PART-1 → Computer arithmetic and errors

In some situations, when working with a PC, we may face the somehow embarassing situation where

$$1+\epsilon=1$$

and $\epsilon\neq 0$. This is because not all numbers can be represented exactly in a computer, and arithmetic mistakes are a consequence. The information in a **single-precision** PC is built upon vectors composed by "0/1"32 bits (4 bytes). A large number of integers can be thus represented exactly, in base 2.

2^{31}	2^{30}						2^3	2^2	2^1	2^0
0	0	0	0	⋯	⋯	0	0	1	0	1	0

The situation with real numbers $x\neq 0$ is more complicated. The representation of x is structured as

$$x=\pm n\cdot b^e,$$

where n is the **mantissa**, b is the **base** (always 2) and e is the exponent. This representation is also called **floating point**.

It is important to notice that, no matter the number of bits employed, a limited bits are assigned for e and n, therefore not all real numbers can be represented.

NUMERI MACCHINA

:= sottoinsieme di $\mathbb{R}$ rappresentabile in aritmetica di macchina := tutti quanti i reali rappresentabili con il th. di rappresentazione in base

$$\mathcal{F}(\beta,t,m,M)=\{0\}\cup\left\{x\in\mathbb{R} : x=sgn(x)\cdot\beta^p\sum_{i=1}^{t}d_i\beta^{-i}\right\} \quad \text{con} \quad -m\le p\le M,\ 0\le d_i\le\beta-1,\ d_1\neq 0$$

TH. DI RAPPRESENTAZIONE IN BASE

Per ogni $x\in\mathbb{R}$, $x\neq 0$, $\exists!\ p\in\mathbb{Z}$ e $\{d_i\}$ successione di interi tali che:

(i) $d_i\neq 0$
(ii) $0\le d_i\le\beta-1$
(iii) $\forall k>0\ \exists j\geq k$ tale che $d_j\neq\beta-1$

per cui $x=sgn(x)\cdot\beta^p\sum_{i=1}^{+\infty}d_i\beta^{-i}$

Prop: $0.\bar{9}=1$

Dim: $0.\bar{9}=\sum_{i=1}^{+\infty}9\cdot 10^{-i}=9\cdot 10^{-1}+9\cdot 10^{-2}+\ldots=$
$=\frac{9}{10}\cdot\underbrace{\left(1+\frac{1}{10}+\frac{1}{10^2}+\ldots\right)}=$

questa è la serie geometrica $\sum_{n=0}^{+\infty}q^n$ di ragione $q=\frac{1}{10}$ che converge a $\frac{1}{1-q}$

$\sum_{n=0}^{+\infty}\left(\frac{1}{10}\right)^n=\frac{1}{1-\frac{1}{10}}=\frac{1}{\frac{9}{10}}=\frac{10}{9}$

$=\frac{9}{10}\cdot\frac{10}{9}=1$ ☒

CARDINALITÁ INSIEME NUMERI MACCHINA

$$\mathcal{F}(\beta,t,m,M)=\underbrace{\{0\}}_{+1}\cup\left\{\underbrace{sgn(x)}_{\times 2\,(+,-)}\cdot\underbrace{\beta^p}\underbrace{\sum_{i=1}^{t}d_i\beta^{-i}},\ -m\le p\le M,\ d_1\neq 0\right\}$$

i β^p sono tanti quanti i p:
da $-m$ a -1 $(=m)$
+ da 1 a M $(=M)$
+ più lo 0 $(=1)$
$=m+M+1$

Conto tutte le configurazioni possibili $(=\beta^t)$
poi ci tolgo quelle che iniziano con 0 $(=\beta^{t-1})$
quindi $\beta^t-\beta^{t-1}=\beta^t(1-\beta^{-1})$

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

MASSIMO (Ω) & MINIMO (ω) INSIEME NUMERI MACCHINA

$$\Omega = \beta^M (0.d_1 d_2 \dots d_t) =$$
$$= \beta^M \left((\beta-1)\beta^{-1} + (\beta-1)\beta^{-2} + \dots + (\beta-1)\beta^{-t} \right) =$$
$$= \beta^M (\beta-1) \left(\beta^{-1} + \beta^{-2} + \dots + \beta^{-t} \right) =$$
$$= \beta^M (\beta-1) \left(-1 + \underbrace{\sum_{n=0}^{t} \left(\frac{1}{\beta}\right)^n} \right) =$$

Serie geometrica di ragione $q = \frac{1}{\beta}$ troncata a t:

$$S = \sum_{n=0}^{t} q^n = 1 + q + q^2 + \dots + q^t$$
$$S - qS = S(1-q) = (1 + \cancel{q} + \cancel{q^2} + \dots + \cancel{q^t}) - (\cancel{q} + \cancel{q^2} + \dots + \cancel{q^t} + q^{t+1}) = 1 + q^{t+1}$$
$$S = \frac{1 - q^{t+1}}{1-q}$$

$$= \beta^M (\beta-1) \left(\frac{1 - \left(\frac{1}{\beta}\right)^{t+1}}{1 - \frac{1}{\beta}} - 1 \right) =$$
$$= \beta^M (\beta-1) \left(\frac{\beta^{t+1} - 1}{\beta^{\cancel{t+1}}} \frac{\cancel{\beta}}{\beta - 1} - 1 \right) =$$
$$= \beta^M \cancel{(\beta-1)} \frac{\beta^{t+1} - 1 - \beta^t(\beta-1)}{\beta^t \cancel{(\beta-1)}} =$$
$$= \beta^M \frac{\beta^t(\cancel{\beta} - \cancel{\beta} + 1) - 1}{\beta^t} =$$
$$= \beta^M \frac{\beta^t - 1}{\beta^t} = \beta^M (1 - \beta^{-t})$$

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

TRONCAMENTO & ARROTONDAMENTO

$x \in \mathbb{R} \longrightarrow \tilde{x} \in \mathcal{F}$ [ovvero da un numero reale x devo passare ad un numero di macchina $\tilde{x}$]

$$x = \underbrace{sgn(x)}_{+,-} \cdot \underbrace{\beta^p}_{10^p} \cdot \underbrace{\sum_{i=1}^{\infty} d_i \beta^{-i}}_{0.d_1 d_2 d_3 \dots}$$

$$\tilde{x} = tronc(x) = sgn(x) \cdot \beta^p \cdot \sum_{i=1}^{t} d_i \beta^{-i}$$

[limito (tronco) la Σ al t-esimo]

$$\tilde{x} = arr(x) = sgn(x) \cdot tronc\left(\beta^p \cdot \left(\sum_{i=1}^{t+1} d_i \beta^{-i}\right) + \frac{1}{2}\beta^{-t}\right) =$$

$$= \begin{cases} tronc(x) & \text{se } d_{t+1} < \frac{1}{2}\beta \\ tronc(x) + \beta^{p-t} & \text{se } d_{t+1} \geq \frac{1}{2}\beta \end{cases}$$

$$\left[\begin{aligned} \tilde{x} &= \underset{p=3}{10^3} \cdot 0.37562\underset{t=5}{|}9\dots = \\ &= tronc(x) + \beta^{p-t} = \\ &= 10^3 \cdot 0.37562 + 10^{-2} = \\ &= 375.62 + 0.01 \\ &= 375.63 \end{aligned}\right]$$

NOTA: mentre con il troncamento sto sempre dentro l'intervallo $[\omega, \Omega]$, con l'arrotondamento potrei andare oltre $\Omega \Rightarrow$ OVERFLOW

$$\boxed{\begin{aligned} &\text{OVERFLOW} := p > M \\ &\text{UNDERFLOW} := p < -m \end{aligned}}$$

ERRORI DI RAPPRESENTAZIONE

Errore assoluto $:= \tilde{x} - x$

Errore relativo $:= \dfrac{\tilde{x} - x}{x}$ $\longrightarrow$ chiamato anche Errore di rappresentazione $\mathcal{E}_x$ in $\mathcal{F}(\beta, t, m, M)$
Quindi studiare il problema significa andare a vedere come questo errore si propaga all'interno dell'algoritmo.

NOTA: se voglio rappresentare l'errore sulle cifre della mantissa $(d_1 d_2 d_3 \dots)$, l'errore assoluto mi dipende da β^p (ovvero dalla grandezza del numero), mentre l'errore relativo no, quindi a parità di mantissa ottengo lo stesso errore

ESEMPIO: i numeri di macchina $\mathcal{F}(\underbrace{2}_{\text{base } 2}, \underbrace{3}_{\text{arrivo fino alla 3° cifra di mantissa}}, \underbrace{1, 1}_{\text{min e Max dell'exp. (in questo caso sempre = 1)}})$ sono una buona definizione perchè gli errori sono contenuti

$\Rightarrow \boxed{2 \cdot 0.d_1 d_2 d_3}$ dentro questa $\mathcal{F}$, tutti i numeri sono rappresentati in questo modo

TH. Se non si verifica overflow o underflow allora:

$$|\text{tronc}(x)-x| < \beta^{p-t}$$

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

DIM. Prendo $a,b \in \mathcal{F}$ consecutivi:

$$a = \beta^p \cdot \sum_{i=1}^{t} d_i \beta^{-i}$$

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

(ho semplicemente sommato 1 alla cifra t-esima)

Quindi se ho $a \leq x < b$, indipendentemente da quanto sono vicino a b, casco sempre in a:

(i) $\text{tronc}(x) = a \Rightarrow |\text{tronc}(x)-x| = |a-x| = |a-x| < |a-b| = \beta^{p-t}$

(ii) $\text{arr}(x) = \begin{cases} a & \text{se } x < \frac{a+b}{2} \\ b & \text{se } x \geq \frac{a+b}{2} \end{cases} \Rightarrow |\text{arr}(x)-x| \leq \frac{b-a}{2} = \frac{1}{2}\beta^{p-t}$

(l'uguaglianza vale solo se $x = \frac{a+b}{2} \Rightarrow$ sono a metà intervallo)

$\square$

TH. (limitazione dell'errore di rappresentazione)
se non si verifica overflow o underflow, allora la quantità μ detta precisione di macchina è così definita:

$$\left|\frac{\tilde{x}-x}{x}\right| < \mu = \begin{cases} \beta^{1-t} & \text{se } \tilde{x} = \text{tronc}(x) \\ \frac{1}{2}\beta^{1-t} & \text{se } \tilde{x} = \text{arr}(x) \end{cases}$$

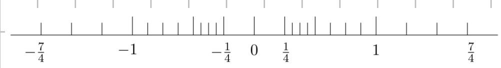
μ non dipende dalla x che sto rappresentando
ma è una limitazione dell'errore relativo (x non deve essere in over/underflow)
Quindi posso dire che μ dipende dalla macchina e non dalla rappresentazione di x!

DIM. $d_1 \neq 0$ (definizione di $\mathcal{F}(\beta,t,m,M)$) $\Rightarrow x \geq \beta^p \cdot d_1 \beta^{-1} \geq \beta^{p-1} \Rightarrow \left|\frac{\tilde{x}-x}{x}\right| \leq \frac{|\tilde{x}-x|}{\beta^{p-1}}$

(i) $\tilde{x} = \text{tronc}(x) \Rightarrow \left|\frac{\tilde{x}-x}{x}\right| < \frac{\beta^{p-t}}{\beta^{p-1}} = \beta^{1-t}$

per maggiorare, diminuisco il denominatore con $\beta^p(0.1) = \beta^{p-1}$

NOTA: β^{p-t} è la distanza tra due numeri di macchina consecutivi



$\mathcal{F}(2,3,1,1)$

(ii) $\tilde{x} = \text{arr}(x) \Rightarrow \left|\frac{\tilde{x}-x}{x}\right| < \frac{1}{2}\frac{\beta^{p-t}}{\beta^{p-1}} = \frac{1}{2}\beta^{1-t}$

anche qui è come sopra ma ricorda che:
$|\text{arr}(x)-x| \leq \frac{1}{2}\beta^{p-t}$, ma l'uguaglianza vale solo quando

$\begin{cases} d_{t+1} = \frac{\beta}{2} \\ d_{t+i} = 0 \quad \forall i \geq 2 \end{cases}$; quindi $x \geq \beta^p\left(d_1\beta^{-1} + d_{t+1}\beta^{-(t+1)}\right) > \beta^{p-1}$

posso riscrivere tutto: $\left|\frac{\text{arr}(x)-x}{x}\right| \leq \frac{1}{2}\frac{\beta^{p-t}}{|x|} < \frac{1}{2}\frac{\beta^{p-t}}{\beta^{p-1}} = \frac{1}{2}\beta^{1-t}$

$\square$

NOTA: ($\mathcal{E}_x$ vs μ)

$$\mathcal{E}_x = \frac{\tilde{x}-x}{x} \quad , \quad |\mathcal{E}_x| < \mu$$

$\Rightarrow \tilde{x} = x(1+\mathcal{E}_x)$ $\Rightarrow$ Quindi $\tilde{x}$ non è altro che una perturbazione di x!

$\Rightarrow$ l'errore assoluto dipende dalla grandezza del numero
l'errore relativo dipende dalle specifiche della macchina

ERRORE NELLE OPERAZIONI MACCHINA

$$x \circledcirc y = \begin{cases} \text{tronc}(x \text{ op } y) \\ \text{arr}(x \text{ op } y) \end{cases} \Rightarrow x \circledcirc y = (x \text{ op } y)(1+\mathcal{E})$$

(op cerchiato = operazione in aritmetica di macchina)

con $|\mathcal{E}| < \mu$
$\mathcal{E}$:= errore locale dell'operazione op, che impongo essere più piccolo della precisione di macchina μ

$$\Rightarrow \mathcal{E}_{Tot} = \frac{(\tilde{x} \circledcirc \tilde{y}) - (x \text{ op } y)}{x \text{ op } y} = \mathcal{E} + \boxed{C_x}\mathcal{E}_x + \boxed{C_y}\mathcal{E}_y$$

COEFF. DI AMPLIFICAZIONE: mi danno la misura di quanto l'errore di rappresentazione di un dato si amplifica nel calcolo di una funzione

In questo modo se ho troncamento o arrotondamento di una operazione op, l'errore commesso è dell'ordine della precisione di macchina μ.

FENOMENO DELLA CANCELLAZIONE NUMERICA := quando rappresento con precisione finita dei numeri reali in un calcolatore, potrei perdere cifre significative, causa una operazione di sottrazione tra due numeri "quasi uguali" (= con le prime t cifre uguali)

ESEMPIO:

$$\begin{aligned} p_1 &= 0.12345789 \\ p_2 &= 0.12345666 \end{aligned}$$

$$d = p_1 - p_2 = 0.00000123 = 0.123 \cdot 10^{-7}$$

Supponendo $\mathcal{F}(10,6,m,M)$ e approssimazione con troncamento:

$$\begin{aligned} \text{tronc}(p_1) &= 0.123457 \\ \text{tronc}(p_2) &= 0.123456 \end{aligned} \Rightarrow \text{tronc}(d) = \text{tronc}(p_1) - \text{tronc}(p_2) = 0.000001 = 10^{-6}$$

Quindi se calcolo l'errore totale $\mathcal{E}_{Tot}$ della differenza, ottengo un valore molto alto:

$$\mathcal{E}_{Tot} = \left| \frac{(p_1-p_2)-(\tilde{p}_1-\tilde{p}_2)}{p_1-p_2} \right| = \left| \frac{d-\text{tronc}(d)}{d} \right| = \left| \frac{0.123\cdot 10^{-7} - 10^{-6}}{0.123 \cdot 10^{-7}} \right| = \left| 1 - \frac{10^4}{123} \right| \simeq 80.3$$

Errore nel Calcolo di una Funzione Razionale

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$
$$[x] \longmapsto x$$

$$f(x) \rightsquigarrow f(\tilde{x}) \rightsquigarrow \psi(\tilde{x})$$

errore di rappresentaz. ($x \rightsquigarrow \tilde{x}$)

errore dovuto all'utilizzo di op invece di op ($f \rightsquigarrow \psi$)

$\Rightarrow$ Sono partito per calcolare $f(x)$, vado a calcolare $\psi(\tilde{x})$

$$\varepsilon_{tot} = \frac{\psi(\tilde{x}) - f(x)}{f(x)} := (*)$$

$$\varepsilon_{in} = \frac{f(\tilde{x}) - f(x)}{f(x)}$$:= Errore Inerente causato dalla rappresentazione in macchina

$$\varepsilon_{alg} = \frac{\psi(\tilde{x}) - f(\tilde{x})}{f(\tilde{x})}$$:= Errore Algoritmico causato dalla rappresentazione in macchina che non produce un numero macchina $\in \mathcal{F}(\beta, t, m, M)$

$$(*) = \frac{\psi(\tilde{x})}{f(x)} - 1 = \frac{\psi(\tilde{x})}{f(\tilde{x})} \frac{f(\tilde{x})}{f(x)} - 1 =$$

$$= (1 + \varepsilon_{alg})(1 + \varepsilon_{in}) - 1 = \cancel{1} + \varepsilon_{alg} + \varepsilon_{in} + \underbrace{\varepsilon_{alg}\, \varepsilon_{in}} - \cancel{1} \doteq \varepsilon_{alg} + \varepsilon_{in}$$

questo pezzo lo posso buttare perché opero sotto l'assunzione che la precisione di macchina $|\mu| \ll 1$ e che gli errori ε_{alg} e ε_{in} siano $O(\mu)$

$\Rightarrow$ Posso trascurare i termini non lineari: ANALISI DELL'ERRORE AL PRIMO ORDINE ($\doteq$)

Studio Errore Inerente (= condizionamento problema nel calcolo di $f(x)$)

$$\varepsilon_{in} = \frac{f(\tilde{x}) - f(x)}{f(x)} = \frac{f(\tilde{x}) - f(x)}{\tilde{x} - x} \; \frac{\tilde{x} - x}{f(x)} =$$

$$= \frac{f(\tilde{x}) - f(x)}{\tilde{x} - x} \; \frac{x}{f(x)} \; \frac{\tilde{x} - x}{x} \doteq \boxed{\frac{f'(x)}{f(x)} \, x} \cdot \varepsilon_x$$

Coefficiente di amplificazione C_x
misura come l'errore sul dato viene amplificato nel calcolo della funzione f

per questa analisi al primo ordine posso pensare allo sviluppo di Taylor di f in $\tilde{x}$:

$$f(\tilde{x}) \doteq f(x) + (x - \tilde{x}) \cdot f'(x)$$

Se quindi il C_x è sempre limitato in modulo da una qualche costante (ovvero posso maggiorare $|C_x|$ con una costante k relativamente piccola alla precisione di macchina), allora si dice che il problema è BEN CONDIZIONATO

Viceversa, se per $x \to \tilde{x}$, $|C_x| \longrightarrow \infty$, allora il problema è MAL CONDIZIONATO nel punto $\tilde{x}$

Studio Errore Algoritmico (= errore nelle operazioni di macchina)

Come visto, è quell'errore che commette l'algoritmo partendo dai valori già trasformati in numeri macchina:

$$\tilde{x} + \tilde{y} \longrightarrow \tilde{x} \oplus \tilde{y} = \begin{cases} \mathrm{tronc}(\tilde{x} + \tilde{y}) \\ \mathrm{arr}(\tilde{x} + \tilde{y}) \end{cases}$$

$$\boxed{\varepsilon_{Tot}} \doteq \boxed{\varepsilon_{alg}} + \boxed{\varepsilon_{in}} = \varepsilon_{alg} + \boxed{C_x \varepsilon_x + C_y \varepsilon_y}$$

$$|\varepsilon_{Tot}| = \left| \frac{\psi(\tilde{x}, \tilde{y}) - f(x, y)}{f(x, y)} \right|$$

$$|\varepsilon_{alg}| = \left| \frac{\psi(\tilde{x}, \tilde{y}) - f(\tilde{x}, \tilde{y})}{f(\tilde{x}, \tilde{y})} \right| = \left| \frac{\mathrm{arr}(\tilde{x} \text{ op } \tilde{y}) - (\tilde{x} \text{ op } \tilde{y})}{\tilde{x} \text{ op } \tilde{y}} \right| < \underbrace{\mu}$$

errore locale limitato dalla precisione di macchina

Quindi, se non ho avuto overflow o underflow, quando approssimo un numero con un numero di macchina, commetto un errore limitato alla precisione di macchina μ

Coefficienti di Condizionamento C_x e C_y per le 4 Operazioni

$$C_x = \frac{\partial}{\partial x} f(x,y) \cdot \frac{x}{f(x,y)}$$

$$C_y = \frac{\partial}{\partial y} f(x,y) \cdot \frac{y}{f(x,y)}$$

$$f(x,y) = x+y \Rightarrow C_x = \frac{x}{x+y}, \quad C_y = \frac{y}{x+y}$$

$$f(x,y) = x-y \Rightarrow C_x = \frac{x}{x-y}, \quad C_y = -\frac{y}{x-y}$$

$\Rightarrow$ in entrambi i casi, per $x \to y$ il coefficiente scoppia: ci può essere un caso di cancellazione numerica

$$f(x,y) = xy \Rightarrow C_x = 1, \quad C_y = 1$$

$$f(x,y) = x/y \Rightarrow C_x = 1, \quad C_y = -1$$

ANALISI DELL'ERRORE CON GRAFI

Supponiamo adesso di avere una funzione razionale che non sia una singola operazione aritmetica $\Psi(\underbrace{\tilde{x}}_{=\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\\ \vdots\\ \tilde{x}_n\end{bmatrix}})$ calcolato al posto di $f(\underbrace{x}_{=\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}})$ lo ottengo sostituendo agli x_i i corrispondenti numeri macchina $\tilde{x}_i$ e sostituendo alle operazioni aritmetiche op, le corrispondenti operazioni di macchina (op) nell'ordine in cui vengono eseguite.

Schematicamente Ψ è ottenuta implementando un algoritmo costituito da p passi del tipo:

$$z^{(i)} = y_1^{(i)} \text{ op } y_2^{(i)} \qquad \text{per } i=1,2,\dots,p$$

sono i dati iniziali o i risultati delle operazioni ai passi precedenti:

$$y_1^{(i)}, y_2^{(i)} \in \left\{x_1, x_2, \dots, x_n, z^{(1)}, z^{(2)}, \dots, z^{(i-1)}\right\}$$

$$f(x) = z^{p}$$

ESEMPIO: $f(x_1, x_2) = x_1^2 - x_2^2$

la posso calcolare con il seguente algoritmo:

$$z^{(1)} = x_1 \cdot x_1$$
$$z^{(2)} = x_2 \cdot x_2$$
$$z^{(3)} = z^{(1)} - z^{(2)}$$

Adesso, per calcolare l'errore $\varepsilon_{Tot}^{(i)}$ del risultato all'i-esimo passo:

$$\varepsilon_{Tot}^{(i)} \doteq \underbrace{\varepsilon^{(i)}}_{\text{errore generato dalla i-esima op.}} + C_1^{(i)} \varepsilon_1^{(i)} + C_2^{(i)} \varepsilon_2^{(i)} \qquad \text{per } i=1,2,\dots,p$$

errori presenti negli operandi della i-esima operazione $\Rightarrow$ possono essere causati dai dati iniziali o accumulati nei risultati intermedi del calcolo

Quello che ancora non sappiamo calcolare è l'errore algoritmico ε_{alg}.
Procediamo con un grafo che descrive la sequenza delle operazioni dell'algoritmo: