Laboratorio di Calcolo Numerico

Aritmetica di macchina e analisi degli errori

Ángeles Martínez Calomardo
http://www.dmsa.unipd.it/~acalomar/DIDATTICA/2012-13
angeles.martinez@unipd.it

Laurea in Matematica
A.A. 2012–2013

Errori

La soluzione al calcolatore di un problema matematico è affetta da errori di vario tipo:

1. Errori dovuti alla modellazione matematica del problema reale ed errori presenti nei dati sperimentali.
2. Errori di troncamento commessi nella trasformazione di un problema matematico (dimensione infinita) in uno di dimensione finita.
3. Errori di arrotondamento dovuti al fatto che sul calcolatore si può rappresentare soltanto un sottoinsieme finito dei numeri reali.

I punti 2 e 3 sono oggetto di studio del Calcolo Numerico.

Effetti disastrosi degli errori numerici

Fallimento del missile Patriot: il giorno 25 Febbraio 1991, durante la prima guerra del golfo, un missile Patriot fallì l'intercettazione di un missile Scud iracheno che centrò il suo obbiettivo causando la morte di 28 soldati americani e un centinaio di feriti.

- Causa: L'orologio interno del sistema misurava il tempo in decimi di secondo, poi questo numero intero veniva moltiplicato per 0.1 per ottenere il tempo in secondi e memorizzato usando soli 24 bit.
- 0.1 non ha un'espansione binaria finita $\longrightarrow$ ad ogni decimo di secondo l'errore che si commette è (circa) 0.95×10^{-7} secondi.
- Il computer che regolava i lanci dei Patriot rimase in funzione per 100 ore il che produsse un errore pari a $0.95 \times 10^{-7} \times 100 \times 3600 \times 10 \approx 0.34$ secondi.
- Lo Scud viaggiava a Mach 5 (1700 m/sec) con un conseguente errore nella traiettoria di circa 600 metri.

Effetti disastrosi degli errori numerici

Esplosione del razzo Ariane 5: il 4 giugno 1996, avvenuta a soli 40 secondi dal lancio dovuta a un overflow per una conversione di un numero reale memorizzato usando 64 bit in un numero intero con soli 16 bit. Il numero che rappresentava la velocità orizzontale del razzo era più grande del massimo numero rappresentabile con soli 16 bit. La missione era costata 7.5 miliardi di dollari.

Altri esempi:

1. Crollo di una piattaforma petrolifera nel mare del Nord (Norvegia) nel 1991.
2. Distruzione del veicolo spaziale Mars Climate orbiter nel 1999.

Altre informazioni alla pagina del professore D. Arnold

`http://www.ima.umn.edu/~arnold/disasters/`

e su

`http://marsprogram.jpl.nasa.gov/msp98/orbiter`

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.

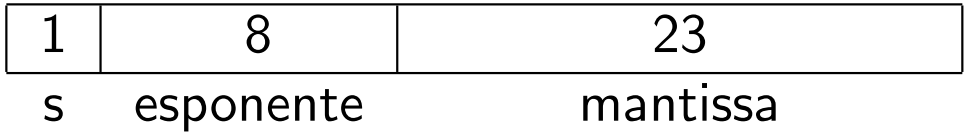
Lo standard ANSI IEEE-754r

Lo standard riserva due dei possibili valori per l'esponente per codificare due situazioni speciali:

- $e^* = 0$, viene riservato per la codifica dello zero ed eventuali numeri denormalizzati.
- $e^* = 255$ o $e^* = 2047$, che corrisponde a un esponente vero pari a 128 (singola) o 1024 (doppia), viene riservato per la codifica di
  - `Inf` (Overflow)
  - `NaN` (Not a Number), ovvero operazioni del tipo $\frac{0}{0}$, $\texttt{Inf} - \texttt{Inf}$, $\frac{\texttt{Inf}}{\texttt{Inf}}$.
- `Inf` viene codificato con mantissa nulla, mentre `NaN` con mantissa $\neq 0$.
- Più informazione sullo standard è reperibile al sito `http://it.wikipedia.org/wiki/IEEE_754`

Singola precisione

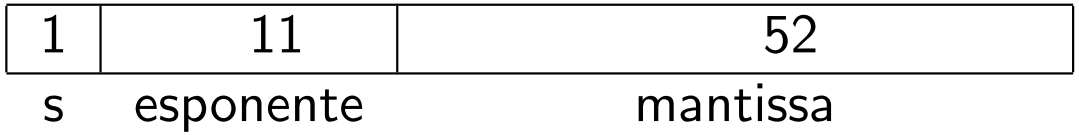
- Ogni numero macchina occupa 32 bit, distribuiti nei tre campi segno, esponente e mantissa come segue:



- Attualmente, l'insieme di numeri macchina in singola precisione è: $F(2,24,-126,127)$.
- Con 23 bit si codificano 24 cifre della mantissa (1 bit nascosto).
- Dei $2^8 = 256$ esponenti possibili, 2 si riservano per usi speciali.
- I numeri rappresentabili in singola precisione sono $2 \cdot (U - L + 1) \cdot (B - 1) \cdot B^{t-1} + 1 = 2 \cdot 254 \cdot 1 \cdot 2^{23} \approx 4.2789 \cdot 10^9$.

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 $2 \cdot (U - L + 1) \cdot (B - 1) \cdot B^{t-1} + 1 = 2 \cdot 2046 \cdot 1 \cdot 2^{52} \approx 1.8438 \cdot 10^{19}$.

Precisione di macchina

Definizione

Si chiama precisione di macchina, e si denota con il simbolo $\mathbf{u}$ *(unit roundoff):*

$$\mathbf{u} = \frac{1}{2} \cdot B^{1-t}$$

- La precisione di macchina rappresenta il massimo errore relativo che si commette nell'approssimare il numero reale x con il suo corrispondente numero macchina $\mathrm{fl}(x)$.
- In singola precisione $\mathbf{u} = 2^{-24} \approx 5.96 \cdot 10^{-8}$
- In doppia precisione $\mathbf{u} = 2^{-53} \approx 1.11 \cdot 10^{-16}$

Esempio di codifica

Vediamo come si rappresenta il numero $x = -8.265625$ secondo lo standard IEEE in doppia precisione.

- $(-8.265625)_{10} = (-1000.010001)_2$
- Il numero $(-1000.010001)_2$ si scrive, in virgola mobile normalizzata secondo lo standard IEEE come $-1.000010001 \cdot 2^3$.
- Della mantissa si immagazzinano solo le cifre dopo la virgola.
- $e^* = 3 + 1023 = 1026$, che in binario corrisponde a 10000000010.
- Il numero x viene codificato con la stringa

$$1 \quad 10000000010 \quad 00001000\underbrace{1000\cdots000}_{43 \text{ zeri}}$$

Esercizio

Usando OCTAVE *confrontare la stringa di bit ottenuta con quella visualizzata per* x *usando il formato* `format bit`.

Massimo e minimo numero rappresentabile

L'insieme F di numeri macchina è limitato sia inferiormente che superiormente.

- Il più piccolo numero rappresentabile ha tutte le cifre della mantissa uguali a 0 tranne la prima (virgola mobile normalizzata) ed il minimo esponente possibile.
- Nello standard IEEE 754 in doppia precisione tale numero è $1 \cdot 2^{-1022} = 2.2251 \cdot 10^{-308}$
- In MATLAB/OCTAVE questo numero è nella variabile `realmin`

```
>> realmin
ans =   2.2251e-308
>>
>> 2^(-1022)
ans =   2.2251e-308
```

Esercizio

Determinare, in maniera analoga a quanto appena effettuato per `realmin`*, il valore del massimo numero macchina in virgola mobile normalizzata, usando doppia precisione secondo lo standard IEEE 754. Confrontare il valore ottenuto con quello della variabile* MATLAB/OCTAVE `realmax`*.*

Massimo numero rappresentabile

Soluzione dell'esercizio

- Il numero più grande rappresentabile al calcolatore ha tutte le cifre della mantissa uguali a $B-1$ ed il massimo esponente possibile.
- Nello standard IEEE 754 in doppia precisione il massimo numero rappresentabile è $(2-2^{-52})\cdot 2^{1023} = 1.7977 \cdot 10^{308}$
- Ne abbiamo la conferma con MATLAB/OCTAVE

```
>> (2-2^(-52))*2^1023
ans =  1.7977e+308

>> realmax
ans =  1.7977e+308

>> format bit
>> realmax
ans = 0111111111101111111111111111111111111111111111111111111111111111
>>
```

Overflow e underflow

Poiché `realmax` è il massimo numero rappresentabile, assegnando ad una variabile un numero maggiore di `realmax` MATLAB/OCTAVE restituisce come valore `Inf`. Questo fenomeno si chiama **overflow**:

```
>>> a=1.e310
a  =  Inf
```

Differentemente, è ancora possibile assegnare ad una variabile un valore inferiore a `realmin`, perché il più piccolo numero rappresentabile in MATLAB/OCTAVE è:

```
>>> realmin/2^52
ans  =   4.9407e-324
```

anche se tutti i numeri macchina compresi tra `realmin` (escluso) e `realmin`$/2^{52}$ fanno parte dei cosidetti *numeri denormalizzati* e cioè del tipo

$$(-1)^s \cdot (0.F) \cdot 2^{-1023}$$

e sono quindi numeri con esponente codificato uguale a zero, come il numero zero, ma con parte frazionaria della mantissa non nulla.

Numeri denormalizzati e Underflow

Esercizio

Verificare che `realmin` *non è il più piccolo numero rapresentabile in* MATLAB/OCTAVE *assegnando ad una variabile* **a** *un valore minore di* `realmin` *e vedendo che si ha ancora un risultato diverso da zero. Che valore viene assegnato ad* **a** *quando proviamo a dare valori più piccoli di* $4.9407E-324$*? Come si chiama questo fenomeno?*

Soluzione:
Il valore rappresentato è zero e il fenomeno si chiama underflow.

```
>> a=1e-308
a =   1.0000e-308
>> a=10^-308
a =   1.0000e-308
>> a=1e-311
a =   1.0000e-311
>> a=1e-323
a =   9.8813e-324
>> a=1e-324
a = 0
```

Nan (Not a number)

Esercizio

Eseguire l'assegnazione $\mathbf{a} = \frac{\mathbf{1}}{\mathbf{0}}$. *Che risultato si ottiene? (Facoltativo: usando* OCTAVE *visualizzare cambiando il formato a* `format bit` *la codifica interna del risultato ottenuto.)*

Esercizio

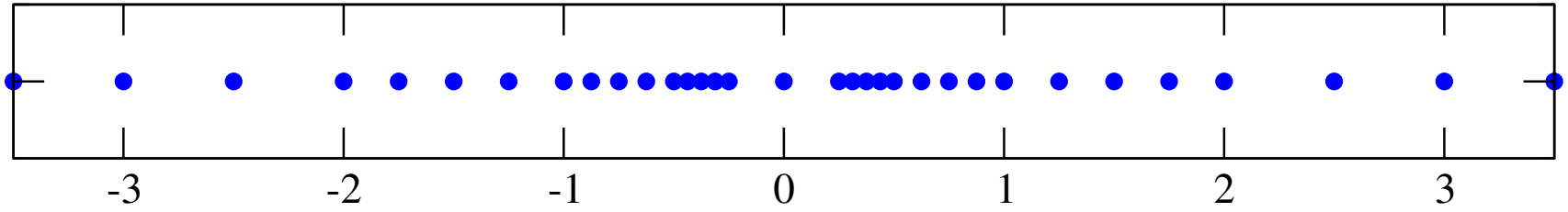
Eseguire l'operazione aritmetica $\mathbf{a} - \mathbf{a}$. *Che risultato si ottiene? (Facoltativo: usando* OCTAVE *visualizzare cambiando il formato a* `format bit` *la codifica interna del risultato ottenuto. Spiegare che criterio si usa per codificare internamente NaN (not a number). Qual è la differenza con Inf?)*

Distanza assoluta tra due numeri macchina consecutivi

- L'insieme $\mathbb{R}$ dei numeri reali è denso.
- Il sottoinsieme dei numeri macchina $\mathbb{F}$, oltre ad essere limitato inferiormente e superiormente, è anche *bucato*, ovvero attorno ad ogni elemento x di $\mathbb{F}$ 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:

```
>>> format long e
>>> 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
```

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:

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

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

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

- 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 $\varepsilon_M = 2^{-52}$.
- In MATLAB/OCTAVE $\varepsilon_M = 2^{-52}$ 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

$$\begin{aligned} \varepsilon_{x,y}^{\oplus} &\leq \left|\frac{x}{x+y}\right|\varepsilon_x + \left|\frac{y}{x+y}\right|\varepsilon_y \\ \varepsilon_{x,y}^{\otimes} &\lessapprox \varepsilon_x + \varepsilon_y \\ \varepsilon_{x,y}^{\oslash} &\leq |\varepsilon_x - \varepsilon_y| \end{aligned}$$

  dove ε_x e ε_y sono tali che $\varepsilon_x, \varepsilon_y < \mathbf{u}$

- 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 algoritmo instabile

Formula risolutiva dell'equazione di secondo grado

- Dato il polinomio di secondo grado $x^2 + 2px - q$, con $\sqrt{p^2+q} \geq 0$ le radici

$$y = -p \pm \sqrt{p^2+q}$$

dell'equazione sono reali.

- La soluzione

$$y = -p + \sqrt{p^2+q} \tag{1}$$

è la maggiore delle 2, ed è non negativa se e solo se $q \geq 0$.

- Questa espressione è potenzialmente instabile per $p \gg q$ a causa della sottrazione tra p e $\sqrt{p^2+q}$.
- A tal proposito, dopo averla implementata in Matlab, verificheremo numericamente la perdita di accuratezza per opportune scelte dei coefficienti p e q.
- Ripetiamo poi lo stesso tipo di indagine con una formula alternativa (e stabile) che si ottiene razionalizzando (1). In altri termini

$$y = -p + \sqrt{p^2+q} = \frac{(-p+\sqrt{p^2+q})(p+\sqrt{p^2+q})}{(p+\sqrt{p^2+q})} = \frac{q}{(p+\sqrt{p^2+q})} \tag{2}$$

Formula risolutiva dell'equazione di secondo grado

Implementazione in Matlab

Scriviamo un programma `radicisecgrado.m` in Matlab che illustri i due algoritmi.

```
% p=4.999999999995*10^(+4); q=10^(-2); sol=10^(-7);
p=1000; q=0.018000000081; sol=0.9*10^(-5);

% ALGORITMO 1: INSTABILE
s=p^2;
t=s+q;
if t >=0
   u=sqrt(t);
else
   fprintf('\n \t [RADICI COMPLESSE]');
end
s1=-p+u;

% ALGORITMO 2: STABILE
s=p^2;
t=s+q;
if t >=0
   u=sqrt(t);
else
   fprintf('\n \t [RADICI COMPLESSE]');
end
v=p+u;
t1=q/v;

% STAMPA DEI RISULTATI
fprintf('\n \t [ALG.1] [1]: %10.19f',s1);
fprintf('\n \t [ALG.2] [1]: %10.19f',t1);
if length(sol) > 0 & (sol ~= 0)
    relerrs1=abs(s1-sol)/abs(sol);
    relerrt1=abs(t1-sol)/abs(sol);
    fprintf('\n \t [REL.ERR.][ALG.1]: %2.2e',relerrs1);
    fprintf('\n \t [REL.ERR.][ALG.2]: %2.2e',relerrt1);
end
```

Formula risolutiva dell'equazione di secondo grado

Tests

Digitiamo quindi da shell Matlab/Octave il comando `radicesecgrado` e otteniamo

```
>> radicesecgrado

   [ALG.1]  [1]:  0.00000899999999772772
   [ALG.2]  [1]:  0.00000900000000000000
   [REL.ERR.][ALG.1]:  2.52e-009
   [REL.ERR.][ALG.2]:  0.00e+000
```

Come previsto, il secondo algoritmo si comporta notevolmente meglio del primo, che compie un errore relativo dell'ordine di circa 10^{-9}.

Esercizio

Testare il codice `radicesecgrado.m` *per l'esempio in cui*

```
p=4.999999999995*10^(+4);  q=10^(-2);  sol=10^(-7);
```

Esercizio proposto

Esercizio

Si determini una formula stabile alternativa per il calcolo della radice positiva dell'equazione di secondo grado $x^2 + 2px - q$ *con* p *e* q *i valori dell'esercizio precedente.*
Suggerimento: si consideri la relazione che esiste tra il prodotto delle radici e il coefficiente q*.*