Laboratorio computazionale

Appunti lezione 2

Stefano Franceschina

12/03/2025

Introduzione

Come abbiamo visto, le limitazioni della rappresentazione in punto mobile si ripercuotono sulle operazioni fondamentali eseguite dal computer. In particolare, quando si esegue un'operazione aritmetica in macchina essa è soggetta a errori dovuti all'arrotondamento. Di seguito vengono illustrate alcune definizioni e concetti relativi alla rappresentazione e all'analisi degli errori.

Operazioni Aritmetiche Approssimate

Si definisce la somma in macchina come:

$$x \oplus y = \mathrm{fl}(x+y) = (x+y)\,(1+\varepsilon_1),$$

dove $\mathrm{fl}(x+y)$ indica il risultato dell'operazione arrotondato secondo la precisione della macchina e $\epsilon_1 = \frac{f'-f}{1+f}$ rappresenta l'approssimazione della somma. Analogamente, per le altre operazioni si ha:

$$x \ominus y = \mathrm{fl}(x-y) = (x-y)\,(1+\varepsilon_2),$$

$$x \otimes y = \mathrm{fl}(x \cdot y) = (x \cdot y)\,(1+\varepsilon_3).$$

È evidente che queste operazioni sono approssimate e che l'insieme $\mathbb{F}$ dei numeri in virgola mobile non è chiuso rispetto a tali operazioni.

La Proprietà Associativa e l'Effetto della Cancellazione

Uno dei problemi principali in aritmetica in punto mobile è la violazione della proprietà associativa dell'addizione. Consideriamo il seguente esempio, sapendo che $e = \frac{\epsilon_{mach}}{2}$

1. (1.0 + e) - 1.0
2. 1.0 + (e - 1.0)

Ci si aspetta che entrambe le operazioni restituiscano lo stesso risultato, pari ad e in particolare. Tuttavia, il risultato ottenuto è diverso nei due casi, a causa del fatto che ϵ_{mach} è la precisione *relativa.* ϵ_{mach} è la distanza minima tra 1 e il floating point number successivo, e quindi la somma di $1.0+e$, dove $e = \frac{\epsilon_{mach}}{2}$, è approssimata a 1.0. Sottraendo 1.0 si ottiene un risultato nullo. Nell'altro caso, invece, sapendo che la distanza tra i numeri $[\frac{1}{2}, 1)$ non è più ϵ_{mach} ma $\frac{\epsilon_{mach}}{2}$, la somma di $e - 1.0$ è approssimata a $-\frac{\epsilon_{mach}}{2}$, che sommato a 1.0 dà e. Questo esempio mostra come la cancellazione possa portare a risultati errati.

Accuratezza e Precisione

L'**accuratezza** indica quanto il valore calcolato si avvicini al valore reale, mentre la **precisione** si riferisce al numero di cifre significative della rappresentazione in macchina. Non è detto che un numero rappresentato con molte cifre (alta precisione) sia anche accurato, poiché un algoritmo mal progettato può produrre risultati con molte cifre ma errati. L'accuratezza relativa può essere espressa anche in scala logaritmica per evidenziare le differenze in termini di ordine di grandezza.
Esaminiamo meglio il concetto di accuratezza: intorduciamo il numero x e la sua approssimazione $\tilde{x}$. L'**accuratezza assoluta** di $\tilde{x}$ è definita come $|x-\tilde{x}|$, mentre l'**accuratezza relativa** è data da $\frac{|x-\tilde{x}|}{|x|}$.
Si può esprimere l'accuratezza relativa in scala logaritmica come $d = -\log_{10}\left|\frac{x-\tilde{x}}{x}\right|$.
Sulle dispense sono riportati alcuni esempi di accuratezza relativa e di scala logaritmica.

Algoritmi e Propagazione degli Errori

Abbiamo visto che due risoluzioni analoghe dal punto di vista aritmetico di uno stesso problema possono portare a risultati diversi una volta risolti in aritmetica in virgola mobile.

Un algoritmo numerico può essere descritto come una funzione

$$\tilde{f} : \mathbb{F}^n \to \mathbb{F}^m,$$

cioè una funzione che prende un numero macchina e ne restituisce un altro. In pratica, un algoritmo è una composizione di funzioni:

$$\tilde{x}_0 \xrightarrow{f_0} \tilde{y}_0 = \tilde{x}_1 \xrightarrow{f_1} \tilde{y}_1 = \tilde{x}_2 \xrightarrow{f_2} \cdots \xrightarrow{f_{N-1}} \tilde{x}_N = \tilde{y},$$

dove $\tilde{y}$ è il risultato finale approssimato.

Per snellire la notazione consideriamo la funzione f come un solo passaggio dell'algoritmo, come se fosse una f_i. Possiamo anche pensare come se l'algoritmo fosse composto da un solo passaggio.
Siamo interessati all'errore, che possiamo manipolare come:

$$\Delta y = \tilde{y} - y = \tilde{f}(\tilde{x}) - f(x) = [\tilde{f}(\tilde{x}) - f(\tilde{x})] + [f(\tilde{x}) - f(x)] = \Delta y_{\text{fl}} + \Delta y_{\text{f}},$$

In questa forma riconosciamo due contributi all'errore:

- L'arrotondamento del risultato dell'algoritmo: Δy_{fl}.
- L'approssimazione iniziale del valore di x: Δy_{f}.

Analizziamo per prima cosa l'errore Δy_{f}. Riscriviamo Δy in funzione di f e sviluppiamo in serie di Taylor per analizzare l'effetto degli errori, questo è possibile in quanto $\tilde{x} = x(1 + \epsilon_x)$ con ϵ_x piccolo.

$$\Delta y_{\text{f}} = f(\tilde{x}) - f(x) = f'(x)\,\Delta x + O(\varepsilon_x^2), \quad \text{con } \Delta x = \tilde{x} - x.$$

In questa maniera, ricordandoci che le precisioni relative di x e y sono $\varepsilon_x = \frac{|\Delta x|}{|x|}$ e $\varepsilon_y = \frac{|\Delta y|}{|y|} = \frac{f(\tilde{x}) - f(x)}{f(x)}$, e definendo il numero di condizionamento K_f della funzione f come $K_f = \left|\frac{x}{f(x)}\, f'(x)\right|$ possiamo scrivere l'errore relativo di y troncato al secondo ordine come:

$$|\varepsilon_y| \approx K_f\, |\varepsilon_x|.$$

É facile verificare che nel caso in cui la funzione f sia composizione di funzioni, il numero di condizionamento della funzione composta è il prodotto dei numeri di condizionamento delle funzioni che la compongono.
Adesso analizziamo l'errore Δy_{fl}. Ci aspettiamo per il valore $f(\tilde{x})$ un errore di arrotondamento pari a $f(\tilde{x})(1+\epsilon_y)$. Ripercorrendo i passaggi precendenti, e cioè sviluppando in serie di Taylor, possiamo scrivere l'errore relativo Δy_{fl} come:

$$\Delta y_{\text{fl}} = \epsilon_y f(x) = \epsilon_y y$$

In questa maniera possiamo combianre i due errori per ottenere l'errore totale:

$$\Delta y = (\epsilon_y + K_f \epsilon_x) y$$

Ulteriori Considerazioni

È utile ricordare che il comportamento degli errori in aritmetica in virgola mobile è regolato dallo standard IEEE 754, che definisce il formato di rappresentazione, le modalità di arrotondamento e le eccezioni. La comprensione del numero di condizionamento e della propagazione degli errori è fondamentale per progettare algoritmi numerici stabili e affidabili. Algoritmi ben condizionati riescono a limitare la crescita degli errori, mentre algoritmi mal condizionati possono amplificare anche piccole imprecisioni.

1 Recap del giorno dopo

Abbiamo visto che un algoritmo è la composizione di certe funzioni, e l'algoritmo è l'approssimazione di un problema matematico esatto. Sta a noi capire l'errore, l'accuratezza assoluta traa la soluzione numerica e la soluzione esatta. Abbiamo visto che l'errore è composto da due parti: l'errore dovuto all'input e l'errore dovuto all'output.