

# Numeri di macchina

Lucia Gastaldi

DICATAM - Sez. di Matematica,  
<http://lucia-gastaldi.unibs.it>



UNIVERSITÀ  
DEGLI STUDI  
DI BRESCIA

- 1 Rappresentazione dei numeri
  - Formato di memorizzazione dei numeri
  - Arrotondamento di un numero reale
- 2 Operazioni di macchina
- 3 Problemi con l'aritmetica Floating Point

# Formato di memorizzazione dei numeri

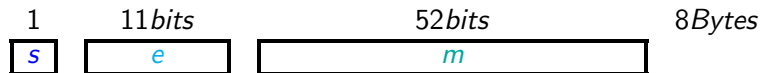
- ▶ **Singola** (o semplice) precisione, 4 Bytes (32 bits)
- ▶ **Doppia** precisione, 8 Bytes (64 bits)

Come vengono utilizzati questi Bytes?

Si considera la forma esponenziale di un numero reale:

$$x = 123456.789 = (-1)^0 1.23456789 \cdot 10^5 = (-1)^s m \cdot \beta^e$$

$s = 0, 1$ ;  $m$  mantissa;  $\beta$  base (es: 2,10);  $e$  esponente



# Insieme dei numeri rappresentabili

$$x = (-1)^s m \cdot \beta^e = (-1)^s (a_1 a_2 \cdots a_t) \cdot \beta^e = (-1)^s \beta^e \sum_{i=1}^t a_i \beta^{-i}$$

con le seguenti restrizioni:

$$L \leq e \leq U \quad \text{tipicamente } L < 0 \text{ e } U > 0 \quad (L = -1021, U = 1024)$$

$$\beta \geq 2$$

$$0 \leq a_i \leq \beta - 1 \quad \text{per } i = 1, \dots, t, \quad a_1 \neq 0$$

Le 11 cifre binarie dell'esponente danno i numeri interi da 0 a 2047.

## Floating point in Matlab

$$x_M = (-1)^s 2^e (1 + f) \quad \text{con } U = 1023 \text{ e } L = -1022.$$

Il segno di  $e$  viene ottenuto salvando  $e + 1023$ , che va da 1 a  $2^{11} - 2$ .

## Codifiche particolari

valore	esponente	mantissa
$\pm 0$	$L - 1$	0
Inf	$U + 1$	0
NaN	$U + 1$	$\neq 0$

# Unicità della rappresentazione

$$.1000 \cdot 10^1 \quad .0100 \cdot 10^2 \quad .0010 \cdot 10^3 \quad .0001 \cdot 10^4$$

sono tutte rappresentazioni di 1.

Per avere **unicità della rappresentazione** si richiede:

$$a_1 \neq 0; \quad \Rightarrow \beta^{-1} \leq m < 1$$

La rappresentazione di  $x$  si dice **normalizzata**.

$$x_{min} = \beta^{L-1} \leq |x| \leq \beta^{U-1} \frac{\beta - \beta^{-t}}{\beta - 1} = x_{max}$$

$realmin=2^{(-1022)}$ ,  $realmax=2^{1023}*(2-2^{(-52)})$

Si può rinunciare alla restrizione che  $a_1 \neq 0$  nel caso  $e = L$  e si ha la rappresentazione **floating-point denormalizzata**.

$$\beta^{-t} \leq m < 1 \quad e \quad x_{min} = \beta^{L-t}$$

# Arrotondamento

$$x = (-1)^s \beta^e \cdot \sum_{i=1}^{\infty} a_i \beta^{-i}$$

**numero reale**

Se  $L \leq e \leq U$  allora l'**arrotondamento** di  $x$   $fl(x)$  è definito come il numero floating point più vicino a  $x$ .

$$\frac{|x - fl(x)|}{|x|} \leq \frac{\beta^{-t} \beta^e}{2 |m| \beta^e} \leq \frac{1}{2} \beta^{1-t} = \frac{1}{2} \epsilon_M = u$$

eps    **precisione di macchina**  
u        **unità di arrotondamento**

eps+1 > 1    eps =  $\beta^{1-t}$   
u = eps/2

## Calcolo di eps

```
Eps=1;
while 1+Eps>1
  Eps=Eps/2;
end
Eps=Eps*2
```

## In Matlab:

```
eps = 2-52
u = 2-53
```

# Operazioni di macchina

## Aritmetica IEC559

Lo standard IEC559 provvede a definire le operazioni sull'insieme dei numeri di macchina in modo che ogni operazione produce un risultato all'interno del sistema stesso.

Supponiamo che  $x$  e  $y$  siano due numeri reali e che  $fl(x)$  e  $fl(y)$  siano i loro arrotondamenti.

Indichiamo con  $\circ$  una delle tre operazioni  $*$ ,  $/$ ,  $\pm$  eseguita in matematica esatta, e con  $\hat{\circ}$  l'operazione in matematica di macchina corrispondente. Si può pensare che l'operazione di macchina venga effettuata nel modo seguente:

$$fl(x)\hat{\circ}fl(y) = fl(fl(x) \circ fl(y)).$$

# Risultati per alcune operazioni eccezionali

eccezione	esempi	risultato
operazione non valida <i>overflow</i>	$0/0, 0 \cdot \infty$	NaN
divisione per zero <i>underflow</i>	$1/0$	Inf
		Inf
		numeri sottonormali

## Forme indeterminate:

$$\frac{0}{0}, \frac{\infty}{\infty}, 0 \cdot \infty, \infty - \infty, 1^\infty, 0^0, \infty^0.$$



# Propagazione degli errori di arrotondamento

Confrontiamo il **risultato esatto** dell'operazione con il **risultato approssimato** ottenuto con l'operazione di macchina, vogliamo quindi valutare il seguente **errore relativo**

$$\frac{|x \circ y - fl(x) \hat{=} fl(y)|}{|x \circ y|}.$$

## Stabilità

Diciamo che un'operazione di macchina è **stabile** se l'errore relativo rimane limitato dagli errori relativi introdotti nell'arrotondamento dei numeri  $x$  e  $y$ .

# Stabilità delle operazioni di macchina

L'errore relativo commesso nell'effettuare un'operazione di macchina può essere scritto come segue:

$$\frac{|x \circ y - fl(x) \hat{\circ} fl(y)|}{|x \circ y|}$$

$$\leq \frac{|x \circ y - fl(x) \circ fl(y)|}{|x \circ y|} \quad \text{propagazione}$$

$$+ \frac{|fl(x) \circ fl(y) - fl(fl(x) \circ fl(y))|}{|x \circ y|} \quad \text{arrotondamento}$$

e si ottiene per il termine di arrotondamento:

$$\frac{|fl(x) \circ fl(y) - fl(fl(x) \circ fl(y))|}{|x \circ y|} \leq u \quad \text{precisione di macchina.}$$

# Stabilità della moltiplicazione

Per la definizione dell'arrotondamento di un numero e della precisione di macchina, si può scrivere:

$$fl(x) = x(1 - \delta_1), \quad fl(y) = y(1 - \delta_2)$$

essendo  $|\delta_1| \leq u$  e  $|\delta_2| \leq u$ .

## Teorema

La moltiplicazione è stabile.

### Dimostrazione

Valutiamo il termine di propagazione dell'errore:

$$\begin{aligned} \frac{|x * y - fl(x) * fl(y)|}{|x * y|} &= \frac{|x * y - x(1 - \delta_1) * y(1 - \delta_2)|}{|x * y|} \\ &= |\delta_1 + \delta_2 + \delta_1\delta_2| \leq 3u. \end{aligned}$$

# Stabilità della divisione

## Teorema

La divisione è stabile.

## Dimostrazione

Valutiamo il termine di propagazione dell'errore:

$$\begin{aligned} \frac{|x/y - fl(x)/fl(y)|}{|x/y|} &= \frac{|x/y - x(1 - \delta_1)/y(1 - \delta_2)|}{|x/y|} \\ &= \left| 1 - \frac{1 - \delta_1}{1 - \delta_2} \right| = \left| \frac{1 - \delta_2 - 1 + \delta_1}{1 - \delta_2} \right| \\ &= \left| \frac{\delta_1 - \delta_2}{1 - \delta_2} \right| \leq 2u. \end{aligned}$$



# Instabilità dell'addizione e della sottrazione

Ripetiamo l'analisi fatta nel caso della moltiplicazione e della divisione:

$$\begin{aligned}\frac{|x + y - (fl(x) + fl(y))|}{|x + y|} &= \frac{|x + y - x(1 - \delta_1) - y(1 - \delta_2)|}{|x + y|} \\ &= \frac{|x\delta_1 + y\delta_2|}{|x + y|} \leq \frac{|x|}{|x + y|}\delta_1 + \frac{|y|}{|x + y|}\delta_2.\end{aligned}$$

Nel caso in cui  $|x + y|$  diventa molto piccolo rispetto ad  $x$  e  $y$  si ha che l'errore commesso può essere amplificato senza nessun controllo.

**L'addizione e la sottrazione non sono operazioni stabili.**



# Effetto dell'ordine delle operazioni

## Primo esempio

```
b=1e-16+1-1e-16
```

```
c=1e-16+1e-16+1
```

```
b-c
```

```
b==c
```

## Secondo esempio

La funzione

$$F(x) = 1 - x * \left( \frac{x+1}{x} - 1 \right)$$

vale identicamente 0 per ogni valore  $x \neq 0$ .

Calcoliamo il valore di  $F(x)/\text{eps}$  per ogni valore intero in due modi diversi:

```
f=@(n) (1-n.*((n+1)./n-1))./eps
```

```
plot(f(1:256),'.')
```

```
g=@(n) (1-n.*((n+1)./n)+n)./eps
```

```
hold on
```

```
plot(g(1:256),'.r')
```

## Cancellazione numerica

Consideriamo la funzione  $h(x) = \sqrt{x+1} - 1$ .

Assegniamo la funzione `h=@(x) sqrt(x+1)-1;`

Facciamo il grafico nell'intervallo  $[-1, 1]$  con il comando  
`fplot(h, [-1, 1])`

Studiamo l'effetto della **cancellazione numerica** per valori di  $x$  vicino a 0.

```
y=0:(1/16):16;  
x=eps*y;  
plot(x,h(x))
```

Usiamo ora l'espressione alternativa `k=@(x) x./(sqrt(x+1)+1);`  
e riproduciamo il grafico sulla figura precedente:

```
hold on  
plot(x,k(x),'r--')  
legend('h','k','Location','NorthWest')
```



# Proprietà associativa

**Le operazioni di macchina non godono della proprietà associativa.**

## Esempio

$$\text{Sia } a = 0.23371258 \cdot 10^{-10}$$

$$b = 0.71133225533678429 \cdot 10^4 \quad c = -0.71133225533677811 \cdot 10^4$$

allora si ha:

$$\begin{aligned}(a + b) + c &= 7.113322553367867e + 03 - 7.113322553367781e + 03 \\ &= 8.549250196665525 \cdot 10^{-11}\end{aligned}$$

$$\begin{aligned}a + (b + c) &= 2.337125800000000e - 11 + 6.184563972055912e - 11 \\ &= 8.521689772055913e - 11\end{aligned}$$

# Disastri dovuti ad errori nelle simulazioni numeriche

- ▶ Bersaglio sbagliato da un missile Patriot durante la Guerra del Golfo (25 Febbraio 1991).  
**Causa** Scarsa considerazione degli errori di arrotondamento.
- ▶ Esplosione del razzo Ariane 5 (4 Giugno 1996).  
**Causa** Conseguenza di un semplice overflow.

Per maggiori informazioni consultare il sito

<http://www.ima.umn.edu/arnold/disasters/disasters.html>

## Bersaglio sbagliato da un missile Patriot durante la Guerra del Golfo

Il 25 Febbraio 1991 durante la Guerra del Golfo un missile americano Patriot non è riuscito ad intercettare un missile Scud iracheno che ha colpito una baracca dell'esercito Americano uccidendo 28 soldati e ferendone altri 100.

### Causa Errori di arrotondamento

$$\frac{1}{3} \times 3 = 1 \quad 0.3 \times 3 = 0.9 \neq 1$$

Invece

$$\frac{1}{10} \times 10 = 1 \quad 0.1 \times 10 = 1$$

Però in rappresentazione binaria si ha

$$\begin{aligned} \frac{1}{10} &= \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^8} + \frac{1}{2^9} + \frac{1}{2^{12}} + \frac{1}{2^{13}} + \dots \\ &= 0.000110011001100110011001100\dots \end{aligned}$$



# Esplosione del razzo Ariane 5

Il 4 Giugno 1996, il razzo Ariane 5 lanciato dalla Agenzia Spaziale Europea è esploso 40 secondi dopo il lancio da Kourou, nella Guiana Francese. Il razzo era al suo primo volo dopo una ricerca di 10 anni costata 7000 milioni di dollari. Il costo del razzo e della pedana di lancio è stato valutato in 500 milioni di dollari.

## Causa

La commissione di indagine ha studiato le cause dell'esplosione e dopo due settimane ha dato il suo responso. Un numero rappresentato in virgola mobile a 64 bit relativo alla velocità del razzo rispetto alla piattaforma è stato convertito in un intero a 16 bit con segno.

Il numero era maggiore di 32768, il più grande numero intero che si può memorizzare col segno in 16 bit. Quindi la conversione è fallita.