ARITMETICA, ERRORI

Letizia SCUDERI

Dipartimento di Scienze Matematiche, Politecnico di Torino

letizia.scuderi@polito.it

A.A. 2017/2018

Il **Calcolo Numerico** è una disciplina della matematica, che propone e analizza **metodi** che consentono di ottenere una **soluzione numerica** di problemi matematici, per i quali metodi di risoluzione analitica non esistono oppure sono eccessivamente onerosi.

Per ottenere la soluzione numerica si utilizzano **algoritmi**, eseguibili da un calcolatore e dedotti dai metodi stessi.

In seguito verranno mostrati metodi per risolvere numericamente alcuni problemi di Approssimazione e di Algebra Lineare.

Per gli algoritmi, che implementano i suddetti metodi, verrà usato il linguaggio di programmazione **Matlab**®.

Uno dei più grandi errori, da evitare quando si programma al calcolatore, è pensare che i risultati numerici ottenuti siano privi di errori oppure affetti da un errore trascurabile.

Alcuni disastri, ormai passati alla storia, sono dovuti proprio ad un uso scorretto del Calcolo Numerico!

Diverse pagine web[1] riportano alcune di queste storie, per esempio,

- nel 1991, durante la prima guerra del Golfo Persico, la caduta su una caserma americana di un missile Scud iracheno, a causa di una mancata intercettazione da parte di un missile americano Patriot;
- nel 1996, l'esplosione in aria del razzo Ariane 5 dopo appena 40 secondi dal lancio.

[1] **http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html**

Esempio

Un esempio con conseguenze meno catastrofiche, ma comunque "inquietanti", è costituito dalla seguente successione:

$$x_1 = 2$$
$$x_n = 2^{n-1/2}\sqrt{1-\sqrt{1-4^{1-n}x_{n-1}^2}} \qquad n \geq 2$$

convergente a π.

In teoria, per definizione di limite di una successione, x_n deve avvicinarsi sempre di più a π al crescere di n.

... segue esempio

Nella pratica, se si implementa l'algoritmo per il calcolo dei termini della successione, al crescere di n, accade che dapprima i termini della successione si avvicinano a π ma, successivamente, a partire da $n = 16$, cominciano ad allontanarsene, come mostrano i valori riportati in tabella.

n	$\pi \approx 3.141592653589793$
1	2.000000000000000
2	2.828427124746190
3	3.061467458920717
4	3.121445152258050
5	3.136548490545931
6	3.140331156954739
7	3.141277250932756
8	3.141513801144145
9	3.141572940365566
10	3.141587725279960
11	3.141591421504635
12	3.141592345611076
13	3.141592576545004
14	3.141592633463248
15	3.141592654807589

n	$\pi \approx 3.141592653589793$
16	3.141592645321215
17	3.141592607375720
18	3.141592910939672
19	3.141594125195191
20	3.141596553704819
21	3.141596553704819
22	3.141674265021757
23	3.141829681889201
24	3.142451272494133
25	3.142451272494133
26	3.162277660168379
27	3.162277660168379
28	3.464101615137754
29	4.000000000000000
30	0

Addirittura si ha che $x_{30} = 0$!!

Rappresentazione dei numeri

Per meglio comprendere il fenomeno che si è verificato, è opportuno illustrare le principali regole che i calcolatori seguono per memorizzare i numeri e operare con essi.

Definizione

Si definisce **rappresentazione floating-point** di un numero reale a la seguente rappresentazione

$$a = (-1)^s p N^q, \ s \in \{0, 1\}, \ p \geq 0 \text{ reale}, \ q \text{ intero},$$

ove N rappresenta la **base** del sistema di numerazione.

Se $N = 10$ il sistema di numerazione si dice **decimale**, se $N = 2$ il sistema si dice **binario**.

Esempio

Per $N = 10$, si ha $a = 0.015 \cdot 10^{-1} = 0.15 \cdot 10^{-2} = 0.0015 \cdot 10^{0}$.

L'esempio evidenzia che ogni numero reale (non nullo) **non ammette un'unica rappresentazione floating-point**. Tale rappresentazione è univocamente determinata, se si impone la seguente condizione

$$N^{-1} \leq p < 1$$

cioè se si impone che la prima cifra di p, dopo il punto decimale, sia diversa da zero.

Per $N = 10$, la condizione precedente diventa $0.1 \leq p < 1$.

Definizione

La rappresentazione floating-point $a = (-1)^s p N^q$ del numero reale a si dice **normalizzata** se p soddisfa la condizione

$$N^{-1} \leq p < 1$$

In tal caso, p e q sono univocamente determinati e

- il numero reale non negativo p si definisce **mantissa** di a;
- l'intero q si definisce **esponente** oppure **caratteristica** di a.

Esempi

Sia $N = 10$.

1. $a = 0.15 \cdot 10^{-2}$, la mantissa è $p = 0.15$, l'esponente è $q = -2$;
2. $a = 12.4 \cdot 10^{0} = 0.124 \cdot 10^{2}$, la mantissa è $p = 0.124$, l'esponente è $q = 2$;
3. $a = -0.0013 \cdot 10^{4} = -0.13 \cdot 10^{2}$, la mantissa è $p = 0.13$, l'esponente è $q = 2$.

Fissata la base N del sistema di numerazione, i valori di s, p, q della rappresentazione floating-point normalizzata $a = (-1)^s pN^q$ individuano **univocamente** il numero reale a. Pertanto, per memorizzare a, è sufficiente memorizzare s, p e q.

Poiché un calcolatore riserva per la memorizzazione di tali quantità uno **spazio finito di memoria**, esso può memorizzare solo mantisse con un numero finito di cifre ed esponenti appartenenti a un certo intervallo.

Si supponga, per esempio, che

- le mantisse p rappresentabili possano avere al massimo t cifre nel sistema di numerazione scelto;
- l'esponente q debba soddisfare le limitazioni $L \leq q \leq U$ con $L < 0$ e $U > 0$ interi.

Numeri di macchina

È evidente che non tutti i numeri reali soddisfino le suddette condizioni. Pertanto, **non tutti i numeri reali sono esattamente rappresentabili su di un calcolatore**.

Definizione

Si definiscono **numeri di macchina** i numeri con mantissa ed esponente esattamente rappresentabili negli spazi a loro riservati dal calcolatore.

L'**insieme dei numeri macchina** è costituito da un numero finito di elementi ed è così definito:

$$\mathcal{F} = \{0\} \cup \{(-1)^s 0.a_1 a_2 ... a_t \cdot N^q, 0 \leq a_i < N, a_1 \neq 0, L \leq q \leq U\}$$

ove

- N è la base del sistema di numerazione;
- s vale 0 (segno positivo) oppure 1 (segno negativo);
- $a_1, a_2, ..., a_t$ sono interi minori o uguali a $N - 1$ e rappresentano le cifre della mantissa[2];
- t è il massimo numero di cifre della mantissa rappresentabili;
- q è l'esponente.

[2]Da qui in avanti, con il termine *cifre della mantissa* $p = 0.a_1 a_2 ... a_t$ si intenderanno le cifre $a_1, a_2, ..., a_t$ che seguono la virgola.

Esempi

Siano $N = 10$, $t = 5$, $L = -127$, $U = 128$. Nell'aritmetica floating-point fissata,

1. $a = 1.58291 \cdot 10^0 = 0.158291 \cdot 10^1$
   non è un numero di macchina;
2. $a = 0.0038245 \cdot 10^0 = 0.38245 \cdot 10^{-2}$
   è un numero di macchina;
3. $a = 12.29 \cdot 10^{128} = 0.1229 \cdot 10^{130}$
   non è un numero di macchina.

Di seguito verrà descritto come un calcolatore tratti i numeri reali che non soddisfano le limitazioni sull'esponente.

La limitazione inferiore $L \leq q$ sull'esponente q comporta che il più piccolo numero positivo, rappresentabile come numero di macchina, sia

$$m = 0.\underbrace{10000000}_{t \text{ cifre}} \cdot N^L$$

Definizione

Si definisce **regione di underflow** l'insieme dei numeri reali diversi da zero e appartenenti a $(-m, m)$.

Esempio

Per $N = 10$, $t = 8$ e $L = -125$, il più piccolo numero positivo rappresentabile è $m = 0.10000000 \cdot 10^{-125}$.

La limitazione superiore $q \leq U$ sull'esponente q comporta che il più grande numero positivo, rappresentabile come numero di macchina, sia

$$M = 0.\underbrace{N-1\,N-1\ldots N-1}_{t \text{ cifre}} \cdot N^U$$

Definizione

Si definisce **regione di overflow** l'insieme dei numeri appartenenti a $(-\infty, -M) \cup (M, +\infty)$.

Esempio

Per $N = 10$, $t = 8$ e $U = 128$, il più grande numero positivo rappresentabile è $M = 0.99999999 \cdot 10^{128}$.

Anche se i dati iniziali e il risultato finale di un processo di calcolo sono numeri di macchina, o comunque arrotondabili a numeri di macchina, le operazioni intermedie possono dare origine a numeri che appartengono alla regione di underflow oppure di overflow.

Quando ciò succede il calcolatore invia all'utente una segnalazione dell'evento: nel primo caso, il risultato parziale viene approssimato con lo zero; nel secondo caso, invece, il risultato viene approssimato con il simbolo *Inf*. Pertanto, in quest'ultimo caso, per ottenere il risultato finale è necessario evitare la regione di overflow.

Gli esempi che seguono mostrano situazioni nelle quali è possibile evitare il verificarsi di fenomeni di overflow.

Esempio 1

Si consideri la media aritmetica

$$\frac{\overline{M} + M}{2}$$

di $\overline{M} = M/2$ e M, ove M è il più grande numero di macchina.

Il risultato è minore di M e quindi può essere rappresentato, eventualmente in maniera approssimata, come numero di macchina. Se però si calcola la media come somma $\overline{M} + M$ seguita dalla divisione per 2, allora $\overline{M} + M$ è maggiore di M, e dunque non è rappresentabile come numero di macchina.

Per ovviare a questo problema, basta calcolare la media aritmetica come

$$\frac{\overline{M}}{2} + \frac{M}{2} \quad \text{oppure} \quad \overline{M} + \frac{M - \overline{M}}{2}$$

Esempio 2

Si consideri la quantità

$$R = \sqrt{x_1^2 + x_2^2}$$

Può accadere che, pur essendo x_1 e x_2 rappresentabili, x_1^2 e/o x_2^2 non lo siano. In questo caso si può ovviare al problema, calcolando R come

$$R = \begin{cases} |x_1|\sqrt{1 + \left(\dfrac{x_2}{x_1}\right)^2}, & \text{se } |x_1| > |x_2| \\ \\ |x_2|\sqrt{\left(\dfrac{x_1}{x_2}\right)^2 + 1}, & \text{se } |x_2| > |x_1| \end{cases}$$

Si esamina ora come vengano rappresentati i numeri reali, la cui mantissa ha più di t cifre.

Sia $a = (-1)^s p N^q$, con $N^{-1} \leq p < 1$ e $L \leq q \leq U$, e si supponga che p abbia più di t cifre.

Per rappresentare a in un sistema floating-point con t cifre per la mantissa, occorre in qualche modo "accorciare" o, meglio, *arrotondare* p a t cifre.

Generalmente si ricorre alla seguente tecnica.

Definizione

Tecnica di arrotondamento "rounding to even": la mantissa p viene approssimata con la mantissa di macchina $\bar{p}$ più vicina e se p è equidistante da due mantisse di macchina consecutive allora p viene approssimata con quella delle due che ha l'ultima cifra pari.

Il numero di macchina corrispondente al numero reale $a = (-1)^s pN^q$ nell'aritmetica floating-point fissata è allora dato da $\bar{a} = (-1)^s \bar{p} N^q$.

Operativamente, per calcolare $\bar{p}$, si aggiunge $1/2 \cdot N^{-t}$ a p e poi si tronca il risultato alla t-esima cifra, con un'eccezione nel caso in cui p sia equidistante da due mantisse di macchina consecutive.

In quest'ultimo caso, p viene approssimato per eccesso o per difetto a seconda che la mantissa di macchina più vicina e con la t-esima cifra pari segua o preceda p.

Pertanto, se

$$p = 0.a_1 a_2 \ldots a_t a_{t+1} \ldots a_n$$

e

$$p \in (\bar{p}_1, \bar{p}_2)$$

ove

$$\bar{p}_1 = 0.a_1 a_2 \ldots a_t \quad \text{e} \quad \bar{p}_2 = \bar{p}_1 + N^{-t}$$

denotano due mantisse di macchina consecutive, allora il numero di macchina $\bar{p}$ corrispondente a p è dato da

$$\bar{p} = \begin{cases} \bar{p}_1, \text{ se } p \in \left(\bar{p}_1, \dfrac{\bar{p}_1 + \bar{p}_2}{2}\right) \text{ oppure } p = \dfrac{\bar{p}_1 + \bar{p}_2}{2} \text{ e } a_t \text{ è } \textit{pari} \\ \bar{p}_2, \text{ se } p \in \left(\dfrac{\bar{p}_1 + \bar{p}_2}{2}, \bar{p}_2\right) \text{ oppure } p = \dfrac{\bar{p}_1 + \bar{p}_2}{2} \text{ e } a_t \text{ è } \textit{dispari} \end{cases}$$

Esempi

Siano $N = 10$ e $t = 3$.

- Se $p = 0.158\underline{1}4$, allora $\bar{p} = 0.158$.
- Se $p = 0.158\underline{5}432$, allora $\bar{p} = 0.159$.
- Se $p = 0.158\underline{8}12$, allora $\bar{p} = 0.159$.
- Se $p = 0.158\underline{5}$, allora $\bar{p} = 0.158$.
- Se $p = 0.159\underline{5}$, allora $\bar{p} = 0.160$.

In seguito con la notazione $\bar{a}$ verrà indicato il numero di macchina corrispondente ad a. Se a è un numero di macchina, allora $a = \bar{a}$; altrimenti $a \approx \bar{a}$.

È evidente che quando $a \approx \bar{a}$ e si memorizza $\bar{a}$ al posto di a, si commette un errore. Per quantificare tale errore, è necessario introdurre una definizione d'errore.

Definizione

Sia $\tilde{x}$ un'approssimazione del numero x. Si definisce **errore assoluto** associato a $\tilde{x}$ la quantità

$$e_a = |x - \tilde{x}|$$

ed **errore relativo** la quantità

$$e_r = \frac{|x - \tilde{x}|}{|x|}, \quad x \neq 0$$

Osservazione

Se x rappresenta un'entità diversa dal numero reale, per esempio una funzione, oppure un vettore o una matrice, e $\tilde{x}$ denota una sua approssimazione, è possibile dare una analoga definizione di errore assoluto/relativo. In tal caso, nella definizione occorre sostituire il valore assoluto $|\cdot|$ con una opportuna norma $||\cdot||$.

In generale, l'errore relativo fornisce una indicazione più realistica e significativa della distanza fra x e $\tilde{x}$, in quanto tiene conto dell'ordine di grandezza delle quantità in esame.

Esempio

Sia $N = 10$.

Se $x = 10^3$ e $\tilde{x} = 999$ si ha $e_a = 1$ ed $e_r = 10^{-3}$.

Se $x = 10^6$ e $\tilde{x} = 999999$ si ha nuovamente $e_a = 1$, mentre $e_r = 10^{-6}$ è più piccolo del valore e_r precedentemente determinato.

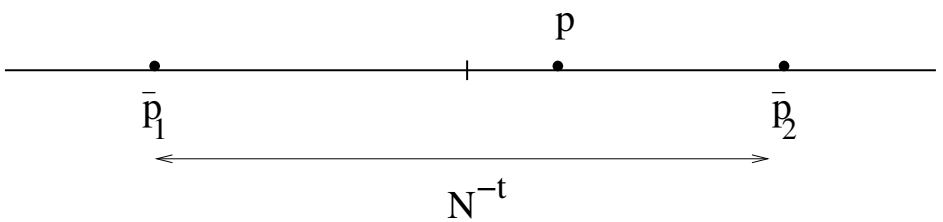
Definizione

Si definisce **errore di arrotondamento** l'errore che si commette quando si sostituisce il numero reale $a \neq 0$ con il corrispondente numero di macchina $\bar{a}$.

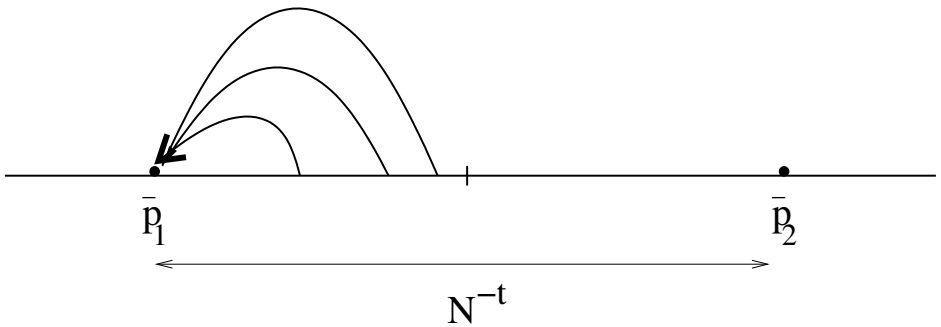
Si vuole determinare ora una stima dell'errore assoluto $|a - \bar{a}|$ e dell'errore relativo $|a - \bar{a}|/|a|$ di arrotondamento.

Sia $a = (-1)^s p N^q$ con $N^{-1} \leq p < 1$ e $\bar{a} = (-1)^s \bar{p} N^q$, dove $\bar{p}$ è stato ottenuto applicando a p la tecnica di arrotondamento precedentemente descritta; dunque, $\bar{p}$ non ha più di t cifre.

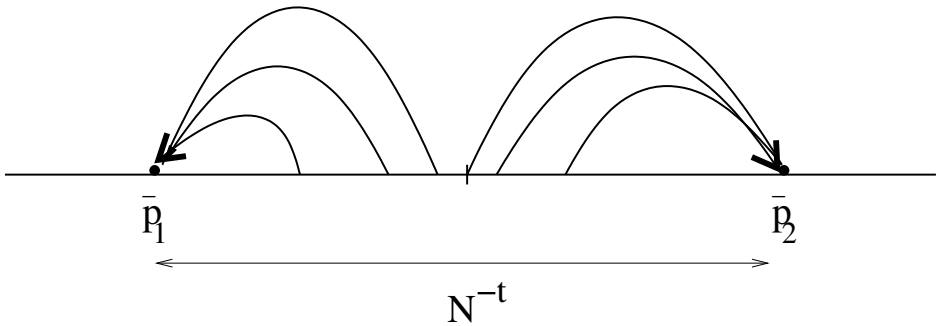
Si supponga che $p \in (\bar{p}_1, \bar{p}_2)$, ove $\bar{p}_1 = 0.a_1 \ldots a_t$ e $\bar{p}_2$ sono due mantisse di macchina consecutive. Tenendo conto che la distanza tra $\bar{p}_1$ e $\bar{p}_2$ è esattamente N^{-t}, si ha



- se $p \in \left(\bar{p}_1, \bar{p}_1 + \frac{1}{2}N^{-t}\right)$ oppure $p = \bar{p}_1 + \frac{1}{2}N^{-t}$ e a_t è pari, allora $\bar{p} = \bar{p}_1$



- se $p \in \left(\bar{p}_1 + \frac{1}{2}N^{-t}, \bar{p}_2\right)$ oppure $p = \bar{p}_1 + \frac{1}{2}N^{-t}$ e a_t è dispari, allora $\bar{p} = \bar{p}_2$



In entrambi i casi, si ha

$$|p - \bar{p}| \leq \frac{1}{2}N^{-t}$$

Pertanto, per l'errore assoluto risulta

$$|a - \bar{a}| = |p - \bar{p}| N^q \leq \frac{1}{2} N^{q-t}$$

e, tenendo conto che $p \geq N^{-1}$, per l'errore relativo risulta

$$\frac{|a - \bar{a}|}{|a|} = \frac{|p - \bar{p}|}{|p|} \leq \frac{1}{2} N^{1-t}$$

Definizione

Si definisce **epsilon di macchina** la quantità

$$eps = N^{1-t}$$

Si definisce **precisione di macchina** la quantità

$$\varepsilon_m = \frac{1}{2} N^{1-t}$$

La precisione di macchina è una costante caratteristica di ogni aritmetica floating-point; essa rappresenta il massimo errore relativo che si commette quando si approssima il numero reale a con il corrispondente numero di macchina $\bar{a}$.

Ponendo $\varepsilon = (\bar{a} - a)/a$, ove $\bar{a}$ è il numero di macchina corrispondente al numero reale a, si deduce la seguente uguaglianza

$$\bar{a} = a(1 + \varepsilon), \quad |\varepsilon| \leq \varepsilon_m$$

che esprime una relazione tra $\bar{a}$ e a.

Osservazione

Matlab implementa le specifiche del sistema standard floating point IEEE 754 del 1985 (divenuto lo standard internazionale IEC 559 del 1989), che in doppia precisione prevede:

la base $N = 2$, la tecnica di arrotondamento precedentemente descritta e 64 bit per la rappresentazione di un numero macchina, di cui

- 1 bit per rappresentare il segno della mantissa;
- 52 bit per rappresentare le cifre della mantissa p (in realtà i bit per la mantissa sono 53, in quanto non si rappresenta il primo bit, che è certamente 1);
- 11 bit per rappresentare l'esponente q.

... continua osservazione

In Matlab sono predefiniti i seguenti valori:

- `realmin` fornisce il valore del più piccolo numero di macchina positivo e non nullo $m \approx 2.2 \cdot 10^{-308}$;
- `realmax` fornisce il valore del più grande numero di macchina $M \approx 1.8 \cdot 10^{308}$;
- `eps` fornisce il valore dell'epsilon di macchina $eps \approx 2.2 \cdot 10^{-16}$ (la precisione di macchina è dunque $\varepsilon_m \approx 1.1 \cdot 10^{-16}$).

Si osservi che 53 cifre per la mantissa in base 2 corrispondono a circa 16 cifre per la mantissa in base 10.

Operazioni di macchina

Il risultato di un'operazione aritmetica tra due numeri di macchina generalmente non è un numero di macchina.

Esempio

Siano $N = 10$, $t = 4$ e

$$\bar{a}_1 = 0.5823 \text{ e } \bar{a}_2 = 0.6214$$

due numeri di macchina. La somma

$$\bar{a}_1 + \bar{a}_2 = 1.2037 = 0.12037 \cdot 10$$

non è un numero di macchina nell'aritmetica floating-point fissata.

In un calcolatore **non è possibile eseguire esattamente le operazioni aritmetiche** $+$, $-$, $\times$ e $/$, ma solo le cosiddette *operazioni di macchina*, che vengono rappresentate con i simboli $\oplus$, $\ominus$, $\otimes$ e $\oslash$.

Definizione

L'**operazione di macchina** $\odot$ associa a due numeri di macchina un terzo numero di macchina, ottenuto arrotondando l'esatto risultato dell'operazione in questione.

Ad esempio, se si denotano con $\bar{a}_1$ e $\bar{a}_2$ due numeri di macchina e con $\oplus$ l'operazione di macchina corrispondente all'operazione esatta di addizione, si ha

$$\bar{a}_1 \oplus \bar{a}_2 = \overline{\bar{a}_1 + \bar{a}_2}$$

Esempio

Siano $N = 10$, $t = 4$, $\bar{a}_1 = 0.5823$ e $\bar{a}_2 = 0.6214$. Il risultato dell'operazione di macchina corrispondente all'operazione di addizione è

$$\bar{a}_1 \oplus \bar{a}_2 = 0.1204 \cdot 10$$

Siano $\bar{a}_1$ e $\bar{a}_2$ due numeri di macchina e sia $\odot$ l'operazione di macchina corrispondente all'operazione $\cdot$ in aritmetica esatta. Si ha allora

$$\bar{a}_1 \oplus \bar{a}_2 = \overline{\bar{a}_1 + \bar{a}_2} = (\bar{a}_1 + \bar{a}_2)(1 + \varepsilon_\oplus)$$

$$\bar{a}_1 \ominus \bar{a}_2 = \overline{\bar{a}_1 - \bar{a}_2} = (\bar{a}_1 - \bar{a}_2)(1 + \varepsilon_\ominus)$$

$$\bar{a}_1 \otimes \bar{a}_2 = \overline{\bar{a}_1 \times \bar{a}_2} = (\bar{a}_1 \times \bar{a}_2)(1 + \varepsilon_\otimes)$$

$$\bar{a}_1 \oslash \bar{a}_2 = \overline{\bar{a}_1 / \bar{a}_2} = (\bar{a}_1 / \bar{a}_2)(1 + \varepsilon_\oslash)$$

con $|\varepsilon_\odot| \leq \varepsilon_m$.

Per le operazioni di macchina rimane valida la proprietà commutativa, ma non valgono in generale le proprietà associativa e distributiva.
Precisamente si ha

$$\bar{a}_1 \oplus \bar{a}_2 = \bar{a}_2 \oplus \bar{a}_1 \quad \text{p. commutativa}$$

$$\bar{a}_1 \otimes \bar{a}_2 = \bar{a}_2 \otimes \bar{a}_1 \quad \text{p. commutativa}$$

ma, in generale,

$$\bar{a}_1 \oplus (\bar{a}_2 \oplus \bar{a}_3) \neq (\bar{a}_1 \oplus \bar{a}_2) \oplus \bar{a}_3 \quad \text{p. associativa}$$

$$\bar{a}_1 \otimes (\bar{a}_2 \otimes \bar{a}_3) \neq (\bar{a}_1 \otimes \bar{a}_2) \otimes \bar{a}_3 \quad \text{p. associativa}$$

$$\bar{a}_1 \otimes (\bar{a}_2 \oplus \bar{a}_3) \neq (\bar{a}_1 \otimes \bar{a}_2) \oplus (\bar{a}_1 \otimes \bar{a}_3) \quad \text{p. distributiva}$$

$$(\bar{a}_1 \otimes \bar{a}_2) \oslash \bar{a}_3 \neq (\bar{a}_1 \oslash \bar{a}_3) \otimes \bar{a}_2 \quad \text{p. distributiva}$$

Un'ulteriore relazione anomala è

$$\bar{a}_1 \oplus \bar{a}_2 = \bar{a}_1$$

quando $|\bar{a}_2| \ll |\bar{a}_1|$, in particolare quando $|\bar{a}_2| < |\bar{a}_1| eps$.
Infatti, in tal caso, il valore di $|\bar{a}_2|$ è "troppo piccolo" rispetto a $|\bar{a}_1|$ e non fornisce alcun contributo nella somma.

Esempio

```
>> format long e
>> 5+1.0e-18
ans =
     5
>> 2017.3+1.0e-14
ans =

     2.017300000000000e+03
```

Pertanto, due espressioni/quantità e_1 ed e_2 equivalenti nell'aritmetica esatta, non risultano generalmente tali nell'aritmetica con precisione finita. Tuttavia, sussiste la seguente definizione.

Definizione

Due espressioni/quantità e_1 ed e_2 si definiscono **equivalenti nell'aritmetica del calcolatore** quando l'errore relativo

$$\frac{|e_1 - e_2|}{|e_1|} \quad \text{oppure} \quad \frac{|e_1 - e_2|}{|e_2|}$$

è dell'ordine della precisione di macchina ε_m o minore.

Ne consegue che ε_m rappresenta la massima precisione (relativa) di calcolo raggiungibile: non ha senso cercare di determinare approssimazioni con precisione relativa inferiore a ε_m.

Esempio 1

Per $N = 10$ e $t = 5$, i numeri 0.99999 e 1 sono equivalenti nell'aritmetica floating-point fissata, perché $|1 - 0.99999| = 0.1 \cdot 10^{-4} \leq \varepsilon_m \equiv 0.5 \cdot 10^{-4}$.

Esempio 2

Consideriamo le espressioni $y1 = (1 + x) - 1$ e $y2 = (1 - 1) + x$, equivalenti in aritmetica esatta qualunque sia il valore di x.

... continua esempio

Nell'aritmetica del calcolatore con precisione $\varepsilon_m \approx 1.1 \cdot 10^{-16}$, le quantità $y1 = (1 + x) - 1$ e $y2 = (1 - 1) + x$ per $x = 0.1$ **sono equivalenti**.
Infatti, si ha

```
>> format long e
>> x = 0.1;
>> y1 = (1+x)-1
y1 =
     1.000000000000001e-01
>> y2 = (1-1)+x
y2 =
     1.000000000000000e-01
>> er = abs(y1-y2)/abs(y2)
er =

     8.326672684688674e-16
```

I diversi risultati $y1$ e $y2$ ottenuti dimostrano che, in aritmetica con precisione finita di calcolo, la proprietà associativa non è valida; tuttavia, poiché la differenza relativa è dell'ordine della precisione di macchina, $y1$ e $y2$ sono equivalenti nell'aritmetica del calcolatore.

... continua esempio

Le quantità $y1 = (1 + x) - 1$ e $y2 = (1 - 1) + x$ per $x = 1.0e - 11$ invece **non sono equivalenti**:

```
>> format long e
>> x = 1.0e-11;
>> y1 = (1+x)-1
y1 =
     1.000000082740371e-11
>> y2 = (1-1)+x
y2 =
     9.999999999999999e-12
>> er = abs(y1-y2)/abs(y2)
er =

     8.274037105959341e-08
```

I risultati $y1$ e $y2$ non sono equivalenti nell'aritmetica del calcolatore, anzi sono molto diversi tra loro! In particolare, si è verificata una perdita di cifre della mantissa di $y1$. Tale fenomeno va sotto il nome di *cancellazione numerica*.

Cancellazione numerica

La cancellazione numerica rappresenta una delle conseguenze più gravi della rappresentazione con precisione finita dei numeri reali.

Per descrivere il fenomeno della cancellazione numerica conviene partire da un esempio numerico.

Esempio

Siano $N = 10$ e $t = 5$. Si ha allora $\varepsilon_m = 0.5 \cdot 10^{-4}$.
Consideriamo i numeri

$$a_1 = 0.15782 \text{ e } a_2 = 0.15735$$

Osserviamo che le prime tre cifre delle mantisse di a_1 e a_2 coincidono; inoltre, $a_1 = \bar{a}_1$ e $a_2 = \bar{a}_2$. Eseguiamo l'operazione di sottrazione in aritmetica esatta e nell'aritmetica fissata:

in aritmetica esatta		in aritmetica finita	
a_1	$= 0.15782$	$\bar{a}_1$	$= 0.15782$
a_2	$= 0.15735$	$\bar{a}_2$	$= 0.15735$
$a_1 - a_2$	$= 0.00047$	$\bar{a}_1 \ominus \bar{a}_2$	$= 0.00047$
	$= 0.47 \cdot 10^{-3}$		$= 0.47 \cdot 10^{-3}$

Le due operazioni hanno fornito lo stesso risultato.

... segue esempio

Consideriamo ora i numeri

$$a_1 = 0.157824831 \text{ e } a_2 = 0.157348212$$

Osserviamo che le prime tre cifre delle mantisse di a_1 e a_2 coincidono; inoltre, $\bar{a}_1 = 0.15782 \neq a_1$ e $\bar{a}_2 = 0.15735 \neq a_2$. In tal caso, risulta

in aritmetica esatta		in aritmetica finita	
a_1	$= 0.157824831$	$\bar{a}_1$	$= 0.15782$
a_2	$= 0.157348212$	$\bar{a}_2$	$= 0.15735$
$a_1 - a_2$	$= 0.000476619$	$\bar{a}_1 \ominus \bar{a}_2$	$= 0.00047$
	$= 0.476619 \cdot 10^{-3}$		$= 0.47000 \cdot 10^{-3}$

Pertanto, la mantissa di $\bar{a}_1 \ominus \bar{a}_2$ ha solo le prime 2 cifre decimali in comune con la mantissa di $a_1 - a_2$. Si è verificata una perdita di tre cifre!

Nel primo caso gli operandi $\bar{a}_1$ e $\bar{a}_2$ non sono affetti dall'errore di arrotondamento e il risultato dell'operazione di sottrazione non presenta alcuna perdita di precisione; ciò è in accordo con la proprietà che tutte le operazioni aritmetiche di macchina possono provocare un errore che non supera mai la precisione di macchina.

Nel secondo caso, invece, entrambi gli operandi $\bar{a}_1$ e $\bar{a}_2$ sono affetti dall'errore di arrotondamento e tali errori vengono amplificati dall'operazione di sottrazione.
Infatti, si ha

$$\frac{|a_1 - \bar{a}_1|}{|a_1|} \approx 0.3 \cdot 10^{-4} < \varepsilon_m \qquad \frac{|a_2 - \bar{a}_2|}{|a_2|} \approx 0.1 \cdot 10^{-4} < \varepsilon_m$$

mentre

$$\frac{|(a_1 - a_2) - (\bar{a}_1 \ominus \bar{a}_2)|}{|a_1 - a_2|} \approx 0.1 \cdot 10^{-1} > \varepsilon_m$$

In generale, la cancellazione numerica può essere così definita.

Definizione

Siano

$$\bar{a}_1 = (-1)^{s_1}\,\bar{p}_1 N^{q_1}, \quad \bar{a}_2 = (-1)^{s_2}\,\bar{p}_2 N^{q_2}$$

le rappresentazioni di macchina associate rispettivamente ai numeri reali

$$a_1 = (-1)^{s_1}\,p_1 N^{q_1}, \quad a_2 = (-1)^{s_2}\,p_2 N^{q_2}$$

La **cancellazione numerica** consiste in una perdita di cifre della mantissa e si verifica quando si esegue l'**operazione di sottrazione fra due rappresentazioni di macchina $\bar{a}_1$ e $\bar{a}_2$ dello stesso segno, circa uguali e almeno uno delle quali sia affetta dall'errore di arrotondamento**, ossia quando

- $s_1 = s_2$
- $q_1 = q_2$
- $\bar{p}_1 \approx \bar{p}_2$
- $\bar{p}_1 \neq p_1$ e/o $\bar{p}_2 \neq p_2$

Esempio

Implementiamo in Matlab il calcolo della quantità

$$y = \frac{(1+x)-1}{x}$$

per i valori di x uguali a 10^{-k}, $k = 1, ..., 15$. Tenendo conto che in precisione infinita di calcolo $y \equiv 1$ qualunque sia x, calcoliamo l'errore assoluto $|1-y|$ associato a y (che in questo caso coincide con l'errore relativo), per ogni valore di x considerato.

```
>> format short e
>> k = 1:15;
>> x = 10.^-k;
>> y = ((1+x)-1)./x;
>> err = abs(1-y);
>> [x' err']
```

... segue esempio

```
ans =
 1.0000e-01 8.8818e-16
 1.0000e-02 8.8818e-16
 1.0000e-03 1.1013e-13
 1.0000e-04 1.1013e-13
 1.0000e-05 6.5512e-12
 1.0000e-06 8.2267e-11
 1.0000e-07 5.8387e-10
 1.0000e-08 6.0775e-09
 1.0000e-09 8.2740e-08
 1.0000e-10 8.2740e-08
 1.0000e-11 8.2740e-08
 1.0000e-12 8.8901e-05
 1.0000e-13 7.9928e-04
 1.0000e-14 7.9928e-04
 1.0000e-15 1.1022e-01
```

Si osservi che, al diminuire dell'ordine di grandezza di x, la perdita di precisione aumenta e il fenomeno della cancellazione è sempre più eclatante!

Talvolta, manipolando opportunamente le espressioni matematiche che definiscono un problema, è possibile evitare il fenomeno della cancellazione numerica; quando ciò non è possibile si dice che la cancellazione è **insita nel problema**.

Esempio 1

Sia $y = \sqrt{x+\delta} - \sqrt{x}$ con $x, x+\delta > 0$.

La cancellazione numerica per $|\delta| << x$ si elimina razionalizzando:

$$y = (\sqrt{x+\delta} - \sqrt{x}) \frac{\sqrt{x+\delta} + \sqrt{x}}{\sqrt{x+\delta} + \sqrt{x}} = \frac{\delta}{\sqrt{x+\delta} + \sqrt{x}}$$

La cancellazione numerica per $\delta \approx -x$ non è, invece, eliminabile.

Esempio 2

Sia $y = \dfrac{e^x - 1}{x}$ con $x \neq 0$.

La cancellazione numerica per $x \approx 0$ si elimina sostituendo e^x con il corrispondente sviluppo di Taylor, centrato in 0 e di ordine n:

$$\begin{aligned} y &= \frac{1}{x}\left(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} + o(x^n) - 1\right) \\ &= 1 + \frac{x}{2!} + \frac{x^2}{3!} + \ldots + \frac{x^{n-1}}{n!} + o(x^{n-1}) \end{aligned}$$

Successivamente per il calcolo numerico di y si sommano tutti e soli i termini che danno contributo alla somma (si ricordi, a tal scopo, la relazione $\bar{a}_1 \oplus \bar{a}_2 = \bar{a}_1$ se $|\bar{a}_2| < |\bar{a}_1| eps$).

... segue esempio

In Matlab si ha

```
>> x = 10^-5;
>> y1 = (exp(x)-1)/x
y1 =
     1.000005000006965e+00
>> y2 = 1+x/factorial(2)+x^2/factorial(3)
y2 =

     1.000005000016667e+00
% y2 e' la somma dei termini fino alla potenza con esponente 2
>> y3 = (1+x/factorial(2)+x^2/factorial(3)+x^3/factorial(4))
y3 =
     1.000005000016667e+00
% y3 e' la somma dei termini fino alla potenza con esponente 3
% poiche' y3=y2, il termine aggiunto non ha dato contributo
>> err = abs(y2-y1)/abs(y2)
err =
     9.701746414095626e-12
```

Osserviamo che, poiché la differenza relativa calcolata è molto maggiore di $\varepsilon_m \approx 1.1 \cdot 10^{-16}$, le quantità $y1$ e $y2$ non sono equivalenti nell'aritmetica del calcolatore. Il valore di $y1$ non è affidabile in quanto è affetto dalla cancellazione numerica, mentre il valore di $y2$ si può ritenere affidabile e accurato, perché è definito da operazioni che non amplificano gli eventuali errori presenti negli operandi.

Esempio 3

Sia $y = \dfrac{1 - \cos(x)}{x^2}$ con $x \in (0, \pi)$, per esempio.

La cancellazione numerica per $x \approx 0$ si elimina sostituendo $\cos(x)$ con il corrispondente sviluppo di Taylor, centrato in 0 e di ordine n.
Alternativamente, si può riscrivere y utilizzando una formula trigonometrica di bisezione:

$$y = \frac{2\sin^2(x/2)}{x^2} = \frac{1}{2}\left(\frac{\sin(x/2)}{x/2}\right)^2$$

Quest'ultima espressione è equivalente a quella iniziale in aritmetica esatta e non è affetta dalla cancellazione numerica.

Condizionamento di un problema numerico

Gli esempi precedenti hanno messo in evidenza che nella risoluzione di un problema gli errori presenti nei dati possono venire trasmessi/amplificati nei risultati.

Nello studio della propagazione degli errori bisogna distinguere il ruolo del *problema* dal ruolo dell'*algoritmo* utilizzato per risolvere tale problema.

A tal scopo si introducono le definizioni di problema numerico e algoritmo, e i corrispondenti concetti di *condizionamento di un problema* e *stabilità di un algoritmo*.

Definizione

Si definisce **problema numerico** una relazione funzionale f

tra i dati x (**input**) e i risultati y (**output**).

I dati x e i risultati y devono essere rappresentabili da numeri, vettori o matrici di numeri di dimensione finita.

La connessione f tra x e y può essere **esplicita**

$$y = f(x)$$

oppure **implicita**

$$f(x, y) = 0$$

Esempi

1. La somma di due numeri reali $y = x_1 + x_2$, con $x_1, x_2 \in \mathbb{R}$ (input) e $y \in \mathbb{R}$ (output), è un problema esplicito.
2. Il sistema lineare

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases}$$

con $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2 \in \mathbb{R}$ (input) e soluzione $x, y \in \mathbb{R}$ (output), è un problema implicito.

Sia $y = f(x)$ oppure $f(x, y) = 0$ un generico problema numerico. Si denotino con

- $\bar{x}$ una perturbazione dei dati x di input,
- $\bar{y}$ i risultati ottenuti a partire dai dati $\bar{x}$ in **precisione infinita di calcolo**.

Definizione

Un problema numerico si dice **ben condizionato** se accade che l'errore relativo associato a $\bar{y}$ è dello stesso ordine di grandezza dell'errore relativo associato a $\bar{x}$ o minore; altrimenti, si dice **mal condizionato**.

Pertanto, un problema è ben condizionato quando le perturbazioni nei dati non influenzano eccessivamente i risultati.

Per studiare il condizionamento di un problema si possono determinare stime del tipo:

$$\frac{|y - \bar{y}|}{|y|} \leq K(f, x) \frac{|x - \bar{x}|}{|x|}$$

Se i dati di input e/o i risultati sono rappresentati da una funzione, oppure un vettore o una matrice, nella stima occorre sostituire il valore assoluto con una norma opportuna.

Definizione

Si definisce **numero di condizionamento** del problema il più piccolo valore di $K(f, x)$ per cui vale la suddetta disuguaglianza.

Se K non è eccessivamente grande, il problema è ben condizionato.

Esempio

Studiamo il condizionamento del problema numerico

$$y = x_1 + x_2, \;\; x_1, x_2 \in \mathbb{R}$$

A tal scopo denotiamo con $\bar{x}_1$ e $\bar{x}_2$ i dati perturbati (ad esempio le rappresentazioni di macchina di x_1 e x_2) e determiniamo una maggiorazione dell'errore relativo associato al risultato perturbato $\bar{y} = \bar{x}_1 + \bar{x}_2$. Riscrivendo $\bar{x}_1$ e $\bar{x}_2$ nella forma

$$\bar{x}_1 = x_1(1+\epsilon_1) \text{ e } \bar{x}_2 = x_2(1+\epsilon_2), \text{ con } \epsilon_i = (\bar{x}_i - x_i)/x_i, \;\; i = 1, 2,$$

si ha

$$\frac{y - \bar{y}}{y} = \frac{x_1 + x_2 - (\bar{x}_1 + \bar{x}_2)}{x_1 + x_2} = \frac{x_1 + x_2 - x_1(1+\varepsilon_1) - x_2(1+\varepsilon_2)}{x_1 + x_2}$$

$$= -\frac{x_1}{x_1 + x_2}\varepsilon_1 - \frac{x_2}{x_1 + x_2}\varepsilon_2$$

... segue esempio

Posto $K_i = |x_i/(x_1 + x_2)|$ $i = 1, 2$ e $K = \max\{K_1, K_2\}$, si deduce

$$\frac{|y - \bar{y}|}{|y|} \leq K_1\varepsilon_1 + K_2\varepsilon_2 \leq K\left(\frac{|x_1 - \bar{x}_1|}{|x_1|} + \frac{|x_2 - \bar{x}_2|}{|x_2|}\right)$$

La quantità K rappresenta dunque il numero di condizionamento del problema; se $x_1 + x_2 \to 0$, allora $K \to \infty$. Quindi il problema può essere mal condizionato quando $x_1 + x_2 \approx 0$, ossia quando i due numeri sono di segno opposto e circa uguali in modulo.

Ne consegue che la somma di due numeri di segno opposto e circa uguali in modulo può causare un'amplificazione degli errori presenti nei due operandi. Nel caso in cui $\bar{x}_1$ e $\bar{x}_2$ siano le rappresentazioni di macchina di due numeri reali x_1 e x_2, il mal condizionamento della somma algebrica fornisce un'interpretazione alternativa del fenomeno della cancellazione numerica.

Stabilità di un algoritmo

Definizione

Per **algoritmo** si intende una sequenza finita di operazioni (aritmetiche e non) che consente di ottenere l'output di un problema a partire dai dati di input.

Data un'aritmetica con precisione finita, si denotino con

- $\bar{x}$ l'arrotondamento dei dati x di input,
- $\bar{y}$ i risultati dell'algoritmo ottenuti a partire dai dati $\bar{x}$ in **precisione infinita di calcolo**,
- $\tilde{y}$ i risultati dell'algoritmo ottenuti a partire dai dati $\bar{x}$ in **precisione finita di calcolo**.

Per giudicare la bontà di un algoritmo per la risoluzione di un problema, bisogna dunque confrontare la risposta $\tilde{y}$ con $\bar{y}$.

Definizione

Un algoritmo si dice **numericamente stabile** se accade che l'errore relativo associato al risultato $\tilde{y}$ ha lo stesso ordine di grandezza della precisione di macchina o minore; altrimenti, si dice **instabile**.

Pertanto, un algoritmo è numericamente stabile quando la sequenza delle operazioni non amplifica eccessivamente gli errori di arrotondamento presenti nei dati.

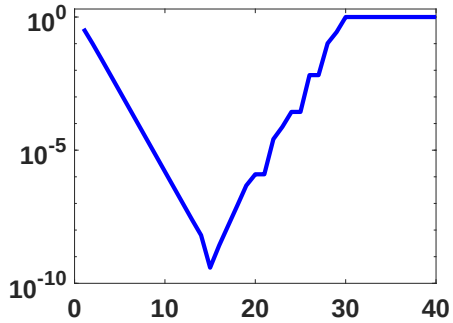
Esempio

L'algoritmo per il calcolo di π, precedentemente descritto,

$$x_1 = 2$$
$$x_n = 2^{n-1/2}\sqrt{1-\sqrt{1-4^{1-n}x_{n-1}^2}} \qquad n \geq 2$$

è instabile a causa del fenomeno della cancellazione numerica. Infatti, al crescere di n, la quantità $z = \sqrt{1-4^{1-n}x_{n-1}^2}$ si avvicina sempre più a 1 e genera una perdita di cifre nella successiva operazione $1 - z$.

Figura: Errore relativo $|\pi - x_n|/|\pi|$ al variare di $n = 1, \ldots, 40$



... segue esempio

Mediante la razionalizzazione è possibile eliminare la cancellazione numerica e l'algoritmo che così si ottiene

$$x_1 = 2$$
$$x_n = x_{n-1}\sqrt{\frac{2}{1+\sqrt{1-4^{1-n}x_{n-1}^2}}} \qquad n \geq 2$$

è stabile.

Figura: Errore relativo $|\pi - x_n|/|\pi|$ al variare di $n = 1, \ldots, 40$