

Capitolo 2

Sistemi lineari

2.1 Richiamo sulle matrici

Date due matrici A e B e $\alpha \in \mathbb{R}$ sono definite le seguenti operazioni:

- $A + B$ per $A, B \in \mathbb{R}^{m \times n}$
- αA
- $A \cdot B$ per $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $A \cdot B \neq B \cdot A$
- $A \in \mathbb{R}^{m \times n} \mapsto A^T \in \mathbb{R}^{n \times m}$ $A = (a_{ij}) \Rightarrow A^T = (a_{ji})$

Ricordiamo le seguenti proprietà:

- $(AB)^T = B^T A^T$
- Elemento neutro $I = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & 1 \end{pmatrix}$: $A \cdot I = A = I \cdot A$
- $A \in \mathbb{R}^{n \times n} \Rightarrow \exists A^{-1} : A \cdot A^{-1} = I = A^{-1} \cdot A$ se A è *non singolare* ($\det A \neq 0$)

Matrici particolari:

- simmetriche: $A \in \mathbb{R}^{n \times n}$ $A = A^T \iff A$ simmetrica
- triangolari superiori $A = \begin{pmatrix} a_{11} & & a_{1n} \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}$
- diagonali $A = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}$

Le matrici notevoli rimangono tali rispetto alle operazioni definite?

	triangolare superiore	diagonale	simmetrica
$A + B$	V	V	V
αA	V	V	V
AB	V	V	F
A^{-1}	V	V	V

Tabella 2.1: Matrici notevoli ed operazioni

Determinante $\det A : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$

Regola di Laplace: $\det A = \pm a_{i1} \cdot \det A_{i1} \pm a_{i2} \cdot \det A_{i2} \pm \dots \pm a_{in} \cdot \det A_{in}$ (segni da scegliere con la “regola della scacchiera”: segno + in alto a sinistra e poi si alternano).

Non vale nessuna regola per $\det(A + B)$, ma valgono:

- $\det(\alpha A) = \alpha^n \det A$
- Formula di Binet: $\det(AB) = \det A \cdot \det B$ (tutti gli oggetti devono essere quadrati)
- $\det(A^{-1}) = \frac{1}{\det A}$ (se esiste A^{-1} il determinante è per forza non nullo)
- $\det(A^T) = \det(A)$
- Le matrici triangolari superiori e diagonali hanno come determinante il prodotto degli elementi sulla diagonale
- Se B è uguale ad A con due righe scambiate, il determinante cambia segno

2.2 Norme

Supponiamo che i dati siano affetti da un errore (vengono da misurazioni o da arrotondamenti del calcolatore): ci vuole un numero che riassume la grandezza di un vettore o una matrice in un unico valore. Questo sarà chiamato *norma*.

2.2.1 Norme vettoriali

2.2.1.1 Norma 2

Dato un vettore $x \in \mathbb{R}^3$ si definisce la norma come la lunghezza del vettore: $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$. In generale si definisce come *norma 2*:

$$x \in \mathbb{R}^n \mapsto \|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2} = \sqrt{x^T x}$$

Proprietà:

1. $\forall x \in \mathbb{R}^n : \|x\|_2 \geq 0; \|x\|_2 = 0 \iff x = 0$

2. $\forall x \in \mathbb{R}^n, \forall \alpha \in \mathbb{R} \quad \|\alpha x\|_2 = |\alpha| \|x\|_2$
3. $\forall x, y \in \mathbb{R}^n \quad \|x + y\|_2 \leq \|x\|_2 + \|y\|_2$ (disuguaglianza triangolare)

2.2.1.2 Generalizzazione della norma

Si chiama *norma* qualunque misura che restituisce un numero da un vettore e rispetta gli assiomi della norma:

1. $\forall x \in \mathbb{R}^n : \|x\| \geq 0; \|x\| = 0 \iff x = 0$
2. $\forall x \in \mathbb{R}^n, \forall \alpha \in \mathbb{R} \quad \|\alpha x\| = |\alpha| \|x\|$
3. $\forall x, y \in \mathbb{R}^n \quad \|x + y\| \leq \|x\| + \|y\|$ (disuguaglianza triangolare)

Due esempi di norme che rispettano questi assiomi sono:

- $\|x\|_1 = |x_1| + |x_2| + \dots + |x_n|$
- $\|x\|_\infty = \max_{i=1..n} |x_i|$

2.2.1.3 Teorema

Date due norme diverse $\|\cdot\|'$ e $\|\cdot\|''$, $\exists \alpha, \beta > 0$ tali che $\forall x \in \mathbb{R}^n$

$$\alpha \|x\|'' \leq \|x\|' \leq \beta \|x\|''$$

Tutte le norme sono metricamente equivalenti.

$\ \cdot\ '$	$\ \cdot\ ''$	α	β
1	∞	1	n
1	2	1	\sqrt{n}
2	∞	1	\sqrt{n}

Tabella 2.2: Teorema del confronto per le norme

2.2.1.4 Sfera e palla unitaria

Si dice *sfera unitaria* l'oggetto creato da una norma così fatto:

$$\{x \in \mathbb{R}^n : \|x\| = 1\}$$

Sostituendo l'uguaglianza con una disuguaglianza si ottiene il volume racchiuso dalla sfera unitaria, cioè la *palla unitaria*.

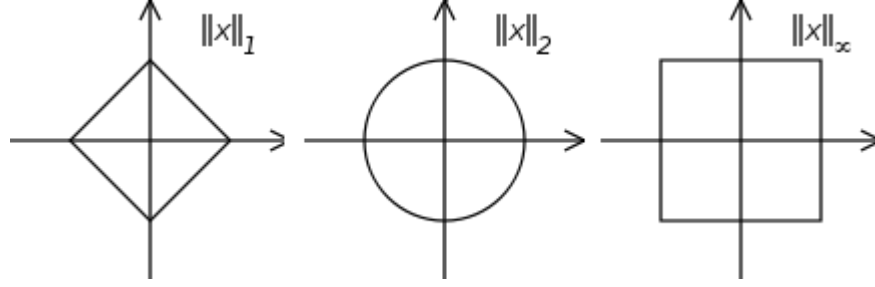


Figura 2.1: Sfera e palla unitaria

2.2.2 Norme matriciali

Per una matrice quadrata si chiama *norma*

$$\|\cdot\| : A \in \mathbb{R}^{n \times n} \mapsto \|A\| \in \mathbb{R}$$

Con le proprietà:

1. $\|A\| \geq 0 \forall A \in \mathbb{R}^{n \times n}; \|A\| = 0 \iff A = 0$
2. $\|\alpha A\| = |\alpha| \|A\| \forall \alpha \in \mathbb{R}, A \in \mathbb{R}^{n \times n}$
3. $\|A + B\| \leq \|A\| + \|B\| \forall A, B \in \mathbb{R}^{n \times n}$
4. $\|A \cdot B\| \leq \|A\| \cdot \|B\| \forall A, B \in \mathbb{R}^{n \times n}$

Non sempre si riesce a definire la norma matriciale imitando quella vettoriale. Un caso in cui si può è derivato dalla norma 2: la *norma di Frobenius* rispetta le quattro proprietà:

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n a_{ij}^2}$$

2.2.2.1 Norma matriciale indotta

Per costruire altre norme di matrici che rispettino le proprietà e siano legate alle norme vettoriali si usa la *norma matriciale indotta*, che aggiunge una proprietà alle quattro già definite.

Data $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ norma matriciale, $A \in \mathbb{R}^{n \times n}$, considero tutti i vettori della sfera unitaria e vado a vedere cosa succede nel caso del prodotto matrice per vettore ($Ax \in \mathbb{R}^n$):

$$\max_{\|x\|=1} \|Ax\| =: \|A\|$$

$\|A\|$ la chiamo *norma indotta* della matrice, si dimostra che soddisfa le proprietà. Interpretando questa come trasformazione, ci si chiede quanto sarà grande il risultato di questa trasformazione.

2.2.2.2 Proposizione

Se $\|\cdot\|$ è una norma matriciale indotta, allora

$$\forall v \in \mathbb{R}^n \quad \|Av\| \leq \|A\| \|v\|$$

Se abbiamo una qualunque norma matriciale, possiamo maggiorare la norma del prodotto matriciale con il prodotto fra norme di matrici (proprietà 4), con questa proposizione, avendo l'ipotesi ulteriore che sia una norma matriciale indotta, possiamo usare anche vettori.

Dimostrazione: Prendo un vettore *normalizzato* $x = \frac{v}{\|v\|}$, esso è nella sfera unitaria infatti $\|x\| = \left\| \frac{v}{\|v\|} \right\| = \|\alpha v\|$ con $\alpha = \frac{1}{\|v\|} \implies \|x\| = |\alpha| \|v\| = \frac{1}{\|v\|} \|v\| = 1$.

Quindi $Ax = A \frac{v}{\|v\|} = \frac{1}{\|v\|} Av$ ($\frac{1}{\|v\|}$ è uno scalare, A una matrice, v un vettore), applico la norma: $\|Ax\| = \|\alpha Av\| = \frac{1}{\|v\|} \|Av\| \implies \|Av\| = \|Ax\| \cdot \|v\| \leq \|A\| \cdot \|v\|$ (per la definizione di norma indotta $\|Ax\|$ è maggiorato dalla norma di A). In conclusione $\|Av\| \leq \|A\| \cdot \|v\|$.

Introduco quindi la proprietà 5:

$\|Av\| \leq \|A\| \cdot \|v\| \quad \forall A \in \mathbb{R}^{n \times n}, v \in \mathbb{R}^n$ solo se $\|\cdot\|$ è indotta.

Osservazione Se $\|\cdot\|$ è indotta allora $\|I\| = \max_{\|x\|=1} \|Ix\| = \max_{\|x\|=1} \|x\| = 1$.

Nel caso di $\|\cdot\|_F$, $\|I\|_F = \sqrt{n \cdot 1} = \sqrt{n} \neq 1$, cioè la norma di Frobenius non è indotta.

2.2.2.3 Norme infinito e uno

La norma infinito (indotta) è anche calcolabile come il massimo delle somme dei moduli per riga:

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

La norma uno indotta invece è il massimo delle somme dei moduli sulle colonne:

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

Prendiamo ad esempio $A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & -4 & -1 \\ 0 & -5 & 2 \end{pmatrix}$

Facendo la somma sulle righe troviamo $\|A\|_\infty = 7$, mentre $\|A\|_1 = 11$.
La norma due indotta esiste ma non è calcolabile esplicitamente.

2.2.2.4 Teorema

$\forall |||', |||''$ norme matriciali (anche non indotte), $\exists \alpha, \beta > 0$ tale che $\forall A \in \mathbb{R}^{n \times n}$

$$\alpha \|A\|'' \leq \|A\|' \leq \beta \|A\|''$$

2.2.3 Errore inerente nei sistemi lineari

L'input del problema è dato dalla matrice dei coefficienti $A \in \mathbb{R}^{n \times n}$ e dalla colonna dei termini noti $b \in \mathbb{R}^n$, l'output è la colonna delle soluzioni $x \in \mathbb{R}^n$ tali che $Ax = b$.

L'errore inerente deriva da input non corretti: normalmente i sistemi lineari hanno un errore sul termine noto (è più raro che i componenti della matrice siano sbagliati). Analizzeremo quindi il problema con queste condizioni.

Supponiamo di avere il vettore $\delta b \in \mathbb{R}^n$ che contiene gli errori commessi sui valori del vettore dei termini noti: devo risolvere $A\tilde{x} = b + \delta b$ con \tilde{x} l'output perturbato e $b + \delta b$ l'input perturbato.

Bisogna calcolare l'errore relativo e il condizionamento del problema.

Errore in input $\frac{\|b + \delta b - b\|}{\|b\|}$ è la norma degli errori assoluti diviso quella del vettore quindi $\varepsilon_b = \frac{\|\delta b\|}{\|b\|}$, senza bisogno di porre la condizione che b sia non nullo poichè in quel caso lo sarebbe anche la soluzione.

Errore in output $\varepsilon_x := \frac{\|\tilde{x} - x\|}{\|x\|}$
 Il condizionamento del problema sarà $c = \frac{\varepsilon_x}{\varepsilon_b}$.

2.2.3.1 Teorema

Se $|||$ indotta, allora $\varepsilon_x \leq \|A\| \cdot \|A^{-1}\| \varepsilon_b$.

Definisco il *condizionamento della matrice* come $\mu(A) = \|A\| \cdot \|A^{-1}\|$.

Dimostrazione Considero $\delta x = \tilde{x} - x$ da cui $\varepsilon_x = \frac{\|\delta x\|}{\|x\|}$ e studio i due componenti.

Considero il denominatore: $Ax = b \Rightarrow \|Ax\| = \|b\| \Rightarrow \frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}$ (applicando la quinta proprietà).

Considero il numeratore: $\tilde{x} = x + \delta x \Rightarrow A(x + \delta x) = b + \delta b$ Semplificando: $A\delta x = \delta b \Rightarrow (A^{-1}A)\delta x = A^{-1}\delta b \Rightarrow \delta x = A^{-1}\delta b$. Applico la norma $\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \|\delta b\|$.

Unendo le due parti: $\varepsilon_x \leq \frac{\|A^{-1}\| \|\delta b\| \|A\|}{\|b\|} = \mu(A) \varepsilon_b$.

Una matrice si dice *ben condizionata* quando $\mu(A)$ è basso.

Esempio $\begin{cases} x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$

Stima del condizionamento: calcoliamo $A = \begin{pmatrix} 1 & 1 \\ 1001 & 1000 \end{pmatrix}$ e $A^{-1} = \begin{pmatrix} -1000 & 1 \\ 1001 & -1 \end{pmatrix}$,

la norma vale $\|A\|_{\infty} = 2001$ e $\|A^{-1}\|_{\infty} = 1002$.

$$\mu(A) = 2001 \cdot 1002 \approx 2 \cdot 10^6$$

Spostando di un centesimo un valore in input devo aspettarmi un errore relativo di 20000.

2.2.3.2 Condizionamento minimo per una matrice

Per la matrice identità vale $\|I\| = 1$.

Essendo $I = A \cdot A^{-1}$, $\|A \cdot A^{-1}\| = \|I\| = 1$, ma $\|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\|$ che è il condizionamento della matrice $\mu(A)$. Quindi $\mu(A) \geq 1$.

2.3 Risoluzione dei sistemi lineari

Per risolvere un sistema lineare si deve procedere prima a trasformare la matrice in una matrice triangolare superiore e poi risolvere questo nuovo sistema.

2.3.1 Risoluzione di sistemi triangolari superiori: sostituzione all'indietro

I sistemi triangolari superiori sono fatti così

$$\begin{pmatrix} a_{11} & & \cdots & a_{1n} \\ & a_{22} & & \\ & & \ddots & \vdots \\ 0 & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Un sistema triangolare superiore è immediatamente risolvibile sull'ultima componente ($a_{nn}x_n = b_n \Rightarrow x_n = \frac{b_n}{a_{nn}}$), quindi passo alla penultima equazione:

$$a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \Rightarrow x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}$$

e itero il procedimento.

2.3.1.1 Sostituzione all'indietro

Questo procedimento è chiamato *sostituzione all'indietro*.

Per $i = n-1, n-2, \dots, 2, 1$:

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$

Costo totale Si ha una operazione di inizializzazione, poi una sequenza di cicli con una divisione e tante moltiplicazioni (all'interno della sommatoria).

$$1 + (n - (n - 1) + 1) + (n - (n - 2) + 1) + \dots + n = 1 + 2 + \dots + n = \frac{n(n+1)}{2} = \frac{n^2+n}{2} \doteq \frac{n^2}{2}$$

In prima approssimazione

$$costo \doteq \frac{n^2}{2}$$

2.3.2 Riduzione a sistema triangolare: algoritmo di eliminazione gaussiana

Una strategia percorribile per ridurre un sistema alla forma triangolare superiore sarebbe moltiplicare per un coefficiente una riga della matrice e sommarla o sottrarla ad un'altra. Lo scopo dell'eliminazione gaussiana è proprio ottenere gli zeri una colonna alla volta.

Alla k -esima colonna abbiamo

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & & & & & & a_{1n}^{(k)} \\ 0 & a_{22}^{(k)} & & & & & \\ & 0 & \ddots & & & & \\ & & & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & & & \vdots & & \\ & & & & a_{ik}^{(k)} & & a_{in}^{(k)} \\ & & & & \vdots & & \\ & & & & a_{nk}^{(k)} & & a_{nn}^{(k)} \end{pmatrix}$$

Per azzerare $a_{ik}^{(k)}$, scelgo di usare l'elemento diagonale $a_{kk}^{(k)}$, detto *perno* o *pivot*:

definisco quindi il moltiplicatore $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$

L'operazione che verrà svolta sarà quindi i -esima riga $-m_{ik} \cdot k$ -esima riga; l'effetto è quello di lasciare invariata la soluzione del sistema ottenendo nella posizione ik uno zero, ovvero $a_{ik}^{(k+1)} = 0$. Va ripetuta su tutte le righe da azzerare, perciò per $i = k + 1 \rightarrow n$. L'analoga operazione per la colonna dei termini noti è $b_i^{(k)} - m_{ik}b_k^{(k)} = b_i^{(k+1)}$.

2.3.2.1 Esecuzione manuale su un esempio

$$A^{(1)} = \begin{pmatrix} 2 & -4 & 1 \\ 6 & -14 & 8 \\ -2 & 0 & 6 \end{pmatrix} b^{(1)} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Il primo passo prevede di azzerare la prima colonna, il pivot è a_{11} , i moltiplicatori per la riga del perno sono $m_{21} = 3$, $m_{31} = -1$: prendo la seconda riga e sottraggo tre volte la prima riga, prendo la terza riga e sottraggo -1 volte la prima riga (cioè sommo); per il termine noto prendo il secondo elemento e gli tolgo tre volte il primo, e così via.

$$A^{(2)} = \begin{pmatrix} 2 & -4 & 1 \\ 0 & -2 & 5 \\ 0 & -4 & 7 \end{pmatrix} \quad b^{(2)} = \begin{pmatrix} 1 \\ -4 \\ 2 \end{pmatrix}$$

Vado alla colonna successiva, il pivot è a_{22} , il moltiplicatore invece $m_{32} = 2$. Devo prendere la seconda riga, raddoppiarla e sottrarla alla terza, e ricopiare le righe fino al perno inclusa.

$$A^{(3)} = \begin{pmatrix} 2 & -4 & 1 \\ 0 & -2 & 5 \\ 0 & 0 & -3 \end{pmatrix} \quad b^{(3)} = \begin{pmatrix} 1 \\ -14 \\ 10 \end{pmatrix}$$

Il nuovo sistema ha le soluzioni dell'originale, adesso si può procedere col metodo della sostituzione all'indietro.

$$\begin{cases} 2x_1 - 4x_2 + x_3 = 1 \\ -2x_2 + 5x_3 = -4 \\ -3x_3 = 10 \end{cases}$$

Le soluzioni sono $x_3 = -\frac{10}{3}$, $x_2 = -\frac{19}{3}$, $x_1 = -\frac{21}{2}$.

2.3.2.2 Algoritmo e costo del metodo

Algoritmo 2.1 Algoritmo di eliminazione gaussiana

$$A^{(1)} = A, \quad b^{(1)} = b$$

$$k = 1, \dots, n-1$$

$$a_{kk}^{(k)} = \text{perno}$$

$$i = k+1, \dots, n$$

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad (\text{una divisione per ogni } i)$$

$$b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)} \quad (\text{una moltiplicazione per ogni } i)$$

$$a_{ik}^{(k+1)} = 0$$

$$j = k+1, \dots, n$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)} \quad (n-k \text{ moltiplicazioni per ogni } i)$$

In totale all'interno del ciclo su i ci sono $n-k+2$ operazioni, considerando solo moltiplicazioni e divisioni, quindi ci sono in totale $(n-k)(n-k+2)$ operazioni per ogni k .

Procediamo alla somma su k :

$$\begin{aligned} \sum_{k=1}^{n-1} (n-k)(n-k+2) &= \sum_{k=1}^{n-1} (n^2 - nk + 2n - nk + k^2 - 2k) = \\ &= \sum_{k=1}^{n-1} (n^2 + 2n - 2(n+1)k + k^2) = \sum_{k=1}^{n-1} (n^2 + 2n) - \sum_{k=1}^{n-1} 2(n+1)k + \\ &\quad \sum_{k=1}^{n-1} k^2 = \end{aligned}$$

$$(n-1)(n^2 + 2n) - 2(n+1) \sum_{k=1}^{n-1} k + \sum_{k=1}^{n-1} k^2 \doteq n^3 - n^3 + \frac{n^3}{3},$$

$$\text{avendo usato le uguaglianze } \sum_{k=1}^{n-1} k \doteq \frac{n^2}{2}, \quad \sum_{k=1}^{n-1} k^2 \doteq \frac{n^3}{3}.$$

Il metodo di Gauss ha costo asintotico di $\frac{n^3}{3}$.

2.3.2.3 Analisi dell'errore

Il metodo di Gauss funziona sempre?

$$A = \begin{pmatrix} -1 & -4 & 0 \\ 2 & 8 & 1 \\ 2 & -3 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ -4 \\ 7 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} -1 & -4 & 0 \\ 0 & 0 & 1 \\ 0 & -11 & 1 \end{pmatrix} \quad b^{(2)} = \begin{pmatrix} 2 \\ 0 \\ 11 \end{pmatrix}$$

Per il prossimo passo il perno sarebbe zero! Bisogna modificare la scelta del perno, prendendo un indice i_0 tale che $a_{i_0 k} \neq 0$. Scambio la seconda e la terza riga sia in A che in b per far tornare il perno sulla diagonale.

$$A^{(3)} = \begin{pmatrix} -1 & -4 & 0 \\ 0 & -11 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad b^{(3)} = \begin{pmatrix} 2 \\ 11 \\ 0 \end{pmatrix}$$

Il metodo di Gauss è *stabile* (errore algoritmico che non supera l'errore inerente) se gli elementi di $A^{(k)}$ non sono molto più grandi degli elementi di partenza A . Altro esempio:

$$A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad x = \begin{pmatrix} 1 \\ 1 - \varepsilon \end{pmatrix}$$

$$m_{21} = \frac{1}{\varepsilon}$$

$$A^{(2)} = \begin{pmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{pmatrix} \quad b^{(2)} = \begin{pmatrix} 1 \\ 1 - \frac{1}{\varepsilon} \end{pmatrix}$$

Se ε è molto piccolo, tale da non essere sentito dalla macchina, fa sballare i calcoli, infatti

$$x_2 = \frac{1 - \frac{1}{\varepsilon}}{-\frac{1}{\varepsilon}} \approx \frac{-\frac{1}{\varepsilon}}{-\frac{1}{\varepsilon}} = 1 \quad x_1 = \frac{1 - x_2}{\varepsilon} = 0 \text{ che è diverso dalla soluzione esatta. Infatti } A^{(2)} \text{ ha un elemento } -\frac{1}{\varepsilon} \text{ molto più grande di quelli di } A.$$

2.3.2.4 Strategia di Pivoting Parziale

Per evitare che ci siano moltiplicatori troppo grandi, si sceglie il perno più grande possibile. Definisco il perno tale che sia il massimo elemento in modulo sulla colonna:

$$|a_{i_0 k}| = \max_{i \geq k} |a_{ik}|$$

Questa strategia, detta di *pivoting parziale*, evita la propagazione degli errori: se viene trovato l'elemento massimo, viene scambiata la riga in cui si trova l'algoritmo (la soluzione non viene modificata).

Viene usata anche in Matlab: le soluzioni si trovano con l'istruzione $x = A \backslash b$.

2.3.2.5 Applicazioni

1. Costruiamo la matrice $(A|I)$ affiancando la matrice A con la matrice identità. Applicando l'algoritmo di Gauss otterremo una matrice così fatta $(I|A^{-1})$ dove A^{-1} è l'inversa. Analogamente, costruendo una matrice del

tipo $(A|b_1|b_2)$ dove b_1 e b_2 sono colonne di termini noti, ricaveremo le soluzioni x_1 e x_2 ;

2. Data una matrice A , se l'algoritmo di Gauss procede senza effettuare scambi di riga, il determinante non cambia e all'ultimo passo avremo una matrice triangolare, il cui determinante è il prodotto degli elementi sulla diagonale: $\det A = a_{11}^{(n)} \cdot \dots \cdot a_{nn}^{(n)}$. Nel caso di scambi di riga, bisogna tenerne traccia, perchè ad ogni scambio cambia il segno del determinante;
3. Definita l'ultima matrice del metodo di Gauss $A^{(n)} =: U$ e costruita una matrice L triangolare inferiore con sulla diagonale tutti 1 e per elementi i moltiplicatori con indici corrispondenti, si afferma il teorema della *fattorizzazione LU*: se il metodo di Gauss non esegue scambi, allora $L \cdot U = A$.

$$L = \begin{pmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{32} & 1 & & \\ \vdots & & & \ddots & \\ m_{n1} & m_{n2} & & \dots & 1 \end{pmatrix}$$

2.3.3 Esercitazione con Matlab

Algoritmo 2.2 Matrici

$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 4 \\ 7 & 8 & 0 & 7 \end{bmatrix}$
 $b = [1 \ 4 \ 7]$
 $A \setminus b$ risolve il sistema
 $R : [1 \ 0 \ 0]'$ (una colonna)
 $\det(A)$
 $\text{inv}(A)$ Inversa

$$H_{ij} = \frac{1}{i+j-1}$$

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

Algoritmo 2.3 Matrici di Hilbert

```

H = hilb(6)
x = ones(6,1) Genera una colonna di tutti uno
b = H * x Calcola i termini noti
db = rand(6,1) * 1e-6 Genera una perturbazione di  $\sim 10^{-6}$ 
xx = H \ (b + db)
eb = norm(db) / norm(b)  $4.89 \cdot 10^{-7}$ 
ex = norm(xx - x) / norm(x) R=0.9157 L'errore in output è  $10^6$  volte più
grande di quello in input
cond(H) calcola il condizionamento di H:  $1.49 \cdot 10^7$ 
for n = 1 : 20, H = hilb(n); muH(n) = cond(H); end
plot(muH)
semilogy(muH) Grafico con y in scala logaritmica
H = hilb(12);
x = ones(12,1);
b = H * x;
xx = H \ b segnala errore.

```

$$W = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \end{bmatrix}$$

$$A_{ij} = \begin{cases} \frac{i}{j} & j \geq i \\ \frac{j}{i} & j < i \end{cases}$$

$$A = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1 & 2/3 & 1/2 \\ 1/3 & 2/3 & 1 & 3/4 \\ 1/4 & 1/2 & 3/4 & 1 \end{pmatrix}$$

Algoritmo 2.4 Matrici notevoli: Wilkinson e Lehmer

$W = wilkinson(6)$

$A = gallery('lehmer', 6)$

$cond(W)$ circa 13

$cond(A)$ circa 29

$dim = 10 : 10 : 200;$

$for\ n = dim, W = wilkinson(n); muW(n) = cond(W); end$

$plot(dim, muW(dim))$ vedo che il condizionamento ha un andamento lineare nella dimensione

Per Lehmer non si capisce, dobbiamo vedere un grafico con entrambi gli assi in scala logaritmica:

$loglog(dim, muA(dim))$ è una retta inclinata ($\Rightarrow \mu$ potenza di n), ma come procede?

$loglog(dim, muA(dim), 'b', dim, dim.^2, 'r', dim, dim.^3, 'g')$ Il condizionamento della matrice di Lehmer va circa come il quadrato rispetto alla dimensione.
