Calcolo Numerico
Soluzioni del Foglio di Esercizi #1

Gianna Del Corso <gianna.delcorso@unipi.it>

Febbraio 2022

Esercizio 1. Si consideri l'insieme dei numeri di macchina $F(10, 3, 7, 6)$ in cui si assume di operare con troncamento.

(a) Assegnato $x = 2022$, se ne calcoli la rappresentazione $\tilde{x} \in F$.

(b) Si determinino tutti i numeri reali tali che la loro rappresentazione in F coincida con $\tilde{x}$,

c) Si calcoli la precisione di macchina u e si determinino due numeri $z \in F$ tali che $\frac{|z-x|}{|x|} < u$.

Soluzione 1. (a) Poichè si opera con troncamento abbiamo $\tilde{x} = 2020$.

(b) Si rappresentano con $\tilde{x}$ tutti i numeri reali nell'intervallo $I = [2020, 2030)$.

(c) $u = 10^{1-t} = 10^{-2}$. Se $z = \tilde{x}$ vale la maggiorazione per il teorema sull'errore di rappresentazione. Un altro z per cui vale questo risultato è ad esempio il valore che troviamo arrotondando il risultato cioè $z = 2030$. In questo caso infatti dalla teoria sappiamo che l'errore è minore di $\cdot 10^{1-t} < u$.

Esercizio 2. Si considerino i due insiemi $F(10, t, m, M)$ con $t = 3, m = 4, M = 5$ e G definito come l'unione di F con l'nsieme dei numeri non nulli della forma $x = \pm 10^{-m}(0.0d_2, \ldots, d_t)$, con $0 \leq d_i \leq 9$ per i=2, ..., t (quindi G contiene anche numeri piccoli non normalizzati).

(a) Si calcolino i minimi positivi non nulli ω e i massimi Ω degli insiemi F e G.

(b) Si determinino le cardinalità degli insiemi F e G.

c) Quale errore relativo di rappresentazione si commette volendo rappresentare $1.4\,10^{-(t+m)}$ in F e in G.

Soluzione 2. (a) $\Omega_F = 10^5(1 - 10^{-3}), \omega_F = 10^{-5}$, $\Omega_G = \Omega_F = 10^5(1 - 10^{-3}), \omega_G = 10^{-m-t} = 10^{-7}$.

(b) $|F| = 1 + 2(m + M + 1)(10^t - 10^{t-1}) = 18001$, $|G| = |F| + 2\,(10^{t-1} - 1) = 18199$.

(c) Poiche' $x = 1.4\,10^{-7} < \omega_F$, abbiamo una situazione di underflow e $\tilde{x} = 0$. Quindi l'errore relativo risulta 1. Se il numero e' rappresentato in G invece abbiamo che $x = 0.0014\,10^{-}4$ nonè un numero di macchina ma vale $\tilde{x} = 0.00110^{-4}$, quindi l'errore relativo è dato da

$$\left|\frac{\tilde{x} - x}{x}\right| = \frac{4\,10^{-8}}{1.4\,10^{-7}} = \frac{2}{7}.$$

Si nota che l'errore è molto più grande della precisone di macchina $u = \frac{1}{2}10^{-2}$, infatti i numeri denormalizzati non sono rappresentati altrettanto bene.

Esercizio 3. Sia $fl(x)$ la rappresentazione, ottenuta per troncamento, nell'insieme $F(10, 3, m, M)$ di un numero $x \in \mathbb{R}$.

(a) Dimostrare che $x_1, x_2 \in \mathbb{R}$ e $0 < x_1 \leq x_2$ implica $fl(x_1) \leq fl(x_2)$.

(b) Dimostrare con un opportuno esempio che $fl(x_1) \leq fl(x_2)$ non implica che $x_1 \leq x_2$.

Soluzione 3. (a) Sia $x_1 = 10^p \sum_{i=1}^{\infty} d_i 10^{-i}$ e $x_2 = 10^q \sum_{i=1}^{\infty} c_i 10^{-i}$. Poichè $x_1 \leq x_2$ possiamo avere due situazioni: $p < q$, ed in questo caso $fl(x_1) = 10^p \sum_{i=1}^{3} d_i 10^{-i} < 10^q \sum_{i=1}^{3} c_i 10^{-i} = fl(x_2)$.

Abbiamo poi il caso che $p = q$. In questo caso, significa che $\sum_{i=1}^{\infty} d_i 10^{-i} \leq \sum_{i=1}^{\infty} c_i 10^{-i}$. Se le prime tre cifre di x_1 e x_2 coincidono allora abbiamo che $fl(x_1) = fl(x_2)$ e quindi la disuguaglianza è verificata. Altrimenti se le prime tre cifre non coincidono deve esistere un $j \in \{1, 2, 3\}$ tale che $d_j \neq c_j$. Sia $j = min\{1, 2, 3\}$ tale che $d_j \neq c_j$. Deve essere $d_j < c_j$ da cui $fl(x_1) < fl(x_2)$.

La dimostrazione può anche essere fatta per assurdo. Supponiamo per assurdo che che $x_1 \leq x_2$ e che $fl(x_1) > fl(x_2)$. Poichè stiamo operando con troncamento e con dati positivi abbiamo che $fl(x_1) \leq x_1$ e $fl(x_2) \leq x_2$. Per le ipotesi fatte dobbiamo avere $fl(x_2) < fl(x_1) \leq x_1 \leq x_2$, ma questo non è possibile perchè quando operiamo con troncamento dobbiamo associare ad un numero il numero di macchina più vicino. Nello scenario precedente dovremmo avere $fl(x_2) = fl(x_1)$, che non è possibile. Abbiamo quindi un assurdo.

(b) Come controesempio possiamo prendere $x_1 = 1115$ e $x_2 = 1110$. Allora $fl(x_1) = 1110 = fl(x_2)$ ma $x_1 > x_2$.

Esercizio 4. Sia $\mathcal{F} = \mathcal{F}(\beta, t, m, M)$ con $m = M$.

a) Si dica quali sono il minimo numero positivo ω ed il massimo numero positivo Ω di $\mathcal{F}$.

b) Si dica, giustificando la risposta, se i numeri $b = 1/\omega$ e $B = 1/\Omega$ appartengono a $\mathcal{F}$.

c) Si esamini in particolare il caso $\beta = 2, t = 8$ ed $m = M = 6$.

Soluzione 4. (a) Come noto $\omega = \beta^{-m-1}$ e $\Omega = \beta^M(1-\beta^- t)$.

(b) Risulta $b = 1/\omega = \beta^{m+1}$. Poichè per ipotesi $m = M$ risulta $b > \Omega$ e quindi $b \notin \mathcal{F}$. Inoltre $B = 1/\Omega = \beta^{-M} \sum_{k=0}^{\infty} \beta^{-tk}$. TrattaTrattandosi di un numero periodico $B \notin \mathcal{F}$.

(c) Risulta $\omega = 2^{?7}, \Omega = 2^6(1-2^-8)$, $b = 2^7$ e $B = 2^6 \sum_{k=0}^{\infty} 2^{-8k}$.

Esercizio 5. Si dica quale è il più grande intero N tale che tutti gli interi nell'intervallo $[-N, N]$ sono rappresentabili esattamente in $\mathcal{F}(2, t, m, M)$. Quale è il valore di N corrispondente per l'aritmetica IEEE in doppia precisione?

Soluzione 5. Affinchè $x = 2^p f$ con f mantissa rappresentata su t bit sia un intero occorre che $p \geq t$. Poiche' vogliamo che siano rappresentabili tutti gli interi in $[-N, N]$ abbiamo due casi: $M \geq t$ e nel caso $N = 2^t(0.11\cdots 1)_2 = 2^t(1-2^{-t}) = 2^t - 1$. Nel caso in cui $M < t$ abbiamo che $N = 2^M - 1$.

Esercizio 6. Sia $f(x_1, \ldots, x_n) = \sum_{i=1}^n x_i^2$.

(a) Si studi il condizionamento del calcolo di $f(x)$.

(b) Si fornisca un algoritmo per il calcolo di $f(x)$ e se ne studi la stabilità.

(c) È noto che

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

Per $n = 10^8$ ed $n = 10^9$ si consideri la somma parziale $S_n = \sum_{k=1}^n \frac{1}{k^2}$ e si confrontino i valori ottenuti utilizzando con Matlab implementando due differenti algoritmi di somma (dal termine più piccolo a quello più grande e viceversa) e si confrontino gli errori relativi ottenuti con i due programmi.

Soluzione 6. (a) Per l'errore inerente vale

$$\varepsilon_{in} \doteq \frac{1}{f(x)} \sum_{i=1}^n x_i \frac{\partial f}{\partial x_i} \varepsilon_{x_i}.$$

Nel nostro caso $\frac{\partial f}{\partial x_i} = 2x_i$, da cui abbiamo $\varepsilon_{in} \doteq \frac{1}{f(x)} \sum_{i=1}^n 2\, x_i^2 \varepsilon_{x_i}$. Passando ai moduli, ed utilizzando la disuguaglianza triangolare abbiamo

$$|\varepsilon_{in}| \doteq |\frac{1}{f(x)} \sum_{i=1}^n 2\, x_i^2 \varepsilon_{x_i}| \leq \frac{1}{f(x)} \sum_{i=1}^n |2x_i^2||\varepsilon_{x_i}|.$$

Poichè $|\varepsilon_{x_i}| < u$ segue che $|\varepsilon_{in}| \leq 2u$. Quindi il problema risulta sempre ben condizionato.

(b) Per calcolare $f(x)$ posso ad esempio sommare i vari termini della serie in avanti, secondo il seguente algoritmo

$$\begin{aligned} p_1 &= x_1\, x_1 \\ p_2 &= x_2\, x_2 \\ &\vdots \quad \vdots \quad \vdots \\ p_n &= x_n\, x_n \\ s_2 &= p_1 + p_2 \\ &\vdots \quad \vdots \quad \vdots \\ s_i &= s_{i-1} + p_i \\ &\vdots \quad \vdots \quad \vdots \\ s_n &= s_{n-1} + p_n \end{aligned}$$

Il calcolo di ogni p_i genera un errore ε_i limitabile in modulo dalla precisione di macchina, mentre l'errore η_i nel calcolo di s_i è del tipo $\delta_i + \frac{p_i}{s_i}\varepsilon_i + \frac{s_{i-1}}{s_i}\eta_{i-1}$. Otteniamo quindi che

$$\varepsilon_{alg} \doteq \delta_n + \frac{p_n}{s_n}\varepsilon_n + \frac{s_{n-1}}{s_n}(\delta_{n-1} + \frac{p_{n-1}}{s_{n-1}}\varepsilon_{n-1} + \frac{s_{n-2}}{s_{n-1}}(\delta_{n-2} + \cdots)\cdots)$$

Otteniamo

$$\varepsilon_{alg} \doteq \sum_{i=1}^{n} \frac{p_i}{s_n}\varepsilon_i + \sum_{i=2}^{n} \frac{s_i}{s_n}\delta_i.$$

Passando ai moduli, applicando la disuguaglianza triangolare e maggiorando $|\delta_i|$ e $|\varepsilon_i|$ con u otteniamo che

$$|\varepsilon_{alg}| < u + (n-1)u = n\,u.$$

(c) Lo script per la somma in avanti risulta

```
% script somma forward
s=0;
for k=1:n
s=s+1/k^2;
end
```

Lo script per la somma all'indietro risulta

```
% script somma backward
s1=0;
for k=n:-1:1
s1=s1+1/k^2;
end
```

Per $n = 10^8$ lo script 1 restituisce

```
s =

   1.644934057834575
 l=pi^2/6;

>> err=abs(s-l)/l

err =

    5.479642961076798e-09
```

Con il secondo script

```
s1 =

   1.644934056848226
>> err1=abs(s1-l)/abs(l)

err1 =

    6.079270981593674e-09
```

Ripetendo l'esecuzione per $n = 10^9$ otteniamo

```
err=abs(s-l)/abs(l)

err =

    5.479642961076798e-09

>> err1=abs(s1-l)/abs(l)

err1 =

    6.079271521541406e-10
```

cioè l'algoritmo più stabile sembra quello nel quale si sommano gli elementi dal più piccolo al più grande.

Esercizio 7. Quale delle due espressioni equivalenti (dal punto di vista matematico)

$$\frac{1-x^4}{1-x} = 1 + x + x^2 + x^3$$

è più stabile? Si studi anche il condizionamento del calcolo dell'espressione precedente.

Soluzione 7. Per quanto riguarda il condizionamento abbiamo:

$$\varepsilon_{in} = \frac{x(1+2x+3x^2)}{1+x+x^2+x^3}\,\varepsilon_x.$$

La quantità $\left|\frac{x(1+2x+3x^2)}{1+x+x^2+x^3}\right|$ è illimitata per $x \to -1$ quindi il problema è mal condizionato per valori prossimi a -1. Risulta invece ben condizionato per $x \to \pm\infty$ in quanto $\lim_{x\to\pm\infty}\left|\frac{x(1+2x+3x^2)}{1+x+x^2+x^3}\right| = 3$.

Per l'errore algoritmico, nel primo caso, cioè quando $f(x) = \frac{1-x^4}{1-x}$ abbiamo la seguente limitazione $|\varepsilon_{alg}| < (3 + 3\frac{x^4}{|1-x^4|})u$. Questa espressione non è limitabile per $x \to \pm 1$. Questo algoritmo è quindi instabile per valori di x prossimi ad ± 1. L'algoritmo è invece stabile per $|x| \to \infty$.

Nel secondo caso, detta $z_1 = 1 + x, z_2 = z_1 + x^2, z_3 = z_2 + x^3$, abbiamo che

$$|\varepsilon^{(2)}_{alg}| < \frac{|z_1| + |z_2| + |z_3|}{|z_3|}u + \frac{x^2 + |x^3|}{|z_3|}u.$$

Poichè z_3 si annulla per $x = -1$ abbiamo che l'errore è illimitato per $x \to -1$. Non abbiamo instabilità invece pr $x \to 1$, o per $|x| \to \infty$.

Esercizio 8. Per un qualsiaiasi $x > 0$ si dica quanto dovrebbe valere x dopo aver eseguito il seguente frammento di programma

```
for k=1:60
       x=sqrt(x);
end
for k=1:60
       x=x*x;
end
```

Si esegua questo script in Matlab e si veda quale valore assume x alla fine del programma e si cerchi di dare una motivazione del risultato ottenuto.

Soluzione 8. In teoria, all'uscita dello script dovrei ottenere il valore x iniziale, in quanto $(x^{2^{-60}})^{2^{60}} = x$. In realtà si osserva che per quasi tutti i valori abbiamo che dopo aver eseguito 60 volte la radice quadrata otteniamo $x = 1$. I successivi 60 elevamenti a potenza lasciano inalterato il valore di x uguale a 1. In altre situazioni, posso ottenere il valore 0 o Inf. Comunque si ottiene sempre un valore molto lontano dall' x di partenza.

Facendo l'analisi dell'errore, indicando con ε_i l'errore locale dovuto all' i-esima estrazione della radice quadrata e con η_i l'errore locale dovuto all' i-esimo elevamento a potenza, e sapendo che il coefficiente di amplificazione della radice è $1/2$ e del quadrato è 2, otteniamo

$$\varepsilon_{alg} \doteq \sum_{k=0}^{59} 2^i \eta_i + \sum_{k=1}^{60} 2^i \varepsilon_i.$$

Ricordandp che nel caso di Matlab $u = 2^{-52}$, si ottiene quindi che $|\varepsilon_{alg}| \leq 3(2^{60} - 1)\,u = 3 \cdot 2^8$ indicante che nessuna cifra del risultato è corretta. Da questo calcolo si ottiene

che, in generale, se anzichè ripetere 60 volte i cicli for, questo sono ripetuti per h volte, l'errore si maggiora con

$$|\varepsilon_{alg}| \doteq |\sum_{k=0}^{h} 2^i \eta_i + \sum_{k=1}^{h} 2^i \varepsilon_i| \leq 3(2^h - 1)\, u.$$

Ad esempio, per avere 10 cifre significative, occorre che prendere $h < 19$.

Esercizio 9. Si vuole calcolare il valore del polinomio

$$p(x) = x^n + 2x^{n-1} + 2^2 x^{n-2} + ... + 2^{n-1}x + 2^n, \quad n > 1,$$

con il metodo di Ruffini-Horner per $0 < x < 1$.

a) Si studi l'errore algoritmico per il caso $n = 3$, in cui $p(x) = x(x(x + 2) + 4) + 8$.

b) Si studi l'errore algoritmico per n generico.