1 I NUMERI E IL CALCOLATORE

1.1 RAPPRESENTAZIONE DEI NUMERI

TEOREMA 1.1(**di rappresentazione**). *Sia $B \geq 2$ un numero naturale. Sia x un numero reale non nullo. Allora esistono e sono unici un numero intero relativo p ed una successione di numeri interi $\{a_i\}, i = 1, 2, ...,$ che soddisfano le proprietà:*

1) $0 \leq a_i \leq B - 1$,

2) $a_1 \neq 0$,

3) *gli a_i non sono tutti uguali a $B - 1$ da un certo indice in poi,*

tali che

$$x = sign(x) B^p \sum_{i=1}^{\infty} a_i B^{-i},$$

dove $sign(x) = 1$ *se* $x > 0$, $sign(x) = -1$ *se* $x < 0$.

□

Poiché $a_1 \neq 0$, la rappresentazione si dice **normalizzata**.
B è la **base** della rappresentazione.
$a_i, i = 1, 2, ...,$ sono le **cifre** della rappresentazione (se $B = 2$ le cifre utilizzabili sono $0, 1$, se $B = 10$, le usuali da 0 a 9, se $B = 16$ si utilizzano le cifre da 0 a 9 ed i simboli A, B, C, D, E, F)
p è la **caratteristica.**
il numero $\sum_{i=1}^{\infty} a_i B^{-i}$ è la **mantissa**.

La rappresentazione è:
a) **finita** (numero finito di cifre) se e solo se $x = \pm\alpha/\beta$, dove α e β sono numeri interi primi fra loro, con β tale che tutti i suoi fattori primi dividono B;
b) **infinita periodica** se $x = \pm\alpha/\beta$, con α e β numeri interi primi fra loro, ma non come sopra, ossia esistono $n^* \geq 1$ e $k \geq 1$ tali che per ogni $i \geq n^*$ si ha che $a_{i+k} = a_i$. Sarà **periodica pura** ($n^* = 1$) se β e B non hanno fattori primi in comune (es: $B = 10, x = 1/3 = 0.\overline{3}$), altrimenti sarà **periodica mista** (es. $B = 10, x = 1/6 = 0.1\overline{6}$);
c) **infinita** non periodica se x è un numero irrazionale.

Per i numeri interi si usa anche la rappresentazione di tipo intero, data dal segno seguito dalle cifre che rappresentano, in una certa base B, il valore assoluto del numero, cioè $x = \pm(a_1 a_2 .....)_B$. Il più grande numero intero di n cifre in base B è è $B^n - 1$.

Se, ad esempio, nel calcolatore abbiamo a disposizione un massimo di 16 bit (cifre binarie $0, 1$) per la rappresentazione in base 2 di un numero intero, dedicando il primo bit alla rappresentazione del segno (ad es. ponendolo $= 0$ se positivo, $= 1$ se negativo) e i rimanenti al valore assoluto, possiamo rappresentare tutti gli interi compresi tra $-(2^{15} - 1)$ e $(2^{15} - 1)$.

DEFINIZIONE 1.2. *Siano B, t, m, M numeri interi con $B \geq 2, t \geq 1, m > 0, M > 0$. Si definisce insieme dei numeri di macchina (Floating Point) in base B con t cifre significative e range $= [-m, M]$ l' insieme:*

$$F(B,t,m,M) = \{0\} \cup \left\{ \begin{array}{c} x \in R : x = sign(x)B^p \sum_{i=1}^{t} a_i B^{-i}, \\ con\ 0 \leq a_i \leq B-1, a_1 \neq 0, -m \leq p \leq M \end{array} \right\}.$$

□

Dunque i numeri di $F(B,t,m,M)$ hanno la rappresentazione finita del tipo $x = \pm(.a_1 a_2 .....a_t)B^p$.

L' insieme $F(B,t,m,M)$ è composto da $2(B-1)B^{t-1}(M+m+1)+1$ numeri reali compresi tra $-\alpha B^M$ e αB^M dove $\alpha = 1 - B^{-t}$. Il più piccolo numero positivo è $.1B^{-m} = B^{-m-1}$.

La rappresentazione di un numero di macchina in un calcolatore richiede la disponibilità di adeguati campi di memoria per la caratteristica, per il segno e per le t cifre della mantissa. Generalmente per la caratteristica, che è un numero intero relativo, viene usata la rappresentazione per traslazione, in modo che lo zero corrisponda all' esponente $-m$. Per quanto riguarda la mantissa, osserviamo che nel caso più frequente in cui $B = 2$, dovendo essere $a_1 \neq 0$ avremo necessariamente che $a_1 = 1$, dunque non è necessario che questo dato venga memorizzato. Ad esempio se $B = 2$, $t = 24$. $m + M + 1 = 256$, possiamo schematizzare una rappresentazione interna del numero $\pm(.a_1 a_2 .....a_{24})B^p$ in una configurazione di 32 bit (cifre binarie $0, 1$) secondo la seguente distribuzione. Il primo gruppo di 8 bit (ogni gruppo di 8 bit è detto *byte*) contiene la caratteristica rappresentata (in base 2) per traslazione, dei rimanenti 24 bit, uno rappresenta il segno (ad es. 0 se positivo, 1 se negativo) e gli altri conterranno le cifre binarie $a_2, ..., a_{24}$. Questa è la rappresentazione dei numeri di $F(2, 24, 128, 127)$.

Consideriamo ora un numero reale qualsiasi avente la rappresentazione

$$x = sign(x)B^p \sum_{i=1}^{\infty} a_i B^{-i}.$$

Se $x \notin F(B,t,m,M)$, si presenta la necessità di approssimarlo tramite un opportuno numero macchina.

Se x è troppo piccolo in valore assoluto, cioè $p < -m$, allora viene segnalata la situazione di **underflow** ed il numero viene approssimato con zero.

Se $p > M$, il numero non viene rappresentato e viene segnalata la situazione di **overflow**.

Indichiamo con $\mathbb{R}(m, M)$ l 'insieme dei numeri reali per i quali $-m \leq p \leq M$. Allora, per $x \in \mathbb{R}(m, M)$ si possono avere i seguenti due tipi di approssimazione tramite un numero di $F(B,t,m,M)$:
1) **Troncamento** (Chopping). Si definisce

$$Chop(x) = sign(x)B^p \sum_{i=1}^{t} a_i B^{-i}.$$

2) **Arrotondamento** (Rounding). Sia $x > 0$. Si definisce

$$Round(x) = Chop(x + \frac{1}{2}B^{p-t}).$$

E' immediato verificare che se la base B è un numero pari allora si ha per $x > 0$:

$$\text{se } a_{t+1} < \frac{B}{2} : Round(x) = Chop(x),$$
$$\text{se } a_{t+1} \geq \frac{B}{2} : Round(x) = Chop(x) + B^{p-t}.$$

Nel caso generale ($x \gtrless 0$) si definisce:

$$Round(x) = sign(x)Round(|x|).$$

□

OSSERVAZIONE. Nell' effettuare l'arrotondamento può verificarsi l' overflow.

□

Valutazione dell' errore.

Ricordando che $a_1 \neq 0$ e che $0 \leq a_i \leq B-1$, per l' errore assoluto si ottiene

$$|x - Chop(x)| = B^p \sum_{i=t+1}^{\infty} a_i B^{-i} \leq B^p(B-1) \sum_{i=t+1}^{\infty} B^{-i}$$
$$= B^p(B-1)\frac{B^{-t-1}}{(1-B^{-1})} = B^{p-t}.$$

Per quanto riguarda l' errore relativo, poichè

$$|x| = B^p \sum_{i=1}^{\infty} a_i B^{-i} \geq B^{p-1},$$

avremo

$$\frac{|x - Chop(x)|}{|x|} \leq B^{1-t}.$$

Per il Rounding si ottiene invece:

$$\frac{|x - Round(x)|}{|x|} \leq \frac{B^{1-t}}{2}.$$

□

OSSERVAZIONE. Nel caso di underflow l'errore relativo è sempre 1.

□

In generale indichiamo con fl la generica funzione di approssimazione (ad esempio *Chop* o *Round*). Si definisce **precisione di macchina** :

$$u = \sup_{x \in \mathbb{R}(m,M)} \frac{|x - fl(x)|}{|x|}.$$

Quindi avremo $u = B^{1-t}$ per $fl = Chop$, $u = \frac{B^{1-t}}{2}$ per $fl = Round$.

In generale quindi considerata una funzione di approssimazione fl, posto

$$\varepsilon := \frac{fl(x) - x}{x},$$

avremo

$$fl(x) = x(1 + \varepsilon)$$
$$\text{con } |\varepsilon| \leq u.$$

La funzione fl comunemente considerata è il Rounding.

1.2 ARITMETICA di MACCHINA

Indichiamo ora con $*$ la generica operazione aritmetica $(+, -, \cdot, /)$.

Può accadere che, anche se gli operandi $x, y \in F(B, t, m, M)$, il risultato esatto dell' operazione cioè $(x * y)$ non appartenga a $F(B, t, m, M)$. Ossia $F(B, t, m, M)$ non è chiuso rispetto alle operazioni aritmetiche.

Noi supponiamo che l' operazione di macchina realmente eseguita, che indichiamo qui con $(*)$, *dia come risultato l' approssimazione di macchina del risultato vero* cioè

$$x(*)y = fl(x * y).$$

Dunque (escludendo underflow e overflow)

$$x(*)y = (x * y)(1 + \varepsilon), \text{ con } |\varepsilon| \leq u.$$

E' evidente che non valgono più le proprietà associativa per la somma né la distributiva per la moltiplicazione rispetto alla somma.

1.3 CONDIZIONAMENTO DELLE OPERAZIONI ELEMENTARI

Siano x e y numeri reali ed ϵ_x e ϵ_y errori relativi associati. Consideriamo dunque i dati perturbati

$$\widetilde{x} = x(1 + \varepsilon_x), \ \widetilde{y} = y(1 + \varepsilon_y)$$

che supponiamo essere dei numeri macchina.

Ci prefiggiamo ora lo scopo di valutare come le perturbazioni ϵ_x e ϵ_y, che supponiamo fin d' ora essere "piccole", possano o meno venire amplificate quando si esegue una generica operazione tra i dati in questione.

Per quanto convenuto, il risultato dell' operazione di macchina sui dati perturbati è

$$\widetilde{x}(*)\widetilde{y} = fl(\widetilde{x} * \widetilde{y}) = (\widetilde{x} * \widetilde{y})(1+\varepsilon), \text{ con } |\varepsilon| \leq u.$$

Diciamo E_{x*y} l' errore relativo di $\widetilde{x} * \widetilde{y}$ (*operazione esatta sui dati perturbati*) rispetto a $(x * y)(\neq 0)$ (*operazione esatta sui dati esatti*), cioè

$$\widetilde{x} * \widetilde{y} = (x * y)(1 + E_{x*y})$$

ossia

$$E_{x*y} = \frac{(\widetilde{x} * \widetilde{y}) - (x * y)}{x * y} = \frac{\{(x(1+\epsilon_x) * y(1+\epsilon_y)\} - (x * y)}{(x * y)}.$$

Indichiamo poi con $E_{x(*)y}$ l' errore relativo di $(\widetilde{x}(*)\widetilde{y})$(*operazione di macchina tra i dati perturbati*) rispetto al risultato esatto $(x * y)$, cioè

$$\begin{aligned} E_{x(*)y} &= \frac{(\widetilde{x}(*)\widetilde{y}) - (x * y)}{x * y} = \frac{(\widetilde{x} * \widetilde{y})(1+\varepsilon) - (x * y)}{(x * y)} \\ &= \frac{(x * y)(1 + E_{x*y})(1+\varepsilon) - (x * y)}{(x * y)} = (1 + E_{x*y})(1+\varepsilon) - 1. \end{aligned}$$

Dunque

$$E_{x(*)y} = E_{x*y} + \varepsilon + E_{x*y}\varepsilon.$$

Supponendo che ε_x , ε_y e u siano molto piccoli rispetto ad 1, l' errore relativo $E_{x(*)y}$ dipende essenzialmente da E_{x*y} ed è dunque questo che dobbiamo esaminare. Ricordando che

$$E_{x*y} = \frac{\{(x(1+\varepsilon_x) * y(1+\varepsilon_y)\} - (x * y)}{(x * y)},$$

vediamo dunque, per le varie operazioni, come E_{x*y} dipende da ε_x , ε_y.

Se, pur essendo ε_x , ε_y "piccoli", l' errore relativo E_{x*y} diventa "grande" diremo che l' operazione in questione, per i dati considerati, è **mal condizionata**. Altrimenti, se non vi è amplificazione degli errori iniziali ε_x , ε_y, essa si dirà **ben condizionata**.

Iniziamo considerando l' operazione di somma algebrica.

Avremo (se $x + y \neq 0$)

$$\begin{aligned} E_{x+y} &= \frac{\{(x(1+\varepsilon_x) + y(1+\varepsilon_y)\} - (x+y)}{(x+y)} \\ &= \frac{x}{x+y}\varepsilon_x + \frac{y}{x+y}\varepsilon_y. \end{aligned}$$

Questa relazione ci dice che il calcolo della somma fra due numeri di segno opposto può essere malcondizionato. In questo caso infatti gli errori ε_x , ε_y possono venire notevolmente amplificati tramite i due coefficienti $(\frac{x}{x+y})$ e $(\frac{y}{x+y})$, dando luogo al così detto fenomeno di "cancellazione" numerica. Si pensi al caso in cui $y \approx -x$ e ε_x , ε_y sono di segno opposto. In tutti gli altri casi è immediato vedere che il condizionamento è buono (non c'è amplificazione degli errori).

Per la moltiplicazione otteniamo (se $x, y \neq 0$)

$$\begin{aligned} E_{x\cdot y} &= \frac{\{(x(1+\varepsilon_x)\cdot y(1+\varepsilon_y)\} - (x\cdot y)}{(x\cdot y)} \\ &= \varepsilon_x + \varepsilon_y + \varepsilon_x \cdot \varepsilon_y. \end{aligned}$$

Dunque la moltiplicazione è sempre ben condizionata. Ciò vale anche per la divisione, infatti si ottiene

$$\begin{aligned} E_{x/y} &= \frac{\{(x(1+\varepsilon_x)/y(1+\varepsilon_y)\} - (x/y)}{(x/y)} \\ &= (\varepsilon_x - \varepsilon_y)/(1+\varepsilon_y) \approx (\varepsilon_x - \varepsilon_y). \end{aligned}$$

Il concetto di **condizionamento** si può estendere in generale a tutti i problemi rappresentati da una legge che associa dei risultati a dei dati. Si riferisce pertanto a tale legge ed ai dati considerati. Esso misura la possibilità che a "piccole" perturbazioni sui dati possano corrispondere "piccole" o "grandi" perturbazioni sui risultati corrispondenti. Nel primo caso il problema si dirà **ben condizionato** nel secondo **mal condizionato**.