

Metodi diretti per sistemi lineari

Sistema di m equazioni in n incognite:

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1, \dots, m$$



(1)

$$Ax = b$$

Soluzione del sistema: n -upla che soddisfi tali equazioni. Trattiamo solo sistemi quadrati

Ovvero tali che: $m = n$, per cui: $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

In tal caso, $\exists_1 x \in \mathbb{R}^n$ soluzione di (1) se e solo se:

1) $\exists A^{-1}$ oppure 2) $\text{rank}(A) = n$ oppure 3) $A\underline{x} = 0 \Rightarrow \underline{x} = 0$.

Teorema di Cramer

Se $\det(A) \neq 0$ \exists_1 soluzione del sistema data da:

$$x_i = \frac{\det(\Delta_i)}{\det(A)}$$

(2)

$$\text{con } \Delta_i = \begin{vmatrix} a_{11} & \dots & b_1 & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \dots & b_n & \dots & a_{nn} \end{vmatrix}$$



i-esima colonna

Costo computazionale di (2): $(n+1)!$ flops.

Se $n = 50$, 10^9 flops \Rightarrow time $\approx 10^{47}$ anni!

La risoluzione numerica di un sistema lineare prevede due possibili strategie: quelle basate sui *metodi diretti* e quelle basate sui *metodi iterativi*. La scelta del tipo di metodo si basa essenzialmente sul tipo di matrice e sulle risorse a disposizione: tempo di calcolo e spazio di memoria. Infatti, mentre i **metodi diretti sono adatti ai sistemi con matrici piene**, i **metodi iterativi sono adatti ai sistemi con matrici sparse**, contenenti cioè molti zeri.

Poiché, come vedremo, il risultato dei metodi diretti è sempre un sistema triangolare, occupiamoci prima di risolvere un tale sistema.

Risoluzione di sistemi triangolari

- Metodo delle sostituzioni in avanti.

Sia dato il seguente sistema lineare 3x3 non degenere:

$$\begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$Lx = b$$

Poiché, per ipotesi, $\det(L) \neq 0 \Rightarrow \ell_{ii} \neq 0$, la soluzione è quindi data da:

$$\begin{cases} x_1 = b_1 / \ell_{11} \\ x_2 = (b_2 - \ell_{21}x_1) / \ell_{22} \\ x_3 = (b_3 - \ell_{31}x_1 - \ell_{32}x_2) / \ell_{33} \end{cases}$$

In generale si ha quindi:

$$x_1 = b_1 / \ell_{11}$$

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij}x_j \right) / \ell_{ii} \quad i = 2, \dots, n$$

Costo computazionale: numero di moltiplicazioni e divisioni = $n(n+1)/2$

numero di addizioni e sottrazioni = $n(n-1)/2$

per un totale di $\approx n^2$ flops.

- Metodo delle sostituzioni indietro.

Si deve risolvere il sistema: $Ux = b$ ovvero:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\Rightarrow x_n = b_n / u_{nn}$$

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

che ha la stessa complessità computazionale del metodo precedente.

Metodi diretti

La soluzione è ottenuta con un numero finito di passi.

Metodo di eliminazione di Gauss

Sia $Ax = b$ con $\det(A) \neq 0$:

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + \dots + a_{nn}x_n = b_n \end{cases}.$$

Sia $a_{11} \neq 0$. Se ciò non si ha si scambia la prima riga con una delle successive in cui il coefficiente di x_1 sia diverso da zero.

Sia $m_{i1}^{(1)} = -\frac{a_{i1}}{a_{11}}$ per $i = 2, \dots, n$ e aggiungiamo alla i -esima equazione la prima equazione

moltiplicata per m_{i1} . Si ha:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}^{(2)}x_2 + \dots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ \vdots \\ a_{n2}^{(2)}x_2 + \dots + a_{nn}^{(2)}x_n = b_n^{(2)} \end{cases}.$$

dove: $a_{ij}^{(2)} = a_{ij} + m_{i1}^{(1)} a_{1j}$ $i, j = 2, \dots, n$

$$b_i^{(2)} = b_i + m_{i1}^{(1)} b_1 \quad i = 2, \dots, n$$

Operiamo allo stesso modo nel secondo passo moltiplicando per $m_{i2} = -\frac{a_{i2}^{(2)}}{a_{22}^{(2)}}$.

Al passo $n-1$ si ottiene un sistema triangolare che si risolve con il metodo della sostituzione all'indietro.

Il costo computazionale del metodo di Gauss è $\approx \frac{4}{3}n^3$.

Perché il metodo di Gauss funzioni è necessario che gli elementi pivotali $a_{ii}^{(i)}$ siano diversi da zero. Purtroppo, il solo fatto che gli elementi diagonali di A siano non nulli non è sufficiente a garantire che nei passi successivi gli elementi pivotali non si annullino. Infatti sia:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix} \quad a_{ii} \neq 0, \quad i=1,2,3$$

Eppure:

$$A^{(2)} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & -1 \\ 0 & -6 & -12 \end{bmatrix} \quad \text{da cui } a_{22}^{(2)} = 0$$

Abbiamo quindi bisogno di condizioni più restrittive su A . Si ha che se tutti i minori principali di A sono non nulli allora anche gli elementi diagonali in tutti i passi di eliminazione, ovvero gli elementi pivotali, saranno non nulli. Poiché la matrice A dell'esempio precedente ha il secondo minore principale uguale a zero, scambiando in $A^{(2)}$ la seconda e la terza riga il metodo funziona.

Per evitare inoltre problemi di arrotondamento si usano le tecniche del *pivot parziale* e del *pivot totale*.

Pivot parziale. Al j -esimo passo si cerca la riga I contenente il massimo elemento della

j -esima colonna: $a_{Ij} = \max_{i \leq n} |a_{ij}|$ e si scambia la riga i con la riga I . Pertanto al primo passo: $a_{11} = \max_{1 \leq i \leq n} |a_{i1}|$.

Pivot totale. Si trova il massimo elemento della matrice: $a_{IJ} = \max_{i,j} |a_{ij}|$ e si scambiano la riga i con la riga I e la colonna j con la colonna J .

Il metodo del pivot totale è più preciso ma bisogna memorizzare l'ordine di eliminazione delle variabili e quindi si occupa molta memoria.

Per far vedere la necessità del pivoting abbiamo il seguente esempio, supponendo di lavorare in un'aritmetica con 4 cifre decimali:

$$\begin{cases} 0.0001x + 1.00y = 1.00 \\ 1.00x + 1.00y = 2.00 \end{cases}$$

$$x = 1.00010 \quad y = 0.99990 \quad \text{soluz. analitica}$$

$$x_G = 0.00 \quad y_G = 1.00 \quad \text{soluz. con Gauss}$$

Riscrivendo il sistema:

$$\begin{cases} 1.00x + 1.00y = 2.00 \\ 0.0001x + 1.00y = 1.00 \end{cases}$$

$$x_G = 1.00, \quad y_G = 1.00$$

Metodi di fattorizzazione. Sono una riformulazione matriciale del metodo di Gauss. Consistono nel trovare una matrice S non singolare e formare un sistema equivalente a quello originale.

$$Ax = b \Rightarrow SAx = Sb, \quad SA = U$$

U = matrice triangolare superiore.

Se S è triangolare inferiore lo è pure S^{-1} :

$$A = S^{-1}U = LU$$

Fattorizzazione **LU** : L triangolare inferiore, $\ell_{ii} = 1 \Rightarrow$ Gauss

Fattorizzazione **LL^T** : L con elementi diagonali positivi \Rightarrow Cholesky

Fattorizzazione **QR** : Q ortogonale, R triangolare superiore \Rightarrow Householder

Riformulazione matriciale del metodo di Gauss

I vantaggi di fattorizzare A nel prodotto LU derivano dal fatto che L ed U non dipendono dal termine noto. Poiché il costo computazionale della procedura di eliminazione è $\approx n^3$ flops si ha un risparmio di operazioni se si devono risolvere più sistemi lineari che hanno la stessa matrice.

Sia :

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}$$

e:

$$L_1 = \begin{bmatrix} 1 & & & 0 \\ m_{21} & 1 & & \\ \vdots & & \ddots & \\ m_{n1} & 0 & & 1 \end{bmatrix} \quad \text{con } m_{i1} = -\frac{a_{i1}}{a_{11}} \quad i = 2, \dots, n$$

Il prodotto L_1A equivale al primo passo di Gauss.

In generale, il passo i -esimo è $L_i A$, dove:

$$L_i = \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ & m_{ji} & 1 & \\ 0 & \vdots & 0 & \ddots \\ & m_{ni} & & 1 \end{bmatrix} \quad \text{con } m_{ji} = -\frac{a_{ji}^{(i)}}{a_{ii}^{(i)}} \quad j = i+1, \dots, n$$

Alla fine si ha: $U = L_{n-1}L_{n-2}\dots L_2L_1A$

Poniamo: $\tilde{L} = L_{n-1}\dots L_1 \Rightarrow U = \tilde{L}A; \quad A = \tilde{L}^{-1}U$ e ponendo $L = \tilde{L}^{-1}$ si ha: $A = LU$.

La soluzione di

$$Ax = b \Leftrightarrow LUx = b$$

si trova in due passi:

- i) si pone: $Ly = b$ e si risolve per y
- ii) da: $Ux = y$ si trova x .

La fattorizzazione LU può essere combinata con il pivoting e con lo scaling dei fattori mediante la pre o post moltiplicazione con matrici di permutazione.

Matrici di permutazione

Una matrice di permutazione è una matrice ottenuta scambiando le righe o le colonne della matrice identità. In particolare, scambiando la riga i con la riga j di I e premoltiplicando la matrice così ottenuta per A si ottiene lo stesso scambio di righe, invece postmoltiplicando si ottiene lo scambio di colonne.

In generale, se vogliamo scambiare la riga i con la riga j dobbiamo premoltiplicare A per la matrice $P^{(i,j)}$ di elementi

$$p_{rs}^{(i,j)} = \begin{cases} 1 & \text{se } r = s = 1, \dots, i-1, i+1, \dots, j-1, j+1, \dots, n \\ 1 & \text{se } r = j, s = i \text{ o } r = i, s = j \\ 0 & \text{altrimenti} \end{cases}$$

Così, ad esempio, se: $P = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}$, il prodotto PA darà uno scambio della prima e

seconda riga, mentre AP darà uno scambio della prima e seconda colonna.

Non c'è comunque unicità nella scelta di L ed U se L ed U sono generiche. Ciò si può vedere in due modi:

$$I. \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} \ell_{11} & & 0 \\ \vdots & \ddots & \\ \ell_{n1} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ & \ddots & \\ 0 & & u_{nn} \end{bmatrix}$$

Uguagliando i termini si hanno n^2 equazioni che però contengono ognuna $\frac{n(n+1)}{2}$ incognite per un totale di $n^2 + n$ incognite; n di esse vanno quindi determinate arbitrariamente.

II. Siano L_1U_1 ed L_2U_2 due fattorizzazioni di A:

$$A = L_1U_1 = L_2U_2 \Rightarrow L_2^{-1}L_1 = U_2U_1^{-1}$$

Poiché la matrice a sinistra è triangolare inferiore e quella a destra è triangolare superiore, perché esse siano uguali devono necessariamente essere diagonali. Indicando tale matrice diagonale con D, si ha:

$$L_1 = L_2D, U_1 = D^{-1}U_2$$

Scegliendo come costanti arbitrarie del punto I. :

$$\ell_{11} = \ell_{22} = \dots = \ell_{nn} = 1$$

si ha il metodo di **Doolittle**, che è il metodo di fattorizzazione equivalente all'eliminazione gaussiana senza pivoting.

Scegliendo invece:

$$u_{11} = u_{22} = \dots = u_{nn} = 1$$

si ha il metodo di **CROUT**.

Da un punto di vista computazionale, è possibile memorizzare le matrici L ed U nella stessa area di memoria di A. Pertanto questi ultimi due metodi sono *metodi compatti* in

quanto permettono di memorizzare L ed U nell'area di memoria di A non essendo necessario memorizzare gli elementi, rispettivamente, ℓ_{ii} o u_{ii} .

Comunque, non sempre esiste una fattorizzazione LU di A.

Esempio: $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Sebbene esista A^{-1} non è possibile fattorizzare A.

Invece la matrice $I + A$, che è singolare, ha una fattorizzazione LU.

$$I+A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = LU$$

Teorema. $A \in \mathbb{C}^{n \times n}$, $\det(A_k) \neq 0 \quad k = 1, \dots, n-1 \Rightarrow \exists_1 L, U : A = LU$

Corollario. Sotto le stesse ipotesi del teorema, esiste un'unica fattorizzazione di Doolittle e

si ha che: $\det(A) = \prod_{i=1}^n u_{ii}$.

Dimostrazione del corollario.

Si dimostra per induzione. Per $k = 1$ si ha: $A_1 = L_1 U_1$ ovvero $a_{11} = \ell_{11} u_{11} = u_{11} \Rightarrow \exists_1 u_{11}$.

Sia vera la tesi per $k-1$, cioè $\exists_1 L_{k-1}, U_{k-1}$ per i quali si abbia:

$$A_{k-1} = L_{k-1} U_{k-1}, \quad \det(A_{k-1}) = \prod_{i=1}^{k-1} u_{ii}$$

e dimostriamo che:

$$A_k = L_k U_k \quad \det(A_k) = \prod_{i=1}^k u_{ii}.$$

Effettuiamo il prodotto a blocchi:

$$\begin{array}{|c|c|} \hline \boxed{L_{k-1}} & 0 \\ \hline \mathbf{v}^T & 1 \\ \hline \end{array} \begin{array}{|c|c|} \hline \boxed{U_{k-1}} & \mathbf{w} \\ \hline 0 & u_{kk} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \boxed{A_{k-1}} & \mathbf{c} \\ \hline \mathbf{b}^T & a_{kk} \\ \hline \end{array}$$

$$L_k \cdot U_k = A_k$$

$L_{k-1}U_{k-1} = A_{k-1}$ vero per le ipotesi di induzione.

$$L_{k-1}w = c \quad \det(L_{k-1}) = 1 \Rightarrow \exists_1 w : L_{k-1}w = c$$

$$v^T U_{k-1} = b^T \quad \det(A_{k-1}) = \det(L_{k-1})\det(U_{k-1}) = \det(U_{k-1}) \Rightarrow \exists_1 v : U_{k-1}^T v = b$$

$$v^T w + u_{kk} = a_{kk}, \text{ ovvero: } u_{kk} = a_{kk} - v^T w$$

Ma v, w, a_{kk} sono unici \Rightarrow anche u_{kk} è unico.

$$\Rightarrow A_k = L_k U_k, \det(A_k) = \det(L_k)\det(U_k) = \prod_{i=1}^k u_{ii} . \quad \bullet$$

Le ipotesi del teorema pero' non sono facili da verificare.

Se A e' tale che $\det(A) \neq 0 \Rightarrow \exists P$ matrice di permutazione :

$$PA = LU$$

Per due tipi di matrici non è necessario uno scambio di righe o di colonne per aversi la fattorizzazione LU: diagonalmente dominanti, simmetriche definite positive.

Metodo di Cholesky.

Teorema.

Sia $A \in \mathfrak{R}^{n \times n}$, $A = A^T$, $x^T A x > 0$ per $\forall x \neq 0 \Rightarrow$ esiste almeno una L triangolare inferiore :

$$A = LL^T$$

Se si impone che $\ell_{ii} > 0$ la fattorizzazione è unica.

Dimostrazione.

Per il criterio di Sylvester: $\det(A_k) > 0 \quad \forall k$.

Per il teorema precedente esiste un'unica fattorizzazione LU. Ponendo:

$$\begin{bmatrix} u_{11} & & 0 \\ \vdots & \ddots & \\ u_{1n} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ & \ddots & \\ 0 & & u_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

$$\text{si ha: } a_{kk} = \sum_{p=1}^k u_{pk}^2 = u_{kk}^2 + \sum_{p=1}^{k-1} u_{pk}^2 \Rightarrow u_{kk}^2 = a_{kk} - \sum_{p=1}^{k-1} u_{pk}^2$$

$$a_{kj} = \sum_{i=1}^k u_{ki} u_{ij} = u_{kk} u_{kj} + \sum_{i=1}^{k-1} u_{ki} u_{ij} \Rightarrow u_{kj} = \left(a_{kj} - \sum_{i=1}^{k-1} u_{ki} u_{ij} \right) / u_{kk} \quad k > j$$

da cui si ha il metodo di Cholesky:

$$u_{11} = \sqrt{a_{11}}$$

$$u_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} u_{ik} u_{jk} \right) / u_{jj} \quad i = 2, \dots, n \quad j = 1, \dots, i-1$$

$$u_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} u_{ik}^2 \right)^{1/2} \quad i = 2, \dots, n$$

Il costo computazionale di questo metodo e' meta' di quello di Gauss e cioe' $\frac{2}{3}n^3$.

Tale metodo è il migliore per le matrici simmetriche definite positive poiché non distrugge la simmetria della matrice.

I metodi fin qui visti utilizzano un numero di operazioni $\propto n^3$ se n è l'ordine della matrice. Mostriamo pero' alcuni casi speciali, come quello delle matrici tridiagonali, per i quali tale costo e' di ordine n se il metodo e' utilizzato ad hoc.

Sistemi tridiagonali (Algoritmo di Thomas)

$a_{ij} = 0 : |i - j| > 1$. Scriviamo la matrice, che ha $3n-2$ elementi, come prodotto di due matrici particolari le cui incognite sono $\alpha_i, i=1, \dots, n$ e $\gamma_i, i=1, \dots, n-1$.

$$\begin{bmatrix} a_1 & c_1 & 0 & & & \\ b_2 & a_2 & c_2 & 0 & & \\ & b_3 & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & & & \\ b_2 & \alpha_2 & & & & \\ & b_3 & \alpha_3 & & & \\ & & \ddots & \ddots & & \\ & 0 & & b_n & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \gamma_1 & & & & \\ & 1 & \gamma_2 & & & \\ & & \ddots & \ddots & & \\ & 0 & & \ddots & \gamma_{n-1} & \\ & & & & 1 \end{bmatrix}$$

$$\begin{aligned} a_1 &= \alpha_1 & \alpha_1 \gamma_1 &= c_1 \\ a_i &= \alpha_i + b_i \gamma_{i-1} & i &= 2, \dots, n \\ \alpha_i \gamma_i &= c_i & i &= 2, \dots, n-1 \end{aligned}$$

\Rightarrow

$$\begin{aligned}
\alpha_1 &= a_1 & \gamma_1 &= c_1/\alpha_1 \\
\alpha_i &= a_i - b_i\gamma_{i-1} & i &= 2, \dots, n \\
\gamma_i &= c_i/\alpha_i & i &= 2, \dots, n-1
\end{aligned}$$

Costo computazionale: $8n - 7$ flops.

I metodi di fattorizzazione modificano la matrice iniziale e a causa dell'effetto del *fill-in*, se la matrice iniziale è *sparsa*, cioè ha molti zeri, si hanno problemi di memoria. In tali casi è più conveniente utilizzare i metodi iterativi.