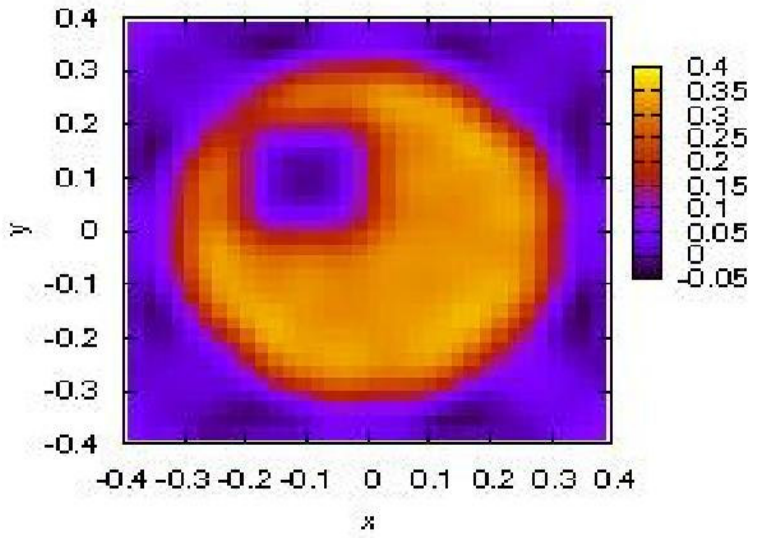
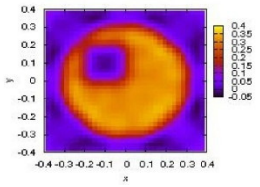


Claudio Estatico
(estatico@dima.unige.it)

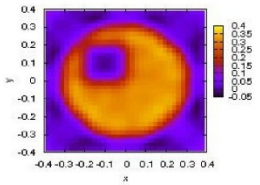
Sistemi lineari: Algoritmo di Gauss (Eliminazione Gaussiana)

Lezione basata su appunti del prof. Marco Gaviano



Sistemi lineari

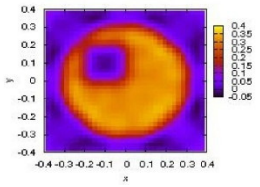
1) **Sistemi lineari.**
2) **Matrice inversa.**
3) **Sistemi triangolari. Algoritmo di sostituzione in avanti e all'indietro.**
4) **Algoritmo di Gauss. Pseudocodice e applicabilità del metodo.**
5) **Metodo del pivoting.**



Sistemi lineari

La risoluzione dei sistemi lineari si rende necessaria in ogni ambito dell'analisi numerica.

Infatti ogni problema di carattere scientifico conduce in qualche modo alla risoluzione di sistemi lineari, anche di notevoli dimensioni (ad esempio milioni di equazioni ed incognite).

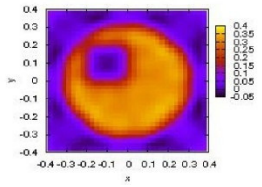


Risoluzione di sistemi lineari

Problema: dato il sistema di m equazioni in n incognite $(x_1,x_2,\ldots,x_n)$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}$$

con $a_{i,j}$ e b_i numeri reali, vogliamo determinare i valori delle incognite $(x_1,x_2,\ldots,x_n)$ che risolvono tutte le m equazioni, ossia i valori di $(x_1,x_2,\ldots,x_n)$ che sostituiti nelle singole equazioni conducono ad m uguaglianze.



In forma compatta rappresentiamo il sistema come

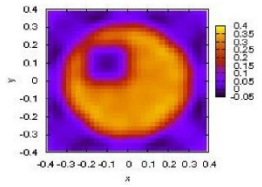
$$Ax = b$$

con A matrice $A=(a_{i,j})$ e $b=(b_i)$, i=1,2,…,m e j=1,…,n.

A è detta matrice dei coefficienti

b è il vettore dei termini noti

In seguito considereremo solo il caso m=n (si osservi che il caso m≠n può essere ricondotto a questo).

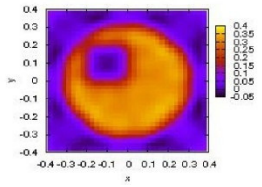


Importanza di avere algoritmi efficienti che risolvono

Ax=b

- **La risoluzione di problemi reali si effettua in molti casi attraverso la risoluzione diretta di sistemi lineari del tipo Ax=b (es. Image deblurring)**

- **La risoluzione di problemi reali può portare alla risoluzione di problemi matematici che conducono alla soluzione di sistemi lineari del tipo Ax=b (es. problemi di minimizzazione, determinazione delle radici di sistemi di equazioni non-lineari).**



Risultati noti

(Esistenza e Unicità della soluzione)

Data il sistema lineare quadrato Ax=b, se A è non singolare, ossia det(A)≠0, allora

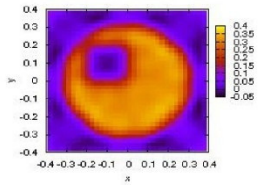
esiste una ed una sola soluzione.

(Algoritmo di risoluzione)

La ben nota regola di Cramer risolve il problema.

$$x_i = \frac{\det(A_i)}{\det(A)} \qquad i = 1,2,...,n$$

$$A = (a_1, a_2, ...a_n), \;\; A_i = (a_1, ..., a_{i-1}, b, a_{i+1}, ...a_n)$$



A parte casi di dimensione molto piccola (non rilevanti nelle applicazioni reali), la regola di Cramer è inutilizzabile. Infatti, se la matrice A è una matrice n×n, calcolando gli n+1 determinanti con il classico metodo di Laplace, la regola di Cramer richiede circa

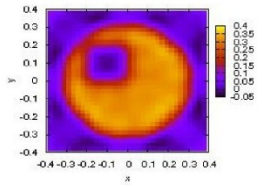
$$3(n+1)! \text{ moltiplicazioni.}$$

Disponendo di un calcolatore a 10^9 flops (1 Giga-flops, ossia 1 miliardo di operazioni al secondo) servirebbero:

per n=15 t=12 ore,

per n=20 t=3240 anni,

per n=100 $t=10^{143}$ anni…..



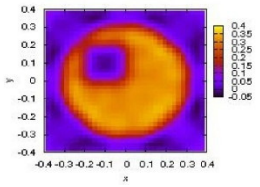
Matrice inversa

La matrice inversa di una matrice A, denotata con A^{-1}, è quella matrice tale che $A^{-1}A=AA^{-1}=I$. Essa esiste ed è unica se se A è non singolare, ossia $\det(A)\neq 0$.

Calcolo dell'inversa A^{-1} di A

Questo problema è strettamente collegato alla risoluzione di Ax=b. Infatti

- **se si ha un metodo per calcolare A^{-1}, si può allora risolvere il sistema Ax=b;**
- **se si ha un metodo per trovare la soluzione di Ax=b, allora si può calcolare A^{-1}.**



Più in dettaglio:

- **se si sa calcolare A^{-1}, allora la soluzione del sistema $Ax=b$ è data da $x = A^{-1}b$ (prodotto matrice-vettore);**

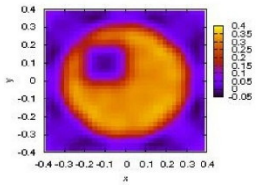
- **se si sa risolvere un sistema lineare con matrice A, allora la risoluzione degli n sistemi lineari**

$$Ax_j = e_j$$

dove $e_j=(0, 0,\ldots,1,\ldots, 0)^t$, permette di calcolare l'inversa A^{-1}.

1 in posizione j-esima

Infatti la matrice $(x_1,x_2,\ldots,x_n)$ formata dagli n vettori soluzioni è l'inversa di A.



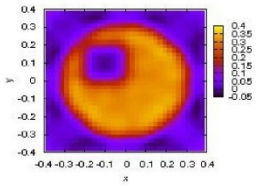
Matrici triangolari

(definizione)

Una matrice U si dice triangolare superiore se tutti i suoi elementi al di sotto della diagonale principale sono nulli.

Una matrice L si dice triangolare inferiore se tutti i suoi elementi al di sopra della diagonale principale sono nulli.

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{21n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & u_{nn} \end{bmatrix} \qquad L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}$$

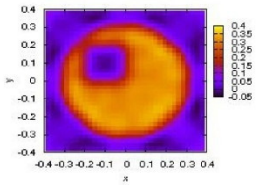


Sistemi triangolari

(definizione)

Un sistema lineare si dice triangolare (superiore, inferiore) se la matrice dei coefficienti è triangolare (superiore, inferiore)

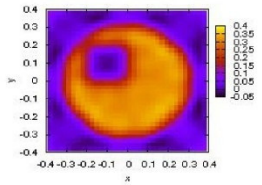
È importante avere degli algoritmi che risolvono un sistema lineare triangolare poiché la risoluzione di un sistema lineare (generico) Ax=b può essere effettuata trasformando preliminarmente tale sistema in un sistema triangolare equivalente (ossia la cui soluzione è la stessa).



Algoritmo per la soluzione di un sistema triangolare inferiore Ly=b, L non singolare (sostituzione in avanti)

$$\begin{cases} l_{11}\,y_1 & = b_1 \\ l_{21}\,y_1 + l_{22}\,y_2 & = b_2 \\ l_{31}\,y_1 + l_{32}\,y_2 + l_{33}\,y_3 & = b_3 \end{cases}$$

- **Ricavo y_1 dalla 1ª equazione**
- **Sostituisco y_1 nella 2ª equazione**
- **Ricavo y_2 dalla 2ª equazione**
- **Sostituisco y_1 e y_2 nella 3ª equazione**
- **Ricavo y_3 dalla 3ª equazione**



Algoritmo di sostituzione in avanti per la risoluzione di un sistema triangolare inferiore e non singolare Ly=b

Pseudocodice:

$y_1=b_1/l_{11}$

for i=2,…,n

  $y_i=b_i$

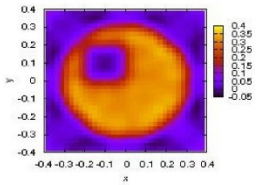
  for j=1,…,i-1

    $y_i=y_i-l_{ij}y_j$

  end

  $y_i=y_i/l_{ii}$

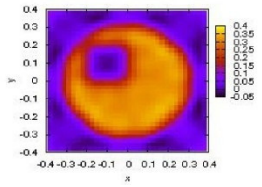
end



Algoritmo per la soluzione di un sistema triangolare superiore Ux=y, U non singolare (sostituz. all'indietro)

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + u_{13}x_3 = y_1 \\ u_{22}x_2 + u_{23}x_3 = y_2 \\ u_{33}x_3 = y_3 \end{cases}$$

- **Ricavo x_3 dalla 3ª equazione**
- **Sostituisco x_3 nella 2ª equazione**
- **Ricavo x_2 dalla 2ª equazione**
- **Sostituisco x_3 e x_2 nella 1ª equazione**
- **Ricavo x_1 dalla 1ª equazione**



Algoritmo di sostituzione all'indietro per la risoluzione di un sistema triangolare superiore non singolare Ly=b

Pseudocodice:

$x_n = y_n / u_{nn}$

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

  $x_i = y_i$

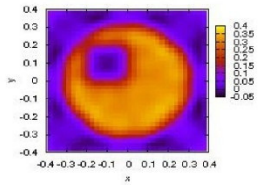
  for $j = i+1,\ldots,n$

    $x_i = x_i - u_{ij} x_j$

  end

  $x_i = x_i / u_{ii}$

end

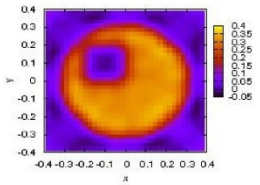


Complessità computazionale di un algoritmo

Un modo abbastanza usuale di valutare l'efficienza di un algoritmo è quello di calcolare il numero di operazioni che vengono eseguite durante la sua esecuzione al variare dei dati input del problema.

Le operazioni possono essere di vario tipo: operazione aritmetiche, confronto tra numeri, lettura di dati, ecc.

Nel caso degli algoritmi numerici si considerano il numero di operazioni aritmetiche (+ , – , * , /). In prima analisi si considerano solo le moltiplicazioni e le divisioni, poiché queste operazioni sono considerate "più costose" rispetto alle addizioni e sottrazioni.

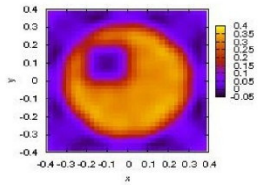


La complessità computazionale dell'algoritmo di sostituzione in avanti per sistemi triangolari è

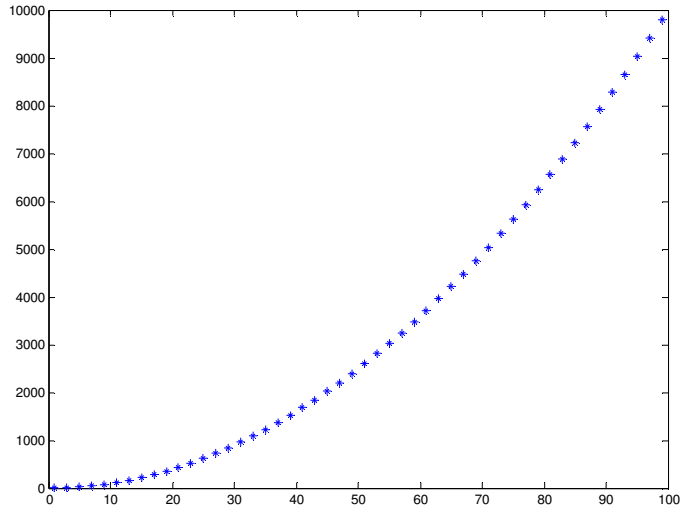
$$operazioni = 1 + \sum_{i=2}^{n} i = \sum_{i=1}^{n} i = \frac{1}{2} n(n+1) \cong \frac{1}{2} n^2$$

Al crescere di n (la dimensione del problema)
il numero di operazioni ('*' e '/') cresce come n^2 ;
si dice che la complessità dell'algoritmo è polinomiale

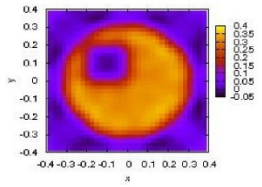
Posto il tempo per l'esecuzione di una moltiplicazione uguale a τ
il tempo di esecuzione dell'algoritmo sarà: $t \approx \tau \cdot (n^2/2)$



Il grafico della funzione "numero di operazioni eseguite dall'algoritmo al variare della dimensione del problema" è (approssimativamente) il seguente



Per il tempo di esecuzione dell'algoritmo, il grafico sarà dello stesso tipo

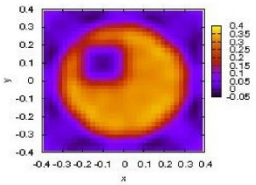


Metodo di Gauss

L'algoritmo di Gauss è un metodo diretto per risolvere sistemi lineari Ax=b, A matrice, n×n, det(A)≠0, ovvero

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = b_1 = a_{1,n+1} \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n = b_2 = a_{2,n+1} \\ \ldots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n = b_n = a_{n,n+1} \end{cases}$$

I coefficienti del sistema (ossia gli elementi di A) e i termini noti (ossia gli elementi di b) possono essere memorizzati in una stessa matrice, detta matrice completa del sistema (si ricordi il Teorema di Rouché-Capelli).



Idea di base del metodo di Gauss "naive" nel caso n=3

sistema lineare (1)

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,1}^{(1)} x_1 + a_{2,2}^{(1)} x_2 + a_{2,3}^{(1)} x_3 = a_{2,4}^{(1)} \\ a_{3,1}^{(1)} x_1 + a_{3,2}^{(1)} x_2 + a_{3,3}^{(1)} x_3 = a_{3,4}^{(1)} \end{cases}$$

Ricavo x_1 dalla 1ª equazione

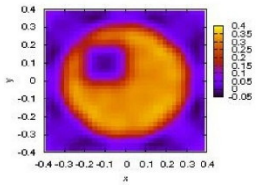
$$x_1 = \frac{a_{1,4}^{(1)} - a_{1,2}^{(1)} x_2 - a_{1,3}^{(1)} x_3}{a_{1,1}^{(1)}}$$

e lo sostituisco nella 2ª e 3ª equazione, ottenendo

sistema lineare (2)

compaiono degli zeri nella prima colonna

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,2}^{(2)} x_2 + a_{2,3}^{(2)} x_3 = a_{2,4}^{(2)} \\ a_{3,2}^{(2)} x_2 + a_{3,3}^{(2)} x_3 = a_{3,4}^{(2)} \end{cases}$$



Da

sistema lineare (2)

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,2}^{(2)} x_2 + a_{2,3}^{(2)} x_3 = a_{2,4}^{(2)} \\ a_{3,2}^{(2)} x_2 + a_{3,3}^{(2)} x_3 = a_{3,4}^{(2)} \end{cases}$$

Ricavo x_2 dalla 2ª equazione

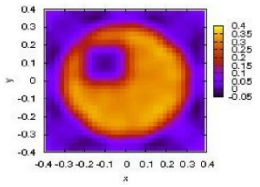
$$x_2 = \frac{a_{2,4}^{(2)} - a_{2,3}^{(2)} x_3}{a_{2,2}^{(2)}}$$

e lo sostituisco nella 3ª equazione, ottenendo

sistema lineare (3)

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,2}^{(2)} x_2 + a_{2,3}^{(2)} x_3 = a_{2,4}^{(2)} \\ a_{3,3}^{(3)} x_3 = a_{3,4}^{(3)} \end{cases}$$

compaiono degli zeri nella seconda colonna



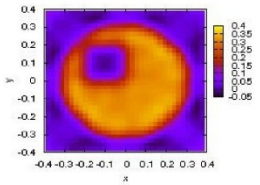
Si osservi che i passaggi sono possibili se

$$a_{1,1}^{(1)} \; e \; a_{2,2}^{(2)} \quad sono \neq 0$$

Il sistema finale è triangolare superiore

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,2}^{(2)} x_2 + a_{2,3}^{(2)} x_3 = a_{2,4}^{(2)} \\ a_{3,3}^{(3)} x_3 = a_{3,4}^{(3)} \end{cases}$$

Se $a_{3,3}^{(3)} \neq 0$ **la soluzione di Ax=b la si trova con l'algoritmo (già visto) che risolve un sistema triangolare superiore ("sostituzione all'indietro").**



Il passaggio dal sistema (1) al sistema (2) si può scrivere

moltiplico la 1ª riga del sist. (1) per

$$m_{2,1} = a_{2,1}^{(1)} / a_{1,1}^{(1)}$$

e la sottraggo alla 2ª; ottengo così:

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,1}^{(1)} x_1 + a_{2,2}^{(1)} x_2 + a_{2,3}^{(1)} x_3 = a_{2,4}^{(1)} \\ a_{3,1}^{(1)} x_1 + a_{3,2}^{(1)} x_2 + a_{3,3}^{(1)} x_3 = a_{3,4}^{(1)} \end{cases}$$

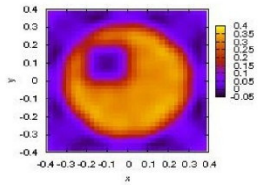
moltiplico la 2ª riga del sist. (1) per

$$m_{3,1} = a_{3,1}^{(1)} / a_{1,1}^{(1)}$$

e la sottraggo alla 3ª; ottengo così:

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + a_{1,3}^{(1)} x_3 = a_{1,4}^{(1)} \\ a_{2,2}^{(2)} x_2 + a_{2,3}^{(2)} x_3 = a_{2,4}^{(2)} \\ a_{3,2}^{(3)} x_2 + a_{3,3}^{(3)} x_3 = a_{3,4}^{(3)} \end{cases}$$

Cioè il sistema (2) è ottenuto sostituendo opportunamente ad una riga, una combinazione lineare della riga stessa con un'altra riga. Il sistema (2) ha la stessa soluzione del sistema (1).



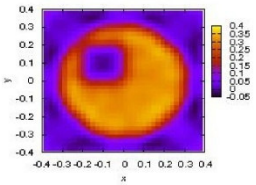
Osservazioni

- **La soluzione del sistema triangolare superiore è la stessa soluzione del sistema iniziale perché i sistemi lineari (0), (1) e (2) sono equivalenti tra loro.**

- **Il nome di Gauss "naive" deriva dal fatto che il metodo è applicabile se e solo se**

$$a_{1,1}^{(1)},\ a_{2,2}^{(2)}\ \ e\ a_{3,3}^{(3)}\ \ sono \neq 0$$

(si osservi che $a_{1,1}^{(1)} \neq 0$ **non è condizione necessaria alla risolvibilità del sistema…).**



Matrice principale di ordine k di una matrice A n×n, (caso n=3) (definizione)

Matrice principale di ordine 1

$$A_1 = [a_{1,1}]$$

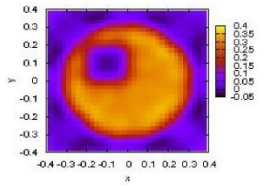
$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

Matrice principale di ordine 2

$$A_2 = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$$

Matrice principale di ordine 3

$$A_3 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$



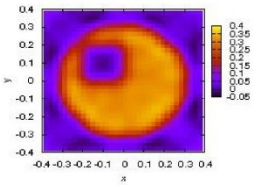
Minori principali (definizione)

I minori principali di ordine k (k=1,2,…,n) di una matrice A, n×n, sono i determinanti det(A_k) delle sue matrici principali

Teorema

Dato un sistema Ax=b, se tutti i suoi minori principali sono ≠ 0, allora si può applicare l'algoritmo di Gauss 'naive'.

L'algoritmo di Gauss 'naive' si generalizza in maniera banale al caso generale n × n.



Gauss 'naive' nel caso generale

Iterazione 1: si ricava x_1 dalla 1ª equazione e lo si sostituisce nella 2ª, 3ª, …, n-esima equazione

per i=2,…, n e j=2,…,n+1:

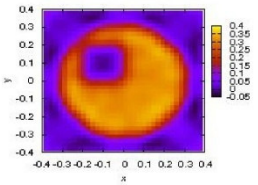
$$a_{i,j}^{(2)} = a_{i,j}^{(1)} - m_{i,1} a_{1,j}^{(1)}$$

$$con \quad m_{i,1} = a_{i,1}^{(1)} / a_{1,1}^{(1)}$$

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + \ldots + a_{1,n}^{(1)} x_n = a_{1,n+1}^{(1)} \\ a_{2,1}^{(1)} x_1 + a_{2,2}^{(1)} x_2 + \ldots + a_{2,n}^{(1)} x_n = a_{2,n+1}^{(1)} \\ \qquad\qquad \ldots \\ a_{n,1}^{(1)} x_1 + a_{n,2}^{(1)} x_2 + \ldots + a_{n,n}^{(1)} x_n = a_{n,n+1}^{(1)} \end{cases}$$

ottengo

$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + \ldots + a_{1,n}^{(1)} x_n = a_{1,n+1}^{(1)} \\ \qquad\qquad a_{2,2}^{(2)} x_2 + \ldots + a_{2,n}^{(2)} x_n = a_{2,n+1}^{(2)} \\ \qquad\qquad\qquad \ldots \\ \qquad\qquad a_{n,2}^{(2)} x_2 + \ldots + a_{n,n}^{(2)} x_n = a_{n,n+1}^{(2)} \end{cases}$$



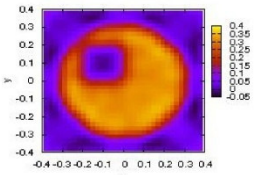
Allo stesso modo si procede con:

- **iterazione 2: si ricava x_2 dalla 2ª equazione e lo si sostituisce nella 3ª, 4ª, …, n-esima equazione**
- **…**
- **iterazione n-1: si ricava x_{n-1} dalla (n-1)ª equazione e lo si sostituisce nella n-esima equazione**

Si ottiene alla fine un sistema triangolare superiore equivalente a quello iniziale, che si risolve facilmente

Coefficienti annullati dalle combinazioni lineari di righe effettuate dall'intero algoritmo

$$
\begin{aligned}
a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + \ldots + a_{1,n}^{(1)} x_n &= a_{1,n+1}^{(1)} \\
a_{2,2}^{(2)} x_2 + \ldots + a_{2,n}^{(2)} x_n &= a_{2,n+1}^{(2)} \\
\ldots & \\
a_{n,n}^{(n)} x_n &= a_{n,n+1}^{(n)}
\end{aligned}
$$



Pseudocodice dell'algoritmo di Gauss "naive" (triangolarizzazione del sistema Ax=b)

Pseudocodice:

for k=1,2,…,n-1

for i=k+1,…,n

$$m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$$

for j=k+1,…,n+1

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$$

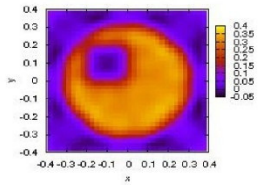
end

end

end

In questo ciclo si eliminano successivamente le variabili

$x_1, x_2, \ldots x_{n-1}$



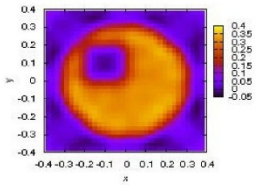
Osservazioni

- **I valori m_{ik} vengono chiamati moltiplicatori**
- **Solo se i minori principali di A sono tutti $\neq 0$ allora si può applicare l'algoritmo "naive"**

Numero di moltiplicazioni e divisioni per la risoluzione del sistema Ax=b mediante Gauss

Risultato fondamentale:

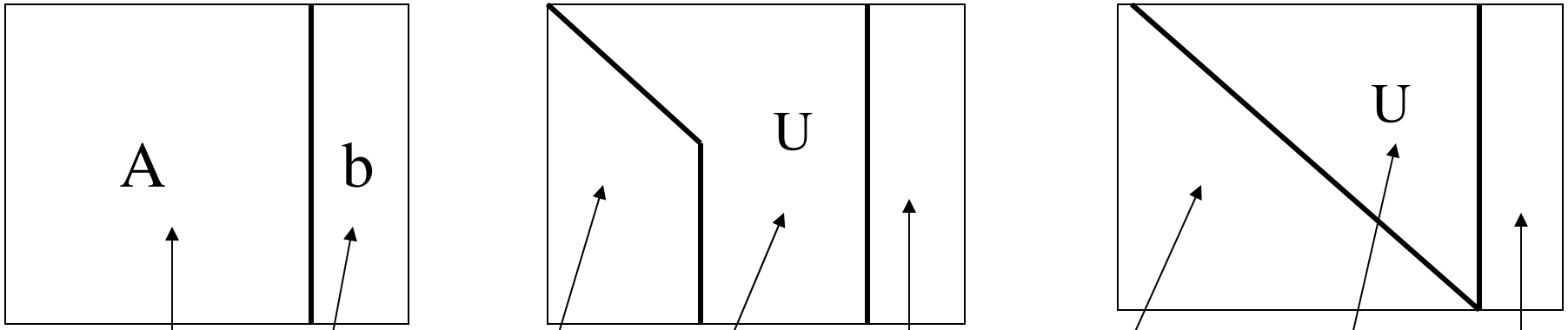
il costo computazionale dell'algoritmo di Gauss "naive" è polinomiale, dello stesso ordine di n^3



Schema di funzionamento dell'algoritmo per la triangolarizzazione del sistema lineare Ax=b

un solo array di lavoro

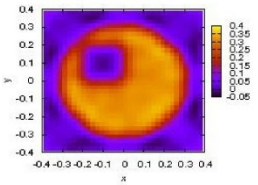
al termine dell'esecuzione



contiene inizialmente A ed il vettore b

zeri, coefficienti, termini noti dei successivi sistemi

zeri, coefficienti, termini noti del sistema triangolare superiore finale



Risoluzione di un sistema lineare Ax=b in cui l'unica ipotesi è che A sia non singolare, ossia det(A) ≠ 0,

(in particolare non si richiede che i minori principali siano ≠ 0, esempio)

soluzione esatta

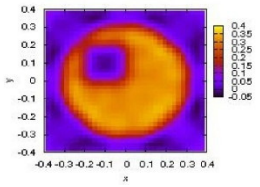
$x_1 = 1,\ x_2 = -1,\ x_3 = 1$

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 + x_2 + 2x_3 = 2 \\ x_1 + 2x_2 + 2x_3 = 1 \end{cases}$$

1ª iterazione

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ + x_3 = 1 \\ + x_2 + x_3 = 0 \end{cases}$$

a questo punto non possiamo applicare Gauss 'naive'

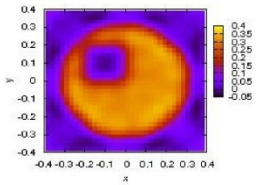


Se scambiamo la 2ª riga con la 3ª otteniamo

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_2 + x_3 = 0 \\ x_3 = 1 \end{cases}$$

Possiamo applicare la 2ª iterazione che non cambia il sistema. L'algoritmo può così procedere fino al termine e si ottiene quindi la soluzione del sistema cercata.

Lo scambio di righe è una tecnica che, inserita nell'algoritmo di Gauss "naive", lo rende sempre applicabile.

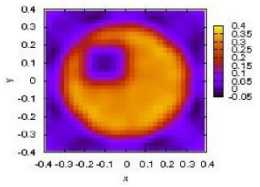


Applicazione di Gauss con scambio di righe (pivoting parziale)

Iterazione 1

Tra tutte le righe, si sceglie una riga in cui il coefficiente di x_1 (pivot) è massimo in valore assoluto. Tale riga (se non è già la 1ª) viene scambiata con la 1ª riga. Si applica ora la prima iterazione dell'algoritmo di Gauss

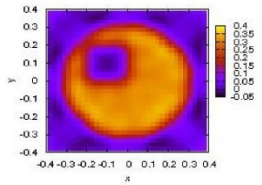
$$\begin{cases} a_{1,1}^{(1)} x_1 + a_{1,2}^{(1)} x_2 + \ldots + a_{1,n}^{(1)} x_n = a_{1,n+1}^{(1)} \\ a_{2,1}^{(1)} x_1 + a_{2,2}^{(1)} x_2 + \ldots + a_{2,n}^{(1)} x_n = a_{2,n+1}^{(1)} \\ \qquad\qquad\qquad \ldots \\ a_{n,1}^{(1)} x_1 + a_{n,2}^{(1)} x_2 + \ldots + a_{n,n}^{(1)} x_n = a_{n,n+1}^{(1)} \end{cases}$$



Iterazione 2

Tra la 2ª e la 3ª, …, la n-esima riga, si sceglie una riga in cui il coefficiente di x_2 (pivot) è massimo in valore assoluto. Tale riga (se non è già la 2ª) viene scambiata con la 2ª riga. Si applica ora la seconda iterazione dell'algoritmo di Gauss

$$\begin{cases} a_{1,1}^{(1)}x_1 + a_{1,2}^{(1)}x_2 + \ldots + a_{1,n}^{(1)}x_n = a_{1,n+1}^{(1)} \\ \boxed{a_{2,2}^{(2)}x_2} + \ldots + a_{2,n}^{(2)}x_n = a_{2,n+1}^{(2)} \\ \boxed{\ldots} \\ \boxed{a_{n,2}^{(2)}x_2} + \ldots + a_{n,n}^{(2)}x_n = a_{n,n+1}^{(2)} \end{cases}$$

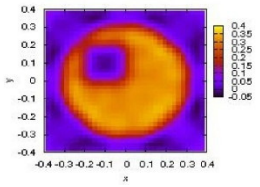


Osservazione

Il pivoting parziale rende sempre applicabile l'algoritmo di Gauss, con la sola ipotesi che $\det(A) \neq 0$

Applicazione di Gauss con scambio di righe e colonne (pivoting totale)

Con questa tecnica si scambiano righe e colonne in modo che il pivot sia il massimo elemento possibile in valore assoluto tra tutti quelli che possono essere presi in considerazione



Osservazione

Generalmente si utilizza il pivoting parziale poiché la sua applicazione è meno costosa.

La tecnica del pivoting oltre a rendere sempre applicabile l'algoritmo di Gauss, generalmente (…) ne migliora l'accuratezza.