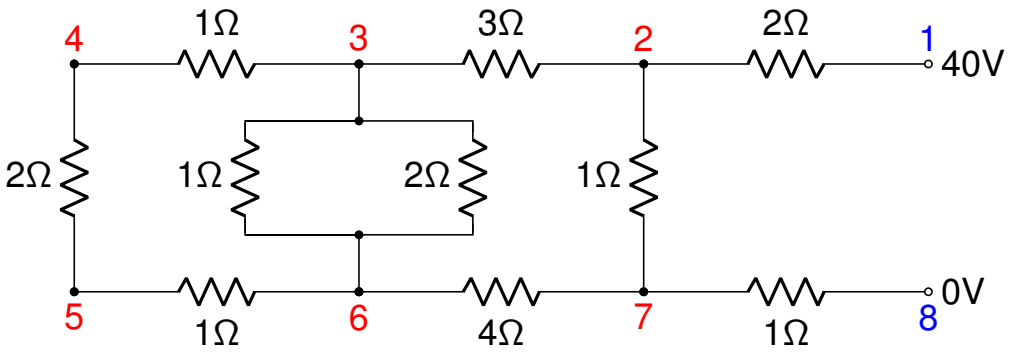
Un'applicazione di Fisica



In questo circuito elettrico, le resistenze sono indicate in ohm (Ω) e i potenziali in volt (V). Si denota la corrente che fluisce dal nodo i al nodo j con I_{ij}. Essa è misurata in ampère.
Per conoscere quali sono i potenziali in ogni nodo ci si serve

- della legge di Ohm:
$$\frac{V_i - V_j}{R_{i,j}} = I_{i,j}$$
- della legge di Kirkhoff: la somma algebrica di tutte le correnti che entrano in un nodo vale 0.

Un'applicazione di Fisica

Seguendo tale legge si mostra che

$$R_{3,6} = \frac{2}{3}$$

Pertanto il problema di conoscere correnti e potenziali nei nodi si risolve impostando il seguente sistema:

$$\begin{aligned}
&\text{nodo 2} && I_{1,2} + I_{7,2} + I_{3,2} = \frac{40 - V_2}{2} + \frac{V_7 - V_2}{1} + \frac{V_3 - V_2}{3} = 0\\
&\text{nodo 3} && I_{2,3} + I_{6,3} + I_{4,3} = \frac{V_2 - V_3}{3} + \frac{V_6 - V_3}{2/3} + \frac{V_4 - V_3}{1} = 0\\
&\text{nodo 4} && I_{3,4} + I_{5,4} = \frac{V_3 - V_4}{1} + \frac{V_5 - V_4}{2} = 0\\
&\text{nodo 5} && I_{4,5} + I_{6,5} = \frac{V_4 - V_5}{2} + \frac{V_6 - V_5}{1} = 0\\
&\text{nodo 6} && I_{5,6} + I_{3,6} + I_{7,6} = \frac{V_5 - V_6}{1} + \frac{V_3 - V_6}{2/3} + \frac{V_7 - V_6}{4} = 0\\
&\text{nodo 7} && I_{6,7} + I_{2,7} + I_{8,7} = \frac{V_6 - V_7}{4} + \frac{V_2 - V_7}{1} + \frac{0 - V_7}{1} = 0
\end{aligned}$$

Si ottiene così un sistema di 6 equazioni in 6 incognite $(V_2, V_3, V_4, V_5, V_6, V_7)^T$.

Come si può scrivere?

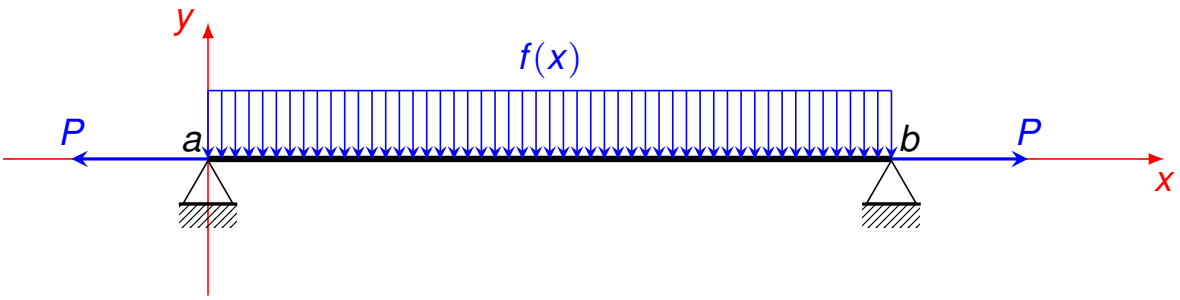
Con un po' di conti, si arriva a

$$\begin{cases} 11\,V_2 & -2\,V_3 & & & & -6\,V_7 = 120 \\ -2\,V_2 & +13\,V_3 & -2\,V_4 & & -9\,V_6 & = 0 \\ & -2\,V_3 & +3\,V_4 & -V_5 & & = 0 \\ & & -V_4 & +3\,V_5 & -2\,V_6 & = 0 \\ & -6\,V_3 & & -4\,V_5 & +11\,V_6 & -V_7 = 0 \\ -4\,V_2 & & & & -V_6 & +9\,V_7 = 0 \end{cases}$$

$$A = \begin{pmatrix} 11 & -2 & & & & -6 \\ -2 & 13 & -2 & & -9 & \\ & -2 & 3 & -1 & & \\ & & -1 & 3 & -2 & \\ & -6 & & -4 & 11 & -1 \\ -4 & & & & -1 & 9 \end{pmatrix} \qquad \boldsymbol{b} = \begin{pmatrix} 120 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Un'applicazione ingegneristica: discretizzazione di un problema ai limiti

Si consideri una trave, appoggiata ai suoi estremi a e b, tesa nella direzione del suo asse da una forza P e sottoposta a un carico trasversale $f(x)\mathrm{d}x$, per elemento di lunghezza $\mathrm{d}x$.



Il **momento flettente** nel punto di ascissa x è soluzione del problema ai limiti

$$-y''(x) + c(x)y(x) = f(x) \qquad x \in (a, b)$$

con $y(a)$ e $y(b)$ assegnati, ove $c(x)$ è definito dal rapporto $P/(E \cdot I(x))$, ove E è il modulo di Young dipendente dal materiale di cui è costituita la trave e $I(x)$ è il momento principale di inerzia della trave nel punto di ascissa x.
Sotto le ipotesi che $c(x)$ e $f(x)$ siano continue in $[a, b]$ e che $c(x) \geqslant 0$ in $[a, b]$, si dimostra che la soluzione $y(x)$ è univocamente determinata.

Modello discreto

Per risolvere il problema applichiamo il **metodo delle differenze finite**: esso consiste nel sostituire al dominio continuo un dominio discreto (un insieme di punti) e all?equazione differenziale una equazione alle differenze per ogni punto considerato, rimpiazzando le derivate con opportuni rapporti incrementali e risolvendo il sistema discretizzato così ottenuto.
Per discretizzare l'equazione, suddividiamo il dominio in $n+1$ sottointervalli di uguale ampiezza (per semplicità): $h = \dfrac{b-a}{n+1}$, $a = x_0 < x_1 < \ldots < x_n < x_{n+1} = b$, $x_i = a + ih$.
Poniamo $y_i = y(x_i)$, $y_i' = y'(x_i)$, $y_i'' = y''(x_i)$, $y_i''' = y'''(x_i)$, e, supposto $y(x) \in C^4\big([a,b]\big)$, dalla formula di Taylor di punto iniziale x_i osserviamo che

$$y_{i-1} = y_i - hy_i' + (h^2/2)y_i'' - (h^3/3!)y_i''' + (h^4/4!)y^{(4)}(x_i - \theta_i h)$$
$$y_{i+1} = y_i + hy_i' + (h^2/2)y_i'' + (h^3/3!)y_i''' + (h^4/4!)y^{(4)}(x_i + \theta_i h)$$

Pertanto, sommando membro a membro, si ricava

$$\begin{aligned} y_i'' &= \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + \frac{h^2}{4!}\Big(y^{(4)}(x_i - \theta_i h) + y^{(4)}(x_i + \theta_i h)\Big) \\ &= \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + \frac{h^2}{12}y^{(4)}(x_i + \psi h) \end{aligned}$$

Modello discreto

Da ciò si ricava la **formula alle differenze centrali** per l'approssimazione della derivata seconda in un punto con un **errore di troncamento** dell'ordine di h^2.
Allora per ogni $i = 1, \ldots, n$, essendo noti $y(a) = \alpha$, $y(b) = \beta$, si può sostituire all'equazione differenziale in x_i l'equazione alle differenze

$$\frac{-y_{i-1} + 2y_i - y_{i+1}}{h^2} + c(x_i)y_i = f(x_i) + \tau_i(y)$$

con $\tau_i(y) = -(h^2/12)y^{(4)}(x_i + \psi h)$, ottenendo un sistema di n equazioni in n incognite. In forma matriciale il sistema si scrive:

$$A\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\tau}(y)$$

con

$$\frac{1}{h^2}\begin{pmatrix} 2+h^2c(x_1) & -1 & & & \\ -1 & 2+h^2c(x_2) & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2+h^2c(x_{n-1}) & -1 \\ & & & -1 & 2+h^2c(x_n) \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} f(x_1) + \alpha/h^2 \\ f(x_2) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) + \beta/h^2 \end{pmatrix} + \boldsymbol{\tau}(y)$$

Metodo alle differenze finite

Il metodo alle differenze consiste nel trascurare il termine di errore e nel prendere come approssimazione per $y(x)$ la soluzione $\boldsymbol{u}$ del sistema di equazioni

$$A\boldsymbol{u} = \boldsymbol{b}$$

Poichè $c(x) \geqslant 0$, si può dimostrare che la matrice A è simmetrica definita positiva. Pertanto essa è non singolare. Dunque la soluzione esiste ed è unica.

I sistemi lineari algebrici

- Molti problemi si possono rappresentare mediante un sistema lineare
- La soluzione di un sistema lineare costituisce un sottoproblema di moltissime applicazioni del calcolo scientifico

Obiettivo: studiare algoritmi efficienti per la risoluzione numerica dei sistemi.

Sistemi di equazioni lineari

Un **sistema lineare** di n equazioni algebriche in n incognite è esprimibile come:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n \end{cases}$$

dove $a_{ij} \in \mathbb{R}$ si dicono coefficienti del sistema, $b_i \in \mathbb{R}$ sono i termini noti e x_i sono le incognite; in notazione matriciale, chiamando A la matrice reale quadrata degli $n \times n$ coefficienti, $\boldsymbol{x} \in \mathbb{R}^n$ il vettore delle incognite e $\boldsymbol{b} \in \mathbb{R}^n$ il vettore dei termini noti (termine noto), si ha

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

IPOTESI: Si assume che A sia non singolare, ossia che $\det(A) \neq 0$ o, equivalentemente, che esiste l'inversa della matrice A. In tal caso la soluzione del sistema è unica ed è data da

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

Dobbiamo calcolare l'inversa?

Il calcolo dell'inversa di A consiste nel calcolare una matrice X tale che

$$AX = I$$

dove I è la matrice identità di ordine n. Ciò equivale a risolvere n sistemi lineari:

$$AX_{*j} = \boldsymbol{e}_j$$

dove X_{*j} è la j-esima colonna della matrice inversa incognita ed $\boldsymbol{e}_j$ è la j-esima colonna dell'identità (j-esimo vettore della base canonica di $\mathbb{R}^n$).
Ovviamente **non conviene** calcolare la soluzione del sistema $A\boldsymbol{x} = \boldsymbol{b}$ come $\boldsymbol{x} = A^{-1}\boldsymbol{b}$, prima di tutto perché il calcolo dell'inversa comporta la risoluzione di n sistemi.

Metodi per la risoluzione di sistemi lineari

- Metodi diretti: con un numero finito di operazioni, in aritmetica esatta, si determina la soluzione esatta; poiché si lavora in aritmetica finita, occorre valutare l'errore di arrotondamento delle operazioni e l'errore inerente.
- Metodi iterativi: la soluzione si ottiene come limite di una successione di approssimazioni alla soluzione; a partire da un vettore che è una approssimazione iniziale a una soluzione, si costruisce una successione di vettori convergenti alla soluzione cercata quando il numero di passi tende all'infinito; poiché il processo deve essere interrotto, occorre analizzare l'errore di troncamento nell'approssimazione determinata (errore inerente, errore di troncamento).

La scelta del metodo dipende:

- dalla struttura della matrice (densa o sparsa, ossia con un numero di elementi non nulli proporzionali alla dimensione della matrice);
- dalla condizione della matrice;
- dalla dimensione.

Per ciascun metodo, occorre analizzare la complessità computazionale, l'errore, la dipendenza dalla struttura della matrice.
La matrice $[A|\boldsymbol{b}]$ di dimensioni $n \times (n+1)$, ottenuta aggiungendo alla matrice A come $(n+1)$-esima colonna il vettore dei termini noti, si dice **matrice completa**.

Casi particolari

- Sia $A = D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ **diagonale**.

Poiché $\det(D) = d_1 d_2 \cdots d_n$, D è non singolare se e solo se $d_i \neq 0$, $i = 1, \ldots, n$.
In tal caso l'inversa è una matrice diagonale data da
$D^{-1} = \mathrm{diag}(1/d_1, 1/d_2, \ldots, 1/d_n)$. Segue che la soluzione di

$$D\boldsymbol{x} = \boldsymbol{b}$$

è ottenuta immediatamente mediante la formula:

$$x_i^* = b_i/d_i, \quad i = 1, \ldots, n$$

```
function [x] = soldiag(A, b);
% SOLDIAG - soluzione di un sistema diagonale
if ( isempty(find(~diag(A))) )
   x = b ./ diag(A);
else
   error('la matrice e'' singolare!');
end
```

Verifica della correttezza con Matlab

Un modo per verificare la correttezza del codice è il seguente (dal prompt dei comandi di Matlab):

```
>> D = diag(rand(6,1));  % matrice diagonale avente come elementi
                         % numeri casuali in (0,1);
>> spy(D)                % spy e' una funzione grafica che
                         % visualizza la struttura  di una matrice
>> ij = find( diag(D) == 0 ) % equivalentemente  ij = find(~diag(D))
% si controlla quali elementi diagonali di D sono nulli
ij =     []
%  Poiche' ij  e' l'insieme vuoto, D non ha elementi diagonali nulli
>> b = D*ones(6,1);    % Si costruisce il termine noto in modo
                       % che la soluzione sia un vettore di 1
>> x = soldiag(D,b)    % si risolve
x =
     1
     1
     1
     1
     1
     1
```


Matrici triangolari

- Sia A una matrice **triangolare** (inferiore se $a_{ij} = 0 \ \forall j > i$, oppure superiore se $a_{ij} = 0 \ \forall j < i$).

Per fissare le idee, sia $A = R = \{r_{ij}\}$, triangolare superiore.

poiché $\det(R) = r_{11} r_{22} \cdots r_{nn}$, R è non singolare se e solo se tutti gli elementi diagonali sono non nulli.

In tal caso si può mostrare che l'inversa di una matrice triangolare superiore (inferiore) è una matrice triangolare superiore (inferiore), con elementi diagonali dati da $1/r_{ii}$, $i = 1, \ldots, n$.

Calcolo dell'inversa di una matrice triangolare superiore

poiché $RR^{-1} = I_n$, denotando con ρ_{ij} gli elementi di R^{-1}, si ha che:

$$r_{i1}\rho_{1j} + r_{i2}\rho_{2j} + \ldots + r_{in}\rho_{nj} = \sum_{k=1}^{n} r_{ik}\rho_{kj} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

Se $i = j$, poiché

$$r_{i1}\rho_{1i} + r_{i2}\rho_{2i} + \ldots + r_{i,i-1}\rho_{i-1,i} + r_{ii}\rho_{ii} + r_{i,i+1}\rho_{i+1,i} + \ldots + r_{in}\rho_{ni} = 1$$

e $r_{ik} = 0$, $k < i$, $\rho_{ki} = 0$, $i < k$, segue

$$r_{ii}\rho_{ii} = 1 \Leftrightarrow \rho_{ii} = \frac{1}{r_{ii}}$$

Se invece $i < j$, allora

$$\begin{aligned} \sum_{k=1}^{n} r_{ik}\rho_{kj} &= r_{i1}\rho_{1j} + r_{i2}\rho_{2j} + \ldots + r_{i,i-1}\rho_{i-1,j} + r_{ii}\rho_{ij} + \ldots \\ &\quad \ldots + r_{ij}\rho_{jj} + r_{i,j+1}\rho_{j+1,j} + \ldots + r_{in}\rho_{nj} \\ &= \sum_{k=i}^{j} r_{ik}\rho_{kj} = r_{ii}\rho_{ij} + r_{i,i+1}\rho_{i+1,j} + \ldots + r_{ij}\rho_{jj} = 0 \end{aligned}$$

Calcolo dell'inversa di una matrice triangolare superiore

Da questa uguaglianza si ricava ρ_{ij}:

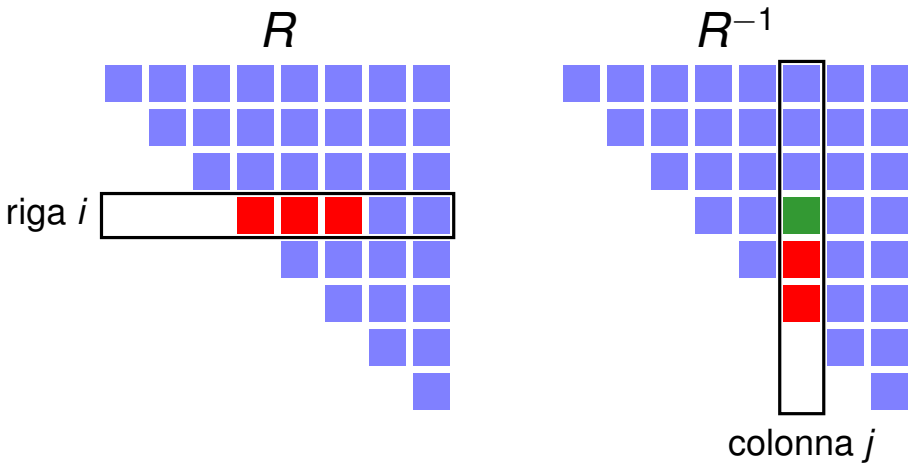
$$\rho_{ij} = -\frac{r_{i,i+1}\rho_{i+1,j} + \ldots + r_{ij}\rho_{jj}}{r_{ii}} = -\frac{\sum_{k=i+1}^{j} r_{ik}\rho_{kj}}{r_{ii}}$$

Dunque per calcolare ρ_{ij}, occorre disporre della i-esima riga di R dalla diagonale fino all'elemento j e della j-esima colonna di R^{-1} dall'$(i+1)$-esimo elemento al j-esimo.

Calcolo dell'inversa di una matrice triangolare superiore

Esempio: calcolo dell'elemento di posizione $(i, j) = (4, 6)$ di R^{-1}.

Voglio calcolare l'elemento verde: ho bisogno degli elementi rossi.



Gli elementi di R^{-1} si possono calcolare, sfruttando le uguaglianze in ordine opportuno (PAVIMENTAZIONE DELLA MATRICE).
Si può calcolare R^{-1} a partire dalla n-esima riga fino alla prima e su ogni riga dall'elemento di posizione n all'elemento diagonale. Usando questo ordinamento si può usare R come spazio di memorizzazione per R^{-1}.

Codice Matlab per il calcolo dell'inversa di una triangolare superiore

```
function [R] = invupper(R)
% INVUPPER - Sovrascrive una matrice triang. sup. invert. con la propria
    inversa
% SYNOPSIS:
%   [R] = invupper(R)
% INPUT:
%   R (double array) - Matrice triangolare superiore da invertire
% OUTPUT:
%   R (double array) - La matrice in input sovrascritta con la propria inversa

[m, n] = size(R);
if ( isempty( find( diag(R) == 0 ) ) )  % oppure find( ~diag(R) )
   R(n,n) = 1 / R(n,n);
   for i = n-1 : -1 : 1
      R(i,i) = 1 / R(i,i);
      for j = n : -1 : i+1
         %  s = 0;
         %  for k = i+1 : j
         %     s = s + R(i,k)*R(k,j);
         %  end
         %  R(i,j) = -s*R(i,i);
         R(i,j) = -( R(i, i+1:j)*R(i+1:j, j) )*R(i,i);
      end
   end
else
   error('la matrice e'' singolare');
end
end
```

Verifica di correttezza

Verifica della correttezza (al prompt dei comandi di Matlab):

```
» R = triu(hilb(6));
% R e' uguale alla parte triangolare sup. di una matrice
% di Hilbert di ordine 6
» ij = find( diag(R)==0 )   % controllo se la matrice e' invertibile
ij =
     0x1 empty double column vector

» T = R * invupper(R)
T =

    1.0000         0         0    0.0000   -0.0000   -0.0000
         0    1.0000         0         0         0   -0.0000
         0         0    1.0000         0         0         0
         0         0         0    1.0000         0         0
         0         0         0         0    1.0000         0
         0         0         0         0         0    1.0000
```

Cosa nascondono quegli zeri dopo la virgola? Impostando la massima accuratezza di visualizzazione con "`format long e`" si scopre ad esempio che

```
>> T(1, 4:6)

ans =
     2.220446049250313e-16    -1.110223024625157e-16    -8.326672684688674e-17
```


Verifica di correttezza

```
>> R
R =
    1.0000    0.5000    0.3333    0.2500    0.2000    0.1667
         0    0.3333    0.2500    0.2000    0.1667    0.1429
         0         0    0.2000    0.1667    0.1429    0.1250
         0         0         0    0.1429    0.1250    0.1111
         0         0         0         0    0.1111    0.1000
         0         0         0         0         0    0.0909


>> invupper(R)
ans =
    1.0000   -1.5000    0.2083    0.1069    0.0618    0.0386
         0    3.0000   -3.7500    0.1750    0.1246    0.0911
         0         0    5.0000   -5.8333    0.1339    0.1073
         0         0         0    7.0000   -7.8750    0.1069
         0         0         0         0    9.0000   -9.9000
         0         0         0         0         0   11.0000
```

Richiami per calcolare la complessità computazionale

Per contare le operazioni si ricordi che:

- la somma dei primi s interi vale

$$1 + 2 + 3 + \ldots + s = \frac{s(s+1)}{2}$$

- la somma dei quadrati dei primi s interi vale

$$1 + 2^2 + 3^2 + \ldots + s^2 = \frac{s(s+1)(2s+1)}{6}$$

Richiami per calcolare la complessità computazionale

Per il calcolo dell'inversa, la complessità computazionale è:

1. n divisioni;
2. $1 + (1+2) + (1+2+3) + \ldots + \big(1 + 2 + \ldots + (n-1)\big) =$
   $= \sum_{i=1}^{n-1}(1 + 2 + \ldots + i) = \sum_{i=1}^{n-1} \frac{i(i+1)}{2} = \sum_{i=1}^{n-1} \frac{1}{2}(i^2 + i) =$
   $\frac{1}{2}\left(\frac{n(n-1)(2n-1)}{6} + \frac{n(n-1)}{2}\right) = \mathcal{O}(n^3/6)$ moltiplicazioni e circa altrettante somme.

- per $\rho_{n-1,n}$: un prodotto + un prodotto per $\frac{1}{r_{n-1,n-1}}$;
- per $\rho_{n-2,n}$, $\rho_{n-2,n-1}$: $2+1$ prodotti e una somma + due prodotti per $\frac{1}{r_{n-2,n-2}}$;
- per $\rho_{n-3,n}$, $\rho_{n-3,n-1}$, $\rho_{n-3,n-2}$: $3+2+1$ prodotti e $1+2$ somme + tre prodotti per $\frac{1}{r_{n-3,n-3}}$;
- ...
- per $\rho_{1,n}$, $\rho_{1,n-1}$, ..., $\rho_{1,2}$: $(n-1) + (n-2) + \ldots + 2 + 1$ prodotti e $(n-2) + \ldots + 2 + 1$ somme + $(n-1)$ prodotti per $\frac{1}{r_{1,1}}$

Algoritmi di sostituzione all'indietro e di eliminazione in avanti

La risoluzione di un sistema triangolare superiore si può ottenere in sole $\mathcal{O}(n^2/2)$ somme e prodotti e n divisioni mediante l'**algoritmo di sostituzione all'indietro**.

In tal caso si ricava x_n dall'ultima equazione. Si sostituisce nella penultima e si ricava x_{n-1} e così via (algoritmo per righe).

$$\begin{cases} r_{11}x_1 + r_{12}x_2 + \quad \ldots \quad + r_{1n}x_n = b_1 & \Rightarrow \quad x_1^* = \dfrac{b_1 - \sum_{j=2}^{n} r_{1j}x_j^*}{r_{11}} \\ r_{22}x_2 + \quad \ldots \quad + r_{2n}x_n = b_2 & \Rightarrow \quad x_2^* = \dfrac{b_2 - \sum_{j=3}^{n} r_{2j}x_j^*}{r_{22}} \\ \vdots & \vdots \\ r_{n-1,n-1}x_{n-1} + r_{n-1,n}x_n = b_{n-1} & \Rightarrow \quad x_{n-1}^* = \dfrac{b_{n-1} - r_{n-1,n}x_n^*}{r_{n-1,n-1}} \\ r_{nn}x_n = b_n & \Rightarrow \quad x_n^* = \dfrac{b_n}{r_{nn}} \end{cases}$$

Si può anche, dopo aver ricavato x_n dall'ultima equazione, sostituire in tutte le precedenti ottenendo un sistema triangolare superiore di dimensione $n-1$; da questo si ricava l'ultima incognita x_{n-1} dall'ultima equazione e si sostituisce nelle precedenti e così via (algoritmo di sostituzione all'indietro per colonne).

Esempio

$$\begin{cases} 3x_1 + 5x_2 - x_3 = 7 \\ 4x_2 + x_3 = 5 \\ -x_3 = -1 \end{cases}$$

Soluzione per righe:

- dalla terza equazione $x_3^* = 1$;
- sostituendo nella seconda si ha $x_2^* = 1$;
- sostituendo x_2 e x_3 nella prima, $x_1^* = 1$

Soluzione per colonne:

- dall'ultima equazione $x_3^* = 1$; si sostituisce nelle precedenti equazioni ottenendo
$$\begin{cases} 3x_1 + 5x_2 = 8 \\ 4x_2 = 4 \end{cases}$$
- dall'ultima equazione $x_2^* = 1$; si sostituisce nella precedente equazione ottenendo
$$\{3x_1 = 3$$
- dall'ultima equazione $x_1^* = 1$.

Codici Matlab per la soluzione di sistemi triangolari superiori

```
function [x] = solupper(R,b)
% SOLUPPER - Soluzione di sistema triang. sup. (per righe)
  n = length(b);
  x = b;                                  % si puo' evitare con:
  x(n) = x(n)/R(n,n);                     % x(n) = b(n)/R(n,n);
  for i = n-1 : -1 : 1                    % e poi mettendo
     % SDOT - BLAS1                       % dentro il ciclo:
     x(i) = x(i) - R(i, i+1:n)*x(i+1:n);  % x(i) = b(i) - R(i,...
     x(i) = x(i)/R(i,i);
  end
end


function [x] = rtrisol(R,b)
% RTRISOL - Soluzione di sistema triang. sup. (per colonne)
  n = length(b);
  x = b;     % Qui NON si puo' evitare di fare questa assegnazione!!
  x(n) = x(n)/R(n,n);
  for j = n-1 : -1 : 1
     % SAXPY - BLAS1
     x(1:j) = x(1:j) - R(1:j, j+1)*x(j+1);
     x(j) = x(j)/R(j,j);
  end
end
```


Sistemi triangolari inferiori

Per i sistemi triangolari inferiori si usa l'**algoritmo di eliminazione in avanti**, ove si ricava x_1 dalla prima equazione, si sostituisce nella seconda e si ricava x_2 e così via (algoritmo per righe).

$$\begin{cases} \ell_{11}x_1 & = b_1 & \Rightarrow & x_1^* = \dfrac{b_1}{\ell_{11}} \\ \ell_{21}x_1 + \ell_{22}x_2 & = b_2 & \Rightarrow & x_2^* = \dfrac{b_2 - \ell_{21}x_1^*}{\ell_{22}} \\ & \vdots & & \vdots \\ \ell_{n1}x_1 + \ell_{n2}x_2 + \ldots + \ell_{nn}x_n & = b_n & \Rightarrow & x_n^* = \dfrac{b_n - \sum_{j=1}^{n-1} \ell_{nj}x_j^*}{\ell_{nn}} \end{cases}$$

Si può anche, dopo aver ricavato x_1 dalla prima equazione, sostituire in tutte le successive ottenendo un sistema triangolare inferiore di dimensione $n-1$; da questo si ricava la prima incognita x_2 dalla prima equazione e si sostituisce nelle successive equazioni e così via (algoritmo di eliminazione in avanti per colonne).

Codici Matlab per la soluzione di sistemi triangolari inferiori

```
function [x] = sollower(L, b)
% SOLLOWER - Soluzione di sistemi triang. inf. (per righe)
  n = length(b);
  x = b;
  x(1) = x(1)/L(1,1);
  for i = 2:n
     % SDOT
     x(i) = x(i) - L(i, 1:i-1)*x(1:i-1);
     x(i) = x(i)/L(i,i);
  end
end


function [x] = ltrisol(L, b)
% LTRISOL - Soluzione di sistemi triang. inf. (per colonne)
  n = length(b);
  x = b;
  x(1) = x(1)/L(1,1);
  for j = 2:n
     % SAXPY
     x(j:n) = x(j:n) - L(j:n, j-1)*x(j-1);
     x(j)   = x(j) / L(j,j);
  end
end
```


Complessità - Analisi dell'errore algoritmico

Per entrambe gli algoritmi in tutte e due le versioni (per righe e per colonne) serve lo stesso numero di operazioni, date da $\frac{n(n-1)}{2}$ prodotti e altrettante somme e n divisioni.
Si dice che sono algoritmi per cui servono $\mathcal{O}(n^2/2)$ flops.

Entrambi gli algoritmi sono metodi diretti, che, in un numero finito di passi, in aritmetica esatta, calcolano la soluzione esatta del sistema. Quando si usa aritmetica finita, invece di calcolare la soluzione esatta $\boldsymbol{x}$ del sistema

$$R\boldsymbol{x} = \boldsymbol{b} \quad \text{oppure} \quad L\boldsymbol{x} = \boldsymbol{b}$$

si determina una soluzione approssimata $\boldsymbol{z}$. Essa può essere ***interpretata*** come soluzione esatta di un sistema perturbato:

$$(R + \delta R)\boldsymbol{z} = \boldsymbol{b} \quad \text{oppure} \quad (L + \delta L)\boldsymbol{z} = \boldsymbol{b}$$

È possibile trovare una maggiorazione per la norma delle matrici di perturbazione:

$$\|\delta R\|_\infty \leqslant 1.01u\frac{n(n+1)}{2}\max|r_{ij}|$$

$$\|\delta L\|_\infty \leqslant 1.01u\frac{n(n+1)}{2}\max|\ell_{ij}|$$

assumendo $(n-1)u \leqslant 0.01$, dove u è la precisione di macchina.

Metodi diretti per il caso generale

$$A\boldsymbol{x} = \boldsymbol{b} \qquad \det(A) \neq 0$$

- **Metodo di Cramer**: si calcola $x_j^* = \dfrac{\det(A_j)}{\det(A)}$, dove $\det(A_j)$ è la matrice ottenuta dalla A sostituendo alla j-esima colonna di A il termine noto $\boldsymbol{b}$. Occorre calcolare $n+1$ determinanti. Se si usa la regola di Laplace per il calcolo del determinante, servono $n!(n-1)$ prodotti per ogni determinante. Pertanto per il calcolo della soluzione servono $n!(n-1)(n+1)$ prodotti, n divisioni e $(n!-1)(n+1)$ somme. Contando solo i prodotti e supponendo che si impieghino 10^{-12} secondi per ogni prodotto, per $n = 20$ servono circa 154 anni per calcolare la soluzione del sistema!! C'è eccessiva complessità computazionale e conseguente accumulo di errori.
- **Calcolo dell'inversa**: comporta la soluzione di n sistemi.
  Si consideri il caso semplice:
  $$7x = 21$$
  Se viene calcolato come $x = (7)^{-1} \cdot 21 = 0.142857 \cdot 21 = 2.99997$, ci vogliono 2 operazioni e c'è un errore.

Metodi diretti per il caso generale

Se invece si usano le proprietà delle equazioni (metodo di sostituzione), si ha $x = 21/7 = 3$; si esegue una sola operazione e non c'è errore.
Dunque occorre evitare il calcolo dell'inversa.

- **Metodi di fattorizzazione**.
  L'idea di base è fattorizzare la matrice A nel prodotto di due matrici "semplici", in modo che sia facile risolvere i due sistemi associati; in particolare, si studiano due tipi di fattorizzazione:
  - **fattorizzazione** LR: si fattorizza la matrice A nel prodotto di una matrice triangolare inferiore per una triangolare superiore; se $A = LR$, si ha
    $$A\boldsymbol{x} = \boldsymbol{b} \quad \Rightarrow \quad L\underbrace{R\boldsymbol{x}}_{\boldsymbol{y}} = \boldsymbol{b} \quad \Rightarrow \quad \begin{cases} L\boldsymbol{y} = \boldsymbol{b} & \textbf{eliminazione in avanti} \\ R\boldsymbol{x} = \boldsymbol{y} & \textbf{sostituzione all'indietro} \end{cases}$$
    In totale, una volta ottenuta la fattorizzazione, servono $\mathcal{O}(n^2/2) + \mathcal{O}(n^2/2) = \mathcal{O}(n^2)$ operazioni.
  - **fattorizzazione** QR: si fattorizza la matrice A nel prodotto di una matrice ortogonale (per cui $Q^{-1} = Q^T$) con una matrice triangolare superiore; se $A = QR$, si ha
    $$A\boldsymbol{x} = \boldsymbol{b} \quad \Rightarrow \quad QR\boldsymbol{x} = \boldsymbol{b} \quad \Rightarrow \quad \underbrace{Q^TQ}_{I}R\boldsymbol{x} = Q^T\boldsymbol{b} \quad \Rightarrow \quad R\boldsymbol{x} = Q^T\boldsymbol{b}$$
    In totale $\mathcal{O}(n^2/2 + n^2)$ operazioni.

Fattorizzazione *LR*

Data $A \in \mathbb{R}^{n\times n}$, si vogliono trovare le condizioni per cui A si può fattorizzare nel prodotto di una matrice triangolare inferiore L con una triangolare superiore R.
poiché deve essere

$$A = LR$$

occorre imporre n^2 uguaglianze $\left(a_{ij} = \sum_{k=1}^{n} \ell_{ik} r_{kj}\right)$ per determinare $n^2 + n$ parametri (numero di elementi non nulli di L e di R).
Pertanto occorre fissare n elementi, attribuendo ad essi un valore arbitrario.
In genere, per convenzione, si fissano uguali a 1 gli elementi diagonali della matrice triangolare inferiore o superiore. In realtà si cerca la fattorizzazione

$$A = LDU$$

dove L è una matrice triangolare inferiore con elementi diagonali 1, D è una matrice diagonale, U è una matrice triangolare superiore con elementi diagonali 1.

$$A = LDU = \begin{array}{cll} \nearrow & LR \quad R = DU & \text{fattorizzazione di Doolittle} \\ \searrow & \tilde{L}U \quad \tilde{L} = LD & \text{fattorizzazione di Crout} \end{array}$$

dove $r_{ij} = d_i\, u_{ij}$ e $\tilde{\ell}_{ij} = \ell_{ij}\, d_j$.
Da una fattorizzazione si può ottenere l'altra.

Fattorizzazione *LDU*

La fattorizzazione $A = LDU$ non esiste sempre. Per esempio, se $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, non esistono L, D e U:

$$\begin{pmatrix} 1 & 0 \\ \ell_{21} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} 1 & u_{12} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} d_1 & d_1 u_{12} \\ \ell_{21} d_1 & \ell_{21} d_1 u_{12} + d_2 \end{pmatrix}$$

$$\begin{aligned} d_1 &= 0 \\ d_1 u_{12} &= 1 \\ \ell_{21} d_1 &= 1 \\ \ell_{21} d_1 u_{12} + d_2 &= 0 \end{aligned}$$

Equazioni incompatibili.

Sottomatrici e minori principali primi

Definizione.

Si dice *sottomatrice principale prima* (o *di testa*) di ordine i di una matrice A l'intersezione delle prime i righe e i colonne. Tale matrice di ordine i si denota con $A^{(i)}$.
Il suo determinante si dice *minore principale di testa* di ordine i di A.

Esempio.

$$A = \begin{pmatrix} 1 & 2 & -1 & 4 \\ 3 & 0 & -3 & 1 \\ 5 & 7 & 9 & 1 \\ 0 & 3 & -2 & 4 \end{pmatrix}$$

Sottomatrici principali prime

$$A^{(1)} = 1 \qquad A^{(2)} = \begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix} \qquad A^{(3)} = \begin{pmatrix} 1 & 2 & -1 \\ 3 & 0 & -3 \\ 5 & 7 & 9 \end{pmatrix} \qquad A^{(4)} = A$$

Minori principali primi

$$\det(A^{(1)}) = 1 \quad \det(A^{(2)}) = -6 \quad \det(A^{(3)}) = -84 \quad \det(A^{(4)}) = 288$$

Condizioni per l'esistenza della fattorizzazione LR

Teorema

Se tutti i minori principali di testa di A di ordine $i = 1, 2, \ldots, n-1$ sono non nulli, allora esistono matrici L triangolare inferiore unitaria (con 1 sulla diagonale), D diagonale, U triangolare superiore unitaria, tali che $A = LDU$. La fattorizzazione è univocamente determinata e vale che

$$\det(A^{(i)}) = d_1 d_2 \cdots d_i \neq 0 \qquad i = 1, \ldots, n-1$$

Inoltre, poiché $\det(A) = d_1 \cdots d_n$, se $d_n \neq 0$, A è non singolare.

Viceversa, se A è decomponibile in modo unico come $A = LDU$, con L triangolare inferiore unitaria, U triangolare superiore unitaria e D diagonale con $d_i \neq 0$, $i = 1, \ldots, n-1$, allora tutte le sottomatrici principali prime eccetto al più l'ultima sono non singolari.

$$A = LDU \Rightarrow A^{(i)} = (L_i \; 0) \begin{pmatrix} D_i & 0 \\ 0 & D_{n-i} \end{pmatrix} \begin{pmatrix} U_i \\ 0 \end{pmatrix} = L_i D_i U_i$$

Esempio

$$\begin{pmatrix} 5 & 2 & 1 \\ -1 & 4 & 1 \\ -2 & 8 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1/5 & 1 & 0 \\ -2/5 & 2 & 1 \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 22/5 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2/5 & 1/5 \\ 0 & 1 & 3/11 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A = L D U$$

$$\det(A^{(1)}) = 5 \qquad \det(A^{(2)}) = 22 \qquad \det(A^{(3)}) = 0$$

Le sottomatrici principali prime di ordine 1 e 2 sono non singolari. Dunque la fattorizzazione esiste. Inoltre

$$\det(A^{(1)}) = d_1 \qquad \det(A^{(2)}) = d_1 d_2 \qquad \det(A^{(3)}) = d_1 d_2 d_3$$

Trasformazioni elementari di Gauss

Sia $\boldsymbol{x} \in \mathbb{R}^n$ con $x_1 \neq 0$. Una trasformazione elementare di Gauss è una matrice triangolare inferiore con 1 sulla diagonale tale che $L_1 \boldsymbol{x} = (x_1, 0, ..., 0)^T$:

$$L_1 = I - \boldsymbol{m}^{(1)} \boldsymbol{e}_1^T = \begin{pmatrix} 1 & 0 & \dots & \\ 0 & 1 & 0 & \dots \\ & & & \\ \dots & & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 \\ m_2 \\ \vdots \\ m_n \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$$

$$L_1 \boldsymbol{x} = \begin{pmatrix} 1 & & & \\ -m_2 & 1 & & \\ \vdots & \vdots & 1 & \\ -m_n & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$-m_i x_1 + x_i = 0 \Rightarrow m_i = \frac{x_i}{x_1} \quad i = 2, \dots, n$$

dove $\boldsymbol{m}^{(1)} = (0, m_2, \dots, m_n)^T$.

Esempio

Sia $\boldsymbol{x} = (3,\ 1,\ -5\ 7)^T$. La trasformazione elementare di Gauss associata al vettore è data da:

$$L_1 = \begin{pmatrix} 1 & & & \\ -1/3 & 1 & & \\ 5/3 & 0 & 1 & \\ -7/3 & 0 & 0 & 1 \end{pmatrix} = I_4 - \begin{pmatrix} 0 \\ 1/3 \\ -5/3 \\ 7/3 \end{pmatrix} (1\ 0\ 0\ 0)$$

$$L_1 \boldsymbol{x} = \begin{pmatrix} 1 & & & \\ -1/3 & 1 & & \\ 5/3 & 0 & 1 & \\ -7/3 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ -5 \\ 7 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Trasformazione elementare di Gauss

In generale, se $x_i \neq 0$ e si vuole trasformare $\boldsymbol{x}$ in un vettore $(x_1, \ldots, x_i, 0, \ldots, 0)^T$, allora $L_i = I - \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T$, con $\boldsymbol{m}^{(i)} = (0, \ldots, 0, m_{i+1}, \ldots, m_n)^T$, $m_j = \dfrac{x_j}{x_i}$, $j = i+1, \ldots, n$.

Esempio.
Sia $\boldsymbol{x} = (0,\ 0,\ -5,\ 7)^T$. La trasformazione elementare di Gauss associata al vettore e che lo trasforma in $(0,\ 0,\ -5,\ 0)^T$, è data da:

$$L_3 = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 0 & 7/5 & 1 \end{pmatrix} = I_4 - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 7/(-5) \end{pmatrix} (0\ 0\ 1\ 0)$$

L_i è non singolare e $L_i^{-1} = I + \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T$. Infatti:

$$(I - \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T)(I + \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T) = I - \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T + \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T - \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T \boldsymbol{m}^{(i)} \boldsymbol{e}_i^T = I$$

perché $\boldsymbol{e}_i^T \boldsymbol{m}^{(i)} = (0\ \ldots\ \overbrace{1}^{i\text{-esima pos.}}\ \ldots\ 0) \begin{pmatrix} 0 \\ \vdots \\ m_{i+1} \\ \vdots \\ m_n \end{pmatrix} = 0.$

Come si costruisce la fattorizzazione $A = LR$?

Usiamo le trasformazioni elementari di Gauss per costruire la fattorizzazione di Doolittle $A = LR$, nell'ipotesi di A con tutti i determinanti delle sottomatrici principali prime (minori) di ordine $i = 1, 2, \ldots, n-1$ non nulli.
Poiché la costruzione è fatta per risolvere $A\boldsymbol{x} = \boldsymbol{b}$, applichiamo le stesse trasformazioni anche a $\boldsymbol{b}$, ossia fattorizziamo la matrice completa $[A|\boldsymbol{b}]$.

Algoritmo di eliminazione di Gauss

Per spiegare il procedimento partiamo con un esempio e poi formalizzeremo.

$$\begin{cases} x_1 + x_2 + 3x_4 = 4 \\ 2x_1 + x_2 - x_3 + x_4 = 1 \\ 3x_1 - x_2 - x_3 + 2x_4 = -3 \\ -x_1 + 2x_2 + 3x_3 - x_4 = 4 \end{cases}$$

La matrice completa è

$$[A|\boldsymbol{b}] = [A_1|\boldsymbol{b}_1] = \left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{array}\right)$$

Eliminiamo la prima incognita da tutte le equazioni eccetto la prima. Ciò vuol dire togliere da ogni equazione a partire dalla seconda un opportuno multiplo della prima equazione, azzerando tutti i coefficienti di x_1 nelle righe successive della matrice dei coefficienti.
In questo caso $A_{2*} - 2A_{1*}$, $A_{3*} - 3A_{1*}$ e infine $A_{4*} - (-1)A_{1*}$, ottenendo:

$$\left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & -4 & -1 & -7 & -15 \\ 0 & 3 & 3 & 2 & 8 \end{array}\right) = [A_2|\boldsymbol{b}_2]$$

Si noti che ciò equivale a premoltiplicare $[A|\boldsymbol{b}_1]$ per la matrice di trasformazione elementare di Gauss che trasforma la prima colonna di A in $(a_{11}, 0, \ldots, 0)^T$. In questo caso la matrice di Gauss è data da $L_1 = I_4 - \boldsymbol{m}^{(1)}\boldsymbol{e}_1^T$ con $\boldsymbol{m}^{(1)} = (0, 2, 3, -1)^T$.

$$L_1 = \begin{pmatrix} 1 & & & \\ -2 & 1 & & \\ -3 & 0 & 1 & \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Algoritmo di eliminazione di Gauss

Adesso ripetiamo cercando di eliminare x_2 dal tutte le equazioni successive alla seconda. Ciò significa togliere dalla terza e quarta equazione un opportuno multiplo della seconda, azzerando i coefficienti di x_2 nella terza e quarta equazione. In analogia a quanto visto prima, è possibile ottenere questo con una trasformazione elementare di Gauss associata alla seconda colonna della matrice, che vada ad annullare tutti gli elementi della seconda colonna sotto quello diagonale. In questo caso $L_2 = I_4 - \boldsymbol{m}^{(2)} \boldsymbol{e}_2^T$ con $\boldsymbol{m}^{(2)} = (0, 0, 4, -3)^T$.

$$L_2[A_2|\boldsymbol{b}_2] = \left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & -13 & -13 \end{array}\right) = [A_3|\boldsymbol{b}_3]$$

Infine se il coefficiente di x_3 nell'ultima equazione non fosse già nullo, con una ultima trasformazione di Gauss si riuscirebbe ad annullare il coefficiente di x_3 nell'ultima equazione. In tal caso $L_3 = I_4 - \boldsymbol{m}^{(3)} \boldsymbol{e}_3^T = I$ con $\boldsymbol{m}^{(3)} = (0, 0, 0, 0)^T$. La matrice di partenza è stata ridotta a forma triangolare. Adesso si può risolvere il sistema (che è equivalente al dato) con l'algoritmo di sostituzione all'indietro ottenendo la soluzione.

$$\begin{pmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ -7 \\ 13 \\ -13 \end{pmatrix}$$

$$x_4^* = 1, \quad x_3^* = 0, \quad x_2^* = 2, \quad x_1^* = -1$$

Metodo di eliminazione di Gauss

Formalizziamo quanto detto nel caso generale.

- Passo 1. $A_1 = A, \quad \boldsymbol{b}_1 = \boldsymbol{b}$.
  Eliminiamo la prima incognita da tutte le equazioni eccetto la prima. Ciò vuol dire togliere da ogni equazione a partire dalla seconda un opportuno multiplo della prima equazione, azzerando tutti i coefficienti di x_1 nelle righe 2,3,...,n della matrice dei coefficienti. Ciò significa premoltiplicare $[A_1|\boldsymbol{b}_1]$ per la matrice di trasformazione elementare di Gauss che trasforma la prima colonna di A in $(a_{11}, 0, ..., 0)^T$. è possibile farlo perché $a_{11} \neq 0$ per ipotesi.

$$L_1[A_1|\boldsymbol{b}_1] = [A_2 \ \boldsymbol{b}_2]$$

$$\begin{pmatrix} 1 & & & \\ -m_{21} & 1 & & \\ \vdots & & \ddots & \\ -m_{n1} & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11} & \dots & \dots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}$$

Metodo di eliminazione di Gauss

dove $L_1 = I - \boldsymbol{m}^{(1)} \boldsymbol{e}_1^T$ e $\boldsymbol{m}^{(1)} = (0, m_{21}, \ldots, m_{n,1})^T$.
Poiché $a_{i1}^{(2)} = 0 = -m_{i1} a_{11} + a_{i1}$ e $a_{11} \neq 0$ (minore di A di ordine 1), segue

$$
\begin{aligned}
m_{i1}^{(1)} &= \frac{a_{i1}}{a_{11}}, \quad i = 2, \ldots, n \\
a_{ij}^{(2)} &= a_{ij} - m_{i1} a_{1j}, \quad i, j = 2, \ldots, n \\
b_i^{(2)} &= b_i - m_{i1} b_1, \quad i = 2, \ldots, n
\end{aligned}
$$

Si osservi che:

$$
\begin{pmatrix} 1 & 0 \\ -m_{21} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22}^{(2)} \end{pmatrix}
$$

Calcolando i determinanti, $1 \cdot \det(A^{(2)}) = a_{11} a_{22}^{(2)} \Rightarrow a_{22}^{(2)} \neq 0$.
$a_{22}^{(2)}$ è detto **perno o pivot**. Essendo non nullo, è possibile costruire

$$
L_2 = I - \boldsymbol{m}^{(2)} \boldsymbol{e}_2^T
$$

dove $\boldsymbol{m}^{(2)} = (0, 0, m_{32}, \ldots, m_{n2})^T$.

Metodo di eliminazione di Gauss

- Passo 2.

$$
L_2 [A_2 | \boldsymbol{b}_2] = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -m_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & -m_{n2} & 0 & \ldots & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \ldots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & a_{32}^{(2)} & \ldots & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \ldots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix} =
$$

$$
= \begin{pmatrix} a_{11} & \ldots & \ldots & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \ldots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \ldots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & a_{3n}^{(3)} & \ldots & a_{nn}^{(3)} & b_n^{(3)} \end{pmatrix}
$$

con

$$
m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}, \quad a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2} a_{2j}^{(2)}, \quad b_i^{(3)} = b_i^{(2)} - m_{i2} b_2^{(2)}, \quad i, j = 3, \ldots, n
$$

Metodo di eliminazione di Gauss

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{pmatrix} \quad \Rightarrow$$

$$\Rightarrow \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} \begin{pmatrix} 1 & & \\ -m_{21} & 1 & \\ -m_{31} & 0 & 1 \end{pmatrix} A^{(3)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{pmatrix}$$

Calcolando i determinanti, si ottiene $1 \cdot 1 \cdot \det(A^{(3)}) = a_{11} a_{22}^{(2)} a_{33}^{(3)} \Rightarrow a_{33}^{(3)} \neq 0$. Di nuovo il perno è non nullo e si può proseguire con un'ulteriore trasformazione di Gauss.

- Passo k.

$$L_{k-1} L_{k-2} \cdots L_1 [A_1 \ \boldsymbol{b}_1] = [A_k | \boldsymbol{b}_k] = \begin{pmatrix} a_{11} & \dots & & & & a_{n1} & b_1 \\ 0 & a_{22}^{(2)} & \dots & & & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \ddots & & & \vdots & \vdots \\ 0 & 0 & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} & b_k^{(k)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} & b_n^{(k)} \end{pmatrix}$$

Metodo di eliminazione di Gauss

Considerando l'intersezione tra le prime k righe e k colonne delle matrici, si ha $\det(A^{(k)}) = a_{11} a_{22}^{(2)} \cdots a_{kk}^{(k)} \Rightarrow a_{kk}^{(k)} \neq 0$.

Preso $L_k = I - \boldsymbol{m}^{(k)} \boldsymbol{e}_k^T$ con $\boldsymbol{m}^{(k)} = (0, \dots, 0, m_{k+1,k}, \dots, m_{nk})^T$, $m_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$, $i = k+1, \dots, n$,

$$L_k [A_k | \boldsymbol{b}_k] = [A_{k+1} | \boldsymbol{b}_{k+1}] =$$

$$= \begin{pmatrix} a_{11} & \dots & & & & & a_{n1} & b_1 \\ 0 & a_{22}^{(2)} & \dots & & & & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & \ddots & & & & \vdots & \vdots \\ 0 & 0 & 0 & a_{kk}^{(k)} & \dots & & a_{kn}^{(k)} & b_k^{(k)} \\ 0 & 0 & 0 & 0 & a_{k+1,k+1}^{(k+1)} & \dots & a_{k+1,n}^{(k+1)} & b_{k+1}^{(k+1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & a_{n,k+1}^{(k+1)} & \dots & a_{nn}^{(k+1)} & b_n^{(k+1)} \end{pmatrix}$$

Metodo di eliminazione di Gauss

Le prime k righe e k colonne restano invariate. Le posizioni (i,k), $i = k+1,\dots,n$ si annullano e

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad i,j = k+1,\dots,n$$
$$b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}$$

Dopo $n-1$ passi si ha:

$$L_{n-1}\cdots L_2 L_1 [A_1|\boldsymbol{b}_1] = [R|\boldsymbol{y}]$$

con $r_{ii} = a_{ii}^{(i)}$, $i = 1,\dots,n$, ossia i **perni o pivot**, $\boldsymbol{y} = (b_1\ b_2^{(2)}\ b_3^{(3)}\ \dots\ b_n^{(n)})^T$. Allora, premoltiplicando ambo i membri per $L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}$, si ha

$$L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}L_{n-1}\cdots L_2L_1[A_1|\boldsymbol{b}_1] = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}[R|\boldsymbol{y}]$$

da cui:

$$A = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}R \qquad \boldsymbol{b} = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}\boldsymbol{y}$$

Detto $L = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}$ segue che

$$A = LR \qquad \boldsymbol{b} = L\boldsymbol{y}$$

Metodo di eliminazione di Gauss

Poiché $L_i^{-1}L_j^{-1}$ per $i < j$ vale

$$(I+\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T)(I+\boldsymbol{m}^{(j)}\boldsymbol{e}_j^T) = I+\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T+\boldsymbol{m}^{(j)}\boldsymbol{e}_j^T+\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T\boldsymbol{m}^{(j)}\boldsymbol{e}_j^T = I+\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T+\boldsymbol{m}^{(j)}\boldsymbol{e}_j^T$$

perché $\boldsymbol{e}_i^T\boldsymbol{m}^{(j)} = 0$ per $i < j$. Pertanto

$$L = \begin{pmatrix} 1 & & & \\ \vdots & 1 & & \\ \vdots & m_{ij} & 1 & \\ \dots & \dots & \dots & 1 \end{pmatrix}$$ dove m_{ij} sono detti **moltiplicatori**

Si è determinata la fattorizzazione e la risoluzione di $L\boldsymbol{y} = \boldsymbol{b}$.
Se A è non singolare ($r_{nn} \neq 0$), resta da risolvere $R\boldsymbol{x} = \boldsymbol{y}$.

Inoltre

$$\det(A) = a_{11}a_{22}^{(2)}\cdots a_{nn}^{(n)}$$

Pertanto si è dimostrato che: se tutti i minori principali (eccetto al più l'ultimo) sono non nulli, i perni $a_{ii}^{(i)}$ per $i = 1,\dots,n-1$ sono non nulli e l'eliminazione di Gauss si può portare a termine (**strategia diagonale**). Se poi $a_{nn}^{(n)} \neq 0$, la matrice è non singolare e $\boldsymbol{x}^* = R^{-1}\boldsymbol{y}$.

Esempio (continua)

Nel caso dell'esempio precedente,

$$[A|\boldsymbol{b}] = [A_1|\boldsymbol{b}_1] = \left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{array}\right)$$

si ha che

$$L = L_1^{-1}L_2^{-1}L_3^{-1} = \begin{pmatrix} 1 & & & \\ 2 & 1 & & \\ 3 & 4 & 1 & \\ -1 & -3 & 0 & 1 \end{pmatrix}, \qquad R = \begin{pmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{pmatrix}$$

$$\boldsymbol{y} = L^{-1}\boldsymbol{b} \begin{pmatrix} 4 \\ -7 \\ 13 \\ -13 \end{pmatrix}$$

$$\det(A) = 13 \cdot 3 = 39$$

Complessità computazionale

La **complessità computazionale** dell'algoritmo di Gauss vale:

- Fattorizzazione:
  - $\frac{n(n-1)}{2}$ divisioni, per il calcolo dei moltiplicatori;
  - $\sum_{k=1}^{n-1}(n-k)^2 = \frac{n(n-1)(2n-1)}{6} = \mathcal{O}\left(\frac{n^3}{3}\right)$ somme e prodotti per il calcolo delle sottomatrici trasformati dalle trasformazioni elementari di Gauss;
- Termine noto: $\frac{n(n-1)}{2}$ prodotti e somme per fattorizzare più altrettanti $\frac{n(n-1)}{2}$ prodotti e somme per la soluzione.
- Determinante: $n-1$ prodotti

Codice Matlab

```
function [L, R, deter] = gauss1(A)
% GAUSS1 - Fattorizzazione di Gauss, versione 1
  n = max(size(A));
  for k = 1 : n-1
     if ( abs( A(k,k) ) < eps * norm(A,inf) )
        error('fattorizzazione non effettuabile.');
     else
        % for i = (k+1) : n
        %     A(i,k) = A(i,k)/A(k,k);
        %     for j = (k+1) : n
        %         A(i,j) = A(i,j)-A(i,k)*A(k,j);
        A((k+1):n, k) = A((k+1):n, k) / A(k,k);
        % operazione di base a livello 2: aggiornamento mediante
            diade
        A((k+1):n, (k+1):n) ...
           = A((k+1):n, (k+1):n) - A((k+1):n, k)*A(k, (k+1):n);
     end
  end
  deter = prod(diag(A));
  R = triu(A);
  L = eye(n) + tril(A, -1);  % si puo' migliorare...
end
```


Fattorizzazione di Gauss per matrici simmetriche

Se A è simmetrica e fattorizzabile mediante l'algoritmo di Gauss,

$$A = LDU \quad A = A^T = U^T D L^T$$

Pertanto $L = U^T$.

$$A = LDL^T$$

L'occupazione di memoria **si dimezza** circa poiché occorre memorizzare solo D ed L. La complessità computazionale diventa $\mathcal{O}(n^3/6)$ somme e prodotti, poiché **circa metà elementi non sono da calcolare.**

Condizioni sufficienti per l'esecuzione dell'algoritmo di Gauss con strategia diagonale

Ci sono due classi di matrici per il cui il metodo di Gauss si può portare a termine senza che nessun perno si annulli:

- le matrici strettamente diagonali dominanti per righe (per colonne) o le matrici non singolari diagonali dominanti per righe (o per colonne);
- le matrici simmetriche definite positive.

Matrici diagonali dominanti

Una matrice si dice **strettamente diagonale dominante per righe (per colonne)** se vale per ogni $i = 1, ..., n$ (oppure per ogni $j = 1, ..., n$):

$$|a_{ii}| > \sum_{i \neq j, j=1}^{n} |a_{ij}| \quad \left(\text{oppure } |a_{jj}| > \sum_{i \neq j, i=1}^{n} |a_{ij}| \right)$$

Una matrice si dice **diagonale dominante per righe (per colonne)** se vale per ogni $i = 1, ..., n$ (oppure per ogni $j = 1, ..., n$):

$$|a_{ii}| \geqslant \sum_{i \neq j, j=1}^{n} |a_{ij}| \quad \left(\text{oppure } |a_{jj}| \geqslant \sum_{i \neq j, i=1}^{n} |a_{ij}| \right)$$

Esempio

La seguente matrice è strettamente diagonale dominante per righe e diagonale dominante per colonne.

$$A = \begin{pmatrix} 2 & -1 & 0 \\ 1 & 4 & -2 \\ 0 & -1 & 2 \end{pmatrix}$$

Infatti per le righe si ha

$$\begin{aligned} 2 &> 1 \\ 4 &> 1 + |-2| \\ 2 &> |-1| \end{aligned}$$

Per le colonne

$$\begin{aligned} 2 &> 1 \\ 4 &> |-1| + |-1| \\ 2 &\geqslant |-2| \end{aligned}$$

Esempio

Una matrice strettamente diagonale dominante è non singolare.
Infatti il primo pivot non può essere nullo. Se così fosse, tutta la prima riga sarebbe nulla e la matrice non sarebbe più strettamente diagonale dominante.
Dopo aver fatto un passo dell'algoritmo di Gauss, si dimostra che la sottomatrice $A_2(2:n, 2:n)$ è ancora strettamente diagonale dominante. Pertanto $a_{22}^{(2)} \neq 0$; ripetendo il ragionamento, si dimostra che tutti i perni sono non nulli e dunque la matrice è non singolare.
Un ragionamento analogo vale per le matrici non singolari diagonali dominanti.

Matrici definite positive

Una matrice $A \in \mathbb{R}^{n\times n}$ simmetrica ($A = A^T$) si dice **definita positiva** se per ogni vettore $\boldsymbol{x}$ non nullo risulta $\boldsymbol{x}^T A\boldsymbol{x} > 0$.

PROPRIETÀ. Sia A una matrice simmetrica definita positiva.

- Una matrice è simmetrica definita positiva se e solo se tutti i suoi autovalori sono numeri reali positivi. Questa è una proprietà caratterizzante delle matrici definite positive.
- Se A è simmetrica definita positiva, essa è non singolare e la sua inversa è simmetrica definita positiva.
- Tutte le sottomatrici principali di testa di una matrice simmetrica definita positiva sono simmetriche definite positive.
- Se A è simmetrica definita positiva, poiché il determinante è il prodotto degli autovalori, $\det(A) > 0$.
- $\max |a_{ij}| \leqslant \max(a_{ii})$.

Matrici definite positive

Poiché tutti i minori principali primi sono positivi, l'algoritmo di eliminazione di Gauss può essere portato a termine, in quanto i perni sono positivi.
Le sottomatrici $A_k(k:n, k:n)$ che si ottengono ad ogni passo sono ancora simmetriche definite positive.
Pertanto

$$A = LR$$

poiché A è simmetrica, segue che $A = LDL^T$, ossia $R = DL^T$.

Inoltre vale il teorema di Von Neumann – Goldstine:

$$\max \left| a_{ij}^{(k)} \right| \leqslant \max(a_{ii})$$

ossia gli elementi che si incontrano nell'algoritmo non diventano mai troppo grandi rispetto agli elementi di A.
Per esempio, al secondo passo, siccome $a_{ij}^{(2)} = a_{ij} - \dfrac{a_{i1}}{a_{11}} a_{1j}$, si ha

$$\max |a_{ij}^{(2)}| \leqslant \max(a_{ii}^{(2)}) = \max\left(a_{ii} - \frac{a_{i1}^2}{a_{11}} \right) \leqslant \max(a_{ii})$$

Teorema di Cholesky

Una matrice $A \in \mathbb{R}^{n\times n}$ simmetrica è definita positiva se e solo se esiste una e una sola matrice $\mathcal{L}$ triangolare inferiore con elementi diagonali positivi tale che $A = \mathcal{L}\mathcal{L}^T$ (equivalentemente, posto $\mathcal{L} = \mathcal{R}^T, A = \mathcal{R}^T\mathcal{R}$).

Dim. "$\Rightarrow$" Se A è simmetrica definita positiva ammette fattorizzazione unica del tipo LDL^T. Inoltre, per ogni $\boldsymbol{x} \neq \mathbf{0}$, per la definizione di matrice definita positiva si ha

$$0 < \boldsymbol{x}^T A\boldsymbol{x} = \boldsymbol{x}^T LDL^T\boldsymbol{x} = \boldsymbol{y}^T D\boldsymbol{y} = \sum_{i=1}^{n} d_i y_i^2$$

dove si è posto $\boldsymbol{y} = L^T\boldsymbol{x} \neq \mathbf{0}$. Di conseguenza $d_i > 0$ per ogni $i = 1, ..., n$, e si può scrivere $D = \Delta\Delta$, con $\Delta = \mathrm{diag}(\sqrt{d_1}, \sqrt{d_2}, ..., \sqrt{d_n})$.
Posto $L\Delta = \mathcal{L}$, si è determinata una matrice triangolare inferiore con elementi diagonali $\mathcal{L}_{ii} = \sqrt{d_i} \cdot 1 > 0$. L'unicità della matrice segue dall'unicità della fattorizzazione LDU per matrici non singolari.

"$\Leftarrow$" Viceversa supponiamo che esista una e una sola matrice $\mathcal{L}$ triangolare inferiore con elementi diagonali positivi tale che $A = \mathcal{L}\mathcal{L}^T$; allora per ogni $\boldsymbol{x} \neq \mathbf{0}$, si ha

$$\boldsymbol{x}^T A\boldsymbol{x} = \boldsymbol{x}^T\mathcal{L}\mathcal{L}^T\boldsymbol{x} = \|\mathcal{L}^T\boldsymbol{x}\|_2^2 \geqslant 0$$

poiché $\mathcal{L}$ è non singolare e $\boldsymbol{x} \neq \mathbf{0}$, $\mathcal{L}^T\boldsymbol{x} \neq \mathbf{0}$ e $\boldsymbol{x}^T A\boldsymbol{x} > 0$, ossia A è definita positiva.

Questa fattorizzazione è detta fattorizzazione di Cholesky e caratterizza le matrici simmetriche definite positive. Se una matrice simmetrica non ammette tale fattorizzazione non è definita positiva.

Fattorizzazione di Cholesky (tecnica compatta)

Le **tecniche compatte** sfruttano le uguaglianze matriciali (prodotto righe per colonne), considerando le singole equazioni tra scalari in un ordine opportuno, detto **pavimentazione della matrice**:

$$A = \mathcal{L}\mathcal{L}^T$$

Se $j \leqslant i$, si ha che l'elemento di A di posizione ij è il prodotto scalare della riga i-esima di $\mathcal{L}$ per la riga j-esima di $\mathcal{L}$ (colonna j-esima di $\mathcal{L}^T$):

$$a_{ij} = \sum_{k=1}^{j} \ell_{ik}\ell_{jk} = \sum_{k=1}^{j-1} \ell_{ik}\ell_{jk} + \ell_{ij}\ell_{jj}$$

$$\Downarrow$$

$$\begin{cases} \ell_{ij} = \dfrac{a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\ell_{jk}}{\ell_{jj}} & j < i \\ \ell_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{j-1} \ell_{ik}^2} & j = i \end{cases}$$

per $j = 1, \ldots, n$.

Fattorizzazione di Cholesky (tecnica compatta)

Si costruisce la matrice $\mathcal{L}$ per colonne; essa si può memorizzare nel triangolo strettamente triangolare inferiore di A, ponendo gli elementi diagonali di $\mathcal{L}$ in un vettore ausiliario p.

p

1°	6°	10°	13°	15°

A

2°				
3°	7°			
4°	8°	11°		
5°	9°	12°	14°	

esempio di pavimentazione

Esempio

$$A = \begin{pmatrix} 4 & 1 & -2 \\ 1 & 5 & 1 \\ -2 & 1 & 4 \end{pmatrix}$$

$$\ell_{11} = \sqrt{4} = 2$$

$$\ell_{21} = \frac{1}{2} \qquad \ell_{22} = \sqrt{5 - \left(\frac{1}{2}\right)^2} = \frac{\sqrt{19}}{2}$$

$$\ell_{31} = \frac{-2}{2} = -1 \quad \ell_{32} = \frac{1 - (-1)\left(\frac{1}{2}\right)}{\frac{\sqrt{19}}{2}} = \frac{3}{\sqrt{19}} \quad \ell_{33} = \sqrt{4 - 1^2 - \left(\frac{3}{\sqrt{19}}\right)^2} = 2\sqrt{\frac{12}{19}}$$

Dunque la matrice A è definita positiva e

$$\mathcal{L} = \begin{pmatrix} 2 & & \\ \frac{1}{2} & \frac{\sqrt{19}}{2} & \\ -1 & \frac{3}{\sqrt{19}} & 2\sqrt{\frac{12}{19}} \end{pmatrix}$$

Esempio

Se invece si ha:

$$A = \begin{pmatrix} 4 & 1 & -2 \\ 1 & 5 & 1 \\ -2 & 1 & 9/19 \end{pmatrix}$$

allora

$$\ell_{11} = \sqrt{4} = 2$$

$$\ell_{21} = \frac{1}{2} \qquad \ell_{22} = \sqrt{5 - \left(\frac{1}{2}\right)^2} = \frac{\sqrt{19}}{2}$$

$$\ell_{31} = \frac{-2}{2} = -1 \quad \ell_{32} = \frac{1 - (-1)\left(\frac{1}{2}\right)}{\frac{\sqrt{19}}{2}} = \frac{3}{\sqrt{19}} \quad \ell_{33} = \sqrt{\frac{9}{19} - 1^2 - \left(\frac{3}{\sqrt{19}}\right)^2} = \sqrt{-1}$$

ℓ_{33} non è calcolabile. La matrice non è definita positiva.

Complessità dell'algoritmo di Cholesky

- $\mathcal{O}(n^3/6)$ somme e altrettanti prodotti;
- $\mathcal{O}(n^2/2)$ divisioni e n estrazioni di radici quadrate

Per evitare le estrazioni di radici si preferisce calcolare la fattorizzazione LDL^T, ossia non fare le divisioni per l_{jj} e non estrarre le radici.

- $\det(A) = \ell_{11}^2 \ell_{22}^2 \cdots \ell_{nn}^2$;
- se una delle quantità sotto radice che si incontrano nel calcolo è negativa, la matrice non è definita positiva;
- se A è associata al sistema $A\boldsymbol{x} = \boldsymbol{b}$, resta da risolvere

$$\begin{cases} \mathcal{L}\boldsymbol{y} = \boldsymbol{b} \\ \mathcal{L}^T\boldsymbol{x} = \boldsymbol{y} \end{cases}$$

Nel caso di $A = LDL^T$, si deve risolvere

$$\begin{cases} L\boldsymbol{y} = \boldsymbol{b} \\ L^T\boldsymbol{x} = D^{-1}\boldsymbol{y} \end{cases}$$

Codice Matlab

```
function [L, deter] = cholesky(A)
% CHOLESKY - Fattorizzazione di Cholesky
  n = max(size(A));
  deter = 1;
  for j = 1:n
     for i = j:n
        s = A(j,i) - A(i, 1:j-1)*A(j, 1:j-1)';
        if (i == j)
           if (s <= 0)
              error('matrice non definita positiva');
           else
              deter = deter*s;
              p(j) = sqrt(s);
           end
        else
           A(i,j) = s/p(j);
        end
     end
  end
  L = diag(p) + tril(A) - diag(diag(A));
end
```

In Matlab...

La funzione pre-definita di Matlab per la fattorizzazione di Cholesky è `chol`:

```
>> R = chol(A)
>> L = chol(A,'lower')   % genera matrice triangolare inferiore
>> R = chol(A,'upper')   % genera matrice triangolare superiore
>> [R,p] = chol(A)       % se A non e' def. pos., p = 1, altr. p = 0
>> [L,p] = chol(A,'lower') % se A non def. pos, p = 1, altr. p = 0
>> [R,p] = chol(A,'upper') % se A non def. pos, p = 1, altr. p = 0
```


Matrici di permutazione

Si dice **matrice elementare di permutazione** una matrice P_{ij} ottenuta dall'identità scambiando due righe (i-esima e j-esima, $i \neq j$) o due colonne:

$$P_{ij} = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & & & 1 & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & 1 & & & & 0 & \\ & & & & & & & & 1 \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{pmatrix}$$

Matrici di permutazione

$P_{ij}A$ ha come effetto di scambiare le righe i-esima e la j-esima di A.

Esempio.

$$P_{13}A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{pmatrix}$$

AP_{ij} ha come effetto di scambiare le colonne i-esima e j-esima di A.

Esempio.

$$AP_{13} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{pmatrix}$$

Matrici di permutazione

Una matrice elementare di permutazione è simmetrica, ossia $P_{ij} = P_{ij}^T$.
Inoltre $P_{ij}P_{ij} = I_n$.
Pertanto una matrice elementare di permutazione è simmetrica, ortogonale e involutoria ($P_{ij}^2 = I_n$).

Definizione.

Si dice matrice di permutazione P il prodotto di permutazioni elementari.

$$P = P_{ij}P_{kl}P_{rs}\ldots P_{uv}$$

$$P^T = (P_{ij}P_{kl}P_{rs}\ldots P_{uv})^T = P_{uv}\cdots P_{rs}P_{kl}P_{ij}$$
$$PP^T = P_{ij}P_{kl}P_{rs}\cdots P_{uv}P_{uv}\ldots P_{rs}P_{kl}P_{ij} = I_n$$

Una matrice P di permutazione è ortogonale.

Fattorizzazione di matrici con trasformazioni elementari di Gauss

Occorre una strategia che permette di **evitare i perni nulli** quando si applica l'algoritmo di Gauss. L'introduzione delle matrici di permutazione ha lo scopo di risolvere sistemi in cui A, pur essendo non singolare, non è fattorizzabile nella forma LR.

Ad esempio, la seguente matrice A non ammette fattorizzazione LR

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad A\boldsymbol{x} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Se si permutano le due equazioni del sistema, si ottiene un sistema equivalente e fattorizzabile secondo Gauss. Ciò significa premoltiplicare ambo i membri del sistema per una matrice di permutazione elementare:

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad PA\boldsymbol{x} = P\boldsymbol{b} \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_2 \\ b_1 \end{pmatrix}$$

Fattorizzazione di una matrice con righe permutate

Teorema

Sia A una matrice $m \times n$. Esiste una matrice di permutazione P $m \times m$ tale che

$$PA = LR$$

ove L è una matrice $m \times m$ triangolare inferiore con 1 sulla diagonale e R è una matrice $m \times n$ trapezoidale superiore, tale che $\mathrm{rank}(A) = \mathrm{rank}(R)$.
Se A è quadrata non singolare, anche R è quadrata delle stesse dimensioni, non singolare.

La dimostrazione è costruttiva.

Consideriamo la prima colonna di A. Se c'è un elemento $a_{r1} \neq 0$, si premoltiplica A per una matrice di permutazione elementare $P_1 = P_{r1}$ che scambia le righe di indici r e 1 per portare l'elemento in posizione **perno** e poi si esegue una trasformazione elementare di Gauss L_1 che annulla tutti gli elementi della prima colonna al di sotto dell'elemento diagonale. Se, al contrario, tutta la prima colonna è nulla, si pone $P_1 = L_1 = I_n$ e si procede:

$$A_2 = L_1 P_1 A_1 \quad \text{con} \quad A_1 = A$$

Fattorizzazione di una matrice con righe permutate

Al passo successivo, si cerca un elemento non nullo sulla seconda colonna, dalla posizione di riga 2 alla riga n. Se esiste tale elemento in una riga s, si porta in posizione **perno**, scambiando la seconda riga con la riga s (mediante la matrice elementare di permutazione $P_2 = P_{s2}$) e poi si esegue una trasformazione di Gauss L_2 per annullare tutti gli elementi al di sotto della posizione **perno**. Altrimenti, si pone $L_2 = P_2 = I_n$ e si prosegue.

Dopo $k = \min\{m-1, n\}$ passi si ottiene

$$L_k P_k \cdots L_3 P_3 L_2 P_2 L_1 P_1 A = R$$

dove $R = A_{k+1}$ è una matrice $m \times n$ trapezoidale superiore.

Fattorizzazione di una matrice con righe permutate

Se $m \leqslant n$, $k = m - 1$,

$$R = \begin{pmatrix} \ddots & \dots & \dots & \dots \\ & \ddots & \dots & \dots \\ & & \ddots & \dots \end{pmatrix}$$

Se $m > n$, $k = n$,

$$R = \begin{pmatrix} \ddots & \dots & \dots \\ & \ddots & \dots \\ & & \ddots \\ & & \end{pmatrix}$$

Gli elementi diagonali di R sono nulli in corrispondenza dei perni nulli.

Caso delle matrici quadrate non singolari

Esempio: caso di una matrice A quadrata di ordine n non singolare. Esiste un elemento diverso da 0 nella prima colonna (altrimenti ci sarebbe una colonna nulla): si eseguono una permutazione per portarlo in posizione perno e una trasformazione di Gauss:

$$L_1 P_1 A = \left(\begin{array}{c|ccc} a_{11}^{(2)} & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ \hline 0 & & & \\ \vdots & & \widetilde{A}_2 & \\ 0 & & & \end{array}\right) = A_2$$

Ora, nella prima colonna di $\widetilde{A}_2$ esiste almeno un elemento non nullo, altrimenti $\det(\widetilde{A}_2) = 0$ e di conseguenza $\det(A_2) = a_{11}\det(\widetilde{A}_2) = 0$, **ma A_2 è non singolare come prodotto di matrici non singolari**.
Al passo j:

$$L_{j-1}P_{j-1}\cdots L_1 P_1 A = \left(\begin{array}{cc|cc} a_{11}^{(j)} & \dots & \dots & a_{1n}^{(j)} \\ 0 & \ddots & \dots & \dots \\ \hline 0 & & a_{jj}^{(j)} & \dots \\ 0 & & & \tilde{A}_j \end{array}\right) = A_j$$

A_j è non singolare perché prodotto di matrici non singolari. Almeno uno degli elementi della prima colonna di $\widetilde{A}_j$ è non nullo (altrimenti $\det(A_j) = 0$).
Pertanto in $k = n-1$ passi si ottiene

$$L_{n-1}P_{n-1}\cdots L_1 P_1 A = R$$

con R non singolare.

Come si arriva a $PA = LR$? Esempio per matrici 4×4

$$L_3 P_3 L_2 P_2 L_1 P_1 A = R$$

Si può sempre scrivere

$$L_3 P_3 L_2 \boldsymbol{P_3 P_3} P_2 L_1 \boldsymbol{P_2 P_3 P_3 P_2} P_1 A = R$$

$$L_3 \underbrace{(P_3 L_2 P_3)}_{\widetilde{L}_2} \underbrace{(P_3 P_2 L_1 P_2 P_3)}_{\widetilde{L}_1} \underbrace{(P_3 P_2 P_1)}_{P} A = R$$

Dunque, ponendo $P = P_3 P_2 P_1$ e osservando che

- $\widetilde{L}_2 = P_3 L_2 P_3$ mantiene la stessa struttura di L_2 (P_3 esegue una permutazione tra la terza riga e una successiva),
- $\widetilde{L}_1 = P_3 P_2 L_1 P_2 P_3$ mantiene la stessa struttura di L_1,

si ha

$$L_3 \widetilde{L}_2 \widetilde{L}_1 PA = R$$

Dunque

$$PA = \widetilde{L}_1^{-1}\widetilde{L}_2^{-1}L_3^{-1}R = LR$$

con $L = \widetilde{L}_1^{-1}\widetilde{L}_2^{-1}L_3^{-1}$.

Come si arriva a $PA = LR$? Esempio per matrici 4×4

Effetto di pre- e post-moltiplicare una matrice di trasformazione elementare di Gauss per una matrice elementare di permutazione:

$$L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m_3^{(2)} & 1 & 0 \\ 0 & -m_4^{(2)} & 0 & 1 \end{pmatrix} \quad \Rightarrow \quad P_3 L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m_4^{(2)} & 0 & 1 \\ 0 & -m_3^{(2)} & 1 & 0 \end{pmatrix}$$

$$\Rightarrow \quad P_3 L_2 P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m_4^{(2)} & 1 & 0 \\ 0 & -m_3^{(2)} & 0 & 1 \end{pmatrix} = \widetilde{L}_2$$

Quando P_3 pre-moltiplica L_2, scambia le righe 3 e 4; quando la post-moltiplica, scambia le colonne 3 e 4.

Dunque le **strutture** di L_2 e di $\widetilde{L}_2$ sono identiche.

Caso generale I

Nel caso generale, si ha

$$L_k P_k L_{k-1} P_{k-1} L_{k-2} P_{k-2} \cdots L_3 P_3 L_2 P_2 L_1 P_1 A = R$$

Si pone

$$\left.\begin{aligned} S_k &= P_k \\ S_{k-j} &= P_k P_{k-1} \cdots P_{k-j} \\ S_{k-j-1} &= S_{k-j} P_{k-j-1} \end{aligned}\right\} j = 1, \ldots, k-2$$

$$S_1 = S_2 P_1 = P_k P_{k-1} \cdots P_2 P_1$$

S_{k-j} è invertibile e $S_{k-j}^{-1} = P_{k-j} \cdots P_{k-1} P_k$.
Introduciamo prodotti di matrici uguali all'identità:

$$L_k P_k L_{k-1} S_k^{-1} S_k P_{k-1} L_{k-2} S_{k-1}^{-1} S_{k-1} P_{k-2} \cdots L_3 S_4^{-1} S_4 P_3 L_2 S_3^{-1} S_3 P_2 L_1 S_2^{-1} S_2 P_1 A = R$$

$$L_k \underbrace{P_k}_{S_k} L_{k-1} S_k^{-1} \underbrace{S_k P_{k-1}}_{S_{k-1}} L_{k-2} S_{k-1}^{-1} \underbrace{S_{k-1} P_{k-2}}_{S_{k-2}} L_{k-3} \cdots L_3 S_4^{-1} \underbrace{S_4 P_3}_{S_3} L_2 S_3^{-1} \underbrace{S_3 P_2}_{S_2} L_1 S_2^{-1} \underbrace{S_2 P_1}_{S_1} A = R$$

$$L_k (S_k L_{k-1} S_k^{-1})(S_{k-1} L_{k-2} S_{k-1}^{-1})(S_{k-2} L_{k-2} S_{k-2}^{-1}) \cdots (S_4 L_3 S_4^{-1})(S_3 L_2 S_3^{-1})(S_2 L_1 S_2^{-1}) S_1 A = R$$

Caso generale II

Si pone $P = S_1 = P_k P_{k-1} \cdots P_1$ e si osserva che

$$S_i L_{i-1} S_i^{-1} = S_i \left(I_n - \boldsymbol{m}^{(i-1)} \boldsymbol{e}_{i-1}^T\right) S_i^{-1} = I_n - S_i \boldsymbol{m}^{(i-1)} \boldsymbol{e}_{i-1}^T S_i^{-1} = I_n - \tilde{\boldsymbol{m}}^{i-1} \boldsymbol{e}_{i-1}^T = \tilde{L}_{i-1}$$

ossia $\tilde{L}_{i-1}$ ha la stessa struttura di L_{i-1}.
Infatti, $S_i = P_k P_{k-1} \cdots P_i$ permuta elementi che stanno dalla posizione i a posizioni di indice maggiore:

$$S_i \boldsymbol{m}^{(i-1)} = \tilde{\boldsymbol{m}}^{(i-1)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ * \\ \vdots \\ * \end{pmatrix} \Bigg\} (i-1) \text{ zeri} \qquad \boldsymbol{e}_{i-1}^T S_i^{-1} = \boldsymbol{e}_{i-1}^T$$

Dunque gli m_{ij} sono solo permutati per righe:

$$L_k \tilde{L}_{k-1} \tilde{L}_{k-2} \cdots \tilde{L}_1 PA = R$$
$$\Rightarrow \quad PA = \underbrace{\tilde{L}_1^{-1} \cdots \tilde{L}_{k-1}^{-1} L_k^{-1}}_{L} R \qquad \text{con} \qquad L = \begin{pmatrix} 1 & & & & & \\ \tilde{m}_{21} & \ddots & & & & \\ \vdots & & 1 & & & \\ \tilde{m}_{i+1,1} & \dots & \tilde{m}_{i+1,i} & 1 & & \\ \vdots & & \vdots & & \ddots & \\ \tilde{m}_{n1} & \dots & \tilde{m}_{ni} & \dots & \dots & 1 \end{pmatrix}$$

Caso generale III

Infine, si dimostra che se $\text{rank}(A) = r$**, allora** $\text{rank}(R) = r$, ossia R ha solo r elementi diag. non nulli:

$$PA = LR \quad \text{rank}(A) = r$$
$$\Updownarrow$$
$$A \text{ ha } r \text{ colonne linearmente indipendenti (siano } j_1, \dots, j_r)$$
$$\Updownarrow$$
$$\lambda_1 A_{*,j_1} + \lambda_2 A_{*,j_2} + \dots + \lambda_r A_{*,j_r} = \mathbf{0} \;\Rightarrow\; \lambda_j = 0 \;\forall j = 1, \dots, r$$
$$\text{e con } A = P^T PA = P^T LR$$
$$\Updownarrow$$
$$\lambda_1 \left(P^T L R_{*,j_1}\right) + \dots + \lambda_r \left(P^T L R_{*,j_r}\right) = \mathbf{0} \;\Rightarrow\; \lambda_j = 0 \;\forall j = 1, \dots, r$$
$$\Updownarrow$$
$$P^T L \left(\lambda_1 R_{*,j_1} + \dots + \lambda_r R_{*,j_r}\right) = \mathbf{0} \;\Rightarrow\; \lambda_j = 0 \;\forall j = 1, \dots, r$$
$$\Updownarrow$$
$$\lambda_1 R_{*,j_1} + \dots + \lambda_r R_{*,j_r} = \mathbf{0} \;\Rightarrow\; \lambda_j = 0 \;\forall j = 1, \dots, r$$
$$\Updownarrow$$
$$\text{rank}(R) = r$$

Caso generale IV

Nel caso in cui occorre risolvere il sistema $A\boldsymbol{x} = \boldsymbol{b}$, con A non singolare, se è nota la fattorizzazione $PA = LR$, allora il sistema si riporta ai seguenti due sistemi:

$$\underbrace{PA}_{LR}\boldsymbol{x} = P\boldsymbol{b} \Rightarrow \begin{cases} L\boldsymbol{y} = P\boldsymbol{b} \\ R\boldsymbol{x} = \boldsymbol{y} \end{cases}$$

Inoltre vale che:

$$\det(A) = (-1)^{\sigma} r_{11} \cdots r_{nn}$$

dove σ è il numero di permutazioni **effettivamente** eseguite sulla ennupla $(1, 2, \ldots, n)$.
Infatti il determinante di ogni matrice elementare di permutazione diversa dall'identità vale -1.

Esempio

$$[A \,|\, \boldsymbol{b}] = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{array}\right)$$

$$P_1 = I_3 \qquad L_1 = \begin{pmatrix} 1 & & \\ -1 & 1 & \\ -1 & 0 & 1 \end{pmatrix} \qquad L_1 P_1 [A \,|\, \boldsymbol{b}] = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right)$$

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad L_2 = I_3 \qquad L_2 P_2 L_1 P_1 [A \,|\, \boldsymbol{b}] = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

$$R = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad L^{-1} P\boldsymbol{b} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad \Rightarrow \qquad x_3^* = 1 \quad x_2^* = -1 \quad x_1^* = 1$$

$$P = P_2 P_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad L^{-1} = L_2 P_2 L_1 P_2^T = L_2 \widetilde{L}_1 \quad L = \widetilde{L}_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & & \\ 1 & 1 & \\ 1 & 0 & 1 \end{pmatrix}$$

$$PA = LR \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & & \\ 1 & 1 & \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{pmatrix} \quad \det(A) = (-1)^1 \cdot 1 \cdot 1 = -1$$

Casi critici

La presenza di un perno nullo è causa d'arresto nell'algoritmo di eliminazione di Gauss. Occorre ricorrere alla **tecnica di ricerca del perno**.

Cosa accade in presenza di un perno piccolo?
Si consideri:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad \Rightarrow \quad [A_2 \,|\, \boldsymbol{b}_2] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.0001 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Teoricamente si può procedere perché $a_{22}^{(2)} \neq 0$:

$$m_{32} = \frac{1}{0.0001} = 10^4 \quad [A_3 \,|\, \boldsymbol{b}_3] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.0001 & 1 & 1 \\ 0 & 0 & -9999 & -10^4 \end{pmatrix}$$

La soluzione calcolata in aritmetica esatta è $x_3^* = 1.0001\ldots$, $x_2^* = -1.0001\ldots$, $x_1^* = 1$.

Casi critici

Usando però aritmetica finita con $t = 4$ cifre decimali:

$$\begin{aligned} \widetilde{x}_3 &= 1.000 \\ \widetilde{x}_2 &= (1 - 1)/10^{-4} = 0 \\ \widetilde{x}_1 &= (1 - 0 - 1)/1 = 0 \end{aligned}$$

Gli errori in $\widetilde{x}_2$ e $\widetilde{x}_1$ sono grandi. **Si noti che il perno $a_{22}^{(2)}$ è piccolo:** 10^{-4}.

Se c'è un piccolo errore nella determinazione di $\widetilde{x}_3$, questo viene amplificato di 10^4 nel calcolo di $\widetilde{x}_2$ e, di conseguenza, si ripercuote nella determinazione di $\widetilde{x}_1$. Allora occorre evitare che i perni siano molto piccoli (e i moltiplicatori grandi).

In sintesi:

- è un problema di stabilità dell'algoritmo;
- a un perno piccolo corrisponde un moltiplicatore grande;
- si può usare la strategia di permutare le righe (strategia di pivoting parziale) anche per "aggiustare le cose", ossia aumentare la stabilità.

Casi critici

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 1.0001 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{array}\right) \quad \Rightarrow \quad [A_2 \,|\, \boldsymbol{b}_2] = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 0.0001 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right)$$

Si può sfruttare la libertà di scelta del perno per rendere l'algoritmo più stabile (ossia meno sensibile agli errori di arrotondamento delle operazioni):

$$P_2[A_2 \,|\, \boldsymbol{b}_2] = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0.0001 & 1 & 1 \end{array}\right) \quad \Rightarrow \quad m_{32} = 10^{-4}$$

$$a_{33}^{(3)} = 1 - 10^{-4} = 0.9999 \quad b_3^{(3)} = 1 \quad [R \,|\, y] = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0.9999 & 1 \end{array}\right)$$

Con $t = 4$ cifre decimali,

$$\bar{x}_3 = 1.0000 \qquad \bar{x}_2 = -1 \qquad \bar{x}_1 = 1$$

Si noti che non c'è stato esagerato accrescimento nei valori di R, perché il moltiplicatore è minore di 1.

Casi critici

Calcolo dei residui.
Si definisce **residuo** il vettore $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{w}$, dove $\boldsymbol{w}$ è il vettore calcolato.

Allora, nell'esempio precedente si ha:

$$\tilde{\boldsymbol{r}} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad \bar{\boldsymbol{r}} = \begin{pmatrix} 0 \\ 10^{-4} \\ 0 \end{pmatrix}$$

Con la strategia che sceglie come perno l'elemento di modulo massimo sulla colonna, il residuo resta piccolo.

Strategia di pivoting parziale

Al passo k, si sceglie come perno l'elemento $a_{rk}^{(k)}$ tale che

$$\left|a_{rk}^{(k)}\right| = \max_{i=k,\ldots,n} \left|a_{ik}^{(k)}\right| \qquad k = 1, 2, \ldots, n-1$$

In tal modo per **i moltiplicatori** vale

$$|m_{ik}| \leqslant 1 \qquad \forall i = k+1, \ldots, n, \quad \forall k = 1, 2, \ldots, n-1.$$

Questa strategia garantisce che il residuo sia piccolo.

La complessità computazionale resta identica a quella dell'algoritmo con strategia diagonale. In più vengono eseguiti una serie di confronti per determinare i perni, pari a $\mathcal{O}(n^2/2)$.

Strategia di pivoting parziale

Attenzione!
Ciò non implica che la soluzione sia accettabile:

$$\left(\begin{array}{cc|c} 0.780 & 0.563 & 0.217 \\ 0.913 & 0.659 & 0.254 \end{array}\right)$$

Se si applica il pivoting parziale con $t = 3$ cifre decimali:

$$m_{21} = \frac{0.780}{0.913} = 0.854 \qquad [R \,|\, \boldsymbol{y}] = \left(\begin{array}{cc|c} 0.913 & 0.659 & 0.254 \\ 0.000 & 0.001 & 0.001 \end{array}\right)$$

$$\widetilde{x}_2 = 1, \ \widetilde{x}_1 = -0.443 \quad \widetilde{\boldsymbol{r}} = \begin{pmatrix} -0.000460 \\ -0.000541 \end{pmatrix} \quad \|\widetilde{\boldsymbol{r}}\|_\infty < 10^{-3}$$

Il residuo è piccolo, ma la soluzione esatta è $\boldsymbol{x}^* = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$.

Questo è un problema di mal condizionamento che non si risolve anche se si usa un algoritmo stabile.

Conclusione: la libertà di scelta del perno è sfruttata per dare maggiore stabilità all'algoritmo, **limitando** l'amplificarsi degli errori di arrotondamento nelle operazioni.

Strategia di pivoting parziale

Esempio.
Si usa $t = 3$ e $\beta = 10$ e chiamiamo τ la **soglia per trascurare valori** (ossia per assumerli numericamente zero):

$$A = \begin{pmatrix} 0.58 & -1.10 & -0.52 \\ -0.56 & 1.12 & 0.56 \\ 0.02 & 0.02 & 0.04 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 0.58 & -1.10 & -0.52 \\ & 0.06 & 0.058 \\ & & 0.0019 \end{pmatrix} \begin{pmatrix} 0.58 & -1.10 & -0.52 \\ & 0.06 & 0.058 \\ & & 0.0019 \end{pmatrix}$$

Se si pone $\tau = 10^{-3}$, si ha $r_{ii} > \tau \ \forall i = 1, 2, 3$ e $\mathrm{rank}(A) = 3$.
Se invece si pone $\tau = 2 \cdot 10^{-3}$, allora $r_{33} < \tau$ e $\mathrm{rank}(A) = 2$.
Se infine si pone $\tau = 10^{-1}$, allora $\mathrm{rank}(A) = 1$.

L'esempio mostra che una piccola variazione di τ può generare una grande variazione del numero di elementi che si assumono nulli. Si parla di **pseudorango numerico** di una matrice.

Esempio

Risolviamo con il metodo di eliminazione di Gauss con pivoting parziale il seguente sistema lineare:

$$\begin{cases} 2x_1 + x_3 = 3 \\ -3x_1 + 2x_2 + 2x_3 = -5 \\ 2x_2 + x_3 = -3 \end{cases}$$

Consideriamo la matrice completa.

$$[A \,|\, \boldsymbol{b}] = \left(\begin{array}{ccc|c} 2 & 0 & 1 & 3 \\ -3 & 2 & 2 & -5 \\ 0 & 2 & 1 & -3 \end{array}\right)$$

Considerata la prima colonna, l'elemento di valore assoluto massimo è in posizione $(2, 1)$. Considerata la matrice di permutazione $P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, si premoltiplica la matrice $[A|b]$ per P_1 per scambiare la prima e la seconda riga:

$$P_1 \left(\begin{array}{ccc|c} 2 & 0 & 1 & 3 \\ -3 & 2 & 2 & -5 \\ 0 & 2 & 1 & -3 \end{array}\right) = \left(\begin{array}{ccc|c} -3 & 2 & 2 & -5 \\ 2 & 0 & 1 & 3 \\ 0 & 2 & 1 & -3 \end{array}\right)$$

Esempio

Si esegue una trasformazione elementare di Gauss (per convenienza i moltiplicatori vengono memorizzati sugli elementi che si annullano):

$$\left(\begin{array}{ccc|c} -3 & 2 & 2 & -5 \\ -\frac{2}{3} & \frac{4}{3} & \frac{7}{3} & -\frac{1}{3} \\ 0 & 2 & 1 & -3 \end{array}\right)$$

Si considera ora la seconda colonna (seconda e terza equazione); poiché l'elemento di modulo massimo sta in posizione $(3,2)$, occorre permutare la seconda e terza riga attraverso la matrice $P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, ottenendo

$$P_2 \left(\begin{array}{ccc|c} -3 & 2 & 2 & -5 \\ -\frac{2}{3} & \frac{4}{3} & \frac{7}{3} & -\frac{1}{3} \\ 0 & 2 & 1 & -3 \end{array}\right) = \left(\begin{array}{ccc|c} -3 & 2 & 2 & -5 \\ 0 & 2 & 1 & -3 \\ -\frac{2}{3} & \frac{4}{3} & \frac{7}{3} & -\frac{1}{3} \end{array}\right)$$

Esempio

Si applica una ulteriore trasformazione di Gauss:

$$\left(\begin{array}{ccc|c} -3 & 2 & 2 & -5 \\ 0 & 2 & 1 & -3 \\ -\frac{2}{3} & \frac{2}{3} & \frac{5}{3} & \frac{5}{3} \end{array}\right)$$

Si ottiene così la matrice triangolare R e il vettore $\boldsymbol{y} = L^{-1}P\boldsymbol{b}$, dove $P = P_2P_1$.

$$R = \begin{pmatrix} -3 & 2 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & \frac{5}{3} \end{pmatrix} \qquad \boldsymbol{y} = \begin{pmatrix} -5 \\ -3 \\ \frac{5}{3} \end{pmatrix}$$

Risolvendo tramite l'algoritmo di sostituzione all'indietro si risolve il sistema $R\boldsymbol{x} = \boldsymbol{y}$, ottenendo

$$x_3^* = 1 \qquad x_2^* = -2 \qquad x_1 = 1$$

Inoltre si determinano L (prendendo la parte strettamente triangolare inferiore) e P

$$L = \begin{pmatrix} 1 & & \\ 0 & 1 & \\ -\frac{2}{3} & \frac{2}{3} & 1 \end{pmatrix} \qquad P = P_2P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Esempio

Si può verificare che $PA = LR$:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 & 0 & 1 \\ -3 & 2 & 2 \\ 0 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & & \\ 0 & 1 & \\ -\frac{2}{3} & \frac{2}{3} & 1 \end{pmatrix} \begin{pmatrix} -3 & 2 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & \frac{5}{3} \end{pmatrix}$$

Vale che $\det(A) = -3 \cdot 2 \cdot \frac{5}{3}(-1)^2 = -10$.

Codice Matlab per fattorizzazione con pivoting parziale

```
function [L, R, P, deter] = gauss2(A)
% GAUSS2 - Fattorizzazione di Gauss con pivoting parziale (versione 2)
[m, n] = size(A);
deter = 1;
temp = zeros(1, n);
P = 1 : n;
tol = eps * norm(A, inf);  % tolleranza verso lo zero (trascurabilita')
for j = 1 : min(m-1,n)
   [amax, ind] = max( abs(A(j:n, j)) );
   ind = ind + j - 1;
   if (ind ~= j)  % pivot non in posizione diagonale: occorre scambiare righe
      % scambio di indici
      aux = P(j);  P(j) = P(ind); P(ind) = aux;
      deter = -deter;
      % scambio di righe
      temp = A(ind,:); A(ind,:) = A(j,:); A(j,:) = temp;
   end
   deter = deter * A(j,j);
   if ( abs(A(j,j)) > tol )
      A(j+1:end, j) = A(j+1:end, j) / A(j,j);
      % operazione di base: aggiornamento mediante diadi
      A(j+1:end, j+1:end) = A(j+1:end, j+1:end) - A(j+1:end, j)*A(j, j+1:end);
   end
end
deter = deter * A(n,n);
R = triu(A);
L = eye(n) + tril(A(1:n,1:n), -1);
end
```

Strategia di pivoting totale

Al passo k si sceglie come perno l'elemento di modulo massimo della sottomatrice di A_k data dalle ultime $m-k+1$ righe ed $n-k+1$ colonne $\left(\widetilde{A}_k\right)$:

$$\left|a_{ij}^{(k)}\right| = \max_{\substack{r=k,\ldots,m \\ s=k,\ldots,n}} \left|a_{rs}^{(k)}\right| \qquad k = 1, 2, \ldots, \min\{m-1, n\}$$

Ciò richiede di eseguire due scambi: uno tra la riga i-esima e la riga k-esima, l'altro tra la colonna j-esima e la colonna k-esima.

Nel caso di una matrice $n \times n$ e di risoluzione di un sistema, lo scambio di colonne comporta un diverso ordinamento delle incognite.

Alla fine del procedimento, se si vogliono le componenti della soluzione nello stesso ordine con cui sono state date, è necessario un *riordinamento del vettore calcolato delle soluzioni*.

Dal punto di vista della complessità computazionale, oltre alle stesse operazioni richieste dall'algoritmo di Gauss, sono necessari $\mathcal{O}(n^3/3)$ confronti.

Strategia di pivoting totale

L'algoritmo si basa sul seguente teorema generale:

Teorema

Sia A una matrice $m \times n$ di rango r. Allora esistono due matrici di permutazione P e Q di ordine m ed n, rispettivamente, tali che:

$$PAQ = LR$$

dove $L \in \mathbb{R}^{m \times m}$ è triangolare inferiore a diagonale unitaria ed $R \in \mathbb{R}^{m \times n}$ è trapezoidale superiore di rango r. La matrice R ha esattamente r righe che, al di sopra della diagonale principale (compresa), possono contenere elementi non nulli:

$$PAQ = L \begin{pmatrix} \ddots & \cdots & \cdots \\ & \ddots & \cdots \\ & & 0 \end{pmatrix} \Bigg\} r \text{ righe} \qquad \operatorname{rank}(A) = r\,.$$

Se A è non singolare, R è triangolare superiore non singolare.

Strategia di pivoting totale

Nel caso la fattorizzazione $PAQ = LR$ sia utilizzata per risolvere il sistema $A\boldsymbol{x} = \boldsymbol{b}$ associato alla matrice A, con A non singolare, la risoluzione si ottiene ponendo:

$$PAQQ^T\boldsymbol{x} = P\boldsymbol{b}$$

Posto $Q^T\boldsymbol{x} = \boldsymbol{z}$, si risolvono i sistemi triangolari inferiore e superiore

$$\begin{cases} L\boldsymbol{y} = P\boldsymbol{b} \\ R\boldsymbol{z} = \boldsymbol{y} \end{cases}$$

e poi si riordinano le incognite secondo la relazione $\boldsymbol{x}^* = Q\boldsymbol{z}^*$, ossia eseguendo sulla soluzione $\boldsymbol{z}^*$ ottenuta le stesse permutazioni fatte sulle colonne di A.

Esempio

$$A = A_1 = \begin{pmatrix} 1 & 1 & 1 & 4 & 1 \\ -2 & -1 & 0 & 1 & 3 \\ -1 & 0 & 1 & 1.7 & 4 \\ 1 & 1.4 & 1.8 & 1 & 3 \\ 0 & 1 & 2 & 3 & 5 \end{pmatrix}$$

Si scambiano la prima riga con la quinta e la prima colonna con la quinta e poi si applica una trasformazione elementare di Gauss:

$$A_2 = L_1(P_1AQ_1) = \begin{pmatrix} 5 & 1 & 2 & 3 & 0 \\ 0 & -1.6 & -1.2 & -0.8 & -2 \\ 0 & -0.8 & -0.6 & -0.7 & -1 \\ 0 & 0.8 & 0.6 & -0.8 & 1 \\ 0 & 0.8 & 0.6 & 3.4 & 1 \end{pmatrix}$$

Si scambiano la seconda riga con la quinta e la seconda colonna con la quarta...

Esempio

. . . e poi si applica una trasformazione elementare di Gauss:

$$A_3 = L_2(P_2 L_1 P_1 A Q_1 Q_2) = \begin{pmatrix} 5 & 3 & 2. & 1 & 0 \\ 0 & 3.4 & 0.6 & 0.8 & 1 \\ 0 & 0 & -0.4765 & -0.6353 & -0.7941 \\ 0 & 0 & 0.7412 & 0.9882 & 1.2353 \\ 0 & 0 & -1.0588 & -1.4118 & -1.7647 \end{pmatrix}$$

Si scambiano la terza riga con la quinta e la terza colonna con la quinta e poi si applica una trasformazione elementare di Gauss:

$$A_4 = L_3(P_3 L_2 P_2 L_1 P_1 A Q_1 Q_2 Q_3) = \begin{pmatrix} 5 & 3 & 2 & 1 & 0 \\ 0 & 3.4 & 0.6 & 0.8 & 1 \\ 0 & 0 & -1.7647 & -1.4118 & -1.0588 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Si conclude che $\mathrm{rank}(R) = 3$.

In aritmetica finita, si deduce uno **pseudorango numerico** che non è detto che coincida con il rango teorico e dipende dalla tolleranza scelta.

Strategia di pivoting totale

- Sebbene il pivoting totale renda l'algoritmo di Gauss più stabile, esso ha un costo computazionale elevato in termini di confronti.
- Nelle librerie di algebra lineare di uso più diffuso e in Matlab, la routine di soluzione di un sistema lineare si basa quasi sempre sull'algoritmo di Gauss con pivoting parziale, che fornisce buoni risultati, senza costi eccessivi (in Matlab l'operatore **barra retroversa**, " \", esegue l'algoritmo di Gauss con pivoting parziale).
- Solo in casi particolari si ricorre al pivoting totale.

Sintesi

In conclusione, la strategia di pivoting ha un duplice scopo:

1. portare a termine l'algoritmo di eliminazione di Gauss su qualunque matrice, mediante la scelta di un perno diverso da zero; questo consente di risolvere sistemi associati a matrici non singolari;
2. rendere più stabile l'algoritmo di fattorizzazione, mediante la scelta di un perno "grande" (pivoting parziale o totale)

Fattorizzazioni in MATLAB

```
[L,R] = lu(A)          % se non e' necessario non c'e' pivoting
                       % (strategia diagonale)
[L,R,P] = lu(A)        % pivoting parziale
[L,R,P,Q] = lu(A)      % pivoting totale (SOLO PER MATRICI SPARSE)
[L,R,P,Q,D] = lu(A)    % con scalatura per righe (SOLO MATRICI SPARSE)
```

A, L, R, P, Q e D sono matrici. Esistono versioni di queste funzioni che restituiscono **vettori** delle permutazioni, invece delle matrici di permutazione.

Attenzione: la versione della funzione che applica l'algoritmo di fattorizzazione con pivoting totale può essere usata **solo con matrici sparse**, cioè definite con la funzione `sparse` di Matlab. In caso contrario la funzione dà errore.

Funzioni per la fattorizzazione LR con pivoting parziale si trovano nei contributi degli utenti sul sito di The MathWorks: File Exchange (si vedano ad esempio `lucp` e `gecp`).

Casi speciali

Ci sono matrici per cui è possibile ottenere una fattorizzazione con ***meno operazioni***, evitando operazioni inutili e risparmiando il tempo di calcolo, e/o ***diminuire l'occupazione di memoria***.

Un esempio è stato già analizzato: nel caso di matrici simmetriche definite positive, l'algoritmo di Cholesky consente di ottenere una fattorizzazione con circa metà operazioni e metà occupazione di memoria.

Altri casi significativi si ottengono per matrici con struttura particolare come le matrici a banda o quelle sparse.

Va tenuto presente che ci sono pro e contro nell'applicazione della strategia di pivoting alle matrici:

- **pro**: rende stabile il processo numerico di fattorizzazione;
- **contro**: modifica la struttura delle matrici; per esempio se A è simmetrica, PA non lo è; per mantenere la simmetria occorre fare permutazioni sulle righe e sulle colonne: PAP^T; se ci sono le condizioni (diagonale dominanza, o proprietà di definita positività), si evita il pivoting.

Matrici a banda

Si dice che A è una matrice a banda con banda superiore s e banda inferiore r se $a_{ij} = 0$, $j - i > s$, $i - j > r$.

$$A = \begin{pmatrix} a_{11} & \dots & a_{1,s+1} & & \\ \vdots & \ddots & \dots & a_{2,s+2} & \\ a_{r+1,1} & \dots & \ddots & \dots & \ddots \\ & a_{r+2,2} & \dots & \ddots & \vdots \\ & & \ddots & \dots & a_{nn} \end{pmatrix}$$

Se non è necessario pivoting, la fattorizzazione $A = LR$ produce una matrice L di banda inferiore r e una matrice R di banda superiore s. Si ha dunque minore occupazione di memoria e minore complessità computazionale (vanno calcolati **ad ogni passo** al più r moltiplicatori e vengono trasformati al più rs elementi).

$$L = \begin{pmatrix} 1 & & & & \\ \vdots & 1 & & & \\ \ell_{r+1,1} & \vdots & 1 & & \\ & \ell_{r+2,2} & \ddots & 1 & \\ & & \dots & \dots & 1 \end{pmatrix} \quad R = \begin{pmatrix} r_{11} & \dots & r_{1,s+1} & & \\ & \ddots & \vdots & r_{2,s+2} & \\ & & \ddots & \vdots & \ddots \\ & & & \ddots & \vdots \\ & & & & r_{nn} \end{pmatrix}$$

Esempio

Consideriamo una matrice a banda con $r = 2$ ed $s = 1$:

$$A = \begin{pmatrix} 5 & 2 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 \\ 2 & 4 & 9 & -1 & 0 \\ 0 & -1 & 2 & 5 & 1 \\ 0 & 0 & 2 & 3 & 7 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.2 & 1 & 0 & 0 & 0 \\ 0.4 & 0.9412 & 1 & 0 & 0 \\ 0 & -0.2941 & 0.2847 & 1 & 0 \\ 0 & 0 & 0.2482 & 0.6146 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} 5 & 2 & 0 & 0 & 0 \\ 0 & 3.4 & 1 & 0 & 0 \\ 0 & 0 & 8.0588 & -1 & 0 \\ 0 & 0 & 0 & 5.2847 & 1 \\ 0 & 0 & 0 & 0 & 6.3854 \end{pmatrix}$$

Senza pivoting, L ed R mantengono la struttura a banda (inferiore e superiore rispettivamente) di A.

Esempio

Se è necessario **pivoting parziale,** L ha al più r elementi non nulli per ogni colonna; R ha banda superiore $s + r$. **Il pivoting parziale distrugge la struttura della matrice e aumenta la complessità computazionale (ogni passo ha** $r(s + r + 1)$ **prodotti)**.

Esempio. $r = 2$, $s = 1$.

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 \\ 2 & 4 & 9 & -1 & 0 \\ 0 & -1 & 2 & 1 & 1 \\ 0 & 0 & 2 & 3 & 7 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.5 & 1 & 0 & 0 & 0 \\ 0.5 & 0 & 1 & 0 & 0 \\ 0 & 0 & -0.4444 & 1 & 0 \\ 0 & -0.2 & -0.6889 & 0.3862 & 1 \end{pmatrix} \qquad R = \begin{pmatrix} 2 & 4 & 9 & -1 & 0 \\ 0 & 5 & 5.5 & -0.5 & 0 \\ 0 & 0 & -4.5 & 0.5 & 0 \\ 0 & 0 & 0 & 3.2222 & 7 \\ 0 & 0 & 0 & 0 & -1.7034 \end{pmatrix}$$

La L non ha più struttura a banda, la R ha banda superiore 3.

Matrici tridiagonali

Si consideri una matrice A tridiagonale strettamente diagonale dominante oppure non singolare diagonale dominante o definita positiva (non serve pivoting). Tre vettori $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{b}$ sono sufficienti a memorizzare gli elementi non nulli delle tre diagonali di A. Supponiamo che si debba risolvere il sistema

$$A\boldsymbol{x} = \boldsymbol{f}$$

$$A = \begin{pmatrix} d_1 & b_1 & & & & \\ c_2 & d_2 & b_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & b_{n-1} \\ & & & & c_n & d_n \end{pmatrix}$$

dove $c_1 = b_n = 0$.

Matrici tridiagonali

Poiché A è a banda con banda superiore e inferiore 1, L è triangolare inferiore unitaria con banda inferiore 1 e R è triangolare superiore a banda superiore 1.

$$A = \begin{pmatrix} 1 & & & \\ \ell_2 & 1 & & \\ & \ddots & 1 & \\ & & \ell_n & 1 \end{pmatrix} \begin{pmatrix} u_1 & s_1 & & \\ & \ddots & \ddots & \\ & & \ddots & s_{n-1} \\ & & & u_n \end{pmatrix} = LR$$

Si sfruttano le uguaglianze matriciali per ottenere la fattorizzazione (**metodo di pavimentazione**), ossia s_i, ℓ_i u_i. Si osservi che:

$$b_i = a_{i,i+1} = \begin{pmatrix} 0 & \dots & \ell_i & \underbrace{1}_{i} & 0 & \dots \end{pmatrix} \begin{pmatrix} \vdots \\ 0 \\ s_i \\ u_{i+1} \\ 0 \\ \vdots \end{pmatrix} = s_i \quad \text{per } i = 1, \dots, n-1$$

Matrici tridiagonali

$$c_i = a_{i,i-1} = \begin{pmatrix} 0 & \dots & \ell_i & \underbrace{1}_{i} & 0 & \dots \end{pmatrix} \begin{pmatrix} \vdots \\ 0 \\ b_{i-2} \\ u_{i-1} \\ 0 \\ \vdots \end{pmatrix} = \ell_i u_{i-1} \quad \Rightarrow \quad \begin{array}{l} \ell_i = \dfrac{c_i}{u_{i-1}} \\ \text{per } i = 2, \dots, n \end{array}$$

$$\begin{cases} d_1 = u_1 \\ \\ d_i = a_{ii} = \begin{pmatrix} 0 & \dots & \ell_i & \underbrace{1}_{i} & 0 & \dots \end{pmatrix} \begin{pmatrix} \vdots \\ 0 \\ b_{i-1} \\ u_i \\ 0 \\ \vdots \end{pmatrix} = \ell_i b_{i-1} + u_i \end{cases} \quad \Rightarrow \quad \begin{array}{l} u_1 = d_1 \\ u_i = d_i - \ell_i b_{i-1} \\ \text{per } i = 2, \dots, n \end{array}$$

L'algoritmo di fattorizzazione risultante è il seguente (**algoritmo di Thomas**):

$u_1 = d_1$
for $i = 2, \dots, n$ **do**
  $\ell_i = \dfrac{c_i}{u_{i-1}}$
  $u_i = d_i - \ell_i b_{i-1}$
end for

Matrici tridiagonali

Restano da risolvere i sistemi

$$Ly = f$$
$$Rx = y$$

Tenendo conto della struttura di L ed R, i due sistemi triangolare inferiore e superiore si risolvono nel seguente modo:

$$f_i = f_i - \ell_i f_{i-1}, \qquad i = 2, ..., n$$

$$f_n = f_n/u_n; \quad f_i = (f_i - b_i f_{i+1})/u_i \quad i = n-1, ..., 1$$

Complessità computazionale

- Fattorizzazione: $n-1$ divisioni, $n-1$ somme e $n-1$ prodotti;
- Soluzione: n divisioni, $2(n-1)$ somme, $2(n-1)$ prodotti.

Anche nel caso di matrici di Hessemberg, l'applicazione del metodo di Gauss (anche con pivoting) comporta un abbassamento della complessità computazionale. Infatti in totale si eseguono $n-1$ divisioni per calcolare i moltiplicatori e $\mathcal{O}(n^2/2)$ prodotti e altrettante somme per il calcolo degli elementi di R.

Matrici sparse

Le matrici sparse sono quelle in cui il numero di elementi non nulli è proporzionale alla dimensione della matrice ($\mathcal{O}(n)$ invece di $\mathcal{O}(n^2)$). **La loro memorizzazione richiede minore occupazione di memoria se si memorizzano solo gli elementi non nulli (memorizzazione sparsa).**
Si ha minore complessità computazionale nelle operazioni (es. prodotto matrice-vettore costa un numero di prodotti e somme pari al numero degli elementi non nulli della matrice).
Per matrici sparse, **se non si esegue un riordinamento delle righe e colonne**, i metodi visti creano dei riempimenti, detti fill-in, che distruggono la struttura della matrice.

Esempio

Il fattore di Cholesky della matrice sparsa A è denso.

$$A = \begin{pmatrix} 4 & 1 & 2 & 1/2 & 2 \\ 1 & 1/2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 1/2 & 0 & 0 & 5/8 & 0 \\ 2 & 0 & 0 & 0 & 16 \end{pmatrix}$$

$$\Downarrow$$

$$\mathcal{L} = \begin{pmatrix} 2 & & & & \\ 0.5 & 0.5 & & & \\ 1 & -1 & 1 & & \\ 0.25 & -0.25 & -0.5 & 0.5 & \\ 1 & -1 & -2 & -3 & 1 \end{pmatrix}$$

Riordinamento

Esistono tecniche che riordinano una matrice per minimizzare il fill-in.
Una delle più note è il criterio di Markowitz: seguendo questo criterio, ad ogni passo k si sceglie come perno l'elemento della sottomatrice $\tilde{A}_k$ per cui la seguente quantità è minima:

$$(row_i - 1)(col_j - 1)$$

dove row_i e col_j sono i numeri di elementi non nulli sulla riga i-esima e sulla j-esima colonna di $\tilde{A}_k$.

Riordinamento

Se una matrice è bfsimmetrica, il criterio di Markowitz equivale al minimum degree ordering.
Quest'ultima tecnica consiste nel costruire il grafo associato alla matrice: se n è la dimensione della matrice, si considera un grafo di n nodi numerati da 1 a n e poi, per ogni elemento $a_{ij} \neq 0$ si genera un arco che connette il nodo i al nodo j.
Si dice grado di un nodo il numero di archi che partono da quel nodo.

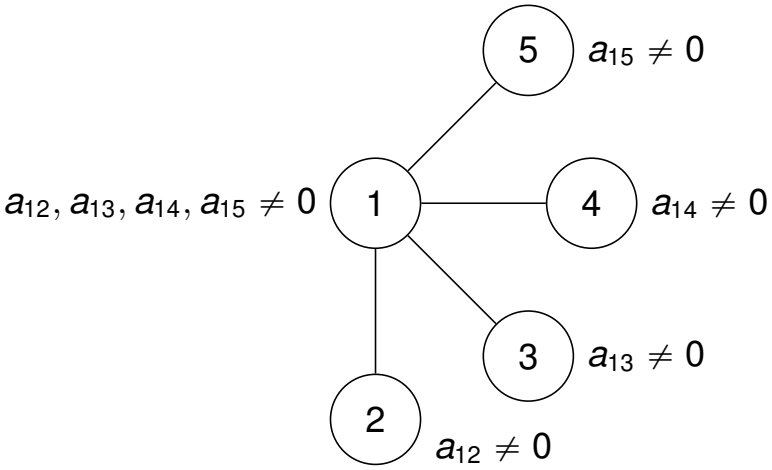
Si ordinano le righe e le colonne della matrice a partire da quella associata al nodo di grado minimo (grado di un nodo=numero di archi che partono da tale nodo) e poi via via considerando i nodi di grado superiore.
Questo è dovuto al fatto che ogni passo di Gauss equivale ad eliminare il nodo associato alla colonna processata e quando si elimina un nodo, tutti i nodi a lui connessi si connettono tra di loro, riflettendo la creazione di elementi non nulli.
Pertanto si scelgono i nodi con minori connessioni prima degli altri.

Riordinamento

In Matlab si può ottenere la permutazione corrispondente a un minimum degree reordering mediante la funzione `symamd` (versione più efficiente di `symmmd`).

```
>> p = symamd(A);
>> L = chol(A(p,p));
```

Il grafo associato alla matrice dell'esempio precedente è il seguente:



Riordinamento

In questo caso l'ordinamento può essere: 2,3,4,5,1. Pertanto la matrice con righe e colonne permutate diventa:

$$PAP^T = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 0 & 0 & 5/8 & 0 & 1/2 \\ 0 & 0 & 0 & 16 & 2 \\ 1 & 2 & 1/2 & 2 & 4 \end{pmatrix}$$

$$\Downarrow$$

$$\mathcal{L} = \begin{pmatrix} 1/\sqrt{2} & & & & \\ 0 & \sqrt{3} & & & \\ 0 & 0 & \sqrt{5}/(2\sqrt{2}) & & \\ 0 & 0 & 0 & 4 & \\ \sqrt{2} & 2/\sqrt{3} & \sqrt{2/5} & 1/2 & 1/\sqrt{60} \end{pmatrix}$$

Si noti che in questo caso non c'è alcun fill-in.

Trasformazioni ortogonali

Si possono definire trasformazioni elementari che permettono di ottenere fattorizzazioni di matrici analoghe alla fattorizzazione di Gauss.
In particolare si intende definire trasformazioni elementari ortogonali, associate a matrici Q la cui inversa coincide con la trasposta.
Tali trasformazioni mantengono la norma euclidea di vettori:

$$\|Q\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^T Q^T Q \boldsymbol{x}} = \sqrt{\boldsymbol{x}^T \boldsymbol{x}} = \|\boldsymbol{x}\|_2$$

e quella euclidea e di Frobenius di matrici:

$$\|QA\|_2 = \sqrt{\lambda_{\max}(A^T Q^T Q A)} = \sqrt{\lambda_{\max}(A^T A)} = \|A\|_2$$
$$\|QA\|_F = \sqrt{\mathrm{trace}(A^T Q^T Q A)} = \sqrt{tr(A^T A)} = \|A\|_F$$

Inoltre si ha

$$\|Q\|_2 = \sqrt{\lambda_{\max}(Q^T Q)} = \sqrt{\lambda_{\max}(I_n)} = 1$$

Trasformazioni elementari di Givens (rotazioni elementari)

Si chiama **trasformazione elementare di Givens** o **matrice di rotazione elementare** G_{ij} di ordine n una matrice che coincide con l'identità di ordine n eccetto nelle posizioni (i,j), (j,i), (i,i) e (j,j), nelle quali stanno due valori c ed s, che dipendono da un solo parametro ϕ:

$$G_{ij} = \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & c & & & & s & & \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & -s & & & & c & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{pmatrix}$$

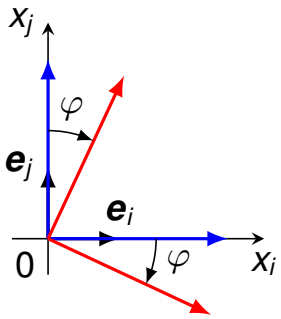
Infatti $c^2 + s^2 = 1$, $c = \cos(\varphi)$, $s = \sin(\varphi)$.

Esempio

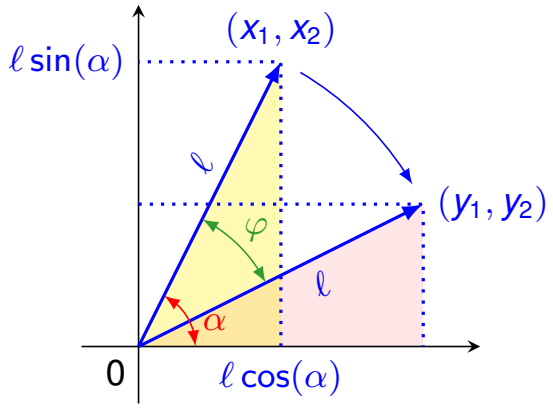
Se $n = 8$, $\varphi = \dfrac{\pi}{4}$, posto $i = 3$, $j = 6$, la matrice di rotazione di Givens $G_{3,6}$ è data da

$$G_{3,6} = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \frac{\sqrt{2}}{2} & & & \frac{\sqrt{2}}{2} & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & -\frac{\sqrt{2}}{2} & & & \frac{\sqrt{2}}{2} & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix}$$

Una matrice G_{ij} esprime una rotazione di ampiezza φ nell'iperpiano individuato dai versori $\boldsymbol{e}_i$ ed $\boldsymbol{e}_j$ in $\mathbb{R}^n$.



Interpretazione geometrica per $n = 2$



Premoltiplicare un vettore per una matrice di Givens equivale ad una rotazione di ampiezza φ. Dato un vettore $\boldsymbol{x} = (x_1, x_2)^T$ e una rotazione di ampiezza φ si ha:

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

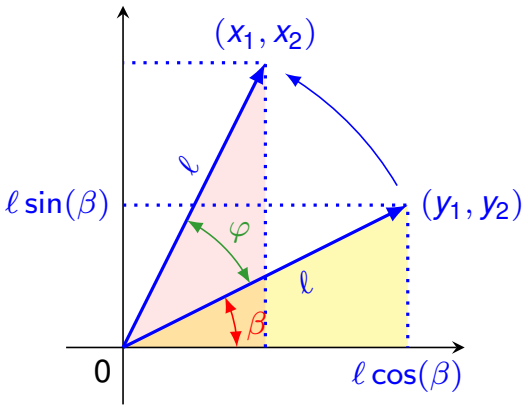
Infatti, poiché $\ell = \sqrt{x_1^2 + x_2^2}$, si ha:

$$x_1 = \ell \cos(\alpha) \qquad x_2 = \ell \sin(\alpha)$$

$$y_1 = \ell \cos(\alpha - \varphi) = \ell\big(\cos(\alpha)\cos(\varphi) + \sin(\alpha)\sin(\varphi)\big) = x_1 \cos(\varphi) + x_2 \sin(\varphi) = x_1 c + x_2 s$$

$$y_2 = \ell \sin(\alpha - \varphi) = \ell\big(\sin(\alpha)\cos(\varphi) - \cos(\alpha)\sin(\varphi)\big) = x_2 \cos(\varphi) - x_1 \sin(\varphi) = x_2 c - x_1 s$$

Interpretazione geometrica per $n = 2$



Viceversa, partendo da $\boldsymbol{y}$, mediante una rotazione di ampiezza φ in senso antiorario si ottiene $\boldsymbol{x}$:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Infatti da

$$y_1 = \ell \cos(\beta) \qquad y_2 = \ell \sin(\beta)$$

discende che

$$x_1 = \ell \cos(\beta + \varphi) = \ell\big(\cos(\beta)\cos(\varphi) - \sin(\beta)\sin(\varphi)\big) = y_1 \cos(\varphi) - y_2 \sin(\varphi) = y_1 c - y_2 s$$
$$x_2 = \ell \sin(\beta + \varphi) = \ell\big(\sin(\beta)\cos(\varphi) + \cos(\beta)\sin(\varphi)\big) = y_2 \cos(\varphi) + y_1 \sin(\varphi) = y_2 c + y_1 s$$

Trasformazioni di Givens

In generale si osserva che:

$$G_{ij}\boldsymbol{x} = \boldsymbol{y} \qquad \text{dove} \qquad \begin{cases} y_k = x_k & k \neq i, j \\ y_i = c \cdot x_i + s \cdot x_j \\ y_j = -s \cdot x_i + c \cdot x_j \end{cases}$$

È sempre possibile trovare un valore di φ per cui $y_j = 0$. Basta trovare c ed s tali che

$$\begin{aligned} c^2 + s^2 &= 1 \\ y_j = 0 &= -s \cdot x_i + c \cdot x_j \end{aligned} \qquad \begin{cases} c = \dfrac{x_i}{\sqrt{x_i^2 + x_j^2}} \\ s = \dfrac{x_j}{\sqrt{x_i^2 + x_j^2}} \end{cases}$$

Formule più stabili

Se $|x_i| < |x_j|$, si pone $t = \dfrac{x_i}{x_j} \Rightarrow s = \dfrac{1}{\sqrt{1 + t^2}}$, $c = t \cdot s$

Se $|x_j| < |x_i|$, si pone $t = \dfrac{x_j}{x_i} \Rightarrow c = \dfrac{1}{\sqrt{1 + t^2}}$, $s = t \cdot c$

La complessità del calcolo di c ed s è di 4 prodotti e quella della sola trasformazione è ancora di 4 prodotti.

Esempio.
Sia $\boldsymbol{x} = (-1, 2, 1, 3)^T$. Si vuole trovare la rotazione che lascia invariati il secondo e quarto elemento e annulla il terzo elemento.
Allora si ha

$$G_{1,3}\boldsymbol{x} = \begin{pmatrix} c & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 1 \\ 3 \end{pmatrix} = \begin{pmatrix} -c+s \\ 2 \\ s+c \\ 3 \end{pmatrix} = \begin{pmatrix} 2/\sqrt{2} \\ 2 \\ 0 \\ 3 \end{pmatrix}$$

Deve essere $s + c = 0$ e $s^2 + c^2 = 1$. Dunque $c = -\dfrac{1}{\sqrt{2}}$, $s = \dfrac{1}{\sqrt{2}}$.


```
function [c, s] = givensrot(x1,x2)
% GIVENSROT - Rotazione elementare di Givens
% si determinano c ed s tali da annullare l'elemento y2
  if (x2 == 0)
     c = 1; s = 0;
  else
     if (abs(x2) >= abs(x1))
        t = x1/x2; s = 1/sqrt(1+t^2); c = s*t;
     else
        t = x2/x1; c = 1/sqrt(1+t^2); s = t*c;
     end
  end
end
```

(Questo codice è migliorabile)

Applicazione di una rotazione a una matrice

Si osserva che **premoltiplicare** una matrice per G_{ij} significa sostituire alle righe i-esima e j-esima una loro combinazione lineare:

$$G_{ij}A = B \qquad \begin{cases} b_{k\ell} = a_{k\ell} & k \neq i,j; \ell = 1,\dots,n \\ b_{i\ell} = c \cdot a_{i\ell} + s \cdot a_{j\ell} & \ell = 1,\dots,n \\ b_{j\ell} = -s \cdot a_{i\ell} + c \cdot a_{j\ell} & \ell = 1,\dots,n \end{cases}$$

Pertanto, si può dimostrare che:

$$G_{ij}G_{ij}^T = \Delta \qquad \delta_{k\ell} = (I)_{k\ell} \quad k \neq i,j;\ \ell = 1,\dots,n$$
$$\delta_{ii} = c^2 + s^2 = 1 \qquad \delta_{ij} = -cs + cs = 0$$
$$\delta_{ji} = -sc + cs = 0 \qquad \delta_{jj} = s^2 + c^2 = 1$$

dunque G_{ij} è ortogonale. Infatti, l'inversa di G_{ij} è G_{ij}^T. Dunque G_{ij} rappresenta un'**isometria** (mantiene invariate le lunghezze, ossia la norma euclidea, e gli angoli). **Postmoltiplicare** A per una matrice G_{ij} vuol dire sostituire alle colonne i-esima e j-esima una loro combinazione lineare:

$$AG_{ij} = B \qquad \begin{cases} b_{k\ell} = a_{k\ell} & \ell \neq i,j;\ k = 1,\dots,n \\ b_{ki} = c \cdot a_{ki} - s \cdot a_{kj} & k = 1,\dots,n \\ b_{kj} = s \cdot a_{ki} + c \cdot a_{kj} & k = 1,\dots,n \end{cases}$$

Complessità del prodotto con una rotazione elementare: $\mathcal{O}(4n)$ prodotti.

Come usare le trasformazioni di Givens?

Data una matrice A, è possibile trovare G_{ij} tale che $B = G_{ij}A$ abbia l'elemento $b_{ji} = 0$:

$$\begin{aligned} c^2 + s^2 &= 1 \\ b_{ji} = 0 &= -s \cdot a_{ii} + c \cdot a_{ji} \end{aligned} \qquad \begin{cases} c = \dfrac{a_{ii}}{\sqrt{a_{ii}^2 + a_{ji}^2}} \\ s = \dfrac{a_{ji}}{\sqrt{a_{ii}^2 + a_{ji}^2}} \end{cases}$$

(è preferibile usare le formule stabili). Inoltre, con $\mathcal{O}(4n)$ prodotti, si trova:

$$\begin{cases} b_{k\ell} = a_{kl} & k \neq i,j;\ \ell = 1,\dots,n \\ b_{i\ell} = c \cdot a_{i\ell} + s \cdot a_{j\ell} & \ell = 1,\dots,n \\ b_{j\ell} = -s \cdot a_{i\ell} + c \cdot a_{j\ell} & \ell = 1,\dots,n \end{cases}$$

Teorema (Teorema generale di fattorizzazione *QR*)

Sia A una matrice $m \times n$. Esiste una matrice Q ortogonale di ordine m tale che $A = QR$, dove R è una matrice trapezoidale superiore $m \times n$. Inoltre $\mathrm{rank}(A) = \mathrm{rank}(R)$.

La dimostrazione dell'esistenza di Q è costruttiva.
Data $A \in \mathbb{R}^{m \times n}$, è possibile costruire successivamente $G_{1,2}, G_{1,3}, \ldots, G_{1,m}$ tali che

$$G_{1,m} \cdots G_{1,3} G_{1,2} A = \begin{pmatrix} * & * & \ldots & * \\ 0 & * & \ldots & * \\ 0 & * & \ldots & * \\ 0 & * & \ldots & * \end{pmatrix}$$

dove $G_{1,2}$ annulla l'elemento in posizione $(2,1)$, $G_{1,3}$ annulla quello in posizione $(3,1)$,...

In seguito, premoltiplicando quanto ottenuto per opportune matrici:

$$G_{2,m} \cdots G_{2,4} G_{2,3} (G_{1,m} \cdots G_{1,3} G_{1,2} A)$$

si annulla la parte della seconda colonna al di sotto della diagonale; premoltiplicando per

$$\begin{cases} G_{3,m} \cdots G_{3,5} G_{3,4} & \text{si annulla la parte sottodiagonale della terza colonna,} \\ G_{4,m} \cdots G_{4,5} & \text{si annulla la parte sottodiagonale della quarta colonna,} \\ \quad \vdots & \\ G_{r,m} \cdots G_{r,r+1} & \text{si annulla la parte sottodiagonale dell'}r\text{-esima colonna,} \end{cases}$$

dove $r = \min(m-1, n)$, e infine si ottiene una matrice trapezoidale superiore. In conclusione

$$\prod_{i=r,\ldots,1} \left(\prod_{j=m,\ldots,i+1} G_{ij} \right) A = R$$

Inoltre, posto $Q^T = \prod_{i=r,\ldots,1;\ j>i} G_{ij}$, si ottiene $Q^T A = R$, da cui:

$$A = QR$$

Fattorizzazione QR

Essendo Q ortogonale e dunque non singolare, il rango di A e di R sono uguali.
In generale,

$$Q = \prod_{i=1,\ldots,r;\ j>i} G_{ij}^T$$

e si può calcolare nel seguente modo:

$Q \leftarrow I$;
for $i = 1, 2, \ldots, r$ **do**
  for $j = i+1, \ldots, m$ **do**
    $Q \leftarrow QG_{ij}^T$
  end for
end for

Se $m = n$, la complessità computazionale è $\mathcal{O}(4n^3/3)$ prodotti e somme e $\mathcal{O}(n^2/2)$ radici quadrate.
Infatti per annullare la colonna k-esima, occorre fare $n-k$ trasformazioni di Givens, ognuna delle quali richiede $4(n-k)$ prodotti e una estrazione di radice, dunque

$$4\sum_{k=1}^{n-1}(n-k)^2 = 4\sum_{k=1}^{n-1}k^2 = \mathcal{O}\left(4\frac{n^3}{3}\right) \quad \text{prodotti}$$

Fattorizzazione QR

Le rotazioni di Givens sono efficienti per ottenere la fattorizzazione QR di matrici sparse.

Per esempio, per matrici tridiagonali sono sufficienti $n-1$ rotazioni di Givens e si ottiene una R triangolare superiore con solo tre diagonali non nulle:

$$G_{n-1,n}\cdots G_{2,3}G_{1,2}A = R = \begin{pmatrix} * & * & * & & \\ & * & * & * & \\ & & * & * & * \\ & & & * & * \\ & & & & * \end{pmatrix}$$

Si ottiene la fattorizzazione con una complessità pari a $\mathcal{O}(12n)$ prodotti e somme e $\mathcal{O}(n-1)$ radici quadrate.

Risoluzione di un sistema mediante QR

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

Essendo nota la fattorizzazione QR della matrice, si ha

$$A = QR \quad \Rightarrow \quad QR\boldsymbol{x} = \boldsymbol{b}$$
$$R\boldsymbol{x} = Q^T\boldsymbol{b} = \prod_{\substack{i=n-1,\dots,1 \\ j=n,\dots,i+1}} G_{ij}\boldsymbol{b}$$

Il prodotto $Q^T\boldsymbol{b}$, si può ottenere applicando a $\boldsymbol{b}$ le stesse trasformazioni che si applicano ad A.

Esempio

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \qquad \boldsymbol{b} = \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix}$$

Per annullare l'elemento di posizione $(2,1)$ è necessaria una rotazione $G_{1,2}$:

$$\begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

$$-2s + 1c = 0 \;\Rightarrow\; \begin{cases} c = 2/\sqrt{5} \\ s = 1/\sqrt{5} \end{cases}$$

$$G_{1,2}A = \begin{pmatrix} 5/\sqrt{5} & 4/\sqrt{5} & 1/\sqrt{5} \\ 0 & 3/\sqrt{5} & 2/\sqrt{5} \\ 0 & 0 & 2 \end{pmatrix} = R \qquad \boldsymbol{y} = G_{1,2}\boldsymbol{b} = \begin{pmatrix} 10/\sqrt{5} \\ 5/\sqrt{5} \\ 2 \end{pmatrix}$$

Pertanto risolvendo il sistema triangolare $R\boldsymbol{x} = \boldsymbol{y}$, la soluzione vale $\boldsymbol{x}^* = (1,1,1)^T$; la matrice Q è data da

$$Q = G_{1,2}^T = \begin{pmatrix} 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{5} & 2/\sqrt{5} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Osservazioni

Teorema

Sia A una matrice $m \times n$, di rango n ($m \geqslant n$). Allora esistono una e una sola matrice Q_1 di dimensioni $m \times n$ a colonne ortonormali ($Q_1^T Q_1 = I_n$) e una e una sola matrice R_1 triangolare superiore di ordine n a elementi diagonali positivi tale che $A = Q_1 R_1$.

Dim. Poiché $A^T A$ è simmetrica definita positiva (A è di rango n), per il teorema di Cholesky esiste una e una sola matrice R_1 triangolare superiore di ordine n a elementi diagonali positivi tale che

$$A^T A = R_1^T R_1$$

Da $R_1^{-T} A^T A = R_1$, posto $Q_1^T = R_1^{-T} A^T \Rightarrow Q_1 = A R_1^{-1}$, si ha che

- Q_1 è una matrice $m \times n$;
- $Q_1^T Q_1 = R_1^{-T} A^T A R^{-1} = R_1^{-T} R_1^T R_1 R_1^{-1} = I_n$, ossia Q_1 è a colonne ortonormali;
- Q_1 è unica. Infatti, se esistesse Q_2 a colonne ortonormali tale che $A = Q_2 R_2$, allora $R_2 = R_1$ per l'unicità del fattore di Cholesky e $Q_2 = A R_1^{-1} = Q_1$, da cui segue l'unicità di Q_1.

Osservazione

Poichè per il teorema generale di fattorizzazione $A = QR$, scrivendo Q come $\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$ con Q_1 matrice $m \times n$, Q_2 matrice $m \times (m-n)$ e $R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ partizionata in modo coerente, segue che

$$A = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1 = \begin{pmatrix} Q_1 & D_1 \end{pmatrix} \begin{pmatrix} D_1 & R_1 \end{pmatrix}$$

dove D_1 è una matrice diagonale con ${D_1}^2 = I_n$, in cui $\delta_i = 1$ se $r_{ii} > 0$ e $\delta_i = -1$ se $r_{ii} < 0$. Allora la fattorizzazione $Q_1 R_1$, che è unica, è un caso particolare del teorema di fattorizzazione generale.

Codice Matlab

```
function [Q, R] = qrfact(A)
% QRFACT - Fattorizzazione QR con rotazioni di Givens
  [m, n] = size(A);
  r = min(m-1, n);
  Q = eye(m);
  for i = 1 : r
     for j = i+1 : m
        if (A(j,i) ~= 0) % ATTENZIONE: meglio usare una soglia...
           [c, s] = givensrot(A(i,i), A(j,i));
           A([i,j], i:n) = [c s; -s c] * A([i,j], i:n);
           Q(:, [i,j]) = Q(:, [i,j]) * [c -s; s c];
        end
     end
  end
  R = triu(A);
  % elementi diagonali di R non negativi
  for i = 1 : min(m,n)
     if (R(i,i) < 0)
        R(i, i:n) = -R(i, i:n);
        Q(:,i)    = -Q(:,i);
     end
  end
end
```


In Matlab

In Matlab esiste una routine che esegue la fattorizzazione *QR*:

```
>> [Q, R] = qr(A)
```

Esiste anche una routine che esegue la rotazione elementare di Givens:

```
>> [G, y] = planerot(x)
```

dove `x` e `y` sono vettori colonna di due elementi con `x(1)` $\neq 0$ e `G` è $G_{1,2}$, la matrice di rotazione di Givens che annulla `y(2)`.

Esistono altre trasformazioni ortogonali dette **trasformazioni (o riflettori) elementari di Householder**, le quali annullano contemporaneamente tutte le componenti di un vettore da un dato indice in poi (come nel caso dell'eliminazione di Gauss, ma con trasformazioni ortogonali).

Metodi per il calcolo dell'inversa di una matrice

- Se A è fattorizzabile nella forma $A = LR$, occorre risolvere gli n sistemi

$$AX = I_n \quad \Rightarrow \quad LRX = I_n$$
$$LY = I_n \qquad RX = Y$$

Ciò comporta $\mathcal{O}(n^3/3)$ prodotti per la fattorizzazione e $\mathcal{O}(2n^3/3)$ prodotti per la soluzione. Si può anche calcolare A^{-1} mediante l'inversione delle matrici R e L ($2\mathcal{O}(n^3/6)$ prodotti), eseguendo poi il prodotto delle inverse

$$A^{-1} = R^{-1}L^{-1}$$

In totale, in entrambe i casi, $\mathcal{O}(n^3)$ prodotti.

- Se A è fattorizzabile nella forma $PA = LR$, occorre risolvere gli n sistemi

$$PAX = PI_n \Rightarrow LRX = P$$
$$LY = P \qquad RX = Y$$

Ciò comporta $\mathcal{O}(n^3/3)$ prodotti per la fattorizzazione, e $\mathcal{O}(2n^3/3)$ prodotti per la soluzione. Si può anche calcolare A^{-1} mediante l'inversione delle matrici R e L ($2\mathcal{O}(n^3/6)$ prodotti), eseguendo poi il prodotto delle inverse e permutando opportunamente le colonne della matrice ottenuta:

$$A^{-1} = R^{-1}L^{-1}P$$

Metodi per il calcolo dell'inversa di una matrice

- Se A è simmetrica definita positiva e quindi esiste la fattorizzazione di Cholesky, $A = \mathcal{L}\mathcal{L}^T$, occorre risolvere gli n sistemi

$$AX = I_n \quad \Rightarrow \quad \mathcal{L}\mathcal{L}^T X = I_n$$
$$\mathcal{L}Y = I_n \qquad \mathcal{L}^T X = Y$$

Ciò comporta $\mathcal{O}(n^3/6)$ prodotti per la fattorizzazione, e $\mathcal{O}(2n^3/3)$ prodotti per la soluzione. Si può anche calcolare A^{-1} mediante l'inversione della matrice $\mathcal{L}$ ($\mathcal{O}(n^3/6)$ prodotti), eseguendo poi il prodotto della trasposta dell'inversa con l'inversa ($\mathcal{O}(n^3/3)$ prodotti):

$$A^{-1} = \mathcal{L}^{-T}\mathcal{L}^{-1}$$

- Se A è fattorizzabile nella forma $A = QR$, occorre risolvere gli n sistemi

$$AX = I_n \quad \Rightarrow \quad QRX = I_n \quad \Rightarrow \quad RX = Q^T$$

Ciò comporta $4\mathcal{O}(n^3/3)$ prodotti per la fattorizzazione (matrici di Givens), e $\mathcal{O}(n^3/2)$ prodotti per la soluzione. Si può anche calcolare A^{-1} mediante l'inversione della matrice R ($\mathcal{O}(n^3/6)$ prodotti), eseguendo poi il prodotto della trasposta dell'inversa di R con Q^T ($\mathcal{O}(n^3/2)$ prodotti):

$$A^{-1} = R^{-1}Q^T$$

- Metodo di Gauss-Jordan

Sia X l'inversa calcolata con uno qualunque dei metodi precedenti. Allora

$$A^{-1} - X = A^{-1}(I - AX)$$

$$\|A^{-1} - X\| = \|A^{-1}(I - AX)\| \leqslant \|A^{-1}\|\|I - AX\| \quad \Rightarrow \quad \frac{\|A^{-1} - X\|}{\|A^{-1}\|} \leqslant \|I - AX\|$$

Dalla piccolezza di $\|I - AX\|$ si deduce che l'inversa è accettabile (ossia l'errore relativo è piccolo).

Metodo di Gauss-Jordan

Lo scopo del metodo è trovare l'inversa della matrice A, risolvendo il sistema $AX = I_n$ mediante l'applicazioni di trasformazioni elementari di Gauss-Jordan che riducono la matrice A a forma diagonale.
Si dice trasformazione elementare di Gauss–Jordan la seguente matrice:

$$M_j = \begin{pmatrix} 1 & & -m_{1j} & & \\ & 1 & -m_{2j} & & \\ & & \vdots & & \\ & & 1 & & \\ & & -m_{ij} & 1 & \\ & & \vdots & & 1 \end{pmatrix}$$

Se $m_{ij} = \dfrac{a_{ij}}{a_{jj}}$, $i = 1, \ldots, n$; $i \neq j$, la colonna j-esima della matrice A che è premoltiplicata per M_j si annulla ad eccezione dell'elemento diagonale, che rimane invariato.
Occorre che in posizione diagonale, il perno a_{jj} sia un elemento non nullo.

A partire dalla prima colonna, quando si arriva alla colonna j-esima, si cerca in tale colonna, dall'elemento diagonale in poi, l'elemento di modulo massimo (in teoria

Metodo di Gauss-Jordan

ne basterebbe uno diverso da zero, ma numericamente...). Si porta tale elemento in posizione perno mediante una permutazione elementare P_j e poi si annulla la colonna j-esima (eccetto nell'elemento diagonale) mediante una trasformazione di Gauss-Jordan.

Dopo n passi, A è ridotta a forma diagonale e poi all'identità, premoltiplicandola per l'inversa della diagonale ottenuta.

Applicando le trasformazioni anche all'identità si ha:

$$\begin{aligned} \underbrace{D^{-1} M_n P_n M_{n-1} P_{n-1} \dots M_2 P_2 M_1 P_1}_{X} \; [A \,|\, I] &= [I \,|\, V] \\ [A \,|\, I] &= [I \,|\, V] \\ \Rightarrow \qquad XA = I_n \quad \text{e} \quad X &= V \end{aligned}$$

per cui $A^{-1} = X = V$.

La complessità computazionale è pari a $\mathcal{O}(n^3)$ prodotti e somme.

In modo analogo si può trovare la soluzione di un sistema, anche se il metodo di Gauss-Jordan per il calcolo della soluzione di un singolo sistema ha una maggiore complessità ($\mathcal{O}(n^3/2)$ prodotti).

Esempio

$$[A \,|\, I] = \left(\begin{array}{ccc|ccc} 2 & 1 & 1 & 1 & 0 & 0 \\ 4 & 1 & 0 & 0 & 1 & 0 \\ -2 & 2 & 1 & 0 & 0 & 1 \end{array}\right)$$

$\Downarrow$ permutazione 1^a e 2^a riga

$$\left(\begin{array}{ccc|ccc} 4 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 \\ -2 & 2 & 1 & 0 & 0 & 1 \end{array}\right)$$

$$\Rightarrow \quad \begin{array}{c} 1/2 \\ -1/2 \end{array} \left(\begin{array}{ccc|ccc} 4 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1/2 & 1 & 1 & -1/2 & 0 \\ 0 & 5/2 & 1 & 0 & 1/2 & 1 \end{array}\right) \begin{array}{l} 2^a \text{ riga} \leftarrow 2^a \text{ riga} - (1/2) \cdot 1^a \text{ riga} \\ 3^a \text{ riga} \leftarrow 3^a \text{ riga} - (-1/2) \cdot 1^a \text{ riga} \end{array}$$

$\Downarrow$ permutazione 2^a e 3^a riga

$$\left(\begin{array}{ccc|ccc} 4 & 1 & 0 & 0 & 1 & 0 \\ 0 & 5/2 & 1 & 0 & 1/2 & 1 \\ 0 & 1/2 & 1 & 1 & -1/2 & 0 \end{array}\right)$$

$$\Rightarrow \quad \begin{array}{c} 2/5 \\ \\ 1/5 \end{array} \left(\begin{array}{ccc|ccc} 4 & 0 & -2/5 & 0 & 4/5 & -2/5 \\ 0 & 5/2 & 1 & 0 & 1/2 & 1 \\ 0 & 0 & 4/5 & 1 & -3/5 & -1/5 \end{array}\right) \begin{array}{l} 1^a \text{ riga} \leftarrow 1^a \text{ riga} - (2/5) \cdot 2^a \text{ riga} \\ \\ 3^a \text{ riga} \leftarrow 3^a \text{ riga} - (1/5) \cdot 2^a \text{ riga} \end{array}$$

$$\Rightarrow \quad \begin{array}{c} -1/2 \\ 5/4 \end{array} \left(\begin{array}{ccc|ccc} 4 & 0 & 0 & 1/2 & 1/2 & -1/2 \\ 0 & 5/2 & 0 & -5/4 & 5/4 & 5/4 \\ 0 & 0 & 4/5 & 1 & -3/5 & -1/5 \end{array}\right) \begin{array}{l} 1^a \text{ riga} \leftarrow 1^a \text{ riga} - (-1/2) \cdot 3^a \text{ riga} \\ 2^a \text{ riga} \leftarrow 2^a \text{ riga} - (5/4) \cdot 3^a \text{ riga} \end{array}$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1/8 & 1/8 & -1/8 \\ 0 & 1 & 0 & -1/2 & 1/2 & 1/2 \\ 0 & 0 & 1 & 5/4 & -3/4 & -1/4 \end{array}\right) = \left[I \,\middle|\, A^{-1} \right]$$

Codice Matlab

```
function [ X ] = gaussJordan(A)
% GaussJordan - Algoritmo di Gauss-Jordan per il calcolo dell'
    inversa
  n = size(A, 1);
  temp = zeros(1, 2*n);
  A = [A eye(n)];
  for k = 1 : n
     [amax, ind] = max( abs(A(k:n, k)) );
     ind = ind + k - 1;
     if (k ~= ind)
        temp      = A(ind, :);
        A(ind, :) = A(k, :);
        A(k, :)   = temp;
     end
     A( [1:(k-1), (k+1):n], k ) = A( [1:(k-1), (k+1):n], k ) / A(k,k
         );
     %  operazione di base: aggiornamento mediante diadi
     A( [1:(k-1), (k+1):n], (k+1):(2*n) ) = ...
              A( [1:(k-1), (k+1):n], (k+1):(2*n) ) - ...
              A( [1:(k-1), (k+1):n], k ) * A(k, (k+1):(2*n));
  end
  X = diag( 1./diag(A(:, 1:n)) ) * A(:, (n+1):(2*n));
end
```


Complessità computazionale per sistemi densi di n equazioni

metodo	prodotti	somme	radici quadrate
Gauss	$n^3/3$	$n^3/3$	
Cholesky	$n^3/6$	$n^3/6$	n
Gauss–Jordan	$n^3/2$	$n^3/2$	
Givens	$4n^3/3$	$2n^3/3$	$n^2/2$

Un semplice esempio:

$$\begin{cases} x_1 + 2x_2 = 3 \\ 0.499x_1 + 1.001x_2 = 1.5 \end{cases}$$

La soluzione è $\boldsymbol{x}^* = (1,1)^T$. Perturbando i dati la soluzione cambia:

matrice dei coefficienti

$$\begin{cases} x_1 + 2x_2 = 3 \\ 0.5x_1 + 1.002x_2 = 1.5 \end{cases}$$

$$\Downarrow$$

$$\tilde{\boldsymbol{x}}^* = (3,0)^T$$

termine noto

$$\begin{cases} x_1 + 2x_2 = 3 \\ 0.499x_1 + 1.001x_2 = 1.4985 \end{cases}$$

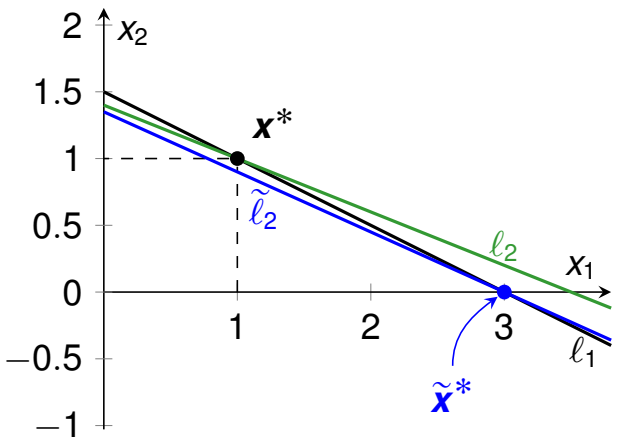
$$\Downarrow$$

$$\overline{\boldsymbol{x}}^* = (2,0.5)^T$$

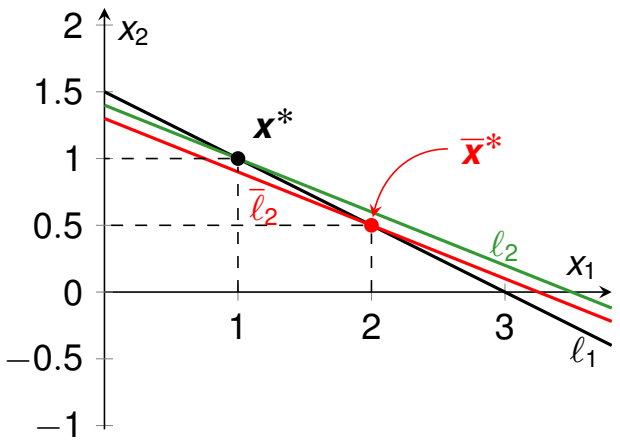
Perturbando di poco i dati iniziali, si trovano soluzioni diverse: si tratta di un **problema mal condizionato**.

Geometricamente, si tratta delle equazioni di due rette ℓ_1, ℓ_2 quasi parallele, di cui si vuole trovare l'intersezione. Perturbando ℓ_2 di poco si ottengono altre due rette $\tilde{\ell}_2$ e $\overline{\ell}_2$ e altri punti di intersezione, $\tilde{\boldsymbol{x}}^* = (3,0)^T$ e $\overline{\boldsymbol{x}}^* = (2,0.5)^T$, che distano poco da punti di ℓ_2 e appartengono a ℓ_1.

perturbazione della matrice



perturbazione del termine noto



N.B.: nei disegni le perturbazioni sono molto amplificate.

Residuo e accuratezza

Il vettore residuo $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{w}^*$ (dove $\boldsymbol{w}^*$ è la soluzione di una equazione perturbata e A e $\boldsymbol{b}$ sono matrice e termine noto del sistema non perturbato) è piccolo in entrambi i casi, pur essendo $\boldsymbol{w}$ significativamente diverso dalla soluzione esatta $\boldsymbol{x}^*$:

$$\boldsymbol{w}^* = (3, 0)^T \qquad \boldsymbol{w}^* = (2, 0.5)^T$$
$$\boldsymbol{r} = \begin{pmatrix} 0 \\ 0.003 \end{pmatrix} \qquad \boldsymbol{r} = \begin{pmatrix} 0 \\ 0.0015 \end{pmatrix}$$

In pratica, invece di risolvere

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

si risolve

$$(A + \Delta A)\boldsymbol{w}_1 = \boldsymbol{b} \qquad A\boldsymbol{w}_2 = (\boldsymbol{b} + \Delta \boldsymbol{b})$$

È davvero possibile che ci siano queste perturbazioni sui dati iniziali?

Possiamo pensare che ΔA e $\Delta \boldsymbol{b}$ siano perturbazioni dei dati dovuti all'approssimazione dei numeri con i numeri di macchina e che $\boldsymbol{w}_1^*$ e $\boldsymbol{w}_2^*$ siano le soluzioni calcolate in aritmetica esatta a partire da dati perturbati.

Pertanto, a causa degli errori di rappresentazione dei dati del problema e degli errori di arrotondamento nelle operazioni, un qualunque metodo numerico su calcolatore determina una soluzione approssimata $\boldsymbol{w}^*$ invece della soluzione esatta $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$.

Come è possibile valutare l'errore $\boldsymbol{e} = \boldsymbol{x}^* - \boldsymbol{w}^*$, visto che $\boldsymbol{x}^*$ non è noto?
Un criterio che si utilizza per decidere se $\boldsymbol{w}^*$ è una approssimazione accettabile consiste nel richiedere che la norma del residuo sia piccola.

$$\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{w}^*$$

Se $\|\boldsymbol{r}\| = 0 \Rightarrow \|\boldsymbol{b} - A\boldsymbol{w}^*\| = 0 \Rightarrow \boldsymbol{w}^* \equiv \boldsymbol{x}^*$.
Tale criterio non è sempre valido.

È davvero possibile che ci siano queste perturbazioni sui dati iniziali?

Infatti, dalla definizione di residuo $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{w}^*$, si ha

$$A\boldsymbol{w}^* = \boldsymbol{b} - \boldsymbol{r}$$

$\boldsymbol{w}^*$ è soluzione esatta di un sistema in cui il termine noto si può ritenere perturbato di una quantità pari a $-\boldsymbol{r}$.
Anche se $\boldsymbol{r}$ ha elementi piccoli, se il problema è mal condizionato, $\boldsymbol{w}^*$ può essere molto diverso da $\boldsymbol{x}^*$.
Si osservi che:

$$\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{w}^* = A\boldsymbol{x}^* - A\boldsymbol{w}^* = A(\boldsymbol{x}^* - \boldsymbol{w}^*) = A\boldsymbol{e}$$

L'**errore assoluto $\boldsymbol{e}$** è soluzione del sistema $\boldsymbol{r} = A\boldsymbol{x}$. Pertanto

$$\boldsymbol{e} = \boldsymbol{x}^* - \boldsymbol{w}^* = A^{-1}\boldsymbol{r} \quad \Rightarrow \quad \|\boldsymbol{x}^* - \boldsymbol{w}^*\| = \|A^{-1}\boldsymbol{r}\| \leqslant \|A^{-1}\|\|\boldsymbol{r}\|$$

Inoltre da

$$A\boldsymbol{x}^* = \boldsymbol{b} \quad \Rightarrow \quad \|\boldsymbol{b}\| \leqslant \|A\|\|\boldsymbol{x}^*\| \quad \Rightarrow \quad \frac{\|\boldsymbol{b}\|}{\|A\|} \leqslant \|\boldsymbol{x}^*\| \quad \Rightarrow \quad \frac{1}{\|\boldsymbol{x}^*\|} \leqslant \frac{\|A\|}{\|\boldsymbol{b}\|}$$

si ha che

$$\frac{\|\boldsymbol{x}^* - \boldsymbol{w}^*\|}{\|\boldsymbol{x}^*\|} \leqslant \|A\|\|A^{-1}\|\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}$$

È davvero possibile che ci siano queste perturbazioni sui dati iniziali?

Pertanto dalla piccolezza del residuo non si può dedurre che l'errore assoluto o l'errore relativo siano piccoli, poichè le quantità $\|A^{-1}\|$ oppure $\|A\|\|A^{-1}\|$ forniscono la connessione tra residuo e accuratezza.

Solo se $\|A^{-1}\|$ oppure $\|A\|\|A^{-1}\|$ sono piccole, si può accettare $\boldsymbol{w}^*$ come soluzione di $A\boldsymbol{x} = \boldsymbol{b}$ se $\|\boldsymbol{r}\|$ è piccola.

Usualmente (con un po' di abuso di nomenclatura), si chiama residuo normalizzato in una data norma $\|\cdot\|_*$ il *vettore* $\boldsymbol{r}/\|\boldsymbol{b}\|_*$, mentre la norma del residuo normalizzato è la quantità *scalare* $\left\|\boldsymbol{r}/\|\boldsymbol{b}\|_*\right\|_* = \|\boldsymbol{r}\|_*/\|\boldsymbol{b}\|_*$.

Esempi

Negli esempi precedenti, con $A = \begin{pmatrix} 1 & 2 \\ 0.499 & 1.001 \end{pmatrix}$, $\|\boldsymbol{b}\|_\infty = 3$ e $\|\boldsymbol{x}^*\|_\infty = 1$, si ha:

$$A^{-1} = \begin{pmatrix} 1.001 & -2 \\ -0.499 & 1 \end{pmatrix} \frac{1}{1.001 - .998}$$

$$\|A\|_\infty = 3 \qquad \|A^{-1}\|_\infty = \frac{3.001}{0.003} = 1000.333$$

$$\|A\|_\infty \|A^{-1}\|_\infty \approx 3001$$

$\boldsymbol{w}^* = (3,0)^T$
$\boldsymbol{r} = \begin{pmatrix} 0 \\ 0.003 \end{pmatrix}$
$\|\boldsymbol{x}^* - \boldsymbol{w}^*\|_\infty = 2$
$\|\boldsymbol{x}^* - \boldsymbol{w}^*\|_\infty / \|\boldsymbol{x}^*\|_\infty = 2$
$\|\boldsymbol{r}\|_\infty = 3 \cdot 10^{-3}$
$\|A^{-1}\|_\infty \|\boldsymbol{r}\|_\infty \approx 3$
$\|A\|_\infty \|A^{-1}\|_\infty \|\boldsymbol{r}\|_\infty / \|\boldsymbol{b}\|_\infty \approx 3$

$\boldsymbol{w}^* = (2,0.5)^T$
$\boldsymbol{r} = \begin{pmatrix} 0 \\ 0.0015 \end{pmatrix}$
$\|\boldsymbol{x}^* - \boldsymbol{w}^*\|_\infty = 1$
$\|\boldsymbol{x}^* - \boldsymbol{w}^*\|_\infty / \|\boldsymbol{x}^*\|_\infty = 1$
$\|\boldsymbol{r}\|_\infty = 1.5 \cdot 10^{-3}$
$\|A^{-1}\|_\infty \|\boldsymbol{r}\|_\infty \approx 1.5$
$\|A\|_\infty \|A^{-1}\|_\infty \|\boldsymbol{r}\|_\infty / \|\boldsymbol{b}\|_\infty \approx 1.5$

Come si vede, il residuo è piccolo ma l'errore è grande: questa è una caratteristica dei sistemi mal condizionati.

Attenzione

Si osservi anche che se si hanno due soluzioni approssimate $\boldsymbol{w}^{(1)}$ e $\boldsymbol{w}^{(2)}$ con residui $\boldsymbol{r}^{(1)}$ e $\boldsymbol{r}^{(2)}$, rispettivamente, non è vero che se $\|\boldsymbol{r}^{(1)}\| < \|\boldsymbol{r}^{(2)}\|$, la prima soluzione sia più accurata della seconda. Infatti,

$\boldsymbol{w}^{(1)} = (3,0)^T$
$\boldsymbol{r}^{(1)} = \begin{pmatrix} 0 \\ 0.003 \end{pmatrix}$
$\|\boldsymbol{r}^{(1)}\|_\infty = 0.003$ $<$

$\boldsymbol{w}^{(2)} = (0.4, 1.302)^T$
$\boldsymbol{r}^{(2)} = \begin{pmatrix} -0.004 \\ -0.002902 \end{pmatrix}$
$\|\boldsymbol{r}^{(2)}\|_\infty = 0.004$

Ma

$$\|\boldsymbol{x}^* - \boldsymbol{w}^{(1)}\|_\infty = 2 \quad > \quad \|\boldsymbol{x}^* - \boldsymbol{w}^{(2)}\|_\infty = 0.6$$

Un teorema generale

Teorema (fondamentale di perturb. della soluzione di un sistema lineare)

Sia $A \in \mathbb{R}^{n\times n}$ non singolare. Dato il sistema $A\boldsymbol{x} = \boldsymbol{b}$, nell'ipotesi di avere sulla matrice A una perturbazione ΔA tale che $\|\Delta A\|\|A^{-1}\| < 1$, dove $\|\cdot\|$ è una norma naturale e $A + \Delta A$ è non singolare, sia $\boldsymbol{w}^*$ la soluzione del sistema perturbato

$$(A + \Delta A)\boldsymbol{w} = \boldsymbol{b} + \Delta \boldsymbol{b}$$

Allora

$$\frac{\|\boldsymbol{x}^* - \boldsymbol{w}^*\|}{\|\boldsymbol{x}^*\|} \leqslant \frac{\mu(A)}{1 - \mu(A)\dfrac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|} \right)$$

dove $\mu(A) = \|A\|\|A^{-1}\|$ si dice **numero di condizione di** A.

In particolare, se $\Delta A = 0$ e $\Delta \boldsymbol{b} = -\boldsymbol{r}$ (come in $A\boldsymbol{w}^* = \boldsymbol{b} - \boldsymbol{r}$),

$$\frac{\|\boldsymbol{x}^* - \boldsymbol{w}^*\|}{\|\boldsymbol{x}^*\|} \leqslant \mu(A)\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}$$

Se $\Delta \boldsymbol{b} = \boldsymbol{0}$,

$$\frac{\|\boldsymbol{x}^* - \boldsymbol{w}^*\|}{\|\boldsymbol{x}^*\|} \leqslant \frac{\mu(A)}{1 - \mu(A)\dfrac{\|\Delta A\|}{\|A\|}} \frac{\|\Delta A\|}{\|A\|}$$

Proprietà del numero di condizione di una matrice

In una norma naturale, vale che $\mu(A) \geqslant 1$.
Infatti

$$1 = \|I_n\| = \|AA^{-1}\| \leqslant \|A\|\|A^{-1}\| = \mu(A)$$

Se $\mu(A) \gg 1$, A è mal condizionata.
Se $\mu(A) \approx 1$, A è ben condizionata.
$\mu(I_n) = 1$.
Se gli elementi di A sono normalizzati in modo che $\|A\| = 1$, un valore di $\mu(A)$ grande si riflette nell'enorme crescita di A^{-1}:

$$A = \frac{1}{2+\epsilon}\begin{pmatrix} 1 & 1 \\ 1 & 1+\epsilon \end{pmatrix} \qquad \|A\|_\infty = 1$$

$$A^{-1} = \frac{2+\epsilon}{\epsilon}\begin{pmatrix} 1+\epsilon & -1 \\ -1 & 1 \end{pmatrix} \qquad \|A^{-1}\|_\infty = \frac{(2+\epsilon)^2}{\epsilon} = \mu_\infty(A)$$

Se $\epsilon = 10^{-k} \Rightarrow \mu_\infty(A) \approx 10^k$.

Il numero di condizionamento di una matrice esprime quanto una matrice è "vicina" alla singolarità.

Numero di condizione (o di condizionamento)

La definizione di numero di condizione come

$$\mu(A) = \|A\|\|A^{-1}\|$$

è data per A non singolare. Si può estendere la definizione al caso di matrici qualunque.

Definizione

Sia $A \in \mathbb{R}^{m\times n}$. Si definisce numero di condizione (o di condizionamento) di A rispetto a una norma naturale $\|\cdot\|$, indotta da una norma vettoriale $\|\cdot\|_V$, il valore

$$\mu(A) = \frac{\max \|A\boldsymbol{x}\|_V}{\min \|A\boldsymbol{x}\|_V}$$

dove il minimo è fatto sui vettori $\|A\boldsymbol{x}\|_V \neq 0$.

Il numero di condizione è il rapporto tra la perturbazione massima e la pertubazione minima non nulla che ogni vettore $\boldsymbol{x} \in \mathbb{R}^n$ subisce per effetto della trasformazione lineare associata ad A.

Numero di condizione (o di condizionamento)

Nel caso della norma euclidea, per definire il numero di condizione, introduciamo la definizione di valori singolari di A.

Definizione

Sia $A \in \mathbb{R}^{m\times n}$. Si dicono valori singolari di A le radici quadrate degli autovalori non nulli di A^TA:

$$\sigma_i = \sqrt{\lambda_i(A^TA)} \neq 0$$

Allora, rispetto alla norma euclidea, il numero di condizione si definisce come:

$$\mu_2(A) = \frac{\sqrt{\lambda_{\max}(A^TA)}}{\sqrt{\lambda_{\min}(A^TA)}} = \frac{\sigma_{\max}}{\sigma_{min}} \qquad \text{dove} \quad \lambda_{\min}(A^TA) = \min_{\lambda\neq 0} \lambda(A^TA).$$

Se $A \in \mathbb{R}^{n\times n}$ è simmetrica, $\mu_2(A) = \dfrac{\sqrt{\lambda_{\max}(A^2)}}{\sqrt{\lambda_{\min}(A^2)}} = \dfrac{|\lambda(A)|_{\max}}{|\lambda(A)|_{\min}}$.

Segue che:

- A normalizzata ($\|A\|_2 = 1$) è mal condizionata se e solo se ha (almeno) un valore singolare “piccolo”.
- A simmetrica normalizzata è mal condizionata se e solo se ha (almeno) un autovalore di modulo “piccolo”.

Una disuguaglianza tra autovalori e valori singolari

Sia $A \in \mathbb{R}^{n\times n}$. Vale che

$$\sigma_{\min}^2(A) \leqslant |\lambda_i(A)|^2 \leqslant \sigma_{\max}^2(A)$$

Si ha che:

- Se A ha un piccolo autovalore in modulo è mal condizionata.
- Se A è mal condizionata non è detto che abbia un piccolo autovalore in modulo.

In genere una matrice è mal condizionata se è vicina alla singolarità, ma non esiste relazione tra condizionamento e valore del determinante.
Se A ha determinante piccolo non è detto che sia mal condizionata.

$$A = \mathrm{diag}(1/2, 1/2, \ldots, 1/2) \qquad \det(A) = \frac{1}{2^n} \qquad \mu_2(A) = \frac{|\lambda|_{\max}}{|\lambda|_{\min}} = 1$$

Se A è mal condizionata non è detto che abbia determinante piccolo:

$$T = \begin{pmatrix} 1 & -1 & \ldots & -1 & -1 \\ & 1 & \ldots & -1 & -1 \\ & & \ddots & \vdots & \vdots \\ & & & 1 & -1 \\ & & & & 1 \end{pmatrix} \qquad \det(T) = 1 \quad \|T\|_\infty = n$$

Una disuguaglianza tra autovalori e valori singolari

$$T^{-1} = \begin{pmatrix} 1 & 2^0 & 2^1 & \ldots & 2^{n-2} \\ & 1 & 2^0 & \ldots & 2^{n-3} \\ & & 1 & 2^0 & \vdots \\ & & & 1 & 2^0 \\ & & & & 1 \end{pmatrix}$$

$$\|T^{-1}\|_\infty = 2^{n-1} \qquad \mu_\infty(T) = n2^{n-1}$$

Esempio

Matrici di Hilbert: $H_n = [h_{ij}]$. Sono matrici simmetriche definite positive.

$$h_{ij} = \frac{1}{i+j-1} \qquad i,j = 1,\ldots,n$$

$$H_4 = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{pmatrix}$$

In aritmetica finita, molti elementi vengono perturbati. In particolare, con $\beta = 10$ e $t = 5$,

$$\widetilde{H}_4 = \begin{pmatrix} 1.00000 & 0.50000 & 0.33333 & 0.25000 \\ 0.50000 & 0.33333 & 0.25000 & 0.20000 \\ 0.33333 & 0.25000 & 0.20000 & 0.16666 \\ 0.25000 & 0.20000 & 0.16666 & 0.14285 \end{pmatrix}$$

Esempio

H_n^{-1} ha elementi

$$\overline{h}_{ij} = \frac{(-1)^{i+j}(n+i-1)!(n+j-1)!}{(i+j-1)!((i-1)!(j-1)!)^2(n-j)!(n-i)!}$$

$$H_4^{-1} = \begin{pmatrix} 16 & -120 & 240 & -140 \\ -120 & 1200 & -2700 & 1680 \\ 240 & -2700 & 6480 & -4200 \\ -140 & 1680 & -4200 & 2800 \end{pmatrix}$$

$$\|H_4\|_\infty \|H_4^{-1}\|_\infty = \frac{25}{12} \cdot 13620 = 28375 \approx 2.8 \cdot 10^4$$

n	$\mu_2(H_n)$	$\mu_\infty(H_n)$
2	$1.505 \cdot 1$	$2.700 \cdot 10$
3	$5.241 \cdot 10^2$	$7.480 \cdot 10^2$
4	$1.551 \cdot 10^4$	$2.837 \cdot 10^4$
5	$4.766 \cdot 10^5$	$9.436 \cdot 10^5$
6	$1.495 \cdot 10^7$	$2.907 \cdot 10^7$
7	$4.754 \cdot 10^8$	$9.852 \cdot 10^8$
8	$1.526 \cdot 10^{10}$	$3.387 \cdot 10^{10}$
9	$4.932 \cdot 10^{11}$	$1.099 \cdot 10^{12}$
10	$1.603 \cdot 10^{13}$	$3.535 \cdot 10^{13}$

Codice Matlab

```
% testHilbert - Risolve sistemi con matrice di Hilbert
for n = 5 : 14
   fprintf('*********************************\n');
   fprintf(' Matrice di Hilbert di ordine %2d\n',n);
   A = hilb(n);
   b = A*ones(n,1);
   [L, R, deter] = gauss1(A);
   y = ltrisol(L, b);
   x = rtrisol(R, y);
   fprintf('Soluzione:\n');
   fprintf('%2.16e \n',x);
   pause
   r = b - A*x;
   muA = norm(A,inf) * norm( gj(A), inf);
   fprintf('norma del residuo = %2.9e \n', norm(r, inf));
   fprintf('mu(A) = %g \n', muA);
   fprintf('errore rel = %2.9e  ', norm(x - ones(n,1), inf));
   fprintf('valore magg = %2.9e \n', muA*norm(r, inf)/norm(b,inf));
   pause
end
```

`gj(A)` calcola l'inversa di A con l'algoritmo di Gauss-Jordan; in Matlab si può usare `inv(A)`.

Stima del numero di condizione

Il calcolo di $\mu(A)$ implica la valutazione di A^{-1}. Ma calcolare A^{-1} vuol dire risolvere $A\alpha_j = \boldsymbol{e}_j$, $j = 1, \ldots, n$, con il costo di risoluzione di n sistemi e gli stessi problemi connessi alla soluzione di $A\boldsymbol{x} = \boldsymbol{b}$ in aritmetica finita.
Allora si usa una stima di $\|A^{-1}\|$ ottenuta nel seguente modo:

- si calcola una soluzione approssimata $\boldsymbol{w}^*$;
- si calcola il residuo in doppia precisione (se possibile) $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{w}^* = A(\boldsymbol{x} - \boldsymbol{w}^*)$;

Pertanto $\boldsymbol{e}^* = \boldsymbol{x}^* - \boldsymbol{w}^*$ è soluzione di

$$A\boldsymbol{e} = \boldsymbol{r}$$

La soluzione calcolata è una approssimazione di $\boldsymbol{e}^* = A^{-1}\boldsymbol{r}$:

$$\boldsymbol{e}^* = A^{-1}\boldsymbol{r}$$

$$\|\boldsymbol{e}^*\| \approx \|\boldsymbol{x}^* - \boldsymbol{w}^*\| \leqslant \|A^{-1}\|\|\boldsymbol{r}\|$$

Pertanto $\dfrac{\|\boldsymbol{e}^*\|}{\|\boldsymbol{r}\|}$ è una sottostima di $\|A^{-1}\|$.

Stima del numero di condizione

Si può anche provare che

$$\|\boldsymbol{r}\| \approx 10^{-t}\|A\|\|\boldsymbol{w}^*\|$$

dove t è il numero di cifre dell'aritmetica usata.

$$\|\boldsymbol{e}^*\| \approx \|\boldsymbol{x}^* - \boldsymbol{w}^*\| \leqslant \|A^{-1}\|\|\boldsymbol{r}\| \approx \|A^{-1}\|\|A\|10^{-t}\|\boldsymbol{w}^*\|$$

da cui discende

$$\mu(A) \approx 10^t \frac{\|\boldsymbol{e}^*\|}{\|\boldsymbol{w}^*\|}$$

È opportuno calcolare $\boldsymbol{r}$ in doppia precisione.

Esempio

$$\begin{cases} x_1 + 2\,x_2 = 3 \\ 0.499\,x_1 + 1.001\,x_2 = 1.5 \end{cases}$$

Si assume $t = 3$:

$$\boldsymbol{w}^* = (0, 1.5)^T$$

$$\boldsymbol{r} = \begin{pmatrix} 3 - 0 - 2 \cdot 1.5 \\ 1.5 - 0.499 \cdot 0 - 1.001 \cdot 1.5 \end{pmatrix} = \begin{pmatrix} 0 \\ -1.5 \cdot 10^{-3} \end{pmatrix}$$

$$\|\boldsymbol{r}\|_\infty = 1.5 \cdot 10^{-3}$$

Si risolve $A\boldsymbol{e} = \boldsymbol{r} \;\Rightarrow\; \boldsymbol{e}^* = \begin{pmatrix} 1.5 \\ -0.75 \end{pmatrix}$, $\|\boldsymbol{e}^*\|_\infty = 1.5$.

Si osservi che $\dfrac{\|\boldsymbol{e}^*\|_\infty}{\|\boldsymbol{r}\|_\infty} = 10^3$ che è una sottostima di $\|A^{-1}\|_\infty = 1000.333$.

Inoltre $\|\boldsymbol{r}\|_\infty \approx 10^{-3}\|A\|_\infty\|\boldsymbol{w}^*\|_\infty = 10^{-3} \cdot 3 \cdot 1.5 = 4.5 \cdot 10^{-3}$. Da ciò discende

$$3001 \approx \mu_\infty(A) \approx 10^3 \frac{\|\boldsymbol{e}^*\|_\infty}{\|\boldsymbol{w}^*\|_\infty} = \frac{1.5 \cdot 10^3}{1.5} = 1000.$$

Codice Matlab

```
% testCondHilbert - Test numero di condizione matrici di Hilbert
n = 10;
% problema test
A = hilb(n);
b = A * ones(n, 1);
% fattorizzazione;
[L, R, deter] = gauss1(A);
y = ltrisol(L, b);
w = rtrisol(R, y);
fprintf('Soluzione:\n');
fprintf('%2.16e \n',w);
% stima dell'errore
r = b - A*w;
z = ltrisol(L, r);
e = rtrisol(R, z);
fprintf('Stima (sottostima) della norma dell''inversa: %g\n',...
  norm(e,inf)/norm(r,inf));
fprintf('Stima (sottostima) di mu(A): %g\n',...
  norm(e,inf)/norm(r,inf)*norm(A,inf));
fprintf('Stima del numero di condizione con precis. 16: %g\n', ...
   1e16*norm(e,inf)/norm(w,inf));
```


Codice Matlab

Output:

```
>> testCondHilbert
Soluzione:
9.9999999875483425e-01
1.0000001067849724e+00
9.9999773786147539e-01
1.0000204794185161e+00
9.9990264184733946e-01
1.0002669070133352e+00
9.9956308847027175e-01
1.0004214011575985e+00
9.9977914079621188e-01
1.0000484987218523e+00
Stima (sottostima) della norma dell'inversa: 1.13743e+12
Stima (sottostima) di mu(A) = 3.3315e+12
Stima del numero di condizione con precisione 16 = 5.04908e+12
>> cond(A)
ans =
   1.6025e+013
>>
```

Stabilità dei metodi diretti

In aritmetica finita, i fattori di A o di PA sono affetti da errore. Detti $\mathcal{L}$ e $\mathcal{R}$ i fattori calcolati, essi possono essere ritenuti fattori esatti di una matrice perturbata mediante una matrice δA:

$$PA + \delta A = \mathcal{L}\mathcal{R} = (L + \delta L)(R + \delta R) = LR + L(\delta R) + (\delta L)R + (\delta L)(\delta R)$$
$$\Rightarrow \quad \delta A = L(\delta R) + (\delta L)R + (\delta L)(\delta R)$$

δA è piccolo (e dunque $\mathcal{L}$ e $\mathcal{R}$ sono accettabili) **se** gli elementi di L e di R non sono troppo grandi rispetto a quelli di A. Si cercano algoritmi che mantengano L e R limitati. Tali algoritmi si dicono **stabili**.
Sia A fattorizzabile e normalizzata in modo che $\max_{i,j} |a_{i,j}| = 1$:

$$A = LR \qquad (PA = LR)$$

Se esistono costanti positive a e b indipendenti dagli elementi di A e dall'ordine di A, tali che

$$|\ell_{ij}| \leqslant a \qquad |r_{ij}| \leqslant b$$

la fattorizzazione LR si dice **stabile in senso forte**.
Se a e b dipendono dall'ordine di A, allora la fattorizzazione di A si dice **stabile in senso debole.**

Stabilità per le fattorizzazioni

- Sia A simmetrica definita positiva:

$$A = LL^T \;\Rightarrow\; 0 < a_{ii} = \sum_{j=1}^{i} \ell_{ij}^2 \;\Rightarrow\; \ell_{ij}^2 \leqslant a_{ii} \leqslant \max_{r,s} |a_{rs}| \;\Rightarrow\; |\ell_{ij}| \leqslant \sqrt{\max_{r,s} |a_{rs}|}$$

$$\frac{|\ell_{ij}|}{\sqrt{\max_{r,s} |a_{rs}|}} \leqslant 1$$

$$\frac{1}{\max_{r,s} |a_{rs}|} A = \frac{1}{\sqrt{\max_{r,s} |a_{rs}|}} L \cdot \frac{1}{\sqrt{\max_{r,s} |a_{rs}|}} L^T = HH^T$$

dove $H = \dfrac{1}{\sqrt{\max_{r,s} |a_{rs}|}} L.$

$$|h_{ij}| = \frac{|\ell_{ij}|}{\sqrt{\max_{r,s} |a_{rs}|}} \leqslant 1$$

Poiché $a = b = 1$, **la fattorizzazione di Cholesky è stabile in senso forte.**

Stabilità per le fattorizzazioni

- **Algoritmo di eliminazione di Gauss con pivoting parziale:** si dimostra che

$$|\ell_{ij}| \leqslant 1 \qquad |r_{ij}| \leqslant 2^{n-1} \max_{r,s} |a_{rs}|$$

Infatti, per la scelta del perno come elemento di modulo massimo sulla colonna k-esima a partire dalla posizione diagonale k, i moltiplicatori sono in modulo minori o uguali a 1 ($|m_{ij}| \leqslant 1 \Rightarrow |\ell_{ij}| \leqslant 1 \Rightarrow a = 1$).

$$\begin{aligned} a_{ij}^{(2)} = a_{ij} - m_{i1} a_{1j} \Rightarrow |a_{ij}^{(2)}| &\leqslant 2 \max_{r,s} |a_{rs}| \\ a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2} a_{2j}^{(2)} \Rightarrow |a_{ij}^{(3)}| &\leqslant 2 \max_{r,s} |a_{rs}^{(2)}| \\ &\leqslant 2^2 \max_{r,s} |a_{rs}| \\ &\vdots \\ a_{ij}^{(n)} = a_{ij}^{(n-1)} - m_{i,n-1} a_{n-1,j}^{(n-1)} \Rightarrow |a_{ij}^{(n)}| &\leqslant 2^{n-1} \max_{r,s} |a_{rs}| \end{aligned}$$

Stabilità per le fattorizzazioni

Per cui $b = 2^{n-1}$. Esistono matrici (esempio di Wilkinson) per le quali tale limite è raggiunto:

$$a_{ij} = \begin{cases} 1 & \text{se } i = j \\ -1 & \text{se } j < i \\ 1 & \text{se } j = n \\ 0 & \text{altrove} \end{cases} \qquad r_{ij} = \begin{cases} 1 & \text{se } i = j,\ j \neq n \\ 2^{i-1} & \text{se } j = n \\ 0 & \text{altrove} \end{cases}$$

L coincide con il triangolo inferiore di A e $r_{nn} = 2^{n-1}$.
Se r_{nn} viene alterato, $\widetilde{r}_{nn} = 2^{n-1} - \epsilon$, $L\widetilde{R}$ è la fattorizzazione esatta di una matrice che differisce dalla A solo in $\widetilde{a}_{nn} = 1 - \epsilon$. Se $\epsilon = 0.5$, $\widetilde{R}$ è circa uguale a R. Ma in tal modo $\widetilde{R}$ diventa fattore di una matrice molto perturbata rispetto ad A.

Stabilità per le fattorizzazioni

```
% testWilkinson - Esempio di Wilkinson
n = 50;
A = -ones(n);
A = tril(A,-1) + diag(ones(n,1));
A(:, end) = ones(n,1);
[L, R, P, deter] = gauss2(A);
fprintf('R(n,n) = %17.16e\n', R(n,n));
Rbar = R;
Rbar(n,n) = R(n,n) - 0.05;
err = 0.05/R(n,n);
fprintf('Rbar(n,n) = %17.16e; errore relativo ris.: %g\n', ...
         Rbar(n,n), err);
Abar = L*Rbar;
Abar = Abar(P,:);
erri = (A(n,n) - Abar(n,n)) / A(n,n);
fprintf('Abar(n,n) = %12.9e; errore relativo input: %g\n', ...
         Abar(n,n), erri);
```

Output

```
>> testWilkinson
R(n,n) = 5.6294995342131200e+14
Rbar(n,n) = 5.6294995342131194e+14; errore relativo ris.: 8.88178e-17
Abar(n,n) = 9.375000000e-01; errore relativo input: 0.0625
```


Stabilità per le fattorizzazioni

Nel caso di $n = 100$, se si effettua la fattorizzazione di Gauss della matrice di Wilkinson su Matlab si ottiene che `R(n,n) = 6.338253001141147e+029`. Eseguendo $L * R$, la matrice prodotto ha nella componente $(100, 100)$ il valore 0.

- Nel caso di **pivoting totale**, si dimostra che:

$$|\ell_{ij}| \leqslant 1 \qquad |r_{ij}| \leqslant f(n) \max_{r,s} |a_{rs}|$$

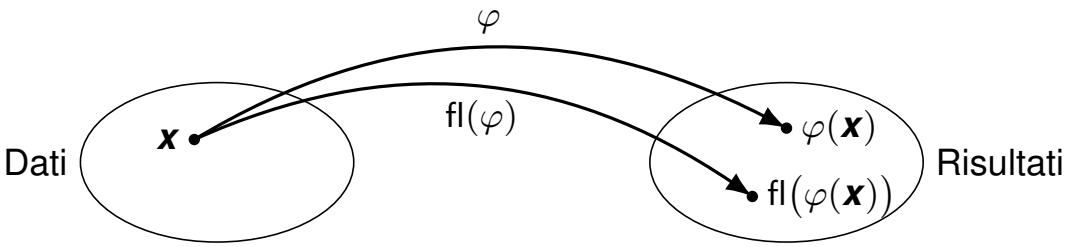
$$f(n) = \sqrt{n}\sqrt{1 \cdot 2 \cdot 3^{1/2} \cdot 4^{1/3} \cdots n^{1/(n-1)}}$$

Non si conoscono matrici per cui vale l'uguaglianza. Per $n \leqslant 4$, si dimostra che $f(n) = n$.

n	$f(n)$ (Gauss piv. tot.)	2^{n-1} (Gauss piv. parz.)
10	19	2^9
20	67	2^{19}
50	530	2^{49}
100	3300	2^{99}

Stabilità per le fattorizzazioni

stabilità : $\begin{cases} \text{Dati esatti} \\ \text{Operazioni con errori} \end{cases}$



- Nel caso di matrici di Hessemberg, l'uso del pivoting parziale produce $a = 1$ e $b = n$
- Nel caso di matrici tridiagonali e matrici diagonali dominanti, $a = 1$, $b = 2$.

Stabilità per le fattorizzazioni

- Per la fattorizzazione ortogonale $A = QR$, vale che:

$$\max_{i,j} |q_{ij}| = \frac{1}{n}\|Q\|_T \leqslant \|Q\|_2 = 1$$

Da $Q^T A = R$, segue per ogni j:

$$\begin{aligned} \max_i |r_{ij}| = \|r_{*,j}\|_\infty \leqslant \|r_{*.j}\|_2 = \|Q^T a_{*,j}\|_2 \\ \leqslant \|Q^T\|_2 \sqrt{n}\, \|a_{*j}\|_\infty = \sqrt{n}\, \max_i |a_{ij}| \end{aligned}$$

Pertanto per una matrice normalizzata si ha che

$$\max_{i,j} |q_{ij}| \leqslant 1 \qquad \max_i |r_{ij}| \leqslant \sqrt{n}$$

In questo caso $a = 1$ e $b = \sqrt{n}$. Dunque **la fattorizzazione è stabile in senso debole.**

Analisi all'indietro dell'errore nella risoluzione di un sistema

Partendo da numeri finiti e supponendo che le operazioni di macchina non alterino la scelta del pivot, si dimostra che la fattorizzazione di Gauss ottenuta in aritmetica finita è la fattorizzazione esatta di

$$(PA + \delta A) = \mathcal{L}\mathcal{R} \quad \|\delta A\|_\infty \leqslant u \cdot n^2 \cdot \max_{i,j} |r_{ij}|$$

dove u è la precisione di macchina.
Le soluzioni calcolate $\tilde{\boldsymbol{y}}^*$ e $\tilde{\boldsymbol{x}}^*$ dei sistemi $\mathcal{L}\boldsymbol{y} = P\boldsymbol{b}$ e $\mathcal{R}\boldsymbol{x} = \tilde{\boldsymbol{y}}^*$ sono soluzioni esatte di

$$\begin{cases} (\mathcal{L} + \delta\mathcal{L})\boldsymbol{y} = P\boldsymbol{b} & \|\delta\mathcal{L}\|_\infty \leqslant 1.01u\dfrac{n(n+1)}{2} \max |\ell_{ij}| \\ (\mathcal{R} + \delta\mathcal{R})\boldsymbol{x} = \tilde{\boldsymbol{y}}^* & \|\delta\mathcal{R}\|_\infty \leqslant 1.01u\dfrac{n(n+1)}{2} \max |r_{ij}| \end{cases}$$

Allora $\tilde{\boldsymbol{x}}^*$ è soluzione esatta di

$$(PA + E)\boldsymbol{x} = P\boldsymbol{b}$$

Analisi all'indietro dell'errore nella risoluzione di un sistema

Si può caratterizzare la matrice E? Andando a ritroso, si trova che $\tilde{\boldsymbol{x}}^*$ risulta soluzione di

$$(\mathcal{L} + \delta\mathcal{L})(\mathcal{R} + \delta\mathcal{R})\boldsymbol{x} = P\boldsymbol{b}$$
$$\big(\mathcal{L}\mathcal{R} + \mathcal{L}(\delta\mathcal{R}) + (\delta\mathcal{L})\mathcal{R} + (\delta\mathcal{L})(\delta\mathcal{R})\big)\boldsymbol{x} = P\boldsymbol{b}$$

Poiché $PA + \delta A = \mathcal{L}\mathcal{R}$, segue

$$\big(PA + \delta A + \mathcal{L}(\delta\mathcal{R}) + (\delta\mathcal{L})\mathcal{R} + (\delta\mathcal{L})(\delta\mathcal{R})\big)\boldsymbol{x} = P\boldsymbol{b}$$

Dunque

$$E = \delta A + \mathcal{L}(\delta\mathcal{R}) + (\delta\mathcal{L})\mathcal{R} + (\delta\mathcal{L})(\delta\mathcal{R})$$

Analisi all'indietro dell'errore nella risoluzione di un sistema

Poichè $\|\mathcal{L}\|_\infty \leqslant n$ e $\|\mathcal{R}\|_\infty \leqslant n \ \max|r_{ij}|$, si ha

$$\begin{aligned}\|E\|_\infty &\leqslant n^2 u \ \max|r_{ij}| + \\ &\quad + n\, 1.01\ u\frac{n(n+1)}{2}\ \max|r_{ij}| + \\ &\quad + 1.01\ u\frac{n(n+1)}{2}\ n\ \max|r_{ij}| + \\ &\quad + (1.01)^2 u^2 \left(\frac{n(n+1)}{2}\right)^2 \max|r_{ij}| \\ &\leqslant u \cdot \max|r_{ij}|(n^2 + 1.01(n^3+n^2) + 1.02\, n^2) \\ &= 1.01 \cdot u \cdot \max|r_{ij}|(3n^2+n^3)\end{aligned}$$

supposto $u\frac{(n+1)^2}{4} < 1$.

Analisi all'indietro dell'errore nella risoluzione di un sistema

Conclusione: a partire da numeri finiti, la soluzione calcolata è soluzione esatta di

$$(PA+E)\boldsymbol{x} = P\boldsymbol{b}$$

dove $\|E\|_\infty \leqslant 1.01 \cdot u(3n^2+n^3)\max|r_{ij}|$. Ma $\max|r_{ij}| \leqslant f(n)\ \max|a_{ij}|$ e dunque

$$\|E\|_\infty \leqslant 1.01 \cdot u(3n^2+n^3) f(n) \cdot \max|a_{ij}|$$

dove

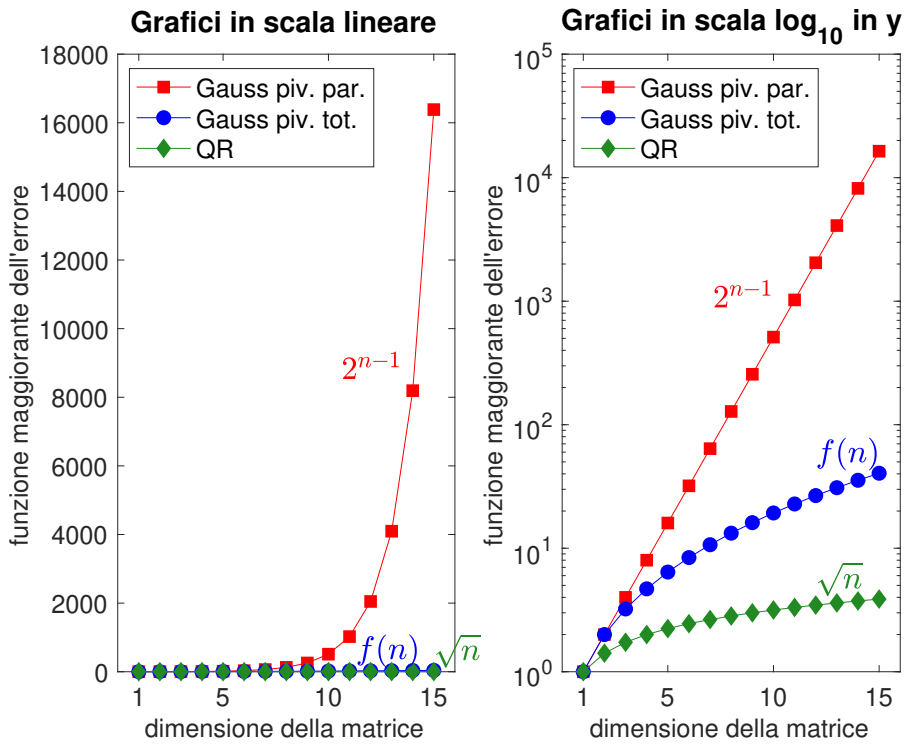
$$f(n) = \begin{cases} 1 & \text{se } A \text{ è simm. def. pos.} \\ n & \text{se } A \text{ è di Hessemberg con piv. parz.} \\ 2^{n-1} & \text{con pivoting parziale} \\ \sqrt{n}\sqrt{1\cdot 2\cdot 3^{1/2}\cdots n^{1/(n-1)}} & \text{con pivoting totale} \\ 2 & \text{per matrici tridiagonali diag. dom.} \\ \sqrt{n} & \text{per la fattorizzazione } QR \end{cases}$$

La stima è estremamente pessimistica per la maggior parte delle matrici. A parte alcuni casi patologici, nella pratica $\|E\|_\infty \leqslant\ u \cdot n \cdot \|A\|_\infty$. Si conclude che l'errore relativo nel caso di algoritmi stabili dipende fortemente dal condizionamento:

$$\frac{\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_\infty}{\|\boldsymbol{x}^*\|_\infty} \leqslant \frac{\mu_\infty(A)}{1-\mu_\infty(A)\dfrac{\|E\|_\infty}{\|A\|_\infty}}\frac{\|E\|_\infty}{\|A\|_\infty}$$

Funzioni maggiorazione dell'errore

Grafici delle funzioni di maggiorazione dell'errore in funzione della dimensione n della matrice per i metodi di fattorizzazione di Gauss con pivoting parziale e totale e per il metodo di fattorizzazione QR:



Metodi iterativi per sistemi lineari: richiami teorici

Prima di affrontare la teoria sui metodi iterativi è necessario richiamare qualche nozione sugli autovalori e su come essi incidono sulla convergenza di una successione di matrici.

Teoremi di localizzazione degli autovalori

Primo Teorema di Gerschgorin

Sia $A \in \mathbb{R}^{n\times n}$. Gli autovalori di A stanno nell'unione dei dischi di Gerschgorin K_i, $i = 1, \ldots, n$, dove

$$K_i = \left\{ z \in \mathbb{C},\ |z - a_{ii}| \leqslant \sum_{\substack{j=1,\ldots,n \\ j\neq i}} |a_{ij}| \right\}$$

K_i è un disco di centro a_{ii} e raggio $\sum_{\substack{j=1,\ldots,n \\ j\neq i}} |a_{ij}|$.

Poiché gli autovalori di A coincidono con gli autovalori di A^T, segue che gli autovalori di A stanno anche nell'unione dei dischi H_i dove

$$H_i = \left\{ z \in \mathbb{C},\ |z - a_{ii}| \leqslant \sum_{\substack{j=1,\ldots,n \\ j\neq i}} |a_{ji}| \right\}.$$

Pertanto gli autovalori stanno sia in $\cup K_i$ che in $\cup H_i$ e dunque appartengono a $(\cup K_i) \cap (\cup H_i)$.

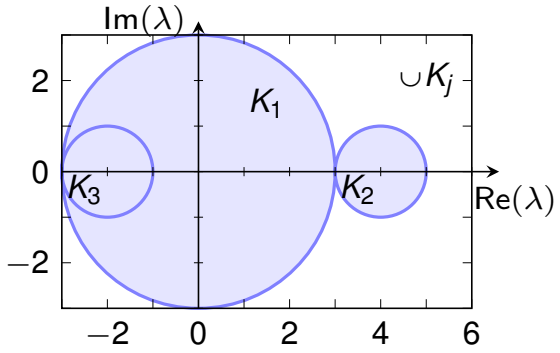
Questi risultati permettono di limitare la regione di ricerca degli autovalori di una matrice.

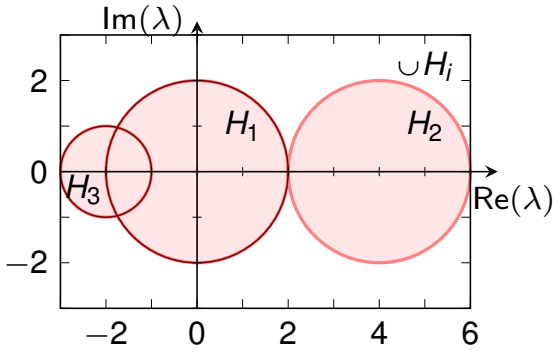
Esempio

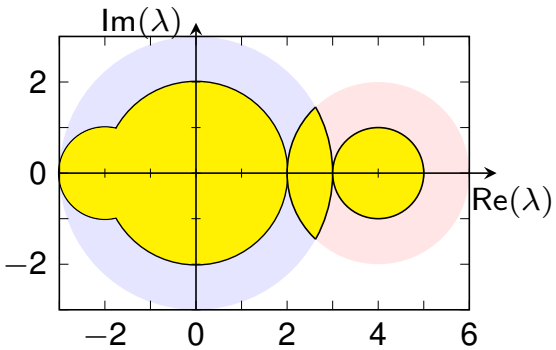
$$A = \begin{pmatrix} 0 & 2 & -1 \\ 1 & 4 & 0 \\ 1 & 0 & -2 \end{pmatrix}$$

K_1 ha centro 0 e raggio 3
K_2 ha centro 4 e raggio 1
K_3 ha centro -2 e raggio 1

H_1 ha centro 0 e raggio 2
H_2 ha centro 4 e raggio 2
H_3 ha centro -2 e raggio 1







regione gialla = $(\cup K_j) \cap (\cup H_i)$

Teoremi di localizzazione degli autovalori

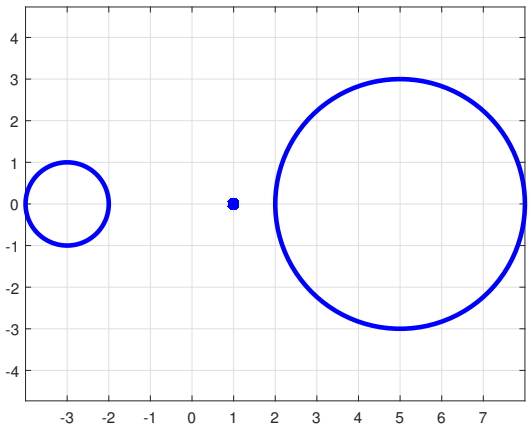
Secondo Teorema di Gerschgorin

Se l'unione M_1 di r dischi di Gerschgorin è disgiunta dall'unione M_2 dei rimanenti $n-r$ dischi, allora r autovalori di A appartengono a M_1 e $n-r$ appartengono a M_2.

Esempio.

$$A = \begin{pmatrix} 5 & -2 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & -3 \end{pmatrix}$$

Autovalori: $\lambda_1 = 4.8730$, $\lambda_2 = -2.8730$, $\lambda_3 = 1.0000$.



Matrici riducibili e irriducibili

Definizione

Sia A una matrice quadrata di ordine n. A si dice **riducibile** se esiste una matrice di permutazione P tale che

$$PAP^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \qquad A_{11} \in \mathbb{R}^{k \times k}$$

con $0 < k < n$. Se non esiste una tale matrice P, allora A è **irriducibile**.

La permutazione P non è unica.

Esempio.

$$A = \begin{pmatrix} 5 & -2 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & -3 \end{pmatrix} \quad \Leftrightarrow \quad PAP^T = \begin{pmatrix} 5 & 1 & -2 \\ -1 & -3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

È riducibile mediante permutazione elementare che scambia seconda e terza riga e colonna ($k = 2$).

Matrici riducibili e irriducibili

Se la matrice A è associata a un sistema e se essa è riducibile, si può sempre ridurre la soluzione del sistema di ordine n alla risoluzione di due sistemi di ordine k ed $n-k$, rispettivamente:

$$A\boldsymbol{x} = \boldsymbol{b}$$
$$PAP^T P\boldsymbol{x} = P\boldsymbol{b}$$

Posto $P\boldsymbol{x} = \boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix}$ e $P\boldsymbol{b} = \boldsymbol{c} = \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix}$, si ha:

$$A_{11}\boldsymbol{y}_1 + A_{12}\boldsymbol{y}_2 = \boldsymbol{c}_1$$
$$A_{22}\boldsymbol{y}_2 = \boldsymbol{c}_2$$

Pertanto si risolve il secondo sistema di dimensione $n-k$ determinando $\boldsymbol{y}_2$ e poi, sostituendo nel primo sistema di dimensione k, si ottiene $\boldsymbol{y}_1$. Permutando opportunamente le componenti si ottiene $\boldsymbol{x} = P^T\boldsymbol{y}$.
Invece di risolvere un sistema di ordine n si risolvono due sistemi di dimensioni minori.

Grafi e matrici

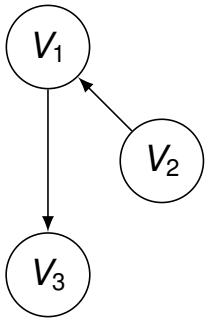
Per vedere se una matrice è riducibile si introduce la seguente definizione.

Definizione

Il grafo orientato associato a una matrice $A \in \mathbb{R}^{n\times n}$ è costituito da n nodi $(V_1, V_2, \ldots, V_n)$ ed è tale che, per ogni $a_{ij} \neq 0$ con $i \neq j$, esiste un arco orientato (i,j) che collega il nodo V_i al nodo V_j.

Esempio.

$$A = \begin{pmatrix} 1 & 0 & 3 \\ 5 & -1 & 0 \\ 0 & 0 & 7 \end{pmatrix}$$



Definizione

Un cammino orientato tra i nodi V_i e V_j è una successione di archi orientati $(i_1, i_2), (i_2, i_3), \ldots, (i_k, i_{k+1})$ consecutivi con $i_1 = i$ e $i_{k+1} = j$. La **lunghezza** del cammino è k.

Definizione

Un **Grafo orientato strettamente connesso** è un grafo orientato tale che per ogni coppia di nodi V_i, V_j con $i \neq j$, esiste un cammino orientato che connette V_i e V_j.

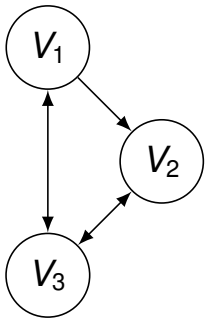
Caratterizzazione delle matrici irriducibili

A è irriducibile $\Leftrightarrow$ il grafo associato è strettamente connesso.

Esempi.

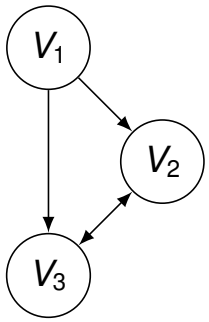
La seguente matrice è irriducibile:

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 0 & 2 & 1 \\ 2 & 4 & 0 \end{pmatrix}$$



La seguente matrice è riducibile (il nodo V_2 non è connesso al nodo V_1 da nessun cammino):

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 0 & 2 & 1 \\ 0 & 4 & 2 \end{pmatrix}$$



Matrici tridiagonali: sono irriducibili $\Leftrightarrow$ $\beta_i \neq 0$, $i = 1, \ldots, n-1$, $\gamma_i \neq 0$, $i = 2, \ldots, n$.

$$\begin{pmatrix} \alpha_1 & \beta_1 & & \\ \gamma_2 & \alpha_2 & \beta_2 & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \gamma_n & \alpha_n \end{pmatrix}$$

Terzo Teorema di Gerschgorin

Se A è una matrice quadrata di ordine n irriducibile e se esiste un autovalore di A che appartiene alla frontiera dell'unione dei dischi di Gerschgorin, allora l'autovalore appartiene alla frontiera di ogni disco.

Esempi.

$$A = \begin{pmatrix} 2 & -2 & 0 \\ -1 & 2 & -1 \\ 0 & -2 & 2 \end{pmatrix}$$

Autovalori: $\lambda_1 = 2$, $\lambda_2 = 4$, $\lambda_3 = 0$.

$$B = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

Autovalori $\lambda_1 = 0.5858$, $\lambda_2 = 2.0000$, $\lambda_3 = 3.4142$.
I valori 4 o 0 non possono essere autovalori!!

Conseguenze

Sia A una matrice strettamente diagonale dominante per righe o per colonne oppure irriducibilmente diagonale dominante (ossia irriducibile e diagonale dominante con almeno una riga o una colonna per cui vale la diseguaglianza in senso stretto). Allora:

- A è non singolare.
  (Se A è strettamente diagonale dominante per righe, qualsiasi disco K_i non contiene l'origine; pertanto gli autovalori di A sono diversi da 0 e A è non singolare.
  Se A è irriducibilmente diagonale dominante, zero può appartenere alla frontiera di $\cup K_i$. Ma in tal caso, poichè A è irriducibile, se zero è autovalore, allora esso deve appartenere alla frontiera di ogni K_i e c'è almeno un disco che non lo contiene).

- Se inoltre $a_{ii} > 0$, allora $\mathrm{Re}\big(\lambda_i(A)\big) > 0$.

- Inoltre, se A è simmetrica, allora A è simmetrica definita positiva.
  (Infatti gli autovalori di A hanno parte reale positiva e sono reali. Una matrice con autovalori reali positivi è simmetrica definita positiva).

Potenze di matrici quadrate

Di particolare importanza è la successione delle potenze di una matrice quadrata.

Definizione

Una matrice $A \in \mathbb{R}^{n\times n}$ si dice convergente se $\lim_{k\to\infty} A^k = \mathbf{0}$, dove con $\mathbf{0}$ si intende la matrice identicamente nulla. Equivalentemente, A è convergente se $\lim_{k\to\infty} \|A^k\| = 0$ o $\lim_{k\to\infty} (A^k)_{ij} = 0 \ \forall i, j$.

Esempio.

$$A = \begin{pmatrix} 1/2 & 0 \\ 1/4 & 1/2 \end{pmatrix} \qquad A^2 = \begin{pmatrix} 1/4 & 0 \\ 1/4 & 1/4 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 1/8 & 0 \\ 3/16 & 1/8 \end{pmatrix} \qquad A^4 = \begin{pmatrix} 1/16 & 0 \\ 1/8 & 1/16 \end{pmatrix}$$

$$A^k = \begin{pmatrix} 1/2^k & 0 \\ k/2^{k+1} & 1/2^k \end{pmatrix} \qquad \lim_{k\to\infty} A^k = \mathbf{0}$$

Raggio spettrale

Teorema (Hirsh)

Sia $A \in \mathbb{R}^{n\times n}$. Per ogni norma naturale $\|\cdot\|$ è

$$\rho(A) \leqslant \|A\|$$

dove $\rho(A) = \max_{i=1,\dots,n} |\lambda_i(A)|$ è il raggio spettrale della matrice A. Inoltre per ogni $\epsilon > 0$ esiste una norma naturale per cui

$$\|A\| \leqslant \rho(A) + \epsilon$$

Condizioni per la convergenza di matrici

Teorema

Sia $A \in \mathbb{R}^{n\times n}$: A è convergente $\Leftrightarrow \rho(A) < 1$.

Dim.
"$\Rightarrow$" Se A è convergente, $\lim_{k\to\infty} A^k = \mathbf{0}$. Allora per ogni $\epsilon > 0$ esiste ν_ϵ tale che $\|A^k\| < \epsilon$ per ogni $k > \nu_\epsilon$.
Poiché (Teorema di Hirsh) $\rho(A^k) \leqslant \|A^k\| < \epsilon$, è

$$\rho(A^k) = \max_i |\lambda_i(A^k)| = \max_i |\lambda_i(A)^k| = \max_i |\lambda_i(A)|^k = \rho(A)^k$$

da cui segue che $\rho(A)^k < \epsilon$. Da $\lim_{k\to\infty} \rho(A)^k = 0$, si conclude $\rho(A) < 1$.

"$\Leftarrow$" Viceversa, se $\rho(A) < 1$, dato che $\exists \mu > 0$ tale che $\rho(A) + \mu < 1$, si ha che esiste una norma naturale per cui $\|A\| \leqslant \rho(A) + \mu < 1$. Allora

$$\|A^k\| = \|A \cdots A\| \leqslant \|A\| \cdots \|A\| \leqslant \|A\|^k \leqslant (\rho(A) + \mu)^k$$

Poiché $\lim_{k\to\infty} (\rho(A) + \mu)^k = 0$, segue $\lim_{k\to\infty} \|A^k\| = \mathbf{0}$ e dunque A è convergente.

Condizioni equivalenti di convergenza di matrici

Sono equivalenti le seguenti proposizioni:

1. A è convergente;
2. $\|A^k\| \to \mathbf{0}$ per $k \to \infty$;
3. $\rho(A) < 1$;
4. $A^k \boldsymbol{x} \to \mathbf{0}$ per $k \to \infty$ per ogni $\boldsymbol{x} \in \mathbb{R}^n$
5. $I_n - A$ è non singolare e la serie di Neumann

$$\sum_{i=0}^{\infty} A^i = (I_n - A)^{-1}$$

è convergente.

Metodi iterativi

Si considera nuovamente il sistema lineare

$$A\boldsymbol{x} = \boldsymbol{b} \qquad A \in \mathbb{R}^{n\times n}, \quad \boldsymbol{b} \in \mathbb{R}^n, \quad \det(A) \neq 0$$

A partire da una qualunque approssimazione iniziale $\boldsymbol{x}^{(0)}$ della soluzione $\boldsymbol{x}^*$, si costruisce una successione di approssimazioni $\{\boldsymbol{x}^{(k)}\}$, via via migliori, che per k tendente all'infinito convergono alla soluzione $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$.

Poiché ci si arresta a un passo $\bar{k} < +\infty$, occorre valutare l'errore di troncamento $\boldsymbol{e}^{(\bar{k})} = \boldsymbol{x}^{(\bar{k})} - \boldsymbol{x}^*$.

A differenza dei metodi diretti, quelli iterativi consentono di mantenere la struttura di una matrice e quindi di usare **tecniche di memorizzazione compatta per matrici sparse** (si vedano le librerie ITPACK ed ELLPACK).
I metodi iterativi si applicano quando la matrice A è sparsa e di grandi dimensioni. La complessitè di una iterazione è pari al costo del prodotto matrice vettore (pari al numero di elementi non nulli della matrice). La velocitè di convergenza è solo lineare, per cui spesso occorre usare tecniche di accelerazione per minimizzare le iterazioni necessarie ad ottenere l'approssimazione entro la tolleranza richiesta.
Matrici con struttura sparsa si incontrano nella risoluzione di equazioni differenziali con condizioni ai limiti, risolte con la tecnica delle differenze finite o degli elementi finiti.

Costruzione di un metodo iterativo

Consideriamo il generico sistema lineare

$$A\boldsymbol{x} = \boldsymbol{b} \qquad A \in \mathbb{R}^{n\times n}, \quad \boldsymbol{b} \in \mathbb{R}^n, \quad \det(A) \neq 0$$

Sia $M \in \mathbb{R}^{n\times n}$ **non singolare**. Il sistema è equivalente a

$$\begin{aligned} M\boldsymbol{x} &= M\boldsymbol{x} + (\boldsymbol{b} - A\boldsymbol{x}) \\ \boldsymbol{x} &= \boldsymbol{x} + M^{-1}(\boldsymbol{b} - A\boldsymbol{x}) \\ \boldsymbol{x} &= \boldsymbol{x} - M^{-1}A\boldsymbol{x} + M^{-1}\boldsymbol{b} \\ \boldsymbol{x} &= \left(I_n - M^{-1}A\right)\boldsymbol{x} + M^{-1}\boldsymbol{b} \\ \boldsymbol{x} &= G\boldsymbol{x} + \boldsymbol{c} \end{aligned}$$

dove $G = I_n - M^{-1}A$ e $\boldsymbol{c} = M^{-1}\boldsymbol{b}$.
La seconda formulazione permette di innescare un procedimento iterativo:

$$\boldsymbol{x}^{(k+1)} = G\boldsymbol{x}^{(k)} + \boldsymbol{c}$$

con $\boldsymbol{x}^{(0)}$ qualunque, o equivalentemente

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + M^{-1}\left(\boldsymbol{b} - A\boldsymbol{x}^{(k)}\right) = \boldsymbol{x}^{(k)} + M^{-1}\boldsymbol{r}^{(k)}$$

dove G si dice **matrice di iterazione** ed $\boldsymbol{r}^{(k)}$ è il **residuo**. Si tratta di un metodo **lineare** (è $\boldsymbol{x}^{(k+1)}$ dipende linearmente da $\boldsymbol{x}^{(k)}$), **stazionario** (è G e $\boldsymbol{c}$ sono costanti), **del prim'ordine** (è $\boldsymbol{x}^{(k+1)}$ dipende solo da $\boldsymbol{x}^{(k)}$).

Convergenza di un metodo iterativo

Definizione

Un metodo iterativo è **convergente** se per ogni $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ iniziale la successione $\{\boldsymbol{x}^{(k)}\}$ generata da

$$\boldsymbol{x}^{(k+1)} = G\boldsymbol{x}^{(k)} + \boldsymbol{c}$$

converge per $k \to \infty$.

Se A e M sono non singolari e se $\{\boldsymbol{x}^{(k)}\}$ converge a $\overline{\boldsymbol{x}}$ a partire da ogni $\boldsymbol{x}^{(0)}$ iniziale, allora $\overline{\boldsymbol{x}}$ è l'unica soluzione del sistema $A\boldsymbol{x} = \boldsymbol{b}$, ossia il metodo iterativo è **consistente**.

Convergenza

Per studiare la convergenza si introduce l'errore di troncamento al passo k:

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

dove $\boldsymbol{x}^*$ è la soluzione del sistema $A\boldsymbol{x} = \boldsymbol{b}$, ovvero del sistema equivalente $\boldsymbol{x} = G\boldsymbol{x} + \boldsymbol{c}$. Si nota che $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)} = A\left(\boldsymbol{x}^* - \boldsymbol{x}^{(k)}\right) = -A\boldsymbol{e}^{(k)}$.

Convergenza

Condizione necessaria e sufficiente perchè $\{\boldsymbol{x}^{(k)}\} \to \boldsymbol{x}^*$ per $k \to \infty$ per ogni $\boldsymbol{x}^{(0)}$ è che $\{\boldsymbol{e}^{(k)}\} \to \boldsymbol{0}$ per $k \to \infty$ per ogni $\boldsymbol{e}^{(0)}$, o anche che $\{\boldsymbol{r}^{(k)}\} \to \boldsymbol{0}$ per $k \to \infty$ per ogni $\boldsymbol{r}^{(0)}$.

$$\boldsymbol{e}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^* = G\boldsymbol{x}^{(k-1)} + \boldsymbol{c} - G\boldsymbol{x}^* - \boldsymbol{c} = G\left(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^*\right) = G\boldsymbol{e}^{(k-1)}$$

$$\boldsymbol{e}^{(k)} = G\boldsymbol{e}^{(k-1)} = G\left(G\boldsymbol{e}^{(k-2)}\right) = G^2\boldsymbol{e}^{(k-2)} = G^2\left(G\boldsymbol{e}^{(k-3)}\right) = G^3\boldsymbol{e}^{(k-3)} = \ldots = G^k\boldsymbol{e}^{(0)}$$

Condizione di convergenza

Teorema

Condizione necessaria e sufficiente affinché un metodo iterativo sia convergente è che $\rho(G) < 1$.

Dim. Il metodo converge se e solo se $\lim_{k\to\infty} \boldsymbol{e}^{(k)} = \boldsymbol{0} \Leftrightarrow \lim_{k\to\infty} G^k \boldsymbol{e}^{(0)} = \boldsymbol{0}$ per ogni $\boldsymbol{e}^{(0)}$ $\Leftrightarrow$ G è convergente $\Leftrightarrow$ $\rho(G) < 1$.

$$\lim_{k\to\infty} \boldsymbol{r}^{(k)} = \boldsymbol{0} \quad \forall \boldsymbol{r}^{(0)} \quad \Leftrightarrow \quad \lim_{k\to\infty} \left(-A\boldsymbol{e}^{(k)}\right) = \boldsymbol{0} \quad \forall \boldsymbol{e}^{(0)} \quad \Leftrightarrow \quad \lim_{k\to\infty} \boldsymbol{e}^{(k)} = \boldsymbol{0} \quad \forall \boldsymbol{e}^{(0)}$$

Condizione sufficiente per la convergenza di un metodo iterativo è che $\|G\| < 1$.

Condizioni necessarie (ma non sufficienti) per la convergenza di un metodo iterativo sono $|\det(G)| < 1$ e $|\mathrm{trace}(G)| < n$.

Osservazione. La traccia di una matrice quadrata è la somma degli autovalori, il determinante è il prodotto degli autovalori

Velocità asintotica di convergenza

Il confronto tra la velocità di convergenza di due metodi può essere fatto solo **asintoticamente**.

Ricordando che

$$\boldsymbol{e}^{(k)} = G^k \boldsymbol{e}^{(0)}$$

si dimostra che

$$\lim_{k\to\infty} \left(\frac{\|\boldsymbol{e}^{(k)}\|}{\|\boldsymbol{e}^{(0)}\|} \right)^{1/k} = \lim_{k\to\infty} \|G^k\|^{1/k} = \rho(G)$$

Si chiama velocità asintotica di convergenza la quantità:

$$\mathcal{R}_\infty(G) = -\ln\left(\rho(G)\right)$$

il numero $\dfrac{1}{\mathcal{R}_\infty(G)}$ è una stima del numero di passi che occorre fare per ridurre l'errore iniziale di $1/\mathrm{e}$:

$$\rho(G)^k \approx \frac{1}{\mathrm{e}} \quad \Rightarrow \quad k \approx -\frac{1}{\ln\left(\rho(G)\right)} = \frac{1}{\mathcal{R}_\infty(G)}$$

Arresto di un procedimento iterativo

- Un possibile test si basa sul controllo di $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|$. Da

$$\begin{aligned}\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} &= G\boldsymbol{x}^{(k)} + \boldsymbol{c} - \boldsymbol{x}^{(k)} = G\boldsymbol{x}^{(k)} + \left(\boldsymbol{x}^* - G\boldsymbol{x}^*\right) - \boldsymbol{x}^{(k)} \\ &= (G - I_n)\boldsymbol{x}^{(k)} - (G - I_n)\boldsymbol{x}^* = (G - I_n)\left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\right)\end{aligned}$$

  segue che, poiché $\rho(G) < 1$ e $G - I_n$ è non singolare, $\boldsymbol{x}^{(k)} - \boldsymbol{x}^* = (G - I_n)^{-1}\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right)$. Dunque,

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| \leqslant \|(I_n - G)^{-1}\|\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|$$

  Ci si arresta se $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| < \epsilon$. Ciò non garantisce che $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|$ sia piccolo, perché dipende da $\|(I_n - G)^{-1}\|$.
- Da

$$\frac{\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|} \leqslant \mu(A)\frac{\|\boldsymbol{r}^{(k+1)}\|}{\|\boldsymbol{b}\|}$$

  ci si può arrestare se $\dfrac{\|\boldsymbol{r}^{(k+1)}\|}{\|\boldsymbol{b}\|} < \epsilon$. Se la matrice A non ha $\mu(A)$ troppo grande, $\boldsymbol{x}^{(k+1)}$ è accettabile come soluzione.

I metodi iterativi non necessitano di analisi di errore di arrotondamento, perché ogni $\boldsymbol{x}^{(k)}$ può essere considerato come iterato iniziale e gli errori influiscono solo sull'ultima iterazione.

Metodi particolari

Ricordiamo che, dati il sistema $A\boldsymbol{x} = \boldsymbol{b}$ e una matrice M non singolare, si riscrive il sistema come $M\boldsymbol{x} = M\boldsymbol{x} + (\boldsymbol{b} - A\boldsymbol{x})$. Questo serve ad innescare il procedimento iterativo seguente:

$$\boldsymbol{x}^{(k+1)} = \left(I_n - M^{-1}A\right)\boldsymbol{x}^{(k)} + M^{-1}\boldsymbol{b}$$

Poiché M influenza il numero di iterazioni con cui si ottiene l'approssimazione desiderata, la scelta ottimale è $M^{-1} = A^{-1}$. Ovviamente non è una scelta praticabile.

È opportuno scegliere M in modo che "assomigli" ad A, ma abbia una struttura che permetta di calcolarne facilmente l'inversa. Un modo per fare questo è cercare una **decomposizione** (o *splitting*) della matrice A:

$$A = M - N$$

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

$$\boldsymbol{x}^{(k+1)} = M^{-1}N\boldsymbol{x}^{(k)} + M^{-1}\boldsymbol{b} = \left(I_n - M^{-1}A\right)\boldsymbol{x}^{(k)} + M^{-1}\boldsymbol{b}$$

$$\boldsymbol{x}^{(k+1)} = G\boldsymbol{x}^{(k)} + \boldsymbol{c}$$

dove M **è una matrice non singolare facile da invertire** (un "pezzo" di A facile da invertire); in tal caso

$$G = M^{-1}N = \left(I_n - M^{-1}A\right), \qquad \boldsymbol{c} = M^{-1}\boldsymbol{b}$$

Una particolare decomposizione di A

$$A = D - L - U = D - (L + U)$$

$$A = \begin{pmatrix} & -U \\ D & \\ -L & \end{pmatrix} \qquad D = \operatorname{diag}(a_{ii}) = \begin{pmatrix} a_{ii} & & 0 \\ & \ddots & \\ 0 & & a_{ii} \end{pmatrix}$$

$$L = \begin{cases} -a_{ij} & j < i \\ 0 & \text{altrove} \end{cases} = \begin{pmatrix} 0 & \cdots & 0 \\ -a_{ij} & \ddots & \vdots \\ & & 0 \end{pmatrix}$$

$$U = \begin{cases} -a_{ij} & j > i \\ 0 & \text{altrove} \end{cases} = \begin{pmatrix} 0 & & -a_{ij} \\ \vdots & \ddots & \\ 0 & \cdots & 0 \end{pmatrix}$$

Metodo di Jacobi (o metodo degli spostamenti simultanei)

Se $a_{ii} \neq 0 \ \forall i = 1, \ldots, n$ allora $A = M - N = D - (L + U)$ con $M = D$, $N = L + U$

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

La matrice di iterazione, detta **matrice di Jacobi**, è data da

$$J = D^{-1}(L + U) = \begin{pmatrix} 1/a_{11} & & \\ & 1/a_{22} & \\ & & \ddots \\ & & 1/a_{nn} \end{pmatrix} \begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -a_{n-1,n} \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & -a_{12}/a_{11} & \cdots & -a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -a_{n-1,n}/a_{n-1,n-1} \\ -a_{n1}/a_{nn} & \cdots & -a_{n,n-1}/a_{nn} & 0 \end{pmatrix}$$

Formulazione per componenti: $$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(- \sum_{\substack{j=1,\ldots,n \\ j \neq i}} a_{ij} x_j^{(k)} + b_i \right)$$

Esempio

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

La soluzione vale $\boldsymbol{x}^* = (1, 2, -1, 1)^T$.

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5} \\ x_2^{(k+1)} &= \frac{1}{11}x_1^{(k)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \\ x_3^{(k+1)} &= -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10} \\ x_4^{(k+1)} &= -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} + \frac{15}{8} \end{aligned} \quad \Leftrightarrow \quad \boldsymbol{x}^{(k+1)} = J\boldsymbol{x}^{(k)} + \boldsymbol{c}$$

$$J = \begin{pmatrix} 0 & 1/10 & -1/5 & 0 \\ 1/11 & 0 & 1/11 & -3/11 \\ -1/5 & 1/10 & 0 & 1/10 \\ 0 & 3/8 & 1/8 & 0 \end{pmatrix} \qquad \boldsymbol{c} = \begin{pmatrix} 3/5 \\ 25/11 \\ -11/10 \\ 15/8 \end{pmatrix}$$

Esempio

k	0	1	2	3	...	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	...	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0533	...	1.9998
$x_3^{(k)}$	0	−1.1000	−0.8052	−1.0493	...	−0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	...	0.9998

Il costo computazionale è pari a un prodotto matrice per vettore per ogni iterazione.

Codice Matlab

```
function [x, k] = jacobi(A, b, x, maxit, tol)
% JACOBI - Metodo iterativo di Jacobi per sistemi lineari
n = size(A,1);
if ( issparse(A) )
    d = spdiags(A, 0);
else
    d = diag( A );
end
if ( any( abs(d) < eps*norm(d, inf) ) )
    error('Elementi diagonali troppo piccoli.');
end
b = b ./ d;
k = 0; stop = 0;
while ( ~stop )
    k = k + 1;
    xtemp = x;
    x = x - (A*x)./d + b; % istruzione vettoriale
    stop = ( norm(xtemp - x, inf) < tol*norm(x, inf) ) ...
           || ( k == maxit );
end
if ( k == maxit )
    warning('Raggiunto numero max di iterazioni maxit = %d', ...
             maxit
end
end % fine della M-function jacobi.m
```


Metodo di Gauss-Seidel (o metodo degli spostamenti successivi)

$$A = M - N = (D - L) - U$$

$$M = D - L \qquad N = U$$

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

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

La matrice di iterazione, detta **matrice di Gauss-Seidel**, è data da
$\mathcal{G} = (D - L)^{-1}U = I_n - (D - L)^{-1}A$.
Allora ogni passo comporta la soluzione di un sistema triangolare inferiore, con l'algoritmo di eliminazione in avanti.

Se $a_{ii} \neq 0$, $\forall i = 1, \ldots, n$, allora una riga di $(D - L)\boldsymbol{x}^{(k+1)} = U\boldsymbol{x}^{(k)} + \boldsymbol{b}$ è data da

$$a_{i1}x_1^{(k+1)} + a_{i2}x_2^{(k+1)} + \ldots + a_{i,i-1}x_{i-1}^{(k+1)} + a_{ii}x_i^{(k+1)} = -a_{i,i+1}x_{i+1}^{(k)} - \ldots - a_{in}x_n^{(k)} + b_i$$

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

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

Esempio

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

La soluzione vale $\boldsymbol{x}^* = (1, 2, -1, 1)^T$.

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{10}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{3}{5} \\ x_2^{(k+1)} &= \frac{1}{11}x_1^{(k+1)} + \frac{1}{11}x_3^{(k)} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \\ x_3^{(k+1)} &= -\frac{1}{5}x_1^{(k+1)} + \frac{1}{10}x_2^{(k+1)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10} \\ x_4^{(k+1)} &= -\frac{3}{8}x_2^{(k+1)} + \frac{1}{8}x_3^{(k+1)} + \frac{15}{8} \end{aligned} \quad \Leftrightarrow \quad \boldsymbol{x}^{(k+1)} = \mathcal{G}\boldsymbol{x}^{(k)} + \boldsymbol{c}$$

$$\mathcal{G} = (D - L)^{-1}U = \begin{pmatrix} 10 & & & \\ -1 & 11 & & \\ 2 & -1 & 10 & \\ 0 & -3 & -1 & 8 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 & -2 & 0 \\ & 0 & 1 & -3 \\ & & 0 & 1 \\ & & & 0 \end{pmatrix}$$

Esempio

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0300	1.0065	1.0009	1.0001
$x_2^{(k)}$	0	2.3272	2.0370	2.0036	2.0003	2.0000
$x_3^{(k)}$	0	−0.9873	−1.0140	−1.0025	−1.0003	−1.0000
$x_4^{(k)}$	0	0.8789	0.9844	0.9983	0.9999	1.0000

Il costo computazionale è pari a un prodotto matrice per vettore per ogni iterazione.

Codice Matlab

```
function [x, k] = gaussSeidel(A, b, x, maxit, tol)
% GAUSSSEIDEL - Metodo iterativo di Gauss-Seidel per sistemi lineari
n = size(A,1);
k = 0; stop = 0;
while ( ~stop )
   k = k + 1; xtemp = x;
   for i = 1 : n
      x(i) = ( -A(i,[1:i-1, i+1:n]) * x([1:i-1, i+1:n]) + b(i) ) ...
             / A(i,i);
   end
   stop = ( norm(xtemp - x, inf) < tol*norm(x, inf) ) ...
          || ( k == maxit );
end
if ( k == maxit )
    warning('Raggiunto numero max di iterazioni maxit = %d', ...
            maxit);
end
end % fine della M-function gaussSeidel.m
```


Convergenza dei metodi di Jacobi e di Gauss-Seidel

Teorema

Se A è una matrice quadrata di ordine n strettamente diagonale dominante per righe o per colonne, oppure irriducibilmente diagonale dominante per righe o per colonne, allora

- il metodo di Jacobi converge;
- il metodo di Gauss-Seidel converge e $\|\mathcal{G}\|_\infty \leqslant \|J\|_\infty$

Dim. (Convergenza del metodo di Jacobi). Se A è strettamente diagonale dominante per righe, allora $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$. Poiché $a_{ii} \neq 0$, $\forall i = 1, \ldots, n$, segue che

$$1 > \max_i \left(\sum_{j \neq i} |a_{ij}|/|a_{ii}| \right) = \|J\|_\infty$$

Allora $\rho(J) \leqslant \|J\|_\infty < 1$ e il metodo è convergente.
Se A è irriducibilmente diagonale dominante per righe, allora $\|J\|_\infty \leqslant 1$. Se fosse $\rho(J) = 1$, esisterebbe un autovalore λ tale che $|\lambda| = 1$ e dunque tale autovalore apparterrebbe alla frontiera dell'unione dei dischi di Gerschgorin (che sono di centro 0 e raggio minore od uguale a 1). Ma se A è irriducibile, lo è anche J. Poichè ogni disco di Gerschgorin deve passare per tale autovalore (per il terzo

Convergenza dei metodi di Jacobi e di Gauss-Seidel

teorema di Gerschgorin) e per almeno un indice i è $\sum_{j \neq i} |a_{ij}|/|a_{ii}| < 1$, allora non può essere $\rho(J) = 1$, bensì $\rho(J) < 1$.
La dimostrazione della convergenza del metodo di Gauss-Seidel e della relazione delle norme delle matrici di iterazione è lasciata per esercizio.

Osservazione. Dal fatto che $\|\mathcal{G}\|_\infty \leqslant \|J\|_\infty$ non si può dedurre che il metodo di Gauss-Seidel converga più velocemente del metodo di Jacobi.

Teorema di Stein-Rosemberg

Se $J \geqslant 0$, allora si può verificare uno solo dei seguenti casi:

- $0 < \rho(\mathcal{G}) < \rho(J) < 1$
- $0 = \rho(\mathcal{G}) = \rho(J)$
- $1 = \rho(\mathcal{G}) = \rho(J)$
- $1 < \rho(J) < \rho(\mathcal{G})$

Se $J \geqslant 0$, i metodi di Gauss-Seidel e di Jacobi convergono entrambi o divergono entrambi. Nel caso in cui convergano entrambi, il metodo di Gauss-Seidel è asintoticamente più veloce del metodo di Jacobi.

Convergenza dei metodi iterativi per matrici definite positive

Teorema

Sia A quadrata di ordine n simmetrica con $a_{ii} > 0$. Allora A è definita positiva se e solo se il metodo di Gauss-Seidel è convergente.

Teorema

Sia A quadrata di ordine n simmetrica e sia $2D - A$ definita positiva. Allora A è definita positiva se e solo se il metodo di Jacobi è convergente.

Metodi iterativi parametrici

$$A = M(\omega) - N(\omega)$$

dove $M(\omega)$ è non singolare.

$$\begin{aligned} M(\omega)\boldsymbol{x}^{(k+1)} &= N(\omega)\boldsymbol{x}^{(k)} + \boldsymbol{b} \\ \boldsymbol{x}^{(k+1)} &= \big(M(\omega)\big)^{-1} N(\omega)\boldsymbol{x}^{(k)} + \big(M(\omega)\big)^{-1}\boldsymbol{b} \\ \boldsymbol{x}^{(k+1)} &= G(\omega)\boldsymbol{x}^{(k)} + \boldsymbol{c}(\omega) \\ G(\omega) &= \big(M(\omega)\big)^{-1} N(\omega) = I_n - \big(M(\omega)\big)^{-1} A \\ \boldsymbol{c}(\omega) &= \big(M(\omega)\big)^{-1}\boldsymbol{b} \end{aligned}$$

- Occorre determinare i valori di ω per cui $M(\omega)$ è non singolare e $\rho\big(G(\omega)\big) < 1$; sia Ω l'insieme di tali valori;
- occorre poi determinare entro Ω il valore ω^* per il quale si ha che

$$\rho\big(G(\omega^*)\big) = \min_{\omega\in\Omega} \rho\big(G(\omega)\big) = \min_{\omega\in\Omega} \max_{i=1,\ldots,n} \big|\lambda_i\big(G(\omega)\big)\big|$$

Metodi iterativi parametrici

Un modo per introdurre tale parametro è usare la tecnica di rilassamento o di estrapolazione entro un metodo noto:

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega x_i^{(k+1/2)}$$

dove $x_i^{(k+1/2)}$ è ottenuto da $x_i^{(k)}$ applicando un passo del metodo che si vuole estrapolare o rilassare.

- $\omega = 1$ fornisce il metodo di partenza;
- con $\omega > 1$ si parla di sovrarilassamento;
- con $\omega < 1$ si parla di sottorilassamento;

Spiegazione della formula di rilassamento

In pratica, invece di usare $x_i^{(k)} + (M^{-1}\boldsymbol{r}^{(k)})_i$ come nuovo iterato di un metodo noto, si usa

$$\begin{aligned} x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \omega x_i^{(k+1/2)} = x_i^{(k)} + \omega\big(x_i^{(k+1/2)} - x_i^{(k)}\big) \\ &= x_i^{(k)} + \omega\Big(x_i^{(k)} + \big(M^{-1}\boldsymbol{r}^{(k)}\big)_i - x_i^{(k)}\Big) \\ x_i^{(k+1)} &= x_i^{(k)} + \omega\big(M^{-1}\boldsymbol{r}^{(k)}\big)_i \end{aligned}$$

Metodo estrapolato di Jacobi

Si ricorda che il metodo di Jacobi è dato da

$$\boldsymbol{x}^{(k+1)} = D^{-1}(L+U)\boldsymbol{x}^{(k)} + D^{-1}\boldsymbol{b} = \big(I_n - D^{-1}A\big)\boldsymbol{x}^{(k)} + D^{-1}\boldsymbol{b}$$

Costruiamo il metodo estrapolato ($A = D - L - U$).

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= (1-\omega)\boldsymbol{x}^{(k)} + \omega\boldsymbol{x}^{(k+1/2)} \\
&= (1-\omega)\boldsymbol{x}^{(k)} + \omega\Big(\big(I_n - D^{-1}A\big)\boldsymbol{x}^{(k)} + D^{-1}\boldsymbol{b}\Big) \\
&= (1-\omega)\boldsymbol{x}^{(k)} + \omega\big(\boldsymbol{x}^{(k)} - D^{-1}A\boldsymbol{x}^{(k)} + D^{-1}\boldsymbol{b}\big) \\
&= (1-\omega)\boldsymbol{x}^{(k)} + \omega\boldsymbol{x}^{(k)} - \omega D^{-1}A\boldsymbol{x}^{(k)} + \omega D^{-1}\boldsymbol{b} \\
&= \boldsymbol{x}^{(k)} - \omega D^{-1}A\boldsymbol{x}^{(k)} + \omega D^{-1}\boldsymbol{b} \\
\boldsymbol{x}^{(k+1)} &= \underbrace{\big(I_n - \omega D^{-1}A\big)}_{J(\omega)}\boldsymbol{x}^{(k)} + \underbrace{\omega D^{-1}}_{M(\omega)^{-1}}\boldsymbol{b}
\end{aligned}$$

Pertanto

$$M(\omega) = \frac{1}{\omega}D$$

$$N(\omega) = M(\omega) - A = \frac{1}{\omega}D - A = \frac{1}{\omega}(D - \omega A)$$

$$\big(M(\omega)\big)^{-1}N(\omega) = I_n - \omega D^{-1}A$$

Metodo estrapolato di Gauss-Seidel (SOR)

Si ricorda che il metodo di Gauss-Seidel è dato da

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right).$$

Si costruisce il metodo estrapolato:

$$\begin{aligned}
x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \omega\frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right) \\
a_{ii}x_i^{(k+1)} &= a_{ii}(1-\omega)x_i^{(k)} + \omega\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right) \\
a_{ii}x_i^{(k+1)} + \omega\sum_{j<i} a_{ij}x_j^{(k+1)} &= a_{ii}(1-\omega)x_i^{(k)} - \omega\sum_{j>i} a_{ij}x_j^{(k)} + \omega b_i
\end{aligned}$$

In notazione matriciale

$$\begin{aligned}
(D - \omega L)\boldsymbol{x}^{(k+1)} &= \Big((1-\omega)D + \omega U\Big)\boldsymbol{x}^{(k)} + \omega\boldsymbol{b} \\
\boldsymbol{x}^{(k+1)} &= (D-\omega L)^{-1}\Big((1-\omega)D + \omega U\Big)\boldsymbol{x}^{(k)} + (D-\omega L)^{-1}\omega\boldsymbol{b}
\end{aligned}$$

Metodo estrapolato di Gauss-Seidel (SOR)

Pertanto si ha che

$$M(\omega) = \frac{1}{\omega}(D - \omega L) \qquad N(\omega) = M(\omega) - A = \frac{1}{\omega}\Big((1-\omega)D + \omega U\Big)$$

$$\mathcal{G}(\omega) = (D - \omega L)^{-1}\Big((1-\omega)D + \omega U\Big)$$

Un esempio di SOR

$$A = \begin{pmatrix} 4 & 2 & 0 \\ -1 & 5 & 3 \\ 0 & 2 & 4 \end{pmatrix} \qquad \boldsymbol{b} = \begin{pmatrix} 6 \\ 7 \\ 6 \end{pmatrix}$$

Si pone $\omega = 1.2$. L'approssimazione iniziale è $\boldsymbol{x}^{(0)} = (0,0,0)^T$.

$$x_1^{(1/2)} = \frac{1}{4}\left(-2x_2^{(0)} + 6\right) = \frac{3}{2} \quad \Rightarrow \quad x_1^{(1)} = (1-\omega)0 + \omega\frac{3}{2} = 1.8$$

$$x_2^{(1/2)} = \frac{1}{5}\left(x_1^{(1)} - 3x_3^{(0)} + 7\right) = 1.76 \quad \Rightarrow \quad x_2^{(1)} = (1-\omega)0 + \omega 1.76 = 2.112$$

$$x_3^{(1/2)} = \frac{1}{4}\left(-2x_2^{(1)} + 6\right) = 0.444 \quad \Rightarrow \quad x_3^{(1)} = (1-\omega)0 + \omega 0.444 = 0.5328$$

Per cui $\boldsymbol{x}^{(1)} = (1.8, 2.112, 0.5328)^T$. Le successive iterazioni forniscono:

$$\boldsymbol{x}^{(2)} = (0.1728,\ 0.9150,\ 1.1440)^T$$
$$\boldsymbol{x}^{(3)} = (1.2162,\ 0.9650,\ 0.9922)^T$$
$$\boldsymbol{x}^{(4)} = (0.9778,\ 1.0070,\ 0.9972)^T$$
$$\boldsymbol{x}^{(5)} = (1.0000,\ 1.0010,\ 1.0000)^T$$

Codice Matlab

```
function [x, k] = sor(A, b, x, maxit, tol, omega)
% SOR - Metodo SOR (Gauss-Seidel estrapolato) per sistemi lineari
n = size(A,1);
k = 0; stop = 0;
while ( ~stop )
   k = k + 1; xtemp = x;
   for i = 1 : n
      x(i) = ( -A(i, [1:i-1, i+1:n]) * x([1:i-1, i+1:n]) + b(i) )...
             / A(i,i);
      x(i) = (1-omega)*xtemp(i) + omega*x(i);
   end
   stop = ( norm(xtemp - x, inf) < tol*norm(x, inf) ) ...
          || ( k == maxit );
end
if ( k == maxit )
   warning('Raggiunto numero max di iterazioni maxit = %d', ...
           maxit);
end
end % fine della M-function sor.m
```


Dominio di ω: $(0,2)$

Teorema di Kahan

$$\rho\big(\mathcal{G}(\omega)\big) \geqslant |\omega - 1|.$$

Dim.

$$\begin{aligned}
\det\big(\mathcal{G}(\omega)\big) &= \det\Big((D-\omega L)^{-1}\big((1-\omega)D+\omega U\big)\Big) \\
&= \det\Big((D-\omega L)^{-1}\Big)\det\Big((1-\omega)D+\omega U\Big) \\
&= \left(\prod_{i=1}^{n} a_{ii}\right)^{-1} \prod_{i=1}^{n}\big((1-\omega)a_{ii}\big) = \left(\prod_{i=1}^{n} a_{ii}\right)^{-1} (1-\omega)^n \left(\prod_{i=1}^{n} a_{ii}\right) \\
&= (1-\omega)^n
\end{aligned}$$

$$\prod_{i=1}^{n} |\lambda_i\big(\mathcal{G}(\omega)\big)| = \big|\det\big(\mathcal{G}(\omega)\big)\big| = \big|(1-\omega)^n\big| \leqslant \Big(\rho\big(\mathcal{G}(\omega)\big)\Big)^n \quad \Rightarrow \quad |1-\omega| \leqslant \rho\big(\mathcal{G}(\omega)\big)$$

Infatti il determinante è il prodotto degli autovalori e il modulo di ogni autovalore è maggiorato dal massimo dei moduli (raggio spettrale).

Perché ci sia convergenza, **è necessario** che $|\omega - 1| < 1$, ossia che $0 < \omega < 2$, anche se questo **non è sufficiente**.

Matrici definite positive e metodo SOR

Teorema di Ostrowski-Reich

Sia A simmetrica con $a_{ii} > 0$ e sia $0 < \omega < 2$. Allora A è definita positiva se e solo se il metodo SOR converge.

Scegliere il parametro ottimale

Il problema di trovare il valore ottimale ω^* di ω per cui

$$\rho\big(\mathcal{G}(\omega^*)\big) = \min_{0<\omega<2} \rho\big(\mathcal{G}(\omega)\big)$$

si risolve solo per particolari classi di matrici. Tra tali classi ci sono le matrici tridiagonali con elementi diagonali non nulli.
Se A è tridiagonale con elementi diagonali non nulli, valgono le seguenti asserzioni:

- se η è autovalore della matrice di Jacobi J, anche $-\eta$ è autovalore di J;
- se η è autovalore di J e vale la relazione

  $$(\lambda + \omega - 1)^2 = \omega^2 \lambda \eta^2$$

  allora λ è autovalore di $\mathcal{G}(\omega)$;
- se λ è un autovalore non nullo di $\mathcal{G}(\omega)$ e vale la relazione

  $$(\lambda + \omega - 1)^2 = \omega^2 \lambda \eta^2$$

  allora η è un autovalore di J.

In particolare, per $\omega = 1$ si ha $\lambda = \eta^2$ e dunque $\big(\rho(J)\big)^2 = \rho\big(\mathcal{G}\big)$ e dunque, se convergono entrambi, si ha $\mathcal{R}_\infty(\mathcal{G}) = 2\mathcal{R}_\infty(J)$.

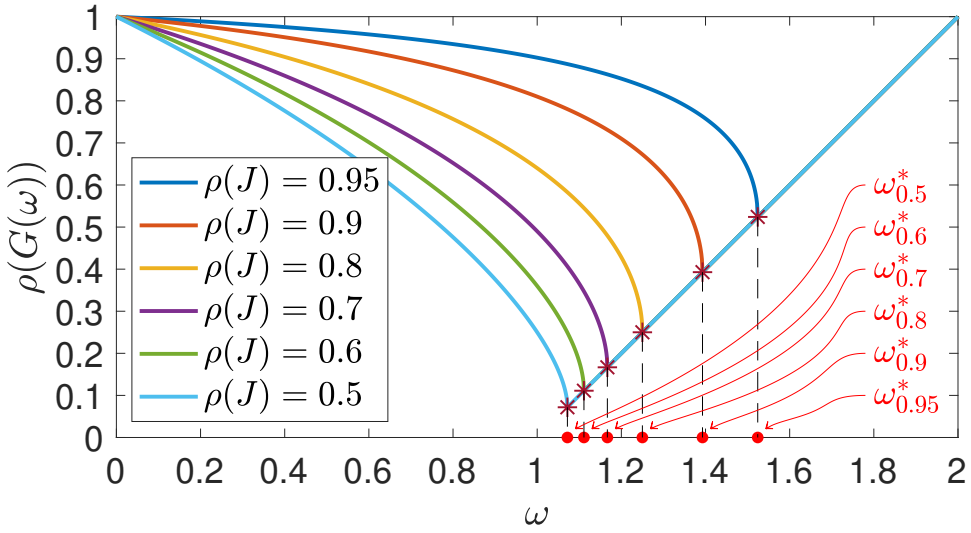
I metodi di Jacobi e SOR convergono entrambi o divergono entrambi.

Scegliere il parametro ottimale

Se $\rho(J) < 1$ e gli autovalori di J sono reali, allora il valore ottimale di ω è

$$\omega^* = \frac{2}{1+\sqrt{1-\rho^2(J)}} > 1 \quad \Rightarrow \quad \rho\big(G(\omega^*)\big) = \frac{1-\sqrt{1-\rho^2(J)}}{1+\sqrt{1-\rho^2(J)}} = \omega^* - 1$$

$$\rho\big(\mathcal{G}(\omega)\big) = \begin{cases} 1-\omega+\dfrac{1}{2}\omega^2\rho^2(J)+\omega\rho(J)\sqrt{1-\omega+\omega^2\rho^2(J)/4} & \text{se } 0<\omega<\omega^* \\ \omega-1 & \text{se } \omega^* \leqslant \omega < 2 \end{cases}$$



Esempio

$$A = \begin{pmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}$$

La matrice è irriducibilmente diagonale dominante e $\rho(J) < 1$.

$$J = \begin{pmatrix} 0 & -3/4 & 0 \\ -3/4 & 0 & 1/4 \\ 0 & 1/4 & 0 \end{pmatrix}$$

$\det(J - \eta I) = -\eta(\eta^2 - 0.625)$, dunque $\eta_1 = 0$, $\eta_{2,3} = \pm\sqrt{0.625}$. Pertanto $\rho(J) = 0.79057$. Gli autovalori di $\mathcal{G}(\omega)$ sono tali che:

$$(\lambda + \omega - 1)^2 = \lambda\omega^2\eta^2$$

$\rho(\mathcal{G}) = 0.625$ dunque SOR converge e $\omega^* = \dfrac{2}{1+\sqrt{1-0.625}} \approx 1.24$. Inoltre $\rho\big(\mathcal{G}(\omega^*)\big) = 0.24$.

Velocità di convergenza	Iterazioni necessarie a ridurre di $1/e$ l'errore iniziale
$\mathcal{R}_\infty(J) = 0.235$	$4.26 \quad (\approx 4 \div 5)$
$\mathcal{R}_\infty(\mathcal{G}) = 0.47$	$2.13 \quad (\approx 2 \div 3)$
$\mathcal{R}_\infty\big(\mathcal{G}(\omega^*)\big) = 1.4271$	$0.70 \quad (\approx 1)$