

1 Lezione del 28-02-25

1.1 Operazioni sui numeri macchina

Abbiamo introdotto l'insieme dei numeri macchina. Vediamo adesso come eseguire **operazioni** fra elementi di questi insiemi.

Notiamo che, di base, dati $x, y \in F(\beta, m, L, U)$, non necessariamente $x \circ y \in F(\beta, m, L, U)$ per le comuni operazioni aritmetiche $+, -, \times, \div$.

Quello che faremo è quindi approssimare tali operazioni, cioè dire:

- $x \oplus y = Rn(x + y)$
- $x \ominus y = Rn(x - y)$
- $x \otimes y = Rn(x \times y)$
- $x \oslash y = Rn(x \div y)$

Effetto di questa approssimazione è negare proprietà note dei reali, come ad esempio l'associativa:

$$(x \oplus y) \oplus z \neq x \oplus (y \oplus z)$$

$$(x \oplus y) \otimes z \neq x \oplus (y \otimes z)$$

Cioè, la valutazione di una formula con ordini diversi ma equivalenti in aritmetica esatta può portare a risultati differenti nell'insieme dei numeri di macchina.

1.1.1 Errore nel calcolo di funzione

Sia $f : \mathbb{R}^m \rightarrow \mathbb{R}$, e si voglia calcolare $f(P_0)$, con $P_0 = (x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}) \in \mathbb{R}^m$. Le operazioni aritmetiche $+, -, \times, \div$ possono essere viste come funzioni di questo tipo. Ci interroghiamo quindi sulla fonte dell'errore nella loro valutazione:

1. Nel caso contenga funzioni irrazionali o trascendenti, f verrà approssimata con una funzione \bar{f} che coinvolge solo operazioni aritmetiche di base $+, -, \times, \div$;
2. \bar{f} viene tradotta in un *algoritmo* \bar{f}_a , ovvero in una formula che coinvolge $\oplus, \ominus, \otimes, \oslash \leftarrow +, -, \times, \div$;
3. Potrebbe essere che $P_0 \notin F(\beta, m, L, U)$, e quindi viene approssimato a $P_1 = Rn(P_0)$.

Quindi, vogliamo $f(P_0)$ ma possiamo solo approssimarla come $\bar{f}_a(P_1)$.

Ad esempio, poniamo di voler calcolare e^π . I passaggi nell'ordine appena visto saranno:

1. Approssimiamo l'esponenziale al secondo grado dello sviluppo di Taylor:

$$e^x \approx 1 + x + \frac{x^2}{2} = \bar{f}(x)$$

2. Si riporta la $\bar{f}(x)$ a $\bar{f}_a(x)$:

$$1 \oplus (x \oplus ((x \otimes x) \oslash 2))$$

3. Si approssima π al numero macchina più vicino:

$$Rn(\pi) = 3.1415 = P_1$$

Avremo quindi la formula finale:

$$1 \oplus (P_1 \oplus ((P_1 \otimes P_1) \odot 2))$$

riporta grafo (o albero) che la rappresenta

Diamo quindi la definizione di **errore totale**:

Definizione 1.1: Errore totale

Data $f : \mathbb{R}^m \rightarrow \mathbb{R}$, un punto $P_0 \in \mathbb{R}^m$ ed un algoritmo \bar{f}_a , l'errore totale è dato da:

$$S_f = \bar{f}_a(P_1) - f(P_0), \quad P_1 = Rn(P_0)$$

1.1.2 Errore di funzioni razionali

Assumiamo per adesso f **funzione razionale** ovvero f si può definire con un numero di operazioni in $+, -, \times, \div$. Assumere funzioni razionali ci permette di prendere $f = \bar{f}$ e $f_a = \bar{f}_a$ (non ci sono irrazionali da riportare ai razionali). Potremo allora dire:

$$S_f = f_a(P_1) - f(P_0) = f_a(P_1) - f(P_1) + f(P_1) - f(P_0)$$

che possiamo dividere in:

$$S_f = S_a + S_d, \quad S_a = f_a(P_1) - f(P_1), \quad S_d = f(P_1) - f(P_0)$$

che chiamiamo rispettivamente **errore totale algoritmico** e **errore totale inerente**.

Allo stesso modo possiamo definire l'**errore relativo**:

$$\begin{aligned} \epsilon_f &= \frac{S_f}{f(P_0)} = \frac{f_a(P_1) - f(P_0)}{f(P_0)} = \frac{f_a(P_1) - f(P_1)}{f(P_0)} + \frac{f(P_1) - f(P_0)}{f(P_0)} \\ &= \frac{f_a(P_1) - f(P_1)}{f(P_1)} \cdot \frac{f(P_1)}{f(P_0)} + \frac{f(P_1) - f(P_0)}{f(P_0)} \end{aligned}$$

che si divide nuovamente in :

$$\epsilon_f = \epsilon_a + \epsilon_d, \quad \epsilon_a = \frac{f_a(P_1) - f(P_1)}{f(P_1)}, \quad \epsilon_d = \frac{f(P_1) - f(P_0)}{f(P_0)}$$

che chiamiamo rispettivamente **errore relativo algoritmico** e **errore relativo inerente**. Questo viene da:

$$\epsilon_f = [\dots] = \frac{f_a(P_1) - f(P_1)}{f(P_1)} \cdot \frac{f(P_1)}{f(P_0)} + \frac{f(P_1) - f(P_0)}{f(P_0)}$$

si nota che $\frac{f(P_1)}{f(P_0)} = 1 + \frac{f(P_1) - f(P_0)}{f(P_0)}$, e quindi:

$$= \epsilon_a \cdot \left(1 + \frac{f(P_1) - f(P_0)}{f(P_0)}\right) + \epsilon_d = \epsilon_a(1 + \epsilon_d) + \epsilon_d = \epsilon_a + \epsilon_d + \epsilon_a \epsilon_d \approx \epsilon_a + \epsilon_d$$

assumendo $\epsilon_a \epsilon_d \approx 0$.

In generale, per limitare $|S_f|$ cercheremo disuguaglianze $|S_a| < \tau_1$, $|S_d| < \tau_2$, da cui:

$$|S_f| < \tau_1 + \tau_2$$

1.1.3 Errore inerente

Avevamo quindi definito l'errore totale inerente:

$$S_d = f(P_1) - f(P_0)$$

Sotto l'ipotesi $f \in C^1(D)$ per $D \subset \mathbb{R}^m$ che contiene P_0 , si può usare lo sviluppo di Taylor di f in P_0 , troncato al primo ordine:

$$\begin{aligned} f(P_1) - f(P_0) &= f(P_0) + \nabla f(\bar{P})^T (P_1 - P_0) - f(P_0) = \nabla f(\bar{P})^T (P_1 - P_0) \\ &= \sum_{j=1}^m \frac{\partial f}{\partial x_j}(\bar{P}) \cdot (x_j^{(1)} - x_j^{(0)}) \approx \sum_{j=1}^m \frac{\partial f}{\partial x_j} P_0 \cdot (x_j^{(1)} - x_j^{(0)}) \end{aligned}$$

dove \bar{P} è un punto che sta sul segmento $\overline{P_1 P_0}$. Da questo potremo dire:

$$S_d = \sum_{j=1}^m \frac{\partial f}{\partial x_j} P_0 \cdot S_j$$

dove $S_j = (x_j^{(1)} - x_j^{(0)})$ è l'**errore di arrotondamento** nella componente j di P_0 , e $\frac{\partial f}{\partial x_j} P_0$ viene detto **coefficiente di amplificazione**.

Per l'errore relativo inerente potremo fare considerazioni simili: metti qual'è come sopra

$$\epsilon_d = \frac{\sum_{j=1}^m \frac{\partial f}{\partial x_j}(P_0) \cdot S_j}{f(P_0)} = \sum_{j=1}^m \frac{x_j^{(1)} - x_j^{(0)}}{x_j^{(0)}} \cdot \frac{\partial f}{\partial x_j}(P_0) \cdot \frac{x_j^{(0)}}{f(P_0)}$$

dove $\epsilon_j = \frac{x_j^{(1)} - x_j^{(0)}}{x_j^{(0)}}$ sarà l'**errore di arrotondamento relativo** nella componente j di P_0 e

$P_j = \frac{\partial f}{\partial x_j}(P_0) \cdot \frac{x_j^{(0)}}{f(P_0)}$ viene detto **coefficiente di amplificazione dell'errore relativo**.

La formula finale sarà quindi:

$$\epsilon_d = \sum_{j=1}^m \epsilon_j P_j$$

I problemi in cui si devono calcolare f i cui coefficient di amplificazione degli errori relativi sono grandi in modulo (o ce n'è almeno uno sufficientemente grande) si dicono **malcondizionati**. Viceversa, se $|P_j|$ è vicino a 1 per ogni j il problema si dice **ben condizionato**, cioè che ϵ_d è di un ordine di grandezza comparabile a $\max(\epsilon_i)$

Notiamo che il condizionamento di un problema dipende solamente dalla sua struttura matematica.

1.1.4 Errori inerenti delle operazioni aritmetiche

Vediamo gli errori inerenti associati alle 4 operazioni aritmetiche $+$, $-$, \times , \div :

Operazione	S_d	ϵ_d
$x \oplus y$	$S_x + S_y$	$\frac{x}{x+y} \epsilon_x + \frac{y}{x+y} \epsilon_y$
$x \ominus y$	$S_x - S_y$	$\frac{x}{x+y} \epsilon_x - \frac{y}{x+y} \epsilon_y$
$x \otimes y$	$y S_x + x S_y$	$\epsilon_x + \epsilon_y$
$x \oslash y$	$\frac{1}{y} S_x - \frac{x}{y^2} S_y$	$\epsilon_x - \epsilon_y$

Notiamo come somme e sottrazioni non amplificano gli errori totali, mentre prodotti e divisioni non amplificano gli errori relativi (riguardo agli errori inerenti). Questo significa che somme e sottrazioni possono avere errori relativi grandi quando $|x + y| \ll \min\{|x|, |y|\}$. Questo effetto viene detto **cancellazione numerica**.

1.1.5 Errore algoritmico

hai fatto un casino boia fra assoluto e totale Avevamo definito un algoritmo $f_a(x)$ di cui vogliamo stimare l'errore algoritmico assoluto $S_a = f_a(P_1) - f(P_1)$. Assumiamo $P_1 = Rn(P_0) \in F(\beta, m, L, U)$, cioè gli operandi come privi di errori iniziali di arrotondamento.

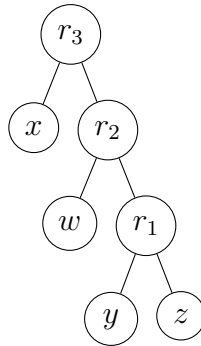
L'idea è di seguire l'errore generato dall'algoritmo sul grafo (o albero) che lo rappresenta sgruttando le relazioni per l'errore inerente nelle 4 operazioni aritmetiche. Prendiamo ad esempio la funzione:

$$f(x, y, z, w) = x \cdot \left(\frac{y}{z} - w \right)$$

Avremo i risultati intermedi:

- $r_1 = \frac{y}{z}$
- $r_2 = r_1 - w$
- $r_3 = r_2 \cdot x$

Di cui riportiamo il grafo:



aggiungi errori formule fallo bellino dove ϵ_3, ϵ_2 e ϵ_1 rappresentano gli errori di troncamento dei risultati intermedi e $\epsilon_{r3}, \epsilon_{r2}$ e ϵ_{r1} rappresentano gli errori inerenti delle singole operazioni per il calcolo dei risultati intermedi.

Partiamo dalla radice per valutare gli errori:

$$\begin{aligned}
 \epsilon_{r3} &= \epsilon_3 + \epsilon_x + \epsilon_{r2} = \epsilon_3 + \epsilon_{r2} \\
 &= \epsilon_3 + \epsilon_2 + \left(\frac{-zw}{y - zw} \right) \cdot \epsilon_w + \frac{y}{y - zw} \cdot \epsilon_{r1} = \epsilon_3 + \epsilon_2 + \frac{y}{y - zw} \cdot \epsilon_{r1} \\
 &= \epsilon_3 + \epsilon_2 + \frac{y}{y - zw} (\epsilon_1 + \epsilon_y - \epsilon_z) = \epsilon_3 + \epsilon_2 + \frac{y}{y - zw} \cdot \epsilon_1 = \epsilon_a
 \end{aligned}$$

Dove per la stima di ϵ_3, ϵ_2 e ϵ_1 , vale $\epsilon_i \leq U$ precisione macchina. Nel caso di errori assoluti vale $|S_i| \leq U \cdot \max(x_i)$ considerata ogni variabile x_i .

Chiaramente, diversi algoritmi equivalenti in aritmetica esatta potranno avere errori algoritmici diversi fatte tutte le approssimazioni.

riporta esempio $x^2 - y^2$, *duemodiuno* x, y *dipendentel'altro* $|\epsilon_a| < 3U$ *fisso*

Abbiamo visto quindi tenciche per la stima di ϵ_a e ϵ_d (S_a e S_d), che ci permettono di calcolare $|\epsilon_f| \leq |\epsilon_a| + |\epsilon_d|$ ($|S_f| \leq |S_a| + |S_d|$).

Un problema classico sarà quello di, data f , un algoritmo risolutivo f_a e una stima degli errori d_{x_i} , di stimare S_f per $P_0 \in D \subset \mathbb{R}^m$.

Il problema inverso potrebbe essere quello di, dato $\tau > 0$, f e un punto $P_0 \in \mathbb{R}^n$, determinatre un algoritmo ed un valore di precisione macchina U tali per cui $|s_f| < \tau$.