CALCOLO DEGLI AUTOVALORI E FONDAMENTI DELLA MATEMATICA NUMERICA (Operazioni floating point)

In questa lezione si tratterà la rappresentazione dei numeri di macchina e la propagazione degli errori durante l'esecuzione di operazioni algebriche con un calcolatore, utilizzando l'aritmetica discreta. Successivamente, sarà approfondita l'aritmetica floating point, comunemente denominata aritmetica in virgola mobile.

Nella lezione precedente è stata illustrata la rappresentazione dei numeri di macchina nella seguente forma generale

$$x = \pm(0.\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t)_\beta \cdot \beta^e,$$

dove x rappresenta un generico numero, denominato numero macchina, composto da un segno $\pm$ (positivo o negativo), dalle cifre significative $\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t$, con t che indica il numero di cifre significative utilizzate per la rappresentazione del numero, dalla base β, assunta costante, e dall'esponente e, che è chiamato caratteristica. L'insieme delle cifre significative $\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t$ con $0 \leq \alpha k \leq \beta - 1$ e $\alpha_1 \neq 0$, costituisce la cosiddetta parte "decimale" del numero, denominata mantissa. L'esponente o caratteristica è compreso tra due estremi $L < 0$ e $U > 0$ (dove $L, U \in \mathbb{N}$), con $L \leq e \leq U$.

Pertanto, per la memorizzazione del numero macchina, il calcolatore utilizzerà una cella (o bit, intesa come unità elementare di memoria) per il segno $\pm$, t celle per le cifre significative e una cella aggiuntiva per l'esponente e.

La Figura 1 illustra l'idealizzazione della retta dei numeri reali rappresentabili dalla macchina. Come si osserva, in questo caso sono presenti lo zero e una zona circostante che non risulta sostanzialmente rappresentabile come numero macchina, poiché esiste un numero minimo positivo indicato come x_L e un numero minimo negativo pari a $-x_L$. Inoltre, sono definiti il numero massimo positivo x_U e il numero massimo negativo $-x_U$. Tra questi estremi si trovano i numeri, rappresentati schematicamente nella Figura 1 mediante dei trattini, ciascuno dei quali identifica un numero macchina. I numeri macchina tendono a concentrarsi nelle vicinanze dello zero, mentre si distribuiscono più distanti man mano che ci si allontana dallo zero verso gli estremi. La regione al di fuori di questi limiti, ovvero per $x < -x_U$ e $x > x_U$ è denominata overflow. Viceversa, la regione attorno allo zero, $(-x_L, 0) \cup (0, x_L)$, è chiamata regione di underflow.

Una questione rilevante da considerare a questo punto consiste nel determinare quanti numeri una macchina sia in grado di rappresentare. Tale risposta dipende, naturalmente, dalle caratteristiche intrinseche della macchina, ossia dalla base utilizzata per la rappresentazione, dagli estremi L e U, nonché dal numero t di cifre significative α_k.

Indichiamo con $\mathbb{F}$ l'insieme dei numeri macchina

$$\mathbb{F} = \text{insieme dei numeri macchina.}$$

Ovviamente, $\mathbb{F}$ è un sottoinsieme dei numeri reali ed è costituito dallo zero unito a tutti i numeri macchina descritti in precedenza, che assumono la forma $x = \pm(0.\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t)_\beta \cdot \beta^e$, ovvero

$$\mathbb{F} = \{0\} \cup \{x = \pm(0.\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t)_\beta \cdot \beta^e, 0 \leq \alpha_k \leq \beta - 1, k = 2, \dots, t, 0 < \alpha_1 \leq \beta - 1, L \leq e \leq U\}.$$

Se intendiamo contare il numero totale di numeri macchina rappresentabili, osserviamo che:

- Vi sono 2 possibili valori per il segno $\pm$ (positivo o negativo);

- La prima cifra significativa α_1 può assumere $\beta - 1$ valori, essendo α_1 compreso tra 1 e $\beta - 1$;
- Ciascuna delle restanti cifre significative α_k, con $k = 2, \dots, t$, può assumere β valori, da 0 a $\beta - 1$, quindi in totale abbiamo β^{t-1} combinazioni;
- L'esponente e può variare tra L e U, estremi inclusi, per un totale di $U - L + 1$ valori.

Combinando tutte queste possibilità, il numero totale N di numeri macchina rappresentabili risulta essere una funzione che dipende da t (numero di cifre significative), dalla base β e dagli estremi U e L dell'esponente e

$$N = N(t, \beta, U, L) = 2(\beta - 1)\beta^{t-1}(U - L + 1) + 1,$$

dove

- il fattore 2 tiene conto dei due possibili segni;
- $(\beta - 1)$ rappresenta le scelte possibili per α_1;
- β^{t-1} corrisponde alle combinazioni possibili per le cifre $\alpha_2, \dots, \alpha_t$;
- $(U - L + 1)$ rappresenta il numero di esponenti possibili;
- il termine finale $+1$ include la rappresentazione dello zero.

Questo numero N rappresenta, dunque, la quantità totale di numeri reali che una macchina è in grado di rappresentare, in base alle proprie caratteristiche intrinseche. Per curiosità, è possibile esaminare le specifiche tecniche di alcuni calcolatori attualmente in uso e determinare quanti numeri macchina essi siano effettivamente in grado di rappresentare.
La Tabella 1 riporta alcune informazioni significative relative a quattro differenti tipologie di calcolatori, ciascuno conforme a uno specifico standard di rappresentazione in virgola mobile. Per ogni calcolatore sono indicati: la base β utilizzata nella rappresentazione (in tutti i casi pari a 2); due modalità di rappresentazione numerica (precisione singola e precisione doppia); il numero di cifre significative t, ovvero la precisione con cui i numeri sono rappresentati (e come si osserva, nella modalità a doppia precisione il valore di t è generalmente superiore, costituendo la principale differenza tra singola e doppia precisione); l'intervallo complessivo degli esponenti possibili $U - L + 1$, cioè la portata della scala esponenziale utilizzabile dal calcolatore. Questi parametri determinano il numero totale di numeri macchina rappresentabili e, più in generale, le capacità numeriche della macchina in termini di precisione e intervallo dinamico.

Nome calcolatore	**β (base)**	**precisione**	**t**	**$U - L + 1$**
IEEE 754 binary16	2	Singola	11	30
IEEE 754 binary32	2	Singola	24	254
IEEE 754 binary64	2	Doppia	53	2046
IEEE 754 binary128	2	Doppia	113	32766

- IEEE 754 binary16 (precisione singola): utilizza 16 celle (o bit) totali, con 1 bit per il segno, 5 bit per l'esponente e 10 bit per la mantissa. Grazie al bit implicito (α_1), la precisione effettiva è di 11 bit. L'esponente può variare da -14 a +15, per un totale di 30 esponenti validi.

- IEEE 754 binary32 (precisione singola): utilizza 32 bit totali, con 1 bit per il segno, 8 bit per l'esponente e 23 bit per la mantissa. La precisione effettiva è di 24 bit. L'esponente può variare da -126 a +127, per un totale di 254 esponenti validi.

- IEEE 754 binary64 (precisione doppia): utilizza 64 bit totali, con 1 bit per il segno, 11 bit per l'esponente e 52 bit per la mantissa. Con il bit implicito, la precisione effettiva è di 53 bit. L'esponente varia da -1022 a +1023, per un totale di 2046 esponenti validi.

- IEEE 754 binary128 (precisione doppia): utilizza 128 bit totali, con 1 bit per il segno, 15 bit per l'esponente e 112 bit per la mantissa. Con il bit implicito, la precisione effettiva è di 113 bit. L'esponente varia da -16382 a +16383, escludendo i valori riservati, per un totale di 32766 esponenti validi.

La Tabella 2 riporta la cardinalità di $\mathbb{F}$, ossia il numero totale di numeri macchina che ciascun calcolatore, riportato in Tabella 1, è in grado di rappresentare.

Nome calcolatore	**precisione**	**Numero totale di numeri macchina N**
IEEE 754 binary16	Singola	$2 \cdot (2-1) \cdot 2^{10} \cdot 30 + 1 \approx 61.4 \cdot 10^3$
IEEE 754 binary32	Singola	$2 \cdot (2-1) \cdot 2^{23} \cdot 254 + 1 \approx 42.6 \cdot 10^9$
IEEE 754 binary64	Doppia	$2 \cdot (2-1) \cdot 2^{52} \cdot 2046 + 1 \approx 7.58 \cdot 10^{18}$
IEEE 754 binary128	Doppia	$2 \cdot (2-1) \cdot 2^{112} \cdot 32766 + 1 \approx 1.51 \cdot 10^{37}$

La seconda colonna della Tabella 2 riporta il risultato della formula

$$N = 2(\beta - 1)\beta^{t-1}(U - L + 1) + 1,$$

in funzione dei parametri indicati nella Tabella 1. In questo modo si ottiene il numero di numeri macchina che ciascun calcolatore può rappresentare. Come si osserva, nella modalità a doppia precisione il numero di valori rappresentabili aumenta significativamente, poiché cresce il parametro t, corrispondente al numero di cifre significative. Pertanto, si riscontra un incremento considerevole dei numeri macchina rappresentabili nella modalità a doppia precisione rispetto a quella a singola precisione.

È possibile osservare, a titolo esemplificativo, un caso puramente accademico (non reale) che consente di effettuare calcoli molto semplici. Se si considera

$$\beta = 2, \qquad t = 3, \qquad L = -2, \qquad U = 3,$$

ovvero, considerando una rappresentazione binaria con sole tre cifre significative e con $L = -2$ e $U = 3$ come estremi possibili per l'esponente, è possibile verificare che il numero totale di numeri macchina rappresentabili utilizzando tali parametri è pari a

$$N = 2 \cdot (2-1) \cdot 2^{2-1} \cdot (3 + 2 + 1) + 1 = 49$$

Abbiamo analizzato quanti numeri una macchina è in grado di rappresentare. Ora ci proponiamo di comprendere quale sia l'errore introdotto dalla macchina nella rappresentazione di un numero reale quando essa è costretta ad arrotondarlo. Questo tipo di errore è noto come errore di round-off o errore di arrotondamento. Ci concentreremo, dunque, sugli errori di arrotondamento associati alla rappresentazione in formato macchina.

Consideriamo un numero reale x in base β

$$x = \pm(0.\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t\alpha_{t+1} \dots)_\beta \cdot \beta^e,$$

dove, a priori, il numero presenta un'infinità di cifre significative, come indicato dai puntini dopo la cifra α_{t+1}. Quando tale numero viene rappresentato in macchina, si ottiene la cosiddetta rappresentazione in virgola mobile (floating point)

$$\mathrm{fl}^t(x) = \pm(0, \alpha_1\alpha_2 \dots \alpha_{t-1}\widetilde{\alpha_t})_\beta \cdot \beta^e,$$

la quale utilizza esattamente t cifre significative.
Come si può osservare, il segno $\pm$ viene mantenuto, e le prime $t-1$ cifre significative $\alpha_1, \alpha_2, \dots, \alpha_{t-1}$ rimangono invariate, ossia sono esattamente corrette. La differenza può sorgere nell'ultima cifra significativa, indicata con $\widetilde{\alpha_t}$, che rappresenta la t-esima cifra nella rappresentazione in macchina, e che, a priori, può differire dalla corrispondente cifra α_t del numero reale originale x. Infine, entrambe le rappresentazioni, x e $\mathrm{fl}^t(x)$, sono espresse nella stessa base β, e l'esponente e rimane invariato. Quindi, possiamo concludere che

$$\mathrm{fl}^t(x) = \pm(0, \alpha_1\alpha_2 \dots \alpha_{t-1}\widetilde{\alpha_t})_\beta \cdot \beta^e,$$

è la rappresentazione di macchina del numero x

$$x = \pm(0.\,\alpha_1\alpha_2 \dots \alpha_k \dots \alpha_{t-1}\alpha_t\alpha_{t+1} \dots)_\beta \cdot \beta^e.$$

La differenza tra x e la sua rappresentazione di macchina floating $\mathrm{fl}^t(x)$ è

$$x - \mathrm{fl}^t(x) = \pm(0.00 \dots 0\alpha_{t+1} \dots)_\beta \cdot \beta^e,$$

se

$$\widetilde{\alpha_t} = \alpha_t, \qquad (\text{con } \alpha_{t+1} < \beta/2),$$

nel caso in cui $\widetilde{\alpha_t} = \alpha_t$ e $\alpha_{t+1} < \beta/2$, ovvero una situazione che si verifica ogniqualvolta la cifra α_{t+1} è strettamente minore di $\beta/2$. In tale circostanza, la differenza $x - \mathrm{fl}^t(x)$ rappresenta il residuo della troncatura, ed è costituita da t cifre decimali (in base β) dopo la virgola tutte pari a zero, seguite da cifre non nulle, a condizione che il numero reale di partenza x presenti ulteriori cifre significative diverse da zero dopo la t-esima. Questo tipo di situazione descrive esattamente ciò che accade quando l'arrotondamento coincide con un troncamento, ossia il semplice taglio della rappresentazione dopo la t-esima cifra significativa, senza incremento della cifra finale.

Se invece l'arrotondamento viene effettuato incrementando di un'unità la t-esima cifra significativa, la differenza tra il numero reale x e la sua rappresentazione in macchina $\mathrm{fl}^t(x)$ risulta

$$x - \mathrm{fl}^t(x) = \pm(0.00 \dots 0(\alpha_t + 1) - 0.00 \dots 0\alpha_t\alpha_{t+1} \dots)_\beta \cdot \beta^e,$$

nel caso in cui $\widetilde{\alpha_t} = \alpha_t + 1$ e

$$\beta/2 \le \alpha_{t+1} < \beta,$$

cioè, quando la $t+1$-esima cifra del numero di partenza x sia compresa fra $\beta/2$ e β. Questa è la situazione che si verifica quando l'arrotondamento avviene per eccesso, aggiungendo uno alla t-esima cifra, poiché la cifra successiva (α_{t+1}) è sufficientemente grande da giustificare l'incremento secondo le regole convenzionali dell'arrotondamento.
A questo punto, ci si può porre il problema di stimare la differenza $x - \mathrm{fl}^t(x)$, ossia di valutare, nei due casi considerati, una maggiorazione del valore assoluto di tale differenza. In altri termini, l'obiettivo è quello di determinare una stima dell'errore assoluto che si commette nella rappresentazione di un numero reale tramite la sua approssimazione in macchina. Questa stima consente di quantificare il massimo scostamento possibile tra il valore reale x e la sua rappresentazione in floating point $\mathrm{fl}^t(x)$, fornendo quindi un limite superiore all'errore di arrotondamento introdotto dal sistema di calcolo.

È un esercizio semplice verificare che il valore assoluto di questa differenza è

$$|x - \mathrm{fl}^t(x)| \leq \left(0.00 \ldots 0\frac{\beta}{2}\right)_\beta \cdot \beta^e = \frac{1}{2}\beta^{e-t}, \qquad \alpha_{t+1} = \frac{\beta}{2},$$

ovvero, un numero che presenta t cifre significative tutte uguali a zero, mentre la $(t+1)$-esima cifra è pari a $\alpha_{t+1} = \frac{\beta}{2}$. Tale numero non è altro che $\frac{1}{2}\beta^{e-t}$. Questa è un'indicazione significativa, poiché ci permette di comprendere che l'errore assoluto commesso è pari a

$$|x - \mathrm{fl}^t(x)| \leq \frac{1}{2}\beta^{e-t}.$$

Ciò evidenzia come l'errore dipenda dalla base β e abbia come esponente la differenza $e - t$. In particolare, si noti che l'esponente include t, ovvero il numero di cifre significative.
È possibile valutare anche l'errore relativo, considerando che, molto spesso, quest'ultimo risulta più significativo rispetto all'errore assoluto. Infatti, l'errore relativo tiene conto dell'ordine di grandezza del numero di partenza, mentre l'errore assoluto ne è indipendente.

L'errore relativo può essere calcolato nel modo seguente. Sia x il numero di partenza

$$x = \pm(0.\alpha_1\alpha_2 \ldots \alpha_t\alpha_{t+1} \ldots)_\beta \cdot \beta^e,$$

possiamo osservare che il numero x, in valore assoluto, è certamente maggiore di $(0.100 \ldots)_\beta \cdot \beta^e$. Infatti, per quanto piccolo possa essere il numero di partenza x, la cifra α_1 sarà almeno pari a 1, il che implica

$$|x| \geq (0.100 \ldots)_\beta \cdot \beta^e = \beta^{e-1} \Longrightarrow \frac{1}{|x|} \leq \beta^{1-e}.$$

Quindi, il valore assoluto di x è maggiore di β^{e-1}. Prendendo i reciproci, troveremo che

$$\frac{1}{|x|} \leq \beta^{1-e}.$$

Pertanto, se andiamo a combinare questa relazione $\frac{1}{|x|} \leq \beta^{1-e}$ con quella precedente $|x - \mathrm{fl}^t(x)| \leq \frac{1}{2}\beta^{e-t}$, troviamo che $x - \mathrm{fl}^t(x)$, in valore assoluto, diviso per il valore assoluto di x, ovvero l'errore relativo nella rappresentazione di macchina, è

$$\frac{|x - \mathrm{fl}^t(x)|}{|x|} \leq \frac{1}{2}\beta^{e-t} \cdot \beta^{1-e} = \frac{1}{2}\beta^{1-t}.$$

Questa stima assume un'importanza notevole, poiché indica che, ogni qualvolta rappresentiamo un numero mediante il calcolatore, l'errore relativo commesso non può eccedere il valore $\frac{1}{2}\beta^{1-t}$

$$\frac{|x - \mathrm{fl}^t(x)|}{|x|} \leq \frac{1}{2}\beta^{1-t}.$$

Nella stima presentata, vi sono due ingredienti fondamentali: la base β e il numero di cifre significative t. Poiché t compare all'esponente con segno negativo, risulta evidente che l'errore relativo commesso diminuisce al crescere di t. Pertanto, è evidente che l'errore relativo che si commette sarà tanto più piccolo quanto più grande sarà t. Dunque, ovviamente, più grande è il numero di cifre significativi che un calcolatore può rappresentare, più piccolo sarà l'errore relativo che si origina quando si rappresenta un qualunque numero reale.
Il numero che abbiamo individuato, il quale dipende esclusivamente da β e t, possiede un significato ben preciso e viene definito come precisione di macchina o zero di macchina

$$u = \frac{1}{2}\beta^{1-t}, \qquad \text{precisione di macchina (o zero di macchina),}$$

dove, u indica la precisione di macchina.
Abbiamo quindi analizzato l'entità dell'errore che si introduce nella rappresentazione di un numero all'interno di un calcolatore. A questo punto, intendiamo esaminare come il calcolatore esegue operazioni sui numeri di macchina e quali errori si manifestano durante tali operazioni.
Passiamo ora a trattare le operazioni di macchina e la propagazione dei corrispondenti errori di arrotondamento. Iniziamo osservando che, se consideriamo la retta che rappresenta schematicamente l'insieme dei numeri macchina (si veda la Figura 1), e prendiamo due numeri macchina $x \in \mathbb{F}$ e $y \in \mathbb{F}$, il risultato della loro somma, differenza, prodotto o quoziente non è necessariamente un numero appartenente ancora a $\mathbb{F}$. Questo comportamento si discosta da quanto accade nel caso dei numeri reali: infatti, eseguendo operazioni lecite in $\mathbb{R}$, si ottiene sempre un altro numero reale.
Dunque, componendo due numeri $x, y \in \mathbb{F}$ mediante una delle operazioni algebriche, è possibile ottenere come risultato un numero di macchina $z \in \mathbb{F}$; tuttavia, è altrettanto possibile che il risultato sia un numero reale $z \in \mathbb{R}\backslash\mathbb{F}$, ossia un numero reale che non appartiene all'insieme dei numeri macchina

$$\mathrm{z} \in \mathbb{F},$$
$$z \in \mathbb{R}\backslash\mathbb{F}.$$

Nel caso in cui $z \in \mathbb{R}\backslash\mathbb{F}$, si possono presentare tre possibili situazioni distinte

1. overflow: $|z| > x_U$,
2. underflow: $|z| < x_L$,
3. $x_L < |z| < x_U$, $z \in \mathbb{R}/\mathbb{F}$.

Nel primo caso, il numero ottenuto risulta essere troppo grande, ovvero ha modulo maggiore di x_U quindi, in questo primo caso parleremo di overflow. Ciò significa che il valore z calcolato appartiene a una delle due regioni non rappresentabili dai numeri macchina: $z > x_U$ e $z < -x_U$.
Il secondo caso, che può essere considerato duale rispetto al primo, si verifica quando il numero z ottenuto è troppo piccolo, ossia ha modulo minore di x_L. In questa situazione si parla di underflow. Di conseguenza, il valore z si colloca nella regione $z \in (-x_L, 0) \cup (0, x_L)$. Si tratta quindi di un numero estremamente vicino allo zero, che tuttavia la macchina non è in grado di rappresentare.
Infine, l'ultimo caso si verifica quando il numero ottenuto ha un modulo compreso tra x_L e x_U, ossia appartiene alla regione $z \in (-x_U, -x_L) \cup (x_L, x_U)$, ma non è un numero macchina, cioè non appartiene a $\mathbb{F}$. In questo caso, il valore z è un numero reale, tuttavia non è rappresentabile dal calcolatore.
Esaminiamo ora alcuni esempi, al fine di comprendere, attraverso casi molto semplici, come tali situazioni possano effettivamente verificarsi. In particolare, intendiamo mostrare come la somma o il prodotto di due numeri macchina possa non risultare in un numero ancora appartenente all'insieme dei numeri macchina.
Consideriamo il seguente esempio

$$t = 3, \qquad \beta = 10, \qquad L = -50, \qquad U = 50,$$

e supponiamo di prendere i numeri x e y

$$x = (0.235)_{10}, \qquad y = (0.900)_{10}.$$

Calcoliamo la somma $x + y$

$$x + y = (1.135)_{10} = (0.1135)_{10} \cdot 10^1 \notin \mathbb{F},$$

ovvero, la somma risulta $x + y = 1.135$ in base decimale. La sua rappresentazione in formato macchina richiede una normalizzazione, ottenendo quindi il numero è $(0.1135)_{10} \cdot 10^1$. Tuttavia, disponendo di sole tre cifre significative, la quarta cifra non può essere rappresentata. Di conseguenza, il numero

ottenuto non appartiene all'insieme dei numeri macchina. Esso dovrà pertanto essere approssimato dal calcolatore mediante un'operazione di round off. Questo costituisce un esempio molto semplice che mostra come, anche a partire dalla somma di due numeri macchina, si possa facilmente ottenere un numero che non è più rappresentabile esattamente come numero macchina.
Un altro esempio è il seguente

$$t = 3, \qquad \beta = 10, \qquad L = -50, \qquad U = 50.$$

Se consideriamo i numeri x e y

$$x = (0.235)_{10} \cdot 10^{40}, \qquad y = (0.200) \cdot 10^{20},$$

e ne facciamo il prodotto $x \cdot y$

$$x \cdot y = (0.470)_{10} \cdot 10^{59}, \qquad \text{overflow},$$

in aritmetica esatta troviamo $(0.470)_{10} \cdot 10^{59}$. Poiché $U = 50$ rappresenta il massimo esponente positivo consentito, il numero ottenuto risulta essere un numero reale che non è più rappresentabile come numero macchina. Ci troviamo, dunque, nella regione di overflow.
I semplici esempi finora esaminati non rappresentano casi eccezionali. In effetti, comportamenti di questo tipo costituiscono quasi la norma nell'aritmetica in virgola mobile. È quindi necessario comprendere quale impatto abbia questa situazione, apparentemente patologica, sulle operazioni effettuate con i numeri macchina. Analizzeremo pertanto come il calcolatore gestisca le operazioni tra numeri macchina e come si comporti per riportare il risultato in un numero macchina, anche quando il risultato originario dell'operazione non appartenga più all'insieme dei numeri macchina.
Introduciamo ora le operazioni di macchina: addizione, sottrazione, prodotto e divisione, e analizziamo lo schema generale che il calcolatore adotta nell'eseguirle. Siano $x, y \in \mathbb{R}$ due numeri reali, le cui rappresentazioni macchina sono rispettivamente $x_M, y_M \in \mathbb{F}$

$$x \to x_M = \mathrm{fl}^t(x),$$
$$y \to y_M = \mathrm{fl}^t(y).$$

Consideriamo la somma

$$x_M + y_M = z_M \in \mathbb{R}.$$

Otteniamo un numero z_M che, a priori, è un numero reale $z_M \in \mathbb{R}$, ma non necessariamente un numero macchina, cioè $z_M \notin \mathbb{F}$. Per riportarlo nell'insieme ammissibile dei numeri macchina, è pertanto necessario applicare un'operazione di floating point, ossia effettuare la rappresentazione macchina di z_M

$$z_M \to \mathrm{fl}^t(z_M) \in \mathbb{F}.$$

A questo punto, si ottiene un numero macchina. In sintesi, l'operazione eseguita è definita come *addizione macchina*, indicata con il simbolo $\oplus$

$$x \oplus y, \qquad \text{addizione macchina},$$

dove si utilizza l'operatore $\oplus$ per distinguere questa addizione dall'addizione classica tra numeri reali, indicata con il simbolo $+$.

Dunque, possiamo rappresentare l'operazione l'addizione di macchina in questo modo

$$x \oplus y = \mathrm{fl}^t[\mathrm{fl}^t(x) + \mathrm{fl}^t(y)],$$

dove prendiamo due numeri $x, y \in \mathbb{R}$, ne facciamo le loro rappresentazioni di macchina $\mathrm{fl}^t(x)$, $\mathrm{fl}^t(y)$, facciamo l'operazione di addizione classica reale $\mathrm{fl}^t(x) + \mathrm{fl}^t(y)$ e troviamo un numero reale, a priori

diverso dal numero di macchine, ed infine facciamo il floating di $\mathrm{fl}^t(x) + \mathrm{fl}^t(y)$. Pertanto, in definitiva troviamo il numero macchina

$$\mathrm{fl}^t[\mathrm{fl}^t(x) + \mathrm{fl}^t(y)] \in \mathbb{F}.$$

L'operazione di floating point applicata alla somma dei valori floating point di x e y corrisponde alla somma, eseguita dal calcolatore, di due numeri reali approssimati in formato macchina.
È evidente che questo paradigma può essere esteso anche alle altre operazioni di macchina, quali il prodotto, la divisione e la sottrazione. Pertanto, esamineremo ora il procedimento da adottare per definire correttamente le operazioni algebriche in macchina.
Abbiamo quindi il prodotto di macchina definito come

$$x \otimes y = \mathrm{fl}^t[\mathrm{fl}^t(x) \times \mathrm{fl}^t(y)],$$

dove per il prodotto di macchina utilizziamo la notazione $\otimes$. Anche in questo caso viene calcolato il floating di x per il floating di y, attraverso la classica operazione di prodotto reale. Questo numero dovrà essere poi tradotto in numero macchina, attraverso l'operazione di arrotondamento floating del risultato $\mathrm{fl}^t(x) \times \mathrm{fl}^t(y)$.
Adottando esattamente lo stesso paradigma e modificando unicamente il significato dell'operazione per sottrazione e divisione, otteniamo

$$x \ominus y = \mathrm{fl}^t[\mathrm{fl}^t(x) - \mathrm{fl}^t(y)],$$
$$x \oslash y = \mathrm{fl}^t[\mathrm{fl}^t(x) \,/\, \mathrm{fl}^t(y)],$$

che definiscono l'operazione di sottrazione di macchina $\ominus$ e l'operazione di divisione di macchina $\oslash$.
Il problema, a questo punto, consiste nel comprendere come tali operazioni di macchina propaghino gli errori di arrotondamento che inevitabilmente vengono introdotti nella rappresentazione dei numeri macchina. In particolare, avendo errori associati agli operandi x e y, è importante analizzare come tali errori si trasmettano e si amplifichino a seguito delle operazioni di macchina.
Per un'operazione generica, analizziamo come calcolare, o più precisamente stimare, quella che viene definita la sensitività delle operazioni $\oplus$, $\ominus$, $\otimes$, $\oslash$ rispetto agli errori di arrotondamento. In particolare, consideriamo un numero reale $x \in \mathbb{R}$ che viene rappresentato con un errore di rappresentazione indicato da $\epsilon(x)$, definito come segue

$$x \in \mathbb{R} \to \epsilon(x) = \frac{|x - \mathrm{fl}^t(x)|}{|x|}.$$

Similmente, $\epsilon(y)$ rappresenterà l'errore relativo alla rappresentazione di y con un numero macchina

$$y \in \mathbb{R} \to \epsilon(y) = \frac{|y - \mathrm{fl}^t(y)|}{|y|}.$$

Possiamo definire l'errore commesso nell'operazione di addizione macchina, indicato come $\epsilon(x \oplus y)$, mediante la seguente espressione:

$$\epsilon(x \oplus y) = \frac{|(x + y) - (x \oplus y)|}{|x + y|},$$

ossia la differenza, in valore assoluto, tra il risultato dell'addizione reale e quello dell'addizione macchina, normalizzata rispetto al valore assoluto del risultato reale. Tale quantità rappresenta l'errore relativo introdotto nel sostituire all'operazione di addizione reale l'operazione di addizione macchina.
Il problema che ci poniamo ora consiste nel determinare come rappresentare questo errore relativo sull'operazione $\oplus$ in funzione degli errori relativi associati a x e a y. In particolare, ci chiediamo se sia possibile esprimere $\epsilon(x \oplus y)$ in termini di $\epsilon(x)$ e $\epsilon(y)$.
La questione che si presenta riguarda il condizionamento dell'operazione di macchina, concetto già introdotto nelle lezioni precedenti. In particolare, ci si chiede se esista una costante C_1 tale che, per ogni

$x, y \in \mathbb{R}$, l'errore relativo sull'operazione di somma sia limitato superiormente da C_1 moltiplicato per il massimo degli errori relativi sugli operandi x e y, ovvero

$$\epsilon(x \oplus y) \leq C_1 \epsilon_{\max}(x, y),$$

dove $\epsilon_{\max}(x, y)$ è il massimo fra $\epsilon(x)$ e $\epsilon(y)$

$$\epsilon_{\max}(x, y) = \max[\epsilon(x), \epsilon(y)].$$

Analogamente, si può indagare l'esistenza di costanti C_2, C_3, C_4 tali da consentire una stima analoga per l'errore relativo associato rispettivamente al prodotto, alla sottrazione e alla divisione

$$\epsilon(x \otimes y) \leq C_2 \epsilon_{\max}(x, y),$$
$$\epsilon(x \ominus y) \leq C_3 \epsilon_{\max}(x, y),$$
$$\epsilon(x \oslash y) \leq C_4 \epsilon_{\max}(x, y).$$

Questa rappresenta l'idea generale alla base del concetto di numero di condizionamento di un'operazione. In altri termini, stiamo verificando se a piccoli errori sugli operandi x e y corrispondano piccoli errori sulle operazioni di macchina $\oplus, \ominus, \otimes, \oslash$.

Un'operazione di macchina si definisce ben condizionata rispetto all'operazione di round off (arrotondamento) se a piccoli errori di arrotondamento sugli operandi x e y corrispondono piccoli errori sul risultato dell'operazione stessa. Ricordiamo che, per ciascun operando x e y, l'errore di round off commesso è al massimo pari a u, dove u rappresenta la precisione di macchina. Quando tale condizione è soddisfatta, si afferma che l'operazione di macchina è ben condizionata. È possibile verificare che questo avviene nella maggior parte dei casi. In particolare, la condizione è verificata per le operazioni di moltiplicazione $\otimes$ e divisione $\oslash$. Lo stesso vale per le operazioni di addizione $\oplus$ (e sottrazione $\ominus$) tra due numeri che abbiano lo stesso segno (o segno opposto, rispettivamente). Di conseguenza, nel caso della moltiplicazione o della divisione tra due numeri macchina, l'errore commesso rimane contenuto e proporzionale agli errori sugli operandi. Analogamente accade per la somma di due numeri con segno uguale, o per la differenza tra due numeri con segno opposto.

Tuttavia, tale condizione non è sempre verificata nel caso della sottrazione tra due numeri macchina che abbiano lo stesso segno e valori assoluti approssimativamente uguali. Esempi di questa situazione erano già stati presentati nella lezione precedente. In tali circostanze, l'operazione si definisce mal condizionata. Pertanto, la sottrazione tra due numeri di valore assoluto simile ma di segno opposto può generare errori molto significativi nel calcolatore. Questo fenomeno, caratterizzato da una grave perdita di accuratezza durante la fase di propagazione dell'errore, è noto come

$$\text{loss of significance error},$$

o anche errore dovuto alla perdita di significato.

Come discusso nella lezione precedente, in tali casi non esistono soluzioni efficaci: questo rappresenta un limite intrinseco di ogni calcolatore. Il rimedio consiste, ove possibile, nell'evitare la sottrazione tra due numeri di valore assoluto approssimativamente uguale e di segno identico, ossia la sottrazione tra due numeri quasi coincidenti.

Esaminiamo un esempio molto semplice, relativo alla risoluzione di un'equazione di secondo grado

$$x^2 - 26x + 1 = 0,$$

la cui prima radice x_1 è uguale a

$$x_1 = 13 + \sqrt{168} = a + b,$$

dove abbiamo definito $a = 13$ e $b = \sqrt{168}$. La seconda radice è

$$x_2 = 13 - \sqrt{168} = a - b.$$

Se lavorassimo con $t = 5$ cifre significative, avremo che

$$a = 13, \qquad b \simeq 12.961,$$

ovvero a è rappresentato correttamente e b è approssimato al valore $b \simeq 12.961$ perché stiamo approssimando la radice quadrata. Dunque, in questo caso, per calcolare x_2 dovremo effettuare la differenza

$$x_2 \simeq 13 - 12.961.$$

In tal caso, potremmo incorrere in un errore significativo nell'operazione di sottrazione. Per evitare questa fase critica, è preferibile riscrivere l'espressione nel seguente modo

$$x_2 = \frac{x_2 x_1}{x_1} = \frac{169 - 168}{13 + \sqrt{168}} \simeq \frac{1}{25.961},$$

Così facendo, non si modifica la sostanza del problema, poiché abbiamo semplicemente moltiplicato e diviso x_2 per lo stesso valore x_1. Tale operazione produce un numeratore pari a 1, che non genera alcun problema, mentre al denominatore si ottiene $13 + \sqrt{168}$, ossia la somma di due numeri dello stesso segno, operazione che non presenta criticità. Questa procedura è quindi ben condizionata e il risultato approssimato è circa $\frac{1}{25.961}$.

Per riassumere quanto esposto finora riguardo ai numeri macchina, abbiamo osservato che l'errore relativo massimo commesso nella rappresentazione di un numero è pari a u, dove $u = \frac{1}{2}\beta^{1-t}$, essendo t il numero di cifre significative. Tale risultato giustifica anche l'utilizzo della rappresentazione in doppia precisione, in quanto l'incremento di t comporta una significativa riduzione dell'errore, come illustrato in alcuni esempi precedenti. Successivamente, abbiamo analizzato la definizione delle operazioni di macchina e la propagazione degli errori ad esse associati. Si è evidenziato che tale propagazione non costituisce generalmente un problema per la maggior parte delle operazioni, ad eccezione del caso in cui si sottraggano numeri di valore approssimativamente uguale. In tale situazione, si possono verificare errori di arrotondamento di entità rilevante. Per tale motivo, è preferibile evitare queste operazioni e adottare algoritmi che consentano di sostituirle con operazioni più sicure e meglio condizionate.