

SISTEMI TRIANGOLARI

Le matrici di $Ax = b$ è triangolare alta o bassa:

$$A = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ & 0 & \times & \times & \times \\ & & 0 & \times & \times \\ & & & 0 & \times \end{pmatrix}$$

Triang. ALTA

$$A = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Triang. BASSA

È consuetudine indicare A con le lettere U

È consuetudine indicare A con L.

Ci sono DUE **SEMPLICI** E **STABILI** (ossie, propagano poco gli errori di arrotondamento) ALGORITMI.
Vediamo il caso delle triang. ALTA.

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1m-1} & u_{1m} \\ 0 & u_{22} & u_{23} & \dots & u_{2m-1} & u_{2m} \\ & 0 & & \ddots & & \\ & & & & u_{m-2,m-2} & u_{m-2,m-1} & u_{m-2,m} \\ & & & & & u_{m-1,m-1} & u_{m-1,m} \\ & & & & & & u_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-2} \\ x_{m-1} \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-2} \\ b_{m-1} \\ b_m \end{pmatrix}$$

Dal basso ho

$$1) \quad u_{mm} x_m = b_m \Rightarrow x_m = \frac{b_m}{u_{mm}}$$

$$2) \quad u_{m-1,m-1} x_{m-1} + u_{m-1,m} x_m = b_{m-1} \Rightarrow x_{m-1} = \frac{b_{m-1} - u_{m-1,m} x_m}{u_{m-1,m-1}}$$

$$3) \quad u_{m-2,m-2} x_{m-2} + u_{m-2,m-1} x_{m-1} + u_{m-2,m} x_m = b_{m-2} \Rightarrow$$

$$x_{m-2} = \frac{b_{m-2} - (u_{m-2,m-1} x_{m-1} + u_{m-2,m} x_m)}{u_{m-2,m-2}}$$

•) Riga i -esima molti dei pesi precedenti!

$$u_{ii} x_i + u_{i,i+1} x_{i+1} + u_{i,i+2} x_{i+2} + \dots + u_{im} x_m = b_i$$

$$x_i = \frac{b_i - (u_{i,i+1} x_{i+1} + u_{i,i+2} x_{i+2} + \dots + u_{im} x_m)}{u_{ii}}$$

$$= \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^m u_{ij} x_j \right)$$

ALGORITMO (SOSTITUZIONI ALL'INDIETRO)

$$x_m = b_m / u_{mm}$$

FOR $i = m-1 : -1 : 1$

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

END

Si dimostra che richiede m^2 operazioni aritmetiche (+, -, x, ÷)
 Infatti, il passo i -esimo richiede

•) calcolo di $\sum_{j=i+1}^m u_{ij} x_j =$

$$= u_{i,i+1} x_{i+1} + u_{i,i+2} x_{i+2} + \dots + u_{im} x_m$$

$m - (i+1) + 1 = m - i$ prodotti (•)
 $m - i - 1$ somme (+)

•) $b_i - \sum_{j=i+1}^m u_{ij} x_j$ richiede 1 somma

•) $b_i - \sum u_{ij} x_j$ richiede 1 somma

•) $(b_i - \sum u_{ij} x_j) / u_{ii}$ richiede 1 divisione

Im totale ho

$$(m-i) + (m-i-1) + 1 = (2m-2i+1) \text{ operazioni.}$$

↙ vale anche per $i=m$

Per ottenere il numero totale di operazioni della somma:

$$\begin{aligned} N_{op}(m) &= \sum_{i=1}^m (2m-2i+1) = \sum_{i=1}^m (2m+1) - 2 \sum_{i=1}^m i = \\ &= (2m+1)m - \frac{m(m+1)}{2} = \\ &= 2m^2 + m - m^2 - \frac{m}{2} = m^2. \end{aligned}$$

Per un sistema triangolare basso $Lx = b$

$$\begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & l_{m3} & \dots & l_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

l'algoritmo parte dall'alto e si sposta verso il basso:

ALGORITMO (SOSTITUZIONE IN AVANTI)

$$x_1 = b_1 / l_{11}$$

FOR $i = 2 : m$

$$x_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right)$$

END.

Si tratta di ALGORITMI STABILI.

METODO DI ELIMINAZIONE GAUSSIANA

Dato il sistema NON singolare $A \underline{x} = \underline{b}$, ricordiamo che se $L \in \mathbb{R}^{m \times m}$ ($A \in \mathbb{R}^{m \times m}$) è NON singolare allora

$$A \underline{x} = \underline{b} \quad \text{è equivalente a} \quad L A \underline{x} = L \underline{b}$$

Il metodo di eliminazione gaussiana applica ripetutamente queste proprietà per ridurre il sistema $A \underline{x} = \underline{b}$ ad uno equivalente triangolare superiore (usabile con l'algoritmo precedente).

Ad esempio se per $n=4$:

$$A^{(1)} = \begin{pmatrix} \text{pivot} \circledast & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix} \longrightarrow \begin{pmatrix} x & x & x & x \\ 0 & \boxed{x} & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{pmatrix} = A^{(2)} = L_1 A^{(1)} \text{ con}$$

| modificati

$$L_1 = \begin{pmatrix} 1 & & & \\ x & 1 & & \\ x & & 1 & \\ x & & & 1 \end{pmatrix}$$

L_1 è NON singolare per cui

$$\textcircled{L_1} A^{(1)} \underline{x} = \textcircled{L_1} \underline{b}^{(1)} \quad \text{con} \quad A^{(1)} = A, \quad \underline{b}^{(1)} = \underline{b}$$

è equivalente a $A^{(1)} \underline{x} = \underline{b}^{(1)}$

Ripeto il procedimento su $L_1 A^{(1)} \underline{x} = L_1 \underline{b}^{(1)}$

$$A^{(2)} = \begin{pmatrix} x & x & x & x \\ 0 & \text{pivot} \circledast & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{pmatrix} \longrightarrow \begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & \boxed{x} & x \\ 0 & 0 & x & x \end{pmatrix} = A^{(3)} = L_2 A^{(2)} \text{ con}$$

Completati

$$L_2 = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & x & 1 & \\ 0 & x & 0 & 1 \end{pmatrix}$$

Abbiamo

$$A \underline{x} = \underline{b} \Leftrightarrow L_1 A \underline{x} = L_1 \underline{b} \Leftrightarrow L_2 L_1 A \underline{x} = L_2 L_1 \underline{b}$$

$$A^{(3)} = \begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{pmatrix} \xrightarrow{\text{pivot}} \begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{pmatrix} = A^{(4)} = L_3 A^{(3)} \quad (\text{con})$$

combinato

$$L_3 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & x & 1 \end{pmatrix}$$

Allora

$$A \underline{x} = \underline{b} \Leftrightarrow \overset{=A^{(2)}}{(L_1 A)} \underline{x} = L_1 \underline{b} \Leftrightarrow$$

$$\overset{=A^{(3)}}{(L_2 L_1 A)} \underline{x} = L_2 L_1 \underline{b} \Leftrightarrow$$

$$\overset{=A^{(4)}}{(L_3 L_2 L_1 A)} \underline{x} = L_3 L_2 L_1 \underline{b}$$

Notare che le operazioni eseguite su A sono le stesse che vengono eseguite su $\underline{b}$.

Vediamo come costruire le matrici L .

Immaginiamo di essere nella situazione seguente

$$A^{(k)} = \left(\begin{array}{c|c} \text{ } & \begin{matrix} a_{kh}^{(k)} & \text{ } & \text{ } \\ \hline \text{ } & \text{ } & \text{ } \end{matrix} \end{array} \right) \leftarrow \begin{matrix} \text{riga} \\ k\text{-esima} \end{matrix}$$

↑
colonne

L'obiettivo è azzerare gli elementi nella zona gialla.
Lo facciamo con OPERAZIONI ELEMENTARI

colonne
k-esima

Lo facciamo con
OPERAZIONI ELEMENTARI
SULLE RIGHE.

Consideriamo la riga k-esima e una riga successiva,
la i-esima ($k < i$):

0 ... 0	$a_{kk}^{(k)}$	
---------	----------------	--

riga $r_k^{(k)}$ di $A_1^{(k)}$

0 ... 0	$a_{ik}^{(k)}$	
---------	----------------	--

riga $r_i^{(k)}$ di $A^{(k)}$



voglio essere questo elemento con
l'OPERAZIONE SEMPLICE!

Sottrazione alla riga i-esima la riga k-esima
moltiplicata per un numero opportuno:

$$r_i^{(k+1)} = r_i^{(k)} - l_{ik} r_k^{(k)}$$

Affinché $r_{ik}^{(k+1)} = 0$ deve essere

$$0 = a_{ik}^{(k)} - l_{ik} \cdot a_{kk}^{(k)} \Rightarrow l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

Questo procedimento, va ripetuto per tutte le righe sotto la
riga k-esima di $A^{(k)}$.

Si dimostra che questa operazione equivale a premoltiplicare
la matrice $A^{(k)}$ per la matrice L_k data da

$$L_k = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 & & \\ & & & & \text{posizione}(k,k) & \\ & & & & & 1 & \\ & & & & & & \ddots & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix}$$

(1)
 $-l_{k+1,k}$
 $-l_{k+2,k}$

$$\left(\begin{array}{c} -c_{k+1, k} \\ -c_{k+2, k} \\ \vdots \\ -c_{n, k} \end{array} \right) \quad \dots \quad 1$$

Perciò, in $n-1$ passi porta il sistema di portanze in uno triangolare alto.

In conclusione abbiamo

con questa operazione abbiamo

$$A \underline{x} = \underline{b} \iff \underbrace{(L_{m-1} \cdots L_2 L_1 A)}_{= A^{(m)}} \underline{x} = L_{m-1} \cdots L_2 L_1 \underline{b}$$

OSSERVAZIONE [FATTORIZZAZIONE LV di A]

Da quanto visto $A^{(m)} = L_{m+1} \dots L_2 L_1 A$ è triangolare alta; si scrive

$$L_{m-1} L_{m-2} \dots L_2 L_1 A = U$$

per cui

$$A = \begin{bmatrix} L_1^{-1} & L_2^{-1} & \dots & L_{m-2}^{-1} & L_{m-1}^{-1} \end{bmatrix} \cdot U$$

Si dimostra che L è triangolare bene e ha espressione

$$L = \begin{pmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & \ddots & & \\ \vdots & \vdots & & \ddots & \\ \ell_{m1} & \ell_{m2} & & & \ell_{m,m-1} & 1 \end{pmatrix}$$

↳ quelli calcolati durante i vari passi.

Le seguenti zze zione

$$A = L V$$

con L ed U di prime e delle FATTORIZZAZIONE LU di A .

VANTAGGI DELLA FATTURIZZAZIONE LT

- ①) permette un calcolo agevole di $|A|$, $|A^{-1}|$, $|A^{-m}|$, ...
Infatti,

$$|A| = |L U| \underset{\substack{\uparrow \\ \text{Bimet}}}{=} |L| \cdot |U| = 1 \cdot |U| = |U|$$

$$|A^{-1}| = \frac{1}{|A|} = \frac{1}{|U|}$$

$$|A^{-m}| = |(A^{-1})^m| = |A^{-1}|^m = \left(\frac{1}{|U|} \right)^m$$

- ②) permette la risoluzione di molti sistemi lineari che hanno la stessa A e differenti b senza dover risolvere l'intero sistema lineare ogni volta.

$$A \underline{x} = \underline{b}_k, \quad k = 1, \dots, m, \quad m \text{ grande.}$$

Procediamo così.

1. Calcolo fattorizzazione LU di A .
 $A = LU$.

2. Risolvo, per ogni $\underline{b}_k$, i sistemi

$$LU \underline{x} = \underline{b}_k, \quad k = 1, \dots, m$$

Ognuno di questi equivale a risolvere un sistema triangolare alto ed uno triangolare basso:

$$L \underbrace{U \underline{x}}_{\substack{\text{"y"} \\ \underline{y}}} = \underline{b}_k \iff (a) \quad L \underline{y} = \underline{b}_k \implies \underline{y} \\ (b) \quad U \underline{x} = \underline{y} \implies \underline{x}$$

Questo modo di fare acquista senso pensando che
LA COMPLESSITÀ COMPUTAZIONALE DELLA

FATTORIZZAZIONE LU è $\frac{2}{3}m^3$ dove m è l'ordine di A . Questo è $\frac{2}{3}$ lo stesso costo della fattorizzazione gaussiana.

Perciò, per risolvere gli m sistemi lineari impiego un tempo proporzionale al numero di operazioni fatte che è

$m \cdot \frac{2}{3}m^3$
come fossero a
e' sistemi

$\frac{2}{3}m^3 + m \cdot (2m^2)$
con fattorizzazione LU.

OSSERVAZIONE.

Per $m = 20$ servono 700 anni con Cray.

Con eliminazione gaussiana servono

$$\frac{2}{3} \cdot 20^3 \cdot 10^{-9} \text{ s} = \frac{2}{3} \cdot 8 \cdot 10^3 \cdot 10^{-9} \approx 5 \times 10^{-6} \text{ s}.$$