è ancora ottimale.

Si introduce quindi il metodo del *gradiente coniugato*: in questo metodo si costruiscono iterativamente delle direzioni  $\mathbf{p}_k$  in modo che siano tutte *A*-coniugate (*A*-ortogonali fra loro):

$$\mathbf{p}_j^T A\mathbf{p}_{k+1} = 0, \quad j = 0, 1, \dots, k.$$

Si può dimostrare che il gradiente coniugato converge alla soluzione esatta in al più  $n$  passi, se  $A \in \mathbb{R}^{n \times n}$  e  $\mathbf{b} \in \mathbb{R}^n$ , supponendo di lavorare in aritmetica esatta.

L'algoritmo che implementa il metodo del gradiente coniugato è il seguente: si inizializzano  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$  e  $\mathbf{p}_0 = \mathbf{r}_0$ , quindi si ripete per  $k = 0, 1, \dots$

$$\begin{aligned} \alpha_k &= \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T A\mathbf{p}_k} \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A\mathbf{p}_k \\ \beta_k &= \frac{\mathbf{p}_k^T A\mathbf{r}_{k+1}}{\mathbf{p}_k^T A\mathbf{p}_k} \\ \mathbf{p}_{k+1} &= \mathbf{r}_{k+1} - \beta_k \mathbf{p}_k \end{aligned}$$

fino a quando non viene raggiunta una certa tolleranza o un massimo numero di iterazioni.

Osserviamo che i metodi del gradiente coniugato e del gradiente coniugato preconditionato sono implementati nella funzione nativa Matlab<sup>®</sup> `pcg`:

```
[x, flag, relres, iter, resvec] = pcg(A, b, tol, maxit, M1, M2, x0)
```

- Output:

1. `x` vettore soluzione all'ultima iterazione del metodo;
2. `iter` numero di iterazioni eseguite.

Se non si è interessati agli altri parametri restituiti da `pcg` se ne può sopprimere l'output nel modo seguente

```
[x, ~, ~, iter] = pcg(____)
```

- Input:

1. `A`, `b`, `tol`, `nmax`, `x0` sono rispettivamente la matrice del sistema lineare, il termine noto, la tolleranza, il numero massimo di iterazioni e il vettore iniziale.
2. `M1` e `M2` sono i fattori della matrice di preconditionamento  $M = M1 * M1$ . Se si vuole applicare il metodo del gradiente coniugato non preconditionato è sufficiente fornire in input `eye(size(A))` sia per `M1` che per `M2`.

Si utilizzi il comando `help pcg` per una guida più dettagliata.

### Esercizio 3.1 (cont.)

4. Risolvere il sistema lineare associato alla forma quadratica  $\varphi_2$  con il metodo del gradiente coniugato utilizzando la funzione Matlab<sup>®</sup> `pcg`

```
x = pcg(A, b, toll, nmax, eye(size(A)), eye(size(A)), x0).
```

### Esercizio 3.2

Data la matrice  $A \in \mathbb{R}^{n \times n}$  e il termine noto  $\mathbf{b} \in \mathbb{R}^n$ :

$$A = \begin{bmatrix} 4 & 1 & 2 & & & & \\ 1 & 4 & 1 & 2 & & & \\ 2 & 1 & 4 & \ddots & \ddots & & \\ & \ddots & & & & & \\ & & 2 & \ddots & \ddots & \ddots & 2 \\ & & & \ddots & \ddots & \ddots & 1 \\ & & & & 2 & 1 & 4 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

Si vuole risolvere il sistema lineare

$$A\mathbf{x} = \mathbf{b} \tag{9}$$

con il metodo del gradiente e del gradiente preconditionato, utilizzando come preconditionatore  $P$  la parte triangolare inferiore di  $A$ .

1. Utilizzando la funzione `richardson.m` implementata nel Laboratorio 06 (nel caso non stazionario), si risolva il sistema lineare per  $n = 5, 6, 7, \dots, 20$  con entrambi i metodi. Si utilizzi una tolleranza di  $10^{-6}$  e un numero massimo di iterazioni pari a 5000, partendo dal vettore  $\mathbf{x}_0 = [0, 0, \dots, 0, 0]^T \in \mathbb{R}^n$ .
2. Si rappresenti su un grafico in scala logaritmica l'andamento del numero di iterazioni in funzione di  $n$ .
3. Si rappresenti su un grafico il numero di condizionamento della matrice del sistema  $K_2(A)$  in funzione di  $n$ .

Utilizzando la funzione Matlab<sup>®</sup> `pcg`, si risolvano i seguenti punti:

4. Risolvere il sistema lineare  $A\mathbf{x} = \mathbf{b}$  per  $n = 5, 6, 7, \dots, 20$  ponendo  $toll = 10^{-6}$  e  $nmax = 5000$ .
5. Rappresentare su un grafico in scala logaritmica l'andamento del numero di iterazioni  $k$  in funzione di  $n$ . Confrontare sullo stesso grafico i risultati trovati utilizzando i metodi del gradiente, gradiente preconditionato e gradiente coniugato.

## 4 Esercizi aggiuntivi

### Esercizio 4.1

Si consideri il sistema lineare  $A\mathbf{x} = \mathbf{b}$ , dove  $A$  è una matrice pentadiagonale

$$A = \text{pentadiag}(1, -2, 6, -2, 1) \in \mathbb{R}^{n \times n}$$

e  $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$  per  $n \geq 1$ . Si considerino in particolare  $n = 300$  e il termine noto  $\mathbf{b}$  tale che la soluzione esatta sia  $\mathbf{x} = \mathbf{1}$ .

1. I metodi di Jacobi e Gauss-Seidel, applicati al sistema lineare  $A\mathbf{x} = \mathbf{b}$ , risultano convergenti per ogni scelta dell'iterata iniziale  $\mathbf{x}^{(0)}$ ? Si motivi la risposta definendo tutta la notazione utilizzata e riportando i principali comandi Matlab<sup>®</sup> usati.
2. Dopo aver risposto al Punto 1), quale dei due metodi iterativi utilizzereste per risolvere il sistema lineare  $A\mathbf{x} = \mathbf{b}$ ? Si motivi la risposta data.
3. Si applichi il metodo iterativo individuato al Punto 2) per risolvere il sistema lineare  $A\mathbf{x} = \mathbf{b}$  usando la funzione Matlab<sup>®</sup> .m corrispondente. Si consideri l'iterata iniziale  $\mathbf{x}^{(0)} = \mathbf{b}$ , la tolleranza sul criterio d'arresto  $tol = 10^{-6}$  e il numero massimo di iterazioni  $nmax=1000$ .

Si riportino: i) il numero di iterazioni  $N_{it}$  effettuate; ii) il valore dell'errore relativo  $e_{rel}^{(N_{it})}$ ; iii) il valore del residuo normalizzato  $r_{norm}^{(N_{it})}$  corrispondente. Si riportino i principali comandi Matlab<sup>®</sup> utilizzati.

Infine, si fornisca la *stima* dell'errore ottenuto a partire dal residuo normalizzato  $r_{norm}^{(N_{it})}$  e lo si confronti con l'errore relativo  $e_{rel}^{(N_{it})}$ .

4. Si consideri ora il metodo del *gradiente* per la soluzione del sistema lineare con la matrice  $A$ . Senza applicare esplicitamente l'algoritmo, si *stim*i l'errore in norma  $A$ , ovvero  $\|\mathbf{x}^{(k)} - \mathbf{x}\|_A$ , dopo  $k = 20$  iterazioni del metodo, sempre considerando  $\mathbf{x}^{(0)} = \mathbf{b}$ . Si giustifichi la risposta data definendo tutta la notazione utilizzata.
5. Si consideri ora il metodo del *gradiente preconditionato* per risolvere il sistema lineare associato alla matrice  $A$ . In particolare, si consideri la seguente matrice di preconditionamento simmetrica e definita positiva:

$$P = \text{tridiag}(-1, \beta, -1) \in \mathbb{R}^{300 \times 300},$$

dipendente dal parametro  $\beta > 0$ . Per quale valore del parametro  $\beta$  a scelta tra 2, 3, 4 e 5, il metodo del gradiente preconditionato converge più rapidamente alla soluzione per ogni scelta dell'iterata iniziale? Si motivi dettagliatamente la risposta data, riportando i principali comandi Matlab<sup>®</sup> usati.

6. Si consideri ora il metodo del *gradiente coniugato* (non preconditionato) per risolvere il sistema lineare  $A\mathbf{x} = \mathbf{b}$ , usando opportunamente la funzione Matlab<sup>®</sup> `pcg` con vettore iniziale  $\mathbf{x}^{(0)} = \mathbf{b}$ , tolleranza  $tol = 10^{-6}$  e un opportuno numero massimo di iterazioni. Si risolva il problema e si riportino: i) il numero di iterazioni effettuate  $N_{it}$  e ii) l'errore  $\|\mathbf{x}^{(N_{it})} - \mathbf{x}\|_A$  corrispondente. Si riportino i principali comandi Matlab<sup>®</sup> utilizzati.

## Esercizio 4.2

Si consideri un sistema lineare  $A\mathbf{x} = \mathbf{b}$ , dove  $A \in \mathbb{R}^{4 \times 4}$  è una matrice simmetrica e definita positiva tale

$$A = \begin{bmatrix} 1 & -3 & -2 & -1 \\ -3 & 10 & -3 & -2 \\ -2 & -3 & 10^2 & -3 \\ -1 & -2 & -3 & 10^3 \end{bmatrix},$$

mentre  $\mathbf{b} = (-5, 2, 92, 994)^T \in \mathbb{R}^4$ .

1. Si considerino i metodi del gradiente e gradiente coniugato, non preconditionati. Senza applicare esplicitamente i metodi, si determinino i fattori di abbattimento degli errori (in norma  $A$ ) dopo  $k = 100$  iterazioni dei metodi.
2. Si consideri il metodo del gradiente. Dopo aver posto  $\mathbf{x}^{(0)} = \mathbf{0} \in \mathbb{R}^4$ , si determinino le prime 3 direzioni di discesa del metodo, genericamente indicate come  $\mathbf{p}^{(0)}$ ,  $\mathbf{p}^{(1)}$ ,  $\mathbf{p}^{(2)} \in \mathbb{R}^4$ . Quali angoli si determinano tra le precedenti direzioni di discesa consecutive?
3. Si ripeta il punto precedente per il metodo del gradiente coniugato.
4. Si implementi il metodo del gradiente coniugato nella funzione Matlab<sup>®</sup> `conjgrad_it.m` avente intestazione

```
[xk,it] = conjgrad_it(A,b,x0,nmax,toll)
```

I parametri in ingresso richiesti dalla funzione sono la matrice  $A \in \mathbb{R}^{n \times n}$ , il termine noto  $\mathbf{b} \in \mathbb{R}^n$ , la guess iniziale  $\mathbf{x}^{(0)} \in \mathbb{R}^n$ , il numero massimo di iterazioni ammesse `nmax` e la tolleranza per il criterio d'arresto `toll`. La funzione restituisce la matrice di dimensione  $n \times (\text{it} + 1)$  contenente tutte le iterate ottenute dal metodo  $\{\mathbf{x}^{(k)}\}_{k=0}^{\text{it}}$  e il numero di iterazioni effettuate `it`. Si utilizzi il criterio del residuo normalizzato (relativo): stop all'iterata  $k$  tale che  $\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|} \leq \text{toll}$ .

5. Si utilizzi la funzione Matlab<sup>®</sup> implementata al punto precedente per applicare il metodo del gradiente coniugato con  $\mathbf{x}^{(0)} = \mathbf{0} \in \mathbb{R}^4$ , `nmax` = 1000 e `toll` =  $10^{-6}$ . Quante iterazioni vengono effettuate?
6. Dopo aver risposto al punto precedente e sapendo che la soluzione esatta del sistema lineare è  $\mathbf{x} = \mathbf{1} \in \mathbb{R}^4$ , si riportino gli errori relativi  $\frac{\|\mathbf{x} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}\|}$  in norma-2 e i residui normalizzati  $\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|}$  vs. il numero di iterazioni  $k$  (in scala semilogaritmica sulle ordinate). Si commenti il risultato ottenuto alla luce della teoria, in particolare con riferimento all'affidabilità del criterio d'arresto in considerazione.