

Calcolo Numerico

Metodi per sistemi lineari

Simone Rebegoldi

Corso di Laurea in Informatica
Dipartimento di Scienze Fisiche, Informatiche e Matematiche



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



Optimization Algorithms
and Software for
Inverse problemS

www.oasis.unimore.it

1. Richiami di algebra lineare e algoritmi di base

Definizione

Per **matrice** intendiamo ogni tabella di elementi del tipo

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix}$$

dove m è il numero di righe ed n è il numero di colonne. Dal punto di vista matematico, è definita come una funzione del tipo

$$\begin{aligned} A : \{1, \dots, m\} \times \{1, \dots, n\} &\rightarrow \mathbb{R} \\ (i, j) &\mapsto a_{ij} \end{aligned}$$

L'insieme delle matrici di m righe ed n colonne è indicato con $\mathbb{R}^{m \times n}$.

- L'elemento generico di una matrice si indica con a_{ij} , dove i è l'indice di riga e j è l'indice di colonna.
- Gli elementi a_{ii} , $i = 1, \dots, \min(n, m)$, sono detti **elementi diagonali**.
- Se $m = n$, la matrice è detta **quadrata di ordine (o dimensione) n** .
- Se $n = 1$, la matrice si riduce ad un **vettore colonna** di m componenti.
- Se $m = 1$, la matrice si riduce ad un **vettore riga** di n componenti.
- Se $m = n = 1$, la matrice si riduce ad uno scalare.
- Si indica con \mathbb{R}^n lo spazio dei vettori colonna di n componenti.

N.B.

La controparte informatica di un vettore o matrice è l'**array**. Il numero totale degli elementi di una matrice è nm , perciò la memorizzazione di una matrice utilizzando il formato double richiede uno spazio di $8nm$ byte.

Definizione

Data $A \in \mathbb{R}^{m \times n}$, la sua **trasposta** è la matrice $A^T \in \mathbb{R}^{n \times m}$ definita come

$$A^T = \begin{pmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{m1} \\ a_{12} & a_{22} & a_{32} & \cdots & a_{m2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{mn} \end{pmatrix}.$$

In altre parole: è la matrice ottenuta scambiando le righe con le colonne di A .

Definizione

Una matrice quadrata di ordine n si dice **simmetrica** se

$$A = A^T.$$

Ciò è equivalente a dire che $a_{ij} = a_{ji}$, $i, j = 1, \dots, n$.

Definizione

Una matrice quadrata D di ordine n si dice **diagonale** se

$$D = \begin{pmatrix} d_{11} & 0 & 0 & \cdots & 0 \\ 0 & d_{22} & 0 & \cdots & 0 \\ 0 & 0 & d_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d_{nn} \end{pmatrix}.$$

Ciò è equivalente a dire che $d_{ij} = 0$, per $i \neq j$.

- La matrice diagonale di ordine n che ha elementi diagonali unitari è la **matrice identità di ordine n** e si indica con I (o con I_n nel caso in cui occorra specificarne la dimensione):

$$I = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Definizione

Una matrice quadrata U si dice **triangolare superiore** se $u_{ij} = 0$ per $i > j$.

Una matrice quadrata L si dice **triangolare inferiore** se $l_{ij} = 0$ per $i < j$.

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}, \quad L = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix}.$$

Osservazioni

- Conoscere la struttura di una matrice (diagonale, triangolare, ...) permette di ridurre l'occupazione di memoria, poichè non è necessario memorizzare esplicitamente gli elementi nulli.
- Ad esempio: una matrice diagonale di ordine n ha al più n elementi non nulli. Per memorizzarli, occorre uno spazio di memoria corrispondente a n numeri floating point.
- Altro esempio: gli elementi non nulli di una matrice triangolare sono $\simeq \frac{n^2}{2}$.
- Una matrice quadrata “piena” richiede invece la memorizzazione di n^2 elementi.

Definizione (somma matriciale)

Date due matrici della stessa dimensione $A, B \in \mathbb{R}^{m \times n}$, la **somma** $A + B$ è la matrice delle stesse dimensioni $C = (c_{ij}) \in \mathbb{R}^{m \times n}$ i cui elementi si ottengono facendo la somma elemento per elemento di A e B :

$$c_{ij} = a_{ij} + b_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

- **Algoritmo della somma matriciale:**

```
FOR  $i = 1, \dots, m$   
  | FOR  $j = 1, \dots, n$   
  |   |  $c_{ij} \leftarrow a_{ij} + b_{ij}$ 
```

Complessità computazionale: nm somme floating point.

Definizione (prodotto di una matrice per uno scalare)

Data una matrice $A \in \mathbb{R}^{m \times n}$ e un numero $\lambda \in \mathbb{R}$, il **prodotto** λA è la matrice delle stesse dimensioni $C = (c_{ij}) \in \mathbb{R}^{m \times n}$ i cui elementi si ottengono moltiplicando i corrispondenti elementi di A per λ :

$$c_{ij} = \lambda a_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

- **Algoritmo del prodotto di una matrice per uno scalare:**

```
FOR  $i = 1, \dots, m$   
  | FOR  $j = 1, \dots, n$   
  |   |  $c_{ij} \leftarrow \lambda a_{ij}$ 
```

Complessità computazionale: nm prodotti floating point.

Definizione (prodotto tra matrici)

Date due matrici $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{n \times p}$, il **prodotto** AB è la matrice $C = (c_{ij}) \in \mathbb{R}^{m \times p}$ i cui elementi sono definiti dalla formula seguente

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad i = 1, \dots, m, \quad j = 1, \dots, p.$$

$$\begin{pmatrix} c_{11} & c_{1j} & c_{1p} \\ \vdots & \vdots & \vdots \\ c_{i1} & c_{ij} & c_{ip} \\ \vdots & \vdots & \vdots \\ c_{m1} & c_{mj} & c_{mp} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{1j} & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{i1} & a_{ij} & a_{in} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{mj} & a_{mn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{1j} & b_{1p} \\ \vdots & \vdots & \vdots \\ b_{i1} & b_{ij} & b_{ip} \\ \vdots & \vdots & \vdots \\ b_{n1} & b_{nj} & b_{np} \end{pmatrix}$$

Definizione (prodotto tra matrici)

Date due matrici $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{n \times p}$, il **prodotto** AB è la matrice $C = (c_{ij}) \in \mathbb{R}^{m \times p}$ i cui elementi sono definiti dalla formula seguente

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad i = 1, \dots, m, \quad j = 1, \dots, p.$$

- **Algoritmo del prodotto matrice-matrice:**

```
FOR  $i = 1, \dots, m$   
  [ FOR  $j = 1, \dots, p$   
    [  $s \leftarrow 0$   
      FOR  $k = 1, \dots, n$   
        [  $s \leftarrow s + a_{ik} b_{kj}$   
          [  $c_{ij} \leftarrow s$ 
```

Complessità computazionale:

- per ogni elemento occorrono n somme e prodotti; siccome gli elementi sono in tutto mp , la complessità ammonterà a nmp ;
- se la matrice è quadrata, la complessità è pari a n^3 .

Definizione (prodotto tra matrici)

Date due matrici $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{n \times p}$, il **prodotto** AB è la matrice $C = (c_{ij}) \in \mathbb{R}^{m \times p}$ i cui elementi sono definiti dalla formula seguente

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad i = 1, \dots, m, \quad j = 1, \dots, p.$$

- **Proprietà del prodotto matrice-matrice**

- Vale la proprietà associativa:

$$A(BC) = (AB)C, \quad \forall A \in \mathbb{R}^{m \times n}, \forall B \in \mathbb{R}^{n \times p}, \forall C \in \mathbb{R}^{p \times q}.$$

- Vale la proprietà distributiva rispetto alla somma matriciale:

$$A(B + C) = AB + AC, \quad \forall A \in \mathbb{R}^{m \times n}, \forall B, C \in \mathbb{R}^{n \times p}.$$

- **Non vale la proprietà commutativa**, nemmeno se le matrici sono quadrate:

$$\exists A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times m} : AB \neq BA.$$

- L'elemento neutro del prodotto matrice-matrice è la matrice identità:

$$AI_n = I_m A.$$

- Vale che $(AB)^T = B^T A^T$.

Definizione (prodotto tra matrice e vettore)

Data una matrice $A \in \mathbb{R}^{m \times n}$ e un vettore $x \in \mathbb{R}^n$, il **prodotto** Ax è il vettore colonna $y = (y_i) \in \mathbb{R}^m$ le cui componenti sono definite dalla formula seguente

$$y_i = \sum_{k=1}^n a_{ik}x_k, \quad i = 1, \dots, m.$$

N.B. E' un caso particolare del prodotto matrice-matrice, in cui la seconda matrice è un vettore colonna.

• Algoritmo del prodotto matrice-vettore

```
FOR  $i = 1, \dots, m$ 
   $s \leftarrow 0$ 
  FOR  $k = 1, \dots, n$ 
     $s \leftarrow s + a_{ik}x_k$ 
   $y_i \leftarrow s$ 
```

Complessità computazionale:

- per ogni elemento occorrono n somme e prodotti; siccome gli elementi sono in tutto m , la complessità ammonterà a mn ;
- se la matrice è quadrata, la complessità è pari a n^2 .

Definizione (prodotto scalare tra vettori)

Dati due vettori $u, v \in \mathbb{R}^n$, si definisce **prodotto scalare tra u e v** il numero s dato da

$$s = u^T v = \sum_{k=1}^n u_k v_k.$$

- **Algoritmo del prodotto scalare tra vettori**

```
s ← 0
FOR k = 1, ..., n
  | s ← s + ukvk
```

Complessità computazionale: n prodotti, n somme.

- Le operazioni vettoriali e matriciali che abbiamo visto finora, hanno una corrispondente implementazione ottimizzata all'interno della **libreria BLAS**

`http://www.netlib.org/blas/`

- Molti software di calcolo scientifico che richiedono la manipolazione di matrici e vettori (incluso MATLAB) utilizzano questa libreria per le operazioni fondamentali.

Definizione (inversa di una matrice)

Una matrice quadrata di ordine n si dice **invertibile** o **non singolare** se esiste una matrice $X \in \mathbb{R}^{n \times n}$ tale che

$$AX = XA = I.$$

La matrice X viene detta **inversa di A** e si indica con $X = A^{-1}$.

• Proprietà dell'inversa

- $(A^{-1})^{-1} = A$.
- $(A^T)^{-1} = (A^{-1})^T = A^{-T}$.
- Se A e B sono invertibili, anche AB è invertibile e

$$(AB)^{-1} = B^{-1}A^{-1}.$$

Definizione (determinante di una matrice)

Data $A \in \mathbb{R}^{n \times n}$, il **determinante di A** è il numero denotato con $\det(A)$ definito in modo ricorsivo come

$$\det(A) = \begin{cases} a_{11}, & \text{se } n = 1 \\ \sum_{j=1}^n (-1)^{i+j} a_{ij} \det(A_{ij}), & \text{se } n > 1 \end{cases}$$

dove

- i è un qualsiasi indice di riga, ovvero $i \in \{1, \dots, n\}$
- A_{ij} è la matrice di ordine $n - 1$ ottenuta eliminando la riga i e la colonna j di A .

- **Complessità computazionale**

Il determinante di una matrice di ordine n richiede n somme, n prodotti e il calcolo di n determinanti di ordine $n - 1$, ciascuno dei quali richiede $n - 1$ somme, $n - 1$ prodotti e il calcolo di $n - 1$ determinanti di ordine $n - 2, \dots$
In totale la complessità del calcolo del determinante tramite la definizione è dell'ordine di $n!$

Proprietà del determinante

- $\det(A) = \det(A^T)$
- $\det(AB) = \det(A) \det(B)$ (Formula di Binet)
- $\det(A^{-1}) = \frac{1}{\det(A)}$
- $\det(\lambda A) = \lambda^n \det(A)$ dove n è l'ordine della matrice
- Se B è una matrice ottenuta scambiando due righe (o colonne) di A , allora $\det(B) = -\det(A)$.
- Una matrice è non singolare se e solo se $\det(A) \neq 0$.

Definizione (autovalore e autovettore di una matrice)

Data $A \in \mathbb{R}^{n \times n}$, uno scalare $\lambda \in \mathbb{C}$ si dice **un autovalore di A** se esiste un vettore $x \in \mathbb{C}^n$, $x \neq 0$, tale che valga la seguente uguaglianza

$$Ax = \lambda x.$$

Il vettore x si dice **autovettore relativo all'autovalore λ** .

- Gli n autovalori di A sono tutte e sole le soluzioni dell'equazione

$$\det(A - \lambda I) = 0.$$

La funzione $p(\lambda) = \det(A - \lambda I)$ è un polinomio di grado n nella variabile λ detto **polinomio caratteristico di A** .

Proprietà

- Se $\lambda_1, \dots, \lambda_n$ sono gli n autovalori di A , allora

$$\det(A) = \lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_n.$$

- Una matrice è non singolare se e solo se tutti i suoi autovalori sono non nulli.
- Se A è non singolare e λ un autovalore di A , allora $1/\lambda$ è un autovalore di A^{-1} .
- Se A è simmetrica, allora tutti i suoi autovalori sono reali.

Definizione (raggio spettrale)

Sia A una matrice quadrata di ordine n e siano $\lambda_1, \dots, \lambda_n$ i suoi autovalori. Si definisce **raggio spettrale di A** il numero reale positivo indicato con $\rho(A)$ e definito nel modo seguente

$$\rho(A) = \max_{i=1, \dots, n} |\lambda_i|.$$

Definizione

Si dice **norma vettoriale** una funzione $\| \cdot \| : \mathbb{R}^n \rightarrow \mathbb{R}$ che soddisfa le seguenti proprietà:

- 1 $\|x\| \geq 0, \forall x \in \mathbb{R}^n$
 $\|x\| = 0 \Leftrightarrow x = 0 \Leftrightarrow x_i = 0, i = 1, \dots, n.$
- 2 $\|\lambda x\| = |\lambda| \|x\|, \forall x \in \mathbb{R}^n, \forall \lambda \in \mathbb{R}.$
- 3 $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n$ (**disuguaglianza triangolare**).

- I tre assiomi che definiscono una norma vettoriale sono tre proprietà naturali quando si vuole misurare una lunghezza di un vettore nello spazio Euclideo.
 \Rightarrow **generalizzazione delle proprietà della norma Euclidea**
- Oltre alla norma Euclidea, esistono altri esempi di norme (prossima slide).

Norma 1

$$\|x\|_1 = \sum_{i=1}^n |x_i|.$$

Norma 2 (o norma euclidea)

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}.$$

Norma p (con $p \geq 1$)

$$\|x\|_p = \left(\sum_{i=1}^p |x_i|^p \right)^{\frac{1}{p}}.$$

Norma ∞ (norma infinito)

$$\|x\|_\infty = \max_{i=1, \dots, n} |x_i|.$$

- Una norma vettoriale associa ad ogni vettore un numero che esprime la sua misura e viene calcolata tenendo conto di tutte le sue componenti.
- E' un'informazione utile per poter confrontare tra loro i vettori.
- Per la norma Euclidea vale la seguente uguaglianza:

$$\|x\|_2^2 = x_1^2 + x_2^2 + \dots + x_n^2 = x^T x.$$

- Le norme vettoriali $1, 2, \infty$ sono **equivalenti**, nel senso che per ogni $x \in \mathbb{R}^n$

$$\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty$$

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$$

$$\frac{\|x\|_1}{\sqrt{n}} \leq \|x\|_2 \leq \|x\|_1.$$

Quindi, per n “contenuto”, le tre norme hanno lo stesso ordine di grandezza.

- Tuttavia, le norme non hanno lo stesso costo: la norma 2 è la più costosa (n prodotti, $n - 1$ somme, una radice quadrata), mentre le norme 1 e ∞ richiedono rispettivamente $n - 1$ somme e $n - 1$ confronti.

Definizione

Si dice **norma matriciale** una funzione $\| \cdot \| : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ che soddisfa le seguenti proprietà:

- 1 $\|A\| \geq 0, \forall A \in \mathbb{R}^{n \times n}$
 $\|A\| = 0 \Leftrightarrow A = 0 \Leftrightarrow a_{ij} = 0, i, j = 1, \dots, n.$
- 2 $\|\lambda A\| = |\lambda| \|A\|, \forall A \in \mathbb{R}^{n \times n}, \forall \lambda \in \mathbb{R}.$
- 3 $\|A + B\| \leq \|A\| + \|B\|, \forall A, B \in \mathbb{R}^{n \times n}$ (**disuguaglianza triangolare**).

A partire da una norma vettoriale $\|\cdot\|$, è possibile definire formalmente una **norma matriciale indotta** nel modo seguente

$$\|A\| = \sup_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Si può dimostrare che le norme indotte dalle norme vettoriali 1, 2, ∞ sono:

Norma 1

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^n |a_{ij}|.$$

Norma 2 (o norma Euclidea)

$$\|A\|_2 = \sqrt{\rho(A^T A)}.$$

Norma ∞

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^n |a_{ij}|.$$

Sia $\bullet \in \{1, 2, \infty\}$. Valgono le seguenti proprietà.

Proprietà submoltiplicativa

Siano $A, B \in \mathbb{R}^{n \times n}$. Allora

$$\|AB\|_{\bullet} \leq \|A\|_{\bullet} \|B\|_{\bullet}.$$

Compatibilità con le norme vettoriali

Siano $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$. Allora

$$\|Ax\|_{\bullet} \leq \|A\|_{\bullet} \|x\|_{\bullet}.$$

- Le norme esprimono l'idea di una misura.
- Hanno lo stesso ruolo che ha il valore assoluto per gli scalari.
- Se $x, y \in \mathbb{R}^n$ il numero $\|x - y\|$ esprime il concetto della distanza tra x e y .
- Per valutare gli errori tra vettori si utilizzano le norme: in particolare la quantità

$$E_r = \frac{\|x - y\|}{\|x\|}.$$

è l'errore relativo (la distanza relativa) tra x e y .

- Le stesse considerazioni valgono anche per le matrici.

2. Sistemi lineari e loro condizionamento

Data una matrice $A \in \mathbb{R}^{n \times n}$, detta matrice dei coefficienti, dato un vettore $b \in \mathbb{R}^n$, detto vettore dei termini noti, ovvero

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix},$$

si vuole calcolare il vettore $x \in \mathbb{R}^n$, $x = (x_1 \ x_2 \ \cdots \ x_n)^T$, che soddisfa l'uguaglianza

$$Ax = b,$$

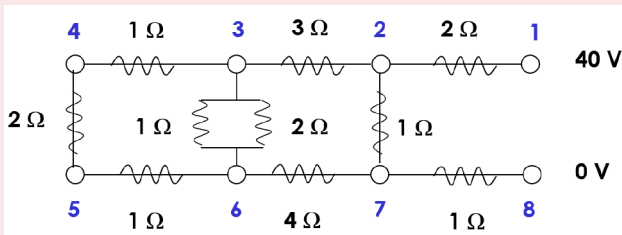
il che equivale a risolvere le seguenti n equazioni lineari nelle incognite x_1, x_2, \dots, x_n

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

L'uguaglianza $Ax = b$ prende il nome di **sistema lineare**.

Problema

Sia dato il circuito in figura.



Noti i potenziali nei nodi 1 e 8 e le resistenze, calcolare i potenziali nei nodi 2 – 7.

- R_{ij} e I_{ij} : resistenza e corrente tra il nodo i e il nodo j ;
- V_i : potenziale nel nodo i ;

- Legge di Ohm:

$$\frac{V_i - V_j}{R_{ij}} = I_{ij}.$$

- Resistenze in parallelo:

$$R_{36} = \frac{1}{1 + \frac{1}{2}} = \frac{2}{3} \Omega.$$

- Legge di Kirchhoff: la somma delle correnti in ciascun nodo è nulla

$$\begin{array}{llll} \text{nodo 2 :} & I_{12} + I_{72} + I_{32} & = & 0 \\ \text{nodo 3 :} & I_{23} + I_{43} + I_{63} & = & 0 \\ \text{nodo 4 :} & I_{34} + I_{54} & = & 0 \\ \text{nodo 5 :} & I_{45} + I_{65} & = & 0 \\ \text{nodo 6 :} & I_{56} + I_{76} + I_{36} & = & 0 \\ \text{nodo 7 :} & I_{67} + I_{27} + I_{87} & = & 0 \end{array} \Leftrightarrow \begin{cases} \frac{(V_1 - V_2)}{2} + \frac{(V_7 - V_2)}{1} + \frac{(V_3 - V_2)}{3} = 0 \\ \frac{(V_2 - V_3)}{3} + \frac{(V_6 - V_3)}{\frac{2}{3}} + \frac{(V_4 - V_3)}{3} = 0 \\ \frac{(V_3 - V_4)}{1} + \frac{(V_5 - V_4)}{\frac{2}{3}} = 0 \\ \frac{(V_5 - V_6)}{1} + \frac{(V_3 - V_6)}{\frac{2}{3}} + \frac{(V_7 - V_6)}{4} = 0 \\ \frac{(V_6 - V_7)}{4} + \frac{(V_2 - V_7)}{1} + \frac{(V_8 - V_7)}{1} = 0 \end{cases}$$

$$\begin{cases} (V_1 - V_2)/2 + (V_7 - V_2)/1 + (V_3 - V_2)/3 = 0 \\ (V_2 - V_3)/3 + (V_6 - V_3)/(2/3) + (V_4 - V_3)/3 = 0 \\ (V_3 - V_4)/1 + (V_5 - V_4)/2 = 0 \\ (V_5 - V_6)/1 + (V_3 - V_6)/(2/3) + (V_7 - V_6)/4 = 0 \\ (V_6 - V_7)/4 + (V_2 - V_7)/1 + (V_8 - V_7)/1 = 0 \end{cases}$$

Siccome $V_1 = 40$ e $V_8 = 0$, il sistema si può riscrivere come

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

ovvero come un sistema $Ax = b$ di 6 equazioni nelle 6 incognite V_2, \dots, V_7 con

$$A = \begin{pmatrix} 11 & -2 & 0 & 0 & 0 & -6 \\ -2 & 12 & -2 & 0 & 0 & 0 \\ 0 & -2 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & -6 & 0 & -4 & 11 & -1 \\ -4 & 0 & 0 & 0 & -1 & 9 \end{pmatrix}, \quad b = \begin{pmatrix} 120 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Definizione

Una matrice quadrata di ordine n si dice **invertibile** o **non singolare** se esiste una matrice $X \in \mathbb{R}^{n \times n}$ tale che $AX = XA = I$.

La matrice X viene detta **inversa di A** e si indica con $X = A^{-1}$.

Proposizione

Le seguenti proprietà sono equivalenti.

- A è non singolare.
- $\det(A) \neq 0$.
- Le righe e le colonne di A formano un insieme di vettori linearmente indipendenti.
- $Ax = 0$ se e solo se $x = 0$.

Teorema

Se A è non singolare, allora esiste un'unica soluzione del sistema lineare $Ax = b$. Tale soluzione è $\bar{x} = A^{-1}b$.

Esempio

Sia dato il sistema lineare $Ax = b$ dove

$$A = \begin{pmatrix} 835 & 667 \\ 333 & 266 \end{pmatrix}, \quad b = \begin{pmatrix} 168 \\ 67 \end{pmatrix}.$$

Consideriamo inoltre il dato perturbato $b + \Delta b$ definito come

$$b + \Delta b = \begin{pmatrix} 168 \\ 66 \end{pmatrix}.$$

Come cambia la soluzione del sistema $Ax = b + \Delta b$ rispetto a quella di $Ax = b$?

- Notiamo che l'errore sul vettore dei dati è

$$\frac{\|b - (b + \Delta b)\|_{\infty}}{\|b\|_{\infty}} = \frac{\|(0 \ 1)^T\|_{\infty}}{\|(168 \ 67)^T\|_{\infty}} = \frac{1}{168} \simeq 6 \cdot 10^{-3}.$$

- Le soluzioni di $Ax = b$ e $Ax = b + \Delta b$ sono rispettivamente

$$x = A^{-1}b = (1 \ -1)^T, \quad x + \Delta x = A^{-1}(b + \Delta b) = (-666 \ 834)^T.$$

Esempio

Sia dato il sistema lineare $Ax = b$ dove

$$A = \begin{pmatrix} 835 & 667 \\ 333 & 266 \end{pmatrix}, \quad b = \begin{pmatrix} 168 \\ 67 \end{pmatrix}.$$

Consideriamo inoltre il dato perturbato $b + \Delta b$ definito come

$$b + \Delta b = \begin{pmatrix} 168 \\ 66 \end{pmatrix}.$$

Come cambia la soluzione del sistema $Ax = b + \Delta b$ rispetto a quella di $Ax = b$?

- Dunque l'errore sulla soluzione è pari a

$$\frac{\|x - (x + \Delta x)\|_{\infty}}{\|x\|_{\infty}} = \frac{\|(667 - 835)^T\|_{\infty}}{\|(1 \ -1)^T\|_{\infty}} = 835.$$

In altre parole: la soluzione del sistema perturbato $Ax = b + \Delta b$ non ha nessuna cifra significativa in comune con quella del sistema $Ax = b$.

Esempio

Sia dato il sistema lineare $Ax = b$ dove

$$A = \begin{pmatrix} 835 & 667 \\ 333 & 266 \end{pmatrix}, \quad b = \begin{pmatrix} 168 \\ 67 \end{pmatrix}.$$

Consideriamo inoltre il dato perturbato $b + \Delta b$ definito come

$$b + \Delta b = \begin{pmatrix} 168 \\ 66 \end{pmatrix}.$$

Come cambia la soluzione del sistema $Ax = b + \Delta b$ rispetto a quella di $Ax = b$?

- In conclusione **il sistema è mal condizionato**, in quanto un errore “piccolo” sui dati ha provocato una “grande” variazione sulla soluzione del sistema, con un fattore di amplificazione dell'errore pari a 10^5 :

$$\frac{835}{6 \cdot 10^{-3}} \simeq 139000 \simeq 10^5.$$

Esempio

Sia dato il sistema lineare

$$\begin{cases} x_1 + 2x_2 & = 3 \\ .499x_1 + 1.001x_2 & = 1.5 \end{cases}.$$

Si consideri il seguente sistema ottenuto perturbando la matrice dei coefficienti:

$$\begin{cases} x_1 + 2x_2 & = 3 \\ .5x_1 + 1.002x_2 & = 1.5 \end{cases}.$$

Come cambia la soluzione del sistema $(A + \Delta A)x = b$ rispetto a quella di $Ax = b$?

- L'errore sulla matrice è dell'ordine di 10^{-3} .
- La soluzione esatta del sistema non perturbato è $(1 \ 1)^T$, mentre quella del sistema perturbato è $(3 \ 0)^T$
 \Rightarrow l'errore sul risultato è dell'ordine di 10^0
 \Rightarrow **il sistema è mal condizionato.**

- In entrambi gli esempi, le righe della matrice dei coefficienti sono “quasi” **linearmente dipendenti**, ossia “quasi” l’una il multiplo dell’altra.
- Ad esempio nel sistema

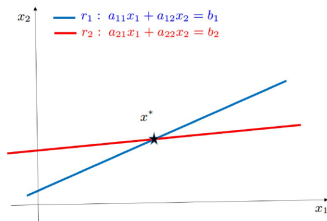
$$\begin{cases} x_1 + 2x_2 & = 3 \\ .499x_1 + 1.001x_2 & = 1.5 \end{cases}$$

la seconda riga è “quasi” la metà della prima.

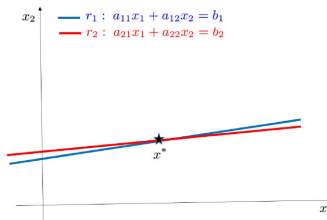
- In generale, un sistema lineare sarà mal condizionato quando la matrice è vicina ad essere non invertibile (singolare), senza realmente esserlo.

Condizionamento di un sistema lineare (interpretazione grafica)

Se $n = 2$, la soluzione di un sistema consiste nel punto x^* in cui si intersecano le rette r_1 ed r_2 di equazione $a_{11}x_1 + a_{12}x_2 = b_1$ e $a_{21}x_1 + a_{22}x_2 = b_2$ rispettivamente.

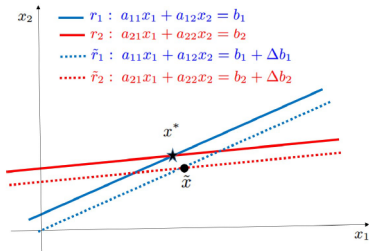


Il caso di malcondizionamento si ha quando r_1 ed r_2 sono “quasi” coincidenti.

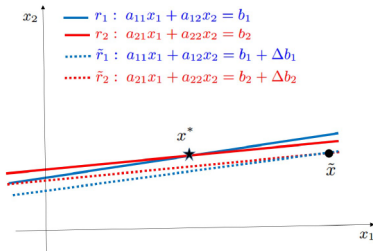


Supponiamo per semplicità di perturbare soltanto il termine noto. Indichiamo con x^* la soluzione di $Ax = b$ e con \tilde{x} la soluzione di $Ax = b + \Delta b$.

Sistema ben condizionato



Sistema mal condizionato



Ipotesi: sia x^* la soluzione di $Ax = b$ e \tilde{x} la soluzione di $Ax = b + \Delta b$.

Obiettivo: ricavare una relazione tra l'errore relativo sulla soluzione e l'errore relativo sui dati di un sistema lineare:

$$\frac{\|x^* - \tilde{x}\|}{\|x^*\|} \leftrightarrow \frac{\|\Delta b\|}{\|b\|}$$

dove compaia un parametro adatto a descrivere il condizionamento del sistema, ossia l'amplificazione dell'errore sulla sua soluzione.

Sia x^* la soluzione del sistema “esatto” e \tilde{x} la soluzione del sistema “perturbato”:

$$Ax^* = b, \quad A\tilde{x} = b + \Delta b.$$

- Osservazione 1

$$A\tilde{x} = b + \Delta b = Ax^* + \Delta b \Leftrightarrow A(\tilde{x} - x^*) = \Delta b \Leftrightarrow \|\tilde{x} - x^*\| = \|A^{-1}\Delta b\|.$$

- Osservazione 2

$$b = Ax^* \Rightarrow \|b\| = \|Ax^*\| \xrightarrow{\text{norma indotta}} \|b\| \leq \|A\| \cdot \|x^*\| \Rightarrow \frac{1}{\|x^*\|} \leq \frac{\|A\|}{\|b\|}.$$

Combinando le due osservazioni, otteniamo che

$$\begin{aligned} \frac{\|\tilde{x} - x^*\|}{\|x^*\|} &\stackrel{\text{Osservaz. 1}}{=} \frac{\|A^{-1}\Delta b\|}{\|x^*\|} \\ &\stackrel{\text{norma indotta}}{\leq} \frac{1}{\|x^*\|} \|A^{-1}\| \cdot \|\Delta b\| \\ &\stackrel{\text{Osservaz. 2}}{\leq} \|A\| \|A^{-1}\| \frac{\|\Delta b\|}{\|b\|}. \end{aligned}$$

Il numero $\kappa(A) = \|A\| \|A^{-1}\|$ è detto **numero di condizionamento della matrice A** .

Abbiamo dimostrato che

$$\frac{\|\tilde{x} - x^*\|}{\|x^*\|} \leq \underbrace{\|A\| \|A^{-1}\|}_{=\kappa(A)} \frac{\|\Delta b\|}{\|b\|}.$$

- La quantità $\kappa(A) \|\Delta b\| / \|b\|$ rappresenta **una stima dell'errore sulla soluzione**:

$$\underbrace{\frac{\|\tilde{x} - x^*\|}{\|x^*\|}}_{\text{non calcolabile (} x^* \text{ non è nota)}} \simeq \underbrace{\kappa(A) \frac{\|\Delta b\|}{\|b\|}}_{\text{calcolabile (se noto l'errore sui dati)}}$$

- Il numero di condizionamento $\kappa(A)$ di A agisce come un fattore di amplificazione tra l'errore sui dati e l'errore sulla soluzione.
- Per ogni matrice non singolare, si ha

$$1 = \|I\| = \|AA^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \kappa(A) \quad \Leftrightarrow \quad \kappa(A) \geq 1.$$

Se $\kappa(A) \simeq 1$, il sistema $Ax = b$ è ben condizionato.

Se $\kappa(A) \gg 1$, allora il sistema $Ax = b$ è mal condizionato.

Esempio

Sia dato il sistema $Ax = b$ dell'esempio precedente, dove

$$A = \begin{pmatrix} 1 & 2 \\ 0.499 & 1.001 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}.$$

- Notiamo che $\det(A) = 3 \cdot 10^{-3}$ e

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

Dunque $\|A\|_{\infty} = 3$, $\|A^{-1}\|_{\infty} = \frac{1}{\det(A)} \cdot 3.001$ e

$$\kappa(A) = \|A\|_{\infty} \|A^{-1}\|_{\infty} = 3.001 \cdot 10^3.$$

Esempio

Sia dato il sistema $Ax = b$ dell'esempio precedente, dove

$$A = \begin{pmatrix} 1 & 2 \\ 0.499 & 1.001 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}.$$

- Sia $b + \Delta b$ il dato perturbato, ovvero

$$b + \Delta b = \begin{pmatrix} 3 \\ 1.4985 \end{pmatrix} \Rightarrow \Delta b = \begin{pmatrix} 0 \\ -0.0015 \end{pmatrix}.$$

Si ha $Ax^* = b$ e $A\tilde{x} = b + \Delta b$ con

$$x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}.$$

Quindi l'errore sulla soluzione e la sua stima sono rispettivamente

$$\frac{\|x^* - \tilde{x}\|_\infty}{\|x^*\|_\infty} = 1 \quad \leftrightarrow \quad \kappa(A) \frac{\|\Delta b\|_\infty}{\|b\|_\infty} = (3.001 \cdot 10^3) \cdot (5 \cdot 10^{-4}) \simeq 1.5.$$

Si può provare un risultato più generale sul condizionamento dei sistemi, nel caso più realistico in cui siano presenti perturbazioni anche sulla matrice dei coefficienti.

Proposizione

Supponiamo che x^* sia la soluzione del sistema non perturbato, mentre \tilde{x} è la soluzione del sistema in cui sia la matrice dei coefficienti che il vettore dei termini noti sono stati perturbati, ovvero

$$Ax^* = b, \quad (A + \Delta A)\tilde{x} = b + \Delta b,$$

dove ΔA e Δb sono rispettivamente le perturbazioni su A e b .
Indichiamo inoltre con e_A ed e_b gli errori relativi presenti nei dati

$$e_A = \frac{\|\Delta A\|}{\|A\|}, \quad e_b = \frac{\|\Delta b\|}{\|b\|}.$$

Allora si può dimostrare (sotto alcune ipotesi) che l'errore relativo sulla soluzione (l'errore relativo inerente) soddisfa la seguente uguaglianza

$$\frac{\|x^* - \tilde{x}\|}{\|x^*\|} \leq \frac{\kappa(A)}{1 - \kappa(A)e_A} (e_A + e_b).$$

3. Metodi diretti per sistemi lineari

Distinguiamo due classi di metodi per la risoluzione numerica dei sistemi lineari.

1. **METODI DIRETTI**: calcolano la soluzione in un numero finito di operazioni.
2. **METODI ITERATIVI**: definiscono una successione infinita di vettori che al limite tendono alla soluzione del sistema.

Ciascun metodo verrà presentato, valutato dal punto di vista della complessità computazionale e stabilità, ed implementato come algoritmo su macchina mediante linguaggio MATLAB.

Consideriamo il sistema $Dx = b$, dove la matrice dei coefficienti è diagonale:

$$D = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}, \quad \det(D) \neq 0.$$

In forma esplicita:

$$d_i x_i = b_i, \quad i = 1, \dots, n.$$

Siccome $\det(D) = d_1 \cdot d_2 \cdot \dots \cdot d_n$, segue che $\det(D) \neq 0$ se e solo se $d_i \neq 0$ per $i = 1, \dots, n$. Dunque l'unico vettore soluzione è dato da $x = (x_1 \ x_2 \ \dots \ x_n)^T$ dove

$$x_i = \frac{b_i}{d_i}, \quad i = 1, \dots, n.$$

Costo computazionale: n quozienti.

2) Metodi diretti: sistemi triangolari inferiori

Consideriamo il sistema $Lx = b$, dove la matrice dei coefficienti è data da

$$L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{n1} & & & l_{nn} \end{pmatrix}, \quad \det(L) \neq 0.$$

Siccome $\det(L) = l_{11}l_{22} \dots l_{nn}$, si ha $\det(L) \neq 0$ se e solo se $l_{ii} \neq 0, i = 1, \dots, n$.

Metodo di sostituzione in avanti

$$\begin{cases} l_{11}x_1 = b_1 \\ l_{21}x_1 + l_{22}x_2 = b_2 \\ \vdots \\ l_{n1}x_1 + l_{n2}x_2 + \dots + l_{nn}x_n = b_n \end{cases} \Rightarrow \begin{cases} x_1 = \frac{b_1}{l_{11}} \\ x_2 = \frac{b_2 - l_{21}x_1}{l_{22}} \\ \vdots \\ x_n = \frac{b_n - \sum_{k=1}^{n-1} l_{nk}x_k}{l_{nn}} \end{cases}$$

$$x_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik}x_k}{l_{ii}}, \quad i = 1, \dots, n.$$

2) Metodi diretti: sistemi triangolari inferiori

Algoritmo basato sul metodo di sostituzione in avanti

```
 $x_1 \leftarrow b_1/l_{11}$   
FOR  $i = 2, \dots, n$   
   $s \leftarrow 0$   
  FOR  $k = 1, \dots, i - 1$   
     $s \leftarrow s + l_{ik}x_k$   
   $x_i \leftarrow (b_i - s)/l_{ii}$ 
```

Complessità computazionale

- Passo i :
 $i - 1$ somme, 1 sottrazione, $i - 1$ prodotti, 1 divisione
 \Rightarrow **$2i$ operazioni**
- Costo totale:

$$\begin{aligned} 1 + \sum_{i=2}^n 2i &= 1 + 2 \sum_{i=2}^n i \pm 2 \\ &= 2 \sum_{i=1}^n i - 1 = 2 \frac{n(n+1)}{2} - 1 = n^2 + n - 1 \end{aligned}$$

avendo usato la formula di Gauss $\sum_{i=1}^n i = \frac{n(n+1)}{2}$.

\Rightarrow **$\mathcal{O}(n^2)$ operazioni.**

Stabilità

L'algoritmo di sostituzione in avanti può diventare instabile quando gli elementi del triangolo inferiore sono molto grandi rispetto agli elementi diagonali.

- Ad esempio, si consideri il seguente sistema 2×2 triangolare inferiore:

$$\begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Indichiamo con $x^* = (x_1^* \ x_2^*)^T$ la soluzione esatta e con $\tilde{x} = (\tilde{x}_1 \ \tilde{x}_2)^T$ la soluzione calcolata in aritmetica finita con l'algoritmo di sostituzione.

Al primo passo dell'algoritmo si ha

$$\tilde{x}_1 = fl(b_1/l_{11}) = \frac{b_1}{l_{11}}(1 + \epsilon_1) = x_1^*(1 + \epsilon_1), \quad |\epsilon_1| \leq u.$$

Supponendo per semplicità che al secondo passo non vengano introdotti ulteriori errori, risulta che

$$\tilde{x}_2 = \frac{b_2 - l_{21}\tilde{x}_1}{l_{22}} = \frac{b_2 - l_{21}x_1^*(1 + \epsilon_1)}{l_{22}} = \frac{b_2 - l_{21}x_1^*}{l_{22}} - \frac{l_{21}x_1^*}{l_{22}}\epsilon_1 = x_2^* - \frac{l_{21}x_1^*}{l_{22}}\epsilon_1.$$

Se $|l_{21}| \gg |l_{22}|/|x_1^*|$ allora l'algoritmo diventa instabile.

2) Metodi diretti: sistemi triangolari superiori

Consideriamo il sistema $Ux = b$, dove la matrice dei coefficienti è data da

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{pmatrix}, \quad \det(U) \neq 0.$$

Siccome $\det(U) = u_{11}u_{22} \dots u_{nn}$, $\det(U) \neq 0$ se e solo se $u_{ii} \neq 0$, $i = 1, \dots, n$.

Metodo di sostituzione all'indietro

$$\begin{cases} u_{nn}x_n = b_n & \Rightarrow x_n = \frac{b_n}{u_{nn}} \\ u_{n-1n-1}x_{n-1} + u_{n-1n}x_n = b_{n-1} & \Rightarrow x_{n-1} = \frac{b_{n-1} - u_{n-1n}x_n}{u_{n-1n-1}} \\ \vdots & \\ u_{11}x_1 + u_{12}x_2 + \dots + u_{1n}x_n = b_1 & \Rightarrow x_1 = \frac{b_1 - \sum_{k=2}^n u_{1k}x_k}{u_{11}} \end{cases}$$

$$x_i = \frac{b_i - \sum_{k=i+1}^n u_{ik}x_k}{u_{ii}}, \quad i = n, n-1, \dots, 1.$$

Algoritmo basato sul metodo di sostituzione all'indietro

```

$$x_n \leftarrow b_n / u_{nn}$$

$$\text{FOR } i = n - 1, \dots, 1$$

$$\left| \begin{array}{l} s \leftarrow 0 \\ \text{FOR } k = i + 1, \dots, n \\ \quad | \quad s \leftarrow s + u_{ik} x_k \\ x_i \leftarrow (b_i - s) / u_{ii} \end{array} \right.$$

```

Complessità computazionale

$\mathcal{O}(n^2)$ operazioni.

Stabilità

L'algoritmo di sostituzione all'indietro può diventare instabile quando gli elementi del triangolo superiore sono molto grandi rispetto agli elementi diagonali.

3) Metodi diretti: caso generale

Consideriamo il sistema

$$Ax = b, \quad \det(A) \neq 0,$$

dove $A \in \mathbb{R}^{n \times n}$ non ha una particolare struttura.

Metodo di Cramer

Si calcola il vettore soluzione $x = (x_1 \ x_2 \ \cdots \ x_n)^T$ come

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, \dots, n$$

ove A_i è la matrice ottenuta sostituendo b alla i -esima colonna di A .

Occorre calcolare $n + 1$ determinanti mediante la regola di Laplace.

Se si usa tale regola, il costo di ogni determinante è $n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1$.

Il costo totale sarà dunque di $(n + 1)n! = (n + 1)!$ operazioni.

- Ad esempio se volessimo calcolare la soluzione di un sistema di dimensione 25 con il metodo di Cramer, il numero totale delle operazioni sarebbe pari a $(n + 1)! = 26! \simeq 4 \cdot 10^{26}$. Utilizzando il supercomputer IBM Summit avente potenza di calcolo di 200 petaFLOPS (petaFLOPS = 10^{15} operazioni Floating Point), servirebbero 63 anni e mezzo di tempo per ottenere la soluzione.

⇒ **Impraticabile!**

3) Metodi diretti: caso generale

Consideriamo il sistema

$$Ax = b, \quad \det(A) \neq 0,$$

dove $A \in \mathbb{R}^{n \times n}$ non ha una particolare struttura.

Calcolo dell'inversa

Si calcola il vettore soluzione $x = (x_1 \ x_2 \ \cdots \ x_n)^T$ come

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

Dal punto di vista numerico, questa non è una buona idea per almeno due motivi:

1. Il calcolo di A^{-1} è computazionalmente costoso.

Infatti richiede la risoluzione di n sistemi lineari:

$$AA^{-1} = I \quad \Leftrightarrow \quad \begin{cases} Ac_1 = e_1 \\ Ac_2 = e_2 \\ \vdots \\ Ac_n = e_n \end{cases}, \quad c_i = \text{i-esima colonna}.$$

Dunque, per risolvere un sistema, ne risolvo n .

3) Metodi diretti: caso generale

Consideriamo il sistema

$$Ax = b, \quad \det(A) \neq 0,$$

dove $A \in \mathbb{R}^{n \times n}$ non ha una particolare struttura.

Calcolo dell'inversa

Si calcola il vettore soluzione $x = (x_1 \ x_2 \ \cdots \ x_n)^T$ come

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

Dal punto di vista numerico, questa non è una buona idea per almeno due motivi:

2. L'algoritmo dell'inversa è poco stabile

Infatti, consideriamo il seguente esempio

$$7x = 21.$$

Usando un calcolatore con $\beta = 10$, $t = 4$ e troncamento, l'algoritmo esegue

$$fl(1/7) = fl(0.142857142857143) = 0.1428$$

$$fl(fl(1/7) \cdot 21) = fl(2.9988) = 2.998.$$

Si è commesso un errore dell'ordine di 10^{-4} (sulla quarta cifra significativa), mentre calcolando $fl(21/7) = 3$ si sarebbe ottenuto il risultato esatto.

Metodi di fattorizzazione

Sono i metodi di preferenza per la risoluzione numerica dei sistemi lineari.

L'idea è quella di **fattorizzare la matrice A nel prodotto di due matrici semplici**, in modo che sia facile risolvere i due sistemi associati.

Studieremo due tipi di fattorizzazione:

- **Fattorizzazione LU (o di Gauss)**

Si fattorizza la matrice A nel prodotto di una matrice triangolare inferiore per una triangolare superiore

$$A = LU.$$

- **Fattorizzazione QR**

Si fattorizza la matrice A nel prodotto di una matrice ortogonale (per cui $Q^{-1} = Q^T$) con una matrice triangolare superiore

$$A = QR.$$

3) Metodi diretti: metodo di Gauss

$$Ax = b, \quad \det(A) \neq 0.$$

Il metodo di Gauss per la risoluzione di un sistema lineare si divide in due fasi.

Fase 1: procedimento di eliminazione (o fattorizzazione) di Gauss

Si calcola una matrice triangolare inferiore L e una matrice triangolare superiore U tali che

$$A = LU.$$

Fase 2: risoluzione del sistema

Dall'uguaglianza $A = LU$ il sistema si riscrive come

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

Il sistema $Ax = b$ viene risolto in due passi.

- Risoluzione del sistema triangolare inferiore $Ly = b$ per sostituzione in avanti
- Risoluzione del sistema triangolare superiore $Ux = y$ per sostituzione all'indietro.

Fase 1: procedimento di eliminazione (o fattorizzazione) di Gauss

- I passi di eliminazione Gaussiana sono $n - 1$.
- Ponendo $A_1 = A$, ad ogni passo k si ottiene una nuova matrice A_{k+1} , $k = 1, \dots, n - 1$ mediante opportune combinazioni lineari delle righe di A_k , in modo che gli elementi delle colonne di A_{k+1} con indice da 1 a k che stanno al di sotto della diagonale principale siano nulli.

3) Metodi diretti: metodo di Gauss

Fase 1, primo passo

- Elemento perno (o pivot): a_{11} .
- Se $a_{11} = 0$, il metodo si arresta e termina senza successo.
- Altrimenti, $a_{11} \neq 0$ e si calcolano i **moltiplicatori di Gauss**

$$m_{i1} = \frac{a_{i1}}{a_{11}}, \quad i = 2, \dots, n.$$

- Per ogni $i = 2, \dots, n$, si sottrae alla riga i la riga 1 moltiplicata per m_{i1} .

$$A \equiv A_1 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} \Rightarrow A_2 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

$$a_{ij}^{(2)} = a_{ij} - m_{i1}a_{1j}, \quad i, j = 2, \dots, n.$$

$$A_2 = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & 0 & 1 & & \\ \vdots & & & \ddots & \\ -m_{n1} & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} = L_1 A_1.$$

3) Metodi diretti: metodo di Gauss

Fase 1, secondo passo

- Elemento perno (o pivot): $a_{22}^{(2)}$.
- Se $a_{22}^{(2)} = 0$, il metodo si arresta e termina senza successo.
- Altrimenti, $a_{22}^{(2)} \neq 0$ e si calcolano i **moltiplicatori di Gauss**

$$m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}, \quad i = 3, \dots, n.$$

- Per ogni $i = 3, \dots, n$, si sottrae alla riga i la riga 2 moltiplicata per m_{i2} .

$$A_2 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix} \Rightarrow A_3 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ 0 & 0 & \cdots & \cdots & \cdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{pmatrix}.$$
$$a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2} a_{2j}^{(2)}, \quad i, j = 3, \dots, n.$$

$$A_3 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -m_{32} & 1 & & \\ \vdots & & & \ddots & \\ 0 & -m_{n2} & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix} = L_2 A_2.$$

3) Metodi diretti: metodo di Gauss

Fase 1, k -esimo passo

- Elemento perno (o pivot): $a_{kk}^{(k)}$
- Se $a_{kk}^{(k)} = 0$, il metodo si arresta e termina senza successo.
- Altrimenti, $a_{kk}^{(k)} \neq 0$ e si calcolano i **moltiplicatori di Gauss**

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k + 1, \dots, n.$$

- Per ogni $i = k + 1, \dots, n$, si sottrae alla riga i la riga k moltiplicata per m_{ik} .

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

$$A_{k+1} = \begin{pmatrix} 1 & & & & & & \\ 0 & 1 & & & & & \\ 0 & 0 & 1 & & & & \\ \vdots & & & \ddots & & & \\ 0 & & & & 1 & & \\ & & & & -m_{k+1,k} & & 1 \\ & & & & \vdots & & \\ 0 & 0 & & & -m_{n,k} & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} & \cdots & a_{1n} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & a_{33}^{(3)} & \cdots & a_{3k}^{(k)} & \cdots & a_{3n}^{(3)} \\ & & & \ddots & & & \\ & & & & & & \\ & & & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}$$

$$A_{k+1} = L_k A_k.$$

3) Metodi diretti: metodo di Gauss

Dopo $n - 1$ passi ottengo

$$U = A_n = L_{n-1}A_{n-1} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} & \cdots & a_{1n} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & a_{33}^{(3)} & \cdots & a_{3k}^{(3)} & \cdots & a_{3n}^{(3)} \\ & & & \ddots & & & \\ & & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & & & \ddots & \\ & & & & & & a_{nn}^{(n)} \end{pmatrix}.$$

- U è la matrice triangolare ottenuta dopo $n - 1$ passi di eliminazione.
- Il procedimento di eliminazione equivale alla seguente successione di prodotti matriciali

$$U = L_{n-1}A_{n-1} = L_{n-1}L_{n-2}A_{n-2} = \dots = L_{n-1}L_{n-2} \dots L_2L_1A.$$

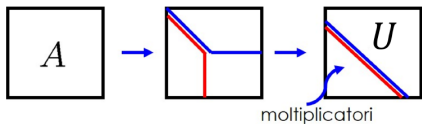
- La matrice L_k è detta **k -esima trasformazione elementare di Gauss** ed è definita in funzione dei moltiplicatori del k -esimo passo.
- Il metodo termina con successo se e solo se $a_{kk}^{(k)} \neq 0$ per $k = 1, \dots, n - 1$.

Algoritmo di eliminazione Gaussiana

Il metodo di Gauss è definito dalle seguenti regole di aggiornamento:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \underbrace{a_{kj}^{(k)} \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right)}_{m_{ik}}, \quad k = 1, \dots, n-1, \quad i, j = k+1, \dots, n.$$

Un'implementazione del metodo che ottimizza l'occupazione di memoria si ottiene sovrascrivendo gli elementi della matrice aggiornata e i moltiplicatori nelle stesse locazioni inizialmente occupate dagli elementi di A .



```
FOR  $k = 1, \dots, n-1$ 
  FOR  $i = k+1, \dots, n$ 
     $a_{ik} \leftarrow a_{ik} / a_{kk}$ 
    FOR  $j = k+1, \dots, n$ 
       $a_{ij} \leftarrow a_{ij} - a_{ik} a_{kj}$ 
```

Algoritmo di eliminazione Gaussiana

Dopo gli $n - 1$ passi dell'algoritmo di eliminazione Gaussiana, nelle locazioni in cui inizialmente erano memorizzati gli elementi di A si trova

a_{11}	a_{12}	a_{13}	\cdots	\cdots	a_{1n}
m_{21}	$a_{22}^{(2)}$	$a_{23}^{(2)}$	\cdots	\cdots	$a_{2n}^{(2)}$
m_{31}	m_{32}	$a_{33}^{(3)}$	\cdots	\cdots	$a_{3n}^{(3)}$
\vdots	\vdots	\vdots	\ddots	\cdots	\vdots
$m_{n-1\ 1}$	$m_{n-1\ 2}$	\cdots	\cdots	$a_{n-1\ n-1}^{(n-1)}$	$a_{n-1\ n}^{(n-1)}$
m_{n1}	m_{n2}	\cdots	\cdots	$m_{n\ n-1}$	$a_{nn}^{(n)}$

Complessità computazionale

```
FOR  $k = 1, \dots, n - 1$   
  | FOR  $i = k + 1, \dots, n$   
  |   |  $a_{ik} \leftarrow a_{ik} / a_{kk}$   
  |   | FOR  $j = k + 1, \dots, n$   
  |   |   |  $a_{ij} \leftarrow a_{ij} - a_{ik} a_{kj}$ 
```

- Passo k , calcolo a_{ij} , $i, j = k + 1, \dots, n$
Per i, j fissati, si esegue 1 sottrazione e 1 prodotto.
Siccome ho $n - k$ indici i e $n - k$ indici j , servono $2(n - k)^2$ operazioni.
- Passo k , calcolo a_{ik} , $i = k + 1, \dots, n$
Per i fissato, si esegue 1 divisione.
Dunque servono $n - k$ operazioni.

Complessità computazionale

```

FOR  $k = 1, \dots, n - 1$ 
  | FOR  $i = k + 1, \dots, n$ 
  | |  $a_{ik} \leftarrow a_{ik} / a_{kk}$ 
  | | FOR  $j = k + 1, \dots, n$ 
  | | |  $a_{ij} \leftarrow a_{ij} - a_{ik} a_{kj}$ 
    
```

- Costo totale

$$\begin{aligned}
 \sum_{k=1}^{n-1} 2(n-k)^2 + (n-k) &= 2 \sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-1} i \\
 &= 2 \left(\frac{(n-1)^3}{3} + \frac{(n-1)^2}{2} + \frac{n-1}{6} \right) + \frac{n(n-1)}{2}
 \end{aligned}$$

avendo usato $\sum_{i=1}^m i = \frac{m(m+1)}{2}$ e $\sum_{i=1}^m i^2 = \frac{m^3}{3} + \frac{m^2}{2} + \frac{m}{6}$ con $m = n - 1$.
 $\Rightarrow \mathcal{O}\left(\frac{2}{3}n^3\right)$ operazioni.

3) Metodi diretti: fattorizzazione LU

Teorema di fattorizzazione di Gauss

Se $a_{kk}^{(k)} \neq 0$, $k = 1, \dots, n-1$, si ha

$$A = LU,$$

dove $U \in \mathbb{R}^{n \times n}$ è una matrice triangolare superiore e $L \in \mathbb{R}^{n \times n}$ è una matrice triangolare inferiore con diagonale unitaria. Più precisamente:

$$U = L_{n-1} \cdot \dots \cdot L_1 A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} & \cdots & a_{1n} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & a_{33}^{(3)} & \cdots & a_{3k}^{(3)} & \cdots & a_{3n}^{(3)} \\ & & & \ddots & & & \\ & & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & & & \ddots & \\ & & & & & & a_{nn}^{(n)} \end{pmatrix}$$

$$L = L_1^{-1} \cdot \dots \cdot L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & & & \\ m_{21} & 1 & & & & & \\ m_{31} & m_{32} & 1 & & & & \\ \cdots & \cdots & \cdots & 1 & & & \\ & & & m_{j+1j} & 1 & & \\ \cdots & \cdots & \cdots & \cdots & & 1 & \\ m_{n1} & m_{n2} & \cdots & m_{nj} & & m_{nn-1} & 1 \end{pmatrix}.$$

Dimostrazione

- L'algoritmo di Gauss applicato ad una matrice A , sotto l'ipotesi che tutti i perni siano diversi da zero, fornisce una matrice U triangolare superiore e le matrici L_k , $k = 1, \dots, n-1$ contenenti i moltiplicatori m_{ik} , $k = 1, \dots, n-1$, $i = k+1, \dots, n$ che realizzano l'uguaglianza seguente

$$L_{n-1}L_{n-2} \cdots L_2L_1A = U.$$

- Osserviamo che tutte le trasformazioni elementari di Gauss sono triangolari con diagonale unitaria; ciò implica che esse siano non singolari (invertibili). Dunque dalla precedente relazione ricaviamo

$$A = L_1^{-1}L_2^{-1} \cdots L_{n-2}^{-1}L_{n-1}^{-1}U.$$

Se indichiamo con $L \in \mathbb{R}^{n \times n}$ la seguente matrice

$$L = L_1^{-1}L_2^{-1} \cdots L_{n-2}^{-1}L_{n-1}^{-1},$$

allora concludiamo che vale la seguente fattorizzazione

$$A = LU.$$

- Resta da provare che L è triangolare inferiore con diagonale unitaria. Più precisamente, dimostriamo che

$$L = \begin{pmatrix} 1 & & & & & \\ m_{21} & 1 & & & & \\ m_{31} & m_{32} & 1 & & & \\ \cdots & \cdots & \cdots & 1 & & \\ & & & m_{j+1j} & 1 & \\ \cdots & \cdots & \cdots & \cdots & & 1 \\ m_{n1} & m_{n2} & \cdots & m_{nj} & m_{nn-1} & 1 \end{pmatrix}.$$

3) Metodi diretti: fattorizzazione LU

- Ricordiamo che al passo k la matrice A_k viene premoltiplicata per

$$L_k = \begin{pmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & 0 & 1 & & & \\ \vdots & & & \ddots & & \\ 0 & & & & 1 & \\ & & & & -m_{k+1,k} & 1 \\ & & & & \vdots & \\ 0 & 0 & & & -m_{n,k} & 1 \end{pmatrix}.$$

- L_k è triangolare con diagonale unitaria $\Rightarrow \det(L_k) = 1 \Rightarrow L_k$ è non singolare.
- Inoltre vale $L_k = I - m^{(k)} e_k^T$, ovvero

$$L_k = \begin{pmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & 0 & 1 & & & \\ \vdots & & & \ddots & & \\ 0 & & & & 1 & \\ & & & & & 1 \\ & & & & & & 1 \end{pmatrix} - \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{n,k} \end{pmatrix}}_{m^{(k)}} \underbrace{\begin{pmatrix} 0 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix}}_{e_k^T}.$$

- Notiamo che l'inversa della trasformazione elementare k -esima di Gauss è

$$L_k^{-1} = I + m^{(k)} e_k^T = \begin{pmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & 0 & 1 & & & \\ \vdots & & & \ddots & & \\ 0 & & & 1 & & \\ & & & m_{k+1,k} & 1 & \\ & & & \vdots & & \\ 0 & 0 & & m_{n,k} & & 1 \end{pmatrix}.$$

Infatti

$$\begin{aligned} (I - m^{(k)} e_k^T)(I + m^{(k)} e_k^T) &= I - m^{(k)} e_k^T + m^{(k)} e_k^T - \underbrace{m^{(k)} e_k^T m^{(k)} e_k^T}_{=0} \\ &= I. \end{aligned}$$

- Inoltre, se $k < j$, il prodotto delle inverse L_k^{-1} e L_j^{-1} è

$$L_k^{-1} L_j^{-1} = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & m_{k+1,k} & 1 & & \\ \vdots & & & \ddots & \\ 0 & & & 1 & \\ & & & m_{j+1,j} & 1 \\ & & & \vdots & \\ 0 & m_{n,k} & & m_{n,j} & 1 \end{pmatrix}$$

Infatti

$$\begin{aligned} L_k^{-1} L_j^{-1} &= (I + m^{(k)} e_k^T)(I + m^{(j)} e_j^T) \\ &= I + m^{(k)} e_k^T + m^{(j)} e_j^T + \underbrace{m^{(k)} e_k^T m^{(j)} e_j^T}_{=0} \\ &= I + m^{(k)} e_k^T + m^{(j)} e_j^T. \end{aligned}$$

3) Metodi diretti: fattorizzazione LU

- Dalla proprietà precedente, segue che

$$\begin{aligned}
 L_1^{-1} L_2^{-1} &= I + m^{(1)} e_1^T + m^{(2)} e_2^T \\
 (L_1^{-1} L_2^{-1}) L_3^{-1} &= (I + m^{(1)} e_1^T + m^{(2)} e_2^T)(I + m^{(3)} e_3^T) \\
 &= I + m^{(1)} e_1^T + m^{(2)} e_2^T + m^{(3)} e_3^T \\
 &\quad + m^{(1)} \underbrace{e_1^T m^{(3)}}_{=0} e_3^T + m^{(2)} \underbrace{e_2^T m^{(3)}}_{=0} e_3^T \\
 &= I + m^{(1)} e_1^T + m^{(2)} e_2^T + m^{(3)} e_3^T \\
 &\vdots \\
 L &= L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} = I + m^{(1)} e_1^T + m^{(2)} e_2^T + \dots + m^{(n-1)} e_{n-1}^T \\
 &= \begin{pmatrix} 1 & & & & & \\ m_{21} & 1 & & & & \\ m_{31} & m_{32} & 1 & & & \\ \dots & \dots & \dots & 1 & & \\ & & & m_{j+1,j} & 1 & \\ \dots & \dots & \dots & \dots & \dots & 1 \\ m_{n1} & m_{n2} & \dots & m_{nj} & m_{nn-1} & 1 \end{pmatrix}
 \end{aligned}$$

il che conclude la dimostrazione. \square

3) Metodi diretti: fattorizzazione LU

Se $A \in \mathbb{R}^{n \times n}$ ammette la fattorizzazione LU , allora

$$\begin{aligned} Ax &= b \\ \Updownarrow \\ L \underbrace{Ux}_y &= b \end{aligned}$$

Dunque $Ax = b$ equivale a risolvere in sequenza i seguenti due sistemi triangolari

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

Il costo totale del metodo è la somma dei due seguenti costi:

- Costo della fattorizzazione: $\mathcal{O}(\frac{2}{3}n^3)$
- Costo della soluzione dei due sistemi triangolari: $\mathcal{O}(2n^2)$.

Osservazione

Applicare le regole di aggiornamento al termine noto durante il processo di fattorizzazione è equivalente alla soluzione del sistema $Ly = b$.

Proposizione (unicità della fattorizzazione LU)

Se $A \in \mathbb{R}^{n \times n}$ è non singolare e ammette la fattorizzazione LU , allora tale fattorizzazione è unica.

Dimostrazione

- Supponiamo per assurdo che esistano due coppie distinte di matrici con le proprietà enunciate nel teorema di fattorizzazione, ovvero:

$$A = L_1 U_1, \quad A = L_2 U_2,$$

ove L_1 e L_2 sono non singolari.

- Se A è non singolare, allora anche U_1 e U_2 sono non singolari. Dunque

$$L_1 U_1 = L_2 U_2 \quad \Rightarrow \quad L_1^{-1} L_2 = U_1 U_2^{-1}.$$

- Notiamo che $L_1^{-1} L_2$ è triangolare inferiore mentre $U_1 U_2^{-1}$ è triangolare superiore, quindi l'uguaglianza è verificata se e solo se le matrici sono entrambe diagonali. Inoltre $L_1^{-1} L_2$ ha diagonale unitaria, quindi deve essere

$$I = L_1^{-1} L_2 = U_1 U_2^{-1} \quad \Leftrightarrow \quad \begin{cases} L_1 = L_2 \\ U_1 = U_2. \end{cases}$$

Proposizione (applicabilità del metodo di Gauss)

Condizione necessaria e sufficiente affinché tutti perni $a_{kk}^{(k)}$ siano diversi da zero (e quindi il metodo di Gauss sia applicabile) è che i **minori principali di A** (ossia i determinanti delle sottomatrici formate dalle prime k righe e k colonne) siano tutti diversi da zero eccetto al più l'ultimo.

3) Metodi diretti: fattorizzazione LU (applicabilità)

Dimostrazione

- Ad ogni passo $k = 1, \dots, n-1$, la matrice $A_k \in \mathbb{R}^{n \times n}$ è data da

$$A_k = L_{k-1} \cdot \dots \cdot L_2 \cdot L_1 \cdot A$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1k} & \dots & a_{1n} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2k}^{(2)} & \dots & a_{2n}^{(2)} \\ & & a_{33}^{(3)} & \dots & a_{3k}^{(k)} & \dots & a_{3n}^{(3)} \\ & & & \ddots & & & \\ & & & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & & & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix} = \begin{pmatrix} 1 & & & & & & \\ -m_{21} & 1 & & & & & \\ -m_{31} & -m_{32} & 1 & & & & \\ \vdots & \vdots & \vdots & & 1 & & \\ -m_{k1} & & & -m_{k,k-1} & 1 & & \\ \vdots & \vdots & \vdots & \vdots & & 0 & 1 \\ & & & & \vdots & \ddots & \\ -m_{n1} & -m_{n2} & \dots & -m_{n,k-1} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3k} & \dots & a_{3n} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} & \dots & a_{kn} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{pmatrix}$$

- Grazie alla struttura triangolare di $L_{k-1} \cdot \dots \cdot L_2 \cdot L_1 \cdot A$, si ha anche la seguente uguaglianza fra sottomatrici:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1k} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2k}^{(2)} \\ & & a_{33}^{(3)} & \dots & a_{3k}^{(k)} \\ & & & & a_{kk}^{(k)} \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & -m_{32} & 1 & & \\ \vdots & \vdots & \vdots & & 1 \\ -m_{k1} & & & -m_{k,k-1} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ a_{31} & a_{32} & \dots & a_{3k} \\ \vdots & \vdots & \dots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix}.$$

3) Metodi diretti: fattorizzazione LU (applicabilità)

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1k} \\ & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2k}^{(2)} \\ & & a_{33}^{(3)} & \dots & a_{3k}^{(k)} \\ & & & \dots & a_{kk}^{(k)} \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & -m_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -m_{k1} & & & -m_{k,k-1} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ a_{31} & a_{32} & \dots & a_{3k} \\ \vdots & \vdots & \dots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix}.$$

- Passando ai determinanti ed applicando la formula di Binet, si ottiene

$$a_{11} \cdot a_{22}^{(2)} \cdot a_{33}^{(3)} \cdot \dots \cdot a_{kk}^{(k)} = 1 \cdot A^{(k)},$$

dove $A^{(k)}$ è il k -esimo minore principale di A .

- Siccome l'uguaglianza sopra riportata deve valere per ogni $k = 1, \dots, n-1$, ne segue che

$$A^{(k)} \neq 0, \quad k = 1, \dots, n-1 \quad \Leftrightarrow \quad a_{kk}^{(k)} \neq 0, \quad k = 1, \dots, n-1,$$

il che conclude la dimostrazione. \square

Teorema di fattorizzazione di Gauss (formulazione alternativa)

Se tutti i minori principali di A sono diversi da zero, tranne al più l'ultimo ($A^{(k)} \neq 0, k = 1, \dots, n-1$), allora esistono una matrice triangolare inferiore L con diagonale unitaria e una matrice triangolare superiore U tali che

$$A = LU.$$

- A differenza della precedente formulazione, questa versione del teorema consente di stabilire nella pratica se una data matrice quadrata ammette o meno la fattorizzazione LU .

Definizione

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

$$\sum_{j=1, j \neq i}^n |a_{ij}| < |a_{ii}|, \quad i = 1, \dots, n.$$

Similmente, $A \in \mathbb{R}^{n \times n}$ si dice **strettamente diagonale dominante per colonne** se

$$\sum_{i=1, i \neq j}^n |a_{ij}| < |a_{jj}|, \quad j = 1, \dots, n.$$

- Ad esempio, la matrice

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

è strettamente diagonale dominante per righe ma non per colonne (vedi terza colonna).

- Le matrici strettamente diagonali dominanti ammettono la fattorizzazione LU (prossima slide).

Proposizione

Una matrice $A \in \mathbb{R}^{n \times n}$ strettamente diagonale dominante per righe (risp. per colonne) ha tutti i perni diversi da zero, ossia:

$$a_{kk}^{(k)} \neq 0, \quad k = 1, \dots, n.$$

Dunque tale matrice è invertibile e ammette la fattorizzazione LU .

Dimostrazione

Supponiamo che A sia diagonale dominante per righe (se è dominante per colonne, il ragionamento è analogo).

- Il primo perno $a_{11}^{(1)}$ non può essere nullo. Se così fosse, tutta la prima riga sarebbe nulla e A non sarebbe più strettamente diagonale dominante.
- Si dimostra che la matrice ottenuta dopo il primo passo del metodo di Gauss è ancora strettamente diagonale dominante per righe. Pertanto, ripetendo il ragionamento sopra riportato, si deduce che $a_{22}^{(2)} \neq 0$.
- Ripetendo il ragionamento per ogni $k = 1, \dots, n$, si dimostra che tutti i perni sono non nulli. Di conseguenza A è invertibile e ha la fattorizzazione LU .

Vantaggi della fattorizzazione LU

1. La fattorizzazione LU è molto utile quando si devono risolvere tanti sistemi lineari che hanno la stessa matrice dei coefficienti:

$$Ax = b_i, \quad i = 1, \dots, N \quad \Leftrightarrow \quad \begin{cases} Ly = b_i \\ Ux = y \end{cases}, \quad i = 1, \dots, N$$

Invece che applicare il metodo di Gauss N volte ad un costo di $\mathcal{O}(2Nn^3/3)$ operazioni, calcolo la fattorizzazione **una volta sola** e risolvo i $2N$ sistemi triangolari associati **ad un costo di $\mathcal{O}(2n^3/3) + \mathcal{O}(2Nn^2)$ operazioni**.

2. Applicando la formula di Binet alla fattorizzazione $A = LU$, si ha che

$$\det(A) = \det(L) \det(U) = 1 \cdot a_{11} \cdot a_{22}^{(2)} \cdot \dots \cdot a_{nn}^{(n)}.$$

Ciò offre un modo computazionalmente sostenibile di calcolare $\det(A)$
 $\Rightarrow \mathcal{O}(2/3n^3)$ **operazioni** + $n - 1$ **prodotti**.

Per contro la regola di Laplace avrebbe richiesto $n!$ operazioni.

3) Metodi diretti: fattorizzazione LU (svantaggi)

Svantaggi della fattorizzazione LU

1. L'algoritmo di Gauss non è applicabile a tutte le matrici.

Esistono infatti delle matrici (anche invertibili) per cui l'algoritmo termina senza successo a causa della presenza di **pivot nulli**.

Ad esempio, l'algoritmo non viene portato a termine per le seguenti matrici

$$\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 0 \\ 2 & 2 & 3 \\ 0 & -1 & 0 \end{pmatrix}.$$

2. L'algoritmo di Gauss può essere instabile numericamente.

Ciò è dovuto alla presenza di **pivot molto piccoli, ma non nulli**, che vengono utilizzati al denominatore dei moltiplicatori di Gauss.

⇒ possibili errori di incolonnamento!

3) Metodi diretti: fattorizzazione LU (svantaggi)

Esempio (di instabilità numerica del metodo di Gauss)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Supponiamo per semplicità di poter derivare la fattorizzazione LU senza errori di arrotondamento:

$$L_1 = \begin{pmatrix} 1 & & \\ -1 & 1 & \\ -1 & 0 & 1 \end{pmatrix} \Rightarrow A_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0.0001 & 1 \\ 0 & 1 & 1 \end{pmatrix} \Rightarrow L_2 = \begin{pmatrix} 1 & & \\ 0 & 1 & \\ 0 & -10000 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0.0001 & 1 \\ 0 & 0 & -9999 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & & \\ 1 & 1 & \\ 1 & 10000 & 1 \end{pmatrix}.$$

Esempio (di instabilità numerica del metodo di Gauss)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Risolvendo il primo sistema triangolare inferiore $Ly = b$, si ottiene

$$\begin{cases} y_1 = 1 \\ y_1 + y_2 = 2 \\ y_1 + 10000y_2 + y_3 = 1 \end{cases} \Rightarrow y = \begin{pmatrix} 1 \\ 1 \\ -10000 \end{pmatrix}.$$

Esempio (di instabilità numerica del metodo di Gauss)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Supponiamo ora di risolvere il sistema triangolare superiore $Ux = y$ usando le regole dell'aritmetica finita con 4 cifre decimali di precisione:

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ 0.0001x_2 + x_3 = 1 \\ -9999x_3 = -10^4 \end{cases} \Rightarrow \begin{aligned} fl(x_3) &= 1 \text{ invece di } 1.00010001.. \\ fl(x_2) &= (1 - 1)/10^{-4} = 0 \\ fl(x_1) &= (1 - 0 - 1)/1 = 0. \end{aligned}$$

Esempio (di instabilità numerica del metodo di Gauss)

Sia dato il sistema

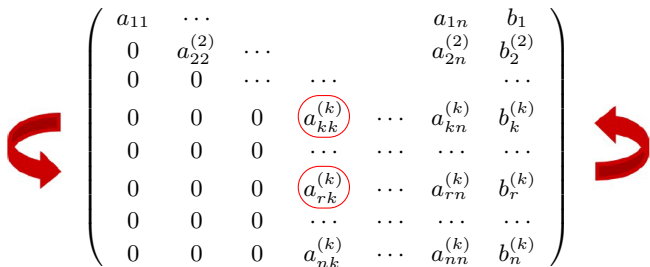
$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Un piccolo errore dell'ordine di 10^{-4} nel calcolo di x_3 si è amplificato in un errore dell'ordine dell'unità nel calcolo delle altre due componenti della soluzione.
- La causa è la scelta di un perno molto piccolo e quindi, di conseguenza, di un moltiplicatore molto grande.
- Come porre rimedio a questa instabilità?

3) Metodi diretti: fattorizzazione LU con pivoting

- Introduciamo una modifica all'algoritmo di Gauss in modo tale che possa essere applicato a tutte le matrici non singolari.
- L'idea è quella di introdurre la possibilità di scambiare le righe della matrice (**pivoting**) durante il procedimento di fattorizzazione.
- Al passo k , si scambiano la riga k e la riga r della matrice A_k . Deve essere $r \geq k$ per non compromettere la struttura triangolare che si sta costruendo. Dopo lo scambio di righe, si procede con il calcolo dei moltiplicatori e della corrispondente trasformazione elementare di Gauss.
- Se gli stessi scambi vengono applicati anche alle componenti del termine noto, ciò equivale a scambiare due equazioni del sistema $Ax = b$.


$$\begin{pmatrix} a_{11} & \cdots & & & a_{1n} & b_1 \\ 0 & a_{22}^{(2)} & \cdots & & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & \cdots & \cdots & & \cdots \\ 0 & 0 & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & a_{rk}^{(k)} & \cdots & a_{rn}^{(k)} & b_r^{(k)} \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)} \end{pmatrix}$$

Definizione

Si dice **matrice di permutazione elementare** una qualunque matrice ottenuta dall'identità scambiando due qualunque righe o due colonne.

$$P_{ij} = \begin{pmatrix} 1 & & & & & & & & & & \\ & \dots & & & & & & & & & \\ & & 1 & & & & & & & & \\ i & - & - & - & 0 & - & - & - & 1 & - & - & - \\ & & & 1 & & & & & & & \\ & & & & \dots & & & & & & \\ & & & & & 1 & & & & & \\ j & - & - & - & 1 & - & - & - & 0 & - & - & - \\ & & & & & & & 1 & & & \\ & & & & & & & & \dots & & \\ & & & & & & & & & 1 & \end{pmatrix}.$$

- P_{ij} è la matrice ottenuta scambiando la riga i con la riga j della matrice I .
- La matrice $P_{ij}A$ si ottiene scambiando la riga i e la riga j di A .
- La matrice AP_{ij} si ottiene scambiando la colonna i e la colonna j di A .

Proposizione

Sia P_{ij} una matrice di permutazione elementare.

1. P_{ij} è non singolare: $\det(P_{ij}) \neq 0$.
2. P_{ij} è simmetrica: $P_{ij} = P_{ij}^T$.
3. P_{ij} è ortogonale: $P_{ij}^{-1} = P_{ij}^T$.

Dimostrazione

1. Ricordando che il determinante cambia segno ad ogni scambio di riga, si ha

$$\det(P_{ij}) = -\det(I) = -1 \neq 0.$$

2. Si ha

$$\begin{aligned} P_{ij}(r, c) &= P_{ij}(c, r) = 0, & (c, r) &\neq (i, j) \\ P_{ij}(i, j) &= P_{ij}(j, i) = 1. \end{aligned}$$

3. Dalla proprietà 2 e dalla definizione di P_{ij} , segue che

$$P_{ij}P_{ij}^T = P_{ij}P_{ij} = I.$$

Definizione

Si dice **matrice di permutazione** un prodotto di matrici di permutazione elementari.

$$P = P_{ij} \cdot P_{hk} \cdot \dots \cdot P_{uv}.$$

- Una matrice di permutazione è ortogonale in quanto prodotto di matrici ortogonali:

$$\begin{aligned} P^{-1} &= P_{uv}^{-1} \cdot \dots \cdot P_{hk}^{-1} \cdot P_{ij}^{-1} \\ &= P_{uv}^T \cdot \dots \cdot P_{hk}^T \cdot P_{ij}^T \\ &= (P_{ij} \cdot P_{hk} \cdot \dots \cdot P_{uv})^T = P^T. \end{aligned}$$

Fattorizzazione LU con pivoting: primo passo

1. Si definisce la matrice di permutazione elementare P_1 che scambia la riga 1 con una qualsiasi riga r tale che $a_{r1} \neq 0$.
 - Siccome A è non singolare, segue che l'elemento a_{r1} esiste.
 - Se $a_{11} \neq 0$ è possibile scegliere $P_1 = I$, ossia non effettuare alcuno scambio.
2. Si calcolano i moltiplicatori associati alla prima colonna della matrice permutata $P_1 A$ e si definisce la trasformazione elementare di Gauss L_1 corrispondente.
3. Si aggiornano gli elementi della matrice, definendo $A_2 = L_1 P_1 A$.

$$A_1 \equiv A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \Rightarrow A_2 = L_1 P_1 A = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \cdots & & & \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}.$$

Osservazione

Definiamo la matrice \tilde{A}_2 come la sottomatrice di A_2 formata dalle sue ultime $n - 1$ righe e colonne:

$$\tilde{A}_2 = \begin{pmatrix} a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & & \vdots \\ a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}.$$

Allora osserviamo che:

- A_2 è non singolare perché prodotto di matrici non singolari. In particolare, vale che

$$\det(A_2) = (-1)^s \det(A)$$

dove $s = 0$ se $P_1 = I$, oppure $s = 1$ se $P_1 \neq I$.

- Per il teorema di Laplace si ha che

$$\det(A_2) = a_{11}^{(2)} \det(\tilde{A}_2).$$

- Dalla precedente uguaglianza e dal fatto che $\det(A) \neq 0$ e $a_{11}^{(2)} \neq 0$, concludiamo che $\det(\tilde{A}_2) \neq 0$, dunque \tilde{A}_2 è non singolare.
 \Rightarrow **Esiste almeno un indice $i \in \{2, \dots, n\}$ tale per cui $a_{i2}^{(2)} \neq 0$.**
 \Rightarrow È possibile procedere all'eventuale scambio di righe e alla trasformazione elementare di Gauss.

3) Metodi diretti: fattorizzazione LU con pivoting

Fattorizzazione LU con pivoting: k -esimo passo

A partire da A_{k-1} , si ottiene la matrice

$$A_k = L_{k-1}P_{k-1} \cdot \dots \cdot L_2P_2L_1P_1A = \begin{pmatrix} a_{11}^{(2)} & & & & a_{1n}^{(2)} \\ & \ddots & & & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix}.$$

- Definendo

$$\tilde{A}_k = \begin{pmatrix} a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ \vdots & & \vdots \\ a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix}$$

si osserva che

$$\det(A_k) = a_{11}^{(1)} a_{22}^{(2)} \cdot \dots \cdot a_{k-1,k-1}^{(k-1)} \det(\tilde{A}_k).$$

- Siccome $\det(A_k) \neq 0$ (ipotesi per induzione) e $a_{jj}^{(j+1)} \neq 0$, $j = 1, \dots, k-1$ (grazie agli scambi di righe), ne segue che \tilde{A}_k è non singolare, quindi esiste almeno un elemento diverso da zero nella sua prima colonna.

3) Metodi diretti: fattorizzazione LU con pivoting

Fattorizzazione LU con pivoting: $(n - 1)$ esimo passo

A partire da A_{n-1} , si ottiene la matrice

$$A_n = L_{n-1}P_{n-1} \cdot \dots \cdot L_2P_2L_1P_1A = \begin{pmatrix} a_{11}^{(2)} & & & & a_{1n}^{(2)} \\ & \ddots & & & \vdots \\ & & a_{kk}^{(k+1)} & \dots & a_{kn}^{(k+1)} \\ & & & \ddots & \vdots \\ & & & & a_{nn}^{(n)} \end{pmatrix}.$$

Teorema di fattorizzazione LU con pivoting

Se A è non singolare, allora esistono una matrice di permutazione P , una matrice triangolare superiore U e una matrice triangolare inferiore L tali che

$$PA = LU.$$

Dimostrazione

La matrice U è definita per costruzione: $U = A_n = L_{n-1}P_{n-1} \cdot \dots \cdot L_2P_2L_1P_1A$.
Si può dimostrare che

- $P = P_{n-1} \cdot \dots \cdot P_1$;
- $L = \prod_{k=1}^{n-1} P_{n-1} \cdot \dots \cdot P_{k+1} L_k^{-1}$.

In altre parole, la k -esima colonna della matrice L contiene i moltiplicatori definiti al passo k permutati secondo gli scambi di righe effettuati nei passi successivi.

3) Metodi diretti: fattorizzazione LU con pivoting

Risoluzione dei sistemi lineari mediante la fattorizzazione LU con pivoting

Per ogni sistema lineare $Ax = b$ con $\det(A) \neq 0$, si può scrivere che

$$Ax = b \quad \Leftrightarrow \quad PAx = Pb \quad \Leftrightarrow \quad LUx = Pb.$$

Ponendo $y = Ux$, segue che risolvere $Ax = b$ è equivalente a risolvere in sequenza i seguenti due sistemi triangolari

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

Grazie alla fattorizzazione $PA = LU$, il determinante di A si può calcolare come

$$\det(A) = (-1)^\sigma u_{11} \cdot \dots \cdot u_{nn},$$

dove σ è il numero di permutazioni non banali effettuate.

3) Metodi diretti: fattorizzazione LU con pivoting

Vantaggi

1. La fattorizzazione $PA = LU$ è applicabile ad ogni matrice invertibile.
Il metodo termina sempre con successo, purché A sia invertibile.

Svantaggi

1. La fattorizzazione $PA = LU$ non è unica.
Infatti essa dipende da come scegliamo di scambiare le righe per portare un elemento non nullo in posizione di perno ad ogni passo del metodo.
⇒ **Di per sé, la fattorizzazione $PA = LU$ non è un algoritmo.**
2. La fattorizzazione $PA = LU$ può essere instabile numericamente.
Gli algoritmi di sostituzione possono diventare instabili quando gli elementi del triangolo sono molto grandi rispetto alla diagonale.
⇒ **Occorre scegliere il perno in modo che i moltiplicatori siano piccoli.**



3) Metodi diretti: fattorizzazione LU con pivoting parziale

Pivoting parziale

È una strategia che consiste nello scegliere come perno l'elemento più grande in valore assoluto tra tutti quelli della prima colonna della matrice \tilde{A}_k .

$$A_k = \begin{pmatrix} a_{11}^{(2)} & & & a_{1n}^{(2)} \\ & \ddots & & \\ & & \begin{matrix} a_{kk}^{(k)} & \dots & \dots & a_{kn}^{(k)} \\ \dots & & & \dots \\ a_{rk}^{(k)} & \dots & \dots & a_{rn}^{(k)} \\ \dots & & & \dots \\ a_{nk}^{(k)} & \dots & \dots & a_{nn}^{(k)} \end{matrix} & \end{pmatrix}, \quad |a_{rk}^{(k)}| = \max_{i \in \{k, \dots, n\}} |a_{ik}^{(k)}|$$

Di conseguenza, i moltiplicatori saranno più piccoli di uno in valore assoluto:

$$|m_{ik}| \leq 1, \quad k = 1, \dots, n-1, \quad i = k+1, \dots, n.$$

3) Metodi diretti: fattorizzazione LU con pivoting parziale

- Costo computazionale

Al passo k occorre effettuare $n - k$ confronti, che hanno il costo di una differenza. Dunque il costo totale è dato da

$$\sum_{k=1}^{n-1} n - k = \sum_{i=1}^{n-1} i \simeq \mathcal{O}\left(\frac{n^2}{2}\right) \text{ operazioni.}$$

N.B. Se A è strettamente diagonale dominante per righe o per colonne, la condizione di pivoting parziale è automaticamente soddisfatta

⇒ Per questo tipo di matrici non è necessario effettuare scambi di righe e si possono evitare i confronti (“costo zero”).

- Stabilità

La fattorizzazione LU con pivoting risulta **stabile nei limiti del condizionamento del problema**: infatti la proprietà $|m_{ik}| \leq 1$ garantisce che non si verifichino errori di incolonnamento nella risoluzione del sistema.

3) Metodi diretti: fattorizzazione LU con pivoting parziale

Esempio (di stabilità numerica del metodo di Gauss con pivoting parziale)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Eseguiamo la fattorizzazione $PA = LU$ senza errori di arrotondamento:

$$P_1 = I, L_1 = \begin{pmatrix} 1 & & \\ -1 & 1 & \\ -1 & 0 & 1 \end{pmatrix} \Rightarrow A_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0.0001 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$P_2 A_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0.0001 & 1 \end{pmatrix}, L_2 = \begin{pmatrix} 1 & & \\ 0 & 1 & \\ 0 & -0.0001 & 1 \end{pmatrix} \Rightarrow A_3 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0.9999 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & & \\ 1 & 1 & \\ 1 & 0.0001 & 1 \end{pmatrix}, \quad U = A_3.$$

Esempio (di stabilità numerica del metodo di Gauss con pivoting parziale)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Risolvendo il sistema triangolare inferiore $Ly = Pb$, si ottiene

$$\begin{cases} y_1 = 1 \\ y_1 + y_2 = 1 \\ y_1 + 0.0001y_2 + y_3 = 2 \end{cases} \Rightarrow y = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

Esempio (di stabilità numerica del metodo di Gauss con pivoting parziale)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.0001 \ 1.0001)^T$.

- Supponiamo ora di risolvere il sistema triangolare superiore $Ux = y$ usando le regole dell'aritmetica finita con 4 cifre decimali di precisione:

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_2 + x_3 = 0 \\ 0.9999x_3 = 1 \end{cases} \Rightarrow \begin{cases} fl(x_3) = 1 \text{ invece di } 1.00010001.. \\ fl(x_2) = -1 \\ fl(x_1) = 1. \end{cases}$$

3) Metodi diretti: fattorizzazione LU con pivoting parziale

Esempio (di stabilità numerica del metodo di Gauss con pivoting parziale)

Sia dato il sistema

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix},$$

avente soluzione esatta $x = (1 \ -1.\overline{0001} \ 1.\overline{0001})^T$.

- L'errore relativo sulla soluzione in norma infinito è piccolo, infatti:

$$\frac{\left\| \begin{pmatrix} 0 \\ -1 + 1.\overline{0001} \\ 1 - 1.\overline{0001} \end{pmatrix} \right\|_{\infty}}{\left\| \begin{pmatrix} 1 \\ -1.\overline{0001} \\ 1.\overline{0001} \end{pmatrix} \right\|_{\infty}} = \frac{0.\overline{0001}}{1.\overline{0001}} \simeq 10^{-4}.$$

In generale, l'introduzione della strategia del pivoting parziale rende l'algoritmo di Gauss più stabile.

3) Metodi diretti: fattorizzazione LU con pivoting totale

Pivoting totale

È una strategia che consiste nello scegliere come perno l'elemento più grande in valore assoluto tra tutti quelli della sottomatrice \tilde{A}_k .

$$A_k = \begin{pmatrix} a_{11}^{(2)} & & & a_{1n}^{(2)} \\ & \ddots & & \\ & & \begin{matrix} a_{kk}^{(k)} & \dots & \dots & a_{kn}^{(k)} \\ \dots & & & \dots \\ a_{rk}^{(k)} & \dots & \dots & a_{rn}^{(k)} \\ \dots & & & \dots \\ a_{nk}^{(k)} & \dots & \dots & a_{nn}^{(k)} \end{matrix} & \end{pmatrix}, \quad |a_{rs}^{(k)}| = \max_{i,j \in \{k, \dots, n\}} |a_{ij}^{(k)}|$$

Di conseguenza, i moltiplicatori saranno più piccoli di uno in valore assoluto:

$$|m_{ik}| \leq 1, \quad k = 1, \dots, n-1, \quad i = k+1, \dots, n.$$

Per realizzare questa strategia, è richiesto lo scambio sia di righe che di colonne.

Teorema di fattorizzazione $PAQ = LU$

Se A è non singolare, allora esistono due matrici di permutazione P e Q , una matrice triangolare superiore U e una matrice triangolare inferiore L tali che

$$PAQ = LU.$$

- Costo computazionale

Al passo k occorre confrontare tutti gli elementi della sottomatrice \tilde{A}_k , che è di ordine $n - k + 1$. Dunque il costo totale è dato da

$$\sum_{i=1}^{n-1} i^2 \simeq \mathcal{O}\left(\frac{n^3}{3}\right) \text{ operazioni.}$$

Il costo del pivoting totale è comparabile a quella della fattorizzazione stessa.
 \Rightarrow Nella pratica si tende ad utilizzare il pivoting parziale, che ha un buon rapporto costi-benefici in termini di stabilità.

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 qualunque sistema associato ad una matrice non singolare;
2. rendere più stabile l'algoritmo di fattorizzazione, mediante la scelta di un perno "grande" (pivoting parziale o totale).

3) Metodi diretti: varianti della fattorizzazione LU per matrici speciali

- La fattorizzazione LU (o di Gauss) con pivoting parziale permette la fattorizzazione di qualsiasi matrice non singolare.
- Si possono ricavare delle varianti di tale fattorizzazione, specifiche per certe classi di matrici, che tengono conto di eventuali proprietà della matrice fattorizzata, al fine di:
 1. risparmiare complessità computazionale;
 2. risparmiare memoria.

Queste implementazioni ad hoc si basano su di alcuni risultati teorici.

- Tratteremo le varianti della fattorizzazione LU delle seguenti classi di matrici:
 1. le matrici simmetriche (fattorizzazione LDL^T);
 2. le matrici simmetriche definite positive (fattorizzazione di Cholesky);
 3. le matrici a banda;
 4. le matrici sparse.

Definizione

Una matrice $A \in \mathbb{R}^{n \times n}$ si dice simmetrica se

$$A = A^T \quad \Leftrightarrow \quad a_{ij} = a_{ji}, \quad i, j = 1, \dots, n.$$

- La memorizzazione ottimizzata di una matrice simmetrica richiede $\simeq \frac{n^2}{2}$ locazioni di memoria.

Teorema di fattorizzazione di Gauss per matrici simmetriche

Se $A \in \mathbb{R}^{n \times n}$ è simmetrica e tutti i suoi minori principali sono diversi da zero, allora esistono una matrice L triangolare inferiore con diagonale unitaria e una matrice diagonale D con elementi diagonali diversi da zero tali che

$$A = LDL^T.$$

3) Metodi diretti: fattorizzazione LU per matrici simmetriche

Dimostrazione

- Dalle ipotesi si ha $A = LU$ con U non singolare e L con diagonale unitaria.
- Definiamo

$$D = \begin{pmatrix} u_{11} & & \\ & \ddots & \\ & & u_{nn} \end{pmatrix} \Rightarrow A = LD \underbrace{D^{-1}U}_{=R}.$$

- La matrice $R = D^{-1}U$ è triangolare superiore e ha diagonale unitaria. Inoltre si dimostra che $R = L^T$. Infatti, dalla simmetria di A si ha che

$$\begin{aligned} A = A^T &\Leftrightarrow LDR = (LDR)^T = R^T DL^T \Rightarrow \underbrace{DRL^{-T}}_{\text{triang. superiore}} = \underbrace{L^{-1}R^T D}_{\text{triang. inferiore}} \\ &\Rightarrow DRL^{-T} \text{ è diagonale} \\ &\Rightarrow RL^{-T} \text{ è diagonale.} \end{aligned}$$

Ma RL^{-T} ha diagonale unitaria, dunque

$$RL^{-T} = I \Rightarrow R = L^T.$$



3) Metodi diretti: fattorizzazione LU per matrici simmetriche

- Il teorema di fattorizzazione di Gauss per matrici simmetriche implica che gli elementi da calcolare sono solo quelli della matrice L e della diagonale di D .
- La complessità computazionale della fattorizzazione può essere dimezzata utilizzando un diverso metodo per calcolare gli elementi di L e D .
- Tale metodo prende il nome di **metodo di pavimentazione**.

3) Metodi diretti: fattorizzazione LU per matrici simmetriche

$$A = LDL^T$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} d_{11} & & & \\ & d_{22} & & \\ & & d_{33} & \\ & & & \ddots \\ & & & & d_{nn} \end{pmatrix} \begin{pmatrix} 1 & l_{21} & l_{31} & \cdots & l_{n1} \\ & 1 & l_{32} & \cdots & l_{n2} \\ & & 1 & \cdots & l_{n3} \\ & & & \cdots & \cdots \\ & & & & 1 \end{pmatrix}$$

$$a_{ij} = (l_{i1} \ l_{i2} \ \cdots \ l_{ij} \ \cdots \ l_{ii-1} \ 1 \ 0 \ \cdots \ 0) \begin{pmatrix} d_{11}l_{j1} \\ d_{22}l_{j2} \\ \vdots \\ d_{j-1,j-1}l_{jj-1} \\ d_{jj} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = d_{jj}l_{ij} + \sum_{k=1}^{j-1} l_{ik}d_{kk}l_{jk},$$

$$j = 1, \dots, n, \ i = j, \dots, n.$$

3) Metodi diretti: fattorizzazione LU per matrici simmetriche

$$a_{ij} = d_{jj}l_{ij} + \sum_{k=1}^{j-1} l_{ik}d_{kk}l_{jk}, \quad j = 1, \dots, n, \quad i = j, \dots, n.$$

Primo passo ($j = 1$)

$$i = 1 \Rightarrow a_{11} = d_{11} \Rightarrow d_{11} = a_{11}$$

$$i > 1 \Rightarrow a_{i1} = d_{11}l_{i1} \Rightarrow l_{i1} = a_{i1}/d_{11}.$$

$$\begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \cdots \\ \cdots & & & & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} d_{11} & & & & \\ & d_{22} & & & \\ & & d_{33} & & \\ & & & \ddots & \\ & & & & d_{nn} \end{pmatrix} \begin{pmatrix} 1 & l_{21} & l_{31} & \cdots & l_{n1} \\ & 1 & l_{32} & \cdots & l_{n2} \\ & & 1 & \cdots & l_{n3} \\ & & & & \\ & & & & 1 \end{pmatrix}$$

3) Metodi diretti: fattorizzazione LU per matrici simmetriche

$$a_{ij} = d_{jj}l_{ij} + \sum_{k=1}^{j-1} l_{ik}d_{kk}l_{jk}, \quad j = 1, \dots, n, \quad i = j, \dots, n.$$

Secondo passo ($j = 2$)

$$i = 2 \Rightarrow a_{22} = d_{22} + d_{11}l_{21}^2 \Rightarrow d_{22} = a_{22} - d_{11}l_{21}^2$$

$$i > 2 \Rightarrow a_{i2} = d_{22}l_{i2} + l_{i1}d_{11}l_{21} \Rightarrow l_{i2} = (a_{i2} - l_{i1}d_{11}l_{21})/d_{22}.$$

$$\begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \cdots \\ \cdots & & & & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} d_{11} & & & & \\ & d_{22} & & & \\ & & d_{33} & & \\ & & & \ddots & \\ & & & & d_{nn} \end{pmatrix} \begin{pmatrix} 1 & l_{21} & l_{31} & \cdots & l_{n1} \\ & 1 & l_{32} & \cdots & l_{n2} \\ & & 1 & \cdots & l_{n3} \\ & & & & \\ & & & & 1 \end{pmatrix}$$

3) Metodi diretti: fattorizzazione LU per matrici simmetriche

Passo j

Al passo j sono già stati calcolati gli elementi

$$d_{kk}, l_{ik}, \quad k = 1, \dots, j-1, \quad i = k+1, \dots, n.$$

Si vogliono calcolare d_{jj} e l_{ij} , $i = j+1, \dots, n$ dalle relazioni

$$a_{ij} = d_{jj}l_{ij} + \sum_{k=1}^{j-1} l_{ik}d_{kk}l_{jk}, \quad i = j, \dots, n.$$

$$i = j \quad \Rightarrow \quad a_{jj} = d_{jj} + \sum_{k=1}^{j-1} l_{jk}^2 d_{kk} \quad \Rightarrow \quad d_{jj} = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_{kk}$$

$$i > j \quad \Rightarrow \quad a_{ij} = d_{jj}l_{ij} + \sum_{k=1}^{j-1} l_{ik}d_{kk}l_{jk} \quad \Rightarrow \quad l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik}d_{kk}l_{jk})/d_{jj}.$$

3) Metodi diretti: fattorizzazione LU per matrici simmetriche

Algoritmo (pseudocodice)

FOR $j = 1, \dots, n$ $d_{jj} \leftarrow a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_{kk}$ FOR $i = j+1, \dots, n$ $l_{ij} \leftarrow (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_{kk} l_{jk}) / d_{jj}$	\Rightarrow	FOR $j = 1, \dots, n$ FOR $k = 1, \dots, j-1$ $p_{jk} \leftarrow l_{jk} d_{kk}$ $d_{jj} \leftarrow a_{jj} - \sum_{k=1}^{j-1} l_{jk} p_{jk}$ FOR $i = j+1, \dots, n$ $l_{ij} \leftarrow (a_{ij} - \sum_{k=1}^{j-1} l_{ik} p_{jk}) / d_{jj}$
--	---------------	---

Complessità computazionale

Al passo j si effettuano $2(j-1) + (j-1)(n-j)$ prodotti. In totale si ha

$$\begin{aligned} 2 \sum_{j=1}^n (j-1) + \sum_{j=1}^n (j-1)(n-j) &= \sum_{j=1}^n (j-1) + n \sum_{j=1}^n (j-1) - \sum_{j=1}^n (j-1)^2 \\ &= \frac{n(n-1)}{2} + n \frac{n(n-1)}{2} - \frac{n(n-1)(2n-1)}{6} \\ &= \frac{n(n-1)}{2} + n(n-1) \left(\frac{n}{2} - \frac{2n-1}{6} \right) \\ &= \frac{n(n-1)}{2} + \frac{n(n-1)(n+1)}{6}, \end{aligned}$$

dunque $\mathcal{O}\left(\frac{n^3}{6}\right)$ prodotti. Contando le somme, si hanno $\mathcal{O}\left(\frac{n^3}{3}\right)$ operazioni.

$$Ax = b \quad \Leftrightarrow \quad L \underbrace{D L^T x}_z = b$$

$$\begin{cases} Lz = b \\ Dy = z \\ L^T x = y. \end{cases}$$

Osservazione

Il primo e il terzo sistema sono triangolari, mentre il secondo è diagonale. Dunque il vettore y si calcola con l'algoritmo di sostituzione dei sistemi diagonali:

$$y_i = \frac{z_i}{d_{ii}}, \quad i = 1, \dots, n.$$

Definizione

Una matrice simmetrica $A \in \mathbb{R}^{n \times n}$ si dice **definita positiva** se vale che

$$\begin{aligned}x^T A x &\geq 0, \quad \forall x \in \mathbb{R}^n \\x^T A x &= 0 \quad \Leftrightarrow \quad x = 0.\end{aligned}$$

- Se A è simmetrica definitiva positiva, tutti i suoi minori principali sono positivi, dunque A soddisfa le ipotesi del teorema di fattorizzazione LDL^T .
- Tuttavia si può ottenere un teorema di fattorizzazione ad hoc.

Teorema di fattorizzazione di Cholesky

Una matrice simmetrica $A \in \mathbb{R}^{n \times n}$ è definita positiva se e solo se esiste una matrice triangolare inferiore \mathcal{L} con elementi diagonali positivi tale che

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

Dimostrazione

Siccome A è simmetrica e definita positiva, si ha

$$A = LDL^T \quad \text{e} \quad x^T LD \underbrace{L^T x}_{=y} > 0, \quad \forall x \neq 0.$$

Sia $x \neq 0$ e $y = L^T x$. Siccome L è non singolare, deve essere $y \neq 0$.

Sostituendo si ha $y^T Dy > 0 \quad \forall y \neq 0$

$\Rightarrow D$ è definita positiva, ossia $d_{ii} > 0, i = 1, \dots, n$.

La tesi segue ponendo $\mathcal{L} = L\Delta$, dove Δ è la matrice diagonale con elementi $\sqrt{d_{ii}}$.

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

Indichiamo con ℓ_{jk} gli elementi di \mathcal{L} . Dal teorema sappiamo che $\mathcal{L} = L\Delta$, ovvero

$$\ell_{jj} = \sqrt{d_{jj}}, \quad \ell_{jk} = l_{jk} \sqrt{d_{kk}}, \quad k = 1, \dots, j-1.$$

Utilizziamo le regole di pavimentazione della fattorizzazione LDL^T per ottenere l'**algoritmo Cholesky**, sostituendo le relazioni precedenti:

$$\begin{array}{l} \text{FOR } j = 1, \dots, n \\ \quad \left[\begin{array}{l} d_{jj} \leftarrow a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_{kk} \\ \text{FOR } i = j+1, \dots, n \\ \quad \mid l_{ij} \leftarrow (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_{kk} l_{jk}) / d_{jj} \end{array} \right. \end{array} \Rightarrow \begin{array}{l} \text{FOR } j = 1, \dots, n \\ \quad \left[\begin{array}{l} \ell_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} \ell_{jk}^2} \\ \text{FOR } i = j+1, \dots, n \\ \quad \mid \ell_{ij} \leftarrow (a_{ij} - \sum_{k=1}^{j-1} \ell_{ik} \ell_{jk}) / \ell_{jj} \end{array} \right. \end{array}$$

Costo computazionale: $\mathcal{O}\left(\frac{n^3}{3}\right)$ operazioni + n estrazioni di radice quadrata.

- Si può dimostrare che, se A è definita positiva, allora gli elementi perno soddisfano automaticamente la condizione di pivoting parziale.
- Si può anche dimostrare che, a differenza dell'algoritmo di Gauss con pivoting parziale o totale, gli elementi della fattorizzazione di Cholesky sono maggiorati da costanti che non dipendono dalla dimensione della matrice (stabilità forte).
- Se la matrice è simmetrica, l'operatore backslash di Matlab tenta per prima cosa di applicare l'algoritmo di Cholesky, finché non trova un radicando negativo, segno che la matrice di partenza non è definita positiva (il teorema di Cholesky caratterizza le matrici definite positive).

3) Metodi diretti: fattorizzazione LU per matrici a banda

Definizione

Una matrice $A \in \mathbb{R}^{n \times n}$ si dice **a banda r, s** se $a_{ij} = 0, \forall j - i > s, \forall i - j > r$. In altre parole: solo s diagonal secondarie superiori ed r diagonal secondarie inferiori contengono (eventualmente) elementi non nulli.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1s+1} & & \\ \cdots & \cdots & \cdots & a_{2s+2} & \\ & & & & \ddots \\ a_{r+11} & & & & \\ & a_{r+22} & & & \\ & & \ddots & & \end{pmatrix}.$$

- Sono interessanti perché molti problemi differenziali provenienti da diverse applicazioni danno luogo a sistemi con questa struttura.
- Possono essere memorizzate in forma compatta in $\mathcal{O}(n(s+r))$ locazioni.
- Si può dimostrare che, nel caso in cui i minori principali siano diversi da zero, le matrici della fattorizzazione $A = LU$ hanno una analoga struttura a banda:

$$L = \begin{pmatrix} 1 & & & \\ \cdots & 1 & & \\ l_{r+11} & \cdots & 1 & \\ & l_{r+22} & \cdots & 1 \end{pmatrix}, \quad U = \begin{pmatrix} r_{11} & \cdots & r_{1s+1} & & \\ & \cdots & \cdots & r_{2s+2} & \\ & & \cdots & \cdots & \\ & & & & r_{nn} \end{pmatrix}.$$

3) Metodi diretti: fattorizzazione LU per matrici a banda

- La complessità della fattorizzazione LU è ridotta nel caso di matrici a banda con minori principali non nulli, poiché gli elementi da calcolare sono solo $\mathcal{O}(n(s+r))$.
- Ciò non è più vero se occorre effettuare degli scambi di righe.

Esempio

Se consideriamo la matrice a banda data da

$$A = \begin{pmatrix} \frac{1}{10} & 5 & & & \\ & 100 & 5 & & \\ & & 2 & \frac{1}{10} & \\ & & & \frac{5}{10} & 5 \\ & & & & \frac{1}{10} \end{pmatrix}$$

risulta $A = LU$ con

$$L = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ \frac{1}{20} & 0 & -\frac{1}{8} & \frac{1}{160} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & & 100 & 5 & \\ & 2 & \frac{1}{10} & 5 & \\ & & 2 & \frac{1}{10} & 5 \\ & & & 2 & \frac{1}{10} \\ & & & & \frac{1}{199} \end{pmatrix}.$$

Con il pivoting parziale L ed U non sono più a banda.

Definizione

Una matrice $A \in \mathbb{R}^{n \times n}$ si dice **sparsa** se il numero di elementi non nulli è $\mathcal{O}(n)$, ossia proporzionale alla dimensione n della matrice e non a n^2 .
In altre parole: il numero di elementi non nulli della matrice è una piccola percentuale rispetto al numero totale degli elementi.

Esempio

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

Gli elementi diversi da zero sono $3n - 2$.
La percentuale di sparsità è $\frac{3n-2}{n^2} \cdot 100 \simeq \frac{3}{n} \cdot 100$.

3) Metodi diretti: fattorizzazione LU per matrici sparse

La memorizzazione di una matrice sparsa avviene tramite il formato **Compressed Column Storage (CCS)**, secondo cui gli elementi non nulli vengono messi in locazioni di memoria contigue

Esempio

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

Secondo il CCS, la matrice viene memorizzata come segue:

	Elementi	Indici riga	Indici colonna	
	4	1	1	
	1	2	1	
	2	3	1	
	1/2	4	1	
	2	5	1	
	1	1	2	
	1/2	2	2	
	2	1	3	
	3	3	3	
	1/2	1	4	
	5/8	4	4	
	2	1	5	
	16	5	5	

double
8 bytes

interi
4 bytes

3) Metodi diretti: fattorizzazione LU per matrici sparse

Fenomeno di fill-in

Non è detto che gli elementi L ed U della fattorizzazione di una matrice sparsa siano altrettanto sparsi.

Esempio

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

$$L = \begin{pmatrix} 1 & & & & \\ \frac{1}{2} & 1 & & & \\ \frac{1}{2} & 1 & 1 & & \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & 1 & \\ \frac{1}{4} & -\frac{1}{2} & -\frac{1}{6} & -\frac{2}{5} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 4 & 1 & 2 & \frac{1}{2} & 2 \\ & -\frac{1}{2} & 2 & -\frac{1}{4} & -1 \\ & & -3 & 0 & 16 \\ & & & \frac{5}{8} & -4 \\ & & & & \frac{1}{15} \end{pmatrix}.$$

Esistono delle **tecniche di permutazione (o di reordering)** delle righe e delle colonne delle matrici sparse che hanno l'obiettivo di minimizzare il riempimento (o fill-in) dei fattori L ed U .

Teorema di fattorizzazione QR

Sia $A \in \mathbb{R}^{n \times n}$ una matrice non singolare. Allora esistono una matrice ortogonale $Q \in \mathbb{R}^{n \times n}$ e una matrice triangolare superiore non singolare $R \in \mathbb{R}^{n \times n}$ tali che

$$A = QR.$$

Dimostrazione

- La dimostrazione si basa su un algoritmo che “costruisce” le matrici Q ed R .
- Tale algoritmo consiste nel premoltiplicare ripetutamente la matrice A per una successione di matrici di trasformazione elementari, dette **trasformazioni elementari di Householder**, sulla falsariga di quanto già visto per la fattorizzazione LU .
- In questo caso, le matrici di trasformazione non sono triangolari, bensì **ortogonali**.

Definizione

Dato un vettore $v \in \mathbb{R}^n$, $v \neq 0$, si definisce **trasformazione elementare di Householder associata a v** la matrice

$$U = I - \frac{1}{\alpha} vv^T, \quad \text{dove } \alpha = \frac{1}{2} \|v\|^2.$$

Proprietà

1. La matrice U sopra definita è **simmetrica**, infatti:

$$U^T = \left(I - \frac{1}{\alpha} vv^T \right)^T = I - \frac{1}{\alpha} (vv^T)^T = I - \frac{1}{\alpha} vv^T = U.$$

2. La matrice U è **ortogonale**, infatti

$$\begin{aligned} U^T U &= U U \quad (U \text{ è simmetrica}) \\ &= \left(I - \frac{1}{\alpha} vv^T \right) \left(I - \frac{1}{\alpha} vv^T \right) \\ &= I - \frac{1}{\alpha} vv^T - \frac{1}{\alpha} vv^T + \frac{1}{\alpha^2} v \underbrace{v^T v}_{\|v\|^2 = 2\alpha} v^T \\ &= I - \frac{2}{\alpha} vv^T + \frac{2}{\alpha} vv^T = I. \end{aligned}$$

Complessità computazionale del prodotto matrice di Householder - vettore

Sia U la trasformazione elementare di Householder associata al vettore $v \neq 0$ e sia y un vettore di \mathbb{R}^n . Si vuole calcolare

$$z = Uy.$$

- Non è necessario calcolare esplicitamente la matrice U , che richiederebbe n^2 prodotti per il termine vv^T . Infatti, applicando le proprietà distributiva e associativa del prodotto matriciale, il calcolo si effettua come

$$z = Uy = \left(I - \frac{1}{\alpha} vv^T \right) y = y - \frac{1}{\alpha} v(v^T y).$$

- Di conseguenza, dati v e y , la complessità computazionale ammonta a **$6n$ operazioni**, secondo quanto riportato sotto:

			prodotti	somme
α	\leftarrow	$\frac{1}{2}\ v\ ^2$	n	n
τ	\leftarrow	$(v^T y)/\alpha$	n	n
w	\leftarrow	τv	n	
z	\leftarrow	$y - w$		n

Proposizione (annullamento componenti tramite trasformazioni di Householder)

Dato $z \in \mathbb{R}^n$, $z \neq 0$, definita U come la trasformazione di Householder associata al vettore $v = z + \sigma e_1$, dove e_1 è la prima colonna della matrice identità di ordine n e $\sigma = \|z\|$, si ha che Uz annulla tutte le componenti di z tranne la prima:

$$Uz = -\sigma e_1 = \begin{pmatrix} -\sigma \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Dimostrazione

Ricordiamo che $U = I - \frac{1}{\alpha} vv^T$, dove

$$\begin{aligned} \alpha &= \frac{1}{2} \|v\|^2 = \frac{1}{2} (z + \sigma e_1)^T (z + \sigma e_1) \\ &= \frac{1}{2} (z^T z + 2\sigma z^T e_1 + \sigma^2) = \frac{1}{2} (\sigma^2 + 2\sigma z_1 + \sigma^2) = \sigma^2 + \sigma z_1. \end{aligned}$$

Allora il prodotto Uz si può scrivere come

$$\begin{aligned} Uz &= \left(I - \frac{1}{\alpha} vv^T \right) z = z - \frac{1}{\alpha} (z + \sigma e_1)(z + \sigma e_1)^T z \\ &= z - \frac{1}{\alpha} (z + \sigma e_1)(z^T z + \sigma e_1^T z) = z - \frac{1}{\alpha} (z + \sigma e_1) \underbrace{(\sigma^2 + \sigma z_1)}_{\alpha} = z - z - \sigma e_1 = -\sigma e_1. \end{aligned}$$

4) Metodi diretti: fattorizzazione QR

- L'idea è quella di utilizzare le trasformazioni di Householder in successione, come fatto con quelle di Gauss nella fattorizzazione LU , al fine di annullare gli elementi del triangolo inferiore della matrice A :

$$U_{n-1} \cdot \dots \cdot U_1 A = R.$$

- Al passo k , la matrice U_k è definita in modo che nel prodotto vengano eliminati tutti gli elementi della colonna k , sulle righe dalla $(k + 1)$ -esima fino alla n -esima.

Fattorizzazione QR : primo passo

- Scriviamo A come la seguente matrice a blocchi

$$A = (a_1 \ a_2 \ \dots \ a_n),$$

dove a_k indica la k -esima colonna di A , $k = 1, \dots, n$.

- Definiamo U_1 la trasformazione elementare di Householder associata al vettore $v_1 = a_1 + \sigma_1 e_1$, dove $\sigma_1 = \|a_1\|$ ed e_1 è la prima colonna di I_n . Si ha

$$A_2 = U_1 A = (U_1 a_1 \ U_1 a_2 \ \dots \ U_1 a_n) = \begin{pmatrix} -\sigma_1 & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & & & \\ 0 & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} \end{pmatrix},$$

dove

$$\begin{pmatrix} a_{1k}^{(2)} \\ a_{2k}^{(2)} \\ \vdots \\ a_{nk}^{(2)} \end{pmatrix} = U_1 a_k, \quad k = 1, \dots, n.$$

Fattorizzazione QR : primo passo

- La sottocolonna estratta dalla 2^a colonna, prendendo gli elementi dalla riga 2 alla riga n , è diversa dal vettore nullo:

$$\mathbb{R}^{n-1} \ni a_2^{(2)} = \begin{pmatrix} a_{22}^{(2)} \\ \vdots \\ a_{n2}^{(2)} \end{pmatrix} \neq 0.$$

Infatti, siccome A è non singolare, si ha che $U_1 a_2 \neq 0$. Se per assurdo assumiamo $a_{12}^{(2)} \neq 0$ e $a_{i2}^{(2)} = 0$, $i = 2, \dots, n$, ciò equivale a dire che

$$U_1 a_2 = a_{12}^{(2)} e_1.$$

Allora $U_1 \left(a_1 + \frac{\sigma_1}{a_{12}^{(2)}} a_2 \right) = 0 \Leftrightarrow a_1 + \frac{\sigma_1}{a_{12}^{(2)}} a_2 = 0$, che implicherebbe che a_1 e a_2 sono linearmente dipendenti.

\Rightarrow Ciò è assurdo, dato che A è non singolare per ipotesi.

Fattorizzazione QR : secondo passo

- Al secondo passo, si parte dalla matrice ottenuta al passo precedente:

$$A_2 = \begin{pmatrix} -\sigma_1 & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & & & & \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}, \quad a_2^{(2)} = \begin{pmatrix} a_{22}^{(2)} \\ \vdots \\ a_{n2}^{(2)} \end{pmatrix} \in \mathbb{R}^{n-1}.$$

- Si definisce la matrice U_2 nel seguente modo:

$$U_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \boxed{I_{n-1} - \frac{1}{\alpha_2} v_2 v_2^T} \\ \vdots & & & \\ 0 & & & \end{pmatrix}$$

dove

- $v_2 = a_2^{(2)} + \sigma_2 e_1^{(n-1)}$, con $e_1^{(n-1)}$ la prima colonna di I_{n-1} e $\sigma_2 = \|a_2^{(2)}\|$;
- $\alpha_2 = \frac{1}{2} \|v_2\|^2$.

Fattorizzazione QR : secondo passo

- Si premoltiplica A_2 per U_2 , ottenendo così la nuova matrice A_3 :

$$A_3 = U_2 A_2 = \begin{pmatrix} -\sigma_1 & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & -\sigma_2 & a_{23}^{(3)} & \cdots & a_{2n}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & & & \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{pmatrix}.$$

- La sottomatrice \tilde{A}_2 formata dalle ultime $n - 1$ righe e colonne di A_2 è non singolare (segue dal teorema di Laplace). Pertanto si possono utilizzare gli stessi argomenti del passo 1 per dimostrare che

$$\mathbb{R}^{n-2} \ni a_3^{(3)} = \begin{pmatrix} a_{33}^{(3)} \\ \vdots \\ a_{n3}^{(3)} \end{pmatrix} \neq 0.$$

4) Metodi diretti: fattorizzazione QR

Fattorizzazione QR : k -esimo passo

Si ha che

$$A_k = \begin{pmatrix} -\sigma_1 & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ & \ddots & \cdots & \cdots \\ & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & \vdots & \cdots & \cdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}, \quad U_k = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \overline{I_{n-k+1} - \frac{1}{\alpha_k} v_k v_k^T} & \end{pmatrix}$$

dove v_k ha dimensione $n - k + 1$ e

$$A_{k+1} = U_k A_k = \begin{pmatrix} -\sigma_1 & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ & -\sigma_2 & a_{23}^{(3)} & \cdots & a_{2n}^{(3)} \\ & & \ddots & \cdots & \cdots \\ & & & -\sigma_k & a_{k+1}^{(k+1)} & \cdots & a_{kn}^{(k+1)} \\ & & & \vdots & \cdots & \cdots & \cdots \\ & & & 0 & a_{n+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} \end{pmatrix}.$$

Fattorizzazione QR : $(n - 1)$ esimo passo

Dopo $n - 1$ passi la matrice è in forma triangolare superiore:

$$\underbrace{U_{n-1} \cdot \dots \cdot U_1}_{Q^T} A = R.$$

- Si dimostra che le matrici U_k sono ortogonali e simmetriche
 $\Rightarrow Q$ è ortogonale.
- Moltiplicando ambo i membri dell'uguaglianza a sinistra per Q si ottiene

$$A = QR,$$

dove per costruzione $Q = U_1 U_2 \dots U_{n-1}$.

4) Metodi diretti: fattorizzazione QR

Algoritmo basato sulla fattorizzazione QR

- Si usa il triangolo superiore di A per gli elementi di R , utilizzando un vettore supplementare $\sigma \in \mathbb{R}^n$ per gli elementi diagonali di R .
- La matrice Q non viene esplicitamente calcolata, ma i vettori v_k e i coefficienti che definiscono le matrici U_k vengono memorizzati nel triangolo inferiore di A e in un vettore supplementare $\alpha \in \mathbb{R}^n$.

$$\begin{aligned}a_{ik} &\leftarrow r_{ik}, & i < k \\a_{ik} &\leftarrow a_{ik}^{(k)}, & i \geq k \\ \sigma_k &\leftarrow r_{kk} = \|a_k^{(k)}\|, & k = 1, \dots, n \\ \alpha_k &\leftarrow \sigma_k^2 + \sigma_k a_{kk}^{(k)}.\end{aligned}$$

- Un algoritmo basato sulla fattorizzazione QR è dunque il seguente:

```
FOR  $k = 1, \dots, n - 1$ 
   $\sigma_k \leftarrow \sqrt{\sum_{i=k}^n a_{ik}^2}$ 
   $a_{kk} \leftarrow a_{kk} + \sigma_k$ 
   $\alpha_k \leftarrow \sigma_k a_{kk}$ 
  FOR  $j = k + 1, \dots, n$ 
     $\tau \leftarrow (\sum_{i=k}^n a_{ik} a_{ij}) / \alpha_k$ 
    FOR  $i = k, \dots, n$ 
       $a_{ij} \leftarrow a_{ij} - \tau a_{ik}$ 
```

4) Metodi diretti: fattorizzazione QR

Complessità computazionale della fattorizzazione QR

Al passo k

$$A_k = \begin{pmatrix} -\sigma_1 & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ & \ddots & \cdots & \cdots \\ & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & \vdots & \cdots & \cdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}, \quad U_k = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \boxed{I_{n-k+1} - \frac{1}{\alpha_k} v_k v_k^T} & \end{pmatrix}$$

	somme	prodotti	radici
$\alpha_k = \frac{1}{2} \ a_k^{(k)}\ ^2$	$n - k + 1$	$n - k + 1$	
$\sigma_k = \sqrt{\alpha_k}$			1
$v_k = a_k^{(k)} + \sigma_k e_1^{(n-k+1)}$	1		
$(I_{n-k+1} - \frac{1}{\alpha_k} v_k v_k^T) a_j^{(k)}$	$2(n - k + 1)(n - k)$	$2(n - k + 1)(n - k)$	
$j = k + 1, \dots, n$			

$$\text{Totale: } \mathcal{O}\left(\frac{4}{3}n^3\right).$$

Risoluzione di un sistema lineare mediante fattorizzazione QR

- Nota la fattorizzazione $A = QR$, un sistema lineare $Ax = b$ può essere risolto come segue

$$Ax = b \quad \Leftrightarrow \quad Q \underbrace{Rx}_y = b$$

$$\begin{cases} y = Q^T b \\ Rx = y \end{cases}$$

- Supponendo di aver memorizzato i vettori v_k nel triangolo inferiore di A , il vettore $y = Q^T b$ si ottiene con il seguente algoritmo:

```

FOR  $k = 1, \dots, n$ 
   $\tau \leftarrow (\sum_{i=k}^n b_i a_{ik}) / \alpha_k$ 
  FOR  $j = k, \dots, n$ 
     $b_j \leftarrow b_j - \tau a_{jk}$ 
    
```


4) Metodi diretti: fattorizzazione QR

Svantaggi della fattorizzazione QR

- La fattorizzazione QR ha un costo più elevato della LU .

Vantaggi della fattorizzazione QR

- La fattorizzazione QR è applicabile anche a matrici rettangolari.
- La fattorizzazione QR è più stabile della fattorizzazione LU .
 - Sappiamo che l'algoritmo di soluzione di un sistema triangolare può diventare instabile quando gli elementi della matrice triangolare diventano "troppo" grandi.
 - In generale, la stabilità delle fattorizzazioni si definisce individuando dei limiti superiori per gli elementi dei fattori. Si parla di **stabilità forte** se questi limiti non dipendono dalla dimensione della matrice, se invece dipendono dalla dimensione della matrice si ha **stabilità debole**.
 - Si può dimostrare che Gauss con pivoting parziale e QR sono stabili debolmente, ma QR è in generale più stabile di Gauss; Cholesky è invece stabile fortemente.

Gauss con pivoting parziale	$ l_{ij} \leq 1, u_{ij} \leq 2^{n-1} \max_{r,s \in \{1, \dots, n\}} a_{rs} $
Cholesky	$ \ell_{ij} \leq a_{ii} $
QR	$ q_{ij} \leq 1, \sqrt{n} \max_{k \in \{1, \dots, n\}} a_{ki} $

4. Metodi iterativi per sistemi lineari

Definizione

Dato $x^{(0)} \in \mathbb{R}^n$, un **metodo iterativo** consiste in una successione di vettori $\{x^{(k)}\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$, detti **iterate**, che converge alla soluzione x^* del problema considerato per $k \rightarrow \infty$, ovvero

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

- Data una norma vettoriale $\|\cdot\|$, è facile dimostrare che

$$\lim_{k \rightarrow \infty} x^{(k)} = x^* \quad \Leftrightarrow \quad \lim_{k \rightarrow \infty} \|x^{(k)} - x^*\| = 0.$$

Una famiglia di metodi iterativi per la soluzione di un sistema lineare

Dato un sistema lineare $Ax = b$, data una matrice $M \in \mathbb{R}^{n \times n}$ non singolare, si può riscrivere equivalentemente il sistema come segue:

$$Ax = b.$$

$$Mx = Mx + b - Ax$$

$$x = (I - M^{-1}A)x + M^{-1}b.$$

Se definiamo

$$\begin{aligned} G &= I - M^{-1}A \\ c &= M^{-1}b \end{aligned}$$

abbiamo dunque riscritto il sistema lineare in forma equivalente come

$$x = Gx + c.$$

Se x^* è la soluzione del sistema di partenza, allora

$$Ax^* = b \quad \Leftrightarrow \quad x^* = Gx^* + c.$$

Una famiglia di metodi iterativi per la soluzione di un sistema lineare

Fissata una matrice M , a partire dalla formulazione equivalente del sistema come

$$x = Gx + c,$$

dove $G = I - M^{-1}A$, si definisce la relazione di ricorrenza

$$x^{(k+1)} = Gx^{(k)} + c, \quad k = 0, 1, 2, \dots$$

che permette di calcolare ogni iterata in funzione della precedente.

- G viene detta la **matrice di iterazione**, mentre M è la **matrice del metodo**.
- Per innescare il procedimento occorre fornire il punto iniziale $x^{(0)} \in \mathbb{R}^n$.
- Diverse scelte di M corrispondono a diversi metodi iterativi.
- Un metodo iterativo si dice **convergente** se per ogni scelta del punto iniziale $x^{(0)} \in \mathbb{R}^n$ la successione generata converge alla soluzione del sistema.
- La convergenza di un metodo dipende dalla scelta di M .

Teorema (condizione sufficiente per la convergenza)

Se $\|G\| < 1$, allora il metodo iterativo

$$x^{(k+1)} = Gx^{(k)} + c, \quad k = 0, 1, 2, \dots$$

è convergente.

Dimostrazione

Studiamo il vettore $e^{(k)} = x^{(k)} - x^*$: convergenza $\iff \lim_{k \rightarrow \infty} e^{(k)} = 0$.

$$\begin{aligned} e^{(k)} &= x^{(k)} - x^* = Gx^{(k-1)} + c - Gx^* - c \\ &= G(x^{(k-1)} - x^*) = Ge^{(k-1)} = G^2e^{(k-2)} = G^ke^{(0)} \quad \text{dove } G^k = \underbrace{G \cdot \dots \cdot G}_k \text{ volte} \end{aligned}$$

$$\lim_{k \rightarrow \infty} x^{(k)} - x^* = \lim_{k \rightarrow \infty} e^{(k)} = \lim_{k \rightarrow \infty} G^k e^{(0)}$$

$$\lim_{k \rightarrow \infty} \|x^{(k)} - x^*\| \leq \lim_{k \rightarrow \infty} \|G^k\| \|e^{(0)}\| \leq \lim_{k \rightarrow \infty} \|G\|^k \|e^{(0)}\| \leq \left(\lim_{k \rightarrow \infty} \|G\|^k \right) \|e^{(0)}\|$$

Se $\|G\| < 1$, allora $\lim_{k \rightarrow \infty} \|G\|^k = 0$ e si ha convergenza per ogni punto iniziale. \square

Teorema (condizione necessaria e sufficiente per la convergenza)

Sia $\rho(G)$ il raggio spettrale di G , ossia il modulo del suo massimo autovalore:

$$\rho(G) = \max_{i \in \{1, \dots, n\}} |\lambda_i(G)|.$$

Il metodo $x^{(k+1)} = Gx^{(k)} + c$, $k = 0, 1, 2, \dots$ è convergente se e solo se $\rho(G) < 1$.

Dimostrazione

Si ha che

$$\lim_{k \rightarrow \infty} x^{(k)} - x^* = \lim_{k \rightarrow \infty} e^{(k)} = \lim_{k \rightarrow \infty} G^k e^{(0)} = \left(\lim_{k \rightarrow \infty} G^k \right) e^{(0)}$$

da cui segue che

$$\lim_{k \rightarrow \infty} x^{(k)} - x^* = 0 \quad \Leftrightarrow \quad \lim_{k \rightarrow \infty} G^k = 0.$$

Si può dimostrare che

$$\lim_{k \rightarrow \infty} G^k = 0 \quad \Longleftrightarrow \quad \rho(G) < 1. \quad \square$$

Osservazione (velocità di convergenza)

$$\|x^{(k)} - x^*\| = \|e^{(k)}\| \simeq [\rho(G)]^k$$

Quanto più $\rho(G)$ è piccolo, tanto più velocemente $x^{(k)}$ converge ad x^* .

- Per definire un algoritmo, occorre individuare una condizione, detta **criterio di arresto**, verificata la quale si arresta il calcolo delle iterate, con la garanzia che l'ultima iterata calcolata approssimi la soluzione del problema entro una certa tolleranza ϵ fissata a priori
- In altre parole, dato ϵ si vorrebbe individuare per quale k si ha

$$\|x^{(k)} - x^*\| \leq \epsilon$$

in corrispondenza del quale arrestare il procedimento.

- Occorre stimare l'errore $\|x^{(k)} - x^*\|$ in base a quantità calcolabili.

Proposizione (stima dell'errore #1)

Sia $\tau > 0$. Se per un certo k si ha che

$$\|x^{(k+1)} - x^{(k)}\| \leq \tau$$

allora

$$\|x^{(k)} - x^*\| \leq \epsilon, \quad \text{con } \epsilon = \tau \|(G - I)^{-1}\|.$$

Dimostrazione

Se x^* è la soluzione del sistema, si ha che $x^* = Gx^* + c$. Dunque

$$\begin{aligned} x^{(k+1)} - x^{(k)} &= Gx^{(k)} + c - x^{(k)} = Gx^{(k)} - Gx^* + x^* - x^{(k)} \\ &= G(x^{(k)} - x^*) - (x^{(k)} - x^*) = (G - I)(x^{(k)} - x^*). \end{aligned}$$

Nelle ipotesi che il metodo sia convergente, si dimostra anche che la matrice $(G - I)$ è non singolare, pertanto si ottiene che

$$(x^{(k)} - x^*) = (G - I)^{-1}(x^{(k+1)} - x^{(k)}).$$

$$\|x^{(k)} - x^*\| \leq \|(G - I)^{-1}\| \|x^{(k+1)} - x^{(k)}\|,$$

da cui segue la tesi. \square

Abbiamo provato che

$$\|x^{(k)} - x^*\| \leq \|(G - I)^{-1}\| \|x^{(k+1)} - x^{(k)}\|.$$

- Se $\|G\| < 1$ allora $\|(G - I)^{-1}\| \leq \frac{1}{1 - \|G\|}$ e quindi

$$\|x^{(k)} - x^*\| \leq \frac{1}{1 - \|G\|} \|x^{(k+1)} - x^{(k)}\|.$$

Si può quindi concludere che il controllo della quantità $\|x^{(k+1)} - x^{(k)}\|$ (incremento) è significativo soltanto se il valore di $\|G\|$ è molto più piccolo di uno, poiché in tal caso l'errore sarà dello stesso ordine di grandezza dell'incremento.

- Nel caso in cui il valore della norma della matrice G è prossimo ad 1, essendo il fattore $\frac{1}{1 - \|G\|}$ elevato, un incremento di norma piccola non garantisce necessariamente un errore assoluto di norma piccola.
- In altre parole, se la convergenza del metodo è molto lenta, allora può succedere che due iterate successive siano vicine tra loro ma entrambe siano ancora lontane dalla soluzione.

Proposizione (stima dell'errore #2)

Sia $\tau > 0$. Per ogni k si definisca il **vettore residuo** del sistema come

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

Se per un certo k si ha

$$\frac{\|r^{(k)}\|}{\|b\|} \leq \tau, \text{ allora } \frac{\|x^{(k)} - x^*\|}{\|x^*\|} \leq \kappa(A)\tau,$$

dove $\kappa(A)$ è il numero di condizionamento della matrice A .

Dimostrazione

Si ha che (vedi analisi del condizionamento dei sistemi lineari)

$$\frac{\|x^{(k)} - x^*\|}{\|x^*\|} \leq \kappa(A) \frac{\|r^{(k)}\|}{\|b\|},$$

da cui segue la tesi. \square

Osservazione

Se la matrice è ben condizionata ($\kappa(A) \simeq 1$) e

$$\frac{\|r^{(k)}\|}{\|b\|} \leq \tau \text{ allora } \frac{\|x^{(k)} - x^*\|}{\|x^*\|} \simeq \tau.$$

- Tale criterio fornisce un'informazione corretta quando $\kappa(A)$ è piccolo, in quanto a norma di residuo piccola corrisponde una norma di errore piccola.
- Al contrario, non è un buon criterio di arresto nel caso in cui la matrice A sia mal condizionata, in quanto se il fattore di amplificazione $\kappa(A)$ è elevato, un residuo piccolo non garantisce un errore piccolo.

Criteri d'arresto

$$\text{a) } \|x^{(k+1)} - x^{(k)}\| < \tau$$

distanza assoluta tra due iterate

$$\text{b) } \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k+1)}\|} < \tau$$

distanza relativa tra due iterate

$$\text{c) } \|r^{(k)}\| = \|Ax^{(k)} - b\| < \tau$$

residuo assoluto

$$\text{d) } \frac{\|r^{(k)}\|}{\|b\|} = \frac{\|Ax^{(k)} - b\|}{\|b\|} < \tau$$

residuo relativo

Spesso per l'arresto del metodo iterativo si impone la verifica simultanea di due delle condizioni precedenti a) e c) oppure b) e d).

Algoritmo basato su un metodo iterativo

- Non viene memorizzata l'intera successione, ma si utilizzano due vettori per l'iterata corrente e il successivo.
- Se il problema è malcondizionato, soddisfare una tolleranza bassa potrebbe avere un costo proibitivo in termini di tempo di calcolo. Per questo motivo si introduce un parametro di salvaguardia N_{max} che rappresenta il numero massimo di iterazioni eseguito il quale l'algoritmo si arresta anche se i criteri di arresto non sono soddisfatti.

```
INPUT:  $x^{corr}$ ,  $A$ ,  $b$ ,  $G$ ,  $c$ ,  $\tau$ ,  $N_{max}$ 
FOR  $k = 0, 1, \dots, N_{max}$ 
   $r \leftarrow b - Ax^{corr}$ 
   $x^{next} \leftarrow Gx^{corr} + c$ 
  IF  $\frac{\|x^{next} - x^{corr}\|}{\|x^{next}\|} < \tau$  AND  $\frac{\|r\|}{\|b\|} < \tau$ 
    return  $x^{next}$ 
  ELSE
     $x^{corr} \leftarrow x^{next}$ 
  END
IF  $k == N_{max}$ 
  print warning message
END
```

Complessità computazionale

- Il costo complessivo di un metodo iterativo dipende dalla tolleranza τ .
- A priori è possibile valutare soltanto il costo computazionale di una iterazione.
- I metodi per la soluzione di sistemi lineari hanno, in generale, un costo per iterazione di un prodotto matrice-vettore, dunque

Costo per iterazione: $\mathcal{O}(n^2)$.

Pertanto i metodi iterativi diventano competitivi con gli approcci basati sulle fattorizzazioni quando:

- l'accuratezza con cui si vuole approssimare la soluzione del sistema è abbastanza bassa da richiedere poche iterazioni;
- la matrice del sistema (e del metodo) è sparsa o ha una struttura particolare per cui il prodotto matrice-vettore ha complessità molto inferiore ad n^2 .

Metodo iterativo per sistemi lineari

$$x^{(k+1)} = Gx^{(k)} + c$$

con

$$\begin{aligned} G &= I - M^{-1}A \\ c &= M^{-1}b \end{aligned}$$

- La scelta ideale, ma non pratica, per avere la soluzione esatta in una sola iterazione sarebbe $M = A$.
- Una scelta ragionevole consiste nel definire M abbastanza simile ad A per avere buone proprietà di convergenza, ma con una struttura 'semplice', diagonale o triangolare.

Metodi di decomposizione

Si basano su una decomposizione di A nella differenza di due matrici M ed N :

$$A = M - N$$

scegliendo M come matrice del metodo.

$$Mx = Mx + b - Ax = Mx + b - (M - N)x$$

$$Mx = Nx + b$$

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

- La nuova iterata $x^{(k+1)}$ è la soluzione del sistema $Mv = p$, dove $p = Nx^{(k)} + b$ è il termine noto che dipende dai dati (N, b) e dall'iterata corrente $x^{(k)}$.
- La matrice M deve essere scelta in modo che la soluzione del sistema si possa ottenere a basso costo (non superiore ad n^2).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & & & a_{3n} \\ \vdots & \vdots & & \ddots & \vdots & \vdots & \\ a_{i1} & \cdots & a_{ii-1} & a_{ii} & a_{ii+1} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & & \ddots & & \\ a_{n-11} & \cdots & \cdots & \cdots & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ a_{n1} & \cdots & \cdots & \cdots & \cdots & a_{nn-1} & a_{nn} \end{pmatrix}$$

$$E = \begin{cases} e_{ij} = -a_{ij} & i > j \\ 0 & i \leq j \end{cases} \quad F = \begin{cases} f_{ij} = -a_{ij} & i < j \\ 0 & i \geq j \end{cases} \quad D = \begin{cases} d_{ii} = a_{ii} \\ d_{ij} = 0 & i \neq j \end{cases}$$

$$A = D - E - F.$$

Da questa decomposizione ricaviamo due metodi iterativi

- Metodo di Jacobi $M = D, N = E + F$.
- Metodo di Gauss-Seidel $M = D - E, N = F$.

In forma matriciale:

$$Dx^{(k+1)} = (E + F)x^{(k)} + b, \quad k = 0, 1, \dots$$

In forma esplicita:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right) / a_{ii}, \quad i = 1, \dots, n, \quad k = 0, 1, \dots$$

- Ogni componente della nuova iterata $x^{(k+1)}$ si calcola in funzione solo dell'iterata precedente $x^{(k)}$.
- L'ordine con cui si calcolano le componenti della nuova iterata successiva è indifferente: *metodo degli spostamenti simultanei*.
- Il metodo si può formalmente scrivere come

$$x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b \Rightarrow x^{(k+1)} = \mathcal{J}x^{(k)} + c.$$

$$\text{dove } \mathcal{J} = D^{-1}(E+F) = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \dots & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \dots & 0 \end{pmatrix}, c = D^{-1}b = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}.$$

In forma matriciale:

$$(D - E)x^{(k+1)} = Fx^{(k)} + b$$

In forma esplicita:

$$\left\{ \begin{array}{lcl} a_{11}x_1^{(k+1)} & = & \sum_{j=2}^n -a_{1j}x_j^{(k)} + b_1 \\ a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} & = & \sum_{j=3}^n -a_{2j}x_j^{(k)} + b_2 \\ a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k+1)} & = & \sum_{j=4}^n -a_{3j}x_j^{(k)} + b_3 \\ \dots & & \\ a_{i1}x_1^{(k+1)} + a_{i2}x_2^{(k+1)} + \dots + a_{ii}x_i^{(k+1)} & = & \sum_{j=i+1}^n -a_{ij}x_j^{(k)} + b_i \\ \dots & & \\ a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \dots + a_{nn}x_n^{(k+1)} & = & b_n \end{array} \right.$$

In forma matriciale:

$$(D - E)x^{(k+1)} = Fx^{(k)} + b, \quad k = 0, 1, \dots$$

Richiede la soluzione di un sistema triangolare inferiore ad ogni passo.

In forma esplicita:

$$x_i^{(k+1)} = \left(- \sum_{j=i+1}^n a_{ij} x_j^{(k)} + b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} \right) / a_{ii}, \quad i = 1, \dots, n, \quad k = 0, 1, \dots$$

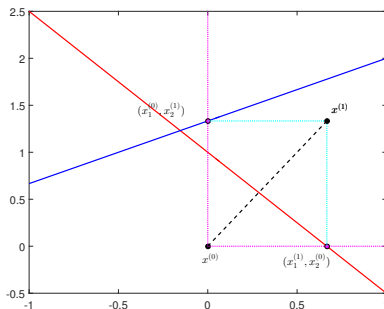
- La componente i -esima della nuova iterata $x_i^{(k+1)}$ si calcola in funzione sia dell'iterata precedente $x^{(k)}$ che delle prime $i - 1$ componenti della nuova iterata stessa, già calcolate.
- L'ordine con cui si calcolano le componenti della nuova iterata successiva è sequenziale: *metodo degli spostamenti successivi*.
- Il metodo si può formalmente scrivere come

$$x^{(k+1)} = \mathcal{G}x^{(k)} + c \text{ dove } \mathcal{G} = (D - E)^{-1}F, \quad c = (D - E)^{-1}b.$$

$$r_1 : a_{11}x_1 + a_{12}x_2 = b_1 \rightarrow r_1 : x_1 = (b_1 - a_{12}x_2)/a_{11}$$

$$r_2 : a_{21}x_1 + a_{22}x_2 = b_2 \rightarrow r_2 : x_2 = (b_2 - a_{21}x_1)/a_{22}$$

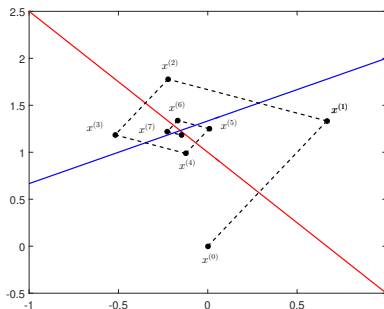
$$\text{Jacobi : } \begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)})/a_{11} \rightarrow (x_1^{(k+1)}, x_2^{(k)}) \in r_1 \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)})/a_{22} \rightarrow (x_1^{(k)}, x_2^{(k+1)}) \in r_2 \end{cases}$$



$$r_1 : a_{11}x_1 + a_{12}x_2 = b_1 \rightarrow r_1 : x_1 = (b_1 - a_{12}x_2)/a_{11}$$

$$r_2 : a_{21}x_1 + a_{22}x_2 = b_2 \rightarrow r_2 : x_2 = (b_2 - a_{21}x_1)/a_{22}$$

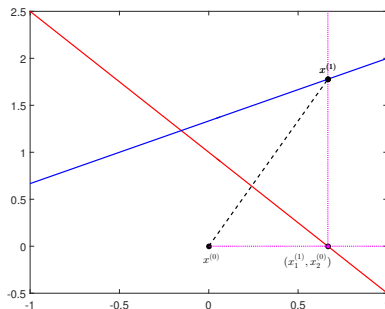
$$\text{Jacobi : } \begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)})/a_{11} \rightarrow (x_1^{(k+1)}, x_2^{(k)}) \in r_1 \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)})/a_{22} \rightarrow (x_1^{(k)}, x_2^{(k+1)}) \in r_2 \end{cases}$$



$$r_1 : a_{11}x_1 + a_{12}x_2 = b_1 \rightarrow r_1 : x_1 = (b_1 - a_{12}x_2)/a_{11}$$

$$r_2 : a_{21}x_1 + a_{22}x_2 = b_2 \rightarrow r_2 : x_2 = (b_2 - a_{21}x_1)/a_{22}$$

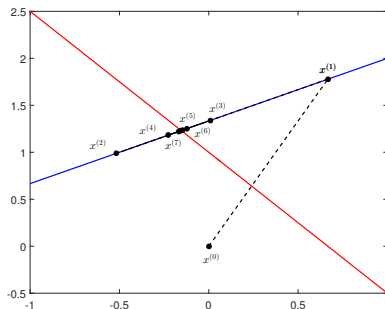
$$\text{Gauss-Seidel : } \begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)})/a_{11} & \rightarrow (x_1^{(k+1)}, x_2^{(k)}) \in r_1 \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)})/a_{22} & \rightarrow (x_1^{(k+1)}, x_2^{(k+1)}) \in r_2 \end{cases}$$



$$r_1 : a_{11}x_1 + a_{12}x_2 = b_1 \rightarrow r_1 : x_1 = (b_1 - a_{12}x_2)/a_{11}$$

$$r_2 : a_{21}x_1 + a_{22}x_2 = b_2 \rightarrow r_2 : x_2 = (b_2 - a_{21}x_1)/a_{22}$$

$$\text{Gauss-Seidel : } \begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)})/a_{11} & \rightarrow (x_1^{(k+1)}, x_2^{(k)}) \in r_1 \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)})/a_{22} & \rightarrow (x_1^{(k+1)}, x_2^{(k+1)}) \in r_2 \end{cases}$$



Teorema

Se A è strettamente diagonale dominante per righe, allora il metodo di Jacobi converge.

Dimostrazione

Ricordiamo che il metodo di Jacobi è definito come $x^{(k+1)} = \mathcal{J}x^{(k)} + c$ dove

$$\mathcal{J} = D^{-1}(E + F) = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \dots & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \dots & 0 \end{pmatrix}, c = D^{-1}b = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}.$$

Per provare la convergenza, è sufficiente mostrare che $\|\mathcal{J}\|_\infty < 1$.

<p>Ipotesi</p> $ a_{ii} > \sum_{j=1, j \neq i}^n a_{ij} $ $\forall i = 1, \dots, n$	<p>Tesi</p> $\ \mathcal{J}\ _\infty < 1$
--	---

$$\|\mathcal{J}\|_\infty = \max_{i \in \{1, \dots, n\}} \sum_{j=1, j \neq i}^n \frac{|a_{ij}|}{|a_{ii}|} < 1.$$

Teorema

Se A è strettamente diagonale dominante per righe, allora il metodo di Gauss-Seidel converge.

Teorema

Se A è simmetrica definita positiva, allora il metodo di Gauss-Seidel converge.

Osservazione

- Una singola iterazione per entrambi i metodi costa, in generale, n^2 operazioni.
- Se la matrice A è sparsa, la complessità è proporzionale al numero di elementi non nulli.
- La complessità dell'intero metodo dipende dalla tolleranza e dalla velocità con cui le iterate si avvicinano alla soluzione.

Osservazione

- Quanto più è piccolo il raggio spettrale della matrice del metodo, tanto più velocemente la successione converge.
- Si può dimostrare che $\rho(\mathcal{G}) \leq \rho(\mathcal{J})$ (il metodo di Gauss-Seidel è non meno veloce di Jacobi).
- In alcuni casi particolari, si può dimostrare che il metodo di Gauss-Seidel è più veloce del metodo di Jacobi (vedi risultato qua sotto).

Proposizione

Se $A \in \mathbb{R}^{n \times n}$ è una matrice tridiagonale non singolare con $a_{ii} \neq 0$, $i = 1, \dots, n$, allora i metodi di Jacobi e di Gauss-Seidel sono entrambi convergenti o entrambi divergenti.

Nel caso di convergenza, il metodo di Gauss-Seidel converge più velocemente di quello di Jacobi, nel senso che $\rho(\mathcal{G}) = \rho(\mathcal{J})^2$.