Sistemi lineari

1. Generalità sui sistemi lineari

Consideriamo un sistema lineare

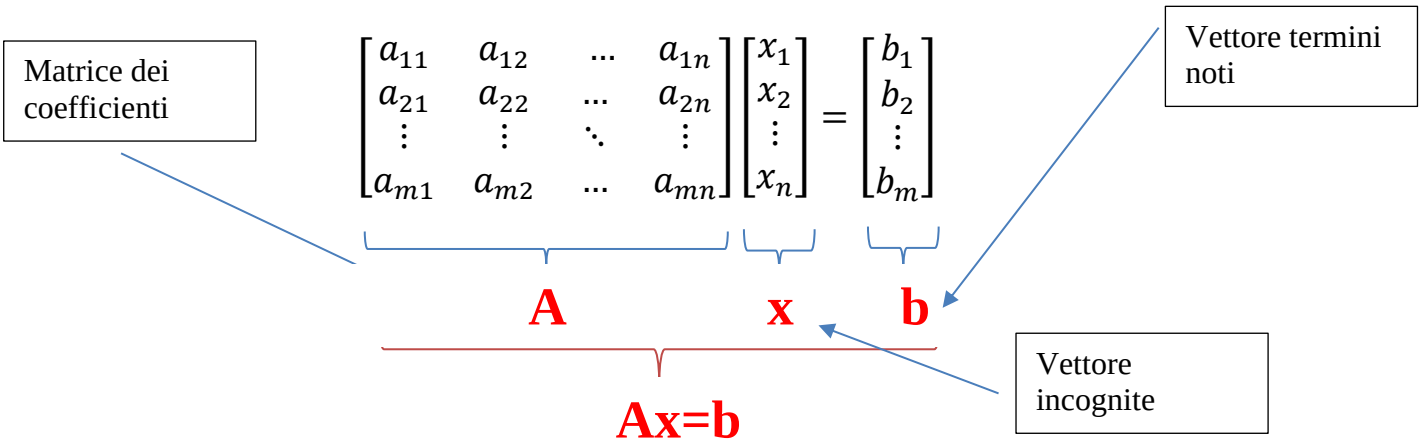
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n = b_3 \\ \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n = b_m \end{cases}$$

Noti i coefficienti a_{ij}, *i=1,..m, j=1,n,* e le componenti del termine noto b_i, i=1,..*m*, si vuole individuare il vettore x incognito di n componenti che soddisfa contemporaneamente le *m relazioni lineari.*

In formato matriciale, il sistema po' essere espresso come:

$$Ax=b$$

con $A \in M(m \times n), x \in R^n, b \in R^m$

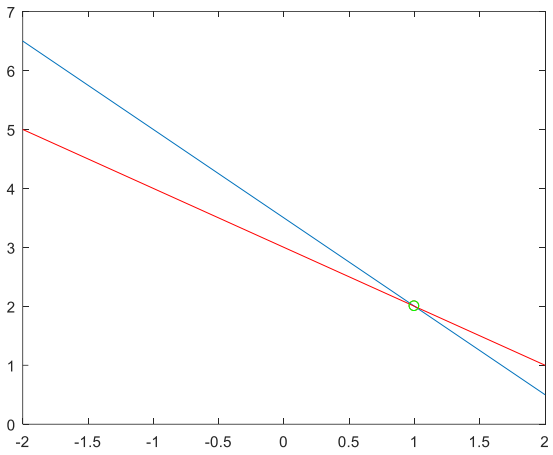


Un sistema lineare si dice **compatibile** se ammette almeno una soluzione, si dice **incompatibile** nel caso non ammetta alcuna soluzione

Interpretazione geometrica di sistemi lineari: $A \in M(2 \times 2), b \in R^2, x \in R^2$

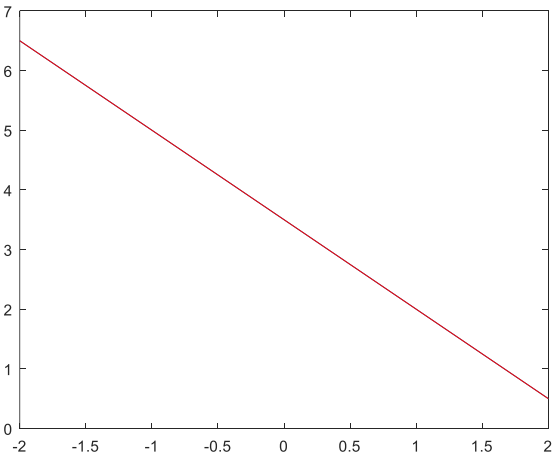
Punto di intersezione tra due rette

$$\begin{cases} 3x + 2y = 7 \\ x + y = 3 \end{cases}$$



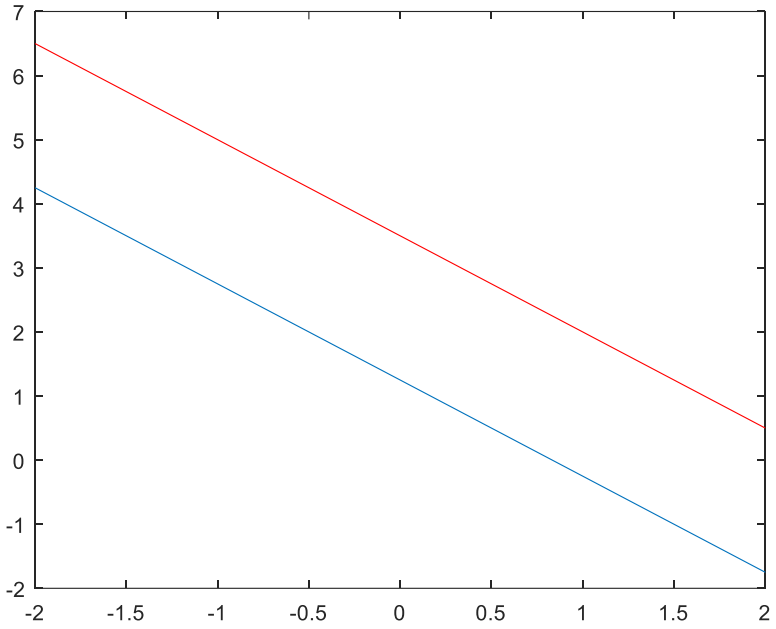
Le due rette coincidono, infinite soluzioni

$$\begin{cases} 3x + 2y = 7 \\ 6x + 4y = 14 \end{cases}$$



Le due rette **sono parallele**, nessuna soluzione.

$$\begin{cases} 3x + 2y = 7 \\ 6x + 4y = 5 \end{cases}$$



Teorema di Rouchè- Capelli

Il sistema lineare Ax=b ammette soluzioni se e solo se la matrice dei coefficienti A e la matrice completa [A b] hanno lo stesso rango:

Rank(A)=r=Rank([A b])

Altrimenti se (Rank(A)≠r≠Rank([A b])) il sistema non ammette soluzioni

Si possono verificare i seguenti casi:

Caso 1) $m<n$, cioè abbiamo **più incognite che relazioni lineari tra di esse**. Il sistema è del tipo

$$\begin{bmatrix} & A & \\ & m < n & \end{bmatrix} \begin{bmatrix} \\ x \\ \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ b \\ \\ \end{bmatrix}$$

e si dice **indeterminato.** Se indichiamo con k il rango della matrice A e $k<n$, allora il sistema ammette ∞^{n-k} soluzioni.

2) Caso $m>n$, cioè abbiamo meno incognite che relazioni lineari tra di esse. Il sistema è del tipo:

$$\begin{bmatrix} & & \\ & A & \\ & m > n & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \\ x \\ \\ \end{bmatrix} = \begin{bmatrix} \\ b \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

e si dice sistema **sovradeterminato.** In questo caso il sistema non ammette una soluzione esatta, ma una soluzione approssimata.

3) Caso $m=n$, cioè il numero di incognite è uguale al numero di relazioni lineari tra di esse. Il sistema è della forma

$$\begin{bmatrix} & A & \\ & m=n & \end{bmatrix}\begin{bmatrix} x \end{bmatrix}=\begin{bmatrix} b \end{bmatrix}$$

e si dice sistema normale e sotto opportune ipotesi può ammettere una ed una sola soluzione.

Per il momento ci interesseremo di sistemi normali, con $n=m$.

Definizione:

A (nxn) è detta **NON singolare** se soddisfa una delle seguenti condizioni equivalenti:

1) $det(A) \neq 0$

2) Esiste la matrice inversa A^{-1} di A

3) rank(A)=n

Altrimenti A è singolare.

Teorema: *Condizione necessaria e sufficiente affinché il sistema lineare Ax=b , $A \in M(n \times n)$, $x, b \in R^n$ ammetta una ed una sola soluzione, comunque si scelga b, è che la matrice A sia a rango massimo (cioè che la matrice A sia invertibile); si ha perciò:*

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

Se il sistema è omogeneo (**b=0**), ed A è non singolare allora esiste solo la soluzione nulla **x=0**.

Nota: il metodo dell'inversa risulta poco efficiente (perché richiede la soluzione di n sistemi lineari), ma anche poco accurato.

Infatti, anche nel caso di una singola equazione, come per esempio,

$$7x = 21;$$

il metodo dell'inversa ci restituisce la soluzione

$$x = 7^{-1} * 21 = 0.142857 * 21 = 2.99997$$

Invece, se si dividono per 7 entrambi i membri dell'equazione, si trova

$$x = 21/7 = 3$$

Un metodo ben noto per la soluzione di sistemi lineari è basato sulla regola di Cramer, che calcola la componente j-esima della soluzione facendo uso del calcolo del determinante (realizzato mediante la formula di Laplace).

$$x_1 = \frac{\det\begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_n & a_{n2} & \dots & a_{nn} \end{vmatrix}}{\det\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}}, \quad x_2 = \frac{\det\begin{vmatrix} a_{11} & b_1 & \dots & a_{1n} \\ a_{21} & b_2 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & b_n & \dots & a_{nn} \end{vmatrix}}{\det\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}}, \quad \dots, \quad x_n = \frac{\det\begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & b_n \end{vmatrix}}{\det\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}}$$

Questo metodo comporta una complessità computazionale di circa $(n+1)!$ operazioni. Questo significa che per risolver un sistema con n=10 occorrerebbe fare 39916800 operazioni. Per n=20 , (n+1)!=10^{21}. Quindi se sono necessari 10^{-9} secondi per fare un prodotto, per risolvere un sistema di n=20 equazioni in n=20 incognite sono necessari 10^{12} secondi, che equivalgono a circa 1/3 di 10^5 anni.

Nella pratica quindi vengono utilizzati altri metodi per la soluzione di un sistema lineare.

2. Metodi numerici per la soluzione di un sistema lineare.

I metodi per la soluzione di sistemi lineari vengono usualmente divisi in due raggruppamenti:

1) **Metodi diretti** ⇒ Questi metodi, in assenza di errori di arrotondamento, conducono alla soluzione esatta in un numero finito di passi. Essi sono adatti per la soluzione di sistemi con **matrice dei coefficienti densa e di moderate dimensioni**.

2) **Metodi iterativi** ⇒ Questi metodi generano una successione di soluzioni, che, sotto opportune ipotesi, convergono alla soluzione del sistema. La matrice dei coefficienti non viene modificata durante il calcolo e quindi è più agevole sfuttarne la sparsità. Sono adatti, quindi, per la soluzione di sistemi con **matrice dei coefficienti di grandi dimensioni e sparsa**. In assenza di errori di arrotondamento conducono alla soluzione esatta in un numero infinito di passi.

Durante il nostro corso ci interesseremo in particolare dei metodi iterativi per la soluzione di sistemi lineari. Di seguito solo un breve accenno al funzionamento dei metodi diretti, che si basano sulla fattorizzazione della matrice dei coefficienti, soffermandoci sulle ipotesi di applicabilità, complessità computazionale e stabilità degli algoritmi su cui si basano

3. Metodi diretti.

I metodi diretti trasformano attraverso un numero finito di passi/operazioni un sistema *lineare generico in un sistema lineare equivalente dotato di una struttura particolare che* ne semplifichi la risoluzione.

Sono basati sulla fattorizzazione di A nel prodotto di due matrici B e C.

Obiettivo della fattorizzazione A = BC è appunto quello di trasformare il sistema lineare Ax = b in un sistema lineare equivalente, la cui risoluzione è più semplice.
Il sistema lineare che si ottiene è della forma

$$BC\,x = b$$

che si può spezzare in due problemi di più facile soluzione

$$\begin{cases} By = b \\ Cx = y \end{cases}.$$

$Cx = y$ con C triangolare superiore (per i tre metodi che tra poco introdurremo), per cui facilmente risolvibile con il metodo delle sostituzioni all'indietro. Il termine noto y può essere ottenuto come la soluzione del sistema lineare By = b dove B è triangolare inferiore (per due dei metodi che tra poco introdurremo) o ortogonale (per il terzo metodo)
Il costo computazionale della fattorizzazione è $O(n^3)$ mentre il costo della risoluzione dei sistemi By = b e Cx = y è $O(n^2)$.

3.1 Metodi diretti e fattorizzazioni associate

1) **Metodo di eliminazione gaussiana**: associato alla **fattorizzazione LU**
A = LU dove L è triangolare inferiore con elementi diagonali 1 e U triangolare superiore.
2) **Metodo di Cholesky** : associato alla **fattorizzazione LL^T o R^TR**
$A = LL^T = R^TR$ dove L è triangolare inferiore con elementi diagonali positivi mentre R è triangolare superiore con elementi diagonali positivi.
3) **Metodo di Householder**: associato alla **fattorizzazione QR**
A = QR dove Q è ortogonale ($Q^{-1} = Q^T$) e R è triangolare superiore.
Osservazioni:
- La fattorizzazione QR esiste sempre
- La fattorizzazione LL^T esiste per matrici simmetriche e definite positive.
- A differenza delle fattorizzazioni LU e LL^T , la fattorizzazione QR non è unica.

Fattorizzazione LU (di Gauss)

Fra i vari tipi esistenti di algoritmi di fattorizzazione riveste un ruolo importante la fattorizzazione LU di Gauss di una matrice A.

Teorema 1: Data $A \in M(nxn)$, sia A_k la sottomatrice principale di testa di A ottenuta considerando le prime k righe e le prime k colonne di A. Se A_k è non singolare per ogni **k=1,..,n-1** allora esiste ed è unica la fattorizzazione LU di A**.**

N.B. Non è esplicitamente richiesta la non singolarità di A per scrivere la fattorizzazione LU.

Con l'ipotesi aggiuntiva che la matrice A abbia determinante diverso da zero allora la fattorizzazione LU può essere utilizzata per risolvere un sistema lineare.

La fattorizzazione LU si basa sull'algoritmo di eliminazione di Gauss, che consiste nel sottoporre il sistema lineare che dobbiamo risolvere ad opportune trasformazioni in modo tale da eliminare successivamente le incognite dalle varie equazioni fino a ridursi ad un sistema triangolare superiore con matrice dei coefficienti U, e termine noto trasformato y, e quindi costruire la matrice L, triangolare inferiore, memorizzando i moltiplicatori utilizzati per trasformare la matrice del sistema A in triangolare superiore.

Complessità computazionale dell'algoritmo di fattorizzazione di Gauss: $\cong \frac{1}{3}\boldsymbol{n^3}$.

Sfruttando il teorema 1 , la soluzione del sistema lineare Ax=b, con A matrice che soddisfa le ipotesi del teorema , si riduce alla soluzione di due sistemi lineari

$$\begin{cases} Ly = b \\ Ux = y \end{cases}.$$

Algoritmo di fattorizzazione di Gauss

Si inizializza la matrice L all'identità.

$$k = 1, \dots, n-1 \qquad \begin{cases} l_{ik} = \dfrac{a_{ik}}{a_{kk}} & i = k+1, \dots, n \\ a_{ij} = a_{ij} - l_{ik} a_{kj} & i, j = k+1, \dots, n \end{cases}$$

Alla fine dell'algoritmo nel triangolo superiore di A viene memorizzato il fattore triangolare superiore U e nella matrice L, che risulterà triangolare inferiore con elementi 1 sulla diagonali, il fattore triangolare inferiore della fattorizzazione.

Def. *Matrice di permutazione P:*

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Questa matrice è ottenuta dalla matrice identità scambiando due righe tra di loro (in questo esempio la prima riga con la terza riga). P prende il nome di **matrice di permutazione.**

Effettuare il prodotto $P \cdot A$ equivale a scambiare le stesse due righe della matrice A (in questo caso la prima riga con la terza riga).

Effettuare il prodotto $A \cdot P$ equivale a scambiare la prima colonna della matrice A con la terza colonna.

Il prodotto di matrici di permutazione è ancora una matrice di permutazione.

Teorema 2: Data una qualunque matrice A non singolare, esiste una matrice di permutazione P non singolare t.c. PA = LU.

N.B. P non è unica, ossia possono esistere diverse matrici P t.c. PA soddisfa le ipotesi del Teorema 1.

Dal punto di vista algoritmo questo porta all'algoritmo di Gauss con pivotaggio.

Al passo *k*, prima di calcolare il moltiplicatore l_{ik}, se $a_{kk} = 0$, si va a cercare nella colonna k-esima, a partire dalla riga k-esima, la posizione di riga *s* in cui si trova il primo elemento diverso da zero.

Se la riga *s è diversa dalla riga k-esima*, si effettua lo scambio tra la riga *k*-esima e la riga *s*-esima della matrice, e poi si procede secondo lo schema dell'algoritmo classico.

Questa scelta garantisce l'esistenza della fattorizzazione LU, per la quale è sufficiente che $a_{kk} \neq 0,\ k = 1, \ldots, n-1$

Esiste un'altra variante dell'algoritmo di Gauss, con **pivotaggio per colonne a perno massimo** (implementata nel pacchetto scipy.linalg di Python) , che al passo *k*, prima di calcolare il moltiplicatore l_{ik}, va a cercare nella colonna *k*-esima, a partire dalla riga *k*-esima, la posizione di riga *s* in cui si trova l'elemento di modulo massimo.

Inizializza la matrice P all'identità

$for\ k = 1, \ldots, n-1$

Calcola nella colonna k-esima, a partire dall'elemento (k,k) l'indice di riga *s* a cui appartiene il massimo in valore assoluto.

Se $s \neq k$

Scambia la riga *s* con la riga *k*, memorizza lo scambio nella matrice P (viene fatto scambiando nelle matrice P la riga *s* con la riga *k*)

$$l_{ik} = \frac{a_{ik}}{a_{kk}} \qquad i = k+1, \ldots, n$$

$$a_{ij} = a_{ij} - l_{ik} a_{kj} \qquad i, j = k+1, \ldots, n$$

Vedremo che per la stabilità della fattorizzazione è meglio utilizzare la variante del pivotaggio per colonne a perno massimo, in modo tale che gli elementi di L risultano tutti minori od uguali ad 1.

In questo caso il sistema lineare $\boldsymbol{Ax} = \boldsymbol{b}$ diventa $\boldsymbol{P}\,\boldsymbol{Ax} = \boldsymbol{P} \cdot \boldsymbol{b}$, cioè gli scambi effettuati premoltiplicando la matrice A per P, devono essere applicati anche al termine noto:

$$\boldsymbol{P} \cdot \boldsymbol{Ax} = \boldsymbol{P} \cdot \boldsymbol{b}$$

Poichè $P\,A = LU$, il sistema diventa

$$\mathrm{LU}\,\boldsymbol{x} = \boldsymbol{P} \cdot \boldsymbol{b}$$

da cui

$$\begin{cases} Ly = P \cdot b \\ Ux = y \end{cases}.$$

Fattorizzazione di Cholesky

Per la fattorizzazione delle matrici simmetriche e definite positive è stato studiato un algoritmo di fattorizzazione, detto di Cholesky, che deriva dal seguente teorema:

Teorema di Cholesky.

Sia A una matrice di ordine n simmetrica e definita positiva, allora esiste una matrice triangolare inferiore L con elementi diagonali positivi, ($(l_{ii} > 0 \quad i = 1, \dots n)$ tale che

$$A = L \cdot L^T$$

Complessità computazionale dell'algoritmo di fattorizzazione di Cholesky: $\cong \frac{1}{6}n^3$.
Sfruttando questo teorema, la soluzione del sistema lineare Ax=b, con A matrice simmetrica e definita positiva, si riduce alla soluzione di due sistemi lineari

$$\begin{cases} Ly = b \\ L^T x = y \end{cases}.$$

Fattorizzazione QR di una matrice

Sia $A \in M(m \times n)$ con $m \geq n$ e rank(A) = n (ossia le colonne di A sono linearmente indipendenti). Allora esistono una matrice $Q \in M(m \times m)$ ortogonale e una matrice $R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \in M(m \times n)$ dove $R_1 \in M(n \times n)$ è una matrice triangolare superiore non singolare, tale che A = QR.

Sfruttando questo teorema, la soluzione del sistema lineare Ax=b, con A matrice quadrato e a rango massimo , si riduce alla soluzione di due sistemi lineari

$$\begin{cases} Qy = b \\ Rx = y \end{cases}.$$

Essendo Q ortogonale la soluzione del sistema Qy=b si riduce a $y = Q^T B$, (poiché l'inversa di una matrice ortogonale coincide con la sua inversa).

Complessità computazionale:

$caso\ m - 1 \geq n \rightarrow \quad \cong \quad mn^2 - n^3/3$

$caso\ m = n \rightarrow \quad \cong \quad 2n^3/3$

Stabilità di un algoritmo di fattorizzazione

Consideriamo la fattorizzazione

$$A = B \cdot C$$

e studiamo l'effetto che ha sui risultati il fatto che questa fattorizzazione venga eseguita operando con i numeri finiti; poiché le operazioni aritmetiche che compaiono in uno degli algoritmi visti per calcolare la fattorizzazione di una matrice *A* vengono eseguite in aritmetica finita, alla fine dell'algoritmo si ottengono, anziché i fattori esatti *B* ed *C* i fattori $\mathcal{B}$ ed $\mathcal{C}$; questi si possono pensare come

$$\mathcal{B}=B+\delta B \qquad \mathcal{C}=C+\delta C, \qquad (2)$$

cioè dati dai fattori esatti più una piccola perturbazione.

Utilizzando la filosofia dell'<u>analisi all'indietro</u> introdotta da Wilkinson, i fattori $\mathcal{B}$ e $\mathcal{C}$ possono essere pensati come <u>fattorizzazione esatta di una matrice perturbata:</u>

$$A+\delta A= \mathcal{B}\cdot \mathcal{C} \qquad (3)$$

Si studia, perciò, da cosa dipenda l'entità della perturbazione δA.

Dalle relazioni (2) e (3) si ottiene

$$A+\delta A= (B+\delta B) \cdot (C+\delta C)$$

cioè

$$A+\delta A= B\cdot C+B\cdot\delta C + \delta B\cdot C +\delta B\cdot\delta C$$

da cui segue per δA la seguente espressione:

$$\delta A= B\cdot\delta C + \delta B\cdot C +\delta B\cdot\delta C$$

Questa relazione mette in evidenza che la perturbazione δA, non solo dipende dalle piccole perturbazioni δB e δC, ma è tanto più grande quanto più grandi sono gli elementi dei fattori B ed C.
Questa osservazione porta al fatto che si definisce la stabilità della fattorizzazione B·C in termini degli elementi di B e di C.

Definizione di stabilità di un algoritmo di fattorizzazione B·C

Data una matrice A i cui elementi sono tutti minori od uguali ad 1, si dice che un algoritmo di fattorizzazione che produce una fattorizzazione B·C della matrice A è numericamente stabile in senso forte, se esistono delle costanti positive *a* e *b*, indipendenti dall'ordine e dagli elementi di *A* tali che

$$|b_{ij}| \leq a \quad |c_{ij}| \leq b$$

Se le costanti a e b dipendono dall'ordine di A si dice che la fattorizzazione B·C è stabile in senso debole.

Stabilità dell'algoritmo di fattorizzazione di Gauss, $A = L \cdot U$

Nel caso in cui si utilizzi la tecnica di pivotaggio a perno massimo per colonne, che nasce da un'opportuna scelta della matrice di Permutazione P del Teorema 2, si ha

$$|l_{ij}| \leq 1$$

$$|u_{ij}| \leq 2^{n-1} \, max|a_{ij}|$$

L'algoritmo di fattorizzazione di Gauss è stabile in senso debole perché la costante che maggiora gli elementi di L non dipende dall'ordine della matrice, mentre ciò non accade per la costante che maggiora gli elementi di U, che dipende in maniera esponenziale dall'ordine della matrice.

Stabilità dell'algoritmo di fattorizzazione di Cholesky, $A = L \cdot L^T$

$$\max_{ij}|l_{ij}| \leq \sqrt{\max_{ij}|a_{ij}|}$$

L'algoritmo di fattorizzazione di Cholesky è stabile in senso forte.

Stabilità dell'algoritmo di fattorizzazione QR, A=Q·R

$$|q_{ij}| \leq 1 \quad |r_{ij}| \leq \sqrt{n}\max_{ij}|a_{ij}|$$

L'algoritmo di fattorizzazione Q·R è stabile in senso debole, perché gli elementi di R in valore assoluto sono maggiorati da $\sqrt{n}\max_{ij}|a_{ij}|$, ma è più stabile della fattorizzazione LU di Gauss, per la quale gli elementi di U in valore assoluto sono maggiorati da $2^{n-1}\max_{ij}|a_{ij}|$,

Soluzione di sistemi con matrici triangolari

Sia $L = \begin{bmatrix} l_{11} & 0 & \dots & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & & \\ \dots & \dots & \dots & \dots & \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix}$ una matrice *triangolare inferiore* ed

$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix}$ una matrice *triangolare superiore*

La soluzione di sistemi lineari con queste matrici dei coefficienti si ottiene facilmente mediante sostituzione.

Soluzione del sistema Lx=b, nel caso in cui la matrice L sia triangolare inferiore

$$
\begin{cases}
l_{11}x_1 & = b_1 \\
l_{21}x_1 + l_{22}x_2 & = b_2 \\
\dots & \\
l_{i1}x_1 + l_{i2}x_2 + \cdots + l_{ii}x_i & = b_i \\
\dots & \\
l_{n1}x_1 + l_{n2}x_2 + \dots + l_{nn}x_n & = b_n
\end{cases}
$$

Sostituzione in avanti (Forward Substitution)

$$
\begin{cases}
x_1 = \dfrac{b_1}{l_{11}} \\
x_2 = (b_2 - l_{21}x_1)/l_{22} \\
\dots \\
x_i = (b_i - (l_{i1}x_1 + l_{i2}x_2 + \dots l_{i,i-1}x_{i-1}))/l_{i,i} \qquad i = 2, \dots, n
\end{cases}
$$

Cioè, in forma algoritmica:

```
for  i=1,2,..,n
    xi=bi
        for j=1,2,...,i-1
          xi=xi-lij·xj
        end for j
   xi=xi/lii
end for i
```

Soluzione del sistema Ux=b, nel caso in cui la matrice U sia triangolare superiore

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + \cdots + u_{1,i}x_i + \cdots & u_{1,n-1}\,x_{n-1} + u_{1n}x_n = b_1 \\ & u_{i,i}x_i + u_{i,i+1}x_{i+1} + \cdots + u_{i,n}\,x_n = b_i \\ & u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1} \\ & u_{nn}x_n = b_n \end{cases}$$

Sostituzione all'indietro (Backward Substitution)

$$\begin{cases} x_n = \dfrac{b_n}{u_{n,n}} \\ x_{n-1} = (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ \dots \\ x_i = (b_i - (u_{i,i+1}x_{i+1} + u_{i,i+2}x_{i+2} + \dots u_{i,n}x_n))/u_{i,i} \qquad i = n-1, n-2, \dots, 1 \end{cases}$$

cioè, in forma algoritmica,

```
for  i=n,n-1,...,1
    xi=bi
        for j=i+1,...,n
            xi=xi-uij·xj
        end for j
xi=xi/uii
end for i
```

Complessità computazionale dell'algoritmo di Forward Substitution (e Backward Substitution) :

La complessità computazionale in termini di operazioni moltiplicative è :

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, quindi dell'ordine di $O\left(\frac{n^2}{2}\right)$.

Calcolo matrice inversa

Il problema di determinare l'inversa A^{-1}di una matrice A quadrata nxn non singolare si riconduce al problema di risolvere n sistemi normali.

$$A \cdot A^{-1} = I$$

Poiché l'i-esima colonna della matrice identità è l'i-esimo vettore della base canonica

$$e_i = \begin{bmatrix} 0 \\ 0 \\ .. \\ 0 \\ 1 \\ 0 \\ ... \\ ... \end{bmatrix}$$ **i-esima riga**

Ed indicando con x_i l'i-esima colonna della matrice inversa A^{-1}, il calcolo dell'inversa equivale a risolvere *n* sistemi lineari con la stessa matrice A (si considera la sua fattorizzazione e la si utilizza per risolvere l'i-esimo sistema lineare) :

$$Ax_i = e_i \qquad i = 1, \dots n$$

La soluzione dell'i-esimo sistema lineare rappresenta l'i-esima colonna della matrice Inversa.

Si può effettuare la fattorizzazione LU della matrice PA, che rimane la stessa per tutti i sistemi lineari e poi utilizzarla per risolvere gli n sistemi lineari:

$$\begin{cases} Ly_i = Pe_i \\ Ux_i = y_i \end{cases} \qquad i = 1,..n$$

La soluzione x_i dell'i-esimo sistema lineare così ottenuta rappresenta l'i-esima colonna della matrice inversa

Calcolo del determinante di una matrice sfruttando la fattorizzazione LU di PA

PA= L U con L matrice triangolare inferiore con elementi diagonali uguali ad 1

U matrice triangolare superiore

$$\det(PA) = \det(LU) = \det(L)\det(U) = \det(U) = \prod_{i=1\ldots n} u_{ii}$$

$$\det(PA) = \det(P)\det(A) = \prod_{i=1\ldots n} u_{ii}$$

$$\det(P) = (-1)^s$$

dove s è il numero di scambi effettuati.

Quindi $\det(A) = (-1)^s \prod_{i=1\ldots n} u_{ii}$

Il rango è dato invece dal numero r degli elementi non nulli sulla diagonale di U

4. Metodi iterativi per la soluzione di sistemi lineari

Per la risoluzione di un sistema lineare $Ax = b$, oltre ai metodi diretti, è possibile utilizzare anche i metodi iterativi che raggiungono la soluzione esatta come limite di un procedimento iterativo.

Mentre i metodi diretti si basano su una fattorizzazione della matrice A e hanno una complessità di $O(n^3)$ i metodi iterativi si basano su una decomposizione della matrice A e presentano una complessità di $O(kn^2)$, dove k è il numero di iterazioni.

Pertanto tali metodi risultano particolarmente convenienti quando la matrice A del sistema è di grandi dimensioni e sparsa. Infatti, quando la matrice A è sparsa, cioè il numero degli elementi non nulli è di molto inferiore al numero degli elementi nulli, applicando i metodi diretti può accadere che vengano generati elementi non nulli in corrispondenza degli elementi nulli della matrice di partenza (fenomeno del **fill-in**).

Questo non avviene applicando i metodi iterativi in quanto essi si limitano ad utilizzare gli elementi non nulli della matrice senza toccare gli elementi nulli.

4.1 Metodi iterativi basati sulla decomposizione di A

Sia A matrice di ordine n e sia dato il sistema

$$Ax = b, \qquad \text{con } \det A \neq 0. \tag{1}$$

Una famiglia di metodi iterativi per la soluzione del sistema lineare (1) si ottiene utilizzando una **decomposizione della matrice *A*** nella forma

$$A = M - N \qquad \text{con } \det M \neq 0.$$

In tal modo il sistema (1) diventa

$$(M-N)\, x = b,$$

cioè

$$Mx = Nx + b$$

e quindi si ottiene

$$x = M^{-1}Nx + M^{-1}b \tag{1.1}$$

La soluzione di (1) risolve anche (1.1)

Questa formulazione suggerisce di considerare metodi iterativi le cui iterazioni siano fornite dalle iterazioni successive

$$x^{(k)} = M^{-1}Nx^{(k-1)} + M^{-1}b \qquad \text{per } k=1,2,\ldots.$$

L'idea dei metodi iterativi è quindi quella di partire da un vettore $x^{(0)}$ iniziale arbitrario, stima iniziale della soluzione del sistema (1) e di costruire una successione di iterati $x^{(k)}$ mediante il seguente procedimento iterativo

$$\boxed{x^{(k)} = Tx^{(k-1)} + q} \qquad \text{per } k=1,2,\ldots. \quad (2)$$

dove $T= M^{-1}N$, è detta la **matrice di iterazione** del metodo iterativo e $q= M^{-1}b$.

I metodi iterativi della forma (2) sono detti usualmente iterativi stazionari, perché T e q non dipendono dall'indice di iterazione k.

4.2 Esempi di metodi iterativi basati sulla decomposizione di *A*

1) **Il metodo di Jacobi** (o degli spostamenti simultanei)

2) **Il metodo di Gauss-Seidel** (o degli spostamenti successivi).

In entrambi i metodi si considera la matrice A del sistema decomposta come somma di 3 matrici, cioè

$$A = D + E + F$$

con

$$D = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}; \quad E = \begin{bmatrix} 0 & & & 0 \\ a_{21} & 0 & & \\ a_{31} & & \ddots & \\ \vdots & & & 0 \\ a_{n1} & & a_{nn-1} & 0 \end{bmatrix}; \quad F = \begin{bmatrix} 0 & a_{12} & & a_{1n} \\ & 0 & & \\ & & \ddots & \\ & & & a_{n-1n} \\ & & & 0 \end{bmatrix}.$$

e si suppone che $a_{ii} \neq 0$ i=1,2,..n.

Metodo di Jacobi

Nel metodo di Jacobi la decomposizione di A nella forma A=M-N si ottiene scegliendo **M=D** e **N= - (E+F)**.

Pertanto il procedimento iterativo (2) diviene

$$x^{(k)} = -D^{-1}(E+F)\,x^{(k-1)} + D^{-1}b \qquad \text{per } k=1,2,.. \quad (3)$$

In termini di componenti la (3), equivale a calcolare la i-esima componente dell'iterato k-esimo come

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j\neq i}}^{n} -a_{ij}x_j^{(k-1)} + b_i}{a_{ii}} \qquad i = 1,2..,n, \quad (4)$$

in quanto la matrice D^{-1} è una matrice diagonale con elementi uguali ai reciproci degli elementi di D.

La matrice di iterazione del metodo di Jacobi è quindi data da

$$\boldsymbol{T_J = M^{-1}N = -D^{-1}(E+F)}$$

Osservazioni:

1) L'algoritmo di Jacobi è definito se gli elementi diagonali di *A* sono diversi da 0, cioè $a_{ii} \neq 0$. In caso contrario, sempre sotto l'ipotesi che *A* sia non singolare, si possono riordinare le equazioni, e le incognite del sistema, in modo da rendere il metodo definito.
2) Nel metodo di Jacobi ogni elemento dell'iterato (k) è indipendente dagli altri, pertanto il metodo è programmabile in forma parallela.

Esempio:

Si consideri il seguente sistema lineare:

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

Il metodo di Jacobi consiste nello scegliere un vettore iniziale $x^0 = (x_1^0, \ldots, x_4^0)$ e nel ricavare

$$\begin{aligned} x_1^{(1)} &= (x_2^{(0)} - 2x_3^{(0)} + 6)/10 \\ x_2^{(1)} &= (x_1^{(0)} + x_3^{(0)} - 3x_4^{(0)} + 25)/11 \\ x_3^{(1)} &= (-2x_1^{(0)} + x_2^{(0)} + x_4^{(0)} - 11)/10 \\ x_4^{(1)} &= (-3x_2^{(0)} + x_3^{(0)} + 15)/8 \end{aligned}$$

Si va poi avanti in questo modo finché non si ritiene di essere sufficientemente vicini alla soluzione. Occorre pertanto definire un criterio d'arresto.

Metodo di Gauss-Seidel

Nel metodo di Gauss-Seidel la decomposizione di *A* nella forma *A=M-N* si ottiene scegliendo ***M=E+D*** **e** ***N= - F***. In questo caso la matrice di iterazione del metodo di

Gauss Seidel è data da $T_G = M^{-1}N = -(E+D)^{-1}F$ ***e la soluzione al passo k si ottiene come***

$$x^{(k)} = -(E+D)^{-1}F\, x^{(k-1)} + (E+D)^{-1}b \qquad \text{per } k=1,2,.. \qquad (5)$$

Per esprimere la soluzione in termini di componenti, partiamo da

$$M\, x^{(k)} = N\, x^{(k-1)} + b$$

Sostituendo ad M la matrice E+D e ad N la matrice –F, otteniamo

$$(E+D)\, x^{(k)} = -F\, x^{(k-1)} + b \qquad \text{per } k=1,2,.. \qquad (5.1)$$

$$D\, x^{(k)} = -E\, x^{(k)} \; -F\, x^{(k-1)} + b \qquad \text{per } k=1,2,..$$

da cui

$$x^{(k)} = -\, D^{-1}\, (E\, x^{(k)} + F\, x^{(k-1)} - b\,) \qquad \text{per } k=1,2,.. \qquad (6)$$

e, in termini di componenti,

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i}{a_{ii}} \qquad i = 1,2..,n \qquad (7)$$

Questo metodo ha la caratteristica di utilizzare, per calcolare la nuova componente i-esima di un iterato, le nuove componenti già calcolate dell'iterato stesso.

Osservazione:

Il metodo di Gauss-Seidel non si presta ad essere parallelizzato in quanto ogni nuova componente dell'iterato (k) dipende da tutte le nuove componenti dello stesso iterato che sono state appena calcolate.

Osservazione:

La (5.1) suggerisce che la soluzione al passo k si ottiene risolvendo il sistema triangolare inferiore avente (D+E) come matrice dei coefficienti e termine noto $b - F\, x^{(k-1)}$.

Esempio:

Si consideri il sistema lineare dell'esempio precedente; partendo dall'iterato iniziale $x^0 = (x_1^0, ..., x_4^0)$ si ottiene:

$$x_1^{(1)} = (x_2^{(0)} - 2x_3^{(0)} + 6)/10$$
$$x_2^{(1)} = (x_1^{(1)} + x_3^{(0)} - 3x_4^{(0)} + 25)/11$$
$$x_3^{(1)} = (-2x_1^{(1)} + x_2^{(1)} + x_4^{(0)} - 11)/10$$
$$x_4^{(1)} = (-3x_2^{(1)} + x_3^{(1)} + 15)/8$$

e si continua così fino a convergenza.

4.3 Convergenza

Le prime domande a cui dobbiamo dare risposta sono: la successione di iterati $x^{(k)}$ è una successione convergente? In caso affermativo, il suo limite è proprio la soluzione x del sistema (1)?

Definizione: Convergenza

Il procedimento iterativo

$$x^{(k)} = Tx^{(k-1)} + q$$

*si dice "**convergente**" se, per ogni vettore iniziale $x^{(0)}$, la successione $\{x^{(k)}\}$ converge ad un vettore limite y;*

$$\lim_{k\to\infty} x^{(k)} = y$$

cioè se, per ogni $\varepsilon>0$, esiste un indice ν tale che per ogni $k>\nu$ si ha

$$|| x^{(k)} - y|| \leq \varepsilon.$$

Il seguente teorema dà una risposta affermativa alla seconda delle domande che ci eravamo posti.

Teorema:

Se il sistema (1) ammette un' unica soluzione x e se il processo iterativo (2) è convergente, allora il vettore limite y coincide con la soluzione x, cioè

$$\lim_{k\to\infty} x^{(k)} = x.$$

Dimostrazione:

Si parte da

$$Mx^{(k)} = Nx^{(k-1)} + b$$

e si considera il limite per $k \to \infty$ di entrambi i membri

$$\lim_{k\to\infty} M\, x^{(k)} = \lim_{k\to\infty} \left(N\, x^{(k-1)} + b\right)$$

Poichè $x^{(k)}$ è per ipotesi una successione convergente ad y segue che

$$My = Ny + b$$

da cui si ottiene $(M-N)y = b$. Essendo $M-N=A$, si ha $Ay=b$ e quindi $y=x$, poiché per ipotesi AX=b ha un'unica soluzione x.

4.4 Convergenza dei Metodi Iterativi

Per effettuare lo studio della convergenza di un metodo iterativo si considera la matrice di iterazione *T*, tipica del metodo stesso.

Inoltre si definisce l' **errore commesso al passo *k*** come il vettore

$$e^{(k)} = x^{(k)} - x, \quad k = 0,1,....$$

e il **vettore residuo al passo *k*** dato da

$$r^{(k)} = Ax^{(k)} - b, \quad k = 0,1,....$$

Si osservi che queste due quantità sono legate dalla seguente relazione:

$$r^{(k)} = Ax^{(k)} - b = Ax^{(k)} - Ax = A(x^{(k)} - x) = Ae^{(k)}.$$

Consideriamo ora la relazione relativa al valore esatto x

$$Mx = Nx + b$$

e la relazione analoga, al generico passo k,

$$M\,x^{(k)} = Nx^{(k-1)} + b.$$

Sottraendo la prima dalla seconda, otteniamo $Me^{(k)} = Ne^{(k-1)}$ da cui

$$e^{(k)} = M^{-1}Ne^{(k-1)} = Te^{(k-1)} = T^2e^{(k-2)} = \ldots = T^ke^{(0)}$$

Affinché il procedimento sia convergente si deve avere che, comunque si sceglie $x^{(0)}$, ciascuna componente del vettore $e^{(k)}$ tenda a 0 per k che tende a ∞.

Questo equivale a cercare delle condizioni per cui

$$\lim_{k\to\infty} T^k e^{(0)} = 0$$

che equivale a

$$\lim_{k\to\infty} T^k = 0.$$

<u>Teorema:</u> Condizione necessaria e sufficiente alla Convergenza

Sia A=M-N una matrice di ordine n, con det(A)≠0, e $T=M^{-1}N$ la matrice di iterazione del procedimento iterativo (2). Condizione necessaria e sufficiente per la convergenza del procedimento iterativo, comunque si scelga il vettore iniziale $x^{(0)}$, al vettore soluzione x del sistema Ax=b , è che

$$\rho(T) < 1$$

ovvero che il raggio spettrale (che corrisponde all'autovalore di modulo massimo) della matrice di iterazione T sia minore di 1.

4.5 Condizioni Sufficienti per la Convergenza:

Il calcolo del raggio spettrale della matrice di iterazione T è, anche utilizzando algoritmi idonei, piuttosto oneroso. Si preferisce quindi o considerare condizioni anche più restrittive, ma facilmente verificabili, o individuare classi di matrici per cui

la convergenza è garantita da risultati teorici. I seguenti teoremi servono a questi scopi.

Il teorema che segue ci garantisce una condizione sufficiente alla convergenza.

Teorema:

Se, per una qualche norma, risulta $||T||<1$*, allora il processo iterativo*

$$x^{(k)} = Tx^{(k-1)} + M^{-1}b \qquad per\ k=1,2,....$$

è convergente per ogni $x^{(0)}$.

Dim:

Dalla definizione di autovalore di una matrice si ha $Tx=\lambda x$, con $x\neq 0$, da cui si ottiene

$$|\lambda|\ ||x|| = ||Tx|| \leq ||T||\ ||x||$$

e quindi

$$|\lambda| \leq ||T||.$$

Ogni autovalore in modulo è minore o uguale della norma della matrice T.

Poiché, per ipotesi, si ha $||T||<1$ allora anche il raggio spettrale di T risulta minore di 1, $\rho(T) < 1$ e, per il teorema precedente, il procedimento iterativo è convergente.

Seguono ora due teoremi che garantiscono la convergenza per classi particolari di matrici.

Teorema:

Se la matrice A è a ***diagonale strettamente dominante****, cioè*

$$|a_{ii}| > \sum_{\substack{k=1 \\ k\neq i}}^{n} |a_{ik}| \quad i=1,2,....,n$$

Allora sia il metodo di Jacobi che quello di Gauss-Seidel convergono e si ha

$$||T_G|| \leq ||T_J|| < 1$$

Teorema:

Se la matrice A è simmetrica e definita positiva, il metodo di Gauss-Seidel è convergente.

4.6 . Velocità di convergenza e condizionamento di una matrice

Un procedimento iterativo, convergente se l'autovalore di modulo massimo è minore di 1, è tanto più velocemente convergente quanto più piccolo è l'autovalore di modulo massimo della matrice di iterazione *T*.

Un problema strettamente legato alla convergenza del procedimento iterativo è quello del mal condizionamento della matrice A. L'influenza del condizionamento di A è infatti responsabile del rallentamento o addirittura della perdita della convergenza nei metodi iterativi.

Esempio:

$$A = \begin{bmatrix} .96326 & .81321 \\ .81321 & .68685 \end{bmatrix} \qquad b = \begin{bmatrix} .88824 \\ .74988 \end{bmatrix}$$

A è una matrice simmetrica definita positiva.

Il metodo di Gauss-Seidel, partendo dai valori iniziali $x^{(0)} = \begin{bmatrix} .33116 \\ .70000 \end{bmatrix}$ non converge alla soluzione

$$x^* = \begin{bmatrix} .39473 \\ .62470 \end{bmatrix}.$$

Questo comportamento si verifica per qualsiasi scelta di $x_2^{(0)}$ compresa tra .4 e .8.

Infatti l'indice di condizionamento di A è K(A)= 8936.

4.7 Accelerazione di un metodo iterativo

Ora ci si chiede se è possibile accelerare la convergenza di un metodo iterativo e addirittura se è possibile rendere convergente un metodo che non lo è. Come risposta a questo quesito è nata una famiglia di metodi nota come **metodi di rilassamento**.
L'idea alla base di questi metodi è la seguente: poiché la velocità di convergenza di un metodo iterativo dipende dal raggio spettrale della matrice di iterazione associata al metodo, un modo per cercare di accelerare la convergenza è quello di far dipendere la matrice di iterazione da un parametro, detto **parametro di rilassamento,** e di scegliere tale parametro in modo tale che la matrice abbia minimo raggio spettrale.

Vediamo come viene generato un metodo di rilassamento a partire dal metodo di Gauss-Seidel.
Il metodo di Gauss-Seidel

$$x^{(k)} = - D^{-1} (E\, x^{(k)} + F\, x^{(k-1)} - b\,) \qquad \text{per } k=1,2,..$$

può essere riscritto nella forma

$$x^{(k)} = x^{(k-1)} + r^{(k)}$$

dove

$$r^{(k)} = x^{(k)} - x^{(k-1)} = -D^{-1}[Ex^{(k)} + Fx^{(k-1)} - b] - x^{(k-1)}. \qquad (8)$$

Modificando il metodo di Gauss-Seidel come

$$x^{(k)} = x^{(k-1)} + \omega r^{(k)} \qquad (9)$$

e scegliendo opportunamente il parametro $\omega>0$ si può accelerare la convergenza in modo significativo. A seconda di come viene scelto tale parametro si distinguono i metodi di rilassamento in

a) ***under-relaxation methods*** con $0< \omega<1$
b) ***over-relaxation methods*** $\omega>1$.

I metodi di ***under-relaxation*** possono essere usati per ottenere convergenza su certi sistemi per cui non si ha convergenza con il metodo di Gauss-Seidel, mentre i metodi di ***over-relaxation*** possono essere usati per accelerare la convergenza in sistemi in cui il metodo di Gauss-Seidel converge ma lentamente.

Questi ultimi metodi vengono chiamati metodi SOR (Successive Over-Relaxation) e vengono spesso impiegati per risolvere sistemi lineari che si incontrano nella soluzione numerica di alcune equazioni alle derivate parziali.

Il metodo SOR si ottiene sostituendo la (8) nella (9), ovvero:

$$x^{(k)} = (1-\omega)x^{(k-1)} - \omega D^{-1}[Ex^{(k)} + Fx^{(k-1)} - b], \qquad (10)$$

da cui si ricava

$$x^{(k)} = (1-\omega)x^{(k-1)} + \omega[-D^{-1}[Ex^{(k)} + Fx^{(k-1)} - b]], \qquad (11)$$

In termini di componenti si ha che la componente i-esima della soluzione al passo k, $x_i^{(k)}$, diviene

$$x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \right] \quad i = 1,2..,n .$$

Questa variante del metodo di Gauss Seidel, dipendente dal parametro ω, è detta anche metodo di Gauss-Seidel estrapolato e può essere vista come

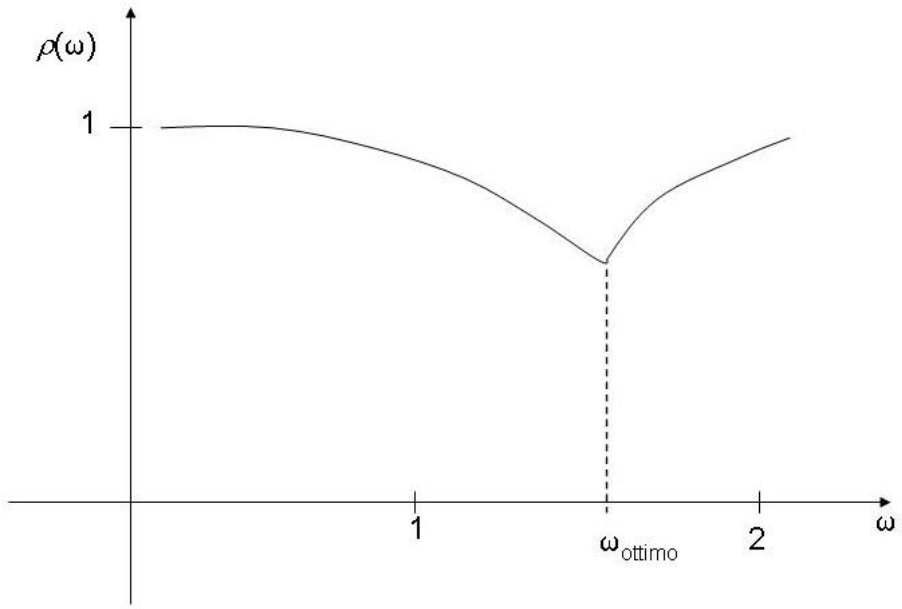
$$\begin{cases} \tilde{x}_i^{(k)} = \dfrac{1}{a_{ii}}\left[b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \right] \\ x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \omega\,\tilde{x}_i^{(k)} \end{cases}$$

La matrice di iterazione del metodo è

$$T_\omega = (D+\omega E)^{-1}[(1-\omega)D - \omega F]. \qquad (12)$$

Nell'applicazione di un metodo di rilassamento dipendente dal parametro nasce il problema della scelta ottimale del parametro ω che, oltre ad assicurare la convergenza, renda minimo il raggio spettrale della matrice di iterazione T_ω in modo da ottenere la massima velocità di convergenza.

E' noto infatti che, per matrici simmetriche definite positive e per $0 < \omega < 2$, il valore di $\rho(T_\omega)$ si mantiene <1, ma c'è un punto in cui è minimo. Il valore corrispondente al minimo del raggio spettrale è detto ω_{ottimo}.



In genere il calcolo di ω_{ottimo} è un problema molto laborioso, per cui si suole andare per tentativi.

4.8 Criterio d'arresto

Poiché con un metodo iterativo non è ovviamente possibile calcolare in generale la soluzione in un numero finito di iterazioni, occorre individuare dei criteri per l'arresto del procedimento. I criteri più comunemente usati, che consistono nel fissare una tolleranza ε che tiene conto della precisione utilizzata nei calcoli, sono i seguenti:

$$||x^{(k)} - x^{(k-1)}|| \leq \varepsilon \qquad (13)$$

oppure se $x^{(k)} \neq 0$

$$\frac{||x^{(k)} - x^{(k-1)}||}{||x^{(k)}||} \leq \varepsilon. \qquad (14)$$

La scelta della tolleranza ε nel criterio d'arresto viene fatta considerando la percentuale d'errore da cui sono affetti i dati iniziali.

La scelta del tipo di norma, in genere, dipende dallo specifico problema in esame; comunque norme comunemente usate sono la norma ∞ e la norma 2