Università di Pisa

Dipartimento di Informatica
Corso di Laurea Triennale in Informatica

Corso 2° anno - 6 CFU

Calcolo Numerico

Professore:
Prof. Luca Germignani

Autore:
Matteo Giuntoni

Anno Accademico 2022/2023

Contents

Calcolo Numerico

Realizzato da: Giuntoni Matteo

A.A. 2022-2023

---

1 Aritmetica di Macchina

Per una macchina la scrittura $(x + y) + z \neq x + (y + z)$. Vediamo dunque che ci sono alcuni punti focali da considerare per far si che una macchina funzioni correttamente:

- Trovare uno standard per come memorizzare i numeri.
- Trovare uno standard per come manipolare i numeri.

Da questi due punti possiamo ricondurci ad un solo problema, come andare a rappresentare i numeri.

1.1 Teorema di rapresentazione

Teorema 1.1.1. Dato $x \in \mathbb{R}, x \neq 0$ esistono e sono univocamente determinati.

1. un intero $p \in \mathbb{Z}$ detto esponente della rappresentazione.
2. una successione di numeri naturiali $\{d_i\}_{i\geq 1}$ con $d_i \neq 0, 0 \leq d_i \leq B-1$ e d_i non definitvamente uguali a $B-1$, dette cire della rappresentazione; tali per cui si ha

$$x = sign(x)B^p \sum_{i=1}^{+\infty} d_i B^{-1}. \tag{1}$$

Andiamo ora ad analizzare il significato di questo teorema. Esso descrive quella che viene chiamata rappresentazione in virgola mobile, in quanto l'esponente p on è determinato in modo da avere la parte intera nulla. Le cose da considerare in questo teorema sono:

- La condizione $d_i \neq 0$ e d_i non definitivamente uguale a $B-1$ sono introdotte per garantire l'unicità delle rappresentazioni. Ad esempio:

  $$B = 10 \text{ abbiamo } 1 = +10^1(1 \cdot 10^{-1}) = +10^2(0 \cdot 10^{-1} + 1 \cdot 10^{-1})$$

  Quindi due rappresentazioni diverse per lo stesso numero, però considerando le condizioni scritte sopra la seconda non risulta accettabile perché la prima cifra è nulla.

- Il caso $x = 0$ non ammette rapresentazione normalizzata. Questa casistica viene trattata dalla macchina in un modo particolare, per questo abbiamo la condizione $x \neq$.
- Questa rapresentazione si estende anche all'insieme dei numeri complessi del tipo $z = a + ib$, utilizzando una rapresentazione come coppie di numeri reali del tipo (a, b).

Possiamo dedurre che visto che stiamo lavornano con registri di meoria di un calcolatore con memoria a numero finito, anche la quantità di cifre rapresentabili saranno a numero finito esso vinene chiamato **insieme dei numeri di macchina**.

Dal teorema di rapresentazione in base di un numero reale può avvenire assegnando delle posizioni di meoria per il segno, per l'esponente e per le cifre della rappresentazione.

Definizione 1.1.1 (Insieme die numeri di macchina)**.** *Si definisce l'insieme dei nuermi di macchina in rappresentazione floating point con t cifre, base B e range $-m, M$ l'insieme dei numeri reali.*

$$\mathbb{F}(B,t,m,M) = \{0\} \cup \{s \in \mathbb{R} \,:\, x = sign(x)B^p \sum_{i=1}^{t} d_i B^{-1}, 0 \leq d_i \leq B-1, d_1 \neq 0, -m \leq p \leq M\}$$

Si osserva in questa definizione che:

- L'insieme $\mathbb{F}$ ha cardinalità finita $N = 2B^{t-1}(B-1)(M+m+1)+1$.
- L'insieme dei numeri di macchina $\mathbb{F}(B,t,m,M)$ è simmetrico rispetto all'origine.
- Possiamo definire $\Omega = B^M(B-1)\sum_{i=1}^{t} b^{-1}$ come il più grande numero macchina e $\omega = B^{-m}B^{-1}$ come invece il più piccolo.
- Posto un $x = B^P \sum_{i=1}^{t} d_i B^{-1}$ possiamo definire il suo successiovo numero di macchina come $y = B^p(\sum_{i=1}^{t-1} d_i B^{-1} + (d_t+1)B^{-t})$. Da qui vediamo che la distanza $y - x = B^p - t$ porta i numeri ad essere non equispaziali fra di loro, quindi la distanza aumento con l'avicinarci a Ω

Esempio 1.1.1. Facciamo ora un esempio in cui andiamo a rappresentare il numero successivo di $x = B^p \sum_{t}^{i=1} d_i B^{-1}$. Esso si può scrivere come $y = B^p\left(\sum_{i=1}^{t-1} d_i B^{-1} + (d_t+1)B^{-t}\right)$.
Mentre si può scrivere la distanza fra questi due valori come $y - x = B^p - t$.

E' stato fissato uno standard IEEE 754 fra gli anni 70/80, questo standard dice che, visto ci sono macchine che hanno metodi di rappresentazione diversi bisogna fissare un standard, esso appunto dice che $B = 2$ ed i registri sono a 32 o 64 bit.

Questa rapresentazione ha uno svantaggio che può sembrare minimo ma non lo è, lo 0 si rappresenta due volte con $-0, +0$. Per ovviare a questo problema si è andato ad abbandonare questa rappresentazione in esponenti ma si rapprensentato i numeri nel seguente modo: $p_1 2^0 + p_1 2^1 + \cdots + p_1 1 s^1 0$ che rappresentano numeri da 0 a $2^1 1 - 1$ quindi 2047 numeri, mentre lo 0 si può scrivere come:

- O tenendo tutti i valori a 0
- Oppure tendendo tutti i valori a 1

In entrambi i casi abbiamo un range di valori che va da $[-1022, 1024]$. A questo punto ho 2^{P-1022} numeri che la macchina rappresneta come $\pm 2^{P-1022}(0.1d_1 \dots d_{52})$.
Impostando questo standard abbiamo $\Omega = 2^{1024}(01\dots1)_2$ e $\omega = 2^{-1022}(101)_2$.

Osservazione 1.1.1. Quando $p = 0$ abbiamo i numeri che si trovano nella porzione della retta dei numeri che è compresa fra $-\omega$ ω e possiamo qui avere anche tuti 0 e quindi si introduce il caso dei numeri denormalizzati.

Se abbiamo l'esponante uguale a tutti 1, la convensione è che tutte le cifre della mantissa sono tutti uguali a 0/1 questo numero indica il $\pm\infty$ altrimenti sta a signifiare NaN (not a number). Questi valori ci permettono di gestire forme indeterminate.

1.2 Errore di rappresentazione

Quando si va a rappresentare un numero reale non nullo $x \in \mathbb{R}$ e con $x \neq 0$ si può andare a commettere degli errori di rappresentazione detto anche **errore relativo di approssimazione**, e di definisce come, prendendo un $\tilde{x} \in \mathbb{F}(B,t,m,M)$

$$\epsilon_x = \frac{\tilde{x} - x}{x} = \frac{\eta x}{x}, x \neq 0$$

Definiamo $|\epsilon_x| = \left|\frac{\tilde{x}-x}{x}\right| \leq \frac{B^{P-t}}{|x|} \leq \frac{B^{P-t}}{B^{P-1}} = B^{1-t} = u$ la u è definita come **precisione di macchina**.

Andiamo inoltre a definire le conidizioni di underflow e overflow. Dato un $x \in \mathbb{R}, x \neq 0$ abbiamo che:

1. Se $|x| < \omega$ o $|x| > \Omega$ overflow. In questo caso si va ad associare il $+\infty$.
2. Se invece $\omega \leq |x| \leq \Omega$ abbiamo underflow. In questo caso allora prendiamo una $x = B^p \sum_{i=1}^{\infty} d_i B^{-1} \longrightarrow B^p \sum_{i=1}^{t} d_i B^{-1} = \tilde{x}$ che è una approssimazione

1.3 Operazioni di macchina

Consideriamo ora due numeri $x, y \in \mathbb{F}$ e chiediamoci perché la macchina non possa fare l'operazione $x + y$. La risposta è che i risultati da questa operazione di ritornano fra i numeri di macchina. Per ovviare a questo problema dovremo usare le Operazioni di macchina che si identificano come $\oplus \ominus \otimes \oslash$. Nel nostro caso l'addizione di macchina $x \oplus y = troncamento(x+y) = (x+y)(1+\epsilon_1)$ con $|\epsilon_1| \leq u$ con e_1 detto errore locale dell'operazione.

Esempio 1.3.1. Supponiamo di dover calcolare in macchina la funzione $f(x) = \frac{x-1}{x}$. In macchina questa funzione corrisponderebbe a $g(\tilde{x}) = (\tilde{x} \ominus 1) \oslash \tilde{x}$. Abbiamo quinidi:

$$g(\tilde{x}) = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1)}{x(1+\epsilon_x)} \cdot (1+\epsilon_1) = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1+\epsilon_2)}{x(1+\epsilon_x)} = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1+\epsilon_2-\epsilon_x)}{x}$$

$$g(\tilde{x}) = (\tilde{x} \oplus 1) \oslash \tilde{x} = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1+\epsilon_2-\epsilon_x)}{x}$$

$$\frac{g(\tilde{x}-f(x))}{f(x)} = \frac{((x-1)/x) + (\epsilon_1+\epsilon_2)((x-1)/x) + \epsilon_2/x - ((x-1)/x)}{(x-1)/x} = \epsilon_1 + \epsilon_2 - \frac{\epsilon_x}{x-1}$$

Esempio 1.3.2. Supponiamo ora di calcolare la funzione $f(x) = \frac{x-1}{x}$ in un altro modo, $g_2(\tilde{x}) = \frac{g_2(\tilde{x})-f(x)}{f(x)}$ ed andiamo a fare l'analisi dell'errore.

$$\frac{g_1(\tilde{x}) - f(x)}{f(x)} \doteq \epsilon_1 + \epsilon_2 + \frac{\epsilon_1}{(x-1)} \text{ Questo è il risultato di un analisi al primo ordine}$$

$$g_2(\tilde{x}) = 1 \ominus \frac{1}{\tilde{x}}(1+\delta_1) = 1 \ominus \frac{1}{x}(1+\delta_1)(1-\epsilon_x) = [1 - \frac{1}{x}(1-\delta_1)(1-\epsilon_1)](1+\delta_2)$$

$$\doteq (1+\delta_1) - \frac{1}{x}(1+\delta_1+\delta_2+\epsilon_x) \doteq (1-\frac{1}{x}) + \delta_2(1-\frac{1}{x}) - \frac{\delta_1}{x} + \frac{\epsilon_x}{x} \doteq \delta_2 - \frac{\delta_1}{x-1} + \frac{\epsilon_x}{x-1}$$

$$\frac{g_2(\tilde{x}) - f(x)}{f(x)} = \delta_2 - \frac{\delta_1}{x-1} + \frac{\epsilon_x}{x-1}$$

Questo è il risultato finale dove $\delta_2 - \frac{\delta_1}{x-1}$ viene definita come parte stabilità mentre $\delta_2 - \frac{\delta_1}{x-1}$ viene chiamato condizionamento,il risulato finale viene definito invece numero stabile.

2 Calcolo degli errori

Supponiamo di avere una funzione $f : [a, b] \to \mathbb{R}$ e $f \neq 0$, per andare a calcolare questa funzione come già visto usiamo un algoritmo che esprime tale valore come risutlato di una sequanza di operazioni aritmentiche. Questa rappresentazione come abbiamo già potuto verificare con esempi produce degli errori di approssimazione. Questi errori possono essere suddivisini in 3 tipologie.

Definizione 2.0.1 (Errore inerente o inevitabile)**.** *Si dice errore inerente o inevitabile generato nel calcolo di $f(x) \neq 0$ la quantità:*

$$\epsilon_{in} = \frac{f(\tilde{x}) - f(x)}{f(x)}$$

Definizione 2.0.2 (Errore algoritmico)**.** *Si dice errore algoritmico generato nel calcolo di $f(x) \neq 0$ la quantità:*

$$\epsilon_{alg} = \frac{g(\tilde{x}) - f(\tilde{x})}{f(\tilde{x})}$$

Definizione 2.0.3 (Errore totale)**.** *Si dice errore algoritmico totale nel calcolo di $f(x) \neq 0$ mediante l'algoritmo specificato da g la quantità:*

$$\epsilon_{tot} = \frac{g(\tilde{x}) - f(x)}{f(x)}$$

Osservazione 2.0.1. Vediamo che se $|\epsilon_{in}|$ è grande il problema si definisce **problema mal condizionato**. Mentre se $|\epsilon_{alg}$ è grande l'algoritmo si dice che **algoritmo è numericamente instabile**.