Calcolo Numerico

Luca De Paulis

11 aprile 2021

Indice

1

Aritmetica di macchina

1.1 RAPPRESENTAZIONE DEI NUMERI DI MACCHINA

Il primo problema da risolvere quando si vuole fare analisi numerica è scegliere un metodo per rappresentare i numeri reali su una macchina. Infatti un generico numero reale ha potenzialmente una scrittura decimale infinita, dunque essendo le risorse disponibili in una macchina *finite* dobbiamo trovare un modo di approssimarlo.

1.1.1 Virgola fissa

Il primo metodo è il metodo **a virgola fissa**: in questo metodo si rappresentano tutti i numeri nella loro forma decimale normale e si considerano esattamente k cifre dopo la virgola.

Questo metodo è molto semplice e ci consente di fare operazioni elementari (come le somme o i prodotti) immediatamente ("in colonna"), tuttavia ha anche degli svantaggi evidenti, come

- il range dei numeri rappresentabili su n cifre/bit è piccolo;
- siccome il numero di bit dedicato ai numeri dopo la virgola è basso, la precisione è molto bassa e assolutamente non adeguata ad applicazioni di analisi numerica.

Per questo è stata inventata la rappresentazione in virgola mobile.

1.1.2 Virgola mobile

Innanzitutto dobbiamo trovare un modo standard per rappresentare un qualsiasi numero reale. Per far ciò ci viene in aiuto il seguente teorema.

Teorema 1.1.1 – Teorema di rappresentazione in base

Sia $x \in \mathbb{R}$, $x \neq 0$ e sia $\beta \geq 2$ una base di rappresentazione.
Allora esistono e sono univocamente determinati un **esponente** $p \in \mathbb{Z}$ e una successione di **cifre** $(d_i)_{i \in \mathbb{N}}$ (con $d_i \in \mathbb{Z}$ per ogni i) per cui vale che

(i) $d_1 \neq 0$;

(ii) $0 \leq d_i \leq \beta - 1$ per ogni i;

(iii) la successione d non è definitivamente uguale a $\beta - 1$, ovvero per ogni $k > 0$ esiste un $j \geq k$ tale che $d_j \neq \beta - 1$;

(iv) si ha che

$$x = \operatorname{sgn}(x) \cdot \beta^p \cdot \left(\sum_{i=1}^{\infty} d_i \beta^{-i} \right).$$

La rappresentazione di x nella forma data dal Theorem 1.1.1 si dice **rappresentazione normalizzata in virgola mobile**.

Esempio 1.1.2. Consideriamo ad esempio il numero reale 123. Per il Theorem 1.1.1 possiamo esprimere 123 come

$$123 = +10^3 \cdot (0.123) = +10^3 \cdot (1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3}).$$

Questa rappresentazione è l'unica possibile se imponiamo che d_1 sia diverso da 0 e che le cifre della rappresentazione siano tutte considerate come *cifre decimali* moltiplicate per una certa potenza della base (in questo caso 3).

La rappresentazione data dal Theorem 1.1.1 è essenzialmente il concetto di *notazione scientifica*: in notazione scientifica portiamo il numero reale in modo che abbia una singola cifra diversa da zero prima della virgola, mentre nella notazione data dal Teorema quella cifra diventa la prima cifra dopo la virgola. Inoltre, il Teorema ci garantisce che questa rappresentazione non è solamente valida in base 10, ma lo è in qualsiasi base (noi lo useremo in particolare in base 2).

Il fattore β^p viene detto **esponente**, mentre la parte decimale (corrispondente alla sommatoria) viene detta **mantissa**.

Osservazione 1.1.1. Le ipotesi (i) e (iii) del Teorema sono fondamentali per l'unicità della rappresentazione. In effetti

- se venisse a mancare la prima condizione avremmo che

$$123 = +10^3 \cdot (0.123) = +10^4 \cdot (0.0123) = +10^5 \cdot (0.00123) = \dots;$$

- se venisse a mancare la terza condizione (che ci dice che la successione d non può terminare con una sequenza infinita di $(\beta - 1)$, ovvero il numero non può finire con $\beta - 1$ periodico) avremmo dei problemi più subdoli, che derivano dalla non esistenza del "9 *periodico*" (dimostrata in appendice, Proposition A.1.1).

  Infatti in qualsiasi base β il numero $0.\overline{(\beta - 1)}$ è uguale a 1, quindi se ammettessimo entrambe le rappresentazioni verrebbe meno l'unicità.

Notazione. Useremo la notazione $(n)_\beta$ per riferirci al numero n espresso in base β.

Ad esempio il numero $(1011)_2$ si riferisce al numero

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11 \text{ (in base 10)}.$$

Per rappresentare i numeri reali in macchina dobbiamo quindi risolvere ancora due problemi:

- la macchina usa (nella maggior parte dei casi) la base 2, quindi dobbiamo poter trasformare da base 10 in base 2;
- dobbiamo rappresentare i numeri infiniti in modo approssimato.

Per quanto riguarda il primo, l'algoritmo per trasformare un numero decimale (in base 10) in un numero in base 2 è il seguente:

1. si trasforma la parte intera (le cifre prima della virgola) in base 2 tramite divisioni successive per 2;
2. si trasforma la parte decimale (le cifre dopo la virgola) in base 2 tramite *moltiplicazioni* successive per 2.

Esempio 1.1.3. Trasformiamo 3.15 in base 2: ovviamente $(3)_{10} = (11)_2$ e quindi rimane solo da trasformare la parte decimale.

L'algoritmo consiste nel prendere il numero decimale 0.15 e moltiplicarlo per 2 ripetutamente: se il risultato ha come cifra prima della virgola uno 0 aggiungiamo uno 0 alla rappresentazione in base 2, altrimenti un 1. Nella pratica:

$$\begin{aligned}
0.15 \cdot 2 &= \boxed{0}.30 \\
0.30 \cdot 2 &= \boxed{0}.60 \\
0.60 \cdot 2 &= \boxed{1}.20 \\
0.20 \cdot 2 &= \boxed{0}.40 \\
0.40 \cdot 2 &= \boxed{0}.80 \\
0.80 \cdot 2 &= \boxed{1}.60 \\
0.60 \cdot 2 &= \boxed{1}.20 \\
&\vdots
\end{aligned}$$

Osserviamo che ripetendo il procedimento la sequenza $(0.60, 0.20, 0.40, 0.80)$ si ripete all'infinito, dunque il numero in base 2 termina con le cifre 1001 ripetute periodicamente. La parte decimale in base 2 corrisponde quindi a $(0.00\overline{1001})_2$.

Il numero completo è quindi

$$(11.00\overline{1001})_2 = 2^2 \cdot (0.1100\overline{1001})_2.$$

Per rappresentare i numeri infiniti dobbiamo approssimare la parte decimale, considerando solo un numero finito t di cifre dopo la virgola. Per far ciò esistono due metodi (che esamineremo più nel dettaglio nel seguito):

- nel caso del **troncamento** si considerano le prime t cifre dopo la virgola e si scartano tutte le successive;
- nel caso dell'**arrotondamento** analizziamo la cifra decimale di posto $t + 1$:
  - se d_{t+1} appartiene all'intervallo $\left[0, \frac{\beta}{2}\right)$ (dove β è la base) consideriamo semplicemente le prime t cifre;
  - altrimenti (se $d_{t+1} \in \left[\frac{\beta}{2}, \beta\right]$ aggiungiamo 1 all'ultima cifra decimale (come riporto, quindi lo propaghiamo in caso sia necessario).

Esempio 1.1.4. Riprendiamo il numero considerato precedentemente, ovvero

$$(3.15)_{10} = 2^2 \cdot (0.1100\overline{1001})_2$$

e approssimiamolo a $t = 8$ cifre decimali.

Nel caso del troncamento basta considerare le prime 8 cifre decimali, quindi il numero troncato è

$$2^2 \cdot (0.11001001)_2.$$

Nel caso dell'arrotondamento invece dobbiamo considerare la nona cifra decimale: espandendo la rappresentazione decimale osserviamo che la nona cifra è

$$0.11001001\underline{1}00\ldots;$$

siccome $1 \in \left[\frac{2}{2}, 2\right] = [1, 2]$ dobbiamo *approssimare per eccesso*, ottenendo

$$2^2 \cdot (0.11001010)_2.$$

1.1.3 Insieme dei numeri di macchina

Siccome un singolo numero può occupare una quantità fissa in memoria (tipicamente 32 o 64 bit) dobbiamo fissare dei limiti per l'esponente e per il numero di cifre decimali che possiamo usare per la rappresentazione. Osserviamo inoltre che, se le nostre risorse sono limitate, aumentando il numero di cifre decimali disponibili dobbiamo necessariamente diminuire lo spazio dedicato a memorizzare l'esponente e viceversa.

Definizione 1.1.5 – Insieme dei numeri di macchina

ia $\beta \geq 2$ una base, t il numero di cifre decimali utilizzabili e siano $m,\ M \in \mathbb{Z}$ gli estremi per l'esponente (ovvero ogni esponente p rappresentabile deve appartenere a $[-m,\ M]$). Si dice allora **insieme dei numeri di macchina** l'insieme

$$\mathcal{F}(\beta, t, m, M) := \{0\} \cup \Big\{ x \in \mathbb{R} \ : x = \operatorname{sgn}(x) \cdot \beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}, \quad d_1 \neq 0, \quad 0 \leq d_i \leq \beta - 1, \quad -m \leq p \leq M \Big\}.$$

Osservazione 1.1.2. Nelle condizioni in cui specifichiamo quali numeri sono rappresentabili compaiono le ipotesi (i) e (ii) del Teorema di rappresentazione in base, ma non l'ipotesi (iii). Quest deriva dal fatto che i numeri di macchina hanno una precisione finita (ovvero hanno al più t cifre decimali), dunque non è possibile avere un numero che termina in $\beta - 1$ periodico.

OMEGA GRANDE Chiamiamo Ω il più grande numero di macchina. Possiamo costruirlo in questo modo:

- scegliamo come segno $+$;
- scegliamo come esponente il massimo esponente possibile, ovvero $p = M$;
- poniamo tutte le t cifre decimali uguali al massimo possibile, che in base β è $\beta - 1$.

Segue quindi che

$$\Omega := +\beta^M \sum_{i=1}^{t} (\beta - 1)\beta^{-i}.$$

Semplificando l'espressione di Ω tramite la formula per la serie geometrica si ottiene

$$\begin{aligned}
\Omega &= +\beta^M \sum_{i=1}^{t} (\beta - 1)\beta^{-i} \\
&= \beta^M (\beta - 1) \sum_{i=1}^{t} (1/\beta)^i \\
&= \beta^M (\beta - 1) \left(\sum_{i=0}^{t} (1/\beta)^i - (1/\beta)^0 \right) \\
&= \beta^M (\beta - 1) \left(\frac{1 - (1/\beta)^{t+1}}{1 - 1/\beta} - 1 \right)
\end{aligned}$$

Moltiplicando numeratore e denominatore della frazione per β:

$$\begin{aligned}
&= \beta^M (\beta - 1) \left(\frac{\beta - (1/\beta)^t}{\beta - 1} - 1 \right) \\
&= \beta^M (\beta - 1) \cdot \frac{\beta - (1/\beta)^t - \beta + 1}{\beta - 1} \\
&= \beta^M \left(1 - \beta^{-t}\right).
\end{aligned}$$

Esempio 1.1.6. Consideriamo l'insieme $\mathcal{F}(10, 3, 2, 2)$: in questo sistema possiamo rappresentare numeri in base 10 con al massimo 3 cifre decimali e con un esponente compreso tra -2 e 2.

Il massimo numero rappresentabile in questo sistema è

$$\Omega = +10^2(0.999) = 10^2(1 - 10^{-3}).$$

OMEGA PICCOLO Chiamiamo ω il più piccolo numero *positivo* rappresentabile in un sistema. Possiamo costruirlo in questo modo:

- siccome è positivo, il segno è necessariamente +;
- come esponente scegliamo il più piccolo esponente possibile, ovvero $p = -m$;
- siccome le cifre decimali non possono essere tutte uguali a 0 poniamo $d_1 = 1$ e $d_i = 0$ per ogni $i > 1$.

Segue quindi che

$$\omega = +\beta^{-m}\left(1 \cdot \beta^{-1}\right) = \beta^{-m}(0.1)_\beta.$$

CARDINALITÀ DELL'INSIEME DEI NUMERI DI MACCHINA Dati $\beta,\ t,\ m,\ M$ calcoliamo $\#\mathcal{F}(\beta, t, m, M)$. Osserviamo che

1. dalla definizione di $\mathcal{F}(\beta, t, m, M)$, dobbiamo contare lo 0 separatamente;
2. per quanto riguarda i numeri non-nulli, per ogni possibile combinazione di esponente/-mantissa abbiamo 2 scelte per il segno (segno positivo o negativo);
3. le scelte per i possibili esponenti corrispondono a tutti i numeri interi nell'intervallo $[-m, M]$, che sono $M + m + 1$;
4. per quanto riguarda la mantissa dobbiamo scegliere t cifre comprese tra 0 e $\beta - 1$ (e lo possiamo fare in β^t modi), ma dobbiamo escludere tutte quelle che iniziano per 0, che sono β^{t-1}: abbiamo quindi $\beta^t - \beta^{t-1}$ scelte per la mantissa di un numero.

In totale si ha quindi che

$$\#\mathcal{F}(\beta, t, m, M) = 1 + 2(M + m + 1)\left(\beta^t - \beta^{t-1}\right).$$

Esempio 1.1.7. Ad esempio se $\beta = 2$, $t = 3$ e $m = M = 1$ si ha che

$$\#\mathcal{F}(2, 3, 1, 1) = 1 + 2(1 + 1 + 1)\left(2^3 - 2^2\right) = 25.$$

1.1.4 Standard IEEE (754)

Lo standard usato al giorno d'oggi per implementare i numeri a virgola mobile è lo standard IEEE (754). Esso definisce due rappresentazioni distinte, una *single precision*, dove ogni numero è rappresentato su 32 bit, ovvero su 4 byte, e una *double precision*, dove ogni numero è rappresentato su 64 bit (o equivalentemente 8 byte). In particolare noi analizzeremo solo il secondo.

I 64 bit vengono divisi nel seguente modo:

- un singolo bit di segno (0 per i numeri positivi, 1 per i negativi);
- 11 bit per rappresentare l'esponente;
- i restanti 52 bit vengono usati per rappresentare la mantissa.

Tuttavia possiamo ottimizzare un po' lo spazio usato facendo delle semplici osservazioni:

1. siccome siamo in base 2 la condizione $d_1 \neq 0$ ci dice che necessariamente $d_1 = 1$: ciò significa che rappresentare d_1 nella mantissa è inutile (ogni mantissa inizierebbe per 1) e quindi possiamo rappresentare le cifre da d_2 in poi. Questo significa che abbiamo in realtà $t = 53$ cifre decimali a disposizione;

2. per evitare di usare un bit per il segno dell'esponente lo rappresentiamo in *traslazione* (anche detto *bias* o *eccesso a*): invece di rappresentare il vero esponente p rappresentiamo p + 1022. Questo ci consente di indicare esponenti negativi senza dover usare strategie di complemento a 2.

L'insieme dei numeri di macchina definito dallo standard IEEE (754) è quindi

$$\mathcal{F}_{\text{IEEE}} \coloneqq \mathcal{F}(2, 53, 1021, 1024).$$

Osserviamo che questa scelta per i limiti degli esponenti non copre tutte le combinazioni possibili su 11 bit: in particolare possiamo rappresentare $1024 + 1021 + 1 = 2046$ esponenti diversi, mentre su 11 bit potenzialmente potremmo rappresentarne fino a $2^{11} = 2048$.

Questo ci permette di rappresentare dei numeri particolari.

ZERO Abbiamo visto che (per il Theorem 1.1.1) lo 0 non ammette una rappresentazione normalizzata. Nello standard IEEE (754) esso viene realizzato ponendo a 0 tutti i campi dell'esponente e della mantissa; tuttavia non viene specificato il segno, per cui esistono due rappresentazioni uguali per lo zero (-0 e $+0$).

Questa rappresentazione non va in conflitto con i numeri normalizzati, in quanto ponendo il campo dell'esponente uguale a undici 0 otterrei un esponente effettivo uguale a -1022, mentre il minimo rappresentabile è -1021.

NUMERI DENORMALIZZATI Se il campo dell'esponente contiene solo zeri ma la mantissa ha almeno un bit diverso da 0 stiamo rappresentando numeri *denormalizzati*. Un tale numero viene interpretato come se fosse $d_1 = 0$ e quindi ci consente di ottenere dei numeri in valore assoluto più piccoli di ω, risolvendo parzialmente i problemi di *underflow*.

Tuttavia questi numeri hanno una precisione minore dei numeri normalizzati, in quanto i primi k bit della mantissa possono essere tutti uguali a 0. Questo significa che invece di avere 53 cifre di precisione ne abbiamo $53 - k$ (dove k dipende dal numero denormalizzato scelto), dunque dobbiamo porre ancora più attenzione agli eventuali errori di precisione.

Il più grande numero denormalizzato è della forma

$$2^{-1021}(0.01\ldots1)_2 = 2^{-1022}(0\overbrace{1\ldots1}^{52\text{ volte}})_2,$$

mentre il più piccolo numero denormalizzato è il numero

$$2^{-1021}(0.0\ldots01)_2.$$

INFINITI E NAN Ponendo il campo dell'esponente uguale a $(11111111111)_2$ (ovvero undici cifre 1) ottengo una rappresentazione che non appartiene all'insieme definito prima (in quanto il massimo esponente rappresentabile è 1024, che in eccesso a 1022 diventa $2046 = (11111111110)_2$). Sfruttando questo fatto possiamo rappresentare tre "numeri" diversi:

- se il campo della mantissa non è formato da tutti 0 il risultato è NaN (*Not a Number*): questa configurazione viene usata ad esempio per i risultati delle forme indeterminate, come $0/0$;
- se il campo della mantissa è formato da tutti 0 il numero viene interpretato come $\pm\infty$ (a seconda del bit di segno): ciò viene usato per rappresentare l'overflow, ovvero numeri che sono più grandi di Ω (o più piccoli di $-\Omega$).

1.2 ERRORI DI APPROSSIMAZIONE

Abbiamo iniziato ad introdurre il concetto di approssimazione nella Section 1.1. Definiamolo ora formalmente.

Definizione 1.2.1 – Troncamento e arrotondamento

Sia $x \in \mathbb{R}$, $x \neq 0$, tale che $\omega < |x| < \Omega$. Consideriamo inoltre la rappresentazione normalizzata di x:

$$x = \mathrm{sgn}(x)\beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}.$$

Allora definiamo

(1) il **troncamento** di x alla t-esima cifra, indicato con $\mathrm{trun}(x)$, dato da

$$\mathrm{trun}(x) := \mathrm{sgn}(x)\beta^p \sum_{i=1}^{t} d_i \beta^{-i};$$

(2) l'**arrotondamento** di x, indicato con $\mathrm{arr}(x)$, dato da

$$\mathrm{arr}(x) := \begin{cases} \mathrm{trun}(x), & \text{se } 0 \leq d_{t+1} < \beta/2 \\ \mathrm{trun}(x) + \beta^{p-t}, & \text{se } d_{t+1} \geq \beta/2. \end{cases}$$

Osservazione 1.2.1. Aggiungere β^{p-t} ad un numero significa aggiungere 1 alla sua t-esima cifra. Infatti (considerando lo stesso x della definizione) si ha che

$$\beta^p \sum_{i=1}^{\infty} d_i \beta^{-i} + \beta^{p-t} = \beta^p \left(\sum_{i=1}^{\infty} d_i \beta^{-i} + \beta^{-t} \right).$$

Esempio 1.2.2. Consideriamo il numero reale

$$\pi = 3.141592653\cdots = 10^1 \cdot (0.3141592653\ldots)_{10}.$$

Il suo troncamento alla quinta cifra sarà 3.1415, mentre il suo arrotondamento alla stessa cifrà (siccome la sesta cifra è $9 \geq 10/2$) sarà

$$3.1415 + 10^{1-5} = 3.1415 + 0.0001 = 3.1416.$$

D'ora in avanti quando parleremo dell'approssimazione di un numero reale x senza voler necessariamente specificare il metodo (troncamento o arrotondamento) usato scriveremo $\mathrm{fl}(x)$ oppure $\widetilde{x}$.

Approssimando numeri reali a numeri di macchina si genera un errore, detto **errore di rappresentazione**.

Definizione 1.2.3 – Errore di rappresentazione

Sia $x \in \mathbb{R}$, $x \neq 0$ tale che $\omega \leq |x| \leq \Omega$. Chiameremo **errore assoluto** la quantità

$$\eta_x := \widetilde{x} - x,$$

mentre chiameremo **errore relativo** o **errore di rappresentazione** la quantità

$$\varepsilon_x := \frac{\widetilde{x} - x}{x}.$$

La qualità più importante dell'errore relativo è il fatto che è sempre **superiormente limitato**, come dimostreremo attraverso un lemma ed un teorema.

Lemma 1.2.4 – Limite superiore all'errore assoluto

Sia $x \in \mathbb{R}$, $x \neq 0$, $\omega \leq x \leq \Omega$. Sia inoltre $x = \mathrm{sgn}(x)\beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}$ la rappresentazione in base di x. Si ha che

(i) $\left|\text{trun}(x) - x\right| < \beta^{p-t}$

(ii) $\left|\text{arr}(x) - x\right| \leq \frac{1}{2}\beta^{p-t}$.

Dimostrazione. Se fosse $x = \Omega$ oppure $x = \omega$ allora $\text{trun}(x) = \text{arr}(x) = x$, da cui l'errore assoluto $\widetilde{x} - x$ è nullo.

Supponiamo quindi $\omega < x < \Omega$ (il caso $-\Omega < x < -\omega$ è speculare). Siccome valgono le ipotesi del Theorem 1.1.1 possiamo scrivere

$$x = \beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}.$$

Esisteranno dunque due numeri di macchina $a, b \in \mathcal{F}$ tali che $a \leq x < b$ e b è il numero di macchina successivo rispetto ad a, ovvero a è il numero di macchina appena più piccolo di x e b è quello appena più grande.

Allora dovrà essere necessariamente

$$a = \beta^p \sum_{i=1}^{t} d_i \beta^{-i} = \text{trun}(x),$$

mentre b dovrà essere

$$b = \beta^p \left(\sum_{i=1}^{t} d_i \beta^{-i} + \beta^{-t} \right) = a + \beta^{p-t},$$

da cui segue anche che $b - a = \beta^{p-t}$.

Si ha quindi

$$\left|\text{trun}(x) - x\right| = |a - x| < |a - b| = \beta^{p-t},$$

dove il minore centrale deriva dal fatto che $x < b$.

Per quanto riguarda l'arrotondamento, osserviamo invece che

- se $d_{t+1} < \beta/2$ (ovvero se $x < 1/2(a + b)$) allora $\text{arr}(x) = \text{trun}(x) = a$;
- altrimenti $\text{arr}(x) = \text{trun}(x) + \beta^{p-t} = a + \beta^{p-t} = b$.

Nel primo caso si ha

$$\left|\text{arr}(x) - x\right| = \left|x - a\right| \leq \left|\frac{a+b}{2} - a\right| = \frac{1}{2}|b - a| = \frac{1}{2}\beta^{p-t},$$

mentre nel secondo si ha

$$\left|\text{arr}(x) - x\right| = \left|b - x\right| \leq \left|b - \frac{a+b}{2}\right| = \frac{1}{2}|b - a| = \frac{1}{2}\beta^{p-t},$$

da cui la tesi. □

Osservazione 1.2.2. Si ha l'uguaglianza $\left|\text{arr}\, x - x\right| = \frac{1}{2}\beta^{p-t}$ se e solo se $x = \frac{1}{2}(a + b)$, ovvero se e solo se x è esattamente il punto medio dell'intervallo $[a, b]$.

Teorema 1.2.5 – Esistenza della precisione di macchina

Esiste un numero $u \in \mathbb{R}$, detto **precisione di macchina**, tale che per ogni $x \in \mathbb{R}$, $x \neq 0$, $\omega \leq |x| \leq \Omega$ vale che

$$|\varepsilon_x| \leq u.$$

In particolare

(i) se $\mathrm{fl}(x) = \mathrm{trun}(x)$ si ha $u = \beta^{1-t}$;

(ii) se $\mathrm{fl}(x) = \mathrm{arr}(x)$ si ha $u = \frac{1}{2}\beta^{1-t}$.

Dimostrazione. Osserviamo che se $x = \pm\beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}$ allora varrà sicuramente che

$$|x| \geq \beta^p (0.10\ldots)_\beta = \beta^p \cdot (1\beta^{-1}) = \beta^{p-1}.$$

Infatti scegliendo il numero di macchina con lo stesso esponente di x ma la minima mantissa possibile otterremo necessariamente un numero più piccolo (o uguale) del valore assoluto di quello di partenza.

Consideriamo allora i due casi.

Troncamento Per il Lemma 1.2.4 sappiamo che $\left|\mathrm{trun}(x) - x\right| < \beta^{p-t}$. Si ha quindi che

$$|\varepsilon_x| = \frac{\left|\mathrm{trun}(x) - x\right|}{|x|} < \frac{\beta^{p-t}}{\beta^{p-1}} = \beta^{1-p}.$$

Arrotondamento Per il Lemma 1.2.4 sappiamo che $\left|\mathrm{arr}(x) - x\right| \leq \frac{1}{2}\beta^{p-t}$. Si ha quindi che

$$|\varepsilon_x| = \frac{\left|\mathrm{arr}(x) - x\right|}{|x|} \leq \frac{1/2\beta^{p-t}}{\beta^{p-1}} = \frac{1}{2}\beta^{1-p}. \qquad \square$$

Osserviamo che il Theorem 1.2.5 ci dice che u è un limite superiore *debole* (nel senso che $|\varepsilon_x| \leq u$ e non $|\varepsilon_x| < u$), tuttavia almeno nel caso del troncamento si vede dalla dimostrazione che vale anche la disuguaglianza stretta.

In realtà la disuguaglianza stretta vale anche nel caso dell'arrotondamento: la dimostrazione completa è lasciata all'appendice (in particolare nella Appendix A.2).

1.3 ARITMETICA DI MACCHINA

Cerchiamo ora di estendere l'aritmetica tra numeri reali ad un'aritmetica per i numeri di macchina. Il primo tentativo è quello di usare semplicemente le solite operazioni definite tra numeri reali, tuttavia non è detto che il risultato di un'operazione tra numeri di macchina sia ancora un numero di macchina.

Esempio 1.3.1. Consideriamo il sistema $\mathcal{F}(10, 3, -5, 5)$ e i due numeri di macchina

$$a := 123 = 10^3(0.123)_{10}, \qquad b := 0.456 = 10^0(0.456).$$

Si ha che $a,\ b \in \mathcal{F}(10, 3, -5, 5)$, tuttavia $a + b = 123.456 = 10^3(0.123456)_{10}$ non è un numero di macchina poiché è rappresentato con $t = 6$ cifre di precisione.

Dobbiamo quindi definire delle nuove operazioni che siano *interne* all'insieme dei numeri di macchina.

Definizione 1.3.2 – Aritmetica di macchina

Si definiscono quattro operazioni

$$\oplus,\ \otimes,\ \ominus,\ \oslash : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$$

tali che per ogni coppia di numeri di macchina $a, b \in \mathcal{F}$ ($b \neq 0$ nel caso di $\oslash$) si ha che

$$a \oplus b := \mathrm{fl}(a + b) \qquad a \ominus b := \mathrm{fl}(a - b)$$
$$a \otimes b := \mathrm{fl}(ab) \qquad a \oslash b := \mathrm{fl}(a/b)$$

Siamo ora interessati a studiare *quanto errore* commettiamo nel considerare i numeri di macchina invece dei numeri reali non approssimati.

Definizione 1.3.3 – Errore locale

Sia $*$ un'operazione tra numeri qualsiasi e sia $\circledast$ la corrispondente operazione su numeri di macchina. Dati $x, y \in \mathcal{F}$ si dice **errore locale** dell'operazione la quantità

$$\varepsilon := \frac{(x \circledast y) - (x * y)}{x * y}.$$

Osserviamo che ogni operazione di macchina può essere scritta in termini della sua corrispondente operazione base e dell'errore locale: in effetti

$$x \circledast y = (x * y)(1 + \varepsilon).$$

Esempio 1.3.4 (Errore nel calcolo della somma). Cerchiamo ora di calcolare l'errore totale commesso nell'approssimazione di un'operazione tra numeri reali (definiremo più avanti precisamente il concetto di errore totale). Prendiamo $a, b \in \mathbb{R}$ che approssimati non vadano in overflow (o underflow) e cerchiamo di calcolare $a + b$ in termini di numeri di macchina.

Siccome i numeri a, b non sono necessariamente già numeri di macchina, l'unico modo per svolgere il calcolo è calcolare la quantità

$$\widetilde{a} \oplus \widetilde{b} = (\widetilde{a} + \widetilde{b})(1 + \varepsilon).$$

Siccome sappiamo che, se ε_a, ε_b sono gli errori di rappresentazione rispettivamente di a e b

$$\widetilde{a} = a(1 + \varepsilon_a), \quad \widetilde{b} = b(1 + \varepsilon_b)$$

otteniamo che

$$\begin{aligned} \widetilde{a} \oplus \widetilde{b} &= (a(1 + \varepsilon_a) + b(1 + \varepsilon_b))(1 + \varepsilon) \\ &= a(1 + \varepsilon_a)(1 + \varepsilon) + b(1 + \varepsilon_b)(1 + \varepsilon). \end{aligned}$$

Nello svolgere il prossimo passaggio trascureremo gli addendi in cui compare il prodotto di due errori: effettueremo quindi un'**analisi al primo ordine**, indicata dal simbolo $\doteq$.

$$\doteq a + b + a\varepsilon_a + b\varepsilon_b + (a + b)\varepsilon.$$

L'errore complessivo che commettiamo è quindi dato dalla differenza tra il risultato ottenuto $(\widetilde{a} \oplus \widetilde{b})$ e il risultato non approssimato $(a + b)$, dividendo per il risultato non approssimato in modo da ottenere un errore relativo:

$$\begin{aligned} \frac{(\widetilde{a} \oplus \widetilde{b}) - (a + b)}{a + b} &\doteq \frac{a + b + a\varepsilon_a + b\varepsilon_b + (a + b)\varepsilon - (a + b)}{a + b} \\ &= \frac{a}{a + b}\varepsilon_a + \frac{b}{a + b}\varepsilon_b + \varepsilon. \end{aligned}$$

Notiamo che questo errore può essere diviso in due parti diverse:

- la parte data da $\frac{a}{a+b}\varepsilon_a + \frac{b}{a+b}\varepsilon_b$ non dipende dal tipo di operazione che sto cercando di svolgere, ma dal fatto che devo approssimare gli argomenti dell'operazione: lo chiameremo **errore inerente**;
- la parte data da ε dipende dalla sequenza di passi scelta per calcolare la somma: lo chiameremo perciò **errore algoritmico**.

Esempio 1.3.5 (Errore di una funzione). Consideriamo la funzione $f : \mathbb{R} \to \mathbb{R}$ definita da

$$f(x) = x^2 - 1.$$

Calcoliamo l'errore totale commesso nel calcolare $f(x)$ tramite operazioni di macchina.

Osserviamo che potenzialmente abbiamo più *algoritmi* (intesi come sequenze di calcoli) per calcolare f: scegliamo ad esempio gli algoritmi di macchina g_1, g_2 definiti da

$$g_1(\widetilde{x}) := (\widetilde{x} \otimes \widetilde{x}) \ominus 1, \qquad g_2(\widetilde{x}) := (\widetilde{x} \ominus 1) \otimes (\widetilde{x} \oplus 1).$$

L'algoritmo g_1 prescrive di calcolare innanzitutto $z := \widetilde{x} \otimes \widetilde{x}$ e dopo calcolare $z \ominus 1$. Se ε_1 è l'errore locale dell'operazione $\otimes$ si ha che

$$z = \widetilde{x}^2(1 + \varepsilon_1);$$

chiamando ε_2 l'errore locale dell'operazione $\ominus$ si ottiene

$$g_1(\widetilde{x}) = (z-1)(1+\varepsilon_2).$$

A questo punto, ricordando che $\widetilde{x} = x(1+\varepsilon_x)$ dove ε_x è l'errore di rappresentazione, si può esprimere $g_1(\widetilde{x})$ in termini di x e dei vari errori. In effetti

$$\begin{aligned}
&(z-1)(1+\varepsilon_2)\\
&= \big(\widetilde{x}^2(1+\varepsilon_1)-1\big)(1+\varepsilon_2)\\
&= \Big(\big(x(1+\varepsilon_x)\big)^2(1+\varepsilon_1)-1\Big)(1+\varepsilon_2)\\
&= \big(x^2(1+\varepsilon_x)^2(1+\varepsilon_1)-1\big)(1+\varepsilon_2)\\
&\doteq \big(x^2(1+2\varepsilon_x)(1+\varepsilon_1)-1\big)(1+\varepsilon_2)\\
&\doteq \big(x^2(1+2\varepsilon_x+\varepsilon_1)-1\big)(1+\varepsilon_2)\\
&= x^2(1+2\varepsilon_x+\varepsilon_1)(1+\varepsilon_2)-(1+\varepsilon_2)\\
&\doteq x^2(1+2\varepsilon_x+\varepsilon_1+\varepsilon_2)-(1+\varepsilon_2)\\
&= x^2-1+2x^2\varepsilon_x+x^2\varepsilon_1+(x^2-1)\varepsilon_2.
\end{aligned}$$

L'errore totale commesso nell'approssimare il risultato della funzione $f(x)$ con la funzione di macchina $g_1(\widetilde{x})$ è dunque

$$\begin{aligned}
\varepsilon_{g_1} &\coloneqq \frac{x^2-1+2x^2\varepsilon_x+x^2\varepsilon_1+(x^2-1)\varepsilon_2-(x^2-1)}{x^2-1}\\
&= \frac{2x^2}{x^2-1}\varepsilon_x+\frac{x^2}{x^2-1}\varepsilon_1+\varepsilon_2.
\end{aligned}$$

Ripetiamo il procedimento per l'algoritmo g_2: siano δ_1, δ_2 e δ_3 i tre errori locali commessi; ovvero

$$\begin{aligned}
z_1 &\coloneqq \widetilde{x}\ominus 1 = (\widetilde{x}-1)(1+\delta_1)\\
z_2 &\coloneqq \widetilde{x}\oplus 1 = (\widetilde{x}+1)(1+\delta_2)\\
g_2(\widetilde{x}) &= z_1\otimes z_2 = z_1z_2(1+\delta_3).
\end{aligned}$$

Si ha quindi

$$\begin{aligned}
&z_1z_2(1+\delta_3)\\
&= (\widetilde{x}-1)(1+\delta_1)(\widetilde{x}+1)(1+\delta_2)(1+\delta_3)\\
&= (\widetilde{x}^2-1)(1+\delta_1)(1+\delta_2)(1+\delta_3)\\
&= \big(x^2(1+\varepsilon_x)^2-1\big)(1+\delta_1)(1+\delta_2)(1+\delta_3)\\
&\doteq \big(x^2(1+2\varepsilon_x)-1\big)(1+\delta_1)(1+\delta_2)(1+\delta_3)\\
&= \big(x^2(1+2\varepsilon_x)(1+\delta_1)-(1+\delta_1)\big)(1+\delta_2)(1+\delta_3)\\
&\doteq \big(x^2(1+2\varepsilon_x+\delta_1)-(1+\delta_1)\big)(1+\delta_2)(1+\delta_3)\\
&= \big(x^2(1+2\varepsilon_x+\delta_1)(1+\delta_2)-(1+\delta_1)(1+\delta_2)\big)(1+\delta_3)\\
&\doteq \big(x^2(1+2\varepsilon_x+\delta_1+\delta_2)-(1+\delta_1+\delta_2)\big)(1+\delta_3)\\
&= x^2(1+2\varepsilon_x+\delta_1+\delta_2)(1+\delta_3)-(1+\delta_1+\delta_2)(1+\delta_3)\\
&\doteq x^2(1+2\varepsilon_x+\delta_1+\delta_2+\delta_3)-(1+\delta_1+\delta_2+\delta_3)\\
&= x^2+2x^2\varepsilon_x+x^2(\delta_1+\delta_2+\delta_3)-1-(\delta_1+\delta_2+\delta_3)\\
&= x^2-1+2x^2\varepsilon_x+(x^2-1)(\delta_1+\delta_2+\delta_3).
\end{aligned}$$

L'errore totale generato dalla funzione g_2 è dunque

$$\begin{aligned}
\varepsilon_{g_2} &\coloneqq \frac{x^2-1+2x^2\varepsilon_x+(x^2-1)(\delta_1+\delta_2+\delta_3)-(x^2-1)}{x^2-1}\\
&= \frac{2x^2}{x^2-1}\varepsilon_x+\delta_1+\delta_2+\delta_3.
\end{aligned}$$

Osserviamo che entrambe le funzioni hanno lo stesso errore inerente (dato dal termine contenente ε_x): in effetti l'errore inerente deriva dall'approssimazione di x in $\widetilde{x}$, e ciò non dipende dall'algoritmo scelto.

Il resto del termine è l'errore algoritmico: siccome abbiamo due algoritmi diversi possiamo chiederci quale dei due sia migliore.

Osserviamo che il secondo non dipende da x, mentre il primo sì: in particolare l'errore algoritmico del primo tende ad infinito quando x tende a ± 1. Si dice quindi che l'algoritmo non è **stabile** per $x \to \pm 1$, mentre il secondo algoritmo è stabile per ogni $x \in \mathbb{R}$.

1.4 TEORIA DEGLI ERRORI

Diamo ora una definizione precisa dei vari tipi di errori.

Consideriamo nel resto della sezione

- una funzione razionale $f : A \to \mathbb{R}$ con $A \subseteq \mathbb{R}$, dove per *razionale* si intende che compaiono solo le operazioni di somma, prodotto, sottrazione e divisione
- un numero reale x qualunque e la sua approssimazione $\widetilde{x}$
- un algoritmo $g : \mathcal{F} \to \mathcal{F}$ per calcolare f, dove per *algoritmo* si intende una sequenza di operazioni tra numeri di macchina usata per calcolare il valore di f.

1.4.1 Errore inerente

Definizione 1.4.1 – Errore inerente

Si dice **errore inerente** del problema la quantità

$$\varepsilon_{in} := \frac{f(\widetilde{x}) - f(x)}{f(x)}.$$

Osserviamo che l'errore inerente misura quanto è grande l'errore nell'approssimare x con il numero di macchina $\widetilde{x}$: se l'errore inerente è grande in un intorno di x nessun algoritmo ci consentirà di calcolare con una buona precisione il risultato della funzione f. In tal caso diremo che il problema è **mal condizionato**, mentre se l'errore inerente è sempre limitato il problema verrà definito **ben condizionato**.

Per studiare il condizionamento di un problema è quindi necessario studiare il suo errore inerente: vogliamo quindi una tecnica per calcolarlo più efficientemente.

Osserviamo che moltiplicando e dividendo per x e per $\widetilde{x} - x$ si ha

$$\varepsilon_{in} = \frac{f(\widetilde{x}) - f(x)}{f(x)} = \frac{f(\widetilde{x}) - f(x)}{\widetilde{x} - x} \cdot \frac{x}{f(x)} \cdot \underbrace{\frac{\widetilde{x} - x}{x}}_{\varepsilon_x}.$$

Il primo termine del prodotto ricorda un rapporto incrementale: se la funzione è derivabile in x possiamo semplificare l'espressione usando il Teorema di Taylor.

Supponiamo quindi che f sia continua e derivabile *due volte* in un intervallo $[a, b]$ contenente x (ovvero $f \in C^2[a, b]$): per il Teorema di Taylor con il resto di Lagrange si ha che esiste un numero ξ_x compreso tra x e $\widetilde{x}$ tale che

$$f(\widetilde{x}) = f(x) + f'(x)(\widetilde{x} - x) + f''(\xi_x)\frac{(\widetilde{x} - x)^2}{2!}$$

con $|\varepsilon_x - x| < |\widetilde{x} - x|$.

Sostituendo otteniamo

$$\begin{aligned}
\varepsilon_{in} &= \frac{f(\widetilde{x}) - f(x)}{\widetilde{x} - x} \cdot \frac{x}{f(x)} \cdot \varepsilon_x \\
&= \frac{f'(x)(\widetilde{x} - x) + f''(\xi_x)\dfrac{(\widetilde{x} - x)^2}{2!}}{\widetilde{x} - x} \cdot \frac{x}{f(x)} \cdot \varepsilon_x \\
&= \left(f'(x) + f''(\xi_x)\frac{\widetilde{x} - x}{2!} \right) \cdot \frac{x}{f(x)} \cdot \varepsilon_x \\
&= f'(x)\frac{x}{f(x)}\varepsilon_x + f''(\xi_x)\frac{\widetilde{x} - x}{2!}\frac{x}{f(x)} \cdot \varepsilon_x.
\end{aligned}$$

Osserviamo ora che il secondo addendo contiene (nascosto) un errore al quadrato, e pertanto in un analisi al prim'ordine viene approssimato a 0. In effetti moltiplicando e dividendo per x si ha

$$\begin{aligned}
& f''(\xi_x)\frac{\widetilde{x} - x}{2!}\frac{x}{f(x)} \cdot \varepsilon_x \\
&= f''(\xi_x)\frac{\widetilde{x} - x}{2!}\frac{x}{f(x)} \cdot \frac{x}{x} \cdot \varepsilon_x \\
&= f''(\xi_x)\frac{x^2}{2!f(x)} \cdot \frac{\widetilde{x} - x}{x} \cdot \varepsilon_x \\
&= f''(\xi_x)\frac{x^2}{2!f(x)} \cdot \varepsilon_x^2 \\
&\doteq 0.
\end{aligned}$$

Abbiamo quindi dimostrato il seguente Teorema.

Teorema 1.4.2 – Caratterizzazione dell'errore inerente

Sia $f : A \subseteq \mathbb{R} \to \mathbb{R}$ e sia x il punto di cui vogliamo calcolare l'immagine $f(x)$. Se $f \in C^2(A)$ si ha che

$$\varepsilon_{in} \doteq f'(x)\frac{x}{f(x)} \cdot \varepsilon_x,$$

dove ε_x è l'errore di rappresentazione di x.
La quantità $c_x := f'(x)\frac{x}{f(x)}$ viene detta **coefficiente di amplificazione**.

Abbiamo quindi scoperto che l'errore inerente è sempre della forma $c_x\varepsilon_x$: possiamo quindi definire formalmente il concetto di condizionamento.

Definizione 1.4.3 – Condizionamento di un problema

Sia $f : A \subseteq \mathbb{R} \to \mathbb{R}$ con $f \in C^2(A)$. Se esiste $x_0 \in A$ tale che

$$\lim_{x \to x_0} |c_x| = +\infty$$

si dice che il problema è **mal condizionato** in un intorno di x_0, altrimenti si dice che il problema è **ben condizionato**.

Esempio 1.4.4. Riprendiamo l'Example 1.3.5: il coefficiente di amplificazione è

$$c_x = \frac{2x^2}{x^2 - 1}$$

il cui valore assoluto tende a $+\infty$ per $x \to \pm 1$. Si ha quindi che il problema è mal condizionato in un intorno di 1 e in un intorno di -1.

Il Theorem 1.4.2 può essere generalizzato per funzioni $\mathbb{R}^n \to \mathbb{R}$.

Corollario 1.4.5 – Errore inerente su funzioni multivariate

Sia $f : \Omega \to \mathbb{R}$ con $\Omega \subseteq \mathbb{R}^n$ aperto e f differenziabile due volte su Ω (ovvero esistono la derivata parziale prima e seconda rispetto ad ogni variabile).
Sia inoltre $\mathbf{x} = (x_1, \dots, x_n)$ il punto di cui vogliamo calcolare $f(\mathbf{x})$. Si ha che

$$\varepsilon_{in} = \sum_{i=1}^{n} c_{x_i} \varepsilon_{x_i}$$

dove $c_{x_i} := \dfrac{x_i}{f(x)} \cdot \dfrac{\partial f}{\partial x_i}$ è il **coefficiente di amplificazione**.

Coefficienti di amplificazione delle operazioni elementari

Il Corollary 1.4.5 può essere molto comodo per calcolare i coefficienti di amplificazione delle operazioni elementari: ciò torna molto utile per il calcolo dell'errore totale.

SOMMA Consideriamo $f(x, y) = x + y$. Si ha che

$$c_x = \frac{x}{x+y} \cdot 1 = \frac{x}{x+y}, \qquad c_y = \frac{y}{x+y} \cdot 1 = \frac{y}{x+y}.$$

SOTTRAZIONE Consideriamo $f(x, y) = x - y$. Si ha che

$$c_x = \frac{x}{x-y} \cdot 1 = \frac{x}{x-y}, \qquad c_y = \frac{y}{x+y} \cdot (-1) = -\frac{y}{x-y}.$$

PRODOTTO Consideriamo $f(x, y) = xy$. Si ha che

$$c_x = \frac{x}{xy} \cdot y = 1, \qquad c_y = \frac{y}{xy} \cdot x = 1.$$

DIVISIONE Consideriamo $f(x, y) = \frac{x}{y}$. Si ha che

$$c_x = \frac{x}{x/y} \cdot \frac{1}{y} = 1, \qquad c_y = \frac{y}{x/y} \cdot \left(-\frac{x}{y^2}\right) = -1.$$

1.4.2 Errore algoritmico

Definizione 1.4.6 – Errore algoritmico

Si dice **errore algoritmico** la quantità

$$\varepsilon_{alg} := \frac{g(\tilde{x}) - f(\tilde{x})}{f(\tilde{x})}.$$

L'errore algoritmico misura quanto è grande calcolare il risultato attraverso l'algoritmo di macchina g invece della funzione non approssimata f *partendo da dati già approssimati*.

Per studiare l'errore algoritmico si usa la tecnica conosciuta come *errore in avanti* (*forward error* in inglese), che può essere resa più compatta attraverso l'uso di grafi.

Formalmente un algoritmo è una sequenza di passi *finita* $z^{(1)}, z^{(2)}, z^{(3)}, \dots, z^{(n)}$, tale che ognuno dei passi calcola una singola operazione sfruttando solo i dati dei passi precedenti e i dati del problema (costanti e variabili) e l'ultimo passo calcola la funzione del problema.

Esempio 1.4.7. L'algoritmo $g(x) = ((x \otimes x) \oplus x) \ominus 3$ può essere espresso dalla sequenza di passi

1. $z^{(1)} = x \otimes x$
2. $z^{(2)} = z_1 \oplus x$
3. $z^{(3)} = z^{(2)} \ominus 3$

Ogni passo che calcola un'operazione può quindi essere scritto nella forma

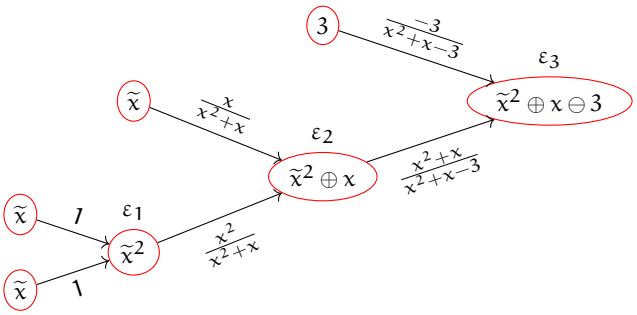
$$z^{(i)} = z^{(h)} \circledast z^{(k)}$$

con $h, k < i$ e $\circledast$ un'operazione qualunque di macchina.

Esempio 1.4.8. Calcoliamo l'errore algoritmico della funzione $f(x) = x^2 + x - 3$ attraverso l'algoritmo di macchina

$$g(\widetilde{x}) = \left(\left(\widetilde{x} \otimes \widetilde{x}\right) \oplus \widetilde{x}\right) \ominus 3.$$

Per far ciò, costruiamo il seguente grafo.



Il grafo è costruito nel seguente modo:

- si considera la sequenza di operazioni da svolgere: in questo caso
  1. $z_1 \coloneqq \widetilde{x} \otimes \widetilde{x}$
  2. $z_2 \coloneqq z_1 \oplus \widetilde{x}$
  3. $z_3 \coloneqq z_2 \ominus 3 = g(\widetilde{x})$
- si disegnano i nodi iniziali, ovvero quelli della prima operazione da svolgere, e sopra di essi si mettono gli errori commessi nel rappresentarli: siccome siamo interessati all'errore algoritmico partiamo già con numeri di macchina, quindi l'errore di rappresentazione è 0 e lo omettiamo;
- si crea un nodo corrispondente a z_1: l'errore commesso nella creazione di questo nodo è l'errore locale del prodotto (qui chiamato ε_1);
- sopra gli archi che portano i nodi $\widetilde{x}$ in z_1 indichiamo il coefficiente di amplificazione c_x dei vari nodi: in questo caso sono entrambi uguali a 1 poiché l'operazione è un prodotto;
- ripetiamo il procedimento: creiamo un nuovo nodo con $\widetilde{x}$ e "errore di creazione" uguale a 0 e lo usiamo per fare l'operazione di somma, creando un nuovo nodo (che corrisponde a z_2) con errore di creazione uguale all'errore locale della somma ε_2;
- per ricavare i coefficienti di amplificazione sfruttiamo le formule ricavate prima, che ci dicono che nella somma $\widetilde{x}^2 \oplus \widetilde{x}$ i coefficienti di amplificazione sono dati da

$$c_{x^2} = \frac{x^2}{x^2 + x}, \qquad c_x = \frac{x}{x^2 + x};$$

- ripetiamo un'ultima volta il ragionamento per quanto riguarda l'ultima operazione.

Possiamo ora calcolare l'errore totale a partire dal grafo: si parte dalla fine del grafo (nodo più a destra) e si calcola la somma tra l'errore del nodo (in questo caso ε_3) e gli errori dei sottoalberi (calcolati ricorsivamente), moltiplicati per il loro coefficiente di amplificazione.

Otteniamo quindi

$$\begin{aligned}
\varepsilon_{alg} &= \varepsilon_3 + \frac{x^2+x}{x^2+x-3}(\dots) + \frac{-3}{x^2+x-3}\cdot 0 \\
&= \varepsilon_3 + \frac{x^2+x}{x^2+x-3}\left(\varepsilon_2 + \frac{x^2}{x^2+x}(\dots) + \frac{x}{x^2+x}\cdot 0\right) \\
&= \varepsilon_3 + \frac{x^2+x}{x^2+x-3}\left(\varepsilon_2 + \frac{x^2}{x^2+x}(\varepsilon_1 + 1\cdot 0 + 1\cdot 0)\right) \\
&= \varepsilon_3 + \frac{x^2+x}{x^2+x-3}\left(\varepsilon_2 + \frac{x^2}{x^2+x}\cdot \varepsilon_1\right).
\end{aligned}$$

1.4.3 Errore totale

Definizione 1.4.9 – Errore totale

Si dice **errore totale** del problema la quantità

$$\varepsilon_{tot} := \frac{g(\widetilde{x}) - f(x)}{f(x)}.$$

Osserviamo chel'errore totale combina i concetti di errore inerente e algoritmico e ci dice quanto è grande l'errore nel calcolare g con il valore di $\widetilde{x}$ rispetto al calcolo esatto di $f(x)$.

Nell'Example 1.3.5 abbiamo notato come l'errore totale fosse dato dalla somma dell'errore algoritmico e dell'errore inerente: questa relazione vale sempre, come dimostrato dal prossimo teorema.

Teorema 1.4.10

$\varepsilon_{tot} \doteq \varepsilon_{in} + \varepsilon_{alg}$.

Dimostrazione. In effetti si ha

$$\begin{aligned}
\varepsilon_{tot} &= \frac{g(\widetilde{x}) - f(x)}{f(x)} \\
&= \frac{g(\widetilde{x}) - f(\widetilde{x}) + f(\widetilde{x}) - f(x)}{f(x)} \\
&= \frac{g(\widetilde{x}) - f(\widetilde{x})}{f(x)} + \frac{f(\widetilde{x}) - f(x)}{f(x)} \\
&= \frac{g(\widetilde{x}) - f(\widetilde{x})}{f(\widetilde{x})}\frac{f(\widetilde{x})}{f(x)} + \frac{f(\widetilde{x}) - f(x)}{f(x)} \\
&= \varepsilon_{alg}\cdot\frac{f(\widetilde{x})}{f(x)} + \varepsilon_{in} \\
&= \varepsilon_{alg}\cdot\frac{f(\widetilde{x}) - f(x) + f(x)}{f(x)} + \varepsilon_{in} \\
&= \varepsilon_{alg}\cdot\left(\frac{f(\widetilde{x}) - f(x)}{f(x)} + \frac{f(x)}{f(x)}\right) + \varepsilon_{in} \\
&= \varepsilon_{alg}\cdot(\varepsilon_{in} + 1) + \varepsilon_{in} \\
&\doteq \varepsilon_{alg} + \varepsilon_{in}.
\end{aligned}$$

□

1.4.4 Errore analitico

Finora abbiamo considerato solo funzioni razionali, ovvero contenenti le quattro operazioni aritmetiche fondamentali. È necessario tuttavia esprimere in macchina funzioni più complesse, come l'esponenziale, il logaritmo, le funzioni trigonometriche e così via.

L'esempio classico è quello della funzione esponenziale exp. La sua proprietà più importante (dal nostro punto di vista) è l'*analiticità*: la funzione esponenziale è uguale in ogni punto alla sua serie di Taylor centrata in 0, ovvero

$$\exp(x) = e^x = \sum_{n=0}^{\infty} \frac{d^n e^x}{dx^n} \frac{x^n}{n!} = \sum_{n=0}^{\infty} e^x \frac{x^n}{n!}.$$

Un modo per approssimarla è quindi sfruttare il Teorema di Taylor (con resto di Peano o Lagrange) *troncando* la serie al k-esimo termine:

$$e^x \approx \sum_{n=0}^{k} e^x \frac{x^n}{n!}.$$

Abbiamo quindi approssimato una funzione qualsiasi con una funzione razionale, generando un errore, chiamato **errore analitico**.

Definizione 1.4.11 – Errore analitico

Sia f una funzione non razionale e $\widetilde{f}$ la sua approssimazione razionale. Si dice **errore analitico** la quantità

$$\varepsilon_{an} := \frac{\widetilde{f}(x) - f(x)}{f(x)}.$$

La sequenza di passaggi per calcolare in macchina il valore di una funzione non razionale f in un punto $x \in \mathbb{R}$ diventa quindi

$$f(x) \overset{\varepsilon_{an}}{\longmapsto} \widetilde{f}(x) \overset{\varepsilon_{in}}{\longmapsto} \widetilde{f}(\widetilde{x}) \overset{\varepsilon_{alg}}{\longmapsto} g(\widetilde{x}).$$

Si può dimostrare il seguente teorema.

Teorema 1.4.12

$\varepsilon_{tot} \doteq \varepsilon_{an} + \varepsilon_{in} + \varepsilon_{alg}$.

1.5 ESEMPI DI CONDIZIONAMENTO E STABILITÀ

In questa sezione faremo alcuni esempi dello studio del condizionamento di problemi e della stabilità di algoritmi di macchina.

Esempio 1.5.1. Consideriamo la funzione $f(x) = \frac{x-1}{x}$ e studiamone il condizionamento per $x \neq 0$.

Osserviamo che la funzione è derivabile (almeno) due volte nell'intervallo $(0, +\infty)$, dunque possiamo calcolare il coefficiente di amplificazione di un dato $x \in (0, +\infty)$.

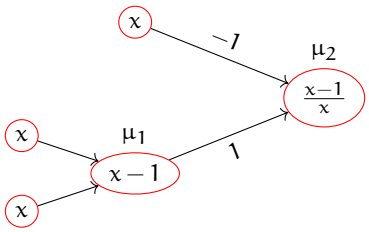
$$c_x = \frac{x}{f(x)} f'(x) = \frac{1}{x-1}.$$

Il problema è ben condizionato se e solo se $|c_x|$ è limitato superiormente per ogni $x \in (0, +\infty)$: tuttavia possiamo osservare che

$$\lim_{x \to 1} \left| \frac{1}{x-1} \right| = +\infty,$$

dunque il problema è mal condizionato in un intorno di 1.

Esempio 1.5.2. Consideriamo l'algoritmo $g_1(x) = (x \ominus 1) \oslash x$ e studiamone la stabilità. Il grafo di g_1 è



L'errore algoritmico totale è quindi dato da

$$\varepsilon_{alg} = \mu_2 + 1 \cdot (\mu_1 + 0 + 0) + 0 = \mu_2 + \mu_1.$$

Dimostriamo ora formalmente che l'algoritmo è stabile, ovvero che $\left|\varepsilon_{alg}\right|$ è sempre limitato superiormente.

In effetti si ha

$$\left|\varepsilon_{alg}\right| = |\mu_1 + \mu_2| \overset{(1)}{\leq} |\mu_1| + |\mu_2| \overset{(2)}{<} 2u$$

dove la disuguaglianza (1) è la disuguaglianza triangolare dei moduli, mentre la (2) viene dal fatto che gli errori locali sono sempre minori della precisione di macchina u.

1.5.1 Problema del calcolo della somma

Studiamo la funzione $f : \mathbb{R}^n_{>0} \to \mathbb{R}$ data da

$$f(x_1, \ldots, x_n) := \sum_{i=1}^{n} x_i.$$

Osserviamo che la funzione ha come dominio $\mathbb{R}^n_{>0}$, dunque tutti gli x_i devono essere strettamente positivi. Inoltre per semplicità scriveremo $\mathbf{x} = (x_1, \ldots, x_n)$.

CONDIZIONAMENTO Studiamo inizialmente il condizionamento del problema studiando l'errore inerente.

$$\begin{aligned}
\varepsilon_{in} &\doteq \sum_{i=1} c_{x_i} \varepsilon_{x_i} \\
&= \sum_{i=1}^{n} \frac{x_i}{f(\mathbf{x})} \frac{\partial f}{\partial x_i} \cdot \varepsilon_{x_i} \\
&= \sum_{i=1}^{n} \frac{x_i}{f(\mathbf{x})} \cdot 1 \cdot \varepsilon_{x_i} \\
&= \sum_{i=1}^{n} \frac{x_i}{f(\mathbf{x})} \varepsilon_{x_i}.
\end{aligned}$$

Per mostrare che il problema è ben condizionato dimostriamo che il valore assoluto

dell'errore inerente è limitato superiormente. In effetti si ha

$$
\begin{aligned}
|\varepsilon_{in}| &\doteq \left| \sum_{i=1}^{n} \frac{x_i}{f(\mathbf{x})} \varepsilon_{x_i} \right| \\
&\overset{(1)}{\leq} \sum_{i=1}^{n} \frac{|x_i|}{|f(\mathbf{x})|} |\varepsilon_{x_i}| \\
&\overset{(2)}{<} \frac{1}{|f(\mathbf{x})|} \sum_{i=1}^{n} |x_i| \cdot u \\
&= \frac{u}{|f(\mathbf{x})|} \sum_{i=1}^{n} |x_i| \\
&\overset{(3)}{=} u,
\end{aligned}
$$

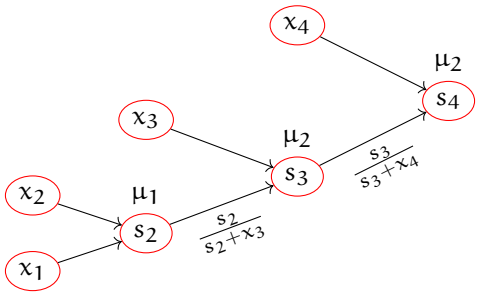
dove (1) è la disuguaglianza triangolare, (2) deriva dal fatto che $|\varepsilon_{x_i}| < u$ per ogni x_i e (3) segue dall'ipotesi che gli x_i siano strettamente positivi: in questo caso infatti il modulo di $f(\mathbf{x}) = x_1 + \cdots + x_n$ è uguale alla somma dei moduli degli x_i.

Si ha quindi che il problema è ben condizionato.

STABILITÀ Consideriamo l'algoritmo che consiste nell'effettuare le somma da sinistra a destra, ovvero

$$g_1(\mathbf{x}) = ((((x_1 + x_2) + x_3) + \cdots + x_{n-1}) + x_n).$$

Nel caso (ad esempio) di $n = 4$ l'algoritmo può essere rappresentato col seguente grafo:



dove $s_i = s_{i-1} + x_i = x_1 + \cdots + x_i$.

Calcoliamo ad esempio l'errore accumulatosi sul nodo s_3: si ha

$$\varepsilon_3 = \mu_3 + \frac{s_2}{s_2 + x_3} \underbrace{(\mu_2 + 0 + 0)}_{\varepsilon_2}.$$

Stessa cosa sul nodo s_4: infatti

$$\varepsilon_4 = \mu_4 + \frac{s_3}{s_3 + x_4} \varepsilon_3.$$

Più in generale si avrà quindi

$$\varepsilon_k = \mu_k + \frac{s_{k-1}}{s_{k-1} + x_k} \varepsilon_{k-1}.$$

Osserviamo che per definizione di s_i possiamo riscrivere $\frac{s_{k-1}}{s_{k-1}+x_k}$ come $\frac{s_{k-1}}{s_k}$ e questa frazione è sempre minore di 1: in questo modo riusciamo a limitare superiormente il modulo dell'errore

algoritmo. Infatti

$$\begin{aligned}\left|\varepsilon_{\text{alg}}\right| = |\varepsilon_n| &= \left|\mu_n + \frac{s_{n-1}}{s_n}\varepsilon_{n-1}\right| \\ &\leq |\mu_n| + \left|\frac{s_{n-1}}{s_n}\right||\varepsilon_{n-1}| \\ &< u + 1 \cdot |\varepsilon_{n-1}| \\ &< u + u + |\varepsilon_{n-2}| \\ &< \dots \\ &< (n-1)u.\end{aligned}$$

L'algoritmo è dunque stabile, ma solo se n non è troppo grande.

2

Algebra lineare numerica

2.1 NORME VETTORIALI

Iniziamo ora lo studio dell'algebra lineare numerica: studieremo algoritmi e problemi numerici sugli spazi vettoriali $\mathbb{F}^n$ e $\mathbb{F}^{n\times n}$, dove $\mathbb{F}$ è il campo dei numeri reali ($\mathbb{R}$) o quello dei numeri complessi ($\mathbb{C}$).

Uno dei nostri obiettivi sarà quello di risolvere sistemi lineari della forma $A\mathbf{x} = \mathbf{b}$ (con A matrice e $\mathbf{x}$, $\mathbf{b}$ vettori) sfruttando algoritmi di macchina sufficientemente efficienti e con errori piccoli; tuttavia quello che abbiamo studiato finora non ci consente di analizzare formalmente il condizionamento e la stabilità di problemi e algoritmi su vettori e matrici.

Esempio 2.1.1. Supponiamo ad esempio di voler risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$, dove

$$A \coloneqq \begin{pmatrix} 1.01 & 1.02 \\ 0.99 & 1 \end{pmatrix}, \qquad \mathbf{b} \coloneqq \begin{pmatrix} 2.03 \\ 1.99 \end{pmatrix}.$$

Risolvendo il sistema (ad esempio tramite il metodo di eliminazione gaussiana) otteniamo che la soluzione è il vettore

$$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Supponiamo ora di *perturbare* un singolo coefficiente della matrice A di 10^{-2}: questa perturbazione nella pratica potrebbe essere un errore di rappresentazione, oppure un errore algoritmico dovuto ai passi precedenti, eccetera. Scegliendo (ad esempio) di modificare il coefficiente in basso a sinistra otteniamo il sistema

$$\begin{pmatrix} 1.01 & 1.02 \\ 1 & 1 \end{pmatrix} \widetilde{\mathbf{x}} = \begin{pmatrix} 2.03 \\ 1.99 \end{pmatrix}.$$

La soluzione di questo sistema è il vettore

$$\widetilde{\mathbf{x}} = \begin{pmatrix} -0.02... \\ 2.01... \end{pmatrix}.$$

Perturbando una singola componente della matrice di 10^{-2} il risultato è stato completamente stravolto: una delle due componenti è raddoppiata, mentre l'altra è addirittura diventata negativa. Sembra ovvio quindi che il problema sia *malcondizionato* (una "piccola" perturbazione dei dati in ingresso ha generato una "grande" perturbazione dei risultati), tuttavia non abbiamo nessuno strumento formale per misurare *quanto* il problema sia malcondizionato, oppure confrontare questo errore con l'errore che otterremo perturbando un'altra componente.

Vogliamo quindi associare ad ogni vettore un *numero reale* non negativo in modo da poter esprimere il concetto di grandezza e di distanza tra vettori.

Definizione 2.1.2 – Norma vettoriale

Si dice **norma vettoriale** su $\mathbb{F}^n$ una funzione $f : \mathbb{F}^n \to \mathbb{R}$ che soddisfi le seguenti proprietà:

(1) $f(\boldsymbol{v}) \geq 0$ per ogni $\boldsymbol{v} \in \mathbb{F}^n$. Inoltre $f(\boldsymbol{v}) = 0$ se e solo se $\boldsymbol{v} = \boldsymbol{0}$.

(2) Per ogni $\alpha \in \mathbb{F}$, $\boldsymbol{v} \in \mathbb{F}^n$ si ha $f(\alpha\boldsymbol{v}) = |\alpha| f(\boldsymbol{v})$.

(3) Per ogni $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{F}^n$ si ha $f(\boldsymbol{v} + \boldsymbol{w}) \leq f(\boldsymbol{v}) + f(\boldsymbol{w})$ (**disuguaglianza triangolare**).

Una norma si indica spesso anche con il simbolo $\|\cdot\|$.

Un esempio banale di norma è il valore assoluto su $\mathbb{R}$, considerando $\mathbb{R}$ come uno spazio vettoriale su se stesso. In effetti:

(1) il valore assoluto di $x \in \mathbb{R}$ è sempre un numero non negativo, ed in particolare è 0 se e solo se $x = 0$;

(2) per ogni coppia di valori $x, y \in \mathbb{R}$ si ha che $|xy| = |x| \cdot |y|$;

(3) vale la disuguaglianza triangolare, e quindi per ogni coppia di valori $x, y \in \mathbb{R}$ si ha $|x + y| \leq |x| + |y|$.

Definizione 2.1.3 – Distanza

Sia X un insieme qualsiasi. Si dice **distanza** su X una funzione

$$d : X \times X \to \mathbb{R}$$

che soddisfi le seguenti proprietà:

(1) $d(x, y) \geq 0$ per ogni $x, y \in X$. Inoltre $d(x, y) = 0$ se e solo se $x = y$.

(2) $d(x, y) = d(y, x)$ per ogni $x, y \in X$.

(3) $d(x, z) \leq d(x, y) + d(y, z)$ per ogni $x, y, z \in X$.

Queste proprietà sono abbastanza semplici se pensiamo al concetto di distanza nella vita di tutti i giorni. Infatti:

(1) la distanza tra due punti è sempre un numero positivo, ed è 0 se e solo se i due punti sono effettivamente lo stesso punto;

(2) è indifferente andare dal primo punto al secondo o viceversa;

(3) la distanza tra i punti x e z è necessariamente minore o uguale alla distanza che percorreremmo se andassimo da x ad un terzo punto y e poi da y a z.

Osserviamo inoltre che nel caso di $\mathbb{F}^n$ esiste una distanza particolarmente facile da ricavare, poiché è direttamente **indotta da una norma**. Infatti se $\|\cdot\|$ è una norma qualsiasi, la funzione $d : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R}$ definita da

$$d(\boldsymbol{v}, \boldsymbol{w}) \coloneqq \|\boldsymbol{v} - \boldsymbol{w}\|$$

è una distanza.

Dimostrazione. Dobbiamo mostrare che la funzione d definita sopra sia effettivamente una distanza.

(1) Per il primo punto della definizione di norma si ha che $d(\boldsymbol{v}, \boldsymbol{w}) = \|\boldsymbol{v} - \boldsymbol{w}\| \geq 0$ per ogni $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{F}^n$. Inoltre questa quantità è 0 se e solo se $\boldsymbol{v} - \boldsymbol{w} = \boldsymbol{0}$, ovvero se e solo se $\boldsymbol{v} = \boldsymbol{w}$.

(2) Si ha che

$$d(v, w) = \|v - w\| = |-1|\|v - w\| \overset{(!)}{=} \|w - v\| = d(w, v),$$

dove il passaggio contrassegnato con (!) deriva dalla seconda proprietà delle norme.

(3) Per la disuguaglianza triangolare delle norme si ha che

$$d(v, u) = \|v - u\| = \|(v - w) + (w - u)\| \leq \|v - w\| + \|w - u\| = d(v, w) + d(w, u). \quad \square$$

2.1.1 Norme vettoriali principali

Introduciamo in questa sezione le principali norme vettoriali.

NORMA 2 O EUCLIDEA La norma più importante è chiamata **norma 2** o **norma euclidea** poiché dà origine alla distanza euclidea. È definita in questo modo:

$$\|v\|_2 := \sqrt{\sum_{i=1}^{n} |v_i|^2}.$$

Nel caso in cui lo spazio vettoriale sia $\mathbb{R}^2$ la norma euclidea ci dà la lunghezza euclidea del vettore v, o equivalentemente la distanza del punto v dall'origine degli assi:

$$\|v\|_2 = \sqrt{v_1^2 + v_2^2}.$$

Il valore assoluto presente nella formula è inutile se lo spazio vettoriale è $\mathbb{R}^n$ ma è necessario se lavoriamo sui complessi: ricordiamo infatti che se $z = a + ib$ è un numero complesso il suo modulo è il numero reale

$$|a + ib| = \sqrt{a^2 + b^2}$$

ed è necessario in quanto il risultato della norma deve essere un numero reale.

Osserviamo inoltre che in $\mathbb{R}^n$ si ha $\|x\|_2 = \sqrt{x^T x}$, dove x^T è il trasposto del vettore colonna x, mentre in $\mathbb{C}^n$ si ha $\|x\|_2 = \sqrt{x^H x}$, dove x^H è il trasposto coniugato del vettore x.

NORMA 1 La **norma 1** è la norma definita da

$$\|v\|_1 = \sum_{i=1}^{n} |v_i|.$$

NORMA INFINITO la **norma infinito** è la norma definita da

$$\|v\|_\infty = \max_{i=1,\dots,n} |v_i|.$$

2.1.2 Equivalenza topologica tra le norme di $\mathbb{F}^n$

Come abbiamo detto all'inizio del capitolo, useremo le norme vettoriali per determinare *quanto* errore abbiamo fatto nel fare un determinato calcolo. Tuttavia non esiste un'unica norma su $\mathbb{F}^n$ e ovviamente norme diverse danno risultati diversi.

Esempio 2.1.4. Consideriamo il vettore $x = (1, -1, 1)^T \in \mathbb{R}^3$. Allora

$$\|x\|_2 = \sqrt{3}, \qquad \|x\|_1 = 3, \qquad \|x\|_\infty = 1.$$

In realtà le norme si comportano tutte *più o meno* allo stesso modo, come ci viene garantito dal seguente Teorema.

Teorema 2.1.5 – Equivalenza topologica tra le norme

Siano f, g due norme su $\mathbb{F}^n$. Allora esistono due costanti $\alpha, \beta \in [0, +\infty)$ tali che per ogni $\boldsymbol{v} \in \mathbb{F}^n$ si ha

$$\alpha g(\boldsymbol{v}) \leq f(\boldsymbol{v}) \leq \beta g(\boldsymbol{v}).$$

Dimostrazione. Data nella **??** dell'Appendice. □

Questo Teorema è molto utile quando vogliamo ad esempio mostrare che un errore tende a 0. Infatti se ad esempio ε è un vettore che misura l'errore commesso e so che $\|\varepsilon\|_2 \to 0$, allora per qualsiasi altra norma p dovranno esistere due costanti $\alpha, \beta \in \mathbb{R}^+$ tali che

$$\alpha\|\varepsilon\|_2 \leq p(\varepsilon) \leq \beta\|\varepsilon\|_2.$$

Ma il membro sinistro e il membro destro di questa disequazione tendono a 0, dunque per il Teorema dei Carabinieri anche il membro centrale dovrà tendere a 0, ovvero la norma dell'errore tende a 0 *a prescindere dalla norma scelta*.

2.2 NORME MATRICIALI

Come per i vettori (ovvero per lo spazio vettoriale $\mathbb{F}^n$) possiamo definire un concetto di norma anche per lo spazio di matrici quadrate $\mathbb{F}^{n\times n}$.

Definizione 2.2.1 – Norma matriciale

Si dice **norma matriciale** su $\mathbb{F}^{n\times n}$ una funzione $f : \mathbb{F}^{n\times n} \to \mathbb{R}$ che soddisfi le seguenti proprietà:

(1) $f(A) \geq 0$ per ogni $A \in \mathbb{F}^{n\times n}$. Inoltre $f(A) = 0$ se e solo se $A = \mathbf{0}$.

(2) Per ogni $\alpha \in \mathbb{F}$, $A \in \mathbb{F}^{n\times n}$ si ha $f(\alpha A) = |\alpha| f(A)$.

(3) Per ogni $A, B \in \mathbb{F}^{n\times n}$ si ha $f(A + B) \leq f(A) + f(B)$ (**disuguaglianza triangolare**).

(4) Per ogni $A, B \in \mathbb{F}^{n\times n}$ si ha $f(AB) \leq f(A) \cdot f(B)$.

Una proprietà immediata delle norme matriciali è che per qualsiasi norma $\|\cdot\|$ si ha che $\|I_n\| \geq 1$ (dove $I_n \in \mathbb{F}^{n\times n}$ è la matrice identità di taglia $n \times n$). Infatti

$$\|I_n\| = \|I_n \cdot I_n\| \leq \|I_n\| \cdot \|I_n\| \implies 1 \leq \|I_n\|,$$

dove l'implicazione si ottiene dividendo entrambi i membri per $\|I_n\|$, che è sicuramente non nullo grazie alla proprietà (1) delle norme.

Osserviamo inoltre che, come nel caso delle norme vettoriali, ogni norma matriciale induce una distanza tra matrici $d : \mathbb{F}^{n\times n} \to \mathbb{F}^{n\times n} \to \mathbb{R}$, definita da

$$d(A, B) := \|A - B\|.$$

Tratteremo in particolare un tipo di norme matriciali, dette *norme matriciali indotte*.

Definizione 2.2.2 – Norma matriciale indotta

Sia $\|\cdot\|$ una norma vettoriale su $\mathbb{F}^n$. Si dice **norma matriciale indotta da** $\|\cdot\|$ la funzione $\|\cdot\|_M : \mathbb{F}^{n\times n} \to \mathbb{R}$ data da

$$\|A\|_M = \max\{\, \|A\boldsymbol{v}\| \;:\; \boldsymbol{v} \in \mathbb{F}^n, \|\boldsymbol{v}\| = 1 \,\}.$$

Per comodità di notazione d'ora in avanti scriveremo $\|\cdot\|$ per indicare una generica norma vettoriale e per la sua norma matriciale indotta: a seconda dell'argomento riusciremo a distinguere in quale dei due casi ci troviamo.

Osserviamo ora che la definizione di norma matriciale indotta è ben posta, ovvero che effettivamente la funzione $\|Ax\|$ ha massimo sull'insieme $S := \{ x \in \mathbb{F}^n : \|x\| = 1 \}$. Possiamo infatti notare che l'insieme S è chiuso e limitato e la funzione $x \mapsto \|Ax\|$ è continua (poiché la norma è una funzione continua), dunque per il Teorema di Weierstrass la funzione ammette un massimo, che sarà quindi il valore di $\|A\|_M$.

La dimostrazione del fatto che la funzione $\|\cdot\|_M$ definisca effettivamente una norma matriciale è lasciata all'appendice.

Osservazione 2.2.1. Non tutte le norme matriciali sono indotte: ad esempio si può dimostrare che la **norma di Frobenius**, definita da

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

non è indotta da alcuna norma vettoriale.

La principale proprietà delle norme matriciali indotte è data dal prossimo Teorema.

Teorema 2.2.3 – Compatibilità delle norme matriciali indotte

Sia $\|\cdot\|$ una norma matriciale indotta da una norma vettoriale. Allora per ogni $A \in \mathbb{F}^{n\times n}$, $x \in \mathbb{F}^n$ si ha

$$\|Ax\| \leq \|A\| \cdot \|x\|.$$

Osserviamo che Ax e x sono vettori, e dunque $\|Ax\|$, $\|x\|$ rappresenta la norma vettoriale dei due vettori, mentre A è una matrice e quindi $\|A\|$ rappresenta la norma matriciale di A.

Prima di dimostrare il Teorema mostriamo un semplice lemma che ci tornerà utile.

Lemma 2.2.4

Sia $x \in \mathbb{F}^n$, $x \neq 0$. Allora $\dfrac{x}{\|x\|} = 1$.

Dimostrazione. Per la proprietà (2) delle norme vettoriali

$$\left\| \frac{x}{\|x\|} \right\| = \frac{1}{\|x\|} \|x\| = 1. \qquad \square$$

Dimostriamo ora il Theorem 2.2.3.

Dimostrazione. Se $x = \mathbf{0}$ allora

$$\|Ax\| = \|\mathbf{0}\| = 0 = \|A\| \cdot \|x\|.$$

Se $x \neq \mathbf{0}$ si ha che $\|x\| \neq 0$, dunque la tesi è equivalente a dimostrare che

$$\frac{1}{\|x\|} \|Ax\| \leq \|A\| = \max_{\|v\|=1} \|Av\|.$$

Osserviamo che $\frac{1}{\|x\|} \in \mathbb{R}$, dunque per la proprietà (2) delle norme vettoriali si ha quindi

$$\frac{1}{\|x\|} \|Ax\| = \left\| A \frac{x}{\|x\|} \right\|.$$

Per il Lemma 2.2.4 sappiamo che $\bar{x} := \frac{x}{\|x\|}$ è un vettore di norma 1, dunque sicuramente vale che

$$\|A\bar{x}\| \leq \max_{\|v\|=1} \|Av\| = \|A\|,$$

ovvero la tesi. □

Il prossimo Teorema ci dà delle formule per calcolare direttamente il valore della norma 1/2/infinito di una matrice.

Teorema 2.2.5

Sia $A \in \mathbb{F}^{n\times n}$. Si ha che

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

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

$$\|A\|_2 := \sqrt{\rho(A^H A)}$$

dove $\rho : \mathbb{F}^{n\times n} \to \mathbb{R}$, dato da

$$\rho(M) := \max\{\, |\lambda| \;:\; \lambda \in \mathbb{C} \text{ autovalore di } M \,\}$$

è il **raggio spettrale** di M.

Osservazione 2.2.2. La norma infinito corrisponde a considerare le *righe* di A, calcolare per ognuna la somma dei moduli dei coefficienti e prendere il valore massimo; per calcolare la norma 1 invece dobbiamo seguire lo stesso procedimento, ma sulle colonne.

Esempio 2.2.6. Prendiamo ad esempio la matrice

$$A := \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}$$

e calcoliamone le varie norme.

Norma-∞ $\|A\|_\infty = \max\{\,|1|+|2|, |-1|+|3|\,\} = 4$.

Norma-1 $\|A\|_1 = \max\{\,|1|+|-1|, |2|+|3|\,\} = 5$.

Norma-2 Per calcolare la norma 2 dobbiamo prima calcolare $A^T A$ (la matrice è a coefficienti reali, dunque la trasposta coniugata è semplicemente la trasposta) e poi calcolare gli autovalori di questa matrice.

$$M := A^T A = \begin{pmatrix} 1 & -1 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 13 \end{pmatrix}.$$

Per trovare gli autovalori consideriamo il polinomio caratteristico

$$p_M(\lambda) = \det(M - \lambda I) = \det\begin{pmatrix} 2-\lambda & -1 \\ -1 & 13-\lambda \end{pmatrix} = (2-\lambda)(13-\lambda) - (-1)(-1) = \lambda^2 - 15\lambda + 25,$$

che ha come radici

$$\lambda_{1/2} = \frac{15 \pm \sqrt{125}}{2}.$$

Si ha quindi

$$\|A\|_2 = \rho(M) = \max\left\{ \left|\frac{15+\sqrt{125}}{2}\right|, \left|\frac{15-\sqrt{125}}{2}\right| \right\} = \frac{15+\sqrt{125}}{2}.$$

2.3 CONDIZIONAMENTO DELLA RISOLUZIONE DI UN SISTEMA LINEARE

Consideriamo il problema della risoluzione di un sistema lineare della forma $Ax = b$, con $A \in \mathbb{F}^{n\times n}$ matrice dei coefficienti, $b \in \mathbb{F}^n$ vettore dei termini noti e $x \in \mathbb{F}^n$ vettore delle incognite.

Siccome vogliamo studiare sistemi che hanno una e una sola soluzione considereremo sempre una matrice A invertibile, il che ci assicura che la soluzione del sistema esista sempre e sia unica.

Abbiamo già visto nell'Example 2.1.1 che una piccola perturbazione nei dati del problema può portare a grandi errori: tramite le norme matriciali e vettoriali possiamo esprimere questo errore tramite un numero reale.

Come nel caso monodimensionale quindi l'errore commesso nell'approssimare la matrice A con $\widetilde{A}$ e il vettore b con $\widetilde{b}$ e quindi risolvere il sistema $\widetilde{A}\widetilde{x} = \widetilde{b}$ rispetto al sistema $Ax = b$ è dato da

$$\frac{\|\widetilde{x} - x\|}{\|x\|}.$$

In particolare noi studieremo il caso in cui l'errore coinvolge solo il vettore dei termini noti (che da b diventa $\widetilde{b}$) e non la matrice dei coefficienti, che rimane invariata.

Un modo per studiare questo errore, e quindi il condizionamento del problema della risoluzione di un sistema lineare, è tramite il limite superiore che ci è dato dal seguente Teorema.

Teorema 2.3.1

Sia $A \in \mathbb{F}^{n\times n}$ una matrice non singolare, $b \in \mathbb{F}^n$ non nullo e sia $\widetilde{b}$ l'approssimazione del vettore dei termini noti. Data una norma vettoriale $\|\cdot\|$ e la sua norma matriciale indotta si ha

$$\frac{\|\widetilde{x} - x\|}{\|x\|} \leq \mu(A) \cdot \frac{\|\widetilde{b} - b\|}{\|b\|}$$

dove $\mu(A) := \|A\| \cdot \|A^{-1}\|$ è detto **numero di condizionamento**.

Dimostrazione. Siccome per definizione x e $\widetilde{x}$ sono rispettivamente soluzione di $Ax = b$ e $A\widetilde{x} = \widetilde{b}$ e A è invertibile si ha che

$$\widetilde{x} - x = A^{-1}\widetilde{b} - A^{-1}b = A^{-1}(\widetilde{b} - b),$$

da cui otteniamo, passando alle norme

$$\|\widetilde{x} - x\| = \|A^{-1}(\widetilde{b} - b)\| \leq \|A^{-1}\| \cdot \|\widetilde{b} - b\|. \tag{$*$}$$

Vogliamo ora dividere per $\|x\|$ il primo membro, per ottenere l'espressione finale. Osserviamo quindi che

$$\|b\| = \|Ax\| \leq \|A\| \cdot \|x\|$$

da cui segue che

$$\frac{1}{\|x\|} \leq \|A\| \cdot \|b\|.$$

Moltiplicando entrambi i membri della ($*$) per $\frac{1}{\|x\|}$ otteniamo quindi

$$\frac{\|\widetilde{x} - x\|}{\|x\|} \leq \frac{1}{\|x\|} \cdot \|A^{-1}\| \cdot \|\widetilde{b} - b\| \leq \frac{\|A\|}{\|b\|} \cdot \|A^{-1}\| \cdot \|\widetilde{b} - b\|,$$

come volevamo. □

2.4 LOCALIZZAZIONE DI AUTOVALORI

Vogliamo ora studiare dei teoremi per *localizzare* gli autovalori di una matrice $\mathbb{C}^{n\times n}$, ovvero per individuare in quali regioni del piano complesso essi sono compresi.

Questo può essere utile per molte applicazioni: ad esempio ricordiamo che

$$\|A\|_2 = \sqrt{\rho\left(AA^T\right)},$$

dove $\rho(M)$ è definito come il massimo degli autovalori di M. Sarebbe quindi comodo avere un modo per determinare velocemente gli autovalori di una matrice M, o comunque approssimarli.

Teorema 2.4.1 – Teorema di Hirsch

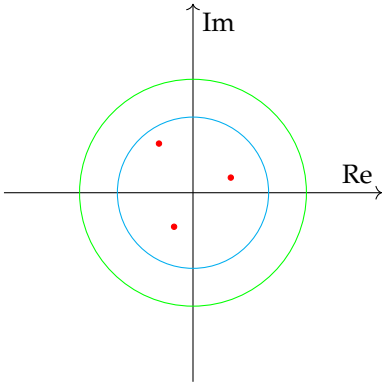
Sia $A \in \mathbb{C}^{n\times n}$ e sia $\|\cdot\|$ una norma matriciale indotta. Allora per ogni $\lambda \in \mathbb{C}$ autovalore di A si ha

$$|\lambda| \leq \|A\|.$$

Questo in particolare ci dice che

$$\rho(A) \leq \|A\|.$$

Da un punto di vista grafico questo significa che rappresentando gli autovalori di A nel piano complesso, essi sono tutti contenuti in un cerchio che ha raggio $\|A\|$, per qualunque norma matriciale indotta.



Dimostriamo ora il Teorema di Hirsch.

Dimostrazione. Per definizione di autovalore deve esistere un vettore $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{x} \neq \mathbf{0}$ tale che

$$A\mathbf{x} = \lambda\mathbf{x}.$$

Sia $\|\cdot\|$ la norma vettoriale che induce la norma matriciale che stiamo considerando: allora

$$|\lambda| \cdot \|\mathbf{x}\| = \|\lambda\mathbf{x}\| = \|A\mathbf{x}\| \leq \|A\| \cdot \|x\|.$$

Siccome $\mathbf{x} \neq \mathbf{0}$ si ha che $\|\mathbf{x}\| > 0$, dunque dividendo entrambi i membri per $\|\mathbf{x}\|$ otteniamo

$$|\lambda| \leq \|A\|. \qquad \square$$

2.4.1 Teoremi di Gershgorin

Vogliamo ora localizzare ancora più precisamente gli autovalori di una matrice.

Definizione 2.4.2 – Cerchi di Gershgorin

Sia $A \in \mathbb{C}^{n\times n}$. Per ogni $1 \le i \le n$ definiamo l'i-esimo **cerchio di Gershgorin** come l'insieme

$$K_i := \left\{ z \in \mathbb{C} \,:\, |z - a_{ii}| \le \sum_{j=1, j\neq i}^{n} |a_{ij}| \right\}.$$

L'i-esimo cerchio è quindi dato da tutti i punti la cui distanza da a_{ii} (l'elemento della diagonale nella i-esima riga) è minore della somma di tutti gli elementi dell'i-esima riga, escluso l'elemento della diagonale. Segue quindi che

- a_{ii} rappresenta il *centro* del cerchio K_i;
- $\sum_{j=1,j\neq i}^{n} |a_{ij}|$, ovvero la somma degli elementi dell'i-esima riga escluso a_{ii}, è il *raggio* del cerchio.

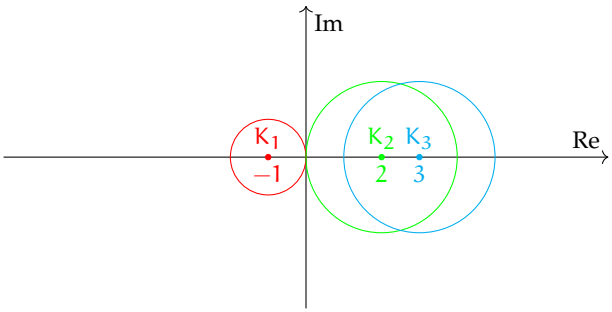
Esempio 2.4.3. Consideriamo la matrice

$$A = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 2 & 1 \\ 2 & 0 & 3 \end{pmatrix}.$$

I corrispondenti cerchi di Gershgorin sono

$$K_1 = \{ z \in \mathbb{C} \,:\, |z+1| \le |-1| + |0| = 1 \} \tag{2.1}$$
$$K_2 = \{ z \in \mathbb{C} \,:\, |z-2| \le |-1| + |1| = 2 \} \tag{2.2}$$
$$K_3 = \{ z \in \mathbb{C} \,:\, |z-3| \le |2| + |0| = 2 \}. \tag{2.3}$$

Rappresentiamoli nel piano complesso:



Teorema 2.4.4 – Primo Teorema di Gershgorin

Sia $A \in \mathbb{C}^{n\times n}$ e K_i i suoi cerchi di Gershgorin. Allora per ogni $\lambda \in \mathbb{C}$ autovalore di A si ha

$$\lambda \in \bigcup_{i=1}^{n} K_i.$$

Dimostrazione. Sia $\lambda \in \mathbb{C}$ un autovalore di A, ovvero tale che esiste $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{x} \neq \mathbf{0}$ tale che $A\mathbf{x} = \lambda\mathbf{x}$. Scrivendo la matrice e i vettori per esteso questo equivale a

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \vdots \\ \lambda x_n \end{pmatrix}.$$

Svolgendo il prodotto otteniamo che per ogni riga $1 \leq i \leq n$ vale che

$$\begin{pmatrix} a_{i1} & \dots a_{in} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = a_{i1}x_1 + \cdots + a_{in}x_n = \sum_{j=1}^{n} a_{ij}x_j = \lambda x_i.$$

Portando fuori dalla sommatoria il termine corrispondente a $j = i$ otteniamo

$$\begin{aligned} &\left(\sum_{j=1, j\neq i}^{n} a_{ij}x_j \right) + a_{ii}x_i = \lambda x_i \\ \iff &\sum_{j=1, j\neq i}^{n} a_{ij}x_j = \lambda x_i - a_{ii}x_i \end{aligned}$$

sempre per ogni $i = 1, \dots, n$.

Sia ora p l'indice per cui

$$\left|x_p\right| = \|\mathbf{x}\|_\infty = \max_{1\leq i\leq n} |x_i|,$$

ovvero p è l'indice della componente di modulo massimo. Siccome $\mathbf{x} \neq \mathbf{0}$ sicuramente c'è almeno una componente diversa da 0, dunque $\left|x_p\right| > 0$.

Riscrivo quindi l'ultima equazione nel caso particolare di $i = p$:

$$\sum_{j=1, j\neq p}^{n} a_{pj}x_j = \lambda x_p - a_{pp}x_p$$

che passando ai moduli diventa

$$\left|\lambda - a_{pp}\right| \cdot \left|x_p\right| = \left| \sum_{j=1, j\neq p}^{n} a_{pj}x_j \right| \leq \sum_{j=1, j\neq p}^{n} \left|a_{pj}\right| \cdot \left|x_j\right|,$$

dove all'ultimo passaggio abbiamo usato la disuguaglianza triangolare dei moduli.

Dividendo per $\left|x_p\right|$ (siccome è sempre strettamente maggiore di 0) si ha

$$\begin{aligned} \left|\lambda - a_{pp}\right| &\leq \frac{1}{\left|x_p\right|} \sum_{j=1, j\neq p}^{n} \left|a_{pj}\right| \cdot \left|x_j\right| \\ &= \sum_{j=1, j\neq p}^{n} \left|a_{pj}\right| \cdot \frac{\left|x_j\right|}{\left|x_p\right|}. \end{aligned}$$

Osserviamo ora che $\frac{\left|x_j\right|}{\left|x_p\right|}$ è sempre minore o uguale a 1: infatti x_p è la componente di modulo massimo, dunque il numeratore della frazione è sempre minore o uguale al denominatore. Si ha quindi

$$\left|\lambda - a_{pp}\right| \leq \sum_{j=1, j\neq p}^{n} \left|a_{pj}\right|,$$

ovvero $\lambda \in K_p$.

Siccome λ appartiene ad un cerchio di Gershgorin in particolare apparterrà all'unione di tutti i cerchi, come volevasi dimostrare. □

Possiamo considerare anche i cerchi *per colonne*, ovvero

$$H_j := \left\{ z \in \mathbb{C} \,:\, \left|z - a_{jj}\right| \leq \sum_{i=1, i\neq j}^{n} \left|a_{ij}\right| \right\}.$$

I cerchi per colonne corrispondono ai cerchi per righe della matrice trasposta A^T: ricordando che ogni matrice ha gli stessi autovalori della sua trasposta si ha che ogni autovalore $\lambda \in \mathbb{C}$ di A deve appartenere all'unione dei cerchi per colonna, ovvero

$$\lambda \in \bigcup_{j=1}^{n} H_j.$$

In particolare varrà quindi

$$\lambda \in \left(\bigcup_{i=1}^{n} K_i\right) \cap \left(\bigcup_{j=1}^{n} H_j\right)$$

per ogni λ autovalore di A.

Enunciamo ora (senza dimostrazione) il Secondo Teorema di Gershgorin.

Teorema 2.4.5 – Secondo Teorema di Gershgorin

Sia $A \in \mathbb{C}^{n\times n}$ tale che k dei cerchi di Gershgorin di A siano disgiunti dai rimanenti $n-k$. Sia inoltre $\mathcal{K}_1$ l'unione dei k cerchi che formano il primo gruppo e $\mathcal{K}_2$ l'unione dei rimanenti.
Allora k autovalori di A appartengono a $\mathcal{K}_1$, mentre i rimanenti $n-k$ appartengono a $\mathcal{K}_2$.

2.4.2 Predominanza diagonale

Definizione 2.4.6 – Matrice a predominanza diagonale

na matrice $A \in \mathbb{C}^{n\times n}$ si dice **a predominanza diagonale per righe** se per ogni riga $i = 1, \dots, n$ si ha

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|,$$

ovvero se il modulo dell'elemento sulla diagonale a_{ii} è strettamente maggiore della somma dei moduli del resto degli elementi della riga i.
Analogamente A si dice **a predominanza diagonale per colonne** se per ogni colonna $j = 1, \dots, n$ si ha

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

Osserviamo che se A è a predominanza diagonale per righe allora A^T lo è per colonne, e viceversa.

Proposizione 2.4.7

Una matrice a predominanza diagonale (per righe o per colonne) è invertibile.

Dimostrazione. Ricordiamo che una matrice è invertibile se e solo se 0 non è autovalore: mostriamo quindi che 0 non appartiene all'unione dei cerchi di Gershgorin. Questo implica che un qualsiasi tipo di predominanza diagonale è sufficiente: se mostriamo che una matrice a predominanza diagonale è invertibile (ovvero che 0 non è autovalore) segue che anche la sua trasposta è invertibile (poiché ha gli stessi autovalori); analogamente se partiamo da una matrice a predominanza diagonale per colonne.

Sia quindi A una matrice a predominanza diagonale per righe, $1 \leq i \leq n$ qualsiasi e consideriamo l'i-esima riga della matrice A. Si ha

$$|0 - a_{ii}| = |a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|,$$

dunque 0 non appartiene all'i-esimo cerchio di Gershgorin.

Siccome $0 \notin K_i$ per ogni i segue anche che $0 \notin \bigcup K_i$, ovvero 0 non può essere autovalore (per il Primo Teorema di Gershgorin). □

2.5 FATTORIZZAZIONE LU

Abbiamo già iniziato a parlare di sistemi lineari e della loro risoluzione precedentemente. Dopo averne studiato il condizionamento studiamo degli algoritmi per trovare efficientemente le soluzioni di sistemi della forma $Ax = b$, dove A è una matrice, b è il vettore dei termini noti e x è il vettore delle incognite.

2.5.1 Risoluzione di un sistema triangolare

Se la matrice A è triangolare superiore è particolarmente semplice risolvere il sistema, in quanto possiamo usare la tecnica di **sostituzione all'indietro**.

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

Infatti l'ultima riga ci fornisce l'equazione $a_{nn}x_n = b_n$, da cui ricaviamo $x_n = \dfrac{b_n}{a_{nn}}$ (a patto che a_{nn} sia diverso da 0); usando questo risultato possiamo risolvere l'equazione della penultima riga, cioè

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

che diventa

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

In generale, tramite questo metodo, la formula per calcolare x_k è

$$x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj}x_j}{a_{kk}}.$$

Un programma MatLab che esprime questo algoritmo è il seguente:

```
% given an upper triangular matrix A of order n
% and a n-dimensional vector b
% it calculates the solution of the system |Ax = b|
function x = upp_triang_solve(A, b)
    x(n) = b(n) / A(n, n)
    for k = n-1 : -1 : 1
        s = 0
        for j = k+1 : n
            s = s + A(k, j)*x(j)
        end
        x(k) = (b(k) - s) / A(k, k)
    end
end
```

Una veloce analisi del costo computazionale dell'algoritmo ci dice che esso ha costo, al caso pessimo, di $O(n^2)$: infatti il ciclo interno ha costo $O(n-k)$, e viene ripetuto per $k = 1, \dots, n-1$, da cui viene il costo

$$\sum_{k=1}^{n-1} O(n-k) = O(n^2).$$

Questo costo non è migliorabile: infatti la matrice ha $O(n^2)$ elementi, da cui non è possibile diminuire il numero di operazioni fatte.

Se la matrice è triangolare inferiore possiamo usare la stessa tecnica ma partendo dalla prima riga, tramite la **sostituzione all'avanti**.

2.5.2 Mosse di Gauss e fattorizzazione LU

Se la matrice A non è triangolare il metodo più usato per risolvere il sistema $Ax = b$ è l'**algoritmo di Gauss**, che sfrutta le cosiddette mosse di Gauss.

Definizione 2.5.1 – Mosse di Gauss

Sia $A \in \mathbb{F}^{n\times n}$ una matrice. Una **mossa di Gauss per riga** è un'operazione sulle righe della matrice con una delle seguenti forme:

(1) scambiare la riga i con la riga j;

(2) moltiplicare la riga i per un coefficiente $\alpha \in \mathbb{F}$, $\alpha \neq 0$;

(3) sottrarre alla riga i la riga j moltiplicata per un coefficiente α.

La strategia è quindi quella di applicare ripetutamente le mosse di Gauss (secondo il cosiddetto **algoritmo di Gauss**) per ridurre la matrice A ad una matrice U triangolare superiore:

$$[A|b] = [A^{(0)}|b^{(0)}] \rightsquigarrow [A^{(1)}|b^{(1)}] \rightsquigarrow \cdots \rightsquigarrow [A^{(n-1)}|b^{(n-1)}] = [U|y].$$

Mostreremo nella prossima sezione che in alcuni casi particolari possiamo descrivere una singola mossa di Gauss come un prodotto tra matrici

$$A^{(k+1)} = E^{(k+1)}A^{(k)}$$

dove $E^{(k+1)}$ sarà una matrice invertibile e triangolare inferiore.

In tal caso segue quindi che

$$U = A^{(n-1)} = E^{(n-1)}A^{(n-2)} = E^{(n-1)}E^{(n-2)}A^{(n-3)} = \cdots = E^{(n-1)} \cdots E^{(1)}A.$$

Siccome abbiamo detto che tutte le matrici $E^{(i)}$ sono invertibili anche il loro prodotto lo sarà: chiamiamo dunque

$$L^{-1} := E^{(n-1)} \cdots E^{(1)},$$

da cui segue che $U = L^{-1}A$, ovvero $A = LU$.

Il sistema $Ax = b$ diventa quindi $LUx = b$, che può essere risolto come

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

Sappiamo che U è triangolare superiore, dunque possiamo risolvere il sistema $Ux = y$ usando i metodi studiati precedentemente. Allo stesso modo, dato che L è data dal prodotto di matrici triangolari inferiori anch'essa è triangolare inferiore, dunque possiamo risolvere anche il sistema $Ly = b$ efficientemente, trovando quindi una soluzione al sistema originale.

Siamo quindi interessati a studiare in che modo e in quali casi si può scrivere A come prodotto di due matrici, di cui una è triangolare inferiore (L, da *lower triangular*) e una è triangolare superiore (U, da *upper triangular*). Diamo dunque la seguente definizione.

Definizione 2.5.2 – Fattorizzazione LU

Sia $A \in \mathbb{F}^{n\times n}$ una matrice. Si dice che A è **fattorizzabile LU** se esistono due matrici $L, U \in \mathbb{F}^{n\times n}$ tali che

(1) L è triangolare inferiore ed ha tutti 1 sulla diagonale principale;

(2) U è triangolare superiore;

(3) vale che
$$A = LU.$$

Il seguente Teorema ci dà delle condizioni sufficienti sulla fattorizzabilità.

Teorema 2.5.3 – Esistenza e unicità della fattorizzazione LU

Sia $A \in \mathbb{F}^{n\times n}$. Se le sottomatrici principali di testa di A di ordine k, con $k = 1, \dots, n-1$, sono invertibili, allora A è fattorizzabile LU e tale fattorizzazione è unica.

Dimostrazione. Dimostriamo la tesi per induzione su n.

- Caso base Supponiamo $n = 1$. Allora la matrice $A = \begin{bmatrix} a_{11} \end{bmatrix}$ non ha sottomatrici, dunque la condizione è vacuamente verificata. Inoltre
  - una possibile fattorizzazione di A è data da
  $$L = \begin{bmatrix} 1 \end{bmatrix}, \qquad U = \begin{bmatrix} a_{11} \end{bmatrix}$$
  - tale fattorizzazione è unica: infatti L deve essere la matrice con un singolo 1 poiché sulla diagonale non può avere valori diversi, da cui segue che $U = A$.

  La tesi è quindi verificata nel caso base.

- Passo induttivo Supponiamo che la tesi sia vera per matrici di ordine m, per ogni $m < n$, e mostriamola per n.

  Supponiamo quindi che A abbia sottomatrici di testa non singolari. Per mostrare che A sia fattorizzabile LU dobbiamo trovare due matrici L, U che rispettano le condizioni sulla fattorizzazione. Scrivendole a blocchi, si ha
$$A = \left(\begin{array}{c|c} A_{n-1} & z \\ \hline x^T & a_{nn} \end{array}\right) = \left(\begin{array}{c|c} \widehat{L} & \mathbf{0} \\ \hline w^T & 1 \end{array}\right)\left(\begin{array}{c|c} \widehat{U} & y \\ \hline \mathbf{0}^T & \beta \end{array}\right)$$
  dove A_{n-1}, z, x^T, a_{nn} sono noti, mentre $\widehat{L}, \widehat{U}, w^T, y$ e β sono incognite.

  Svolgendo il prodotto a blocchi devono quindi valere le seguenti condizioni:
$$\begin{cases} A_{n-1} = \widehat{L}\widehat{U} + \mathbf{0}\mathbf{0}^T = \widehat{L}\widehat{U} \\ z = \widehat{L}y + \beta\mathbf{0} = \widehat{L}y \\ x^T = w^T\widehat{U} + 1 \cdot \mathbf{0}^T = w^T\widehat{U} \\ a_{nn} = w^T y + \beta. \end{cases}$$

  (1) Osserviamo che la prima equazione corrisponde alla fattorizzazione LU di A_{n-1}. Siccome A_{n-1} è la sottomatrice di testa di A di ordine $n-1$ le sue sottomatrici di testa sono uguali alle sottomatrici di testa di A, dunque sono invertibili per ipotesi.

  Per l'ipotesi induttiva segue quindi che la fattorizzazione LU di A_{n-1} esiste ed è unica, dunque $\widehat{L}, \widehat{U}$ esistono e sono uniche.

(2) Siccome la matrice $\widehat{L}$ è triangolare inferiore segue che il suo determinante è il prodotto degli elementi della diagonale, ovvero $\det \widehat{L} = 1$. Segue quindi che il sistema $\widehat{L}y = z$ ammette una e una sola soluzione, dunque y è univocamente determinato.

(3) Applicando l'operatore di trasposizione ad entrambi i membri dell'equazione si ottiene il sistema equivalente

$$\widehat{U}^T w = x.$$

La matrice $\widehat{U}^T$ è invertibile: infatti $A_{n-1} = \widehat{L}\widehat{U}$ e $\widehat{L}$ invertibile implica che $\widehat{U} = \widehat{L}^{-1}A_{n-1}$. Per il Teorema di Binet segue quindi che

$$\det \widehat{U} = \det \widehat{L}^{-1} \cdot \det A_{n-1}.$$

$\widehat{L}^{-1}$ è certamente invertibile, mentre A_{n-1} è invertibile in quanto è la sottomatrice di testa di A di ordine $n-1$, e per ipotesi tutte le sottomatrici di testa di A sono invertibili. Segue quindi che $\widehat{U}$ è invertibile, da cui anche $\widehat{U}^T$ è invertibile, dunque il vettore w esiste ed è unico.

(4) L'ultima equazione è equivalente a

$$\beta = a_{nn} - w^T y.$$

Abbiamo dimostrato che w e y esistono e sono unici, dunque anche β esiste ed è univocamente determinato.

Segue quindi che la matrice A è fattorizzabile LU, da cui la tesi. □

2.6 ALGORITMO DI GAUSS

Durante lo studio della fattorizzazione LU abbiamo fatto l'ipotesi che, in alcune condizioni, ogni mossa di Gauss può essere rappresentata dalla moltiplicazione a sinistra per una qualche matrice invertibile e triangolare inferiore.

Vogliamo ora dare una formalizzazione dell'algoritmo di Gauss che ci spiega quali sono queste matrici, come sono fatte, perché funzionano e in quali condizioni esistono.

Ricordiamo che l'algoritmo di Gauss per ridurre a scalini (cioè ad una matrice triangolare superiore) funziona così:

1. sia $A \in \mathbb{R}^{n\times n}$ una matrice invertibile, poniamo $k := 1$;
2. consideriamo l'elemento in posizione (k, k): se è nullo scambiamo la riga k con una riga successiva;
3. per ogni riga i successiva alla k-esima, sottraiamo da R_i la riga R_k moltiplicata per $\frac{a_{ik}}{a_{kk}}$: in questo modo annulliamo tutti gli elementi della colonna k-esima che sono dopo la riga k;
4. poniamo $k := k + 1$ e torniamo al passo 2.

Definiamo ora il tipo di matrici che sfrutteremo per trasformare questo calcolo sulle righe in un'operazione di prodotto matriciale.

Definizione 2.6.1 – Matrice elementare di Gauss

Sia $E \in \mathbb{R}^{n\times n}$. E si dice **matrice elementare di Gauss** se esistono $k \in \mathbb{N}$, $v \in \mathbb{R}^n$ tali che

1. v è nullo nelle prime k posizioni, ovvero

$$v_1 = \cdots = v_k = 0,$$

2. si ha
$$E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T,$$
dove $\boldsymbol{e}_k$ è il k-esimo vettore della base canonica di $\mathbb{R}^n$.

Esempio 2.6.2. Se $k = 2$ ad esempio otteniamo

$$\begin{aligned}
E &= I_n - \boldsymbol{v}\boldsymbol{e}_2^T \\
&= I_n - \begin{pmatrix} 0 \\ 0 \\ v_3 \\ \vdots \\ v_k \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \end{pmatrix} \\
&= \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} - \begin{bmatrix} 0 & & & \\ & 0 & & \\ & -v_3 & 0 & \\ & \vdots & & \ddots \\ & -v_n & & 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & -v_3 & 1 & \\ & \vdots & & \ddots \\ & -v_n & & 1 \end{bmatrix}.
\end{aligned}$$

Proprietà delle matrici elementari di Gauss

Le matrici elementari di Gauss hanno diverse proprietà che ci torneranno utili nel dimostrare che effettivamente modellano l'algoritmo di Gauss.

Proposizione 2.6.3 – Proprietà delle Matrici Elementari di Gauss

(1) Le matrici elementari di Gauss sono triangolari inferiori, e gli elementi della diagonale sono tutti uguali ad 1.

(2) Le matrici elementari di Gauss sono invertibili. Inoltre, se $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ è una matrice elementare di Gauss, allora la sua inversa è ancora una matrice elementare di Gauss, data da
$$E^{-1} = I_n + \boldsymbol{v}\boldsymbol{e}_k^T.$$

(3) Dato $x \in \mathbb{R}^n$ con $x_k \neq 0$ esiste una matrice elementare di Gauss E tale che
$$E\boldsymbol{x} = (x_1, \dots, x_k, 0, \dots, 0)^T.$$
La matrice è della forma $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$, dove
$$\boldsymbol{v} = \left(0, \dots, 0, \frac{x_{k+1}}{x_k}, \dots, \frac{x_n}{x_k}\right)^T.$$

(4) Siano $E_k = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$, $E_h = I_n - \boldsymbol{w}\boldsymbol{e}_h^T$ due matrici elementari di Gauss, con $h > k$. Allora
$$E_k E_h = I_n - \boldsymbol{v}\boldsymbol{e}_k^T - \boldsymbol{w}\boldsymbol{e}_h^T.$$

(5) Sia $\boldsymbol{y} \in \mathbb{R}^n$, E_k una matrice elementare di Gauss. Allora il calcolo di $E_k\boldsymbol{y}$ ha un costo computazionale di $O(n - k)$ flops (*floating point operations*).

Dimostriamo separatamente le varie proprietà.

Dimostrazione della Proprietà (1). È sufficiente dimostrare che $\boldsymbol{v}\boldsymbol{e}_k^T$ sia triangolare inferiore con gli zeri sulla diagonale, poiché da questo segue che $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ è triangolare inferiore con 1 sulla diagonale.

Siano quindi $i, j \in \mathbb{N}$ con $i \leq j \leq n$ due indici e mostriamo che e_{ij}, ovvero l'elemento della matrice E in posizione (i, j) (che osserviamo essere un elemento sulla diagonale o sopra la diagonale, in quanto abbiamo richiesto che $j \geq i$) è uguale a 0.

Per definizione di prodotto tra matrici si ha che

$$e_{ij} = v_i \cdot \delta_{jk}$$

dove

$$\delta_{jk} := \begin{cases} 1, & \text{se } j = k \\ 0, & \text{se } j \neq k \end{cases}$$

è l'elemento di posizione j-esima del vettore $\boldsymbol{e}_k$.

Consideriamo due casi:

- se $i \leq k$ allora $v_i = 0$ per definizione di matrice elementare di Gauss, dunque $e_{ij} = 0$;
- se $i > k$ allora $j \geq i > k$, dunque $\delta_{jk} = 0$ (poiché j è strettamente maggiore di k), dunque $e_{ij} = 0$.

Segue quindi la tesi. □

Dimostrazione della Proprietà (2). Dalla proprietà (1) segue che il determinante di una matrice elementare è il prodotto degli elementi sulla diagonale poiché è triangolare inferiore, cioè è 1 e dunque le matrici elementari di Gauss sono invertibili.

Mostriamo ora che l'inversa di $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ è effettivamente $E^{-1} = I_n + \boldsymbol{v}\boldsymbol{e}_k^T$.

$$\begin{aligned} EE^{-1} &= (I_n - \boldsymbol{v}\boldsymbol{e}_k^T)(I_n + \boldsymbol{v}\boldsymbol{e}_k^T) \\ &= I_n + \boldsymbol{v}\boldsymbol{e}_k^T - \boldsymbol{v}\boldsymbol{e}_k^T - \boldsymbol{v}\boldsymbol{e}_k^T\boldsymbol{v}\boldsymbol{e}_k^T \\ &= I_n - \boldsymbol{v}\boldsymbol{e}_k^T\boldsymbol{v}\boldsymbol{e}_k^T. \end{aligned}$$

Per mostrare che EE^{-1} è l'identità basta quindi mostrare che $\boldsymbol{v}\boldsymbol{e}_k^T\boldsymbol{v}\boldsymbol{e}_k^T$ è la matrice nulla. Osserviamo innanzitutto che per associatività del prodotto tra matrici

$$\boldsymbol{v}\boldsymbol{e}_k^T\boldsymbol{v}\boldsymbol{e}_k^T = \boldsymbol{v}(\boldsymbol{e}_k^T\boldsymbol{v})\boldsymbol{e}_k^T$$

e il termine tra parentesi è uno scalare, per cui commuta con il prodotto tra matrici, ottenendo

$$(\boldsymbol{e}_k^T\boldsymbol{v}) \cdot \boldsymbol{v}\boldsymbol{e}_k^T.$$

Per definizione di prodotto tra vettori si ha che

$$\boldsymbol{e}_k^T\boldsymbol{v} = \sum_{i=1}^{n} \delta_{ik} v_i.$$

Distinguiamo ora due casi:

- quando $i \neq k$ si ha che $\delta_{ik} = 0$, dunque $\delta_{ik} v_i = 0$;
- quando $i = k$ si ha che $v_i = 0$ (poiché $\boldsymbol{v}$ è nullo nelle prime k posizioni), dunque $\delta_{ik} v_i = 0$.

Segue quindi che tutti i termini della somma sono nulli, da cui $\boldsymbol{e}_k^T\boldsymbol{v} = 0$, dunque $EE^{-1} = I_n$ come volevamo. □

Dimostrazione della Proprietà (3). Poniamo $\boldsymbol{v} = (0, \dots, 0, v_{k+1}, \dots, v_n)$ e dimostriamo che per ogni $i > k$ si ha che

$$v_i = \frac{x_i}{x_k}.$$

Per definizione di matrice elementare di Gauss si ha che

$$E\boldsymbol{x} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -v_{k+1} & \ddots & \\ & & \vdots & & \\ & & -v_n & & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Considerando questo prodotto tra matrici come un sistema lineare (le cui incognite però sono i v_i), le prime k righe corrispondono alle equazioni

$$1 \cdot x_i = x_i,$$

che sono banalmente verificate.

Sia quindi $i > k$. L'equazione codificata dalla riga i-esima è dunque

$$-v_i x_k + 1 \cdot x_i = 0,$$

da cui ricaviamo che

$$v_i = \frac{x_i}{x_k},$$

come volevamo. □

Dimostrazione della Proprietà (5). Sia $\boldsymbol{z} \coloneqq E_k \boldsymbol{y}$. Allora

$$\boldsymbol{z} = (I_n - \boldsymbol{v}\boldsymbol{e}_k^\top) \cdot \boldsymbol{y} = \boldsymbol{y} - \boldsymbol{v}\boldsymbol{e}_k^\top \boldsymbol{y} = \boldsymbol{y} - y_k \boldsymbol{v},$$

dove y_k è la k-esima coordinata del vettore $\boldsymbol{y}$, e l'ultima uguaglianza viene dal fatto che il prodotto del trasposto del k-esimo vettore della base canonica (cioè $\boldsymbol{e}_k^\top$) con un qualsiasi altro vettore $\boldsymbol{w} \in \mathbb{R}^n$ dà come risultato la k-esima coordinata di $\boldsymbol{w}$.

Ma allora

$$z_i = \begin{cases} y_i, & \text{se } i \leq k \\ y_i - y_k v_i, & \text{se } i > k. \end{cases}$$

Per calcolare la coordinata i-esima (con $i > k$) di $\boldsymbol{z}$ abbiamo quindi bisogno di 2 operazioni floating point (un prodotto e una differenza) e le coordinate di indice maggiore di k sono $n-k$, dunque in totale abbiamo bisogno di un numero di operazioni nell'ordine di $O(n-k)$. □

2.6.1 Algoritmo di Gauss tramite matrici elementari

Possiamo ora dimostrare che l'algoritmo di Gauss può essere espresso come una sequenza di moltiplicazioni a sinistra per delle matrici elementari di Gauss.

Sia innanzitutto $A \in \mathbb{R}^{n \times n}$ una matrice e definiamo $A^{(k)}$ come la matrice ottenuta al k-esimo passo dell'algoritmo di Gauss (in particolare quindi $A = A^{(0)}$).

Dato che ad ogni passo azzeriamo tutti gli elementi della matrice sotto la diagonale, la matrice $A^{(k-1)}$ sarà della forma

$$A^{(k-1)} = \begin{bmatrix} a_{11}^{(k-1)} & & & * & & \\ & \ddots & & & & \\ & & a_{k-1,k-1}^{(k-1)} & & & \\ & & & a_{kk}^{(k-1)} & \ddots & \\ & 0 & & a_{k+1,k}^{(k-1)} & & \\ & & & \vdots & * & \\ & & & a_{nk}^{(k-1)} & & a_{nn}^{(k-1)} \end{bmatrix}$$

dove l'area contrassegnata da 0 contiene tutti zeri, mentre le aree contrassegnate da $*$ contengono gli altri elementi della matrice, che possono essere uguali o diversi da 0.

Il nostro scopo è ottenere la matrice $A^{(k)}$, in cui sono azzerati anche tutti gli elementi della colonna k-esima che si trovano al di sotto dell'elemento della diagonale (ovvero di a_{kk}).

Per far ciò sfruttiamo la Proprietà (3): supponendo che $a_{kk}^{(k-1)} \neq 0$ possiamo considerare la matrice elementare di Gauss

$$E_k := I_n - \mathbf{m}^{(k)} \boldsymbol{e}_k^T,$$

dove

$$m_i^{(k)} = \begin{cases} 0, & \text{se } i \leq k \\ \dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, & \text{se } i > k. \end{cases}$$

Tale matrice elementare (per la proprietà (3)) annulla tutti i valori sotto $a_{kk}^{(k-1)}$.

Consideriamo allora il prodotto $E_k A^{(k-1)}$: se $\boldsymbol{A}_1^{(k-1)}, \ldots, \boldsymbol{A}_n^{(k-1)}$ sono le colonne di $A^{(k-1)}$ si ha che

$$\begin{aligned} E_k A^{(k-1)} &= E_k \cdot \left[\boldsymbol{A}_1^{(k-1)} \;\middle|\; \ldots \;\middle|\; \boldsymbol{A}_k^{(k-1)} \;\middle|\; \ldots \;\middle|\; \boldsymbol{A}_n^{(k-1)} \right] \\ &= \left[E_k \boldsymbol{A}_1^{(k-1)} \;\middle|\; \ldots \;\middle|\; E_k \boldsymbol{A}_k^{(k-1)} \;\middle|\; \ldots \;\middle|\; E_k \boldsymbol{A}_n^{(k-1)} \right]. \end{aligned}$$

A questo punto possiamo notare che

- se $i < k$ il prodotto $E_k \boldsymbol{A}_i^{(k-1)}$ è uguale a $A_i^{(k-1)}$, ovvero moltiplicare per E_k **non cambia le prime** $k-1$ **colonne di** A: infatti il calcolo fatto nella Dimostrazione della Proprietà (3) notiamo che i primi k elementi vengono sempre mantenuti, mentre i successivi (gli elementi di posto $j = k+1, \ldots, n$) vengono aggiornati tramite la legge

  $$a_{ij}^{(k-1)} - m_j^{(k)} a_{ik}^{(k-1)}.$$

  Tuttavia, siccome $a_{ik}^{(k-1)} = 0$ segue che anche gli elementi di posto successivo al k-esimo rimangono invariati;

- se $i = k$ il prodotto $E_k \boldsymbol{A}_i^{(k-1)}$ **azzera tutti gli elementi sotto** $a_{kk}^{(k-1)}$ (l'abbiamo definita proprio per questo scopo);

- se $i > k$ il prodotto $E_k \boldsymbol{A}_i^{(k-1)}$ cambia solo gli elementi **al di sotto della** k**-esima riga** (per lo stesso motivo delle prime k colonne).

Dunque la matrice $A^{(k)}$, che corrisponde al k-esimo passaggio dell'algoritmo di Gauss, è data da

$$A^{(k)} := E_k A^{(k-1)}.$$

Studiamo quindi come sono fatti gli elementi di $A^{(k)}$: essi sono

$$a_{ij}^{(k)} = \begin{cases} a_{ij}^{(k-1)}, & \text{se } i = 1,\dots,k, \quad j = 1,\dots,n \\ a_{ij}^{(k-1)} = 0, & \text{se } i = k+1,\dots,n,\ j = 1,\dots,k-1 \\ 0, & \text{se } i = k+1,\dots,n,\ j = k \\ a_{ij}^{(k-1)} - m_i^{(k)} a_{kj}^{(k-1)}, & \text{se } i = k+1,\dots,n,\ j = k+1,\dots,n \end{cases}$$

Questo è effettivamente quello che si fa quando si calcola l'algoritmo di Gauss applicando le mosse per righe. Segue quindi che il metodo per risolvere un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ consiste nel moltiplicare ripetutamente a sinistra per le matrici elementari definite sopra:

$$A\boldsymbol{x} = \boldsymbol{b} \rightsquigarrow E_1 A\boldsymbol{x} = E_1\boldsymbol{b} \rightsquigarrow \cdots \rightsquigarrow (E_{n-1}\cdots E_1 A)\boldsymbol{x} = E_{n-1}\cdots E_1\boldsymbol{b}.$$

Notiamo che così facendo dobbiamo anche trasformare il vettore dei termini noti $\boldsymbol{b}$, che al passo k-esimo diventerà

$$b_i^{(k)} := \begin{cases} b_i^{(k-1)}, & \text{se } i \le k \\ b_i^{(k-1)} - m_i^{(k)} b_k^{(k-1)}, & \text{se } i > k. \end{cases}$$

La matrice $U := E_{n-1}\cdots E_1 A = A^{(n-1)}$ ricavata all'ultimo passo è triangolare superiore, quindi possiamo risolvere il sistema

$$U\boldsymbol{x} = \boldsymbol{b}^{(n-1)}.$$

Ancora meglio, possiamo sfruttare quanto fatto finora per fattorizzare la matrice A in forma LU.

Infatti, come abbiamo visto nella sezione precedente, possiamo considerare la matrice

$$L^{-1} := E_{n-1}\cdots E_2 E_1,$$

ovvero

$$L = E_1^{-1} E_2^{-1} \cdots E_{n-1}^{-1}.$$

Per la proprietà (2), ognuna delle matrici E_i^{-1} è una matrice elementare di Gauss; inoltre per la proprietà (4) le matrici sono moltiplicate nell'*ordine corretto*, dunque possiamo calcolare L senza svolgere i prodotti. Vale quindi la seguente

Proposizione 2.6.4

Se l'algoritmo di Gauss è applicabile (ovvero se $a_{ii}^{(i-1)} \neq 0$ per ogni $i = 1,\dots,n-1$) allora A è fattorizzabile LU e

$$L = \begin{pmatrix} 1 & & & \\ m_1^{(1)} & \ddots & & \\ \vdots & m_2^{(2)} & \ddots & \\ m_n^{(1)} & m_n^{(2)} & \cdots & m_n^{(n-1)} \quad 1 \end{pmatrix}.$$

In particolare la condizione su A vista in questa sezione, ovvero che $a_{ii}^{(i-1)}$ sia non zero per ogni $i = 1,\dots,n-1$, e la condizione sufficiente per la fattorizzazione LU sono equivalenti, come confermato dal seguente Teorema (che non dimostreremo).

Proposizione 2.6.5

Sia $A \in \mathbb{R}^{n\times n}$ e sia $A_{(k)}$ la matrice ottenuta al k-esimo passo dell'algoritmo di Gauss. Vale che le sottomatrici principali di testa di A di ordine k (per $k = 1, \dots, n-1$) sono tutte invertibili se e solo se $a_{kk}^{(k-1)} \neq 0$ per ogni $k = 1, \dots, n-1$.

2.6.2 Costo computazionale dell'Algoritmo di Gauss

Calcoliamo il costo computazionale dell'algoritmo di Gauss.

Al passo k-esimo (in cui passiamo da $A^{(k-1)}$ a $A^{(k)} := E_k A^{(k-1)}$) dobbiamo calcolare:

- il vettore dei moltiplicatori $m^{(k)}$,
- gli elementi di indice (i, j) con $i = k+1, \dots, n$, $j = k+1, \dots, n$.

Per quanto riguarda il vettore dei moltiplicatori, esso ha $n-k$ componenti non nulle e ognuna di queste componenti è calcolabile con una singola divisione, dunque sono sufficienti $n-k$ operazioni.

Invece per gli elementi che vengono modificati al passaggio k-esimo ricordiamo che

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_i^{k-1} a_{kj}^{k-1}.$$

Dunque per ognuno di essi abbiamo bisogno di 2 operazioni (un prodotto e una differenza), e vi sono $(n-k)^2$ elementi con indici nell'intervallo richiesto, dunque sono necessarie $2(n-k)^2$ operazioni.

Per svolgere tutti i passi dell'algoritmo di Gauss avremo quindi bisogno di

$$\sum_{k=1}^{n-1} (n-k) + 2(n-k)^2$$

operazioni floating point. Ponendo $i := n-k$ otteniamo quindi

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

che è il costo computazionale dell'algoritmo di Gauss nel caso generale.

Osservazione 2.6.1. Nell'ultima sequenza di uguaglianze abbiamo usato i fatti che

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}, \qquad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$$

2.6.3 Scambio di righe

Abbiamo visto che il metodo delle matrici elementari di Gauss funziona solo assumendo che $a_{kk}^{(k-1)} \neq 0$. Tuttavia, ciò non ci consente di risolvere semplici sistemi lineari, come ad esempio

$$\begin{pmatrix} 0 & 1 & -2 \\ 1 & 1 & 1 \\ -3 & -1 & 7 \end{pmatrix} x = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}.$$

Infatti $a_{11}^0 = 0$, dunque non possiamo trovare una matrice elementare di Gauss che annulli gli altri coefficienti della prima riga. Abbiamo quindi bisogno di un metodo sistematico per poter **scambiare** le righe della matrice A.

Definizione 2.6.6 – Matrice di permutazione

Una matrice $P \in \mathbb{R}^{n\times n}$ si dice **matrice di permutazione** se esattamente un elemento di P è uguale ad 1 per ogni riga e per ogni colonna, ovvero se P è una matrice ottenuta permutando le righe (o le colonne) della matrice identità I_n.

Si può dimostrare che moltiplicare a sinistra per una matrice di permutazione corrisponde a scambiare le righe di una matrice: in questo modo possiamo effettuare tutti gli scambi necessari a far avvenire con successo l'algoritmo di Gauss *a priori*.

Partendo dal sistema $A\mathbf{x} = \mathbf{b}$ andiamo quindi a risolvere il sistema $PA\mathbf{x} = P\mathbf{b}$.

Vale in particolare il seguente risultato

Proposizione 2.6.7

Sia $A \in \mathbb{R}^{n\times n}$ una matrice invertibile. Allora esiste $P \in \mathbb{R}^{n\times n}$ matrice di permutazione tale che PA è fattorizzabile LU, ovvero tale che esistano uniche L triangolare inferiore con 1 sulla diagonale, U triangolare superiore, tali che

$$PA = LU.$$

A

Teoremi secondari

A.1 "9-PERIODICO"

Proposizione A.1.1 – 9-periodico

In base 10 i numeri $0.\overline{9}$ e 1 sono uguali.

Forniamo due diverse dimostrazioni di questa proposizione.

Prima dimostrazione. Dalle formule per trasformare i numeri periodici in frazioni sappiamo che $0.\overline{9} = {}^{9}/_{9} = 1$. □

Seconda dimostrazione. Espandendo la definizione di numero periodico otteniamo che

$$0.\overline{9} = 0.999\cdots = 9\cdot 10^{-1} + 9\cdot 10^{-2} + 9\cdot 10^{-3} + \cdots = \sum_{i=1}^{\infty} 9\cdot 10^{-i}.$$

Sfruttando la formula della serie geometrica si ottiene che

$$\begin{aligned}
\sum_{i=1}^{\infty} 9\cdot 10^{-i} &= 9\cdot \sum_{i=1}^{\infty}\left(\frac{1}{10}\right)^{i} \\
&= 9\cdot\left(\left(\sum_{i=0}^{\infty}\left(\frac{1}{10}\right)^{i}\right) - \left(\frac{1}{10}\right)^{0}\right) \\
&= 9\cdot\left(\frac{1}{1-{}^{1}/_{10}} - 1\right) \\
&= 9\cdot\left(\frac{10}{9} - 1\right) \\
&= 9\cdot\frac{1}{9} \\
&= 1.
\end{aligned}$$

□

La proposizione vale in generale in una base β qualsiasi ($\beta \geq 2$).

Proposizione A.1.2

In base β ($\beta \geq 2$) vale che $0.\overline{(\beta-1)} = 1$.

Dimostrazione. La dimostrazione è uguale alla seconda dimostrazione della Proposition A.1.1. □

A.2 LA PRECISIONE DI MACCHINA È UN LIMITE SUPERIORE STRETTO

In questa sezione mostriamo che la precisione di macchina $\mathbf{u}$ è sempre strettamente maggiore all'errore relativo ε_x per ogni $x \in \mathbb{R}$ tale che $x \neq 0$ e $\omega \leq |x| \leq \Omega$, anche usando l'arrotondamento come metodo di approssimazione.

Dalla dimostrazione del Theorem 1.2.5 si ha che

$$|\varepsilon_x| = \frac{1}{2}\beta^{1-p} = \mathbf{u}$$

se e solo se la disuguaglianza

$$\frac{\left|\mathrm{arr}(x) - x\right|}{|x|} \leq \frac{1/2\beta^{p-t}}{\beta^{p-1}}$$

è in realtà un'uguaglianza, ovvero se e solo se valgono le seguenti due condizioni:

- $\left|\mathrm{arr}(x) - x\right| = \frac{1}{2}\beta^{p-t}$
- $|x| = \beta^{p-1}$.

Tuttavia, come abbiamo notato nella Remark 1.2.2, la prima condizione vale se e solo se

$$|x| = \frac{1}{2}(a + b) = \frac{1}{2}(a + a + \beta^{p-t}) = a + \frac{1}{2}\beta^{p-t}.$$

È quindi necessario che

$$\begin{aligned}
\beta^{p-1} &= a + \frac{1}{2}\beta^{p-t} \\
&= \beta^{p}\left(\sum_{i=1}^{t} d_i \beta^{-i}\right) + \frac{1}{2}\beta^{p-t} \\
&= \beta^{p}\left(\sum_{i=1}^{t} d_i \beta^{-i} + \frac{1}{2}\beta^{-t}\right) \\
\iff \beta^{-1} &= \sum_{i=1}^{t} d_i \beta^{-i} + \frac{1}{2}\beta^{-t}.
\end{aligned}$$

Tuttavia ciò è assurdo: infatti il termine al primo membro ha una singola cifra decimale diversa da zero, ovvero quella di β^{-1}, mentre il termine del secondo membro ha anche delle cifre non nulle nelle posizioni β^{-t} e/o β^{-t-1}.

Segue quindi che $|\varepsilon_x| < u$ in qualsiasi caso.

A.3 DIMOSTRAZIONE DELL'EQUIVALENZA TOPOLOGICA TRA NORME

In questa sezione dimostreremo il Theorem 2.1.5: per far ciò sfrutteremo il seguente lemma.

Lemma A.3.1

Sia $\|\cdot\|$ una norma su $\mathbb{F}^n$. Allora la funzione

$$x \mapsto \|x\|$$

è una funzione continua.

Dimostrazione. Sia $x_0 \in \mathbb{F}^n$ un punto qualsiasi e dimostriamo che la norma è continua in x_0.

Sia quindi $\varepsilon > 0$ qualsiasi: vogliamo determinare un $\delta > 0$ tale che per ogni $x \in \mathbb{F}^n$ con $\|x - x_0\| < \delta$ si ha $\big|\|x\| - \|x_0\|\big| < \varepsilon$.

Osserviamo che

$$\|x\| = \|x - x_0 + x_0\| \leq \|x_0\| + \|x - x_0\|,$$

da cui $\|x\| - \|x_0\| \leq \|x - x_0\|$. Scambiando i ruoli di x e x_0 otteniamo che $\|x_0\| - \|x\| \leq \|x_0 - x\| = \|x - x_0\|$, quindi combinando le due disuguaglianze si ha

$$\big|\|x\| - \|x_0\|\big| \leq \|x - x_0\|.$$

Poniamo dunque $\delta = \varepsilon$. Allora

$$\big|\|x\| - \|x_0\|\big| \leq \|x - x_0\| < \delta = \varepsilon,$$

cioè la norma è continua in x_0, da cui segue (per generalità di x_0) che la norma è una funzione continua. □

Possiamo quindi dimostrare il Theorem 2.1.5.

B

Prerequisiti di Algebra Lineare

B.1 SPAZI VETTORIALI

Iniziamo rivedendo i concetti fondamentali sugli spazi vettoriali.

Definizione B.1.1 – Campo

Si dice **campo** un insieme $\mathbb{F}$ dotato di due operazioni, chiamate solitamente *somma* e *prodotto* e indicate con i simboli $+$ e $\cdot$, tali che

(S1) la somma è associativa;

(S2) la somma è commutativa;

(S3) esiste un elemento $0_{\mathbb{F}} \in \mathbb{F}$ (indicato anche 0 se il campo è sottointeso) che è elemento neutro per la somma;

(S4) ogni elemento ammette un opposto rispetto alla somma, ovvero per ogni $\alpha \in \mathbb{F}$ esiste $-\alpha \in \mathbb{F}$ tale che
$$\alpha + (-\alpha) = 0;$$

(P1) il prodotto è associativo;

(P2) il prodotto è commutativo;

(P3) esiste un elemento $1_{\mathbb{F}} \in \mathbb{F}$ (indicato anche con 1 se il campo è sottointeso) che è elemento neutro per il prodotto;

(P4) ogni elemento diverso da 0 ammette un inverso rispetto al prodotto, ovvero per ogni $\alpha \in \mathbb{F} \setminus \{0\}$ esiste $\alpha^{-1} \in \mathbb{F}$ tale che
$$\alpha \cdot \alpha^{-1} = 1;$$

(D) vale la proprietà distributiva del prodotto rispetto alla somma.

Esempi di campo sono i numeri reali $\mathbb{R}$, i numeri razionali $\mathbb{Q}$, i numeri complessi $\mathbb{C}$. D'ora in avanti indicheremo con $\mathbb{F}$ un generico campo (in particolare $\mathbb{F}$ nella pratica sarà uno tra $\mathbb{R}$ e $\mathbb{C}$).

Definizione B.1.2 – Spazio vettoriale

Si dice **spazio vettoriale** sul campo $\mathbb{F}$ un insieme V dotato di due operazioni, una somma $+ : V \times V \to V$ e un *prodotto per scalare* $\cdot : \mathbb{F} \times V \to V$, che soddisfano le seguenti proprietà:

(S1) la somma è associativa;

(S2) la somma è commutativa;

(S3) la somma ammette un elemento neutro, indicato con $\mathbf{0}$ o 0_V;

(S4) ogni elemento di V ammette un opposto rispetto alla somma, ovvero per ogni $\boldsymbol{v} \in V$ esiste $-\boldsymbol{v} \in V$ tale che
$$\boldsymbol{v} + (-\boldsymbol{v}) = \mathbf{0};$$

(P1) il prodotto per scalare distribuisce sulla somma, ovvero per ogni $\boldsymbol{v}, \boldsymbol{w} \in V$ e per ogni $\alpha \in \mathbb{F}$ si ha
$$\alpha \cdot (\boldsymbol{v} + \boldsymbol{w}) = \alpha\boldsymbol{v} + \alpha\boldsymbol{w};$$

(P2) la somma di scalari distribuisce sul prodotto per scalare, ovvero per ogni $\alpha, \beta \in \mathbb{F}$, $\boldsymbol{v} \in V$ si ha
$$(\alpha + \beta)\boldsymbol{v} = \alpha\boldsymbol{v} + \beta\boldsymbol{v};$$

(P3) il prodotto per scalari è associativo, ovvero per ogni $\alpha, \beta \in \mathbb{F}$, $\boldsymbol{v} \in V$ si ha
$$\alpha(\beta\boldsymbol{v}) = (\alpha\beta)\boldsymbol{v};$$

(P4) l'elemento neutro del prodotto (in $\mathbb{F}$) è neutrale rispetto al prodotto per scalare, ovvero per ogni $\boldsymbol{v} \in V$ si ha
$$1_{\mathbb{F}} \cdot \boldsymbol{v} = \boldsymbol{v}.$$

Il campo $\mathbb{F}$ viene spesso detto **campo degli scalari** di V.

Indicheremo solitamente gli elementi del campo degli scalari con lettere greche minuscole ($\alpha, \beta, \dots$) e gli elementi dello spazio vettoriale con lettere in grassetto ($\boldsymbol{v}, \boldsymbol{w}, \dots$).
Esempi di spazi vettoriali sono

- l'insieme dei vettori colonna a componenti in $\mathbb{F}$ con n elementi, indicato con $\mathbb{F}^n$;
- l'insieme dei polinomi a coefficienti in $\mathbb{F}$, indicato con $\mathbb{F}[x]$;
- l'insieme delle matrici $m \times n$ a coefficienti in $\mathbb{F}$, indicate con $\mathbb{F}^{m\times n}$.

Definizione B.1.3 – Sottospazio

Sia V uno spazio vettoriale su $\mathbb{F}$. Un **sottospazio** di V è un insieme $U \subseteq V$ che sia anch'esso uno spazio vettoriale su $\mathbb{F}$.

Definizione B.1.4 – Span di un sottoinsieme

Sia V uno spazio vettoriale su $\mathbb{F}$ e sia $S \subseteq V$. Allora si dice **span** di S l'insieme
$$\operatorname{span}(S) \coloneqq \{\, \alpha_1\boldsymbol{v}_1 + \cdots + \alpha_k\boldsymbol{v}_k \;:\; \alpha_i \in \mathbb{F}, \boldsymbol{v}_i \in S \,\}.$$

Proposizione B.1.5

Per ogni $S \subseteq V$ si ha che $\operatorname{span}(S)$ è un sottospazio vettoriale di V.

B.2 INDIPENDENZA E BASI

Assumeremo implicitamente che V sia uno spazio vettoriale su $\mathbb{F}$.

Definizione B.2.1 – Indipendenza lineare

I vettori $v_1, \dots, v_n \in V$ formano un insieme di **vettori linearmente indipendenti** se si ha che

$$\alpha_1 v_1 + \cdots + \alpha_n v_n = 0 \implies \alpha_1 = \cdots = \alpha_n = 0_{\mathbb{F}}.$$

Definizione B.2.2 – Insieme di generatori

I vettori $v_1, \dots, v_n \in V$ formano un insieme di **generatori per** V se per ogni $v \in V$ esistono $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ tali che

$$v = \alpha_1 v_1 + \dots \alpha_n v_n.$$

Equivalentemente $v_1, \dots, v_n$ sono generatori se

$$\operatorname{span}\{ v_1, \dots, v_n \} = V.$$

Definizione B.2.3 – Base

L'insieme formato dai vettori $v_1, \dots, v_n$ forma una **base** per V se valgono entrambe le seguenti condizioni:

1. $\{ v_1, \dots, v_n \}$ è un insieme di vettori linearmente indipendenti;
2. $\{ v_1, \dots, v_n \}$ è un insieme di generatori per V.

Chiameremo **base canonica** di $\mathbb{F}^n$ l'insieme formato dai vettori

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \dots, \quad e_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

Non tutti gli spazi vettoriali ammettono una base *finita*: ad esempio nessun sottoinsieme finito di $\mathbb{F}[x]$ lo genera, e quindi non può esistere una base.

Definizione B.2.4 – Dimensione di uno spazio vettoriale

Sia V uno spazio vettoriale su $\mathbb{F}$ e sia $\mathcal{B}$ una base di V. Si dice **dimensione di** V **su** $\mathbb{F}$ la quantità

$$\dim_{\mathbb{F}} V := \#\mathcal{B}.$$

Quando il campo degli scalari è sottointeso scriveremo semplicemente $\dim V$.

Ad esempio:

- la dimensione di $\mathbb{F}^n$ è $\dim \mathbb{F}^n = n$ (basta considerare la base canonica);
- la dimensione di $\mathbb{F}^{m \times n}$ è $\dim \mathbb{F}^{m \times n} = mn$.

Teorema B.2.5 – Teorema della dimensione

Sia V uno spazio vettoriale su $\mathbb{F}$ di dimensione n. Allora ogni base di V ha n elementi.

B.3 APPLICAZIONI LINEARI

Definizione B.3.1 – Applicazione lineare

Siano U, V due spazi vettoriali su un campo $\mathbb{F}$. Si dice **applicazione lineare** una funzione $f : U \to V$ tale che

1. $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$ per ogni $\mathbf{x}, \mathbf{y} \in U$;
2. $f(\alpha\mathbf{x}) = \alpha f(\mathbf{x})$ per ogni $\mathbf{x} \in U, \alpha \in \mathbb{F}$.

Una volta fissate due basi $\mathcal{B}_U$ e $\mathcal{B}_V$ rispettivamente del dominio e del codominio, ad esempio

$$\mathcal{B}_U = \{ \mathbf{u}_1, \dots, \mathbf{u}_n \}, \qquad \mathcal{B}_V = \{ \mathbf{v}_1, \dots, \mathbf{v}_m \},$$

riusciamo ad esprimere l'applicazione f in modo univoco tramite una matrice $A_f \in \mathbb{F}^{m\times n}$.

In effetti per ogni vettore $\mathbf{u}_j$ nella base $\mathcal{B}_U$ l'immagine del vettore mediante f, ovvero $f(\mathbf{u}_j)$, è un elemento di V e può pertanto essere espresso in termini degli elementi della base $\mathcal{B}_V$ di V: esisteranno cioè degli scalari $\alpha_{1j}, \dots, \alpha_{mj}$ tali che

$$f(\mathbf{u}_j) = \alpha_{1j}\mathbf{v}_1 + \dots \alpha_{mj}\mathbf{v}_m.$$

Questi scalari formeranno la j-esima colonna della matrice associata, che avrà quindi la forma

$$A_f = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mn} \end{pmatrix}$$

Definizione B.3.2 – Immagine e nucleo di un'applicazione lineare

Sia $f : U \to V$ un'applicazione lineare.

(1) Si dice **kernel** (o **nucleo**) di f l'insieme

$$\ker f := \{ \mathbf{u} \in U \,:\, f(\mathbf{u}) = \mathbf{0}_V \}.$$

(2) Si dice **immagine** di f l'insieme

$$\operatorname{Im} f := \{ f(\mathbf{u}) \in V \,:\, \mathbf{u} \in U \}.$$

Si ha che $\ker f$ è un sottospazio di U e $\operatorname{Im} f$ è un sottospazio di V. Inoltre vale il seguente teorema.

Teorema B.3.3 – Teorema della dimensione

$$\dim V = \dim \operatorname{Im} f + \dim \ker f.$$

B.4 INVERTIBILITÀ

Definizione B.4.1 – Matrice invertibile

Una matrice $A \in \mathbb{F}^{n\times n}$ si dice invertibile se esiste $A^{-1} \in \mathbb{F}^{n\times n}$ tale che

$$AA^{-1} = A^{-1}A = I_n$$

dove $I_n \in \mathbb{F}^{n\times n}$ è l'identità.

Elenchiamo alcune condizioni equivalenti per l'invertibilità di una matrice:

- A è invertibile,
- $\det A \neq 0$,
- $\ker A = \{\mathbf{0}\}$,
- A è di rango massimo,
- le colonne di A sono linearmente indipendenti,
- le righe di A sono linearmente indipendenti,
- $0 \in \mathbb{F}$ non è autovalore di A (lo vedremo nella prossima sezione).

B.5 AUTOVETTORI E AUTOVALORI

Definizione B.5.1 – Autovettori e autovalori

Sia $A \in \mathbb{F}^{n\times n}$ una matrice. Un vettore $\mathbf{x} \in \mathbb{F}^n$, $\mathbf{x} \neq \mathbf{0}$ si dice **autovettore** di A se esiste $\lambda \in \mathbb{F}$ tale che

$$A\mathbf{x} = \lambda\mathbf{x}.$$

Tale λ si dice **autovalore** relativo all'autovettore $\mathbf{x}$.

Il modo più semplice per calcolare gli autovalori di una matrice è tramite il polinomio caratteristico.

Definizione B.5.2 – Polinomio caratteristico

Sia $A \in \mathbb{F}^{n\times n}$ una matrice. Si dice **polinomio caratteristico** di A il polinomio

$$p_A(x) = \det(A - xI_n),$$

dove $I_n \in \mathbb{F}^{n\times n}$ è la matrice identità.

Proposizione B.5.3

Sia $A \in \mathbb{F}^{n\times n}$ una matrice. Allora $\lambda \in \mathbb{F}$ è autovalore di A (relativo a qualche autovettore) se e solo se $p_A(\lambda) = 0$, ovvero se e solo se λ è radice del polinomio caratteristico.

La strategia per trovare gli autovalori consiste quindi nel calcolare il polinomio caratteristico e trovarne tutte le radici.

Esempio B.5.4. Consideriamo ad esempio la matrice

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

e calcoliamone gli autovalori.

$$\begin{aligned} p_A(x) &= \det(A - xI_n) \\ &= \det\left(\begin{pmatrix} 3 & -2 \\ 4 & -1 \end{pmatrix} - x\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) \\ &= \det\begin{pmatrix} 3-x & -2 \\ 4 & -1-x \end{pmatrix} \\ &= (3-x)(-1-x) - (-2)\cdot 4 \\ &= x^2 - 2x + 5. \end{aligned}$$

Le radici di questo polinomio sono

$$\lambda_{1/2} = 1 \pm \sqrt{1-5} = 1 \pm \sqrt{-4} = 1 \pm 2i$$

che corrispondono agli autovalori complessi di A.

Facciamo un po' di considerazioni sugli autovalori.

(1) Ogni matrice reale o complessa *ha esattamente* n *autovalori complessi*: infatti per il Teorema Fondamentale dell'Algebra ogni polinomio ha esattamente n radici complesse (non necessariamente tutte distinte).

(2) Se la matrice è a coefficienti reali anche il polinomio caratteristico lo sarà. In tal caso se $\lambda \in \mathbb{C}$ è radice allora anche $\overline{\lambda} \in \mathbb{C}$ lo è (*le radici sono a coppie*).

(3) Se n è dispari allora una matrice ha almeno una radice reale.

(4) Una matrice è invertibile se e solo se 0 è autovalore. Infatti:

($\Longrightarrow$) Se A è invertibile allora $\ker A = \{\mathbf{0}\}$, ovvero per ogni vettore $\mathbf{x} \in \mathbb{F}^n$ si ha

$$A\mathbf{x} \neq \mathbf{0} = 0\mathbf{x}.$$

Dunque in particolare nessun autovettore ha come autovalore 0.

($\Longleftarrow$) Se 0 è autovalore allora esiste un vettore $\mathbf{x} \in \mathbb{F}^n$ tale che

$$A\mathbf{x} = 0\mathbf{x} = 0,$$

ovvero il kernel di A non è banale, da cui A non è invertibile.