Laboratorio di Calcolo Numerico

Aritmetica di macchina. Stabilità degli algoritmi.

Ángeles Martínez Calomardo
http://www.math.unipd.it/~acalomar/DIDATTICA/
angeles.martinez@unipd.it

Laurea in Informatica
A.A. 2018–2019

Lo standard ANSI IEEE-754r

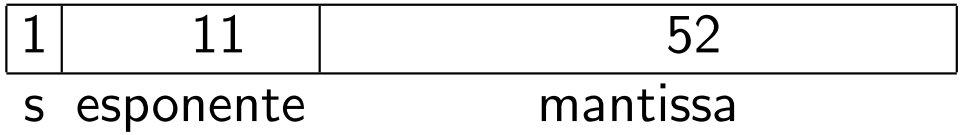
- Scritto nel 1985 e modificato nel 1989 e, più recentemente, nel 2008 costituisce lo standard ufficiale per la rappresentazione **binaria** dei numeri all'interno del calcolatore e l'aritmetica di macchina (il nome dello standard in inglese "Binary floating point arithmetic for microprocessor systems").
- Secondo lo standard un numero non nullo normalizzato si scrive come

$$x = (-1)^s \cdot (1 + f) \cdot 2^{e^* - bias}.$$

- La mantissa si rappresenta dunque come $1.d_1 d_2 \ldots d_\tau$ essendo $f = 0.d_1 d_2 \ldots d_\tau$.
- τ identifica il numero di bit usato per codificare la parte frazionaria della mantissa. Il numero di cifre totali per la mantissa è $t = \tau + 1$.
- Il vero esponente del numero e si immagazzina in traslazione come $e* = e + bias$.
- In questa maniera non serve un bit di segno per l'esponente.
- Il *bias* in singola precisione vale 127 mentre in doppia 1023.

Doppia precisione

- Ogni numero macchina occupa 64 bit, distribuiti nei tre campi segno, esponente e mantissa come segue:



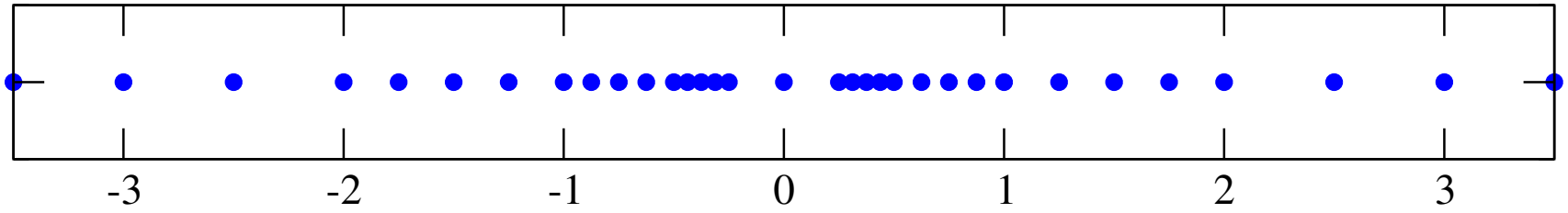
- L'insieme di numeri macchina in doppia precisione è: $F(2, 53, -1022, 1023)$.
- Con 52 bit si codificano 53 cifre della mantissa (1 bit nascosto).
- Dei $2^{11} = 2048$ esponenti possibili, 2 si riservano per usi speciali.
- I numeri rappresentabili in doppia precisione sono $2 \cdot (U - L + 1) \cdot (B - 1) \cdot B^{t-1} + 1 = 2 \cdot 2046 \cdot 1 \cdot 2^{52} \approx 1.8438 \cdot 10^{19}$.

Distanza assoluta tra due numeri macchina consecutivi

- L'insieme $\mathbb{R}$ dei numeri reali è denso.
- Il sottoinsieme dei numeri macchina $\mathcal{F}$, oltre ad essere limitato inferiormente e superiormente, è anche *bucato*, ovvero attorno ad ogni elemento x di $\mathcal{F}$ esiste un piccolo intervallo vuoto, tra x e il suo successivo numero macchina x_+.
- Essendo $x = (-1)^s \cdot (1 + 0.d_1 d_2 d_3 \cdots d_\tau) \cdot 2^E$ e $x_+ = (-1)^s \cdot (1 + 0.d_1 d_2 d_3 \cdots d_\tau + 1) \cdot 2^E$, la **distanza assoluta** tra x e x_+ è:

$$\Delta x = |x - x_+| = 2^{-52} \cdot 2^E.$$

- Questo valore in MATLAB/OCTAVE si ottiene scrivendo `eps(x)`.
- Si noti che questa distanza è uguale per tutti i numeri macchina aventi lo stesso esponente.
- I numeri macchina sono più addensati quanto più piccoli sono e la loro separazione aumenta man mano che aumenta il loro valore assoluto.



Sulla distanza assoluta tra numeri macchina

Esempi

La distanza $|x - x_+|$ determina l'ordine di grandezza del minor numero che sommato ad un numero macchina x fornirà come risultato un numero maggiore di x. Ad esempio:

```
>> 1+eps(1)
ans =   1.00000000000000e+00

>> 1+eps(1) -1
ans =   2.22044604925031e-16

>> 1000+eps(1)
ans =   1.00000000000000e+03

>> 1000+eps(1)-1000
ans =   0.00000000000000e+00
```

vediamo che il calcolatore non è in grado di interpretare come un incremento diverso da zero il numero eps(1) quando viene sommato a 1000, mentre lo riconosce come numero non nullo quando viene sommato a 1. Il più piccolo incremento del numero 1000 riconosciuto dal calcolatore e dell'ordine di eps(1000):

```
>> 1000+eps(1000)
ans =   1.00000000000000e+03
>> 1000+eps(1000)-1000
ans =   1.13686837721616e-13
```

Sulla distanza assoluta tra numeri macchina

Esercizio

Esercizio

Al variare di $x = 10^{-1}, 10^{-2}, 10^{-3}, \cdots, 10^{-15}$, *si calcoli* $(1 + x) - 1$. *Si confronti il valore numerico calcolato con* MATLAB/OCTAVE *con il valore esatto, cioè* x *(impostare il formato di visualizzazione a* `format long e` *).*

Calcolare l'errore relativo e commentare l'andamento degli errori al variare di x.

Qual è la percentuale dell'errore relativo per $x = 10^{-15}$ *?*

Distanza relativa e precisione di macchina

- La **distanza relativa** tra x e il suo elemento successivo x_+ si ottiene dividendo quella assoluta per il numero x:

$$\frac{|x - x_+|}{|x|} = \frac{2^{e-\tau}}{p \cdot 2^e} = \frac{2^{-\tau}}{p}.$$

- Si può vedere che la distanza relativa tra due numeri macchina consecutivi ha un andamento periodico.
- La massima distanza relativa tra due numeri macchina consecutivi è:

$$\epsilon_M = 2^{-\tau}$$

  che si ottiene quando la mantissa p è uguale a 1.
- Nello standard IEEE 754 in doppia precisione $\epsilon_M = 2^{-52}$.
- La precisione di macchina definita anteriormente come il massimo errore relativo di arrotondamento, coincide anche con

$$\mathbf{u} = \frac{\epsilon_M}{2}.$$

- Infatti in doppia precisione secondo lo standard IEEE 754 il valore della precisione di macchina u è pari a $2^{-53} \approx 1.11 \cdot 10^{-16}$, poiché $\epsilon_M = 2^{-52}$.
- In MATLAB/OCTAVE $\epsilon_M = 2^{-52}$ è rappresentato dalla variabile `eps`

Aritmetica di macchina e propagazione degli errori

Domanda: Come si propagano gli errori di rappresentazione quando si effettuano delle operazioni aritmetiche con i numeri macchina?

- Si può dimostrare che

$$\begin{aligned} \epsilon_{x,y}^{\oplus} &\leq \left|\frac{x}{x+y}\right|\epsilon_x + \left|\frac{y}{x+y}\right|\epsilon_y \\ \epsilon_{x,y}^{\otimes} &\lessapprox \epsilon_x + \epsilon_y \\ \epsilon_{x,y}^{\oslash} &\leq |\epsilon_x - \epsilon_y| \end{aligned}$$

  dove ϵ_x e ϵ_y sono tali che $\epsilon_x, \epsilon_y < \mathbf{u}$

- Le operazioni macchina prodotto e divisione introducono un errore dell'ordine della precisione di macchina.
- Con la somma (sottrazione) non si può garantire che il risultato dell'operazione sia affetto da un errore relativo piccolo. In particolare l'errore per la somma è grande quando $x \approx -y$ (**Cancellazione numerica**).

Cancellazione numerica e stabilità di un algoritmo

Definizione

Un metodo numerico (formula, algoritmo) si dice **stabile** *se non propaga gli errori (inevitabili) dovuti alla rappresentazione dei numeri nel calcolatore. Altrimenti si dice* **instabile**.

- La cancellazione numerica genera delle formule instabili.
- Per evitare i problemi legati alla cancellazione numerica occorre trasformare le formule in altre numericamente più stabili.
- La stabilità è un concetto legato all'algoritmo usato per risolvere un determinato problema.

Esempio di cancellazione numerica

Formula risolutiva dell'equazione di secondo grado

- Si vuole risolvere l'equazione $ax^2 + bx + c = 0$.
- Se $a \neq 0$ le radici sono:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Quando $4ac \ll b^2$ sarà $b^2 - 4ac \approx b^2$ e quindi $\sqrt{b^2 - 4ac} \approx |b|$.
- Come conseguenza si avrà cancellazione numerica nel calcolo di quella radice in cui si sottraggono due numeri quasi uguali:

$$\text{In } x_1 \quad \text{se} \quad b > 0 \text{ poichè } \sqrt{b^2 - 4ac} \approx b \longrightarrow x_1 \approx \frac{-b + b}{2a}$$

$$\text{In } x_2 \quad \text{se} \quad b < 0 \text{ poichè } \sqrt{b^2 - 4ac} \approx -b \longrightarrow x_2 \approx \frac{-b - (-b)}{2a}$$

- Per risolvere il problema si calcola una radice con una **formula stabile** nella quale non si manifesta cancellazione numerica

$$x_1 = \frac{-b - sign(b)\sqrt{b^2 - 4ac}}{2a}$$

e si ricava l'altra radice da $\frac{c}{a} = x_1 \cdot x_2$

Calcolo π

Eseguiamo un codice Matlab che valuti le successioni $\{u_n\}$, $\{z_n\}$, definite rispettivamente come

$$\begin{cases} s_1 = 1, \ s_2 = 1 + \frac{1}{4} \\ u_1 = 1, \ u_2 = 1 + \frac{1}{4} \\ s_{n+1} = s_n + \frac{1}{(n+1)^2} \\ u_{n+1} = \sqrt{6\, s_{n+1}} \end{cases}$$

e

$$\begin{cases} z_1 = 1, \ z_2 = 2 \\ z_{n+1} = 2^{n-\frac{1}{2}} \sqrt{1 - \sqrt{1 - 4^{1-n} \cdot z_n^2}} \end{cases} \tag{1}$$

che *teoricamente* convergono a π.

Calcolo π

Implementiamo poi la successione, diciamo $\{y_n\}$, che si ottiene *razionalizzando* (1), cioè moltiplicando numeratore e denominatore di

$$z_{n+1} = 2^{n-\frac{1}{2}} \sqrt{1 - \sqrt{1 - 4^{1-n} \cdot z_n^2}}$$

per

$$\sqrt{1 + \sqrt{1 - 4^{1-n} \cdot z_n^2}}$$

e calcoliamo u_m, z_m e y_m per $m = 2, 3, \ldots, 40$ (che teoricamente dovrebbero approssimare π).
Infine disegniamo in un unico grafico l'andamento dell'errore relativo di u_n, z_n e y_n rispetto a π aiutandoci con l'help di Matlab relativo al comando `semilogy`.

Calcolo π: metodo 1

In seguito scriviamo un'implementazione di quanto richiesto commentando i risultati. Si salvi in un file `pigreco.m` il codice

```
% SEQUENZE CONVERGENTI "PI GRECO".

% METODO 1.
s(1)=1; u(1)=1;
s(2)=1.25; u(2)=s(2);
for n=2:40
    s(n+1)=s(n)+(n+1)^(-2);
    u(n+1)=sqrt(6*s(n+1));
end
rel_err_u=abs(u-pi)/pi;

fprintf('\n');
```

Calcolo π: metodo 2

```
% METODO 2.
format long
z(1)=1;
z(2)=2;
for n=2:40
    c=(4^(1-n)) * (z(n))^2; inner_sqrt=sqrt(1-c);
    z(n+1)=(2^(n-0.5))*sqrt( 1-inner_sqrt );
end
rel_err_z=abs(z-pi)/pi;

fprintf('\n');
```

Calcolo π: metodo 3

```
% METODO 3.
y(1)=1;
y(2)=2;
for n=2:40
    num=(2^(1/2)) * abs(y(n));
    c=(4^(1-n)) * (y(n))^2;
    inner_sqrt=sqrt(1-c);
    den=sqrt( 1+inner_sqrt );
    y(n+1)=num/den;
end
rel_err_y=abs(y-pi)/pi;
```

Calcolo π: plots

```
% SEMILOGY PLOT.
semilogy(1:length(u),rel_err_u,'k.');
hold on;
semilogy(1:length(z),rel_err_z,'m+');
semilogy(1:length(y),rel_err_y,'ro');
hold off;
```

Di seguito digitiamo sulla shell di Matlab/Octave

```
>> pigreco
```

Plot risultati

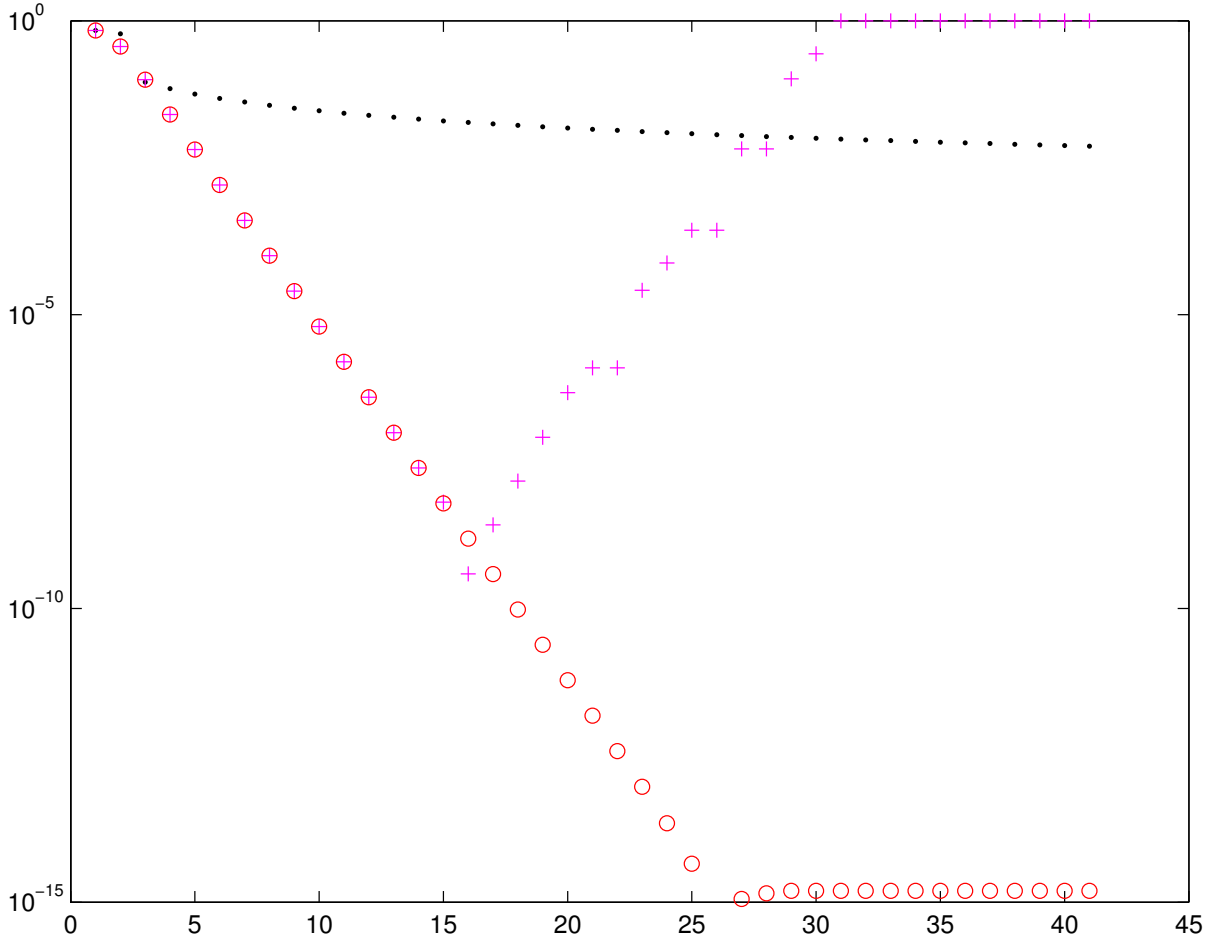


Figura: Errore relativo commesso con le 3 successioni, rappresentate rispettivamente da ., + e o.

Discussione risultati

- La prima successione converge molto lentamente a π, la seconda diverge mentre la terza converge velocemente a π.
- Per alcuni valori $\{z_n\}$ e $\{y_n\}$ coincidono per alcune iterazioni per poi rispettivamente divergere e convergere a π. Tutto ciò è naturale poichè le due sequenze sono analiticamente (ma non numericamente) equivalenti.
- Dal grafico dell'errore relativo, la terza successione, dopo aver raggiunto errori relativi prossimi alla precisione di macchina, si assesta ad un errore relativo di circa 10^{-15} (probabilmente per questioni di arrotondamento).

L'algoritmo 2 in dettaglio

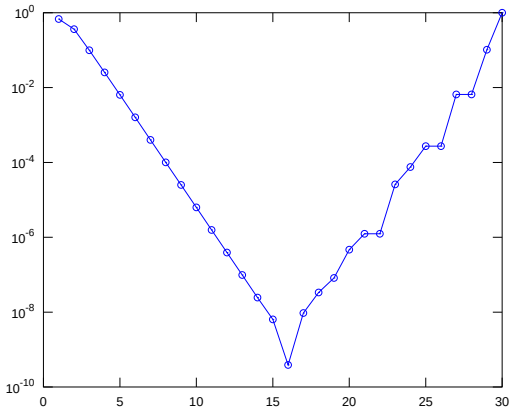
Successione approssimante π

Nell'approssimare il valore di π con la formula ricorsiva

$$\begin{aligned} z_2 &= 2 \\ z_{n+1} &= 2^{n-0.5}\sqrt{1-\sqrt{1-4^{1-n}z_n^2}}, \quad n = 2, 3, \ldots, \end{aligned}$$

si ottiene la seguente successione di valori (dove si è posto $c = 4^{1-n}z_n^2$).

$n+1$	c	$1-\sqrt{1-c}$	z_{n+1}	$\frac{\lvert z_{n+1}-\pi\rvert}{\pi}$
...	...	...	...	...
10	1.505e-04	7.529e-05	3.14157294036	6.27e-06
11	3.764e-05	1.882e-05	3.14158772527	1.57e-06
12	9.412e-06	4.706e-06	3.14159142150	3.92e-07
13	2.353e-06	1.176e-06	3.14159234561	9.80e-08
14	5.882e-07	2.941e-07	3.14159257654	2.45e-08
15	1.470e-07	7.353e-08	3.14159263346	6.41e-09
16	3.676e-08	1.838e-08	3.14159265480	3.88e-10
17	9.191e-09	4.595e-09	3.14159264532	2.63e-09
18	2.297e-09	1.148e-09	3.14159260737	1.47e-08
19	5.744e-10	2.872e-10	3.14159291093	8.19e-08
...	...	...	...	...
28	2.220e-15	1.110e-15	3.16227766016	6.58e-03
29	5.551e-16	3.330e-16	3.46410161513	1.03e-01
30	1.665e-16	1.110e-16	4.00000000000	2.73e-01
31	5.551e-17	0.000e+00	0.00000000000	1.00e+00
32	0.000e+00	0.000e+00	0.00000000000	1.00e+00



Una successione ricorrente.

Consideriamo la successione $\{I_n\}$ definita da

$$I_n = e^{-1} \int_0^1 x^n \, e^x \, dx \tag{2}$$

- $n = 0$: $I_0 = e^{-1} \int_0^1 e^x \, dx = e^{-1}(e^1 - 1)$.
- integrando per parti

$$\begin{aligned} I_{n+1} &= e^{-1} \left(x^{n+1} \, e^x \, |_0^1 - (n+1) \int_0^1 x^n \, e^x \, dx \right) \\ &= 1 - (n+1) \, I_n . \end{aligned}$$

- $I_n > 0$, decrescente e si prova che $I_n \to 0$ come $1/n$.

Problema.

Calcoliamo I_n per $n = 1, \ldots, 100$:

- mediante la successione *in avanti*

$$\begin{cases} I_0 = e^{-1}(e^1 - 1) \\ I_{n+1} = 1 - (n+1)\, I_n. \end{cases} \tag{3}$$

- mediante la successione *all'indietro*

$$\begin{cases} I_{1000} = 0 \\ I_{n-1} = (1 - I_n)/n. \end{cases}$$

Si noti che se $I_{n+1} = 1 - (n+1)\, I_n$ allora $I_n = (1 - I_{n+1})/(n+1)$ e quindi $I_{n-1} = (1 - I_n)/n$.

Successione ricorrente IMPLEMENTAZIONE in Matlab

- Scriviamo il codice in un file `succricorrente.m`.
- Occorre salvare i valori calcolati in due vettori chiamati **s** (successione in avanti) e **t** (successsione all'indietro).
- Per la successione in avanti partire da I_1 anziché da I_0.
- Per la successione all'indietro partire da $I_{1000} = 0$.
- Creare un grafico semilogaritmico con i valori $I_1, I_2, \ldots, I_{100}$ calolati dai due algoritmi che, ricordiamo, sono matematicamente equivalenti ma non numericamente equivalenti.