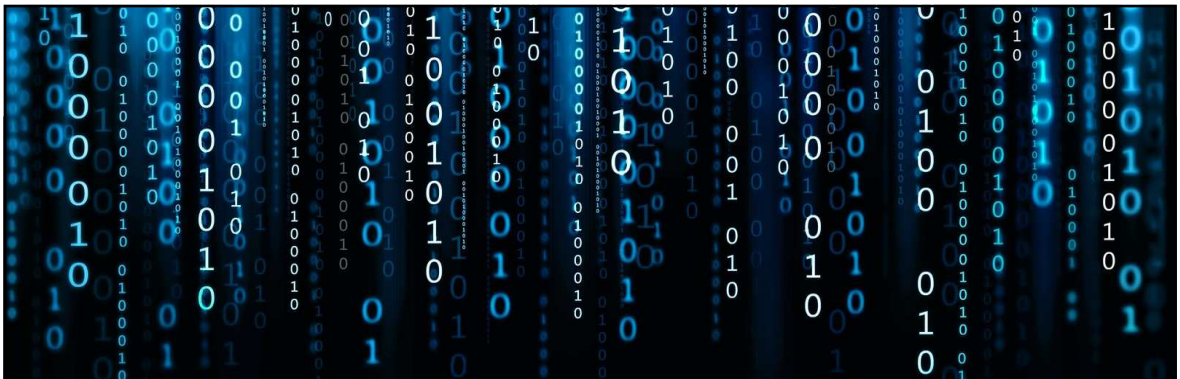


LABORATORIO DI PROGRAMMAZIONE
Corso di laurea in matematica

13 – GLI ERRORI DI ROUND-OFF

Marco Lapegna
Dipartimento di Matematica e Applicazioni
Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

Gli errori nella risoluzione di un problema

- Un processo di risoluzione di un problema scientifico e' soggetto numerosi tipi di errore
  - Imprecisione degli strumenti di misura
  - Semplificazione nel modello matematico
  - Errori di rappresentazione dei dati reali nella memoria del computer

Uno dei problemi del calcolo scientifico e' valutare
l'accuratezza del risultato calcolato da un algoritmo

- **Esempio**:
  sia $x=10.1294$ e una sua approssimazione (generica) $x^*= 10.1253$

Un modo per misurare la bonta' della approssimazione e' calcolare

$$E_A = |x = x^*| = |10.1294 - 10.1253| = 0.0041 = 0.41x10^{-2}$$

ERRORE ASSOLUTO

osservazione

- $x=10.1294$ e $x^*= 10.1253$ hanno *2* cifre decimali in comune
  (x^* e' una approssimazione di x corretta a ***2* cifre decimali**)

$$E_A = |x = x^*| = 0.41x10^{-2} < 10^{-2}$$

- **In generale**

se x^* e' una approssimazione corretta a m cifre decimali si ha che

$$E_A = |x = x^*| < 10^{-m}$$

L'errore assoluto fornisce informazioni sulle **cifre decimali esatte**

esempio

- $x_1 = 10.1294$ e $x_1^* = 10.1253$ si ha $E_A = |10.1294 - 10.1253| = 0.41x10^{-2}$
- $x_2 = 2410.1294$ e $x_2^* = 2410.1253$ si ha $E_A = |2410.1294 - 2410.1253| = 0.41x10^{-2}$

I due errori assoluti sono uguali, ma "intuitivamente" la seconda approssimazione e' migliore della prima (perche' commette lo stesso errore su un dato piu' grande)

Un modo per tenere conto dell'ordine di grandezza del numero da approssimare e'

$$E_R = |x - x^*| / |x|$$

ERRORE RELATIVO

- x_1 e x_1^* ➔ $E_R = |10.1294 - 10.1253| / |10.1294| = 0.0004 = 0.4x10^{-3}$
- x_2 e x_2^* ➔ $E_R = |2410.1294 - 2410.1253| / |2410.1294| = 0.0000017 = 0.17x10^{-5}$

Nel secondo caso l'errore relativo e' piu' piccolo

osservazione

- x_2 e x_2^* hanno complessivamente 6 cifre in comune (4 intere e 2 decimali)
  (x_2* e' una approssimazione di x_2 corretta a ***6* cifre significative**)

$$E_R = |x_2^* - x_2| / |x_2| = 0.17x10^{-5} < 10^{-6+1}$$

- **In generale**

se x* e' una approssimazione corretta a m cifre significative si ha che

$$E_R = |x - x^*| / |x| < 10^{-m+1}$$

L'errore relativo fornisce informazioni sulle **cifre significative esatte**

Gli errori di rappresentazione

Esempio: In un sistema aritmetico floating point normalizzato
$F = \{ b=10 , t=5 , Emin=-9 , Emax=9 \}$

$x = 10.4534$ non e' esattamente rappresentabile ➔ $fl(x) = 0.10453x10^2$

Che errore si commette rappresentando *x* con *fl(x)*?

Osservazione: La mantissa di $fl(x)$ contiene le **cifre significative** del numero

Studiamo l'errore relativo $E_R = |x - fl(x)| / |x|$

Errore relativo di round-off (di rappresentazione)

Errore di round off

$x = 10.4532$ e $fl(x) = 0.10453x\ 10^2$
hanno $t = 5$ cifre in comune

$$E_R = |x - fl(x)| / |x| = 0.000019 = 0.19\ x10^{-4} < 10^{-4} = 10^{1-5} = 10^{1-t}$$

IN GENERALE

Ci chiediamo qual'e' il **massimo errore relativo** che si commette
rappresentando *x* con *fl(x)*

$x = f\ x\ b^e$ $fl(x) = f'\ x\ b^e$ $(1/b \le f < 1\ )$
f e f' hanno t cifre in comune

Quindi

$$E_R = |x - fl(x)| / |x| = |f - f'| / |f| \le b^{1-t}$$ (nel caso di troncamento)

$$E_R = |x - fl(x)| / |x| = |f - f'| / |f| \le b^{1-t}/2$$ (nel caso di arrotondamento)

Definizione

Il **massimo errore relativo** che si commette
rappresentando *x* con *fl(x)*

$$u = max\ |x - fl(x)| / |x| \le b^{1-t}/2$$

e' detto **Massima Accuratezza Relativa**
(e' una delle costanti macchina)

Inoltre, posto $\delta = (fl(x) - x)/x$ $fl(x) = x(1+\delta)$ con $|\delta| \le u$

Esempi:

$F = \{ b=10 , t=5 , Emin=-9 , Emax=9 \}$ $u = 0.5 \times 10^{-4}$

$F = \{ b=2 , t=23 , Emin=-127 , Emax=128 \}$ *(IEEE s.p.)* $u=2^{-23} \sim 0.119x10^{-6}$

$F = \{ b=2 , t=52 , Emin=-1023 , Emax=1024 \}$ *(IEEE d.p.)* $u=2^{-52} \sim 0.222x10^{-15}$

Gli errori nelle operazioni

Esempio:

- $F = \{ b=10 , t=4 , Emin=-9 , Emax=9 \}$
- $x = 0.9983x10^{2}$ e $y=0.4652x10^{-1}$

In aritmetica esatta $z = x+y = 0.9987652\ x10^{2}$ (non rappresentabile)

Indichiamo la somma eseguita dal calcolatore con il simbolo $+_{fp}$

Vogliamo studiare **l'errore relativo di round-off commesso nel calcolare**

$$z^{*} = x +_{fp} y$$

$$E_R = | z - z^{*} | / | z | = |(x+y) - (x +_{fp} y)| / |x+y|$$

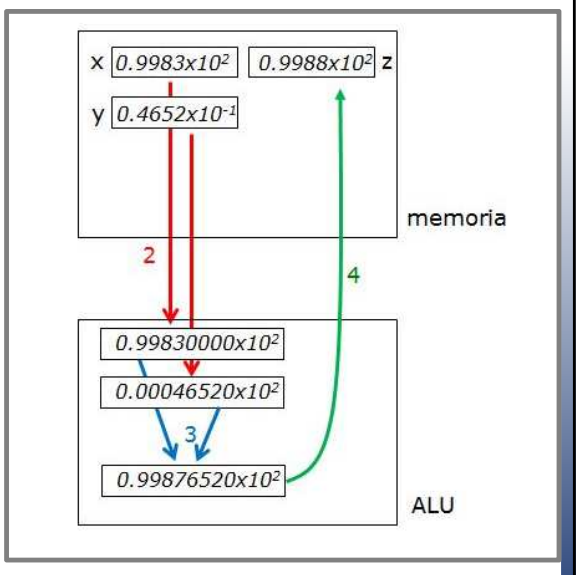
La somma floating point $+_{fp}$

1: calcolo della differenza degli ordini di grandezza
- $d=2+1 = 3$

2: prelevamento dei dati dalla memoria e posizionamento nei registri a d.p. della ALU, effettuando uno shift della mantissa
- $x=0.9983x10^{2} \rightarrow 0.99830000x10^{2}$
- $y=0.4652x10^{-1} \rightarrow 0.00046520x10^{2}$

3: somma delle mantisse nella ALU
- $0.99876520x10^{2}$

4: memorizzazione (con arrotondamento) del risultato e eventuale normalizzazione
- $z^{*} = x +_{fp} y = 0.9988x10^{2}$



Esecuzione di una somma f.p.

I registri della ALU a d.p. garantiscono una maggiore accuratezza

Errori di round off nelle operazioni aritmetiche

$z = x + y = 0.9987652\ x10^2$ $\qquad$ $z^* = x +_{fp} y = 0.9988x10^2$

Osserviamo che, a causa dei registri a d.p. della ALU
$z^* = fl(x+y) = fl(z)$

Quindi

$$E_R = |z - z^*| / |z| = |z - fl(z)| / |z| \leq b^{1-t}/2 = u$$

In generale, indicata con
-Op una operazione aritmetica
-Op_{fp} la sua corrispondente operazioni in aritmetica a precisione finita

$$E_R = |(x\ Op\ y) - (x\ Op_{fp}\ y)| / |x\ Op\ y| \leq b^{1-t}/2 = u$$

Inoltre

$$(x\ Op_{fp}\ y) = (x\ Op\ y)\ (1+\delta) \quad con \quad |\delta| \leq u$$

Due osservazioni

$F = \{ b=10 , t=4 , Emin=-9 , Emax=9 \}$

1. $a = 0.5496x10^2$ $\quad$ $b = 0.8714x10^1$ $\quad$ $c = 0.1493x10^{-1}$

$(a +_{fp} b) +_{fp} c = 0.6367x10^2 +_{fp} 0.1493x10^{-1} = 0.6368x10^2$
$a +_{fp} (b +_{fp} c) = 0.5496x10^2 +_{fp} 0.8729x10^1 = 0.6369x10^2$

La proprieta' associativa dell'addizione non vale

2. $a = 0.2240x10^2$ $\quad$ $b = 0.7653x10^1$ $\quad$ $c = 0.3329x10^2$

$(a +_{fp} b)\ x_{fp} c = 0.3035x10^2 +_{fp} 0.3329x10^2 = 0.1010x10^4$
$(a\ x_{fp} c) +_{fp} (b\ x_{fp} z) = 0.7457x10^3 +_{fp} 0.2648x10^3 = 0.1011x10^4$

La proprieta' distributiva della moltiplicazione rispetto all'addizione non vale

Esempio

$F = \{ b=10 , t=3 , Emin=-9 , Emax=9 \}$

Si vuole eseguire

$$1 +_{fp} x = fl(1+x) \text{ con } x = 0.436x10^{-2}$$

Passo 1: (confronto esponenti) $d=3$

Passo 2: (shift esponenti) $1 = 0.100000x10^{1}$ $x = 0.000436x10^{1}$

Passo 3: (somma mantisse) $0.100000x10^{1} + 0.000436x10^{1} = 0.100436x10^{1}$

Passo 4: (arrotondamento e memorizzazione del risultato): $1 +_{fp} x = 0.100x10^{1}$

CIOE'

$$\mathbf{1 +_{fp} x = fl(1+x) = 1}$$

Epsilon macchina

In un sistema aritmetico floating point esiste un insieme di numeri che non fornisce contributo alla somma con 1

Definizione:

In un sistema aritmetico f.p., il piu' piccolo numero ε tale che
$1 +_{fp} \varepsilon = fl(1+ \varepsilon) > 1$

e' detto **epsilon macchina**

Si verifica facilmente che:

$$\varepsilon = u = b^{1-t} /2$$

Un algoritmo per l'epsilon macchina

Conoscere ε equivale a conoscere t

Un modo per calcolare ε e' basato su un algoritmo iterativo che ad ogni passo

- Divide un valore a per 2
- Verifica se $1+a > 1$

La procedura non ha parametri di input
Restituisce il valore dell'epsilon macchina

Esecuzione in aritmetica standard IEEE
- s.p. eps=0.1192093 $x10^{-6}$
- d.p. eps=0.2220446 $x10^{-15}$

N.B. La divisione per 2 (la base) riduce gli errori di r.o.

```
procedure epsmac(out: eps)
var: eps, a, s  : real

a=1
repeat
  eps=a
  a=a/2
  s=1+a
until (s==1)

end procedure
```

Procedura per il calcolo dell'epsilon macchina

In generale

Dato un numero f.p. $x>0$, ci si chiede qual'e' il piu' piccolo numero $y>0$ tale che

$$x +_{fp} y = fl(x + y) > x$$

Dividendo per x si ottiene

$$1 +_{fp} y/x = fl(1 + y/x) > 1$$

Ci si e' ricondotti alla definizione di epsilon macchina, quindi

$$y/x = \varepsilon \quad \text{da cui} \quad y = \varepsilon|x|$$

Una applicazione dell'epsilon macchina

Si vuole calcolare con un algoritmo il limite della successione:

$0.5 , 0.75 , 0.875 , \ldots , (2^n-1)/2^n$

Tale successione e' equivalente alla somma

$0.5 + 0.25 + 0.125 + \ldots$

Si vuole

- la massima accuratezza
- la minima complessita' computazionale

IDEA:

Utilizzare una struttura repeat e interrompere quando il termine generico non fornisce contributo

Criterio di arresto naturale

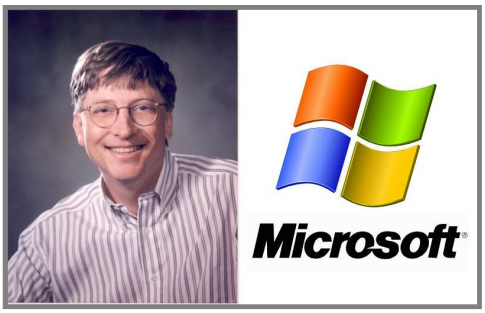
```
begin somma
var: eps, a, sum  : real

epsilon(eps)
a=1
sum=0
repeat
  a=a/2
  sum=sum+a
until (a < sum*eps)

end somma
```

Un semplice esempio di criterio di arresto naturale

Un po' di storia (13)

Bill Gates (1955) e Steve Jobs (1955-2011)

- B.Gates si interessa all'informatica a 13 anni, usando il computer della scuola. Si iscrive ad Harward nel 1973 e nel 1975 fonda la Microsoft assieme a Paul Allen
- Inizialmente sviluppa ambienti BASIC e sistemi operativi per piccoli sistemi Altair, DEC e IBM. Nel 1983 sviluppa la prima versione di Windows, il s.o. piu' usato al mondo negli anni '80 e '90 del XX sec.
- Per anni considerato l'uomo piu' ricco del mondo, dal 2008 dirige una fondazione umanitaria



Bill Gates negli anni '80
(courtesy of Computer History Museum)

- S.Jobs costruisce i primi computer nel garage di casa a 20 anni con Steve Wozniak. Fondano la Apple computer nel 1976 per vendere i loro prodotti.
- Da sempre convinto della stretta integrazione tra hardware e software e della centralita' dell'utente nell'utilizzo delle tecnologie, progetta prodotti di grande innovazione (dal Macintosh fino all'iPad e all'iPhone)
- E' anche tra i fondatori della Pixar Studios che ha rivoluzionato la produzione dei cartoni animati Disney.

- B.Gates e S.Jobs, concorrenti ma anche amici, realizzano l'obiettivo di "portare un computer in ogni casa", trasformando profondamente la societa'

Steve Jobs negli anni '80