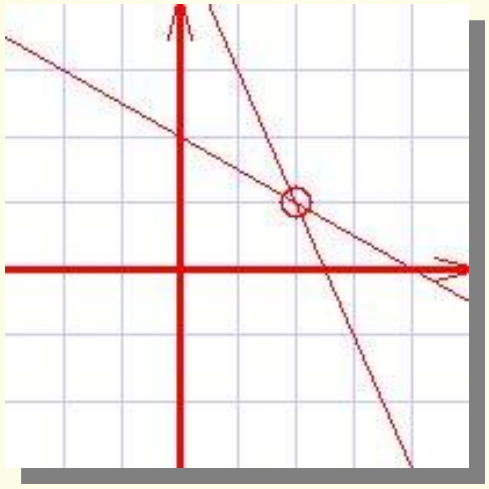
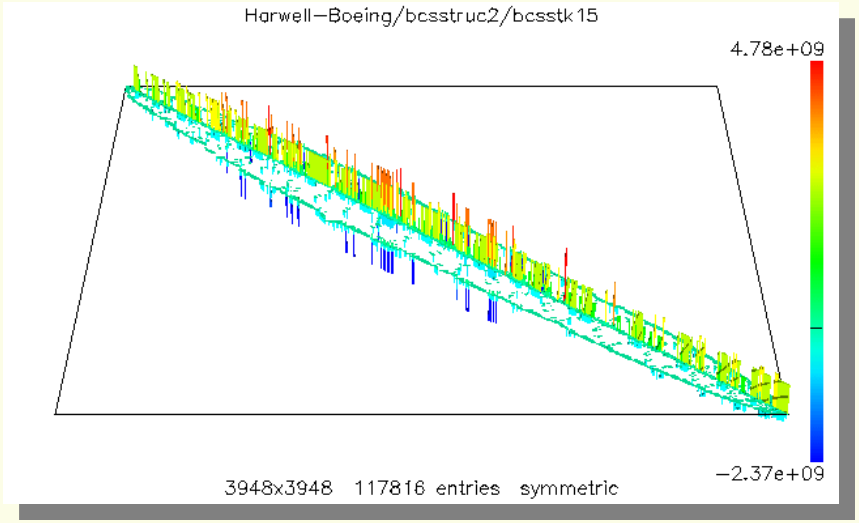
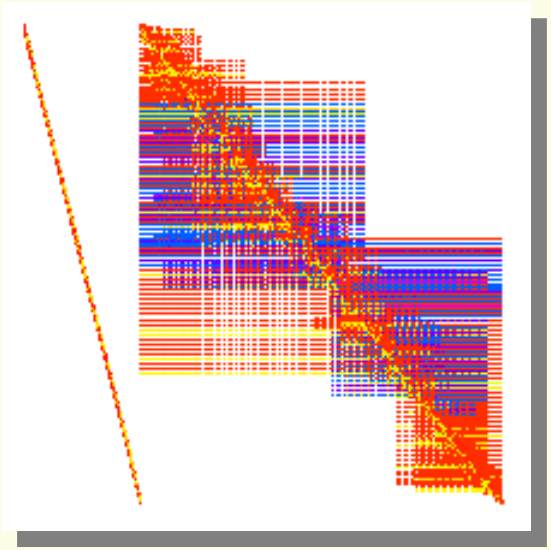
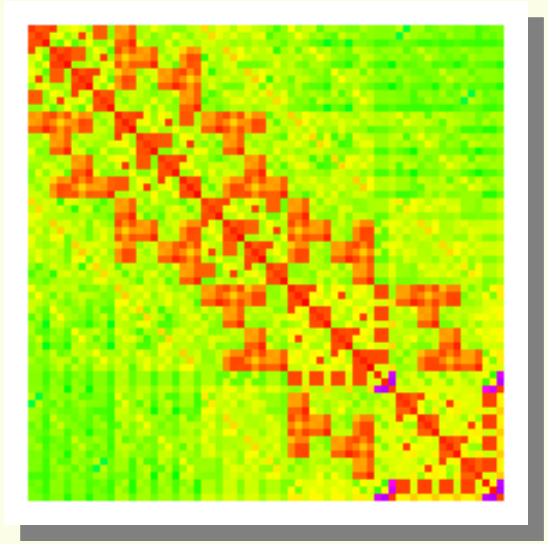
Sistemi lineari









Sistemi lineari

Un sistema lineare:

$$\begin{cases} 3x_1 & +2x_2 & +x_3 & = & 14 \\ -x_1 & +2x_2 & & = & 1 \\ x_1 & +x_2 & -2x_3 & = & 3 \end{cases}$$

viene più comodamente espresso in forma matriciale: $\mathbf{Ax} = \mathbf{b}$

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1 \\ -1 & 2 & 0 \\ 1 & 1 & -2 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 14 \\ 1 \\ 3 \end{bmatrix}$$

$\mathbf{A}$ è la matrice dei coefficienti del sistema, $\mathbf{x}$ il vettore delle incognite, $\mathbf{b}$ il vettore dei termini noti.

N.B.: nel seguito di questo capitolo (**L3**) ci occuperemo di sistemi quadrati, ovvero di sistemi in cui il numero delle incognite è uguale a quello delle equazioni.

Sistemi lineari

> L'importanza dell'algebra lineare in tutti i settori del calcolo numerico difficilmente può essere sopravvalutata.
>
> Guido Buzzi Ferraris, 1998

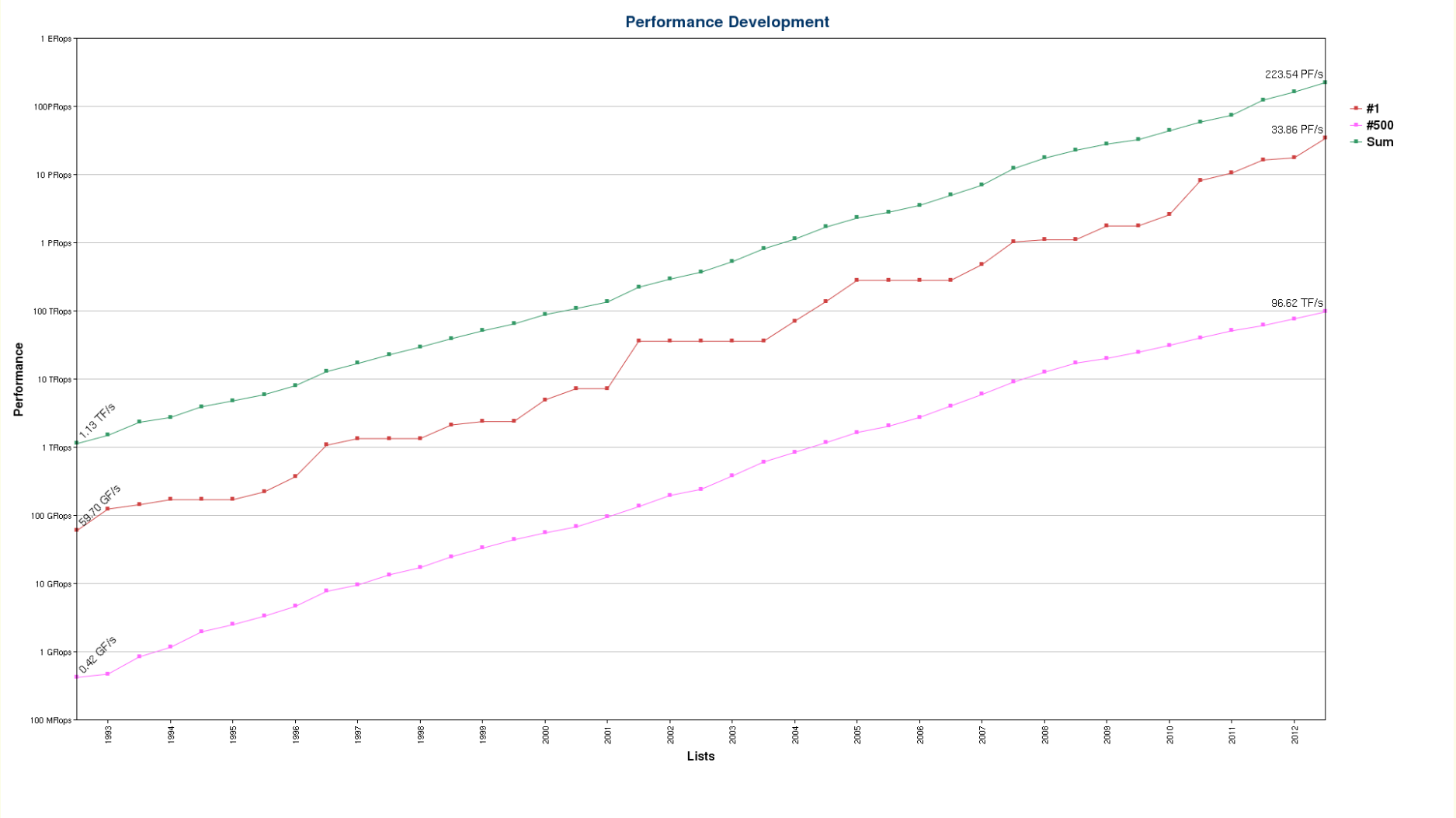
La presenza di sistemi lineari nella matematica applicata è assolutamente di primo rilievo.

I sistemi lineari compaiono direttamente nella formulazione di problemi reali ma ancor più spesso nascono come passaggio intermedio nella risoluzione numerica di altri problemi. La stessa potenza di calcolo dei supercomputer è misurata in Gflop/s e Tflop/s nella risoluzione di un sistema lineare (libreria numerica LINPACK).

Test Linpack: Il computer più veloce del mondo Tianhe-2 della China's National University of Defense Technology raggiunge i 33.86 PFlops (16,000 nodi con 3,120,000 processori, 1,024,000 GB di RAM, 17.8 MW potenza assorbita. Dimensione massima sistema lineare a matrice piena: 9,960,000 (dati aggiornati al Giugno 2013).

La famiglia di processori Intel i7 ha una potenza di 10-50 Gflop/s.

Prestazioni dei 500 top computer



Tipici utilizzi dei 500 top computer

As supercomputers represent a dramatic leap forward in terms of simulation precision and calculation speed, they can be applied to a variety of fields that use computational science. They hold the promise of contributing to the generation of world-class breakthroughs such as:

• Analyzing the **behavior of nanomaterials** through **simulations** and contributing to the early development of such next-generation semiconductor materials, particularly nanowires and carbon nanotubes, that are expected to lead to future fast-response, low-power devices.

• **Predicting** which compounds, from among a massive number of **drug candidate molecules**, will prevent illnesses by binding with active regions on the proteins that cause illnesses, as a way to reduce drug development times and costs (pharmaceutical applications).

• **Simulating the actions of atoms and electrons** in dye-sensitized **solar cells** to contribute to the development of solar cells with higher energy-conversion efficiency.

• **Simulating seismic wave** propagation, strong motion, and **tsunamis** to predict the effects they will have on human-made structures; predicting the extent of earthquake-impact zones for disaster prevention purposes; and contributing to the design of quake-resistant structures.

• Conducting high-resolution (400-m) **simulations** of **atmospheric circulation models** to provide detailed predictions of weather phenomena that elucidate localized effects, such as cloudbursts.

http://www.riken.jp/engn/r-world/info/release/press/2011/111102/index.html

Sistemi lineari

È richiesta la risoluzione numerica di un sistema lineare nei seguenti casi:

- interpolazione polinomiale,
- sistemi algebrici non lineari,
- equazioni differenziali ordinarie e a derivate parziali,
- equazioni integrali,
- ottimizzazione multidimensionale,
- regressioni lineari e non lineari,
- ……

Richiami di Analisi

- Un sistema lineare $\mathbf{Ax} = \mathbf{b}$ è detto **omogeneo** se $\mathbf{b} = \mathbf{0}$ ovvero se tutti gli elementi del vettore dei termini noti sono nulli.
- Un sistema lineare **non è omogeneo** se $\mathbf{b} \neq \mathbf{0}$.

Teorema

Dato un sistema lineare $\mathbf{Ax} = \mathbf{b}$ in n equazioni, le seguenti affermazioni sono equivalenti:

- Per ogni $\mathbf{b}$ esiste una soluzione
- Per ogni $\mathbf{b}$ esiste una ed una sola soluzione
- Le righe e le colonne della matrice $\mathbf{A}$ sono linearmente indipendenti
- Il sistema omogeneo $\mathbf{Ax} = \mathbf{0}$ ha una sola soluzione: $\mathbf{x} = \mathbf{0}$.
- $\mathbf{det(A)} \neq \mathbf{0}$
- $\mathbf{A}^{-1}$ esiste (cioè $\mathbf{A}$ non è singolare)
- $\text{Rango}(\mathbf{A}) = n$

Classificazione sistemi lineari

DIMENSIONE

Uno dei criteri per classificare un sistema lineare è la sua dimensione.

Si parla di sistemi di piccole, medie e grandi dimensioni.

Tali termini sono funzione delle capacità medie dei computer le quali sono funzione del periodo storico specifico in cui si opera.

L'occupazione di memoria di un sistema lineare è principalmente funzione delle dimensioni della matrice $\mathbf{A}$. Se $\mathbf{A}$ è di ordine $n \times n$ e si lavora in doppia precisione l'occupazione di memoria è pari a: $8n^2$ byte. Conseguentemente un sistema lineare costituito da **8000** equazioni occupa in doppia precisione 512 MB.

Soltanto 20-25 anni fa si considerava di **medie** dimensioni un sistema di 30-50 equazioni e **grande** un sistema di 100 equazioni.

Oggi utilizzando un comune personal computer si inizia a parlare di sistemi di **grandi** dimensioni quando si abbia qualche migliaio di equazioni (se la matrice $\mathbf{A}$ è densa) o altrimenti anche qualche centinaio di migliaia di equazioni (se la matrice $\mathbf{A}$ è sparsa).

Classificazione sistemi lineari

DENSITÀ o SPARSITÀ

Un altro criterio di classificazione è relativo alla struttura della matrice $\mathbf{A}$.

Se la matrice $\mathbf{A}$ è praticamente piena o comunque una significativa percentuale della matrice ha elementi non nulli (70-80%) il sistema è detto **denso**.

Se la matrice $\mathbf{A}$ ha un'elevata percentuale di elementi nulli (>80-90%) il sistema è detto **sparso**.

La sparsità può essere a sua volta **non strutturata** o **strutturata** (si parla di matrice diagonale, a banda, a bande, a blocchi, …).

Nel caso di matrici sparse è possibile memorizzare i soli termini non nulli riducendo significativamente l'allocazione totale di memoria. Anche i conseguenti tempi di risoluzione possono subire una drastica riduzione nel caso di matrici sparse se confrontati con corrispondenti matrici dense. Le differenze di tempi di CPU per la soluzione del sistema diventano significative allorché le dimensioni della matrice $\mathbf{A}$ sono elevate (migliaia di equazioni).

Classificazione algoritmi risolutivi

METODI DIRETTI

Sono metodi che impiegano un numero finito di passi (operazioni) per ottenere la soluzione. Questi metodi identificano l'esatta soluzione x del problema purché tutte le operazioni aritmetiche siano eseguite in modo esatto. Il metodo più usato è quello di eliminazione di Gauss (nella versione moderna implementato come fattorizzazione tramite le trasformazioni di Gauss).

METODI ITERATIVI

La loro applicazione principale è per sistemi di grandi dimensioni (principalmente quelli prodotti dalla discretizzazione delle equazioni differenziali alle derivate parziali). La soluzione del sistema passa attraverso formule iterative. Si parte da un valore di primo tentativo e si itera. Se il metodo converge si approssima vieppiù la soluzione.

Risoluzione manuale di un sistema

Si desidera illustrare il metodo di risoluzione manuale di un sistema algebrico. Iniziamo con il seguente sistema lineare:

$$\begin{aligned} 3x_1 \quad -2x_2 \quad -x_3 \quad &= \quad 0 \quad && (E1) \\ 6x_1 \quad -2x_2 \quad +2x_3 \quad &= \quad 6 \quad && (E2) \\ -9x_1 \quad +7x_2 \quad +x_3 \quad &= \quad -1 \quad && (E3) \end{aligned}$$

(1) Si elimina x_1 dalle equazioni $E2$ ed $E3$ sottraendo il doppio di $E1$ da $E2$ e -3 volte $E1$ da $E3$. Si ottiene:

$$\begin{aligned} 3x_1 \quad -2x_2 \quad -x_3 \quad &= \quad 0 \quad && (E1) \\ \quad 2x_2 \quad +4x_3 \quad &= \quad 6 \quad && (E2) \\ \quad x_2 \quad -2x_3 \quad &= \quad -1 \quad && (E3) \end{aligned}$$

N.B.: Il vettore dei termini noti non è cambiato soltanto perché $b_1 = 0$.

Risoluzione manuale di un sistema

(2) Si elimina x_2 dall'equazione $E3$ sottraendo ½ volte $E2$ da $E3$. Si ottiene:

$$\begin{aligned} 3x_1 \quad -2x_2 \quad -x_3 &= 0 \qquad (E1) \\ 2x_2 \quad +4x_3 &= 6 \qquad (E2) \\ -4x_3 &= -4 \qquad (E3) \end{aligned}$$

N.B.: ora il vettore dei termini noti è stato modificato in b_3 in quanto $b_2 \neq 0$.

(3) Si risolve all'indietro il sistema (*back substitution*) e si ottengono: x_3, x_2, x_1.

$$x_3 = x_2 = x_1 = 1$$

Introduzione all'eliminazione di Gauss

Il sistema lineare originale $\mathbf{Ax} = \mathbf{b}$ ha la seguente struttura matriciale:

$$\mathbf{A} = \begin{bmatrix} 3 & -2 & -1 \\ 6 & -2 & 2 \\ -9 & 7 & 1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 6 \\ -1 \end{bmatrix}$$

Spesso si compatta la scrittura del problema in un'unica matrice (*augmented matrix*):

$$\left[\mathbf{A}|\mathbf{b}\right] = \left[\begin{array}{ccc|c} 3 & -2 & -1 & 0 \\ 6 & -2 & 2 & 6 \\ -9 & 7 & 1 & -1 \end{array}\right]$$

Introduzione all'eliminazione di Gauss

Le operazioni da effettuare per la risoluzione del sistema sono:

(1) Si elimina x_1 dalle equazioni 2 e 3. Si moltiplica a tal fine la riga 1 per 2 e la si sottrae alla riga 2; quindi si moltiplica la riga 1 per –3 e la si sottrae alla riga 3:

$$\left[\mathbf{A}\middle|\mathbf{b}\right] = \left[\begin{array}{ccc|c} 3 & -2 & -1 & 0 \\ 0 & 2 & 4 & 6 \\ 0 & 1 & -2 & -1 \end{array}\right]$$

(2) Si elimina x_2 dall'equazione 3. Si moltiplica a tal fine la riga 2 per ½ e la si sottrae alla riga 3:

$$\left[\mathbf{A}\middle|\mathbf{b}\right] = \left[\begin{array}{ccc|c} 3 & -2 & -1 & 0 \\ 0 & 2 & 4 & 6 \\ 0 & 0 & -4 & -4 \end{array}\right]$$

(3) Si procede con la sostituzione all'indietro determinando: x_3, x_2, x_1.

Introduzione all'eliminazione di Gauss

Partendo dalla struttura generale di un sistema lineare:

$$\left[\mathbf{A}\middle|\mathbf{b}\right]=\left[\begin{array}{ccc|c} a_{1,1}^{(0)} & \cdots & a_{1,n}^{(0)} & b_1^{(0)} \\ \vdots & \ddots & \vdots & \vdots \\ a_{n,1}^{(0)} & \cdots & a_{n,n}^{(0)} & b_n^{(0)} \end{array}\right]$$

Dopo n-1 passaggi si ottiene il sistema equivalente:

$$\left[\mathbf{A}\middle|\mathbf{b}\right]=\left[\begin{array}{cccc|c} a_{1,1}^{(0)} & \cdots & & a_{1,n}^{(0)} & b_1^{(0)} \\ 0 & a_{2,2}^{(1)} & & a_{2,n}^{(1)} & \vdots \\ \vdots & & \ddots & \vdots & b_{n-1}^{(n-2)} \\ 0 & \cdots & 0 & a_{n,n}^{(n-1)} & b_n^{(n-1)} \end{array}\right]$$

Introduzione all'eliminazione di Gauss

Si è cioè ottenuto un sistema lineare triangolare destro (triangolare alto) $\mathbf{Rx} = \mathbf{c}$:

$$\begin{bmatrix} r_{1,1} & \dots & & r_{1,n} \\ 0 & r_{2,2} & & r_{2,n} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & r_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Risolvendo il sistema triangolare destro (triangolare alto) all'indietro si ottengono: x_n, quindi x_{n-1} fino a x_1:

$$x_n = \frac{c_n}{r_{n,n}}$$

$$x_k = \frac{c_k - \left(r_{k,k+1} x_{k+1} + \dots + r_{k,n} x_n \right)}{r_{k,k}} \qquad k = n-1, \dots, 1$$

Matrici triangolari destre

Definizione: se una matrice quadrata ha tutti i coefficienti nulli sotto la diagonale principale viene detta **triangolare destra** o **triangolare alta** e viene solitamente indicata con: $\mathbf{R}$ o $\mathbf{U}$ (**R**ight, **U**pper).

N.B.: se uno dei coefficienti sulla diagonale della matrice $\mathbf{R}$ è nullo allora la matrice è **singolare**.

Matrici triangolari destre

Algoritmo

Risoluzione di un sistema lineare avente una matrice triangolare destra.

Input: dimensione n della matrice, coefficienti $r_{i,j}$ matrice destra, termini noti b_i.

Output: nel vettore $\mathbf{b}$ vi è la soluzione del sistema.

Flop: entrambi gli algoritmi richiedono $n^2/2$ flop.

Versione per riga

```
for i = n to 1 step -1
   for k = i+1 to n
      b(i) = b(i) - r(i,k) * b(k)
   next k
   b(i) = b(i) / r(i,i)
next i
```

Versione per colonna

```
for i = n to 1 step -1
   b(i) = b(i) / r(i,i)
   for k = i-1 to 1 step -1
      b(k) = b(k) - r(k,i) * b(i)
   next k
next i
```

Matrici triangolari sinistre

Definizione: se una matrice quadrata ha tutti i coefficienti nulli sopra la diagonale principale viene detta **triangolare sinistra** o **triangolare bassa** e viene solitamente indicata con: $\mathbf{L}$ (**L**eft, **L**ower).

N.B.: se uno dei coefficienti sulla diagonale della matrice $\mathbf{L}$ è nullo allora la matrice è **singolare**.

Matrici triangolari sinistre

Algoritmo

Risoluzione di un sistema lineare avente una matrice triangolare sinistra.

Input: dimensione n della matrice, coefficienti $l_{i,j}$ matrice sinistra, termini noti b_i.

Output: nel vettore $\mathbf{b}$ vi è la soluzione del sistema.

Flop: entrambi gli algoritmi richiedono $n^2/2$ flop.

Versione per riga

```
for i = 1 to n
   for k = 1 to i-1
      b(i) = b(i) - L(i,k) * b(k)
   next k
   b(i) = b(i) / L(i,i)
next i
```

Versione per colonna

```
for i = 1 to n
   b(i) = b(i) / L(i,i)
   for k = i+1 to n
      b(k) = b(k) - L(k,i) * b(i)
   next k
next i
```

Matrici triangolari sinistre e destre

N.B.: dato che la trasposta di una matrice triangolare destra è una matrice triangolare sinistra e la trasposta di una sinistra è una matrice destra allora è possibile risolvere i seguenti problemi numerici utilizzando gli algoritmi indicati in precedenza:

$$\mathbf{R}^T\mathbf{x} = \mathbf{b} \quad \rightarrow \quad \mathbf{L}\mathbf{x} = \mathbf{b}$$

$$\mathbf{L}^T\mathbf{x} = \mathbf{b} \quad \rightarrow \quad \mathbf{R}\mathbf{x} = \mathbf{b}$$

Sull'inversa di una matrice

L'**analisi classica** indica che la soluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$ è: $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

Sembrerebbe quindi necessario, ai fini della determinazione del vettore delle incognite $\mathbf{x}$, dover effettuare il prodotto della matrice inversa $\mathbf{A}^{-1}$ per il vettore dei termini noti $\mathbf{b}$.

L'**analisi classica** detta anche le condizioni per l'invertibilità della matrice $\mathbf{A}$: non deve essere singolare ovvero il suo determinante deve essere diverso da zero.

N.B.: vedremo come passando alla risoluzione numerica di problemi reali tramite l'uso del computer (algoritmi numerici) molte delle condizioni dettate dall'analisi classica non trovino più applicazione o perdano di effettivo significato.

Sull'inversa di una matrice

L'**analisi numerica** si discosta **nettamente** da quella classica in quanto è possibile dimostrare che:

- La soluzione di un sistema lineare previa inversione della matrice $\mathbf{A}$ è **più instabile** rispetto a possibili alternative (es.: fattorizzazione di Gauss)
- Per invertire una matrice quadrata di ordine n sono necessari n^3 flop.
- Per risolvere un sistema tramite fattorizzazione di Gauss sono necessari $n^3/3$ flop.

N.B.: una matrice dovrebbe essere invertita solo quando **esplicitamente** necessario. In genere è quasi sempre possibile evitare l'inversione della matrice quando tale operazione compare in una formula.

Sull'inversa di una matrice

Vantaggi numerici

- La risoluzione di un sistema lineare denso tramite fattorizzazione è 3 volte più veloce che tramite inversione della matrice $\mathbf{A}$.
- La precisione numerica della fattorizzazione è maggiore dell'operazione di inversione.
- Se la matrice $\mathbf{A}$ è sparsa si possono avere ulteriori risparmi di tempo di calcolo rispetto alla inversione.
- Se la matrice $\mathbf{A}$ è sparsa si può avere un risparmio di memoria enorme rispetto alla inversione.
- La fattorizzazione di matrici sparse può richiedere anche ordini di grandezza di n, n^2 flop.

Metodo di eliminazione di Gauss

Si riprendono ora e si sistematizzano i concetti introdotti in precedenza.

Dato il sistema lineare $\mathbf{Ax} = \mathbf{b}$, si considera la matrice rettangolare risultante dalla matrice $\mathbf{A}$ e dal vettore $\mathbf{b}$.

Poiché i coefficienti di $\mathbf{A}$ e $\mathbf{b}$ vengono iterativamente modificati, si assegna ad essi un indice contatore che all'inizio viene posto uguale a zero:

$$\left[\mathbf{A}\middle|\mathbf{b}\right] = \left[\begin{array}{ccc|c} a_{1,1}^{(0)} & \cdots & a_{1,n}^{(0)} & b_1^{(0)} \\ \vdots & \ddots & \vdots & \vdots \\ a_{n,1}^{(0)} & \cdots & a_{n,n}^{(0)} & b_n^{(0)} \end{array}\right]$$

La notazione con l'indice come apice dei coefficienti serve soltanto per comprendere l'evoluzione dinamica della procedura numerica. Nella memoria del computer viceversa i coefficienti `a(i,j)` e `b(i)` vengono ricoperti durante l'esecuzione dell'algoritmo.

Metodo di eliminazione di Gauss

N.B.: Il metodo di eliminazione di Gauss si basa sul fatto teorico che la soluzione del sistema non cambia se sui coefficienti della matrice $[\mathbf{A}|\mathbf{b}]$ vengono eseguite le seguenti operazioni:

(1) Moltiplicazione di tutti gli elementi di una riga per una costante non nulla;

(2) Sostituzione di una riga con una combinazione lineare di quella riga e di altre;

(3) Permutazione tra righe.

L'**analisi classica** fornisce un importante risultato teorico:

Se la matrice $\mathbf{A}$ non è singolare **almeno uno** dei coefficienti: $a_{i,1}^{(0)} \quad (i = 1,\ldots n)$ è diverso da zero.

N.B.: se $a_{1,1}^{(0)} = 0$ è sufficiente effettuare uno scambio di righe (la soluzione non risulta modificata).

Metodo di eliminazione di Gauss

Il risultato del sistema **non** cambia se si eseguono le seguenti operazioni:

$$a_{i,j}^{(1)} = a_{i,j}^{(0)} - \ell_{i,1} a_{1,j}^{(0)} \qquad (i = 2,3,...,n) \qquad (j = 1,2,...,n)$$

$$b_i^{(1)} = b_i^{(0)} - \ell_{i,1} b_1^{(0)} \qquad (i = 2,3,...,n)$$

Con i moltiplicatori dati dalla relazione:

$$\ell_{i,1} = \frac{a_{i,1}^{(0)}}{a_{1,1}^{(0)}} \qquad (i = 2,3,...,n)$$

L'obiettivo di queste operazioni è quello di rendere nulli i coefficienti della prima colonna $a_{i,1}^{(1)}$ $(i = 2,n)$ al di sotto del coefficiente $a_{1,1}^{(0)}$ in modo da eliminare la variabile x_1 dalle equazioni $2,3,\ldots,n$.

Metodo di eliminazione di Gauss

I nuovi coefficienti dopo la **prima iterazione** sono:

$$\left[\mathbf{A}\middle|\mathbf{b}\right]=\left[\begin{array}{cccc|c} a_{1,1}^{(0)} & a_{1,2}^{(0)} & \dots & a_{1,n}^{(0)} & b_1^{(0)} \\ 0 & a_{2,2}^{(1)} & \dots & a_{2,n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n,2}^{(1)} & \dots & a_{n,n}^{(1)} & b_n^{(1)} \end{array}\right]$$

Il procedimento viene ripetuto per le colonne 2, 3,…. Dopo la $(k-1)$-esima iterazione la matrice $\mathbf{A}$ e il vettore $\mathbf{b}$ hanno la seguente forma:

$$\left[\mathbf{A}\middle|\mathbf{b}\right]=\left[\begin{array}{ccccccc|c} a_{1,1}^{(0)} & a_{1,2}^{(0)} & \dots & a_{1,k}^{(0)} & \dots & a_{1,n}^{(0)} & b_1^{(0)} \\ 0 & a_{2,2}^{(1)} & \dots & \dots & \dots & a_{2,n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & & & \vdots & \vdots \\ 0 & 0 & & a_{k,k}^{(k-1)} & \dots & a_{k,n}^{(k-1)} & b_k^{(k-1)} \\ \vdots & \vdots & & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{n,k}^{(k-1)} & \dots & a_{n,n}^{(k-1)} & b_n^{(k-1)} \end{array}\right]$$

Metodo di eliminazione di Gauss

L'**analisi classica** fornisce un altro importante risultato teorico: se la matrice $\mathbf{A}$ non è singolare almeno uno dei coefficienti $a_{i,k}^{(k-1)}$ $(i = k, n)$ è diverso da zero.

Nuovamente si può supporre che $a_{k,k}^{(k-1)} \neq 0$. In caso contrario è sufficiente effettuare uno scambio di righe.

Definizione: il coefficiente $a_{k,k}^{(k-1)}$ viene chiamato k-esimo **pivot**.

Nuovamente, il risultato del sistema non cambia se si eseguono le seguenti operazioni:

$$a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - \ell_{i,k} a_{k,j}^{(k-1)} \qquad (i = k+1,\ldots,n) \qquad (j = k,\ldots,n)$$

$$b_i^{(k)} = b_i^{(k-1)} - \ell_{i,k} b_k^{(k-1)} \qquad (i = k+1,\ldots,n)$$

Con i moltiplicatori dati dalla relazione:

$$\ell_{i,k} = \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} \qquad (i = k+1,\ldots,n)$$

Metodo di eliminazione di Gauss

Giunti all'iterazione $k = n - 1$ i coefficienti sono:

$$\left[\mathbf{A}|\mathbf{b}\right] = \left[\begin{array}{cccc|c} a_{1,1}^{(0)} & a_{1,2}^{(0)} & \dots & a_{1,n}^{(0)} & b_1^{(0)} \\ 0 & a_{2,2}^{(1)} & \dots & a_{2,n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & a_{n,n}^{(n-1)} & b_n^{(n-1)} \end{array}\right]$$

N.B.: la matrice $\mathbf{A}$ è stata trasformata in triangolare destra. Il sistema lineare è cioè stato trasformato in $\mathbf{Rx} = \mathbf{c}$. Se la matrice $\mathbf{A}$ è non singolare i coefficienti $r_{i,i}$ sono diversi da zero poiché sono i **pivot** utilizzati durante il procedimento di eliminazione di Gauss.

Il sistema triangolare destro può essere infine risolto con gli algoritmi descritti in precedenza. Si rammenta che tali algoritmi richiedono $n^2/2$ flop. Dato che la triangolarizzazione della matrice $\mathbf{A}$ richiede $n^3/3$ flop, il totale delle operazioni per risolvere il sistema con il metodo di Gauss è pari a $n^3/3 + n^2/2$ flop. Per valori di n sufficientemente grandi $n^3/3 >> n^2/2$. In questi casi si dice che la risoluzione di un sistema lineare denso con il metodo di Gauss richiede $n^3/3$ flop.

Metodo di eliminazione di Gauss

È stato appena detto che i coefficienti $r_{i,i}$ sono diversi da zero. Di conseguenza il sistema $\mathbf{Rx} = \mathbf{c}$ è **teoricamente** risolubile senza problemi.

In realtà il procedimento presentato fino ad ora **NON** deve essere utilizzato per implementare un programma di calcolo perché potrebbe produrre risultati completamente **errati**, fuorvianti.

Metodo di eliminazione di Gauss

Esempio: il sistema lineare $\begin{cases} 10x_1 + 50000x_2 = 50000 \\ 0.4x_1 - 0.3x_2 = 0.1 \end{cases}$

ha **soluzione esatta**: $x_1 = 0.9999 \quad x_2 = 0.9998$

Se però si lavora in *floating point* con una precisione di quattro cifre significative si ha:

il moltiplicatore $l_{2,1}$ è uguale a `fl(0.4/10.)=0.04`. Il coefficiente $a_{2,2}$ diventa:

a_{22} `= fl(-0.3-0.04*50000.) = -2000.`

mentre il termine noto diventa: b_2 `= fl(0.1+0.04*50000.) = -2000.`

Si ottiene come **soluzione numerica**: x_1 `= 0.` x_2 `= 1.`

Domanda: Qual è la fonte di errore? Dove si è sbagliato?

Norme di vettori

Per poter rispondere alle domande poste in precedenza occorre introdurre il concetto di norma di vettore e di matrice.

Definizione: la norma p di un vettore $\mathbf{x} \in \mathbf{R}^n$ è una funzione reale e non negativa degli elementi del vettore. La si indica con: $\|\mathbf{x}\|_p$

La norma di un vettore soddisfa le seguenti proprietà:

1. **Positività**: $\|\mathbf{x}\| \geq 0$ per tutti gli $\mathbf{x} \in \mathbf{R}^n$ e $\|\mathbf{x}\| = 0$ se e solo se $\mathbf{x} = 0$
2. **Omogeneità**: $\|c\mathbf{x}\| = |c| \|\mathbf{x}\|$ per tutti gli $\mathbf{x} \in \mathbf{R}^n$ e $c \in \mathbf{R}$
3. **Disuguaglianza triangolare**: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ per tutti gli $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$

Norme di vettori

Le principali norme per un vettore sono:

- Norma Massima o Infinita: $\|\mathbf{x}\|_\infty = \max|x_k| \qquad (k = 1,\ldots,n)$
- Norma Euclidea o Norma-2: $\|\mathbf{x}\|_2 = \sqrt{\sum_{k=1}^{n} x_k^2}$
- Norma-1: $\|\mathbf{x}\|_1 = \sum_{k=1}^{n} |x_k|$
- Norma di Hölder o p-norma: $\|\mathbf{x}\|_p = \left(\sum_{k=1}^{n} |x_k|^p \right)^{1/p}$

N.B.: La p-norma racchiude le tre norme: Massima, Euclidea e Norma-1 rispettivamente per $p = \infty, 2, 1$

Esempio: $\mathbf{z} = [3, -9, 0, 2]$ ⇒ $\|\mathbf{z}\|_1 = 14 \quad \|\mathbf{z}\|_2 = \sqrt{94} \quad \|\mathbf{z}\|_\infty = 9$

Norme di vettori

Definizione: la distanza tra due vettori $\mathbf{x}$ e $\mathbf{y}$ è data dalla norma: $\|\mathbf{x}-\mathbf{y}\|$

Definizione: angolo α tra due vettori: $\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos(\alpha) \Rightarrow \alpha = \arccos\left(\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}\right)$

Si noti che se $\mathbf{x}^T\mathbf{y} = 0 \Rightarrow \alpha = \frac{\pi}{2}$ cioè i due vettori sono ortogonali.

Definizione: se $\hat{\mathbf{x}}$ è un'approssimazione di $\mathbf{x}$, data una norma qualsiasi, si definisce **errore assoluto** della stima $\varepsilon_a = \|\hat{\mathbf{x}}-\mathbf{x}\|$

Definizione: se $\hat{\mathbf{x}}$ è un'approssimazione di $\mathbf{x}$, data una norma qualsiasi, si definisce **errore relativo** della stima $\varepsilon_r = \frac{\|\hat{\mathbf{x}}-\mathbf{x}\|}{\|\mathbf{x}\|}$

N.B.: se l'errore relativo ε_r viene misurato con la norma massima $\|\bullet\|_\infty$ e risulta uguale a 10^{-p} allora il massimo dei coefficienti di $\mathbf{x}$ ha approssimativamente p cifre significative.

N.B.: dato che la norma condensa in un unico valore le caratteristiche dei coefficienti del vettore, le informazioni che fornisce possono essere completamente fuorvianti per alcuni componenti del vettore stesso.

Norme di matrici

Definizione: la norma p di una matrice $\mathbf{A} \in \mathbf{R}^{mxn}$ è una funzione reale e non negativa degli elementi della matrice. La si indica con: $\|\mathbf{A}\|_p$

La norma di una matrice soddisfa le seguenti proprietà:

1. **Positività**: $\|\mathbf{A}\| \geq 0$ per tutte le $\mathbf{A} \in \mathbf{R}^{mxn}$ e $\|\mathbf{A}\| = 0$ se e solo se $\mathbf{A} = 0$
2. **Omogeneità**: $\|c\mathbf{A}\| = |c|\|\mathbf{A}\|$ per tutte le $\mathbf{A} \in \mathbf{R}^{mxn}$ e $c \in \mathbf{R}$
3. **Disuguaglianza triangolare**: $\|\mathbf{A}+\mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ per tutte le $\mathbf{A},\mathbf{B} \in \mathbf{R}^{mxn}$

Norme di matrici

Le principali norme per una matrice sono:

- Norma Totale: $\|\mathbf{A}\|_T = \max(m,n) \cdot \max_{i=1,m;\, j=1,n} |a_{i,j}|$
- Norma Riga o Massima: $\|\mathbf{A}\|_R = \|\mathbf{A}\|_\infty = \max_{i=1,m} \sum_{j=1}^{n} |a_{i,j}|$
- Norma Colonna o Norma-1: $\|\mathbf{A}\|_C = \|\mathbf{A}\|_1 = \max_{j=1,n} \sum_{i=1}^{m} |a_{i,j}|$
- Norma di Frobenius o di Shur o Euclidea: $\|\mathbf{A}\|_F = \sqrt{\sum a_{i,j}^2}$
- Norma-2 o Spettrale: $\|\mathbf{A}\|_2 = \sqrt{\max\left(\mathrm{eig}\left(\mathbf{A}^T\mathbf{A}\right)\right)}$

N.B.: La norma Spettrale o Norma-2 di una matrice $\mathbf{A}$ richiede il calcolo degli autovalori della matrice $\mathbf{A}^T\mathbf{A}$. Tale calcolo (in genere condotto tramite fattorizzazione SVD (Singular Value Decomposition)) è molto pesante in termini di tempo di CPU (circa 18 volte il tempo necessario per una fattorizzazione con il metodo di Gauss).

Norme di matrici e di vettori

Definizione: la norma a di una matrice è **compatibile** con quella b di un vettore se risulta: $\|\mathbf{Ax}\|_b \leq \|\mathbf{A}\|_a \|\mathbf{x}\|_b$

Le seguenti norme sono compatibili:

$\|\mathbf{A}\|_T$ e $\|\mathbf{A}\|_R$ sono compatibili con $\|\mathbf{x}\|_\infty$

$\|\mathbf{A}\|_T$ e $\|\mathbf{A}\|_C$ sono compatibili con $\|\mathbf{x}\|_1$

$\|\mathbf{A}\|_T$ $\|\mathbf{A}\|_F$ e $\|\mathbf{A}\|_2$ sono compatibili con $\|\mathbf{x}\|_2$

Condizionamento di un sistema lineare

Si desidera rammentare che un **problema** (quindi un sistema) può essere bene o mal **condizionato**.

Viceversa un **algoritmo** può essere stabile o non **stabile**.

Nell'**analisi numerica** il **determinante** di un sistema lineare **NON** permette assolutamente di valutare il buono o cattivo condizionamento del problema.

NON è assolutamente vero che se il determinante di un sistema lineare è grande allora non ci sono problemi nella risoluzione dello stesso, mentre se tende a zero allora il sistema è praticamente singolare.

Condizionamento di un sistema lineare

Esempio: si consideri il sistema con matrice diagonale $m \times m$ con elementi tutti uguali a 0.1. Il determinante della matrice vale 10^{-m} ed è molto piccolo se m è grande. Ciononostante il sistema è molto ben condizionato.

Esempio: la matrice: $\begin{bmatrix} \left(1.+10^{-7}\right)10^{n} & 10^{n} \\ 10^{n} & 10^{n} \end{bmatrix}$ ha determinante pari a 10^{2n-7}.

Tale determinante è molto grande se n è grande. Al contempo il sistema lineare è molto mal condizionato in quanto le due righe sono praticamente linearmente dipendenti (matrice pressoché singolare).

Insegnamento: le grandezze dell'analisi classica, dedotte per via numerica non permettono di affermare che una certa proprietà sia valida quando il valore calcolato è prossimo a quello teorico proposto dall'analisi classica (vedi determinante nullo per una matrice singolare).

Condizionamento di un sistema lineare

Si rammenta che:

Un problema è ben condizionato se piccole perturbazioni sui dati non provocano grandi variazioni della soluzione.

Per poter analizzare il condizionamento di un sistema lineare $\mathbf{Ax} = \mathbf{b}$ occorre quindi perturbare singolarmente i dati del problema cioè $\mathbf{A}$ e $\mathbf{b}$ ed analizzare l'influenza prodotta sul vettore delle incognite $\mathbf{x}$.

Condizionamento di un sistema lineare

Perturbazione del vettore dei termini noti $\mathbf{b}$

Si perturba il vettore $\mathbf{b}$ di una quantità $\delta\mathbf{b}$. Si ottiene: $\mathbf{A}\left(\mathbf{x}+\delta\mathbf{x}_b\right)=\mathbf{b}+\delta\mathbf{b}$

Il pedice b per il vettore $\mathbf{x}$ indica la perturbazione del vettore $\mathbf{b}$.

Dato che $\mathbf{Ax} = \mathbf{b}$ allora si deduce che: $\mathbf{A}\delta\mathbf{x}_b=\delta\mathbf{b}$

e in assenza di errori di arrotondamento l'analisi classica assicura che: $\delta\mathbf{x}_b=\mathbf{A}^{-1}\delta\mathbf{b}$

Utilizzando una qualsiasi norma si ottiene: $\left\|\delta\mathbf{x}_b\right\|=\left\|\mathbf{A}^{-1}\delta\mathbf{b}\right\|\le\left\|\mathbf{A}^{-1}\right\|\left\|\delta\mathbf{b}\right\|$

Osservazione: l'**errore assoluto** della soluzione, $\delta\mathbf{x}_b$, viene quindi amplificato rispetto ad un errore assoluto sui termini noti, $\delta\mathbf{b}$, per un fattore pari a: $\left\|\mathbf{A}^{-1}\right\|$

Condizionamento di un sistema lineare

Inoltre dato che vale: $\|\mathbf{b}\| = \|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$ è possibile dedurre:

$$\frac{\|\delta \mathbf{x}_b\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\|\|\mathbf{A}\| \frac{\|\delta \mathbf{b}\|}{\|\mathbf{b}\|}$$

Osservazione: l'**errore relativo** della soluzione, $\delta \mathbf{x}_b$, viene amplificato, rispetto ad un errore relativo sui termini noti, per un fattore pari a: $\kappa(\mathbf{A}) = \|\mathbf{A}^{-1}\|\|\mathbf{A}\|$

Definizione: il prodotto: $\|\mathbf{A}^{-1}\|\|\mathbf{A}\|$ viene definito **numero di condizionamento** della matrice $\mathbf{A}$ (talvolta è anche indicato con $\mathrm{cond}(\mathbf{A})$).

Condizionamento di un sistema lineare

Perturbazione della matrice dei coefficienti $\mathbf{A}$

Si perturba la matrice $\mathbf{A}$ di una quantità $\delta\mathbf{A}$. Si ottiene: $\left(\mathbf{A}+\delta\mathbf{A}\right)\left(\mathbf{x}+\delta\mathbf{x}_a\right)=\mathbf{b}$

Il pedice a per il vettore $\mathbf{x}$ indica la perturbazione della matrice $\mathbf{A}$.

Per analogia con le considerazioni precedenti si ottiene:

$$\frac{\left\|\delta\mathbf{x}_a\right\|}{\left\|\mathbf{x}+\delta\mathbf{x}_a\right\|}\leq\left\|\mathbf{A}^{-1}\right\|\left\|\mathbf{A}\right\|\frac{\left\|\delta\mathbf{A}\right\|}{\left\|\mathbf{A}\right\|}=\kappa\left(\mathbf{A}\right)\frac{\left\|\delta\mathbf{A}\right\|}{\left\|\mathbf{A}\right\|}$$

Osservazione: ancora una volta il **numero di condizionamento** ha un ruolo fondamentale per la determinazione del peso esercitato sulla soluzione da parte delle perturbazioni dei dati $\mathbf{A}$ e $\mathbf{b}$.

Il **reciproco** del numero di condizionamento misura la distanza tra la matrice $\mathbf{A}$ e l'insieme delle matrici singolari. Il numero di condizionamento dipende dalla norma adottata ma è sempre maggiore di 1.

Condizionamento di un sistema lineare

Definizione: un sistema lineare è ben condizionato se il **numero di condizionamento** della matrice $\mathbf{A}$ è prossimo ad 1.

Esempio: la matrice: $\begin{bmatrix} \left(1.+10^{-7}\right)10^{n} & 10^{n} \\ 10^{n} & 10^{n} \end{bmatrix}$ ha $\kappa\left(\mathbf{A}\right)=10^{7}$ utilizzando la Norma Totale. È quindi possibile affermare che il sistema ad essa connesso è decisamente mal condizionato.

N.B.: il logaritmo decimale del numero di condizionamento approssima il numero **massimo** di cifre significative che verranno perse durante la soluzione del sistema. Se il calcolo viene eseguito con d cifre significative ed il numero di condizionamento vale 10^{a}, la soluzione avrà in genere $d-a-1$ cifre significative.

Condizionamento di un sistema lineare

Esempio: se il numero di condizionamento di un sistema è 1.e4 e la precisione dei calcoli (macheps) è 1.e-7 allora $d-a-1 = 7-4-1 = 2$. La soluzione avrà quindi in genere una accuratezza di 2 cifre. Se x_1=1.234567 e x_2=0.0012345 ci si deve aspettare che la terza cifra di x_1 e già la prima cifra significativa di x_2 siano errate.

Esempio: il sistema $\begin{cases} x_1 = 1 \\ 10^{-8} x_2 = 10^{-8} \end{cases}$ è mal condizionato, infatti $\kappa(\mathbf{A}) = 10^8$

Se il vettore dei termini noti varia soltanto di 1.e-8 e se il macheps = 1.e-7, allora il sistema risultante diviene:

$$\begin{cases} x_1 = 1 + 1.\mathrm{E}-8 = 1. \\ 10^{-8} x_2 = 2.10^{-8} \end{cases}$$

è possibile notare come in questo caso la soluzione $x_1 = 1$ e $x_2 = 2$ veda per la seconda incognita un errore relativo del 100%.

Trasformazione di Gauss

Si riprende ora il metodo di Gauss e si passa ad una sua interpretazione più moderna volta ad incrementare l'efficacia dell'analisi numerica e della formulazione del problema.

Nuovo approccio

1. Il metodo di Gauss non è più visto come strumento per **eliminare** le variabili bensì come strumento per **trasformare** il sistema lineare in uno equivalente.

2. Il sistema trasformato ha la **matrice** dei coefficienti $\mathbf{R}$ **triangolare destra** quindi risulta di facile risoluzione.

3. La matrice originale $\mathbf{A}$ deve essere pensata come fattorizzata in due matrici $\mathbf{L}$, $\mathbf{R}$ il cui prodotto fornisce appunto la matrice $\mathbf{A}$ di partenza: $\mathbf{LR} = \mathbf{A}$

Trasformazione di Gauss

Ipotesi semplificativa: per il momento si ipotizza che, con riferimento al metodo di Gauss illustrato in precedenza, i coefficienti di pivot: $a_{kk}^{(k-1)} \neq 0$

Dopo la $(k\text{-}1)$-esima iterazione del metodo di Gauss si è giunti alla struttura:

$$\left[\mathbf{A}|\mathbf{b}\right] = \left[\begin{array}{ccccccc|c} a_{1,1}^{(0)} & a_{1,2}^{(0)} & \dots & a_{1,k}^{(0)} & \dots & a_{1,n}^{(0)} & b_1^{(0)} \\ 0 & a_{2,2}^{(1)} & \dots & \dots & \dots & a_{2,n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & & & \vdots & \vdots \\ 0 & 0 & & a_{k,k}^{(k-1)} & \dots & a_{k,n}^{(k-1)} & b_k^{(k-1)} \\ \vdots & \vdots & & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{n,k}^{(k-1)} & \dots & a_{n,n}^{(k-1)} & b_n^{(k-1)} \end{array}\right]$$

Con i moltiplicatori: $\ell_{i,k} = \dfrac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} \qquad (i = k+1, \dots, n)$

Trasformazione di Gauss

A questo punto è utile considerare una matrice $\mathbf{M}_k$ ottenuta aggiungendo alla matrice identità $\mathbf{I}$ i moltiplicatori $\ell_{i,k}$ cambiati di segno in corrispondenza delle righe $i=k+1,\ldots n$ e della colonna k.

$$\mathbf{M}_k = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & & & & \vdots \\ \vdots & 0 & 1 & \ddots & & & & \vdots \\ \vdots & \vdots & -\ell_{k+1,k} & 1 & \ddots & & & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & -\ell_{j,k} & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & -\ell_{n,k} & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

N.B.: l'attenzione nel prosieguo viene focalizzata soltanto sulla matrice $\mathbf{A}$.

Trasformazione di Gauss

Definizione: la matrice $\mathbf{M}_k$ viene chiamata Matrice di trasformazione di Gauss.

Si noti che si ottiene la matrice $\mathbf{A}^{(k)}$ dalla relazione: $\mathbf{A}^{(k)} = \mathbf{M}_k \mathbf{A}^{(k-1)}$

Esempio: si consideri una matrice $\mathbf{A}$ di ordine 4: $\mathbf{A}^{(0)} = \begin{bmatrix} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & a_{1,4}^{(0)} \\ a_{2,1}^{(0)} & a_{2,2}^{(0)} & a_{2,3}^{(0)} & a_{2,4}^{(0)} \\ a_{3,1}^{(0)} & a_{3,2}^{(0)} & a_{3,3}^{(0)} & a_{3,4}^{(0)} \\ a_{4,1}^{(0)} & a_{4,2}^{(0)} & a_{4,3}^{(0)} & a_{4,4}^{(0)} \end{bmatrix}$

La prima matrice di trasformazione è: $\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\ell_{2,1} & 1 & 0 & 0 \\ -\ell_{3,1} & 0 & 1 & 0 \\ -\ell_{4,1} & 0 & 0 & 1 \end{bmatrix}$

Si ha che: $\mathbf{A}^{(1)} = \mathbf{M}_1 \mathbf{A}^{(0)}$

Trasformazione di Gauss

Proseguendo si ha: $\mathbf{M}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\ell_{3,2} & 1 & 0 \\ 0 & -\ell_{4,2} & 0 & 1 \end{bmatrix}$ quindi: $\mathbf{A}^{(2)} = \mathbf{M}_2\mathbf{A}^{(1)} = \mathbf{M}_2\mathbf{M}_1\mathbf{A}^{(0)}$

Infine: $\mathbf{M}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\ell_{4,3} & 1 \end{bmatrix}$ e: $\mathbf{A}^{(3)} = \mathbf{M}_3\mathbf{A}^{(2)} = \mathbf{M}_3\mathbf{M}_2\mathbf{A}^{(1)} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1\mathbf{A}^{(0)}$

Alla fine la matrice originaria $\mathbf{A}^{(0)}$ si è trasformata in: $\mathbf{A}^{(3)} = \begin{bmatrix} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & a_{1,4}^{(0)} \\ 0 & a_{2,2}^{(1)} & a_{2,3}^{(1)} & a_{2,4}^{(1)} \\ 0 & 0 & a_{3,3}^{(2)} & a_{3,4}^{(2)} \\ 0 & 0 & 0 & a_{4,4}^{(3)} \end{bmatrix}$

Trasformazione di Gauss

N.B.: la matrice $\mathbf{A}^{(n-1)}$ ottenuta tramite le $n-1$ matrici di trasformazione è **triangolare destra**.

Le matrici $\mathbf{M}_i$ $(i = 1,\ldots,n-1)$ sono **triangolari sinistre**. Anche il loro prodotto $\mathbf{M}$ è una matrice triangolare sinistra: $\mathbf{M} = \mathbf{M}_{n-1}\mathbf{M}_{n-2}\ldots\mathbf{M}_2\mathbf{M}_1$

La matrice inversa di $\mathbf{M}$ ovvero $\mathbf{M}^{-1}$ è ancora una matrice **triangolare sinistra**. In genere la si indica con $\mathbf{L}$:

$$\mathbf{L} = \mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \ell_{2,1} & 1 & 0 & 0 \\ \ell_{3,1} & \ell_{3,2} & 1 & 0 \\ \ell_{4,1} & \ell_{4,2} & \ell_{4,3} & 1 \end{bmatrix}$$

Trasformazione di Gauss

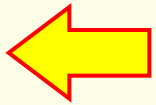
Operativamente dato che ad ogni trasformazione della matrice $\mathbf{A}^{(i)}$ si libera la sua corrispondente colonna i-esima, è possibile ivi memorizzare gli elementi delle matrici $\mathbf{M}_i$ con risparmio di occupazione di memoria. Alla fine si ottiene:

$$\mathbf{A}_{work}^{(n-1)} = \mathbf{A}_{work}^{(3)} = \begin{bmatrix} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & a_{1,4}^{(0)} \\ \ell_{2,1} & a_{2,2}^{(1)} & a_{2,3}^{(1)} & a_{2,4}^{(1)} \\ \ell_{3,1} & \ell_{2,3} & a_{3,3}^{(2)} & a_{3,4}^{(2)} \\ \ell_{4,1} & \ell_{4,2} & \ell_{4,3} & a_{4,4}^{(3)} \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} \\ \ell_{2,1} & r_{2,2} & r_{2,3} & r_{2,4} \\ \ell_{3,1} & \ell_{3,2} & r_{3,3} & r_{3,4} \\ \ell_{4,1} & \ell_{4,2} & \ell_{4,3} & r_{4,4} \end{bmatrix}$$

N.B.: si è indicato con il pedice *work* le matrici di lavoro a livello di algoritmo programmativo volte al summenzionato risparmio di occupazione di memoria.

N.B.: dato che la trasformazione della matrice $\mathbf{A}$ ha portato alla $\mathbf{R}$: $\mathbf{MA} = \mathbf{R}$

Allora: $\mathbf{A} = \mathbf{M}^{-1}\mathbf{R} = \mathbf{LR}$



È possibile affermare che la matrice $\mathbf{A}$ **è stata fattorizzata** nelle matrici $\mathbf{L}$ e $\mathbf{R}$.

Trasformazione di Gauss

Il risultato appena indicato è di **fondamentale** importanza. È possibile affermare che la matrice $\mathbf{M}$ trasforma la $\mathbf{A}$ rendendola triangolare. Il prodotto delle matrici $\mathbf{L}$ e $\mathbf{R}$ fornisce la matrice $\mathbf{A}$ di partenza.

Conseguenze e vantaggi

1. Una volta fattorizzata la matrice $\mathbf{A}$ è possibile risolvere più problemi aventi termini noti $\mathbf{b}$ distinti.

2. Una volta fattorizzata la matrice $\mathbf{A}$ è possibile risolvere il problema: $\mathbf{A}^T\mathbf{x} = \mathbf{c}$ senza dover nuovamente fattorizzare la matrice trasposta: $\mathbf{A}^T$.

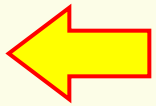
3. Le matrici $\mathbf{L}$ e $\mathbf{R}$ possono essere ricavate anche con tecniche diverse dal processo di eliminazione di Gauss.

4. Se la matrice $\mathbf{A}$ viene modificata è possibile con opportune tecniche ottenere le corrispondenti matrici $\mathbf{L}$ e $\mathbf{R}$ che fattorizzano la nuova matrice $\mathbf{A}^*$.

Risoluzione del sistema

Una volta fattorizzata la matrice $\mathbf{A}$ nelle matrici $\mathbf{L}$ e $\mathbf{R}$ si risolve il sistema originale $\mathbf{Ax} = \mathbf{b}$ nel seguente modo:

$$\mathbf{Ax} = \mathbf{LRx} = \mathbf{L}\left(\mathbf{Rx}\right) = \mathbf{Ly} = \mathbf{b}$$



1. Si risolve dapprima il sistema triangolare sinistro: $\mathbf{Ly} = \mathbf{b}$ che richiede soltanto $n^2/2$ flop.

2. Noto $\mathbf{y}$ si risolve il sistema triangolare destro: $\mathbf{Rx} = \mathbf{y}$ che richiede anch'esso soltanto $n^2/2$ flop.

3. La soluzione è il vettore originale delle incognite: $\mathbf{x}$.

N.B.: la fattorizzazione $\mathbf{LR}$ della matrice $\mathbf{A}$ richiede $n^3/3$ flop. La soluzione del sistema richiede n^2 flop. In genere per valori sufficientemente grandi di n si dice che la risoluzione complessiva del sistema $\mathbf{Ax=b}$ tramite metodo di Gauss richiede: $n^3/3$ flop.

Fattorizzazione LR

Algoritmo

Fattorizzazione **LR** di una matrice quadrata **A**.

Input: dimensione n della matrice, coefficienti $a_{i,j}$

Output: coefficienti $l_{i,j}$ e $r_{i,j}$ nella matrice trasformata $a_{i,j}$.

Flop: l'algoritmo richiede $n^3/3$ flop.

```
for k = 1 to n-1
   for i = k+1 to n
      if a(k,k) = 0 then STOP 'ERRORE'
      a(i,k) = a(i,k) / a(k,k)
      for j = k+1 to n
         a(i,j) = a(i,j) - a(i,k)*a(k,j)
      next j
   next i
next k
```

Ricerca del pivot

Finora sono stati illustrati il metodo di Gauss e quello di trasformazione di Gauss sotto l'ipotesi di pivot non nulli: $a_{kk}^{(k-1)} \neq 0$.

Si desidera ora discutere e analizzare il significato di pivot non nullo e gli effetti della scelta di un pivot sulla risoluzione del sistema non lineare.

Esempio: è dato il sistema $\begin{cases} 0.0002x_1 + x_2 = 1.0 \\ 0.4x_1 - 0.3x_2 = 0.1 \end{cases}$

Se si adotta come primo pivot il coefficiente $a_{1,1}$ e si lavora con una precisione di 4 cifre significative si ottiene x_1=0. e x_2=1. mentre la soluzione esatta è x_1=0.9999 e x_2=0.9998.

- Cosa è successo ???
- Come mai si ottiene un risultato così errato anche se $a_{1,1} \neq 0$???

Ricerca del pivot

Ancora una volta è bene rammentare che esiste una significativa differenza tra analisi classica ed analisi numerica. L'analisi classica lavora in modo esatto (precisione infinita) l'analisi numerica viceversa lavora con precisione finita (*floating point*).

Ciò vuol dire che se per l'analisi classica il pivot **deve** essere **non nullo** per la risolubilità del sistema, viceversa questa condizione non è sufficiente né accettabile per l'analisi numerica.

Dobbiamo quindi tutelarci scegliendo un pivot significativamente diverso da zero, effettuando a tal fine uno **scambio** di **righe** o eventualmente anche di **colonne**.

Esistono due eccezioni per le quali lo scambio di righe/colonne non è necessario:

1. Per matrici simmetriche definite positive
2. Per matrici diagonalmente dominanti ovvero quando: $\left|a_{i,i}\right| > \sum_{j=1;\, j\neq i}^{n} \left|a_{i,j}\right|$

Ricerca del pivot

Nella pratica esistono due strategie per la scelta del pivot. In entrambi i casi la scelta del pivot è limitata alle righe ed alle colonne della matrice $\mathbf{A}$ che non sono state ancora utilizzate in precedenza.

Pivot globale: alla iterazione k-esima viene scelto come pivot il coefficiente che risulta massimo fra tutti i coefficienti rimasti a disposizione. Si sceglie cioè:

$$\left|a_{p,q}^{(k-1)}\right| = \max_{k \le i \le n; k \le j \le n} \left|a_{i,j}^{(k-1)}\right|$$

- Se risulta $p \neq k$ si scambia la riga p con la riga k
- Se risulta $q \neq k$ si scambia la colonna q con la colonna k
- Se si scambiano le righe occorre scambiare anche i corrispondenti termini noti (vettore $\mathbf{b}$)
- Se si scambiano le colonne occorre scambiare anche le corrispondenti variabili (vettore $\mathbf{x}$)

Ricerca del pivot

Pivot parziale: alla iterazione k-esima viene scelto come pivot il massimo fra tutti i coefficienti rimasti a disposizione della sola colonna k-esima:

$$\left|a_{p,k}^{(k-1)}\right| = \max_{k \le i \le n} \left|a_{i,k}^{(k-1)}\right|$$

- Se risulta $p \neq k$ si scambia la riga p con la riga k
- Se si scambiano le righe occorre scambiare anche i corrispondenti termini noti

N.B.: utilizzando la strategia del **pivot globale** gli errori di calcolo rimangono ragionevolmente contenuti. Al contrario utilizzando il **pivot parziale** non è possibile garantire che gli errori di calcolo non vengano amplificati. Nella pratica la differenza tra le due strategie di pivot è spesso nulla e quindi si preferisce il **pivot parziale** in quanto richiede un minor tempo di elaborazione.

Ricerca del pivot

Per rappresentare uno **scambio di righe** della matrice $\mathbf{A}$ è sufficiente mantenere $\mathbf{A}$ inalterata ed introdurre la matrice identità $\mathbf{I}$. Risulta $\mathbf{IA} = \mathbf{A}$.

Si effettua lo scambio di righe sulla matrice identità, che si trasforma nella matrice $\mathbf{P}$. Il prodotto $\mathbf{PA}$ è equivalente allo scambio delle corrispondenti righe della matrice $\mathbf{A}$ di partenza.

Esempio: si desidera scambiare le righe 1 e 2 della matrice $\mathbf{A}$ di ordine 3.

Si ottiene $\mathbf{P}$ scambiando le righe 1 e 2 della matrice identità di ordine 3.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{PA} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} a_{2,1} & a_{2,2} & a_{2,3} \\ a_{1,1} & a_{1,2} & a_{1,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

Ricerca del pivot

Per rappresentare uno **scambio di colonne** della matrice $\mathbf{A}$ è sufficiente mantenere $\mathbf{A}$ inalterata ed introdurre la matrice identità $\mathbf{I}$. Risulta $\mathbf{AI} = \mathbf{A}$.

Si effettua lo scambio di colonne sulla matrice identità, che si trasforma nella matrice $\mathbf{P}$. Il prodotto $\mathbf{AP}$ è equivalente allo scambio delle corrispondenti colonne della matrice $\mathbf{A}$ di partenza.

Esempio: si desidera scambiare le colonne 2 e 3 della matrice $\mathbf{A}$ di ordine 3.

Si ottiene $\mathbf{P}$ scambiando le colonne 2 e 3 della matrice identità di ordine 3.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{AP} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,3} & a_{1,2} \\ a_{2,1} & a_{2,3} & a_{2,2} \\ a_{3,1} & a_{3,3} & a_{3,2} \end{bmatrix}$$

Bilanciamento

Se si utilizza nella risoluzione con il metodo di Gauss la strategia del **pivot parziale** **non è sufficiente** scegliere come pivot il massimo valore sulla colonna di ricerca come indicato in precedenza.

Esempio: si consideri il sistema $\begin{cases} 0.0002x_1 + x_2 = 1.0 \\ 0.4x_1 - 0.3x_2 = 0.1 \end{cases}$

Con il pivot parziale le due righe vengono scambiate dato che risulta $a_{1,1}<a_{2,1}$.

Se si calcola la soluzione con una precisione di quattro cifre significative si ottiene $x_1=1.$, $x_2=1.$ che è un risultato molto prossimo alla soluzione esatta $x_1=0.9999$, $x_2=0.9998$. In questo caso la strategia del pivot parziale ha funzionato.

Bilanciamento

Si moltiplichi ora la prima riga per 50,000. L'analisi classica ci assicura che il sistema è equivalente a quello di partenza e perciò anche la soluzione dovrebbe essere la stessa. Dopo la moltiplicazione si ottiene:

$$\begin{cases} 10x_1 & +50,000x_2 & = & 50,000 \\ 0.4x_1 & -0.3x_2 & = & 0.1 \end{cases}$$

in questo caso la strategia del **pivot parziale** non prescrive lo scambio delle righe in quanto $a_{1,1}>a_{2,1}$.

La soluzione del sistema però, utilizzando una precisione di 4 cifre significative, vale: $x_1=0.$, $x_2=1.$ che è un risultato completamente errato rispetto alla soluzione esatta $x_1=0.9999$, $x_2=0.9998$.

Bilanciamento

Insegnamento: non è vero che un pivot pari a 0.0000001 sia piccolo mentre un pivot pari a 1,000,000,000 sia grande. Ciò che conta è il valore del pivot rispetto agli altri coefficienti della matrice.

Per ovviare al problema presentato in precedenza occorre quindi **bilanciare** i coefficienti della matrice $\mathbf{A}$ così che abbiano tutti lo stesso ordine di grandezza.

Esempio: sbilanciamento dei coefficienti delle righe della matrice $\mathbf{A}$

$$\begin{cases} 10,000x_1 & +15,000x_2 & = & 50,000 \\ 0.2x_1 & -0.003x_2 & = & 0.08 \end{cases}$$

Esempio: sbilanciamento dei coefficienti delle colonne della matrice $\mathbf{A}$

$$\begin{cases} 10,000x_1 & +0.001x_2 & = & 5. \\ 20,000x_1 & -0.003x_2 & = & 7. \end{cases}$$

Bilanciamento

Esempio: sbilanciamento dei coefficienti delle righe e delle colonne della $\mathbf{A}$

$$\begin{cases} 3x_1 & +10,000x_2 & = & 200 \\ 20,000x_1 & -10^{10}x_2 & = & 1,000,000 \end{cases}$$

N.B.: uno **sbilanciamento delle colonne** si ha quando vengono utilizzate unità di misura consistenti ma differenti tra le diverse incognite (ad esempio tonnellate e milligrammi, oppure chilometri e micron).

Lo **sbilanciamento delle righe** è invece legato spesso, in ambito chimico, alle equazioni stechiometriche e a quelle di bilancio energetico (entalpico). Ad esempio:

$$\sum x_i - \sum y_i = 0$$
$$\sum H_{IN} - \sum H_{OUT} - Q_{DISP} = 0$$

Bilanciamento

In genere si lascia all'utente bilanciare le colonne (unità di misura uguali per le incognite della stessa specie). Al contrario è l'algoritmo risolutivo che si incarica del bilanciamento delle righe.

Il **bilanciamento** può essere condotto in **due modi** differenti:

1. Dividendo ogni riga per la somma dei valori assoluti dei coefficienti di quella riga.

2. Dividendo ogni riga per il massimo dei valori assoluti dei coefficienti di quella riga.

Occorre poi scegliere se applicare il bilanciamento solo alla matrice originale o dopo ogni trasformazione k-esima in quanto i coefficienti della matrice $\mathbf{A}^{(k)}$ continuano a cambiare nel corso delle iterazioni.

Bilanciamento

In realtà è possibile scegliere il pivot ottimale operando un **bilanciamento implicito**. I coefficienti della riga stessa **non** vengono cioè modificati.

Alla k-esima iterazione si determina l'indice p per cui si ha:

$$\left|a_{p,q}^{(k-1)}\right| = \max_{k\le i\le n} \frac{\left|a_{i,k}^{(k-1)}\right|}{\max\limits_{k\le j\le n}\left|a_{i,j}^{(k-1)}\right|}$$

Se risulta $p \neq k$ si scambia la riga p con la riga k e ci si ferma lì, senza effettuare il vero e proprio bilanciamento. Quello che importa è l'individuazione dell'indice p che permette a sua volta l'individuazione ed utilizzo del pivot ottimale.

Se infatti si operasse il bilanciamento si introdurrebbero ulteriori inutili errori di arrotondamento che potrebbero addirittura risultare più nocivi del beneficio apportato dal bilanciamento stesso.

Bilanciamento

Esempio: si torni a considerare il sistema:

$$\begin{cases} 10x_1 & +50,000x_2 & = & 50,000 \\ 0.4x_1 & -0.3x_2 & = & 0.1 \end{cases}$$

senza bilanciamento le righe non vengono scambiate perché $a_{1,1}>a_{2,1}$. Così facendo però il risultato finale è completamente errato.

Il bilanciamento implicito richiede invece lo scambio delle righe in quanto:

$$10./50,000. < 0.4/0.4$$

N.B.: nella realtà è stato sufficiente sapere che le due righe debbono essere scambiate. Come sottolineato in precedenza, le due righe **NON** vengono fisicamente divise per 50,000 la prima e per 0.4 la seconda.

Calcolo del determinante

Ancora una volta è possibile misurare la distanza tra analisi classica ed analisi numerica.

Se il calcolo del determinante di una matrice dovesse essere eseguito utilizzando la sua definizione sarebbero necessari: $(n-1)\cdot n!$ flop. Operazione assolutamente impraticabile già per modeste dimensioni della matrice $\mathbf{A}$.

In realtà se $\mathbf{A}$ è stata fattorizzata nelle matrici $\mathbf{LR}$ allora è sufficiente rammentare che il determinante di una matrice triangolare è dato dal prodotto dei coefficienti sulla diagonale e che in generale:

se $\mathbf{A} = \mathbf{B}\,\mathbf{C}$ → $\det(\mathbf{A}) = \det(\mathbf{B})\,\det(\mathbf{C})$

Quindi il determinante della matrice $\mathbf{A}$ coincide con il prodotto dei coefficienti della diagonale della matrice $\mathbf{R}$ dato che $\det(\mathbf{L}) = 1$.

N.B.: ciò è valido soltanto se non è stato operato alcun pivoting sulla matrice $\mathbf{A}$ di partenza. In caso contrario occorre considerare anche la matrice di permutazione $\mathbf{P}$ e calcolare il determinante di $\mathbf{P}^{-1}\mathbf{L}$ che anziché essere necessariamente uguale a 1 può essere anche uguale a -1 (in base al numero pari o dispari di permutazioni effettuate).

Calcolo della matrice inversa

Come detto in precedenza, per risolvere il sistema $\mathbf{Ax} = \mathbf{b}$ non occorre invertire la matrice $\mathbf{A}$. Ciò oltre a richiedere più tempo comporta anche una precisione minore. È un errore affermare che la soluzione del sistema passa per l'inversione della matrice $\mathbf{A}$.

Nel caso particolare in cui si desideri comunque conoscere la matrice inversa $\mathbf{A}^{-1}$ è sufficiente considerare che si sta cercando una matrice $\mathbf{X} = \mathbf{A}^{-1}$ tale che:

$$\mathbf{AX} = \mathbf{I}$$

Dopo aver fattorizzato la matrice $\mathbf{A}$ occorre risolvere gli n sistemi che hanno come matrice complessiva (*augmented matrix*):

$$\left[\mathbf{A}\middle|\mathbf{I}\right] = \left[\begin{array}{cccc|cccc} a_{1,1} & a_{1,2} & \dots & a_{1,n}^{(0)} & 1 & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & & a_{2,n}^{(1)} & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\ a_{n,1} & \dots & \dots & a_{n,n}^{(n-1)} & 0 & \dots & 0 & 1 \end{array}\right]$$

Metodo di Cramer

Uno dei metodi numerici per la risoluzione di un sistema lineare, di cui si è sentito probabilmente parlare già ai tempi del liceo, è il metodo di **Cramer**.

Le singole soluzioni x_i del sistema $\mathbf{Ax} = \mathbf{b}$ vengono espresse come quozienti di **determinanti** di ordine n.

N.B.: il costo per il calcolo del determinante a partire dalla sua definizione è pari a: $(n-1)\cdot n!$ flop. Dato che per risolvere il sistema con il metodo di Cramer occorre calcolare $(n+1)$ determinanti, il numero complessivo di flop risulta essere $(n+1)\cdot(n-1)\cdot n!$. Da ciò si deduce che il metodo di Cramer è assolutamente improponibile. Non è assolutamente fattibile.

Esempio: per risolvere un sistema lineare 20×20 con un personal computer convenzionale sono necessari circa:

- **Metodo di Cramer**: 2.15 milioni di anni
- **Metodo di Gauss**: 0.0003 secondi

Sistemi lineari strutturati

Seguono alcuni **cenni** ai sistemi lineari strutturati.

Si desidera innanzitutto rammentare che non è richiesta fattorizzazione per:

- Sistemi diagonali (la soluzione è diretta dato che le singole equazioni sono disgiunte cioè in ogni equazione compare una sola variabile)
- Sistemi triangolari destri
- Sistemi triangolari sinistri

Matrici simmetriche

Una matrice $\mathbf{S}$ è **simmetrica** se $\mathbf{S}^T = \mathbf{S}$.

Una matrice $\mathbf{S}$ è **simmetrica definita positiva** se $\mathbf{x}^T\mathbf{S}\mathbf{x} > 0$ per qualsiasi vettore $\mathbf{x} \neq 0$.

N.B.: grazie alla simmetria della matrice è possibile memorizzare soltanto la metà dei coefficienti della matrice $\mathbf{S}$.

Se la matrice $\mathbf{S}$ è simmetrica definita positiva allora è possibile applicare la **fattorizzazione di Cholesky** che non richiede la ricerca del pivot massimo e lo scambio delle righe. Il numero di flop richiesto è: $n^3/6$ ossia la metà del metodo di Gauss.

Se la matrice è simmetrica ma non definita positiva non è possibile utilizzare il metodo di Cholesky in quanto si dovrebbe calcolare la radice quadrata di un numero negativo.

A. L. Cholesky (1875-1918) ufficiale francese, sviluppò il metodo di fattorizzazione omonimo allo scopo di risolvere le equazioni normali ottenute applicando il metodo dei minimi quadrati al *fitting* di dati geodetici.

Matrici a banda

Una **matrice a banda** è una matrice che oltre alla diagonale principale ha gli altri coefficienti raccolti sulle diagonali prossime a quella principale. Il resto dei coefficienti è invece nullo.

Ad esempio si parla di sistema tridiagonale se:

$$\begin{array}{ccccccc} d_1 & u_1 & 0 & \dots & \dots & 0 & = & b_1 \\ \ell_1 & d_2 & u_2 & \ddots & & \vdots & = & b_2 \\ 0 & \ell_2 & d_3 & u_3 & \ddots & \vdots & = & b_3 \\ \vdots & \ddots & \ell_3 & \ddots & \ddots & 0 & = & \vdots \\ \vdots & & \ddots & \ddots & d_{n-1} & u_{n-1} & = & b_{n-1} \\ 0 & \dots & \dots & 0 & \ell_{n-1} & d_n & = & b_n \end{array}$$

Matrici a banda

N.B.: nei problemi di ingegneria chimica la presenza di sistemi a banda (non solo tridiagonali ma anche con ampiezze di banda maggiori e non necessariamente equivalenti) è molto frequente.

Esempi tipici sono le unità di strippaggio, assorbimento o distillazione, quelle unità cioè dove si ha un collegamento tra lo stadio i-esimo quello $(i\text{-}1)$-esimo e quello $(i+1)$-esimo.

Altro esempio nasce dalla discretizzazione spaziale o spazio-temporale di un sistema di equazioni differenziali alle derivate parziali. Ancora una volta a livello di equazioni c'è uno scambio di informazioni (legame tra variabili) tra nodi: centrale, precedente e seguente.

L'ampiezza della banda può crescere quando per i singoli nodi si hanno più variabili locali. Si pensi ad esempio ad una colonna di distillazione: per ogni stadio si hanno: T, P, x_i, y_i, $H^v{}_i$, $H^l{}_i$, …. In questo caso si parla di matrici tridiagonali a blocchi. Ogni elemento della banda non è un semplice coefficiente bensì un blocco ovvero una matrice quadrata contenente le singole variabili locali.

Matrici a banda

Il numero di operazioni necessarie per la risoluzione di un **sistema tridiagonale** a blocchi è dell'ordine di $6n$ flop.

Più in generale se si considera una **matrice a banda** avente u diagonali sopra quella principale e w diagonali sotto quella principale, il numero totale di diagonali non nulle è: $u + w + 1$. Per tale matrice il numero di flop necessari per la fattorizzazione è pari a $nw(w+u)$.

Metodi iterativi

Come indicato in precedenza esistono metodi diretti e metodi iterativi per la risoluzione di un sistema lineare.

Quando le dimensioni del sistema divengono significative (alcune migliaia di equazioni) i metodi iterativi prendono il sopravvento su quelli diretti. Occorre anche evidenziare che in genere, quando il numero di equazioni è così elevato, la matrice dei coefficienti non è densa o piena. Al contrario, la matrice dei coefficienti è sparsa e spesso strutturata. I metodi iterativi sono principalmente utilizzati nella soluzione di problemi CFD (Computational Fluid Dynamics) o comunque qualora sia coinvolta la soluzione di equazioni differenziali alle derivate parziali tramite metodi "a elementi finiti". Il dominio spaziale di integrazione di tali equazioni viene suddiviso in una griglia (mesh) opportuna. Alla base dell'algoritmo risolutivo sta la soluzione di un sistema di equazioni lineari il cui numero è legato al numero di elementi della griglia di discretizzazione spaziale.

Metodi iterativi

La procedura iterativa di soluzione trova la seguente formulazione.

Il sistema $\mathbf{Ax} = \mathbf{b}$ può essere riscritto come:

$$\mathbf{Sx} = \left(\mathbf{S} - \mathbf{A}\right)\mathbf{x} + \mathbf{b}$$

dove $\mathbf{S}$ è un'opportuna matrice ausiliaria.

Se le matrici $\mathbf{A}$ e $\mathbf{S}$ godono di certe proprietà è possibile risolvere iterativamente il sistema tramite la formula:

$$\mathbf{Sx}^{(i+1)} = \left(\mathbf{S} - \mathbf{A}\right)\mathbf{x}^{(i)} + \mathbf{b} = \mathbf{c}$$

Partendo da un punto di primo tentativo $\mathbf{x}^{(0)}$, più o meno prossimo alla soluzione, si itera il procedimento. Se tali proprietà sono rispettate, allora la procedura converge alla soluzione.

La procedura iterativa viene bloccata quando la norma della differenza dei vettori delle soluzioni, tra due iterazioni consecutive, è inferiore alla precisione richiesta.

Metodi iterativi

Esistono principalmente tre formulazioni del problema in base al valore della matrice ausiliaria: $\mathbf{S}$.

1. La matrice $\mathbf{S}$ è la matrice identità: metodo di **Richardson**.
2. La matrice $\mathbf{S}$ è la diagonale principale della matrice $\mathbf{A}$: metodo di **Jacobi**.
3. La matrice $\mathbf{S}$ è la porzione triangolare sinistra della matrice $\mathbf{A}$: metodo di **Gauss-Seidel**.

N.B.: la ricerca scientifica, sui metodi iterativi per la soluzione di sistemi lineari di grandi dimensioni, relativi a specifiche matrici strutturate, scaturenti da metodi agli elementi finiti, è molto attiva e specializzata. I tre metodi summenzionati sono soltanto la punta di un iceberg.

Letture aggiuntive

About computer performance

The performance of a computer is a complicated issue, a function of many interrelated quantities. These quantities include the application, the algorithm, the size of the problem, the high-level language, the implementation, the human level of effort used to optimize the program, the compiler's ability to optimize, the age of the compiler, the operating system, the architecture of the computer, and the hardware characteristics. Issues such as load on the system, accuracy of the clock, compiler options, version of the compiler, size of cache, bandwidth from memory, amount of memory, etc can effect the performance even when the processors are the same.

What is Linpack?

The Linpack package is a collection of Fortran subroutines for solving various systems of linear equations. (http://www.netlib.org/Linpack/) The software in Linpack is based on a decompositional approach to numerical linear algebra. The general idea is the following. Given a problem involving a matrix, one factors or decomposes the matrix into a product of simple, well-structured matrices which can be easily manipulated to solve the original problem. The package has the capability of handling many different matrix types and different data types, and provides a range of options. Linpack itself is built on another package called the BLAS. Linpack was designed in the late 70's and has been superseded by a package called LAPACK

How does the Linpack Benchmark performance relate to my application?

The performance of the Linpack benchmark is typical for applications where the basic operation is based on vector primitives such as added a scalar multiple of a vector to another vector. Many applications exhibit the same performance as the Linpack Benchmark. However, results should not be taken too seriously. In order to measure the performance of any computer it’s critical to probe for the performance of your applications. The Linpack Benchmark can only give one point of reference. In addition, in multiprogramming environments it is often difficult to reliably measure the execution time of a single program. We trust that anyone actually evaluating machines and operating systems will gather more reliable and more representative data.

Prestazioni computer

Year	Computer	# processors	Cycle time	Mflop/s
2003	HP Integrity Server rx2600 (1 proc 1.5GHz)	1	1.5 GHz	1635
2002	Intel Pentium 4 (3.06 GHz)	1	2.06 GHz	1414
2001	Fujitsu VPP5000/1	1	3.33 nsec	1156
2000	Fujitsu VPP5000/1	1	3.33 nsec	1156
1999	CRAY T916	4	2.2 nsec	1129
1995	CRAY T916	1	2.2 nsec	522
1994	CRAY C90	16	4.2 nsec	479
1993	CRAY C90	16	4.2 nsec	479
1992	CRAY C90	16	4.2 nsec	479
1991	CRAY C90	16	4.2 nsec	403
1990	CRAY Y-MP	8	6.0 nsec	275
1989	CRAY Y-MP	8	6.0 nsec	275
1988	CRAY Y-MP	1	6.0 nsec	74
1987	ETA 10-E	1	10.5 nsec	52
1986	NEC SX-2	1	6.0 nsec	46
1985	NEC SX-2	1	6.0 nsec	46
1984	CRAY X-MP	1	9.5 nsec	21
1983	CRAY 1	1	12.5 nsec	12
...				
1979	CRAY 1	1	12.5 nsec	3.4

Test LINPACK convenzionale su matrice 100x100

Prestazioni computer

Year	Computer	# of processors	Gflop/s	Size of problem
2003	Earth Simulator Computer, NEC	5104	35610	1,041,216
2001	ASCI White-Pacific, IBM SP Power 3	7424	7226	518,096
2000	ASCI White-Pacific, IBM SP Power 3	7424	4938	430,000
1999	ASCI Red Intel Pentium II Xeon core	9632	2379	362,880
1998	ASCI Blue-Pacific SST, IBM SP 604E	5808	2144	431,344
1997	Intel ASCI Option Red (200 MHz Pentium Pro)	9152	1338	235,000
1996	Hitachi CP-PACS	2048	368.2	103,680
1995	Intel Paragon XP/S MP	6768	281.1	128,600
1994	Intel Paragon XP/S MP	6768	281.1	128,600
1993	Fujitsu NWT	140	124.5	31,920
1992	NEC SX-3/44	4	20	6,144
1991	Fujitsu VP2600/10	1	4	1,000

Test LINPACK per computer altamente paralleli

Matrici piene in doppia precisione

Bibliografia

- Atkinson K. E., "Elementary Numerical Analysis", John Wiley & Sons, (1993)
- Buzzi Ferraris G., "Metodi numerici e Software in C++", Addison Wesley, (1998)
- Dongarra J., *Performance of Various Computers Using Standard Linear Equations Software*, University of Tennessee, Knoxville TN, Computer Science Technical Report Number CS-89–85

- http://www.cise.ufl.edu/~davis/sparse/
- http://www.top500.org/lists/linpack.php
- http://icl.cs.utk.edu/projects/papi/