

Matematica Numerica 1

Laurea Triennale in Matematica

Valeria Simoncini

Dipartimento di Matematica, UniBo

`valeria.simoncini@unibo.it`

a.a. 2026-2027

Aritmetica in precisione finita. Esempio di cancellazione.

Calcolo al computer di $f(x) = \frac{1-\cos(x)}{x^2}$ in $x_* = 1.2 \cdot 10^{-5}$

$$\cos(x_*) = 0.9999\,9999\,99 \text{ (troncato a 10 decimali)}$$

$$1 - \cos(x_*) = 0.0000\,0000\,01$$

Quantità che è dell'ordine di grandezza dell'errore!

E si ha:

$$\frac{1-\cos(x_*)}{x_*^2} = \frac{10^{-10}}{1.44 \cdot 10^{-10}} = 0.6944 > \frac{1}{2}$$

★ Non c'è neppure una cifra significativa corretta

⇒ La **sottrazione aggrava** l'importanza di errori precedenti

Strategia risolutiva: Scrivere f come $f(x_*) = \frac{1}{2} \left(\frac{\sin(\frac{x_*}{2})}{\frac{x_*}{2}} \right)^2 = 0.5$

Rappresentazione dei numeri su calcolatore. Cenni

★ I numeri reali con un numero *infinito* di cifre decimali possono essere **approssimati** da un numero con rappresentazione finita.

$$0.001 \quad \Rightarrow \quad 1 \cdot 10^{-3}$$

Rappresentazione in virgola mobile (floating point, $\mathbb{F}$).

$$x_{(2)} = (-1)^s (a_0 \cdot \underbrace{a_1 a_2 \cdots a_t}_{m=\text{mantissa}}) 2^E, \quad a_0 \equiv 1$$

$a_i \in \{0, 1\}$, a_0 prima cifra significativa. $t = n$. cifre significative,
 $E \in [E_{\min}, E_{\max}]$

Esempio: $(10)_{(2)} = (1010)_2 \Rightarrow (1.)010 \cdot 2^3$, $E = 3$, $m = 010$

Esempio: $1/10 = (0.000\overline{1100})_2 \Rightarrow (1.)100\overline{1100} \cdot 2^{-4}$, $E = -4$, $m = 100\overline{1100}$

tre campi: segno ($0 = +$), esponente E , mantissa m

$m \in [0, \beta^t - 1]$, estremo destro per: $\beta^0 + \beta^1 + \dots + \beta^{t-1} = \frac{\beta^t - 1}{\beta - 1} \leq \beta^t - 1$

Rappresentazione numeri su calcolatore. IEEE standards, primi anni 1980

Alcuni valori particolari non rappresentabili direttamente:

valore	esponente	mantissa
± 0	$E_{\min} - 1$	0
$\pm \infty = x/0$	$E_{\max} + 1$	0
NaN=0/0	$E_{\max} + 1$	± 0

Overflow: valore superiore al range dei valori rappresentabili

Underflow: valore inferiore al range dei valori rappresentabili (non zero, vicinissimo a zero)

Precisione macchina (u, ε , eps): distanza tra 1 ed il numero floating point successivo, $1 + \varepsilon$

Numeri floating-point

Singola Precisione: $N = 32$ bits: 1 per il segno, 8 bits per l'esponente, 23 bits per la mantissa (cioè, (0...01))

$E_{\min} = -126$, $E_{\max} = 127$. Quindi

$$(N_{\min})_{(2)} = (1.00 \dots 0) \cdot 2^{-126} \approx 1.2 \cdot 10^{-38} \quad (N_{\max})_{(2)} = (1.11 \dots 1) \cdot 2^{127} \approx 3.4 \cdot 10^{38}$$

minimo e massimo numero *rappresentabili* sulla macchina

$1 + 2^{-23}$ è il numero più vicino ad 1 e maggiore di 1, $u := 2^{-23} = 1.2 \cdot 10^{-7}$

Doppia Precisione: $N=64$ bits: 1 bit per il segno, 11 bits per l'esponente, 52 bits per la mantissa (cioè, (0...01))

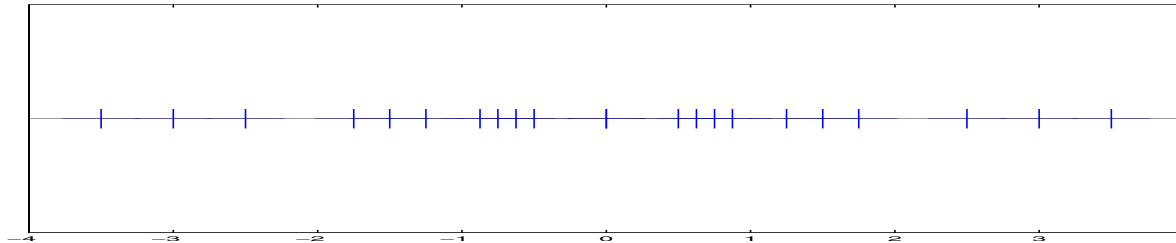
$E_{\min} = -1022$, $E_{\max} = 1023$. Quindi

$$(N_{\min})_{(2)} = (1.00 \dots 0) \cdot 2^{-1022} \approx 2.2 \cdot 10^{-308}, \quad (N_{\max})_{(2)} = (1.11 \dots 1) \cdot 2^{1023} \approx 1.8 \cdot 10^{308}$$

$1 + 2^{-52}$ è il numero più vicino ad 1 e maggiore di 1, $u := 2^{-52} = 2.2 \cdot 10^{-16}$

u : **unità di round-off** nella precisione scelta (unità di precisione macchina)

Numeri floating-point



I valori di $\mathbb{F}$ non sono equispaziati !

Un esercizio

Implementare un algoritmo che calcola l'unità di arrotondamento macchina (ε).

- (i) partire da $u = 1$
- (ii) Porre $u \leftarrow \frac{1}{2}u$ finchè $u + 1 \not\approx 1$

Un modello di aritmetica in $\mathbb{F}$

- In generale, se $x \in \mathbb{R}$ non è detto che $x \in \mathbb{F}$
- Anche se $x, y \in \mathbb{F}$, non è detto che $x \text{ op } y$ sia in $\mathbb{F}$

Quindi

- Definiamo il troncamento di $x \in \mathbb{R}$ in $\mathbb{F}$
- Definiamo un'aritmetica in $\mathbb{F}$

Se $x \in \mathbb{R}$, allora indichiamo $fl(x) = (-1)^s(.a_1a_2 \dots \tilde{a}_t)\beta^e$ dove

$$\tilde{a}_t = \begin{cases} a_t & \text{se } a_{t+1} < \beta/2 \\ a_t + 1 & \text{se } a_{t+1} \geq \beta/2 \end{cases}$$

(arrotondamento dell'ultima cifra significativa al più vicino)

$\Rightarrow$ *i)* $fl(x) = x$ se $x \in \mathbb{F}$, *ii)* $fl(x) \leq fl(y)$ se $x \leq y$

(monotonia)

Operazioni in aritmetica finita (doppia precisione)

Proposizione. Se $x \in \mathbb{R}$ appartiene all'intervallo di $\mathbb{F}$, allora vale
 $fl(x) = x(1 + \delta), \quad |\delta| \leq \frac{1}{2}\varepsilon, \varepsilon = 2.2 \cdot 10^{-16}$

Aritmetica floating point... i falsi amici

★ Siccome le operazioni subiscono dei troncamenti:

$$(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$$

★ Alcune operazioni sono più pericolose di altre:

$$a \ominus b \quad !!!$$

Esempio: b, b', b'' numeri significativi, $g = \textit{garbage}$

$x =$	1	0	1	1	0	0	1	0	1	b	b	g	g	e
$y =$	1	0	1	1	0	0	1	0	1	b'	b'	g	g	e
$x - y =$	0	0	0	0	0	0	0	0	0	b''	b''	g	g	e
$x - y =$	b''	b''	g	g	?	?	?	?	?	?	?	?	?	e-9

Un esempio

```
>> b=(1e-16)+1+(-1e-16)
```

```
1.0000000000000000
```

```
>> c=(1e-16)+(-1e-16)+1
```

```
1
```

```
>> b-c
```

```
ans =
```

```
-1.110223024625157e-16
```

```
>> b==c
```

```
ans =
```

```
logical
```

```
0
```

Errori nelle operazioni floating point

* IEEE Standards: Il risultato di una op. elementare $(+, -, \cdot, /)$ di due numeri floating point dev'essere un arrotondamento corretto del risultato vero; $a \oplus b = fl(a + b) = (a + b)(1 + \delta)$, ecc.

Moltiplicazione: (analogo per la divisione)

$$x(1 + \epsilon_x) \cdot y(1 + \epsilon_y) = xy + xy\epsilon_y + xy\epsilon_x + xy\epsilon_y\epsilon_x \approx xy(1 + \underbrace{\epsilon_y + \epsilon_x}_{=: \epsilon_{xy}})$$

Somma e sottrazione:

$$x(1 + \epsilon_x) + y(1 + \epsilon_y) = (x + y) \left(1 + \underbrace{\frac{x}{x+y}\epsilon_x + \frac{y}{x+y}\epsilon_y}_{\epsilon_{x+y}} \right)$$

se $sgn(x) = sgn(y)$ allora $|\epsilon_{x+y}| \leq |\epsilon_x| + |\epsilon_y|$

$\Rightarrow$	$ \epsilon_{x+y} $	può essere grande quando	$ x + y \ll x , y $
---------------	--------------------	--------------------------	------------------------

Costo computazionale

Quante operazioni macchina (flops) costano le operazioni ?

Alcuni esempi:

$$x, y \in \mathbb{R}: fl(x) \text{ op } fl(y) \Rightarrow 1 \text{ flop}$$

$$x, y \in \mathbb{R}^n: fl(x) + fl(y) \Rightarrow n \text{ flops}$$

$$x \in \mathbb{R}^n, \alpha \in \mathbb{R}: fl(\alpha)fl(x) \Rightarrow n \text{ flops}$$

$$x, y \in \mathbb{R}^n, \alpha \in \mathbb{R}: fl(y) + fl(\alpha)fl(x) \Rightarrow 2n \text{ flops (d)axpy}$$

$$x, y \in \mathbb{R}^n: fl(y)^T fl(x) \Rightarrow 2n - 1 \text{ flops (d)dot}$$

$$x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}: fl(A)fl(x) \Rightarrow (2n - 1)m \text{ flops}$$

(prod.matrice-vettore)