

# Tracce di calcolo numerico<sup>1</sup>

Prof. Marco Vianello - Dipartimento di Matematica, Università di Padova  
aggiornamento: 24 settembre 2016

## 1 Sistema floating-point e propagazione degli errori

### 1.1 Rappresentazione dei numeri reali, errore di troncamento e di arrotondamento

1. si ricordi che, fissata una base (un numero naturale  $\beta > 1$ ), ogni numero  $x \in \mathbb{R}$  si può scrivere (rappresentazione a virgola fissa) come

$$x = \text{sign}(x) (c_m \dots c_1 c_0 . c_{-1} \dots c_{-n} \dots)_\beta$$
$$= \text{sign}(x) \left( \sum_{j=0}^m c_j \beta^j + \sum_{j=1}^{\infty} c_{-j} \beta^{-j} \right)$$

dove  $c_j, c_{-j} \in \{0, 1, \dots, \beta - 1\}$  sono le cifre della rappresentazione in base  $\beta$  (ad esempio  $\{0, 1\}$  in base 2,  $\{0, \dots, 9\}$  in base 10); chiamiamo  $\sum_{j=0}^m c_j \beta^j$  parte intera del numero e  $\sum_{j=1}^{\infty} c_{-j} \beta^{-j}$  parte frazionaria del numero

2. si ricordino le principali proprietà della somma geometrica e della serie geometrica di ragione  $a \neq 1$ , ovvero  $S_n = \sum_{j=0}^n a^j$  e  $S = \sum_{j=0}^{\infty} a^j$   
traccia:  $aS_n - S_n = (a - 1)S_n = a^{n+1} - 1$ , ...
3. perché la serie che rappresenta la parte frazionaria converge?  
traccia: si utilizzi il confronto con la serie geometrica di ragione  $a = 1/\beta$ , osservando che  $c_{-j} \leq \beta - 1$  (si tratta del criterio di confronto tra serie a termini non negativi); si osservi che la parte frazionaria sta in  $[0, 1]$ , controllando ad esempio che  $(0.999 \dots)_{10} = (0.111 \dots)_2 = 1$
4. la parte frazionaria di un numero irrazionale è infinita (perché?); la parte frazionaria di un numero razionale può essere finita o infinita a seconda della base:  $1/3 = (0.333 \dots)_{10}$  (verificarlo) ma  $1/3 = (0.1)_3$
5. si dimostri, usando le serie, che l'errore di troncamento ad  $n$  cifre della parte frazionaria in base  $\beta$  è  $\leq \beta^{-n}$   
traccia: l'errore di troncamento non è altro che il resto della serie corrispondente alla parte frazionaria, ...
6. si dia un'interpretazione geometrica (con un disegno, ad esempio nel caso di 2 cifre decimali) del fatto che il massimo errore di arrotondamento ad  $n$  cifre è la metà del massimo errore di troncamento (con l'usuale regola di arrotondamento, base pari: si tiene la cifra com'è se la prima trascurata è minore di  $\beta/2$ , si aumenta la cifra di una unità se la prima trascurata è maggiore o uguale di  $\beta/2$ )

---

<sup>1</sup>argomenti e quesiti contrassegnati da \* sono più impegnativi, se non si è in grado di fare la dimostrazione bisogna comunque sapere (e saper usare) gli enunciati e capire di cosa si sta parlando

## 1.2 Sistema floating-point

1. si mostri (con qualche esempio) che ogni numero reale si può anche scrivere (rappresentazione normalizzata a virgola mobile in base  $\beta$ ) come

$$x = \text{sign}(x)\beta^p(0.d_1 \dots d_t \dots)_\beta = \text{sign}(x)\beta^p \sum_{j=1}^{\infty} d_j \beta^{-j}$$

$d_j \in \{0, 1, \dots, \beta-1\}$ ,  $d_1 \neq 0$ , dove chiamiamo  $\sum_{j=1}^{\infty} d_j \beta^{-j}$  *mantissa* e  $p \in \mathbb{Z}$  *esponente* della rappresentazione; a cosa serve la normalizzazione  $d_1 \neq 0$ ?

2. la mantissa di un numero reale sta in  $[0, 1]$ , ma non è la sua parte frazionaria
3. i numeri irrazionali hanno parte frazionaria (e mantissa) infinita
4. si studi l'insieme dei numeri floating-point

$$\mathbb{F}(\beta, t, L, U) = \{\mu = \pm(0.\mu_1\mu_2 \dots \mu_t)\beta^p, \mu_j \in \{0, 1, \dots, \beta-1\}, \mu_1 \neq 0, p \in [L, U] \subset \mathbb{Z}\}$$

traccia:

- $\text{card}(\mathbb{F}) = 1 + 2(\beta-1)\beta^{t-1}(U-L+1)$  (sugg.:  $\mathbb{F}$  è simmetrico,  $\mathbb{F}^- = -\mathbb{F}^+$ ; si contino le possibili mantisse e i possibili esponenti)
  - $\min \mathbb{F}^+ = \beta^{L-1}$  (sugg.: chi è la minima mantissa?)
  - $\max \mathbb{F}^+ = \beta^U(1 - \beta^{-t})$  (sugg.: utilizzare la somma geometrica per calcolare la massima mantissa)
  - i numeri floating-point sono razionali
  - si rifletta sul fatto che la densità è variabile calcolando la distanza tra numeri macchina consecutivi; dove e come cambia tale densità?
  - si rifletta sul concetto di raggio assoluto e relativo dell'intorno di approssimazione corrispondente ad ogni numero macchina, calcolando la precisione di macchina  $\varepsilon_M$  che è il massimo errore relativo di arrotondamento a  $t$  cifre di mantissa
  - quando l'intorno associato ad un numero floating-point non è simmetrico?
  - quali sono i numeri reali rappresentabili (ovvero approssimabili per arrotondamento a  $t$  cifre di mantissa) tramite questi numeri floating-point?
5. si disegnino  $\mathbb{F}(10, 1, -1, 1)$  e  $\mathbb{F}(10, 2, -2, 2)$ ; perché i numeri floating-point contigui ad 1 sono (in notazione posizionale classica) nel primo caso 0.9 e 2 (non 1.1) e nel secondo caso 0.99 e 1.1 (non 1.01)?
  6. i numeri floating-point “grandi” (in modulo) sono interi e hanno moltissime cifre nulle; in un sistema floating-point con  $t$  cifre di mantissa in base  $\beta$ , i numeri interi con più di  $t$  cifre vengono arrotondati (se rappresentabili)
  7. la precisione di macchina,  $\varepsilon_M = \beta^{1-t}/2$ , non è il più piccolo numero floating-point positivo (che invece è ...)

8. si discuta un modello di codifica dei reali in base 2 con una sequenza di 64 bit, la cosiddetta “precisione doppia”  
(traccia: riservando un bit per il segno, 52 (+1) bit per la mantissa che è come avere 53 bit perché la prima cifra di mantissa deve essere 1, e 11 bit per l’esponente di cui 1 per il segno e 10 per il valore assoluto, si calcolino la precisione di macchina e gli estremi  $L$  ed  $U$  dell’intervallo degli esponenti; quali sono gli ordini di grandezza di  $\varepsilon_M$ ,  $\max \mathbb{F}^+$  e  $\min \mathbb{F}^+$  in base 10?)

### 1.3 Propagazione degli errori

1. in un’aritmetica a base 10 con 16 cifre di mantissa,  $1 + 10^{-15}$  viene calcolato correttamente ma  $1 + 10^{-16} = 1$ ; perché? (si osservi che questo è un esempio di non unicità dell’elemento neutro)
2.  $*$  si può dimostrare che vale anche  $\varepsilon_M = \min \{\mu \in \mathbb{F}_+ : 1 + \mu > 1\}$
3. detti  $\varepsilon_x = |x - \tilde{x}|/|x|$  e  $\varepsilon_y = |y - \tilde{y}|/|y|$ ,  $x, y \neq 0$ , gli errori relativi sui dati si studi la stabilità delle operazioni aritmetiche con numeri approssimati, mostrando che per ciascuna operazione aritmetica  $\star$  si ha una stima del tipo “somma pesata degli errori”

$$\varepsilon_{x \star y} = \frac{|(x \star y) - (\tilde{x} \star \tilde{y})|}{|x \star y|} \leq w_1(x, y)\varepsilon_x + w_2(x, y)\varepsilon_y, \quad x \star y \neq 0,$$

calcolando  $w_1, w_2$  nei vari casi (moltiplicazione, divisione, addizione, sottrazione; traccia: si utilizzi la disuguaglianza triangolare); quali operazioni si possono considerare “stabili”? in quali situazioni la sottrazione fa perdere precisione? si facciano esempi

4. detto  $\varepsilon_f$  l’errore relativo su una funzione (derivabile) con variabile approssimata,  $\varepsilon_x = |x - \tilde{x}|/|x|$ ,  $x \neq 0$ , utilizzando la formula di Taylor si ricavi la “formula degli errori”

$$\varepsilon_{f(x)} \approx \text{cond}(f(x))\varepsilon_x, \quad \text{cond}(f(x)) = |xf'(x)/f(x)|, \quad f(x) \neq 0$$

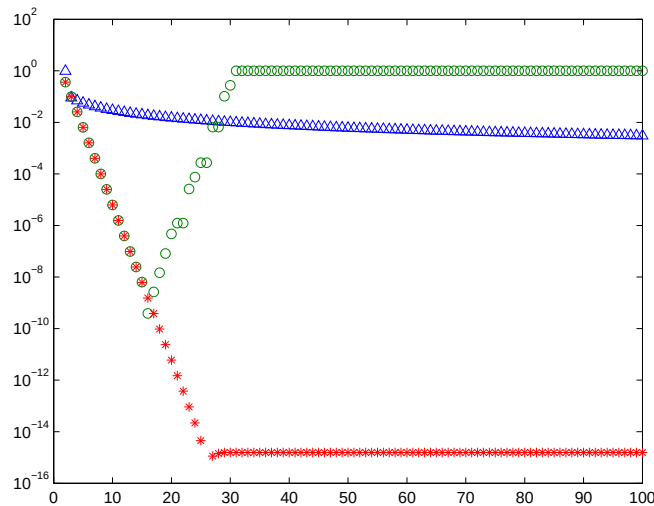
5. si consideri  $f(x) = ((1+x)-1)/x$ ,  $x \neq 0$ ; in Matlab si ha che  $f(10^{-15}) = 1.110223024625157$ , invece  $((1 + 2^{-50}) - 1)/2^{-50} = 1$ , dove  $2^{-50} \approx 10^{-15}$ ; perché?
6. si consideri  $f(x) = 1 - \sqrt{1 - x^2}$ ,  $|x| \leq 1$ , e si calcoli  $\text{cond}(f(x))$ ; in Matlab si ha che  $f(10^{-4}) = 5.000000080634948\text{e-}09$  ma il valore esatto (alla precisione di macchina) è  $5.000000012500000\text{e-}09$ ; la perdita di precisione (quantificarla) è compatibile con la formula degli errori? in caso contrario, qual’è il problema e come si può superarlo?
7. si facciano esempi in cui la proprietà associativa non è valida in aritmetica di macchina per overflow oppure per effetto dell’arrotondamento
8. la formula risolutiva classica per le equazioni di secondo grado  $ax^2 + bx + c = 0$ ,  $a \neq 0$ ,  $\Delta = b^2 - 4ac > 0$ , perde precisione in aritmetica di macchina se  $b^2 \gg 4|ac|$ ; si quantifichi la perdita di precisione calcolando i pesi della sottrazione corrispondente e si ricavi una formula “stabilizzata” (traccia: per la stabilizzazione, si osservi ad esempio che per  $b > 0$ ,  $\sqrt{\Delta} - b = (\sqrt{\Delta} - b)(\sqrt{\Delta} + b)/(\sqrt{\Delta} + b) = \dots$ )
9. si riscrivano, se necessario, le seguenti espressioni per ovviare a possibili instabilità in aritmetica di macchina:

$$E_1(x) = x^5/(2 + x^4 + 0.1|x|) - x + 100(x^4 + 5)^{1/4}$$

$$E_2(x) = \sqrt{x^2 + 1} - |x| + 100x$$

$$E_3(x) = 3\sqrt{x^4 + 2} - (10x^7 + 1)/(x^5 + 1) + 10x^2$$

10. la successione definita da  $x_2 = 2$ ,  $x_{n+1} = 2^{n-1/2} \sqrt{1 - \sqrt{1 - 4^{1-n} x_n^2}}$ ,  $n = 2, 3, \dots$  (“successione di Archimede”), soddisfa  $\lim_{n \rightarrow \infty} x_n = \pi$ , ma se usata per approssimare  $\pi$  diventa instabile in aritmetica di macchina (qual’è la sottrazione che fa perdere precisione? si calcolino i fattori di amplificazione dell’errore in funzione di  $n$ ); come può essere stabilizzata?
11. in figura gli errori relativi (in scala logaritmica) nell’approssimazione di  $\pi$  con la successione  $S_n = \sqrt{6 \sum_{j=1}^n \frac{1}{j^2}}$ ,  $n = 1, 2, 3, \dots$  (triangolini; stabile ma a convergenza lentissima, si stimi l’errore assoluto tramite il resto della serie  $\sum_{j=1}^{\infty} \frac{1}{j^2} = \pi^2/6$ ), con la successione di Archimede del punto (10) (pallini, convergente ma instabile) e con la corrispondente versione stabilizzata (asterischi); perchè con entrambe le versioni della successione di Archimede l’errore diventa ad un certo punto praticamente costante?



#### 1.4 Costo computazionale degli algoritmi numerici

1. si discuta il costo computazionale dell’algoritmo di Hörner per il calcolo dei valori di un polinomio; esempio per un polinomio cubico:  $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 = ((a_3x + a_2)x + a_1)x + a_0$
2. esiste un algoritmo veloce per il calcolo di una potenza tramite codifica binaria dell’esponente, basato sulla proprietà

$$a^n = a^{\sum_{j=0}^m c_j 2^j} = \prod_{j=0}^m a^{c_j 2^j},$$

dove  $\{c_j\}$  sono le cifre della codifica binaria di  $n \in \mathbb{N}$  e  $m$  è la parte intera di  $\log_2(n)$ ; qual’è il costo computazionale (come numero di operazioni aritmetiche floating-point)

in funzione di  $n$ ? qual'è il costo nel caso l'algoritmo venga applicato al calcolo di una potenza di matrice?

3. \* perché il calcolo di  $\exp(x)$  (per  $x > 1$ ) usando la formula  $\exp(x) = (\exp(x/m))^m$ ,  $m > [x]$  e la formula di Taylor per  $\exp(x/m)$ , è più efficiente dell'utilizzo diretto della formula di Taylor?  
(traccia: si stimi il modulo del resto  $n$ -esimo della formula di Taylor di  $\exp(x)$  per  $|x| \leq 1$  e per  $|x| > 1$ ; qual'è il comportamento per  $n \rightarrow \infty$ ?)
4. si dimostri che il costo computazionale della regola di Laplace per il determinante è  $\geq n!$  flops (si conti il numero di moltiplicazioni)
5. il metodo di eliminazione gaussiana può essere usato per calcolare il determinante di una matrice quadrata (come?), con costo computazionale  $\mathcal{O}(n^3)$  flops (da confrontare con il costo  $\mathcal{O}(n!)$  flops della regola di Laplace, che la rende inutilizzabile già per valori relativamente piccoli di  $n$ )
6. \* si dimostri che il costo computazionale asintotico del metodo di eliminazione gaussiana è  $2n^3/3$  flops (traccia: si incastri il costo tra due integrali)