SISTEMI LINEARI

Letizia SCUDERI

Dipartimento di Scienze Matematiche, Politecnico di Torino

letizia.scuderi@polito.it

A.A. 2022/2023

Sistemi lineari

La risoluzione di un sistema lineare è un problema che si presenta in moltissime applicazioni,

- sia esplicitamente come modello (formulazione matematica) di un fenomeno fisico;
- sia come passo intermedio o finale nella risoluzione numerica del modello in questione, rappresentato, per esempio, da equazioni differenziali.

L'applicazione stessa di metodi numerici, per esempio per l'approssimazione di funzioni, per la risoluzione di equazioni non lineari e di equazioni differenziali, può richiedere la risoluzione di sistemi lineari.

Prima di analizzare i metodi numerici per la risoluzione di sistemi lineari, si richiamano alcune nozioni riguardanti vettori e matrici.
Ricordiamo anzitutto le definizioni di norma di vettore e di matrice, tipicamente utilizzate per il calcolo dell'errore assoluto e/o relativo associato ad approssimazioni di tipo vettoriale o matriciale.

Norme di vettore e di matrice

Sia $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ un vettore colonna.

Nel seguito verranno considerate le seguenti norme di vettore:

- $||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$;
- $||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$, **norma euclidea**;
- $||\mathbf{x}||_\infty = \max_{1 \le i \le n} |x_i|$.

Comandi MATLAB

- `norm(x,1)` fornisce la norma `1` del vettore `x`;
- `norm(x,2)` oppure `norm(x)` fornisce la norma `2` del vettore `x`;
- `norm(x,inf)` fornisce la norma `inf`inito del vettore `x`.

Sia $\mathbf{A} = (a_{ij})_{i=1,\dots,m, j=1,\dots,n} \in \mathbb{R}^{m,n}$ una matrice di dimensioni $m \times n$. Nella trattazione dei sistemi lineari verranno utilizzate le seguenti norme di matrice:

- $||\mathbf{A}||_1 = \max\limits_{1 \le j \le n} \sum\limits_{i=1}^{m} |a_{ij}|$;
- $||\mathbf{A}||_2 = \sqrt{\rho(\mathbf{A}^T\mathbf{A})}$ **norma spettrale**, ove $\rho(\mathbf{B}) = \max\limits_{1 \le i \le n} |\lambda_i|$, con λ_i autovalore di $\mathbf{B}$, è detto **raggio spettrale** di $\mathbf{B}$;
- $||\mathbf{A}||_\infty = \max\limits_{1 \le i \le m} \sum\limits_{j=1}^{n} |a_{ij}|, \quad (||\mathbf{A}||_\infty = ||\mathbf{A}^T||_1)$.

Comandi MATLAB

- `norm(A,1)` fornisce la norma `1` della matrice `A`;
- `norm(A,2)` fornisce la norma `2` della matrice `A`;
- `norm(A,inf)` fornisce la norma `inf`inito della matrice `A`.

Definizione

Data una norma di matrice e una di vettore, si dice che le due norme sono **compatibili** se

$$||\mathbf{Ax}|| \leq ||\mathbf{A}||\,||\mathbf{x}||$$

per ogni matrice $\mathbf{A} \in \mathbb{R}^{n,n}$ e vettore $\mathbf{x} \in \mathbb{R}^n$.

Osservazioni

Le norme precedentemente definite 1, 2 e ∞ di matrice e di vettore sono compatibili.

Inoltre, per ogni norma di matrice compatibile con una norma di vettore, vale la seguente relazione

$$\rho(\mathbf{A}) \leq ||\mathbf{A}||$$

Definizione

Una matrice $\mathbf{A} \in \mathbb{R}^{n,n}$ si dice a **diagonale dominante per righe** se

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \quad \text{per} \quad i = 1, \ldots, n$$

Esempio

La matrice

$$\mathbf{A} = \begin{pmatrix} 4 & 1 & -1 \\ 2 & -7 & 1 \\ 3 & -2 & 9 \end{pmatrix}$$

è a diagonale dominante per righe perché

$$|4| > |1| + |-1|, \qquad |-7| > |2| + |1|, \qquad |9| > |3| + |-2|$$

Definizione

Una matrice $\mathbf{A} \in \mathbb{R}^{n,n}$ si dice a **diagonale dominante per colonne** se

$$|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}| \quad \text{per} \quad j = 1, \ldots, n$$

Esempio

La matrice

$$\mathbf{A} = \begin{pmatrix} 9 & 1 & -1 \\ 2 & -7 & 1 \\ 3 & -2 & 4 \end{pmatrix}$$

è a diagonale dominante per colonne, perché

$$|9| > |2| + |3|, \qquad |-7| > |1| + |-2|, \qquad |4| > |-1| + |1|,$$

ma non è a diagonale dominante per righe.

Definizione

Una matrice **simmetrica A** si dice **definita positiva** se

$$\mathbf{x}^T\mathbf{A}\mathbf{x} > 0 \text{ per ogni } \mathbf{x} \neq \mathbf{o};$$

oppure, equivalentemente, se

$$\lambda_i(\mathbf{A}) > 0 \text{ per ogni } i = 1, 2, ..., n$$

con $\lambda_i(\mathbf{A})$ autovalore di $\mathbf{A}$; oppure, equivalentemente, se

$$\det(\mathbf{A}_k) > 0 \text{ per ogni } k = 1, 2, ..., n$$

ove $\mathbf{A}_k$ è la matrice di ordine k formata dall'intersezione delle prime k righe e k colonne di $\mathbf{A}$.

Esempio

La matrice $\mathbf{A} = \mathbf{B}^T\mathbf{B}$, con $\mathbf{B} \in \mathbb{R}^{n,n}$ non singolare, è simmetrica e definita positiva.

Definizione

Una matrice $\mathbf{A} \in \mathbb{R}^{n,n}$ si dice **ortogonale** se le sue colonne (righe) formano un sistema ortonormale; in questo caso, denotata con $\mathbf{I}$ la matrice identità, si ha $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}$, quindi $\mathbf{A}^{-1} = \mathbf{A}^T$.

Definizione

Si definisce **matrice di permutazione**, una matrice ottenuta permutando le righe della matrice identità.

Osservazioni

Le matrici di permutazione in ogni riga e in ogni colonna hanno un solo elemento diverso da zero e uguale a 1.

Le matrici di permutazione sono matrici ortogonali.

Come mostra l'esempio che segue, se si moltiplica una matrice $\mathbf{A}$ o un vettore $\mathbf{b}$ a **sinistra** per un'opportuna matrice di permutazione $\mathbf{P}$, si possono realizzare scambi di righe in $\mathbf{A}$ oppure di componenti in $\mathbf{b}$.

Esempio

Consideriamo la seguente matrice di permutazione

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

e calcoliamo i prodotti **PA** e **Pb**, con $\mathbf{A} \in \mathbb{R}^{3,3}$ e $\mathbf{b} \in \mathbb{R}^3$. Si ha

$$\mathbf{PA} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \end{pmatrix}$$

e

$$\mathbf{Pb} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} b_3 \\ b_2 \\ b_1 \end{pmatrix}$$

Condizionamento di un sistema lineare

Si consideri il sistema lineare di n equazioni nelle n incognite $x_1, \ldots, x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \quad \ldots \quad \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

In forma matriciale si ha

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \iff \mathbf{Ax} = \mathbf{b},$$

Si assuma che la matrice $\mathbf{A}$ sia non singolare (ovvero, che abbia determinante non nullo); in tal caso, il sistema assegnato ammette una e una sola soluzione.

Prima di descrivere i metodi numerici di base per la risoluzione del suddetto sistema, si studia il condizionamento del problema.

Si denotino con $\bar{\mathbf{A}}$ e $\bar{\mathbf{b}}$ i dati perturbati e con $\bar{\mathbf{x}}$ la soluzione in **aritmetica esatta** del sistema perturbato

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

Nello studio del condizionamento si deve confrontare l'errore relativo

$$||\mathbf{x} - \bar{\mathbf{x}}||/||\mathbf{x}||$$

associato alla soluzione perturbata $\bar{\mathbf{x}}$, con gli errori relativi

$$||\mathbf{A} - \bar{\mathbf{A}}||/||\mathbf{A}|| \quad \text{e} \quad ||\mathbf{b} - \bar{\mathbf{b}}||/||\mathbf{b}||$$

associati ai dati perturbati.

Si può dimostrare il seguente risultato.

Teorema

Se $||\mathbf{A} - \bar{\mathbf{A}}|| < 1/(2||\mathbf{A}^{-1}||)$, il sistema $\bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}}$ ammette una e una sola soluzione e

$$\frac{||\mathbf{x} - \bar{\mathbf{x}}||}{||\mathbf{x}||} \leq 2\,K(\mathbf{A}) \left(\frac{||\mathbf{A} - \bar{\mathbf{A}}||}{||\mathbf{A}||} + \frac{||\mathbf{b} - \bar{\mathbf{b}}||}{||\mathbf{b}||} \right),$$

ove

$$K(\mathbf{A}) = ||\mathbf{A}||\,||\mathbf{A}^{-1}|| \geq 1$$

viene definito **numero di condizionamento** del sistema lineare $\mathbf{Ax} = \mathbf{b}$.

Pertanto,

- se $K(\mathbf{A}) \approx 1$, la matrice $\mathbf{A}$ è detta **ben condizionata** e a piccole perturbazioni sui dati corrispondono perturbazioni sulla soluzione al più dello stesso ordine di grandezza di quelle sui dati;
- se $K(\mathbf{A}) >> 1$, la matrice $\mathbf{A}$ si dice **mal condizionata** e a piccole perturbazioni sui dati *possono* corrispondere grandi perturbazioni sulla soluzione.

Osservazione 1

Se si denota con $\bar{\mathbf{b}} = \mathbf{b} + \delta\mathbf{b}$ una perturbazione del termine noto e con $\bar{\mathbf{x}} = \mathbf{x} + \delta\mathbf{x}$ la soluzione, ottenuta in aritmetica esatta, del sistema lineare con matrice $\mathbf{A}$ e termine noto $\bar{\mathbf{b}}$, è immediato dimostrare la seguente disuguaglianza:

$$\frac{||\mathbf{x} - \bar{\mathbf{x}}||}{||\mathbf{x}||} \leq ||\mathbf{A}||||\mathbf{A}^{-1}|| \frac{||\mathbf{b} - \bar{\mathbf{b}}||}{||\mathbf{b}||}$$

Infatti, poiché $\mathbf{Ax} = \mathbf{b}$, si può scrivere

$$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b} \quad \Rightarrow \quad \mathbf{A}\delta\mathbf{x} = \delta\mathbf{b} \quad \Rightarrow \quad \delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b}$$

Considerando una norma di vettore e una di matrice a essa compatibile, si ha

$$||\delta\mathbf{x}|| = ||\mathbf{A}^{-1}\delta\mathbf{b}|| \leq ||\mathbf{A}^{-1}||||\delta\mathbf{b}|| \quad \Rightarrow \quad ||\mathbf{x} - \bar{\mathbf{x}}|| \leq ||\mathbf{A}^{-1}||||\mathbf{b} - \bar{\mathbf{b}}||$$

Combinando quest'ultima diseguaglianza con la seguente

$$||\mathbf{b}|| = ||\mathbf{Ax}|| \leq ||\mathbf{A}||||\mathbf{x}|| \quad \Rightarrow \quad \frac{1}{||\mathbf{b}||} \geq \frac{1}{||\mathbf{A}||||\mathbf{x}||} \quad \Rightarrow \quad \frac{1}{||\mathbf{x}||} \leq \frac{||\mathbf{A}||}{||\mathbf{b}||},$$

si ottiene disuguaglianza desiderata.

Osservazione 2

La dimostrazione del precedente teorema, in cui si considera anche una perturbazione $\delta\mathbf{A}$ della matrice $\mathbf{A}$, è più complessa.

Osservazione 3

Denotato con $\lambda_i(\mathbf{B})$ il generico autovalore della matrice $\mathbf{B}$, si dimostra che:

$$K_2(\mathbf{A}) = \frac{\sqrt{\max_i |\lambda_i(\mathbf{A}^T\mathbf{A})|}}{\sqrt{\min_i |\lambda_i(\mathbf{A}^T\mathbf{A})|}}$$

Se $\mathbf{A}$ è simmetrica, allora

$$K_2(\mathbf{A}) = \frac{|\max_i \lambda_i(\mathbf{A})|}{|\min_i \lambda_i(\mathbf{A})|}$$

Se $\mathbf{A}$ è simmetrica e definita positiva, allora

$$K_2(\mathbf{A}) = \frac{\max_i \lambda_i(\mathbf{A})}{\min_i \lambda_i(\mathbf{A})}$$

Esempi classici di sistemi lineari mal condizionati sono quello associato alla **matrice di Hilbert**, simmetrica e definita positiva,

$$\mathbf{H}_n = \begin{pmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{pmatrix}$$

e quello alla **matrice di Vandermonde**

$$\mathbf{V}_n = \begin{pmatrix} x_1^n & \cdots & x_1 & 1 \\ x_2^n & \cdots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n+1}^n & \cdots & x_{n+1} & 1 \end{pmatrix}$$

non singolare se $x_i \neq x_j$ per $i \neq j$.

Comandi MATLAB

- `cond(A,1)` fornisce il numero di condizionamento in norma `1` del sistema $\mathbf{Ax} = \mathbf{b}$;
- `cond(A,2)` fornisce il numero di condizionamento in norma `2` del sistema $\mathbf{Ax} = \mathbf{b}$;
- `cond(A,inf)` fornisce il numero di condizionamento in norma `inf`inito del sistema $\mathbf{Ax} = \mathbf{b}$;
- `hilb(n)` genera la matrice di Hilbert $\mathbf{H}_n$ di ordine `n`;
- `vander(x)` genera la matrice di Vandermonde $\mathbf{V}_n$ associata al vettore `x` con componenti $x_1, \ldots, x_{n+1}$.

Metodi numerici

Si descrivono ora alcuni metodi numerici e gli algoritmi, che implementano i suddetti metodi, per la risoluzione di sistemi lineari di ordine n.

Per ciascun algoritmo verrà fornito il **costo computazionale**, ovvero il numero di operazioni aritmetiche [1] che esso richiede per la sua esecuzione.

Nel caso dei sistemi lineari il costo è legato alla dimensione n del sistema e, in generale, viene quantificato il numero di operazioni per valori di n grandi.

[1] Da qui in avanti, con il termine *operazione aritmetica (flop)* si intenderà la coppia $(+, \times)$ somma-prodotto.

Metodi diretti

I metodi numerici per la risoluzione di sistemi lineari vengono suddivisi in due classi: **metodi diretti** e **metodi iterativi**.
Le principali proprietà dei metodi diretti per la risoluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$ sono:

1. $\mathbf{x}$ viene determinata mediante un numero finito di passi (al più $n - 1$ se n è l'ordine del sistema);
2. $\mathbf{x}$ in aritmetica con precisione infinita di calcolo viene determinata in maniera esatta; in aritmetica con precisione finita di calcolo viene determinata con una precisione la cui entità non dipende dalle richieste dell'utente;
3. i metodi diretti modificano la matrice dei coefficienti $\mathbf{A}$;
4. i metodi diretti sono efficienti per matrici dense e di piccole-medie dimensioni.

La scelta di un metodo diretto ed efficiente per la risoluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$ dipende dalla struttura e dalle proprietà della matrice $\mathbf{A}$.
Se la matrice dei coefficienti $\mathbf{A}$ è **diagonale**, con $a_{kk} \neq 0,\ k = 1, \ldots, n$, la soluzione del sistema lineare

$$\begin{cases} a_{11}x_1 & & & & & = b_1 \\ & a_{22}x_2 & & & & = b_2 \\ & & a_{33}x_3 & & & = b_3 \\ & & & \ddots & & \vdots \\ & & & & a_{nn}x_n & = b_n \end{cases}$$

si ricava immediatamente mediante le seguenti n divisioni:

$$\begin{aligned} &\text{for } k = 1, \ldots, n \\ &\qquad x_k = b_k / a_{kk} \\ &\text{end} \end{aligned}$$

Se la matrice dei coefficienti **A** è **triangolare superiore**, con $a_{kk} \neq 0,\ k = 1, \ldots, n$, la soluzione del sistema lineare

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + & \ldots & & + a_{1n}x_n & = b_1 \\ & \ddots & & \ddots & \vdots \\ & a_{n-1n-1}x_{n-1} & & + a_{n-1n}x_n & = b_{n-1} \\ & & & a_{nn}x_n & = b_n \end{cases}$$

si ottiene ricavando l'incognita x_n dall'ultima equazione, x_{n-1} dalla penultima equazione (dopo aver sostituito in essa il valore di x_n),..., x_1 dalla prima (dopo aver sostituito in essa i valori di $x_2, \ldots, x_n$).
Lo schema di calcolo è il seguente:

$$\begin{aligned} & x_n = b_n / a_{nn} \\ & \text{for } k = n-1, \ldots, 1 \\ & \qquad x_k = (b_k - \textstyle\sum_{j=k+1}^{n} a_{kj}x_j)/a_{kk} \\ & \text{end} \end{aligned}$$

Tale procedura è nota con il nome di **metodo di sostituzione all'indietro**. Il costo computazionale in termini di operazioni aritmetiche è essenzialmente $n^2/2$ per n grande.

Se la matrice dei coefficienti **A** è **triangolare inferiore**, con $a_{kk} \neq 0,\ k = 1, \ldots, n$, la soluzione del sistema lineare

$$\begin{cases} a_{11}x_1 & & & & = b_1 \\ a_{21}x_1 & +a_{22}x_2 & & & = b_2 \\ & \ddots & \ddots & & \vdots \\ a_{n1}x_1 & +a_{n2}x_2+ & \ldots & +a_{nn}x_n & = b_n \end{cases}$$

si ottiene ricavando l'incognita x_1 dalla prima equazione, x_2 dalla seconda equazione (dopo aver sostituito in essa il valore di x_1),..., x_n dall'ultima (dopo aver sostituito in essa i valori di $x_1, \ldots, x_{n-1}$).
Lo schema di calcolo è il seguente:

$$\begin{aligned} & x_1 = b_1/a_{11} \\ & \text{for } k = 2, \ldots, n \\ & \qquad x_k = (b_k - \textstyle\sum_{j=1}^{k-1} a_{kj}x_j)/a_{kk} \\ & \text{end} \end{aligned}$$

Tale procedura è nota con il nome di **metodo di sostituzione in avanti**.
Il costo computazionale in termini di operazioni aritmetiche è essenzialmente $n^2/2$ per n grande.

Se la matrice dei coefficienti $\mathbf{A}$ non ha una struttura particolare, il metodo diretto più noto e più utilizzato è senza dubbio il metodo delle eliminazioni di Gauss.

Si ricorda che il metodo delle eliminazioni di Gauss consta essenzialmente di due fasi:

1. trasformazione, in $n-1$ passi, del sistema assegnato $\mathbf{Ax} = \mathbf{b}$ nel sistema $\mathbf{Ux} = \bar{\mathbf{b}}$ equivalente a quello assegnato (ovvero che ammette la stessa soluzione $\mathbf{x}$), con $\mathbf{U}$ matrice triangolare superiore;
2. risoluzione del sistema $\mathbf{Ux} = \bar{\mathbf{b}}$ mediante la tecnica di sostituzione all'indietro.

Il costo computazionale del metodo delle eliminazioni di Gauss in termini di operazioni aritmetiche è essenzialmente $n^3/3$ per n grande.

Per garantire una **migliore stabilità numerica** dell'algoritmo di Gauss e, quindi, per ridurre l'amplificazione degli errori di arrotondamento che vengono generati nel corso delle trasformazioni, a ogni passo conviene operare un preliminare scambio di equazioni prima di procedere con le trasformazioni. Tale strategia è nota sotto il nome di **pivoting parziale**.
Il pivoting è **superfluo** quando:

- **A** è a diagonale dominante per colonne;
- **A** è simmetrica e definita positiva.

Comando MATLAB

`x = A\b` calcola la soluzione `x` di `Ax=b` con il metodo delle eliminazioni di Gauss con pivoting parziale. A seconda delle caratteristiche della matrice `A` (diagonale, triangolare, simmetrica e definita positiva,...), il comando richiama un algoritmo specifico, ottimizzato rispetto al numero delle operazioni aritmetiche.

Fattorizzazioni di matrici

Il metodo delle eliminazioni di Gauss con pivoting parziale realizza la seguente fattorizzazione della matrice **A**:

$$\mathbf{PA} = \mathbf{LU}$$

ove **U** è una matrice triangolare superiore, **L** è triangolare inferiore con diagonale unitaria e **P** è una matrice di permutazione, definita dagli scambi richiesti dalla strategia del pivoting.

Il costo computazionale della fattorizzazione $\mathbf{PA} = \mathbf{LU}$ della matrice **A** è uguale a quello del metodo delle eliminazioni di Gauss e vale all'incirca $n^3/3$ per n grande.

Comando MATLAB

`[L,U,P] = lu(A)` calcola i fattori `L`, `U`, e `P` della fattorizzazione $\mathbf{PA} = \mathbf{LU}$ di `A`.

Si osservi che per la risoluzione di un sistema lineare $\mathbf{Ax} = \mathbf{b}$ si può procedere utilizzando la fattorizzazione $\mathbf{PA} = \mathbf{LU}$ nel seguente modo:

$$\mathbf{Ax} = \mathbf{b} \Longrightarrow \mathbf{PAx} = \mathbf{Pb} \Longrightarrow \mathbf{L}\underbrace{\mathbf{Ux}}_{\mathbf{y}} = \mathbf{Pb}$$

da cui segue la risoluzione dei due seguenti sistemi triangolari

$$\begin{cases} \mathbf{Ly} = \mathbf{Pb} & \Rightarrow \mathbf{y} \\ \mathbf{Ux} = \mathbf{y} & \Rightarrow \mathbf{x} \end{cases}$$

Se i fattori $\mathbf{L}, \mathbf{U}, \mathbf{P}$ della fattorizzazione $\mathbf{PA} = \mathbf{LU}$ non si devono calcolare perché già noti, il costo computazionale della risoluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$ è soltanto n^2; altrimenti, il costo è pari a quello della fattorizzazione $\mathbf{PA} = \mathbf{LU}$, cioè $n^3/3$ (essendo, per n grande, il costo della risoluzione dei due sistemi lineari triangolari trascurabile rispetto al costo della fattorizzazione).

Per una matrice **A simmetrica e definita positiva** esiste ed è unica la fattorizzazione

$$\mathbf{A} = \mathbf{R}^T \mathbf{R}$$

ove $\mathbf{R}$ è una matrice triangolare superiore con elementi positivi sulla diagonale principale.
Tale fattorizzazione è detta **fattorizzazione di Choleski** e viene dimostrata nel seguente teorema.

Teorema

Se $\mathbf{A}$ è una matrice simmetrica e definita positiva allora esiste un'unica matrice triangolare inferiore $\mathbf{L}_1 (= \mathbf{R}^T)$ con elementi diagonali positivi tale che

$$\mathbf{A} = \mathbf{L}_1 \mathbf{L}_1^T$$

Dimostrazione

È possibile dimostrare che se i determinanti delle matrici $\mathbf{A}_k$, formate dall'intersezione delle prime k righe e k colonne di una qualsiasi matrice $\mathbf{A}$, sono diversi da zero per $k = 1, ..., n-1$, allora esiste ed è unica la fattorizzazione LU di $\mathbf{A}$.

Poiché per ipotesi $\mathbf{A}$ è simmetrica e definita positiva, e dunque verifica $\det(\mathbf{A}_k) > 0$ per $k = 1, ..., n$, allora possiamo senz'altro affermare che esiste un'unica matrice $\mathbf{L}$ triangolare inferiore con diagonale unitaria e un'unica matrice $\mathbf{U}$ triangolare superiore, tali che

$$\mathbf{A} = \mathbf{LU}$$

Se denotiamo con $\mathbf{D}$ la matrice diagonale i cui elementi principali sono quelli di $\mathbf{U}$ e con $\mathbf{U}_1$ la matrice triangolare superiore con diagonale unitaria tale che $\mathbf{DU}_1 = \mathbf{U}$, allora possiamo scrivere

$$\mathbf{A} = \mathbf{LDU}_1$$

... continua dimostrazione

Poiché $\mathbf{A} = \mathbf{LU}$ è simmetrica ed $\mathbf{L}$ è unica allora $\mathbf{U}_1 = \mathbf{L}^T$; infatti, da

$$\mathbf{A} = \mathbf{LDU}_1 \quad \text{segue} \quad \mathbf{A} = \mathbf{A}^T = \mathbf{U}_1^T \mathbf{DL}^T,$$

con $\mathbf{U}_1^T$ triangolare inferiore con diagonale unitaria. Per l'unicità di $\mathbf{L}$, si ha allora $\mathbf{L} = \mathbf{U}_1^T$ e $\mathbf{DU}_1 = \mathbf{U} = \mathbf{DL}^T$. Pertanto, risulta univocamente

$$\mathbf{A} = \mathbf{LDL}^T$$

Inoltre, poiché $\mathbf{A}$ è definita positiva, gli elementi diagonali della matrice $\mathbf{D}$ sono tutti positivi in quanto anche $\mathbf{D}$ è definita positiva; infatti, per $\mathbf{x} \neq \mathbf{o}$, si ha

$$0 < \mathbf{x}^T \mathbf{Ax} = \mathbf{x}^T \mathbf{LDL}^T \mathbf{x} = (\mathbf{L}^T \mathbf{x})^T \mathbf{D} (\mathbf{L}^T \mathbf{x}) = \mathbf{y}^T \mathbf{Dy}$$

con $\mathbf{y} = \mathbf{L}^T \mathbf{x} \neq \mathbf{o}$, essendo $\mathbf{x} \neq \mathbf{o}$ e $\mathbf{L}$ non singolare.

... continua dimostrazione

Pertanto, denotando con $\mathbf{D}^{\frac{1}{2}}$ la matrice i cui elementi diagonali sono definiti come la radice quadrata degli elementi diagonali della matrice $\mathbf{D}$, allora $\mathbf{D} = \mathbf{D}^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}$ e

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T = \mathbf{L}\mathbf{D}^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\mathbf{L}^T = \mathbf{L}_1\mathbf{L}_1^T$$

avendo posto $\mathbf{L}_1 = \mathbf{L}\mathbf{D}^{\frac{1}{2}}$. Per costruzione $\mathbf{L}_1$ è triangolare inferiore con elementi diagonali positivi e, per l'unicità di $\mathbf{L}$ e di $\mathbf{D}$, $\mathbf{L}_1$ è unica.

Il calcolo del fattore di Choleski $\mathbf{R}$ si ottiene mediante un algoritmo il cui costo computazionale è essenzialmente $n^3/6$ per n grande.

Comando MATLAB

`R = chol(A)` calcola il fattore `R` triangolare superiore della fattorizzazione di Choleski $\mathbf{A} = \mathbf{R}^T\mathbf{R}$, della matrice simmetrica e definita positiva `A`.

Per la risoluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$, con $\mathbf{A}$ simmetrica e definita positiva, nota la fattorizzazione di Choleski $\mathbf{A} = \mathbf{R}^T\mathbf{R}$, un algoritmo efficiente è il seguente:

$$\mathbf{Ax} = \mathbf{b} \Longrightarrow \mathbf{R}^T \underbrace{\mathbf{Rx}}_{\mathbf{y}} = \mathbf{b} \Longrightarrow \begin{cases} \mathbf{R}^T\mathbf{y} = \mathbf{b} & \Rightarrow \mathbf{y} \\ \mathbf{Rx} = \mathbf{y} & \Rightarrow \mathbf{x} \end{cases}$$

Un'altra fattorizzazione per una generica matrice $\mathbf{A}$ di dimensioni $m \times n$ è la **fattorizzazione QR**. In particolare, si può dimostrare che esiste una matrice ortogonale $\mathbf{Q}$ di dimensioni $m \times m$ tale che

$$\mathbf{A} = \mathbf{QR}$$

dove gli elementi r_{ij} della matrice $\mathbf{R}$ sono nulli per $i > j$. Se $m = n$, $\mathbf{R}$ è una matrice triangolare superiore.

Comando MATLAB

`[Q,R] = qr(A)` calcola i fattori `Q` e `R` della fattorizzazione $\mathbf{A} = \mathbf{QR}$ di `A`.

Questa fattorizzazione consente di costruire un algoritmo stabile per la risoluzione del sistema $\mathbf{Ax} = \mathbf{b}$, con $\mathbf{A}$ di ordine n e non singolare:

$$\mathbf{Ax} = \mathbf{b} \Longrightarrow \mathbf{QRx} = \mathbf{b} \Longrightarrow \mathbf{Rx} = \mathbf{Q}^T\mathbf{b}$$

Tuttavia il calcolo delle matrici $\mathbf{Q}$ ed $\mathbf{R}$ richiede $2n^3/3$ operazioni aritmetiche mentre il metodo di Gauss ne richiede solamente $n^3/3$.

Un'altra importante fattorizzazione di una generica matrice $\mathbf{A}$ di dimensioni $m \times n$ è la **decomposizione ai valori singolari (SVD)**.
In particolare, si può dimostrare che esistono due matrici ortogonali $\mathbf{U}$ di dimensioni $m \times m$ e $\mathbf{V}$ di dimensioni $n \times n$ tali che

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

dove $\mathbf{S}$ è diagonale, ovvero

$$s_{ij} = \begin{cases} \sigma_i & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, $p = \min\{m, n\}$. Gli elementi diagonali della matrice $\mathbf{S}$ sono detti **valori singolari** della matrice $\mathbf{A}$. Essi coincidono con la radice quadrata degli autovalori (tutti non negativi) della matrice $\mathbf{A}^T\mathbf{A}$.

Comando MATLAB

`[U,S,V] = svd(A)` calcola i fattori `U`, `S`, e `V` della fattorizzazione $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ di `A`.

Si può dimostrare che il rango della matrice $\mathbf{A}$ coincide con il numero dei valori singolari non nulli.

Anche questa fattorizzazione consente di costruire un algoritmo stabile per la risoluzione del sistema $\mathbf{Ax} = \mathbf{b}$, con $\mathbf{A}$ di ordine n e non singolare:

$$\mathbf{Ax} = \mathbf{b} \Longrightarrow \mathbf{USV}^T\mathbf{x} = \mathbf{b} \Longrightarrow \mathbf{S}\underbrace{\mathbf{V}^T\mathbf{x}}_{\mathbf{y}} = \mathbf{U}^T\mathbf{b} \Longrightarrow \begin{cases} \mathbf{Sy} = \mathbf{U}^T\mathbf{b} \Rightarrow \mathbf{y} \\ \mathbf{x} = \mathbf{Vy} \end{cases}$$

Tuttavia il calcolo delle matrici $\mathbf{U}$, $\mathbf{V}$ ed $\mathbf{S}$ richiede $32n^3/3$ operazioni aritmetiche mentre il metodo di Gauss ne richiede solamente $n^3/3$.

Metodi iterativi

I metodi diretti non sono generalmente adeguati per sistemi di dimensioni elevate. Tali sistemi, che scaturiscono nella maggior parte dalle applicazioni, generalmente ammettono una matrice dei coefficienti **sparsa**. Una matrice di dimensioni $n \times n$ si ritiene sparsa quando il numero degli elementi diversi dallo zero è di ordine n.

La presenza di sparsità in una matrice rappresenta a priori un vantaggio, dal momento che memorizzando solo gli elementi non nulli, si possono ottenere notevoli risparmi di memoria e di operazioni.

MATLAB per esempio consente di memorizzare le matrici in un formato sparso, secondo cui si prendono in considerazione soltanto gli elementi non nulli.

Comandi MATLAB

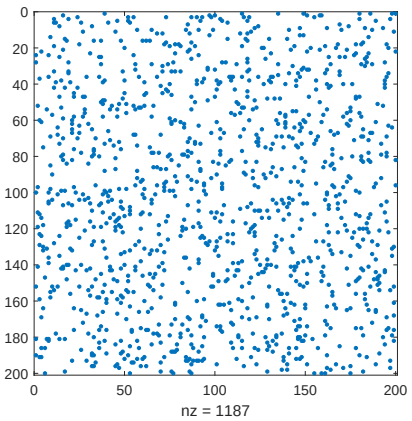
- `sparse(A)` consente di memorizzare solo gli elementi non nulli di `A`;
- `spy(A)` consente di rappresentare graficamente l'insieme degli elementi non nulli di `A`, ovvero il cosiddetto *pattern*;
- `nnz(A)` fornisce il numero degli elementi non nulli di `A`;
- `spdiags(A,d,m,n)` crea una matrice sparsa `m` x `n`, i cui elementi sulle diagonali, specificate dalle componenti del vettore `d`, coincidono con gli elementi delle colonne di `A`.

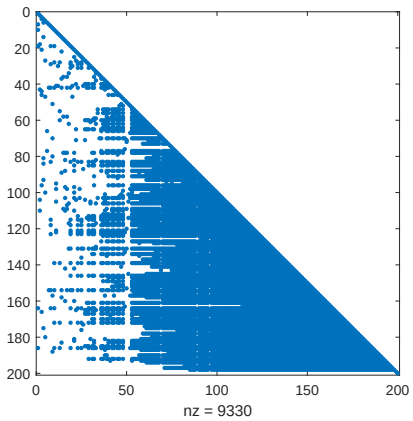
Quando una matrice è memorizzata in formato sparso allora MATLAB richiama automaticamente tecniche opportune che consentono di ottimizzare i tempi di calcolo e l'occupazione di memoria.

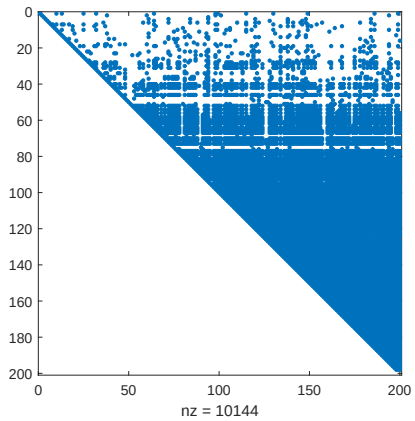
Questo vantaggio è tuttavia utilizzabile nel caso di matrici a banda. Nel caso invece di una matrice sparsa senza una particolare struttura, il procedimento di eliminazione può introdurre nuovi elementi diversi dallo zero. Si può verificare durante l'eliminazione un fenomeno di **fill-in** (riempimento).

Esempio

Sia **A** una matrice sparsa di ordine 200×200, con un numero di elementi diversi da zero pari a $nz = 1187$.
Se si calcola la fattorizzazione $PA = LU$ della matrice **A**, le matrici **L** e **U** non sono sparse a causa del fenomeno di fill-in; per entrambe si ha $nz \approx 10000$.







In questi casi si pone il problema di riordinare opportunamente la matrice, al fine di ridurre al minimo, per un fissato metodo, il fill-in. Naturalmente, nel caso generale, la ricerca di un tale ordinamento non è un problema di facile soluzione (ricordiamo, a tale scopo, che l'ordinamento può influenzare la stabilità del metodo).

Una interessante alternativa ai metodi diretti, quando la matrice è sparsa e di grandi dimensioni, è fornita dai metodi iterativi e dai metodi di tipo gradiente.

Per descrivere il generico metodo iterativo per la risoluzione di un sistema lineare, partiamo dal seguente sistema

$$\mathbf{Ax} = \mathbf{b} \iff \begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \quad \ldots \quad \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

e supponiamo che $a_{ii} \neq 0$ per ogni $i = 1, \ldots, n$. Tale condizione, essendo $\mathbf{A}$ non singolare, si può realizzare mediante opportuni scambi di equazioni.

A partire da un arbitrario vettore iniziale $\mathbf{x}^{(0)}$, con un metodo iterativo si determina una sequenza di vettori $\left\{\mathbf{x}^{(k)}\right\}_{k=1,2,\ldots}$ che, sotto opportune condizioni, converge alla soluzione esatta $\mathbf{x}^*$ del sistema:

$$\mathbf{x}^{(0)} \Rightarrow \left\{\mathbf{x}^{(k)}\right\}_{k=1,2,\ldots} : \lim_{k\to\infty} (\mathbf{x}^* - \mathbf{x}^{(k)}) = \mathbf{o} \text{ (vettore nullo)}$$

Metodi di Jacobi e Gauss Seidel

Definiamo dapprima due classici metodi iterativi. A tale scopo ricaviamo dall'i-esima equazione per $i = 1, \ldots, n$ l'incognita x_i:

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}, \quad a_{ii} \neq 0$$

e, a partire dal vettore $\mathbf{x}^{(0)}$, generiamo la successione di vettori $\left\{\mathbf{x}^{(k)}\right\}_{k=1,2,\ldots}$ la cui i-esima componente, per $i = 1, ..., n$, è definita dall'espressione:

$$x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}, \ k = 1, 2, \ldots$$

Tale formula iterativa definisce il cosiddetto **metodo di Jacobi**.

Ponendo invece

$$x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}, \ k = 1, 2, \ldots$$

otteniamo il cosiddetto **metodo di Gauss-Seidel**.

ESEMPIO

$$\begin{cases} 3x_1 - x_2 = 2 \\ x_1 + 2x_2 = 3 \end{cases} \Longrightarrow \begin{cases} x_1 = \frac{2+x_2}{3} \\ x_2 = \frac{3-x_1}{2} \end{cases} \qquad \mathbf{x}^* = (1,1)^T$$

METODO DI JACOBI: $\begin{cases} x_1^{(k)} = \frac{2+x_2^{(k-1)}}{3} \\ x_2^{(k)} = \frac{3-x_1^{(k-1)}}{2} \end{cases} \qquad \mathbf{x}^{(0)} = (0,0)^T$

$\mathbf{x}^{(0)}$	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$
0	2/3	7/6	19/18	...
0	3/2	7/6	11/12	...

$$\frac{||\mathbf{x}^* - \mathbf{x}^{(3)}||_\infty}{||\mathbf{x}^*||_\infty} \approx 0.8 \cdot 10^{-1}$$

... continua

METODO DI GAUSS-SEIDEL: $\begin{cases} x_1^{(k)} = \frac{2+x_2^{(k-1)}}{3} \\ x_2^{(k)} = \frac{3-x_1^{(k)}}{2} \end{cases}$ $\mathbf{x}^{(0)} = (0,0)^T$

$\mathbf{x}^{(0)}$	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$
0	2/3	19/18	107/108	...
0	7/6	35/36	217/216	...

$$\frac{||\mathbf{x}^* - \mathbf{x}^{(3)}||_\infty}{||\mathbf{x}^*||_\infty} \approx 0.9 \cdot 10^{-2}$$

I metodi di Jacobi e di Gauss-Seidel fanno parte di quella classe più generale di procedimenti iterativi che si deducono nel seguente modo:

$$\mathbf{Ax} = \mathbf{b} \Longleftrightarrow (\mathbf{D} + \mathbf{C})\mathbf{x} = \mathbf{b} \Longleftrightarrow \mathbf{Dx} = -\mathbf{Cx} + \mathbf{b}$$
$$\Longrightarrow \mathbf{Dx}^{(k)} = -\mathbf{Cx}^{(k-1)} + \mathbf{b}, \ k = 1, 2, \ldots$$

ove, a ogni iterazione k, $\mathbf{x}^{(k)}$ è la soluzione di un sistema lineare con matrice dei coefficienti $\mathbf{D}$ e termine noto $\mathbf{b} - \mathbf{Cx}^{(k-1)}$.

Pertanto la matrice $\mathbf{D}$ deve essere:

1. non singolare, ovvero tale da garantire l'esistenza e l'unicità della soluzione $\mathbf{x}^{(k)}$ a ogni passo k;
2. di forma semplice (triangolare, diagonale), ovvero tale che la soluzione $\mathbf{x}^{(k)}$ del sistema possa ottenersi, a ogni passo k, con un algoritmo relativamente semplice e poco costoso;
3. tale da garantire la convergenza, per $k \to \infty$, di $\mathbf{x}^{(k)}$ alla soluzione $\mathbf{x}^*$ del sistema originario.

Osserviamo che, nel caso del metodo di Jacobi

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

le matrici $\mathbf{D}$ e $\mathbf{C}$ dello sdoppiamento $\mathbf{A} = \mathbf{D} + \mathbf{C}$ sono così definite

$$\mathbf{D} = \begin{pmatrix} a_{11} & 0 & \ldots & 0 \\ 0 & a_{22} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & a_{nn} \end{pmatrix}, \quad \mathbf{C} = \mathbf{A} - \mathbf{D};$$

nel caso del metodo di Gauss-Seidel

$$\sum_{j=1}^{i} a_{ij} x_j^{(k)} = b_i - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}, \quad i = 1, \ldots, n$$

si ha

$$\mathbf{D} = \begin{pmatrix} a_{11} & 0 & \ldots & 0 \\ a_{21} & a_{22} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}, \quad \mathbf{C} = \mathbf{A} - \mathbf{D}$$

Quando si implementa un procedimento iterativo occorre definire dei criteri d'arresto.

Un criterio d'arresto consiste senz'altro nel fissare un numero massimo *kmax* di iterazioni.

Un altro criterio d'arresto riguarda la bontà dell'approssimazione $\mathbf{x}^{(k)}$ e consiste nel fissare una tolleranza relativa $tol_r (\geq \varepsilon_m)$ o assoluta tol_a e nell'arrestare il processo all'iterazione $\bar{k}$ se $\mathbf{x}^{(\bar{k})}$ soddisfa la seguente disuguaglianza

$$||\mathbf{x}^{(\bar{k}+1)} - \mathbf{x}^{(\bar{k})}|| \leq tol_r ||\mathbf{x}^{(\bar{k}+1)}||$$

oppure

$$||\mathbf{x}^{(\bar{k}+1)} - \mathbf{x}^{(\bar{k})}|| \leq tol_a$$

Osserviamo che tale criterio d'arresto ha senso quando il metodo converge "rapidamente".

Function MATLAB

```
function [x,k,ier] = Jacobi(A,b,x0,tol,kmax)
if prod(diag(A)) == 0
   x = []; k = []; ier = -1;
   return
end
D = diag(diag(A));
C = A-D;
for k = 1:kmax
   x = D\(b-C*x0);
   if norm(x-x0) <= tol*norm(x);
      ier = 1;
      return
   end
   x0 = x;
end
ier = 0;
```

Function MATLAB

```
function [x,k,ier] = Gauss_Seidel(A,b,x0,tol,kmax)
if prod(diag(A)) == 0
   x = []; k = []; ier = -1;
   return
end
D = tril(A);
C = A-D;
for k = 1:kmax
   x = D\(b-C*x0);
   if norm(x-x0) <= tol*norm(x);
      ier = 1;
      return
   end
   x0 = x;
end
ier = 0;
```

Convergenza dei metodi iterativi

Per lo studio della convergenza dei metodi iterativi, definiamo il vettore errore

$$\mathbf{e}^{(k)} := \mathbf{x}^* - \mathbf{x}^{(k)}$$

e sottraiamo membro a membro le seguenti due equazioni:

$$\begin{aligned} \mathbf{D}\mathbf{x}^* &= \mathbf{b} - \mathbf{C}\mathbf{x}^* \\ \mathbf{D}\mathbf{x}^{(k)} &= \mathbf{b} - \mathbf{C}\mathbf{x}^{(k-1)} \\ \hline \\ \mathbf{D}\mathbf{e}^{(k)} &= -\mathbf{C}\mathbf{e}^{(k-1)} \end{aligned}$$

Abbiamo pertanto

$$\mathbf{e}^{(k)} = -\mathbf{D}^{-1}\mathbf{C}\mathbf{e}^{(k-1)} = \mathbf{B}\mathbf{e}^{(k-1)}$$

ove abbiamo posto

$$\mathbf{B} := -\mathbf{D}^{-1}\mathbf{C} = -\mathbf{D}^{-1}(\mathbf{A} - \mathbf{D}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$$

La matrice $\mathbf{B}$ è detta **matrice di iterazione**.

Tenendo conto che l'uguaglianza $\mathbf{e}^{(k)} = \mathbf{B}\mathbf{e}^{(k-1)}$ vale per ogni $k = 1, 2, \ldots$, si ha

$$\mathbf{e}^{(k)} = \mathbf{B}\mathbf{e}^{(k-1)} = \mathbf{B}^2\mathbf{e}^{(k-2)} = \ldots = \mathbf{B}^k\mathbf{e}^{(0)}$$

da cui deduciamo che

$$\lim_{k\to\infty} \mathbf{e}^{(k)} = \mathbf{o} \text{ (vettore nullo) } \forall \mathbf{x}^{(0)} \iff \lim_{k\to\infty} \mathbf{B}^k = \mathbf{O} \text{ (matrice nulla)}$$

Poiché si dimostra che

$$\lim_{k\to\infty} \mathbf{B}^k = \mathbf{O} \iff \rho(\mathbf{B}) < 1$$

ove $\rho(\mathbf{B})$ è il raggio spettrale della matrice, cioè $\rho(\mathbf{B}) = \max_{1\le i\le n} |\lambda_i|$ con λ_i autovalore di $\mathbf{B}$, allora possiamo dedurre

$$\lim_{k\to\infty} \mathbf{e}^{(k)} = \mathbf{o}\ \forall \mathbf{x}^{(0)} \iff \rho(\mathbf{B}) < 1$$

Convergenza dei metodi di Jacobi e Gauss-Seidel

Ricordando che per le norme 1,2,∞ sussiste la proprietà $\rho(\mathbf{A}) \leq ||\mathbf{A}||$, possiamo senz'altro affermare che, se per almeno una delle suddette norme vale $||\mathbf{B}|| < 1$, allora $\lim_{k\to\infty} \mathbf{e}^{(k)} = \mathbf{o} \quad \forall \mathbf{x}^{(0)}$

Sia $\mathbf{D} = \texttt{diag(diag(A))}$ (metodo di Jacobi) oppure $\mathbf{D} = \texttt{tril(A)}$ (metodo di Gauss-Seidel) e $\mathbf{B} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$. Si può dimostrare che

- se $\mathbf{A}$ è a diagonale dominante per righe allora $||\mathbf{B}||_\infty < 1$
- se $\mathbf{A}$ è a diagonale dominante per colonne allora $||\mathbf{B}||_1 < 1$

Vale pertanto il seguente teorema.

Teorema

Se $\mathbf{A}$ è a diagonale dominante per righe o per colonne, allora i metodi di Jacobi e di Gauss-Seidel, applicati al sistema lineare $\mathbf{Ax} = \mathbf{b}$, convergono qualunque sia il vettore iniziale.

Si può inoltre dimostrare il seguente teorema.

Teorema

Se $\mathbf{A}$ è simmetrica e definita positiva, allora il metodo di Gauss-Seidel converge.

Osservazioni

1. La convergenza del metodo di Gauss-Seidel non implica la convergenza del metodo di Jacobi e viceversa. Se entrambi convergono, in generale il metodo di Gauss-Seidel converge più rapidamente.
2. La rapidità di convergenza del metodo iterativo $\mathbf{D}\mathbf{x}^{(k)} = -\mathbf{C}\mathbf{x}^{(k-1)} + \mathbf{b},\ k = 1, 2, \ldots$ dipende dal raggio spettrale $\rho(\mathbf{B})$ della matrice di iterazione $\mathbf{B} = -\mathbf{D}^{-1}\mathbf{C} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$: quanto più esso è piccolo, tanto più è rapida la convergenza del corrispondente metodo.

Metodi di rilassamento

Definiamo ora un'altra classe di metodi iterativi detti metodi di rilassamento. A partire dalla formula che definisce il metodo di Gauss-Seidel

$$x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}, \ k = 1, 2, \dots$$

sottraiamo $x_i^{(k-1)}$, per $i = 1, 2, ..., n$, da ambo i membri e otteniamo:

$$r_i^{(k-1)} = x_i^{(k)} - x_i^{(k-1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}, \ k = 1, 2, \dots$$

dove $r_i^{(k-1)}$ rappresenta la correzione da apportare a $x_i^{(k-1)}$ per ottenere $x_i^{(k)}$:

$$x_i^{(k)} = x_i^{(k-1)} + r_i^{(k-1)}, \ k = 1, 2, \dots, \ i = 1, .., n$$

Quest'ultima relazione definisce ancora il metodo di Gauss-Seidel. Essa tuttavia suggerisce l'introduzione di un parametro $\omega \neq 0$ al fine di migliorare la correzione su $x_i^{(k-1)}$:

$$x_i^{(k)} = x_i^{(k-1)} + \omega r_i^{(k-1)}, \ k = 1, 2, \ldots, \ i = 1, .., n$$

Tale relazione definisce il **metodo di rilassamento**: il metodo è detto di **sottorilassamento** se $\omega < 1$, mentre è detto di **sovrarilassamento** se $\omega > 1$ (se $\omega = 1$ si ha il metodo di Gauss-Seidel). Quest'ultimo è detto metodo **SOR**, dalle iniziali di Successive Over Relaxation.

Per riscrivere il metodo in forma matriciale, introduciamo le seguenti matrici che, per comodità, definiamo con i seguenti comandi MATLAB :

`D=diag(diag(A))`, `L=tril(A,-1)`, `U=triu(A,1)`.

Si ha pertanto $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ e

$$\mathbf{D}\mathbf{x}^{(k)} = \mathbf{b} - \mathbf{L}\mathbf{x}^{(k)} - \mathbf{U}\mathbf{x}^{(k-1)}$$

Sottraendo $\mathbf{D}\mathbf{x}^{(k-1)}$ da ambo i membri, si ha

$$\underbrace{\mathbf{D}\mathbf{x}^{(k)} - \mathbf{D}\mathbf{x}^{(k-1)}}_{\mathbf{D}\mathbf{r}^{(k-1)}} = \mathbf{b} - \mathbf{L}\mathbf{x}^{(k)} - (\mathbf{D} + \mathbf{U})\mathbf{x}^{(k-1)}$$

da cui

$$\mathbf{D}\mathbf{x}^{(k)} = \mathbf{D}\mathbf{x}^{(k-1)} + \omega\mathbf{D}\mathbf{r}^{(k-1)} = \mathbf{D}\mathbf{x}^{(k-1)} + \omega[\mathbf{b} - \mathbf{L}\mathbf{x}^{(k)} - (\mathbf{D} + \mathbf{U})\mathbf{x}^{(k-1)}]$$

In definitiva, abbiamo

$$(\mathbf{D} + \omega\mathbf{L})\mathbf{x}^{(k)} = \omega\mathbf{b} + [(1-\omega)\mathbf{D} - \omega\mathbf{U}]\mathbf{x}^{(k-1)}$$

Ricordando il risultato di convergenza dei metodi iterativi, abbiamo che la matrice di iterazione è data da:

$$\mathbf{B} = (\mathbf{D} + \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} - \omega\mathbf{U}]$$

e il metodo di rilassamento risulterà convergente se e solo se $\rho(\mathbf{B}) < 1$.

La scelta del parametro ω deve rendere $\rho(\mathbf{B})$ quanto più piccolo è possibile. In alcuni casi è possibile determinare il valore di ω_{opt} per il quale $\rho(\mathbf{B})$ assume valore minimo.

Si dimostra che $\rho(\mathbf{B}) \geq 1$ quando $\omega \leq 0$ oppure $\omega \geq 2$ e, pertanto, per questi valori di ω, il metodo non può convergere.

La convergenza è invece assicurata per tutti i valori di $0 < \omega < 2$ quando A è simmetrica e definita positiva.

Function MATLAB

```
function [x,k,ier] = SOR(A,b,x0,omega,tol,kmax)
if prod(diag(A)) == 0
   x = []; k = []; ier = -1;
   return
end
D = diag(diag(A));
L = tril(A,-1);
U = triu(A,1);
for k = 1:kmax
   x = (D+omega*L)\(omega*b+((1-omega)*D-omega*U)*x0);
   if norm(x-x0) <= tol*norm(x);
      ier = 1;
      return
   end
   x0 = x;
end
ier = 0;
```

Metodi di discesa

Se la matrice dei coefficienti $\mathbf{A} \in \mathbb{R}^{n,n}$ del sistema lineare $\mathbf{Ax} = \mathbf{b}$ è **simmetrica e definita positiva**, allora il sistema lineare può essere risolto con un **metodo di discesa**.

Per definire un metodo di discesa, associamo al sistema lineare $\mathbf{Ax} = \mathbf{b}$ il seguente funzionale (quadratico)

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{b}$$

definito come **energia del sistema**.

I metodi di discesa sono metodi iterativi che si basano sulla seguente proprietà.

Teorema

Sia $\mathbf{A}$ simmetrica e definita positiva. Allora

$\mathbf{x}^*$ è soluzione del sistema $\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{x}^*$ è il punto di minimo del funzionale $\phi(\mathbf{x})$

Dimostrazione

Il minimo di $\phi(\mathbf{x})$ si ottiene annullando il suo gradiente, $\nabla\phi(\mathbf{x}) = \left(\frac{\partial\phi}{\partial x_1}, \ldots, \frac{\partial\phi}{\partial x_n}\right)^T$. Tenendo conto che $\mathbf{A}^T = \mathbf{A}$ e che

$$\frac{\partial\phi}{\partial x_k} = \frac{\partial}{\partial x_k}\left(\frac{1}{2}\sum_{i=1}^{n} x_i\left(\sum_{j=1}^{n} a_{ij}x_j\right) - \sum_{i=1}^{n} x_i b_i\right)$$

$$= \frac{1}{2}\left(\sum_{j=1}^{n} a_{kj}x_j + \sum_{i=1}^{n} a_{ik}x_i\right) - b_k = \sum_{j=1}^{n} a_{kj}x_j - b_k = (\mathbf{Ax})_k - (\mathbf{b})_k,$$

risulta

$$\nabla\phi(\mathbf{x}) = \mathbf{Ax} - \mathbf{b}$$

Pertanto, se $\mathbf{x}^*$ è un punto di minimo per $\phi(\mathbf{x})$, allora $\nabla\phi(\mathbf{x}^*) = \mathbf{o}$ e, quindi, $\mathbf{Ax}^* - \mathbf{b} = \mathbf{o}$, cioè $\mathbf{x}^*$ è soluzione del sistema.

... continua

Viceversa, poiché $\mathbf{A}$ è non singolare, essendo simmetrica e definita positiva, denotiamo con $\mathbf{x}^*$ l'unica soluzione del sistema $\mathbf{Ax} = \mathbf{b}$. Abbiamo allora che $\nabla\phi(\mathbf{x}^*) = \mathbf{o}$ e $\mathbf{x}^*$ è l'unico punto di estremo relativo per $\phi(\mathbf{x})$. Per verificare che esso è un punto di minimo, calcoliamo la matrice Hessiana del funzionale $\phi(\mathbf{x})$,

$$\mathbf{H}\phi(\mathbf{x}) = \begin{pmatrix} \phi_{x_1 x_1} & \phi_{x_1 x_2} & \cdots & \phi_{x_1 x_n} \\ \phi_{x_2 x_1} & \phi_{x_2 x_2} & \cdots & \phi_{x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{x_n x_1} & \phi_{x_n x_2} & \cdots & \phi_{x_n x_n} \end{pmatrix}$$

con $\phi_{x_h x_k} := \dfrac{\partial^2 \phi}{\partial x_h \partial x_k}$.

... continua

Poiché

$$(\mathbf{H})_{hk} = \phi_{x_h x_k} = \frac{\partial}{\partial x_h}\left(\sum_{j=1}^{n} a_{kj}x_j - b_k\right) = a_{kh} = a_{hk} = (\mathbf{A})_{hk},$$

risulta $\mathbf{H}\phi(\mathbf{x}) = \mathbf{A}$. Tenendo conto che $\mathbf{A}$ è simmetrica e definita positiva, $\mathbf{x}^*$ è dunque un punto di minimo per ϕ.

Per lo studio del metodo è utile introdurre il concetto di vettore residuo.

Definizione

Si definisce **residuo** dell'equazione $\mathbf{Ax} = \mathbf{b}$ relativo a un vettore $\mathbf{x}$, il vettore

$$\mathbf{r}(\mathbf{x}) = \mathbf{b} - \mathbf{Ax}$$

Osservazione

Dalla precedente dimostrazione si deduce che:

$$\nabla\phi(\mathbf{x}) = \mathbf{Ax} - \mathbf{b} = -\mathbf{r}(\mathbf{x})$$

Esempio

Per $n = 1$ il funzionale $\phi(\mathbf{x})$ rappresenta una parabola con la concavità rivolta verso l'alto.
Per $n = 2$, $\phi(\mathbf{x})$ rappresenta un paraboloide ellittico, e così via.

Per rappresentare graficamente il funzionale $\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$ per $n = 2$, scegliamo

$$\mathbf{A} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad \text{e} \quad \mathbf{b} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix},$$

in modo tale che il minimo coincida con $\mathbf{x}^* = (1, 1)^T$.
Il corrispondente funzionale è allora

$$\phi(x_1, x_2) = \frac{1}{2}(\lambda_1 x_1^2 + \lambda_2 x_2^2) - \lambda_1 x_1 - \lambda_2 x_2$$

... continua

Nelle Figure 1 e 2 sono rappresentati i funzionali per i valori $\lambda_1 = \lambda_2 = 1$ e $\lambda_1 = 1$, $\lambda_2 = 5$, rispettivamente. I grafici rappresentano un paraboloide circolare e uno ellittico.

Figura 1: $\lambda_1 = \lambda_2 = 1$

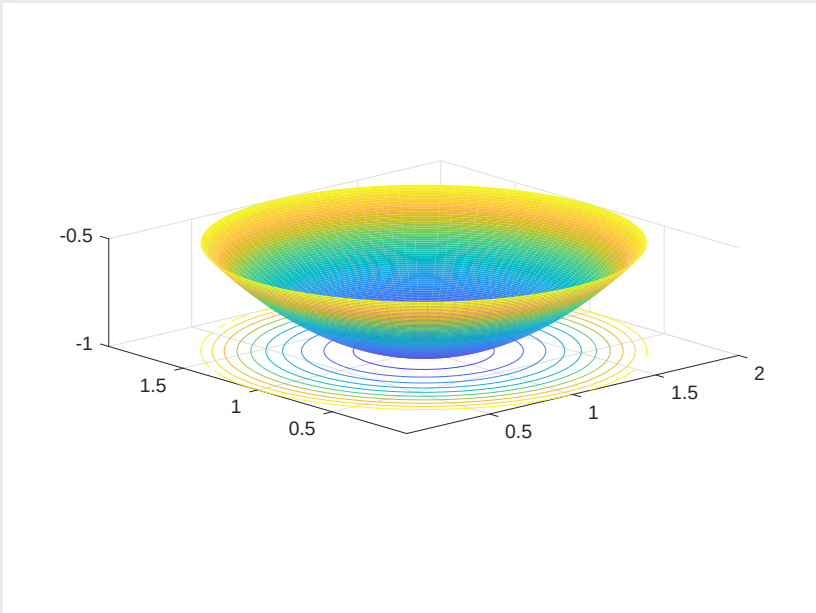
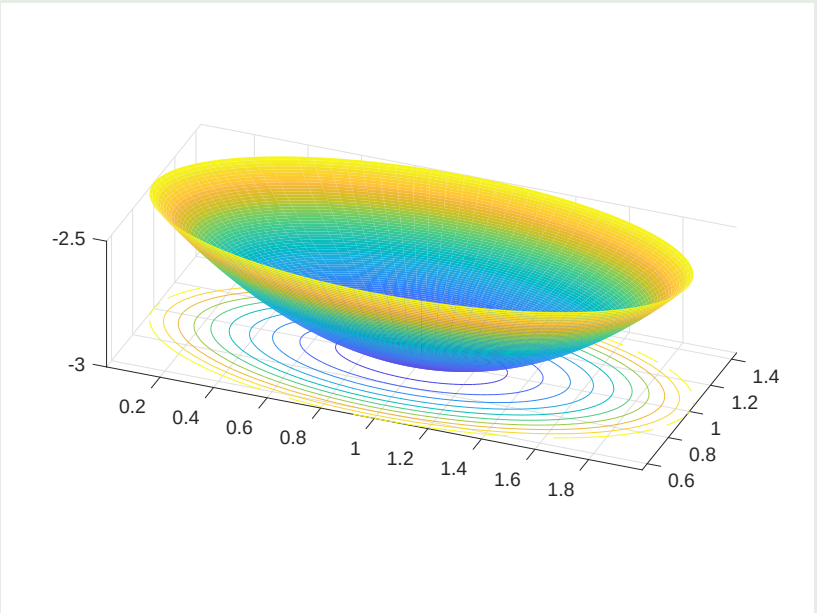


Figura 2: $\lambda_1 = 1, \lambda_2 = 5$



Le **curve di livello**, ovvero le proiezioni verticali sul piano (x_1, x_2) delle curve di intersezione della superficie $x_3 = \phi(x_1, x_2)$ con i piani orizzontali di equazione $x_3 = C$, sono delle ellissi concentriche, oppure delle circonferenze se $\lambda_1 = \lambda_2$, con centro $\mathbf{x}^*$, minimo del funzionale ϕ.

La struttura del funzionale ϕ, e in particolare la proprietà $\phi(\mathbf{x}^*) = \min_{\mathbf{x}\in\mathbb{R}^n} \phi(\mathbf{x})$, suggerisce una strategia per approssimare $\mathbf{x}^*$: costruire una successione $\mathbf{x}^{(k)}$ avente la proprietà che il valore del funzionale decresca nel passaggio da $\mathbf{x}^{(k)}$ a $\mathbf{x}^{(k+1)}$, avvicinandosi via via al valore minimo $\mathbf{x}^*$. Un metodo costruito in base a tale strategia è detto **metodo di discesa**.

Un metodo di discesa è definito da un relazione ricorsiva del tipo:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}, \quad k = 0, 1, 2, ...$$

ove il vettore $\mathbf{d}^{(k)}$ rappresenta il **vettore di discesa**, mentre lo scalare α_k rappresenta il **passo di discesa**.

Un vettore $\mathbf{d}^{(k)} \neq \mathbf{o}$ è ammissibile come vettore di discesa se

$$\frac{\partial \phi(\mathbf{x}^{(k)})}{\partial \mathbf{d}^{(k)}} = (\mathbf{d}^{(k)})^T \nabla \phi(\mathbf{x}^{(k)}) < 0,$$

ossia se muovendosi di un passo sufficientemente piccolo nella direzione e nel verso di $\mathbf{d}^{(k)}$ il funzionale $\phi(\mathbf{x})$ decresce.

I vari metodi si differenziano per la scelta del vettore di discesa $\mathbf{d}^{(k)}$. Infatti, una volta fissato $\mathbf{d}^{(k)}$, il passo di discesa α_k è univocamente determinato dalla condizione che il funzionale abbia la massima riduzione in quella direzione, cioè tale che

$$\phi(\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}) = \min_{\alpha \in \mathbb{R}} \phi(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$

Calcoliamo α_k. A tale scopo riscriviamo la funzione $\varphi(\alpha) = \phi(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)})$:

$$\begin{aligned}
\varphi(\alpha) =& \frac{1}{2}(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}) - (\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)})^T\mathbf{b} \\
&= \frac{1}{2}\left((\mathbf{x}^{(k)})^T\mathbf{A}\mathbf{x}^{(k)} + \alpha(\mathbf{d}^{(k)})^T\mathbf{A}\mathbf{x}^{(k)} + \alpha(\mathbf{x}^{(k)})^T\mathbf{A}\mathbf{d}^{(k)} + \alpha^2(\mathbf{d}^{(k)})^T\mathbf{A}\mathbf{d}^{(k)}\right) \\
&- (\mathbf{x}^{(k)})^T\mathbf{b} - \alpha(\mathbf{d}^{(k)})^T\mathbf{b} \\
&= \frac{1}{2}(\mathbf{x}^{(k)})^T\mathbf{A}\mathbf{x}^{(k)} - (\mathbf{x}^{(k)})^T\mathbf{b} + \alpha(\mathbf{d}^{(k)})^T(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}) + \frac{1}{2}\alpha^2(\mathbf{d}^{(k)})^T\mathbf{A}\mathbf{d}^{(k)} \\
&= \phi(\mathbf{x}^{(k)}) - \alpha(d^{(k)})^T\mathbf{r}^{(k)} + \frac{1}{2}\alpha^2(\mathbf{d}^{(k)})^T\mathbf{A}\mathbf{d}^{(k)}
\end{aligned}$$

avendo usato $(\mathbf{d}^{(k)})^T\mathbf{A}\mathbf{x}^{(k)} = (\mathbf{x}^{(k)})^T\mathbf{A}\mathbf{d}^{(k)}$ e posto $\mathbf{r}^{(k)} := \mathbf{r}(\mathbf{x}^{(k)})$.
Osserviamo che $\varphi(\alpha)$ è una parabola con la concavità verso l'alto, essendo $(\mathbf{d}^{(k)})^T\mathbf{A}\mathbf{d}^{(k)} > 0$.

Per determinare il punto di minimo calcoliamo $\varphi'(\alpha)$:

$$\varphi'(\alpha) = -(\mathbf{d}^{(k)})^T \mathbf{r}^{(k)} + \alpha(\mathbf{d}^{(k)})^T \mathbf{A}\mathbf{d}^{(k)}$$

Da $\varphi'(\alpha) = 0$ si deduce che il punto di minimo è dato da

$$\alpha_k = \frac{(\mathbf{d}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{d}^{(k)})^T \mathbf{A}\mathbf{d}^{(k)}}$$

Da $(\mathbf{d}^{(k)})^T \nabla\phi(\mathbf{x}^{(k)}) < 0$ segue che

$$(\mathbf{d}^{(k)})^T \mathbf{r}^{(k)} = -(\mathbf{d}^{(k)})^T \nabla\phi(\mathbf{x}^{(k)}) > 0$$

e, quindi, $\alpha_k > 0$.

I diversi metodi di discesa si distinguono per la diversa scelta del vettore $\mathbf{d}^{(k)}$.

Il metodo del gradiente

Per effettuare una scelta appropriata del vettore $\mathbf{d}^{(k)}$, ricordiamo che la direzione del gradiente è quella lungo la quale la funzione varia più rapidamente: la massima crescita si ha nella direzione e verso del gradiente, la massima decrescita si ha nella direzione del gradiente, ma nel verso opposto a esso.
Pertanto una scelta naturale per il vettore $\mathbf{d}^{(k)}$ consiste nel porre

$$\mathbf{d}^{(k)} = -\nabla\phi(\mathbf{x}^{(k)}) = \mathbf{r}^{(k)}$$

vale a dire muoversi nella direzione di massima decrescita per ϕ (**"steepest descent"**). Il corrispondente metodo di discesa è detto **metodo del gradiente** oppure **metodo della massima pendenza**.

Per il metodo del gradiente si ha allora

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^T \mathbf{A}\mathbf{r}^{(k)}}$$

$$\mathbf{r}^{(k+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)} = \mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}) = \mathbf{r}^{(k)} - \alpha_k \mathbf{A}\mathbf{r}^{(k)},$$

e, tenendo conto dell'espressione di α_k, si ha

$$(\mathbf{r}^{(k+1)})^T \mathbf{r}^{(k)} = (\mathbf{r}^{(k)} - \alpha_k \mathbf{A}\mathbf{r}^{(k)})^T \mathbf{r}^{(k)} = (\mathbf{r}^{(k)})^T \mathbf{r}^{(k)} - \alpha_k (\mathbf{r}^{(k)})^T \mathbf{A}\mathbf{r}^{(k)} = 0$$

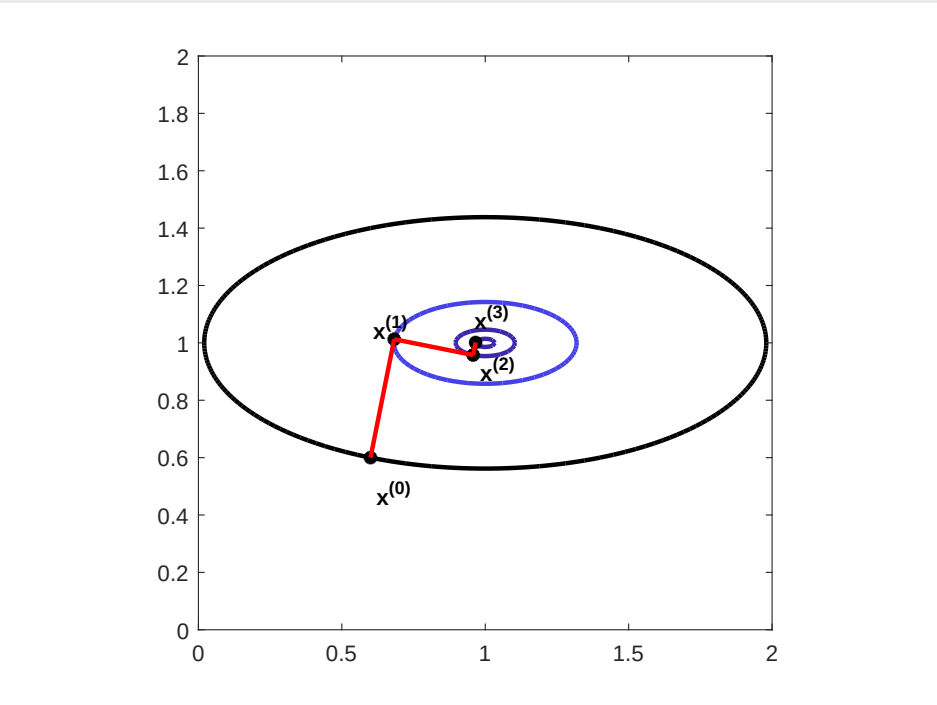
Pertanto, a ogni passo, $\mathbf{d}^{(k+1)} = \mathbf{r}^{(k+1)}$ è ortogonale al vettore $\mathbf{d}^{(k)} = \mathbf{r}^{(k)}$ del passo precedente, ossia i gradienti sono a due a due ortogonali.
La scelta $\mathbf{d}^{(k)} = \mathbf{r}^{(k)}$, anche se in ciascun punto $\mathbf{x}^{(k)}$ sfrutta la direzione della massima pendenza, come vedremo, può non essere la migliore, in particolare quando la matrice è mal condizionata.

Esempio

Per mostrare il comportamento del metodo nel caso $n = 2$, scegliamo

$$\mathbf{A} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad \text{e} \quad \mathbf{b} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix},$$

in modo tale che il minimo coincida con $\mathbf{x}^* = (1, 1)^T$.
Partendo da $\mathbf{x}^{(0)} = (0.6, 0.6)^T$, per $\lambda_1 = 1$ e $\lambda_2 = 5$ il metodo del gradiente converge seguendo una traiettoria a zig-zag e graficamente si ha



... continua esempio

Nel caso $n = 2$, il metodo del gradiente ha dunque il seguente comportamento:

si parte da $\mathbf{x}^{(0)}$ e si scende lungo la direzione del gradiente, che è ortogonale alla curva di livello passante per $\mathbf{x}^{(0)}$.

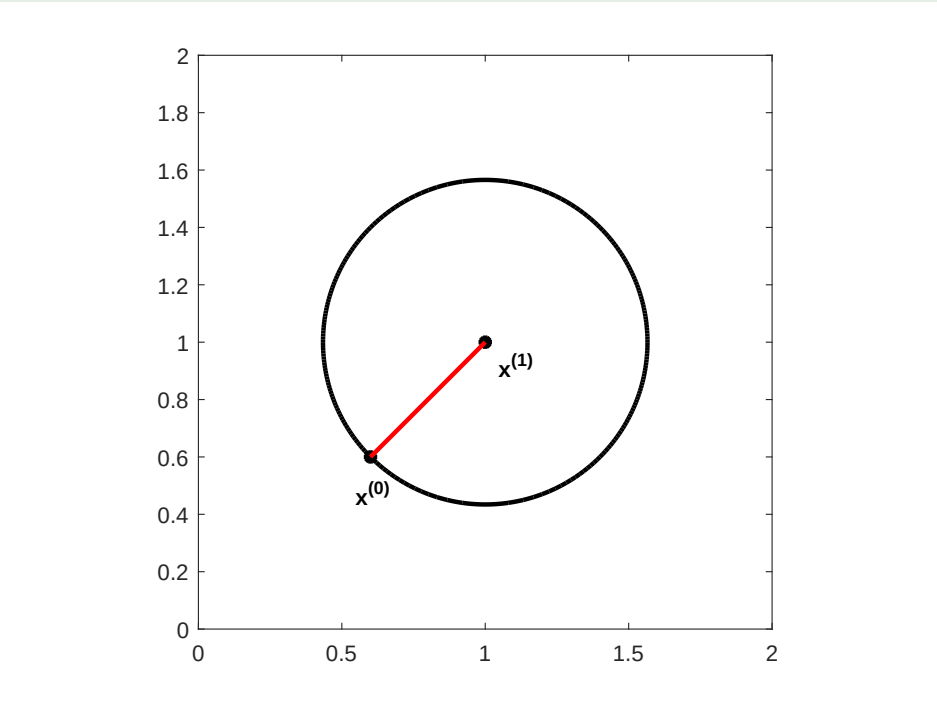
Si giunge in $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)}$ che rappresenta il punto in cui il funzionale $\phi(\mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)})$ assume valore minimo al variare di α; questo valore si raggiunge per $\alpha = \alpha_0$. In $\mathbf{x}^{(1)}$ la direzione del gradiente $\mathbf{d}^{(0)} = -\nabla\phi(\mathbf{x}^{(0)})$ è tangente alla curva di livello $z = \phi(\mathbf{x}^{(1)})$.

Quindi, si riparte da $\mathbf{x}^{(1)}$ lungo la direzione $\mathbf{d}^{(1)} = -\nabla\phi(\mathbf{x}^{(1)})$ del gradiente, ortogonale alla direzione $\mathbf{d}^{(0)}$, per raggiungere $\mathbf{x}^{(2)}$; e così via....

Pertanto, si ha che due direzioni consecutive $\mathbf{d}^{(k)}$ e $\mathbf{d}^{(k+1)}$ sono ortogonali tra loro, e due direzioni alternate $\mathbf{d}^{(k)}$ e $\mathbf{d}^{(k+2)}$ sono parallele tra loro.

... continua esempio

Per $\lambda_1 = \lambda_2 = 1$ il metodo del gradiente converge in una sola iterazione, perchè la direzione del gradiente passa per il centro ($\mathbf{x}^{(1)} = \mathbf{x}^*$).



Nell'implementazione del metodo del gradiente, al posto del criterio di arresto precedentemente utilizzato $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}||_2 \leq tol||\mathbf{x}^{(k+1)}||_2$, considereremo il seguente criterio:

$$||\mathbf{r}^{(k)}||_2 \leq tol||\mathbf{b}||_2, \quad tol \geq 2\varepsilon_m$$

Tale criterio di arresto scaturisce dalle seguenti motivazioni.

- Poiché $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} = \mathbf{A}\mathbf{x}^* - \mathbf{A}\mathbf{x}^{(k)} = \mathbf{A}\mathbf{e}^{(k)}$, si ha che, essendo $\mathbf{A}$ non singolare, $\mathbf{e}^{(k)} = \mathbf{o}$ se e solo se $\mathbf{r}^{(k)} = \mathbf{o}$. Quindi ha senso richiedere che $\mathbf{r}^{(k)}$ sia "piccolo". Ma quanto "piccolo"?
- Poiché $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b} - \mathbf{r}^{(k)}$, si ha che $\mathbf{r}^{(k)}$ è quella quantità che sommata a $\mathbf{b}$ consente di ottenere $\mathbf{x}^{(k)}$. In precisione finita di calcolo, $\mathbf{r}^{(k)}$ dà contributo a $\mathbf{b}$ solo se sufficientemente grande, ovvero $||\mathbf{r}^{(k)}||_2 > eps||\mathbf{b}||_2$. Ne consegue che se $||\mathbf{r}^{(k)}||_2 \leq eps||\mathbf{b}||_2$, non ha senso procedere con il metodo iterativo in quanto l'approssimazione $\mathbf{x}^{(k)}$ rimane invariata. Da tale osservazione scaturisce il criterio d'arresto $||\mathbf{r}^{(k)}||_2 \leq tol||\mathbf{b}||_2$, con $tol \geq eps = 2\varepsilon_m$.
- Anche le function MATLAB che implementano i metodi di discesa utilizzano il criterio di arresto sul residuo $||\mathbf{r}^{(k)}||_2$, e non sulla quantità $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}||_2$.

Scriviamo la function MATLAB che implementa il metodo del gradiente, tenendo conto che per esso le formule caratterizzanti sono le seguenti:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}, \quad \alpha_k = \frac{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^T \mathbf{A}\mathbf{r}^{(k)}}, \quad \mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k \mathbf{A}\mathbf{r}^{(k)}$$

Function MATLAB

```
function [x,k,ier] = gradiente(A,b,x,tol,kmax)
ier = 0;
r = b-A*x;
for k = 1:kmax
  s = A*r;
  alpha = r'*r/(r'*s);
  x = x+alpha*r;
  r = r-alpha*s;
  if norm(r) <= tol*norm(b)
    ier = 1;
    break
  end
end
```

Osservazioni

- A ogni passo occorre calcolare un prodotto matrice per vettore, il cui costo in termini delle operazioni aritmetiche di moltiplicazione (e di addizione), è essenzialmente n^2 per n grande. Gli altri passi dell'algoritmo (calcolo di prodotti scalari e aggiornamento di vettori) richiedono un numero di operazioni dell'ordine di n. Pertanto il costo complessivo del metodo del gradiente è essenzialmente dato da kn^2 per n grande, ove k è il numero di iterazioni eseguite.
- Per ridurre il costo computazionale si è usata l'espressione di $\mathbf{r}^{(k+1)}$ in funzione di $\mathbf{r}^{(k)}$, anche se in pratica per garantire una migliore stabilità dell'algoritmo è opportuno dopo un certo numero di passi calcolare il residuo con la relazione $\mathbf{r}^{(k+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)}$.

Per fornire un risultato di convergenza per il metodo del gradiente, definiamo al passo k l'errore

$$\mathbf{e}^{(k)} := \mathbf{x}^* - \mathbf{x}^{(k)}$$

e introduciamo la norma vettoriale, detta **norma dell'energia**,

$$||\mathbf{x}||_A = \sqrt{\mathbf{x}^T \mathbf{A}\mathbf{x}}$$

Si può dimostrare il seguente teorema.

Teorema

Siano $\lambda_{\max}$ e $\lambda_{\min}$ il massimo e il minimo autovalore della matrice simmetrica e definita positiva $\mathbf{A}$ di ordine n. Sia inoltre $K_2(\mathbf{A}) = \lambda_{max}/\lambda_{min} \geq 1$ il numero di condizionamento spettrale della matrice $\mathbf{A}$. La successione $\{\mathbf{x}^{(k)}\}$ generata dal metodo del gradiente converge qualunque sia il vettore iniziale $\mathbf{x}^{(0)}$ e il corrispondente errore $\mathbf{e}^{(k)}$ soddisfa per ogni k la seguente stima:

$$||\mathbf{e}^{(k)}||_A \leq \left(\frac{K_2(\mathbf{A}) - 1}{K_2(\mathbf{A}) + 1}\right)^k ||\mathbf{e}^{(0)}||_A$$

Osservazione

Osserviamo che il metodo converge perché il fattore di riduzione

$$\frac{K_2(\mathbf{A})-1}{K_2(\mathbf{A})+1} = 1 - \frac{2}{K_2(\mathbf{A})+1}$$

è minore di 1.

Inoltre la velocità di convergenza dipende da tale fattore: quanto più esso è piccolo, tanto più è rapida la convergenza. Poiché il fattore di riduzione è tanto più vicino a zero quanto più $K_2(\mathbf{A})$ si avvicina a 1, la convergenza della successione $\{\mathbf{x}^{(k)}\}$ è tanto più rapida quanto più la matrice è ben condizionata.

Quando $K_2(A) = 1$, cioè quando gli autovalori sono tutti uguali, il metodo converge in una sola iterazione. Quando, invece $K_2(A) >> 1$, la convergenza del metodo può essere molto lenta e il numero delle iterazioni è proporzionale a $K_2(A)$.

Osservazione

Il criterio d'arresto sul residuo può non essere affidabile quando la matrice $\mathbf{A}$ è malcondizionata. Infatti, si può dimostrare che

$$\frac{||\mathbf{e}^{(k)}||_2}{||\mathbf{x}^*||_2} \leq K_2(\mathbf{A}) \frac{||\mathbf{r}^{(k)}||_2}{||\mathbf{b}||_2}$$

e pertanto, pur essendo piccolo il residuo, l'errore potrebbe non essere altrettanto piccolo.

In caso di cattivo condizionamento, anche il criterio di arresto $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}||_2 \leq tol||\mathbf{x}^{(k+1)}||_2$, potrebbe non essere affidabile; infatti, in questo caso, a causa della convergenza lenta del metodo, due iterate successive potrebbero essere vicine tra loro (e quindi il criterio risulta soddisfatto) ma entrambe lontane dalla soluzione esatta.

Metodo del gradiente coniugato

La scelta della direzione di massima decrescita locale del funzionale ϕ come direzione di discesa, può dunque non essere la migliore, in particolare quando la matrice è malcondizionata. In tal caso, per $n = 2$ per esempio, le curve di livello sono ellissi molto allungate e muoversi lungo la direzione del gradiente può portare a una modesta riduzione del funzionale.
La direzione ottimale sarebbe quella passante per la soluzione $\mathbf{x}^*$, perché ci permetterebbe di giungere alla soluzione in un solo passo!

Un esame del problema mostra che la direzione

$$\mathbf{d}^{(k)} = \begin{cases} \mathbf{r}^{(0)}, & \text{se } k = 0 \\ \mathbf{r}^{(k)} + \beta_k \mathbf{d}^{(k-1)}, & \text{se } k \geq 1 \end{cases}$$

dove β_k è determinato in modo tale che

$$(\mathbf{d}^{(k)})^T \mathbf{A}\mathbf{d}^{(k-1)} = 0$$

rappresenta una direzione di discesa conveniente.

I vettori $\mathbf{d}^{(k)}$ e $\mathbf{d}^{(k-1)}$ che soddisfano la suddetta relazione sono detti **A-coniugati** e da questa definizione discende il nome del corrispondente metodo di discesa, che è detto **metodo del gradiente coniugato**.

Si dimostra che la direzione $\mathbf{d}^{(k)}$ è **A**-coniugata non solo con $\mathbf{d}^{(k-1)}$ ma con tutte le precedenti, cioè

$$(\mathbf{d}^{(j)})^T \mathbf{A}\mathbf{d}^{(k)} = 0 \qquad \forall j, k \geq 0,\ j \neq k$$

e che i residui sono tutti ortogonali tra loro

$$(\mathbf{r}^{(j)})^T \mathbf{r}^{(k)} = 0 \qquad \forall j, k \geq 0,\ j \neq k$$

Osservazione

Ricordiamo che nel caso del metodo del gradiente solo due residui consecutivi sono ortogonali tra loro, ovvero solo due direzioni consecutive sono ortogonali. Due direzioni alternate sono invece parallele. Pertanto, esistono solo due direzioni linearmente indipendenti.

Osservazione

Nel caso del metodo del gradiente coniugato, i residui $\mathbf{r}^{(0)}$, $\mathbf{r}^{(1)}$,...,$\mathbf{r}^{(n-1)}$ formano un sistema ortogonale in $\mathbb{R}^n$; se essi sono tutti diversi dal vettore nullo formano una base in tale spazio.

Pertanto, l'$n+1$-esimo residuo $\mathbf{r}^{(n)}$, essendo ortogonale a ciascuno di essi, sarà nullo, e dunque $\mathbf{x}^{(n)} = \mathbf{x}^*$. In altri termini, il metodo del gradiente coniugato fornisce, in aritmetica infinita e quindi in assenza degli errori dovuti all'aritmetica di macchina, la soluzione del sistema lineare in al più n iterazioni.

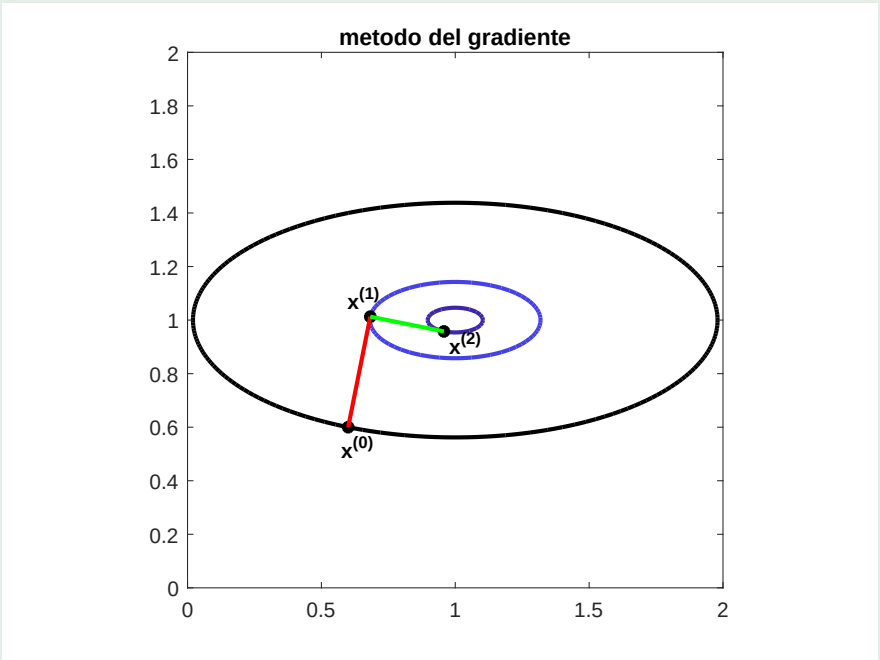
In effetti tale metodo fu proposto come metodo diretto. Esso viene considerato come un metodo iterativo perché se n è molto grande e la matrice non è malcondizionata, si ottiene una buona approssimazione della soluzione dopo un numero di iterazioni molto minore di n. Pertanto il metodo del gradiente coniugato risulta molto conveniente per trattare problemi di grosse dimensioni.

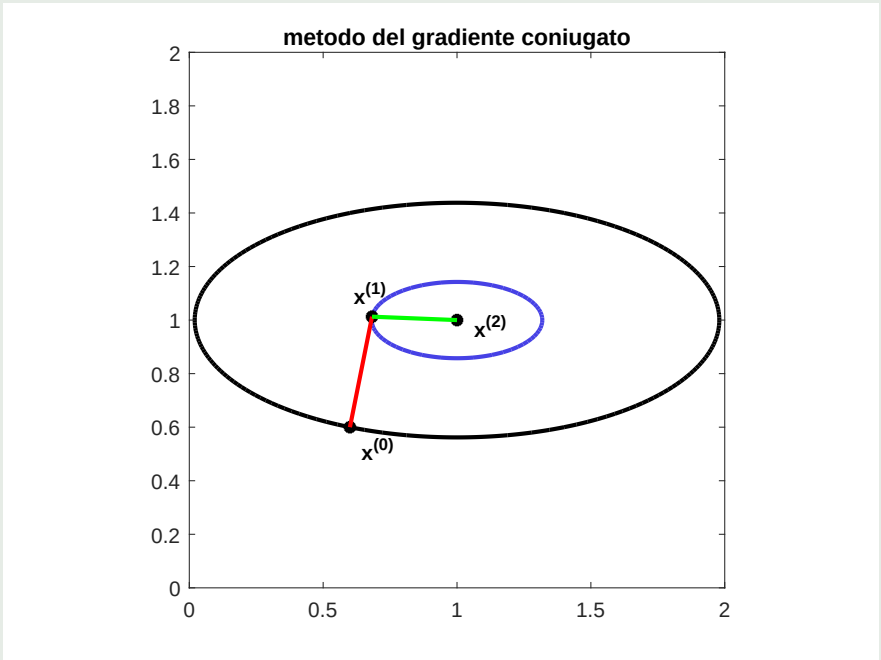
Esempio

Per confrontare, nel caso $n = 2$, il comportamento del metodo del gradiente e quello del gradiente coniugato scegliamo

$$\mathbf{A} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$$

in modo tale che il minimo coincida con $\mathbf{x}^* = (1, 1)^T$.
Partendo da $\mathbf{x}^{(0)} = (0.6, 0.6)^T$, per $\lambda_1 = 1$ e $\lambda_2 = 5$ graficamente si ha





Il metodo del gradiente coniugato converge in sole due iterazioni.

Osservazione

- Come per il metodo del gradiente, anche per il metodo del gradiente coniugato, l'operazione che presenta in generale un maggiore costo computazionale è quella relativa alla moltiplicazione della matrice $\mathbf{A}$ per il vettore $\mathbf{d}^{(k)}$. Essendo il costo di quest'ultimo pari a n^2 moltiplicazioni (e addizioni), il costo complessivo è circa kn^2, ove k è il numero di iterazioni eseguite.
- Se il metodo richiedesse un numero di passi dell'ordine di n e se $\mathbf{A}$ non fosse sparsa, il costo complessivo sarebbe dell'ordine di n^3, superiore a quella del metodo di Choleski. Tuttavia, nella risoluzione di sistemi di equazioni lineari che scaturiscono dalla discretizzazione di problemi differenziali, il numero di iterazioni richieste è di solito molto inferiore alla dimensione della matrice e la matrice è sparsa.

Per quanto riguarda la stima di convergenza dell'errore $\mathbf{e}^{(k)}$ associato alla successione $\{\mathbf{x}^{(k)}\}$ generata dal metodo del gradiente coniugato, vale il seguente teorema.

Teorema

La successione $\{\mathbf{x}^{(k)}\}$ generata dal metodo del gradiente coniugato converge qualunque sia il vettore iniziale $\mathbf{x}^{(0)}$ in al più n iterazioni e il corrispondente errore $\mathbf{e}^{(k)}$ soddisfa per ogni k la seguente stima:

$$||\mathbf{e}^{(k)}||_A \leq 2 \left(\frac{\sqrt{K_2(\mathbf{A})} - 1}{\sqrt{K_2(\mathbf{A})} + 1} \right)^k ||\mathbf{e}^{(0)}||_A$$

Osservazioni

- Se $K_2(\mathbf{A}) \approx 1$ sono sufficienti pochi passi, mentre se $K_2(\mathbf{A}) >> 1$ è possibile che siano necessari n passi per ottenere un'approssimazione accettabile della soluzione, e se n è molto grande è possibile che non si riesca ad ottenere un'approssimazione accettabile a causa degli errori di arrotondamento.
- I metodi del gradiente e del gradiente coniugato ammettono lo stesso costo computazionale; mentre, per quanto riguarda la velocità di convergenza, il fattore di riduzione del metodo del gradiente dipende da $K_2(\mathbf{A})$ e quello del gradiente coniugato da $\sqrt{K_2(\mathbf{A})}$. Poiché quest'ultimo valore risulta più piccolo, il metodo del gradiente coniugato converge più velocemente del metodo del gradiente. Tuttavia, se la matrice è malcondizionata la convergenza di entrambi i metodi potrebbe risultare lenta.

Comandi MATLAB

- `x = pcg(A,b)` determina la soluzione `x` del sistema lineare `Ax=b` simmetrico e definito positivo mediante il metodo del gradiente coniugato, a partire dal vettore iniziale $\mathbf{x}^{(0)} = \mathbf{o}$.
- `x = pcg(A,b,tol,kmax)` determina la soluzione `x` imponendo il criterio d'arresto $||\mathbf{r}||_2/||\mathbf{b}||_2 \leq$`tol` e il massimo numero di iterazioni `kmax`. Se `tol = []` allora di default `tol`$= 10^{-6}$; se `kmax = []` allora `kmax`$= \min\{n, 20\}$.

Sistemi lineari precondizionati

L'analisi della convergenza dei metodi di discesa per un sistema lineare $\mathbf{Ax} = \mathbf{b}$ mette in evidenza il ruolo cruciale del numero di condizionamento $K_2(\mathbf{A})$ della matrice $\mathbf{A}$: i suddetti metodi convergono lentamente se la matrice $\mathbf{A}$ è malcondizionata.

Per ovviare a questo inconveniente, si può sostituire il sistema $\mathbf{Ax} = \mathbf{b}$ con il sistema equivalente

$$\mathbf{P}^{-1}\mathbf{Ax} = \mathbf{P}^{-1}\mathbf{b}$$

dove $\mathbf{P}$ è una matrice non singolare avente le seguenti proprietà:

1. $K_2(\mathbf{P}^{-1}\mathbf{A}) << K_2(\mathbf{A})$;
2. gli elementi di $\mathbf{P}$ sono calcolabili in modo poco costoso;
3. il prodotto matrice-vettore $\mathbf{y} = \mathbf{P}^{-1}\mathbf{z}$ (ovvero la risoluzione del sistema $\mathbf{Py} = \mathbf{z}$) deve avere un costo confrontabile con quello del prodotto matrice-vettore per il calcolo di $\mathbf{y} = \mathbf{Az}$

Una matrice $\mathbf{P}$ avente le suddette proprietà si dice **matrice di precondizionamento** per $\mathbf{A}$.

Osserviamo che la matrice che realizza in modo ottimale la prima condizione è la matrice $\mathbf{P} = \mathbf{A}$, per la quale si ha

$$K_2(\mathbf{P}^{-1}\mathbf{A}) = K_2(\mathbf{I}) = 1!$$

Ma $\mathbf{P} = \mathbf{A}$ non soddisfa l'ultima delle precedenti condizioni. Tuttavia l'osservazione suggerisce di scegliere $\mathbf{P}$ come una opportuna approssimazione della matrice $\mathbf{A}$.

Nel caso di una matrice $\mathbf{A}$ simmetrica e definita positiva, una strategia che porta a una matrice di precondizionamento efficace consiste nel costruire una **fattorizzazione incompleta di Choleski** della matrice $\mathbf{A}$.

Precisamente, indicata con $\mathbf{A} = \mathbf{R}^T\mathbf{R}$ la fattorizzazione classica di Choleski ove il fattore $\mathbf{R}$ è triangolare superiore con elementi diagonali positivi, si sceglie $\mathbf{P} = \bar{\mathbf{R}}^T\bar{\mathbf{R}}$, con $\bar{\mathbf{R}}$ triangolare superiore i cui elementi sono definiti nel seguente modo:

$$\bar{r}_{ij} = \begin{cases} r_{ij}, & \text{se } a_{ij} \neq 0 \\ 0, & \text{se } a_{ij} = 0 \end{cases}$$

In questo modo $\mathbf{P}$ è sparsa tanto quanto $\mathbf{A}$. Inoltre per un'ampia classe di metodi di discretizzazione che conducono a matrici $\mathbf{A}$ simmetriche e definite positive, risulta $K_2(\mathbf{P}^{-1}\mathbf{A}) << K_2(\mathbf{A})$ e un sistema $\mathbf{Py} = \mathbf{z}$ può essere facilmente risolto mediante la risoluzione dei due sistemi triangolari $\bar{\mathbf{R}}^T\mathbf{w} = \mathbf{z}$ e $\bar{\mathbf{R}}\mathbf{y} = \mathbf{w}$.

Metodo del gradiente coniugato precondizionato

Si noti che la matrice $\mathbf{P}^{-1}\mathbf{A}$ non è simmetrica e, quindi, i metodi di discesa non possono essere applicati direttamente al sistema $\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}$.

Per ovviare a questo inconveniente si può sostituire il problema originario simmetrico e definito positivo $\mathbf{A}\mathbf{x} = \mathbf{b}$ con il problema

$$(\bar{\mathbf{R}}^T)^{-1}\mathbf{A}\mathbf{x} = (\bar{\mathbf{R}}^T)^{-1}\mathbf{b} \Rightarrow \underbrace{(\bar{\mathbf{R}}^T)^{-1}\mathbf{A}\bar{\mathbf{R}}^{-1}}_{\bar{\mathbf{A}}} \underbrace{\bar{\mathbf{R}}\mathbf{x}}_{\bar{\mathbf{x}}} = \underbrace{(\bar{\mathbf{R}}^T)^{-1}\mathbf{b}}_{\bar{\mathbf{b}}} \Rightarrow \begin{cases} \bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}} \Rightarrow \bar{\mathbf{x}} \\ \bar{\mathbf{R}}\mathbf{x} = \bar{\mathbf{x}} \Rightarrow \mathbf{x} \end{cases}$$

Osserviamo che $\bar{\mathbf{A}}$ è simmetrica

$$\bar{\mathbf{A}}^T = ((\bar{\mathbf{R}}^T)^{-1}\mathbf{A}\bar{\mathbf{R}}^{-1})^T = (\bar{\mathbf{R}}^{-1})^T\mathbf{A}^T\bar{\mathbf{R}}^{-1} = (\bar{\mathbf{R}}^{-1})^T\mathbf{A}\bar{\mathbf{R}}^{-1} = \bar{\mathbf{A}}$$

e definita positiva

$$\mathbf{x}^T\bar{\mathbf{A}}\mathbf{x} = \mathbf{x}^T((\bar{\mathbf{R}}^T)^{-1}\mathbf{A}\bar{\mathbf{R}}^{-1})\mathbf{x} = (\bar{\mathbf{R}}^{-1}\mathbf{x})^T\mathbf{A}(\bar{\mathbf{R}}^{-1}\mathbf{x}) = \mathbf{y}^T\mathbf{A}\mathbf{y} > 0,$$

essendo $\mathbf{A}$ simmetrica e definita positiva e $\mathbf{y} = \bar{\mathbf{R}}^{-1}\mathbf{x} \neq \mathbf{o}$ per ogni $\mathbf{x} \neq \mathbf{o}$. Inoltre,

$$K_2(\bar{\mathbf{A}}) = K_2((\bar{\mathbf{R}}^T)^{-1}\mathbf{A}\bar{\mathbf{R}}^{-1}) = K_2((\bar{\mathbf{R}}^T)^{-1}\mathbf{R}^T\mathbf{R}\bar{\mathbf{R}}^{-1}) \approx K_2(\mathbf{I})$$

È pertanto possibile applicare il metodo del gradiente coniugato al sistema $\bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}}$. Tale procedura prende il nome di **metodo del gradiente coniugato precondizionato.**

Comandi MATLAB

- `L = ichol(A)` calcola il fattore `L`$= \bar{\mathbf{R}}^T$ del precondizionatore $\mathbf{P} = \bar{\mathbf{R}}^T\bar{\mathbf{R}}$ che si ottiene mediante la fattorizzazione incompleta di Choleski di `A`. La matrice `A` deve essere definita in formato sparso; quindi, nel caso non lo fosse, usare la seguente sintassi `L = ichol(sparse(A))`.
- `x = pcg(A,b,tol,kmax,M,N,x0)` determina la soluzione `x` del sistema lineare `Ax=b` con il metodo del gradiente coniugato precondizionato a partire dal vettore iniziale `x0`. Il precondizionatore utilizzato è $\mathbf{P} = \mathbf{MN}$. Se si vuole utilizzare il precondizionatore che si ottiene mediante la fattorizzazione incompleta di Choleski, allora `M=L` e `N=L'`. Se `M=[]` e `N=[]` il metodo non utilizza precondizionatori.

Osservazione

Se la matrice $\mathbf{A}$ non è simmetrica e definita positiva, non è possibile applicare i metodi di discesa al sistema $\mathbf{Ax} = \mathbf{b}$.
Tenendo conto che $\mathbf{A}^T\mathbf{A}$ è simmetrica e definita positiva se e solo se $\mathbf{A}$ non è singolare, un modo per ricondursi a un sistema simmetrico e definito positivo consiste nel risolvere il sistema

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

Tuttavia questa procedura non è conveniente in quanto si dimostra che $K_2(\mathbf{A}^T\mathbf{A}) = (K_2(\mathbf{A}))^2$ e il prodotto matrice-vettore $\mathbf{y} = \mathbf{A}^T\mathbf{Az}$ costa il doppio rispetto al prodotto $\mathbf{y} = \mathbf{Az}$.

Sono stati quindi proposti metodi alternativi tra i quali molto popolare è il metodo GMRes.

Proprietà generali dei metodi iterativi

Riassumiamo infine le proprietà dei metodi iterativi per la risoluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$:

1. $\mathbf{x}$ viene determinata come limite di una successione di vettori convergente;
2. $\mathbf{x}$ in aritmetica con precisione infinita di calcolo viene determinata in maniera approssimata; in aritmetica con precisione finita di calcolo può essere determinata con una precisione soddisfacente le richieste dell'utente;
3. i metodi iterativi non modificano la matrice dei coefficienti $\mathbf{A}$;
4. i metodi iterativi sono efficienti per matrici sparse e di grandi dimensioni.