Calcolo Numerico

Luca De Paulis

20 febbraio 2021

INDICE

1 ARITMETICA DI MACCHINA

1.1 RAPPRESENTAZIONE DEI NUMERI DI MACCHINA

Il primo problema da risolvere quando si vuole fare analisi numerica è scegliere un metodo per rappresentare i numeri reali su una macchina. Infatti un generico numero reale ha potenzialmente una scrittura decimale infinita, dunque essendo le risorse disponibili in una macchina *finite* dobbiamo trovare un modo di approssimarlo.

1.1.1 Virgola fissa

Il primo metodo è il metodo **a virgola fissa**: in questo metodo si rappresentano tutti i numeri nella loro forma decimale normale e si considerano esattamente k cifre dopo la virgola.

Questo metodo è molto semplice e ci consente di fare operazioni elementari (come le somme o i prodotti) immediatamente ("in colonna"), tuttavia ha anche degli svantaggi evidenti, come

- il range dei numeri rappresentabili su n cifre/bit è piccolo;
- siccome il numero di bit dedicato ai numeri dopo la virgola è basso, la precisione è molto bassa e assolutamente non adeguata ad applicazioni di analisi numerica.

Per questo è stata inventata la rappresentazione in virgola mobile.

1.1.2 Virgola mobile

Innanzitutto dobbiamo trovare un modo standard per rappresentare un qualsiasi numero reale. Per far ciò ci viene in aiuto il seguente teorema.

Teorema 1.1.1 **Teorema di rappresentazione in base.** *Sia $x \in \mathbb{R}$ e sia $\beta \geqslant 2$ una base di rappresentazione. Allora esistono e sono univocamente determinati un **esponente** $p \in \mathbb{Z}$ e una successione di **cifre** $(d_i)_{i\in\mathbb{N}}$ (con $d_i \in \mathbb{Z}$ per ogni i) per cui vale che*

(i) $d_1 \neq 0$;

(ii) *per ogni* i *vale che* $0 \leqslant d_i \leqslant \beta - 1$;

(iii) *la successione* d *non è definitivamente uguale a* $\beta - 1$, *ovvero per ogni* $k > 0$ *esiste un* $j \geqslant k$ *tale che* $d_j \neq \beta - 1$;

(iv) *vale che*

$$x = \operatorname{sgn}(x) \cdot \beta^p \cdot \left(\sum_{i=1}^{\infty} d_i \beta^{-i} \right).$$

La rappresentazione di x nella forma data dal Teorema 1.1.1 si dice **rappresentazione normalizzata in virgola mobile**.

Esempio 1.1.2. Consideriamo ad esempio il numero reale 123. Per il Teorema 1.1.1 possiamo esprimere 123 come

$$123 = +10^3 \cdot (0.123) = +10^3 \cdot (1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3}).$$

Questa rappresentazione è l'unica possibile se imponiamo che d_1 sia diverso da 0 e che le cifre della rappresentazione siano tutte considerate come *cifre decimali* moltiplicate per una certa potenza della base (in questo caso 3).

La rappresentazione data dal Teorema 1.1.1 è essenzialmente il concetto di *notazione scientifica*: in notazione scientifica portiamo il numero reale in modo che abbia una singola cifra diversa da zero prima della virgola, mentre nella notazione data dal Teorema quella cifra diventa la prima cifra dopo la virgola. Inoltre, il Teorema ci garantisce che questa rappresentazione non è solamente valida in base 10, ma lo è in qualsiasi base (noi lo useremo in particolare in base 2).

Il fattore β^p viene detto **esponente**, mentre la parte decimale (corrispondente alla sommatoria) viene detta **mantissa**.

Osservazione 1.1.1. Le ipotesi (i) e (iii) del Teorema sono fondamentali per l'unicità della rappresentazione. In effetti

- se venisse a mancare la prima condizione avremmo che

$$123 = +10^3 \cdot (0.123) = +10^4 \cdot (0.0123) = +10^5 \cdot (0.00123) = \dots;$$

- se venisse a mancare la terza condizione (che ci dice che la successione d non può terminare con una sequenza infinita di $(\beta - 1)$, ovvero il numero non può finire con $\beta - 1$ periodico) avremmo dei problemi più subdoli, che derivano dalla non esistenza del "9 *periodico*" (dimostrata in appendice, Proposizione A.1.1).

  Infatti in qualsiasi base β il numero $0.\overline{(\beta - 1)}$ è uguale a 1, quindi se ammettessimo entrambe le rappresentazioni verrebbe meno l'unicità.

Notazione. Useremo la notazione $(n)_\beta$ per riferirci al numero n espresso in base β.

Ad esempio il numero $(1011)_2$ si riferisce al numero

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11 \text{ (in base 10)}.$$

Per rappresentare i numeri reali in macchina dobbiamo quindi risolvere ancora due problemi:

- la macchina usa (nella maggior parte dei casi) la base 2, quindi dobbiamo poter trasformare da base 10 in base 2;
- dobbiamo rappresentare i numeri infiniti in modo approssimato.

Per quanto riguarda il primo, l'algoritmo per trasformare un numero decimale (in base 10) in un numero in base 2 è il seguente:

1. si trasforma la parte intera (le cifre prima della virgola) in base 2 tramite divisioni successive per 2;
2. si trasforma la parte decimale (le cifre dopo la virgola) in base 2 tramite *moltiplicazioni* successive per 2.

Esempio 1.1.3. Trasformiamo 3.15 in base 2: ovviamente $(3)_{10} = (11)_2$ e quindi rimane solo da trasformare la parte decimale.

L'algoritmo consiste nel prendere il numero decimale 0.15 e moltiplicarlo per 2 ripetutamente: se il risultato ha come cifra prima della virgola uno 0 aggiungiamo uno 0 alla rappresentazione in base 2, altrimenti un 1. Nella pratica:

$$\begin{aligned}
0.15 \cdot 2 &= \boxed{0}.30 \\
0.30 \cdot 2 &= \boxed{0}.60 \\
0.60 \cdot 2 &= \boxed{1}.20 \\
0.20 \cdot 2 &= \boxed{0}.40 \\
0.40 \cdot 2 &= \boxed{0}.80 \\
0.80 \cdot 2 &= \boxed{1}.60 \\
0.60 \cdot 2 &= \boxed{1}.20 \\
&\vdots
\end{aligned}$$

Osserviamo che ripetendo il procedimento la sequenza $(0.60, 0.20, 0.40, 0.80)$ si ripete all'infinito, dunque il numero in base 2 termina con le cifre 1001 ripetute periodicamente. La parte decimale in base 2 corrisponde quindi a $(0.00\overline{1001})_2$.

Il numero completo è quindi

$$(11.00\overline{1001})_2 = 2^2 \cdot (0.1100\overline{1001})_2.$$

Per rappresentare i numeri infiniti dobbiamo approssimare la parte decimale, considerando solo un numero finito t di cifre dopo la virgola. Per far ciò esistono due metodi (che esamineremo più nel dettaglio nel seguito):

- nel caso del **troncamento** si considerano le prime t cifre dopo la virgola e si scartano tutte le successive;
- nel caso dell'**arrotondamento** analizziamo la cifra decimale di posto $t+1$:
  - se d_{t+1} appartiene all'intervallo $\left[0, \frac{\beta}{2}\right)$ (dove β è la base) consideriamo semplicemente le prime t cifre;
  - altrimenti (se $d_{t+1} \in \left[\frac{\beta}{2}, \beta\right]$ aggiungiamo 1 all'ultima cifra decimale (come riporto, quindi lo propaghiamo in caso sia necessario).

Esempio 1.1.4. Riprendiamo il numero considerato precedentemente, ovvero

$$(3.15)_{10} = 2^2 \cdot (0.1100\overline{1001})_2$$

e approssimiamolo a $t = 8$ cifre decimali.

Nel caso del troncamento basta considerare le prime 8 cifre decimali, quindi il numero troncato è

$$2^2 \cdot (0.11001001)_2.$$

Nel caso dell'arrotondamento invece dobbiamo considerare la nona cifra decimale: espandendo la rappresentazione decimale osserviamo che la nona cifra è

$$0.11001001\underline{1}00\ldots;$$

siccome $1 \in \left[\frac{2}{2}, 2\right] = [1, 2]$ dobbiamo *approssimare per eccesso*, ottenendo

$$2^2 \cdot (0.11001010)_2.$$

1.1.3 Insieme dei numeri di macchina

Siccome un singolo numero può occupare una quantità fissa in memoria (tipicamente 32 o 64 bit) dobbiamo fissare dei limiti per l'esponente e per il numero di cifre decimali che possiamo usare per la rappresentazione. Osserviamo inoltre che, se le nostre risorse sono limitate, aumentando il numero di cifre decimali disponibili dobbiamo necessariamente diminuire lo spazio dedicato a memorizzare l'esponente e viceversa.

Definizione 1.1.5

Insieme dei numeri di macchina. Sia $\beta \geqslant 2$ una base, t il numero di cifre decimali utilizzabili e siano m, $M \in \mathbb{Z}$ gli estremi per l'esponente (ovvero ogni esponente p rappresentabile deve appartenere a $[-m, M]$). Si dice allora **insieme dei numeri di macchina** l'insieme

$$\mathcal{F}(\beta, t, m, M) \coloneqq \{0\} \cup \Big\{ x \in \mathbb{R} \ : x = \operatorname{sgn}(x) \cdot \beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}, \quad d_1 \neq 0, \quad 0 \leqslant d_i \leqslant \beta - 1, \quad -m \leqslant p \leqslant M \Big\}.$$

Osservazione 1.1.2. Nelle condizioni in cui specifichiamo quali numeri sono rappresentabili compaiono le ipotesi (i) e (ii) del Teorema di rappresentazione in base, ma non l'ipotesi (iii). Quest deriva dal fatto che i numeri di macchina hanno una precisione finita (ovvero hanno al più t cifre decimali), dunque non è possibile avere un numero che termina in $\beta - 1$ periodico.

OMEGA GRANDE Chiamiamo Ω il più grande numero di macchina. Possiamo costruirlo in questo modo:

- scegliamo come segno $+$;
- scegliamo come esponente il massimo esponente possibile, ovvero $p = M$;
- poniamo tutte le t cifre decimali uguali al massimo possibile, che in base β è $\beta - 1$.

Segue quindi che

$$\Omega \coloneqq +\beta^M \sum_{i=1}^{t} (\beta - 1)\beta^{-i}.$$

Semplificando l'espressione di Ω tramite la formula per la serie geometrica si ottiene

$$\begin{aligned}
\Omega &= +\beta^M \sum_{i=1}^{t} (\beta - 1)\beta^{-i} \\
&= \beta^M (\beta - 1) \sum_{i=1}^{t} (1/\beta)^i \\
&= \beta^M (\beta - 1) \left(\sum_{i=0}^{t} (1/\beta)^i - (1/\beta)^0 \right) \\
&= \beta^M (\beta - 1) \left(\frac{1 - (1/\beta)^{t+1}}{1 - 1/\beta} - 1 \right)
\end{aligned}$$

Moltiplicando numeratore e denominatore della frazione per β:

$$\begin{aligned}
&= \beta^M(\beta-1)\left(\frac{\beta-(1/\beta)^t}{\beta-1}-1\right)\\
&= \beta^M(\beta-1)\cdot\frac{\beta-(1/\beta)^t-\beta+1}{\beta-1}\\
&= \beta^M\left(1-\beta^{-t}\right).
\end{aligned}$$

Esempio 1.1.6. Consideriamo l'insieme $\mathcal{F}(10,3,2,2)$: in questo sistema possiamo rappresentare numeri in base 10 con al massimo 3 cifre decimali e con un esponente compreso tra -2 e 2.

Il massimo numero rappresentabile in questo sistema è

$$\Omega = +10^2(0.999) = 10^2(1-10^{-3}).$$

OMEGA PICCOLO Chiamiamo ω il più piccolo numero *positivo* rappresentabile in un sistema. Possiamo costruirlo in questo modo:

- siccome è positivo, il segno è necessariamente $+$;
- come esponente scegliamo il più piccolo esponente possibile, ovvero $p = -m$;
- siccome le cifre decimali non possono essere tutte uguali a 0 poniamo $d_1 = 1$ e $d_i = 0$ per ogni $i > 1$.

Segue quindi che

$$\omega = +\beta^{-m}\left(1\cdot\beta^{-1}\right) = \beta^{-m}(0.1)_\beta.$$

CARDINALITÀ DELL'INSIEME DEI NUMERI DI MACCHINA Dati β, t, m, M calcoliamo $\#\mathcal{F}(\beta,t,m,M)$. Osserviamo che

1. dalla definizione di $\mathcal{F}(\beta,t,m,M)$, dobbiamo contare lo 0 separatamente;
2. per quanto riguarda i numeri non-nulli, per ogni possibile combinazione di esponente/mantissa abbiamo 2 scelte per il segno (segno positivo o negativo);
3. le scelte per i possibili esponenti corrispondono a tutti i numeri interi nell'intervallo $[-m, M]$, che sono $M+m+1$;
4. per quanto riguarda la mantissa dobbiamo scegliere t cifre comprese tra 0 e $\beta-1$ (e lo possiamo fare in β^t modi), ma dobbiamo escludere tutte quelle che iniziano per 0, che sono β^{t-1}: abbiamo quindi $\beta^t-\beta^{t-1}$ scelte per la mantissa di un numero.

In totale si ha quindi che

$$\#\mathcal{F}(\beta,t,m,M) = 1+2(M+m+1)\left(\beta^t-\beta^{t-1}\right).$$

Esempio 1.1.7. Ad esempio se $\beta=2$, $t=3$ e $m=M=1$ si ha che

$$\#\mathcal{F}(2,3,1,1) = 1+2(1+1+1)\left(2^3-2^2\right) = 25.$$

1.1.4 Standard IEEE (754)

Lo standard usato al giorno d'oggi per implementare i numeri a virgola mobile è lo standard IEEE (754). Esso definisce due rappresentazioni distinte, una *single precision*, dove ogni numero è rappresentato su 32 bit, ovvero su 4 byte, e una *double precision*, dove ogni numero è rappresentato su 64 bit (o equivalentemente 8 byte). In particolare noi analizzeremo solo il secondo.

I 64 bit vengono divisi nel seguente modo:

- un singolo bit di segno (0 per i numeri positivi, 1 per i negativi);
- 11 bit per rappresentare l'esponente;
- i restanti 52 bit vengono usati per rappresentare la mantissa.

Tuttavia possiamo ottimizzare un po' lo spazio usato facendo delle semplici osservazioni:

1. siccome siamo in base 2 la condizione $d_1 \neq 0$ ci dice che necessariamente $d_1 = 1$: ciò significa che rappresentare d_1 nella mantissa è inutile (ogni mantissa inizierebbe per 1) e quindi possiamo rappresentare le cifre da d_2 in poi. Questo significa che abbiamo in realtà $t = 53$ cifre decimali a disposizione;
2. per evitare di usare un bit per il segno dell'esponente lo rappresentiamo in *traslazione* (anche detto *bias* o *eccesso a*): invece di rappresentare il vero esponente p rappresentiamo $p + 1022$. Questo ci consente di indicare esponenti negativi senza dover usare strategie di complemento a 2.

L'insieme dei numeri di macchina definito dallo standard IEEE (754)è quindi

$$\mathcal{F}(2, 53, 1021, 1024).$$

Osserviamo che questa scelta per i limiti degli esponenti non copre tutte le combinazioni possibili su 11 bit: in particolare possiamo rappresentare $1024 + 1021 + 1 = 2046$ esponenti diversi, mentre su 11 bit potenzialmente potremmo rappresentarne fino a $2^{11} = 2048$.

Questo ci permette di rappresentare dei numeri particolari.

ZERO Abbiamo visto che (per il Teorema 1.1.1) lo 0 non ammette una rappresentazione normalizzata. Nello standard IEEE (754)esso viene realizzato ponendo a 0 tutti i campi dell'esponente e della mantissa; tuttavia non viene specificato il segno, per cui esistono due rappresentazioni uguali per lo zero (-0 e $+0$).

Questa rappresentazione non va in conflitto con i numeri normalizzati, in quanto ponendo il campo dell'esponente uguale a undici 0 otterrei un esponente effettivo uguale a -1022, mentre il minimo rappresentabile è -1021.

INFINITI E NAN Ponendo il campo dell'esponente uguale a $(11111111111)_2$ (ovvero undici cifre 1) ottengo una rappresentazione che non appartiene all'insieme definito prima (in quanto il massimo esponente rappresentabile è 1024, che in eccesso a 1022 diventa $2046 = (11111111110)_2$). Sfruttando questo fatto possiamo rappresentare tre "numeri" diversi:

- se il campo della mantissa non è formato da tutti 0 il risultato è NaN (*Not a Number*): questa configurazione viene usata ad esempio per i risultati delle forme indeterminate, come $0/0$;
- se il campo della mantissa è formato da tutti 0 il numero viene interpretato come $\pm\infty$ (a seconda del bit di segno): ciò viene usato per rappresentare l'overflow, ovvero numeri che sono più grandi di Ω (o più piccoli di $-\Omega$).

NUMERI DENORMALIZZATI Se il campo dell'esponente contiene solo zeri ma la mantissa ha almeno un bit diverso da 0 stiamo rappresentando numeri *denormalizzati*. Un tale numero viene interpretato come se fosse $d_1 = 0$ e quindi ci consente di ottenere dei numeri in valore assoluto più piccoli di ω, risolvendo parzialmente i problemi di *underflow*.

Tuttavia questi numeri hanno una precisione minore dei numeri normalizzati, in quanto i primi k bit della mantissa possono essere tutti uguali a 0. Questo significa che invece di avere 53 cifre di precisione ne abbiamo $53 - k$ (dove k dipende dal numero denormalizzato scelto), dunque dobbiamo porre ancora più attenzione agli eventuali errori di precisione.

A APPENDICE A

A.1 TEOREMI DI CONTORNO

Proposizione A.1.1 **9-periodico.** *In base 10 i numeri $0.\overline{9}$ e 1 sono uguali.*

Forniamo due diverse dimostrazioni di questa proposizione.

Prima dimostrazione. Dalle formule per trasformare i numeri periodici in frazioni sappiamo che $0.\overline{9} = 9/9 = 1$. □

Seconda dimostrazione. Espandendo la definizione di numero periodico otteniamo che

$$0.\overline{9} = 0.999\cdots = 9\cdot 10^{-1} + 9\cdot 10^{-2} + 9\cdot 10^{-3} + \cdots = \sum_{i=1}^{\infty} 9\cdot 10^{-i}.$$

Sfruttando la formula della serie geometrica si ottiene che

$$\begin{aligned}
\sum_{i=1}^{\infty} 9\cdot 10^{-i} &= 9\cdot \sum_{i=1}^{\infty} \left(\frac{1}{10}\right)^{i} \\
&= 9\cdot \left(\left(\sum_{i=0}^{\infty} \left(\frac{1}{10}\right)^{i}\right) - \left(\frac{1}{10}\right)^{0}\right) \\
&= 9\cdot \left(\frac{1}{1-1/10} - 1\right) \\
&= 9\cdot \left(\frac{10}{9} - 1\right) \\
&= 9\cdot \frac{1}{9} \\
&= 1.
\end{aligned}$$

□

La proposizione vale in generale in una base β qualsiasi ($\beta \geqslant 2$).

Proposizione A.1.2 *In base β ($\beta \geqslant 2$) vale che $0.\overline{(\beta-1)} = 1$.*

Dimostrazione. La dimostrazione è uguale alla seconda dimostrazione della Proposizione A.1.1. □