Analisi Numerica

Giovanni Norbedo

2024-2025

Indice

Introduzione . . . 2
- Tipi di errori . . . 2
- Sistema posizionale . . . 2
- Perché la serie della parte frazionaria converge? . . . 3
- Cambio di base da base 2 a base 10 . . . 3
- Cambio di base da base 10 a base 2 . . . 3
  - Parte intera . . . 3
  - Parte decimale . . . 3
- Rappresentazione in virgola mobile normalizzata . . . 4
- Numeri macchina . . . 4
- Numeri nel calcolatore . . . 5
  - Singola precisione . . . 5
  - Doppia precisione . . . 5
- Half precision . . . 5

Approssimazione di un numero reale . . . 6
- Underflow e overflow . . . 6

Errori assoluti e relativi . . . 6
- Maggiorazione dell'errore assoluto . . . 6
  - Errore assoluto per troncamento . . . 6
  - Errore assoluto per arrotondamento . . . 6
- Maggiorazione dell'errore relativo . . . 7
  - Errore relativo per troncamento . . . 7
  - Errore relativo per arrotondamento . . . 7

Precisione di macchina . . . 7

Standard ANSI IEEE-754r . . . 7

Massimo e minimo numero rappresentabile . . . 8
- Massimo numero rappresentabile . . . 8
- Minimo numero rappresentabile . . . 8

Distanza assoluta tra due numeri macchina consecutivi . . . 8

Distanza relativa tra due numeri macchina consecutivi . . . 8

Precisione di macchina (2) . . . 9

Errori nelle Operazioni Macchina . . . 9
- Errore relativo risultante da operazioni macchina . . . 9
- Propagazione degli Errori . . . 9
- Errori nell'addizione dimostrazione . . . 9
- Errori nel prodotto dimostrazione . . . 10
- Osservazioni . . . 10

Cancellazione Numerica . . . 10

Esempio 1 . . . 10
Esempio 2 . . . 11
Proprietà delle Operazioni: Non Associatività . . . 11
Esempio di Non Associatività . . . 11
Esempio con Overflow/Underflow . . . 11
Cancellazione numerica e stabilità di un algoritmo . . . 11
Esempii di Algoritmo Instabile . . . 12
Esempio 1 . . . 12
Formula risolutiva delle equazioni di secondo grado . . . 12
Successione Ricorrente . . . 12
Instabilità della Formula Ricorsiva . . . 13
Analisi dell'Errore . . . 13
Formula Alternativa Stabile . . . 13
Smorzamento dell'Errore . . . 13
Condizionamento di un Problema . . . 14
Definizione . . . 14
Esempio . . . 14
Numero di Condizionamento . . . 14
Esempio . . . 15

Calcolo degli Zeri di una Funzione **15**
Introduzione . . . 15
Esistenza dello zero - Teorema di Bolzano . . . 15
Unicità dello zero . . . 16
Metodi per il Calcolo degli Zeri . . . 16

Introduzione

Tipi di errori

- **Errori di modellazione matematica** del problema reale ed **errori presenti nei dati** sperimentali.
- **Errori di troncamento**: da problema matematico (dimensione infinita) a problema numerico (dimensione finita).
- **Errori di arrotondamento**: sul calcolatore, posso rappresentare solo un sottoinsieme finito dei numeri reali.

Sistema posizionale

Definizione: Fissata la *base* $B \in \mathbb{N}$, $B > 1$, e un numero $x \in \mathbb{R}$ finito di cifre $d_k, k = -m, -m+1, \dots, n-1, n$, di definisce x_B la **rappresentazione posizionale** di x in base B:

$$x_B = (-1)^s \cdot \sum_{k=-m}^{n} d_k \cdot B^k \quad d_n \neq 0$$

Esempi:
$(867.0985)_{10} = (-1)^0 \left(8 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0 + 0 \cdot 10^{-1} + 9 \cdot 10^{-2} + 8 \cdot 10^{-3} + 5 \cdot 10^{-4}\right)$
$(-10110.0001)_2 = (-1)^1 \left(1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-4}\right)$

Osservazione:
$\forall x \in \mathbb{R}$, fissata B,

$$x_B = (-1)^s \cdot \sum_{k=0}^{n} d_k \cdot B^k + \sum_{k=1}^{\infty} d_{-k} \cdot B^{-k}$$

Perché la serie della parte frazionaria converge?

Dimostrazione:
Considero la serie geometrica di ragione B^{-1} (criterio di confronto tra serie a termini non negativi).
Poiché $d_{-k} \leq B-1, \quad \forall k \in \mathbb{N}$, allora

$$\sum_{k=1}^{\infty}(B-1)\cdot B^{-k} = (B-1)\cdot \sum_{k=1}^{\infty} B^{-k} = (B-1)\cdot \frac{B^{-1}}{1-B^{-1}} \text{ convergente.}$$

Nota: $(0.999\ldots)_{10} = (0.111\ldots)_2 = 1$
Un numero razionale può avere rappresentazione data da un numero finito di cifre in una base e infinito in un'altra:
$\frac{1}{3} = 0.333\ldots = 0.1_3$

Cambio di base da base 2 a base 10

Definizione: Dato un numero x in base 2, si può convertire in base 10 tramite la formula

$$x_{10} = \sum_{k=0}^{n} d_k \cdot 2^k + \sum_{k=1}^{\infty} d_{-k} \cdot 2^{-k}$$

Esempio: $(10001000.01)_2 = 2^7 + 2^3 + 2^{-2} = 128 + 8 + 0.25 = 136.25$

Cambio di base da base 10 a base 2

Parte intera **Definizione**: Dato un numero x in base 10, si può convertire in base 2 tramite le seguenti operazioni:

1. Divido la parte intera di x per 2 e scrivo il resto.
2. Divido il quoziente precedente per 2 e scrivo il resto.
3. Continuo fino a che il quoziente è 0.
4. Leggo i resti in ordine inverso.

Parte decimale **Definizione**: Dato un numero x in base 10, si può convertire in base 2 tramite le seguenti operazioni:

1. Moltiplico la parte decimale di x per 2 e scrivo la parte intera.
2. Continuo fino a che la parte decimale è 0 oppure se si ripetono periodicamente le stesse cifre.
3. Leggo le parti intere in ordine.

Esempio: 389.1_{10}

Parte intera:

389	1
194	0
97	1
48	0
24	0
12	0
6	0
3	1
1	1
0	1

$\Rightarrow (110000101)_2$

Parte decimale:

0.2	0
0.4	0
0.8	0
1.6	1
1.2	1
0.4	0
0.8	0
1.6	1
1.2	1

$\Rightarrow (00011001100110011\ldots)_2$

$\Rightarrow 389.1_{10} = (110000101.0\overline{0011})_2$

Rappresentazione in virgola mobile normalizzata

Definizione: Dato un numero $x \in \mathbb{R}$, si definisce la **rappresentazione in virgola mobile normalizzata** come

$$x = (-1)^s \cdot B^e \cdot \sum_{k=1}^{\infty} d_k \cdot B^{-k} \quad \text{con} \begin{cases} d_1 > 0 \\ 0 \leq d_k < B-1 \\ e \in \mathbb{Z} \end{cases}$$

oppure si può scrivere come

$x = \pm p \cdot B^e \quad \text{con } B^{-1} \leq p < 1$

dove p è detto **mantissa** e e è detto **esponente**.

Esempi:
In base 10:
$x = 0.00745 \Rightarrow x = 0.745 \cdot 10^{-2}$
$x = 70408.102 \Rightarrow x = 0.70408102 \cdot 10^5$

In base 2: $x = 11001.111 \Rightarrow x = 0.11001111 \cdot 2^5$

Numeri macchina

Nel calcolatore i numeri reali sono rappresentati in virgola mobile normalizzata, con t cifre di mantissa e e esponente, con $L \leq e \leq U$.

Fissata una base B (di solito $B = 2$), fissati $t, L < 0, U > 0$, si definisce l'insieme dei numeri macchina $\mathbb{F}(B, t, L, U)$ come

$\mathbb{F}(B,t,L,U) = \{x | x = (-1)^s \cdot B^e \cdot \sum_{k=1}^{t} d_k \cdot B^{-k}\} \cup \{0\}$, con $d_1 > 0, 0 \leq d_k < B-1, e \in [L, U]$

Lo **zero** è rappresentato come $0 = 0 \cdot B^0 \cdot 0$.

Un numero $x \in \mathbb{F}(B, t, L, U)$ è scritto come

$x = (-1)^s \cdot (0.d_1 d_2 d_3 \dots d_t)_B \cdot B^e$

Numeri nel calcolatore

Base 2: $B = 2$, cifre $d_k \in \{0, 1\}$

$x = (-1)^s \cdot (0.d_1 d_2 d_3 \dots d_t)_2 \cdot 2^e$

- 1 bit per il segno
- t bit per la mantissa
- l'esponent

Singola precisione 32 bit:

- 1 bit per il segno
- 8 bit per l'esponente
- 23 bit per la mantissa (1 bit nascosto)

$F(2, 24, -126, 127)$

Dei $2^8 = 256$ esponenti, 2 sono riservati per i numeri speciali (infinito e NaN), quindi rimangono $2^8 - 2 = 254$ esponenti.

I numeri rappresentabili sono: $2 \cdot (U - L + 1) \cdot (B - 1) \cdot B^{t-1} + 1 = 2 \cdot 254 \cdot 2^{23} + 1 \approx 4.3 \cdot 10^9$

Doppia precisione 64 bit:

- 1 bit per il segno
- 11 bit per l'esponente
- 52 bit per la mantissa (1 bit nascosto)

$F(2, 53, -1022, 1023)$

Dei $2^{11} = 2048$ esponenti, 2 sono riservati per i numeri speciali (infinito e NaN), quindi rimangono $2^{11} - 2 = 2046$ esponenti.

I numeri rappresentabili (cardinalità) sono: $2 \cdot (U - L + 1) \cdot (B - 1) \cdot B^{t-1} + 1 = 2 \cdot 2046 \cdot 2^{52} + 1 \approx 1.8 \cdot 10^{19}$

Nota: $(B - 1)$ è il numero massimo di cifre rappresentabili in base B. B^{t-1} è il numero di cifre rappresentabili nella mantissa.

Half precision

16 bit:

- 1 bit per il segno
- 5 bit per l'esponente
- 10 bit per la mantissa (1 bit nascosto)

$F(2, 11, -14, 15)$

$2^5 = 32$ esponenti, 2 sono riservati per i numeri speciali (infinito e NaN), quindi rimangono $2^5 - 2 = 30$ esponenti.

I numeri rappresentabili sono: $2 \cdot (U - L + 1) \cdot (B - 1) \cdot B^{t-1} + 1 = 2 \cdot 30 \cdot 2^9 + 1 \approx 3.1 \cdot 10^4$

Approssimazione di un numero reale

In $\mathbb{F}(B,t,L,U)$, dato un numero reale $x = p \cdot B^e \in \mathbb{R}$, se ha più di t cifre nella mantissa, si approssima con il numero macchina $fl(x) \in \mathbb{F}(B,t,L,U)$ in due modi:

- **Troncamento**: nella mantissa p si cancellano le cifre oltre la t-esima.
- **Arrotondamento**: nella mantissa p si aggiunge $\frac{B}{2} \cdot B^{-(t+1)}$ e poi si tronca a t cifre.

Esempio: Considero $x = 0.745645897, t = 6$, per troncamento ho che $fl(x) = tr(x) = 0.745645$ e per arrotondamento $fl(x) = tr(x + 0.0000005) = tr(0.745646397) = 0.745646$.

Osservazione: Arrontondare equivale a sommare 1 alla t-esima cifra della mantissa, d_t, se la successiva cifra, d_{t+1}, è $\geq \frac{B}{2}$, altrimenti la cifra t-esima rimane invariata.

Underflow e overflow

In $\mathbb{F}(B,t,L,U)$, nell'approssimare x con $fl(x)$, si possono verificare due situazioni:

- se l'esponente $e > U$, si ha **overflow**: $fl(x) = \infty$
- se l'esponente $e < L$, si ha **underflow**: $fl(x) = 0$

Errori assoluti e relativi

Maggiorazione dell'errore assoluto

Sia $x \in \mathbb{R}$ e x^* la sua approssimazione in $\mathbb{F}(B,t,L,U)$, si definiscono:

- **Errore assoluto**: $|x - x^*|$
- **Errore relativo**: $\dfrac{|x - x^*|}{|x|}, \quad x \neq 0$

Nel caso si abbia a che fare con enti diversi da numeri real (funzioni, vettori, matrici) le definizioni sono le stesse a patto di sostituire il valore assoluto con un'opportuna norma.

Errore assoluto per troncamento $|x - fl(x)| = |p \cdot B^e - \bar{p} \cdot B^e| = |(p - \bar{p}) \cdot B^e| \leq B^{-t} B^e$

$p = 0.d_1 d_2 d_3 \dots d_t | d_{t+1} d_{t+2} \dots$
$\bar{p} = 0.d_1 d_2 d_3 \dots d_t$
$|p - \bar{p}| = 0.00 \dots | d_{t+1} d_{t+2} \dots$

$$|p - \bar{p}| = \sum_{k=t+1}^{\infty} d_k \cdot B^{-k} \leq (B-1) \cdot \sum_{k=t+1}^{\infty} B^{-k} = (B-1) \cdot \frac{B}{} \dots$$

Esempio: $x = 0.745645897, fl(x) = 0.745645, t = 6$
$|x - fl(x)| = |0.745645897 - 0.745645| = 0.000000897 \leq 10^{-6}$

Errore assoluto per arrotondamento $|x - fl(x)| = |p \cdot B^e - \bar{p} \cdot B^e| \leq \frac{B}{2} \cdot B^{-(t+1)} B^e$

Fra due numeri macchina consecutivi c'è una distanza di $\frac{B}{2} \cdot B^{-(t+1)}$.

Caso A: $d_{t+1} \geq \frac{B}{2}$

$$|p - \bar{p}| = \frac{B}{2} \cdot B^{-(t+1)}$$

Caso B: $d_{t+1} < \frac{B}{2}$

$$|p - \bar{p}| = \left(\frac{B}{2} - 1\right) \cdot B^{-(t+1)} + \sum_{k=t+2}^{\infty} d_k \cdot B^{-k} \leq \left(\frac{B}{2} - 1\right) \cdot B^{-(t+1)} + (B-1) \cdot \sum_{k=t+2}^{\infty} B^{-k} = \dots$$

Esempio: $x = 0.745645897, fl(x) = 0.745646, t = 6$
$|x - fl(x)| = |0.745645897 - 0.745646| = 0.000000103 \leq 5 \cdot 10^{-7}$

Maggiorazione dell'errore relativo

Errore relativo per troncamento $\dfrac{|x - fl(x)|}{|x|} \leq \dfrac{|p - \bar{p}| \cdot B^e}{p \cdot B^e} = \dfrac{|p - \bar{p}|}{p} \leq \dfrac{B^{-t}}{B^{-1}} = B^{1-t}$

Errore relativo per arrotondamento $\dfrac{|x - fl(x)|}{|x|} \leq \dfrac{|p - \bar{p}| \cdot B^e}{p \cdot B^e} = \dfrac{|p - \bar{p}|}{p} \leq \dfrac{\frac{B}{2} \cdot B^{-(t+1)}}{B^{-1}} = \dfrac{B}{2} \cdot B^{-t} = \dfrac{1}{2} \cdot B^{1-t}$

Attualmente, la maggior parte dei sistemi implementa la tecnica di arrotondamento perché produce mediamente errori più piccoli.

Precisione di macchina

Definizione: Si definisce **precisione di macchina** (unit roundoff) il più piccolo numero positivo rappresentabile in $\mathbb{F}(B, t, L, U)$, indicato con $u = \frac{1}{2} \cdot B^{1-t}$.

La precisione di macchina rappresenta il massimo errore relativo che si commette nell'approssimare il numero reale x con il suo corrispondente numero macchina $fl(x)$ per arrotondamento.

Esempi:

- $F(2, 24, -126, 127)$, $u = \frac{1}{2} \cdot 2^{1-24} = 2^{-24} \approx 6.0 \cdot 10^{-8}$
- $F(2, 53, -1022, 1023)$, $u = \frac{1}{2} \cdot 2^{1-53} = 2^{-53} \approx 1.1 \cdot 10^{-16}$
- $F(2, 11, -14, 15)$, $u = \frac{1}{2} \cdot 2^{1-11} = 2^{-11} \approx 4.9 \cdot 10^{-4}$

Standard ANSI IEEE-754r

Scritto nel 1985 e modificato nel 1989 e, più recentemente, nel 2008
costituisce lo standard ufficiale per la rappresentazione binaria dei numeri all'interno del calcolatore e l'aritmetica di macchina (il nome dello standard in inglese "Binary floating point arithmetic for microprocessor systems").

Secondo lo standard un numero non nullo normalizzato si scrive come

$x = (-1)^s \cdot (1 + f) \cdot 2^{e^* - \text{bias}}$

La mantissa si rappresenta dunque come $1.d_1 d_2 \dots d_\tau$ essendo

$f = 0.d_1 d_2 \dots d_\tau$

τ identifica il numero di bit usato per codificare la parte frazionaria della mantissa. Il numero di cifre totali per la mantissa è $t = \tau + 1$.

Il vero esponente del numero e si immagazzina in traslazione come
$e^* = e + \text{bias}$.

In questa maniera non serve un bit di segno per l'esponente.

Il bias in singola precisione vale 127 mentre in doppia 1023.

Lo standard riserva due dei possibili valori per l'esponente per codificare due situazioni speciali:

- $e^* = 0$, viene riservato per la codifica dello zero ed eventuali numeri denormalizzati.
- $e^* = 255$ (singola precisione) o $e^* = 2047$ (doppia precisione), che corrisponde a un esponente vero pari a 128 (singola) o 1024 (doppia), viene riservato per la codifica di:
  - **Inf (Overflow)**
  - **NaN (Not a Number)**, ovvero operazioni del tipo $\frac{0}{0}, \quad \infty - \infty, \quad \frac{\infty}{\infty}$

Inf viene codificato con mantissa nulla, mentre NaN con mantissa $\neq 0$.

Massimo e minimo numero rappresentabile

F è limitato inferiormente e superiormente.

Massimo numero rappresentabile

Il massimo numero rappresentabile è il numero più grande che si può rappresentare in $\mathbb{F}(B, t, L, U)$, ovvero con tutte le cifre della mantissa uguali a $B - 1$ e l'esponente massimo:

$M = (1 - B^{-t}) \cdot B^U$

Minimo numero rappresentabile

Il minimo numero rappresentabile è il numero più piccolo che si può rappresentare in $\mathbb{F}(B, t, L, U)$, ovvero con le cifre della mantissa tutte uguali a 0 tranne la prima, e l'esponente minimo:

$m = B^L$???

Distanza assoluta tra due numeri macchina consecutivi

$x = (-1)^s \cdot (1 + 0.d_1 d_2 \dots d_\tau) \cdot B^e$
$x_+ = (-1)^s \cdot (1 + 0.d_1 d_2 \dots d_\tau + 1) \cdot B^e$

$\Delta x = |x - x_+| = B^{-\tau} \cdot B^e = B^{e-\tau}$

Questa distanza è uguale per tutti i numeri macchina aventi lo stesso esponente.
L'incremento/decremento di una unità dell'esponente comporta un incremento/decremento di in fattore pari alla base della distanza assoluta tra due numeri macchina consecutivi.

Distanza relativa tra due numeri macchina consecutivi

$$\frac{|x - x_+|}{|x|} = \frac{B^{e-\tau}}{p \cdot B^e} = \frac{B^{-\tau}}{p}$$

Si può vedere che la distanza relativa tra due numeri macchina consecutivi ha un andamento periodico.

La massima distanza relativa tra due numeri macchina consecutivi è:

$$\varepsilon_M = B^{-\tau}$$

che si ha quando $p = 1$.

Nello standard IEEE-754r in doppia precisione $\varepsilon_M = 2^{-52}$.

Precisione di macchina (2)

La precisione di macchina definita precedentemente come il massimo errore relativo di arrotondamento, coincide anche con $\boldsymbol{u} = \frac{1}{2}\varepsilon_M$.

In un computer che usa doppia precisione secondo lo standard IEEE-754r il valore della precisione di macchina è pari a $u = 2^{-53}$.

Errori nelle Operazioni Macchina

Ricordiamo che $\varepsilon_x = \dfrac{|x - fl(x)|}{|x|} \leq u$.

Errore relativo risultante da operazioni macchina

$x \oplus y$ per definizione è l'approssimazione di $x + y$ in $\mathbb{F}(B, t, L, U)$.

$\varepsilon^{\oplus}_{x,y} = \dfrac{|(x + y) - (x \oplus y)|}{|x + y|}$

Propagazione degli Errori

- **Somma:**

$$\epsilon_{\oplus}(x, y) \leq \left|\frac{x}{x+y}\right| \epsilon_x + \left|\frac{y}{x+y}\right| \epsilon_y$$

- **Prodotto:**

$$\epsilon_{\otimes}(x, y) \leq\approx \epsilon_x + \epsilon_y$$

- **Divisione (o altre operazioni):**

$$\epsilon_{\oslash} x, y \leq |\epsilon_x - \epsilon_y|$$

con ϵ_x e ϵ_y tali che $\epsilon_x, \epsilon_y \leq u$.

Errori nell'addizione dimostrazione

$x \neq 0, y \neq 0, x + y \neq 0, fl(fl(x) + fl(y)) = fl(x) + fl(y)$, cioè la somma di due numeri macchina è un numero macchina.

$$\begin{aligned}
\epsilon_{\oplus}(x, y) &= \frac{|(x + y) - (x \oplus y)|}{|x + y|} = \frac{|(x + y) - (fl(x) + fl(y))|}{|x + y|} \\
&\leq \frac{|(x - fl(x))|}{|x + y|} + \frac{|(y - fl(y))|}{|x + y|} \\
&= \frac{|(x - fl(x))| \cdot |x|}{|x + y| \cdot |x|} + \frac{|(y - fl(y))| \cdot |y|}{|x + y| \cdot |y|} \\
&= \frac{|(x - fl(x))|}{|x|} \cdot \frac{|x|}{|x + y|} + \frac{|(y - fl(y))|}{|y|} \cdot \frac{|y|}{|x + y|} \\
&\leq \epsilon_x \cdot \frac{|x|}{|x + y|} + \epsilon_y \cdot \frac{|y|}{|x + y|}
\end{aligned}$$

□

Errori nel prodotto dimostrazione

$x \neq 0, y \neq 0, x \cdot y \neq 0, fl(fl(x) \cdot fl(y)) = fl(x) \cdot fl(y)$, cioè il prodotto di due numeri macchina è un numero macchina.

$$\begin{aligned}
\epsilon_{\otimes}(x,y) &= \frac{|(x \cdot y) - (x \otimes y)|}{|x \cdot y|} = \frac{|(x \cdot y) - (fl(x) \cdot fl(y))|}{|x \cdot y|} \\
&= \frac{|x \cdot y - x \cdot fl(y) + x \cdot fl(y) - fl(x) \cdot fl(y)|}{|x \cdot y|} \\
&= \frac{|x \cdot (y - fl(y)) + fl(y) \cdot (x - fl(x))|}{|x \cdot y|} \\
&\leq \frac{|x \cdot (y - fl(y))|}{|x \cdot y|} + \frac{|fl(y) \cdot (x - fl(x))|}{|x \cdot y|} \\
&= \frac{|y - fl(y)|}{|y|} + \frac{|x - fl(x)|}{|x|} \\
&= \epsilon_y + \epsilon_x
\end{aligned}$$

Osservazioni

1. Le operazioni macchina **prodotto** e **divisione** introducono un errore dell'ordine della precisione di macchina.
2. Con la **somma** (e la sottrazione) non si può garantire che il risultato dell'operazione sia affetto da un errore relativo piccolo. In particolare, l'errore per la somma diventa grande quando $x \approx -y$. Questo fenomeno è noto come **cancellazione numerica**.

Cancellazione Numerica

La cancellazione numerica rappresenta la perdita di cifre significative nel risultato, dovuta alla sottrazione di due numeri quasi uguali.

Esempio 1

- Consideriamo l'aritmetica di macchina $F(10, 5, *, *)$ con:
  - $a = 0.73415507$
    $fl(a) = 0.73416$
  - $b = 0.73415448$
    $fl(b) = 0.73415$
- Valore esatto:
  $a - b = 0.59 \cdot 10^{-6}$
- In aritmetica macchina:
  $fl(a - b) = 10^{-5}$
- L'errore relativo diventa: $\frac{|(a-b)-fl(a-b)|}{|a-b|} \approx 1595\%$
- La precisione di macchina in questo caso è: $u = \frac{1}{2}B^{1-t} = \frac{1}{2} \cdot 10^{-4} = 5 \cdot 10^{-5}$

L'errore relativo è molto maggiore della precisione di macchina. Noi vogliamo che l'errore relativo sia dell'ordine della precisione di macchina!

Esempio 2

- Consideriamo ora l'aritmetica di macchina $F(10, 6, *, *)$ con:
  - $a = 0.147554326$
    $fl(a) = 0.147554$
  - $b = 0.147251742$
    $fl(b) = 0.147252$
- Valore esatto: $a - b = 0.000302584$
- In aritmetica macchina: $fl(a - b) = 0.000302$
- L'errore relativo risulta: $\frac{|0.000302584-0.000302|}{0.000302584} \approx 1.9 \cdot 10^{-3} \quad (0.19\%)$
- Precisione di macchina: $u = \frac{1}{2} \cdot 10^{-5} = 5 \cdot 10^{-6}$

Anche in questo caso l'errore relativo è molto maggiore della precisione di macchina.

Proprietà delle Operazioni: Non Associatività

In aritmetica macchina alcune proprietà delle operazioni sui numeri reali non sono valide. In particolare, la **proprietà associativa** non vale:

$$(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$$

Esempio di Non Associatività

- Consideriamo:
  - $(1 \oplus 10^{-15}) \oplus 1 = 1.11 \cdot 10^{-15}$
  - $(1 \oplus 1) \oplus 10^{-15} = 10^{-15}$

L'ordine in cui vengono effettuate le operazioni influisce sul risultato.
Il problema dell'esempio è dovuto alla differenza di ordine di grandezza tra i numeri sommati, che non permette di rappresentare correttamente il risultato: si ha una perdita di cifre significative.

Esempio con Overflow/Underflow

- Siano:
  - $a = 1.0 \cdot 10^{308}$
  - $b = 1.1 \cdot 10^{308}$
  - $c = -1.001 \cdot 10^{308}$
- Calcolando:
  - $a \oplus (b \oplus c) = 1.0 \cdot 10^{308} \oplus (0.99 \cdot 10^{307}) = 1.099 \cdot 10^{308}$
  - $(a \oplus b) \oplus c = \text{Inf} \oplus c = \text{Inf}$

Questo esempio evidenzia come la violazione dell'associatività possa portare a problemi di overflow o underflow.

Cancellazione numerica e stabilità di un algoritmo

Un metodo numerico (formula, algoritmo) si dice **stabile** se non propaga gli errori (inevitabili) dovuti alla rappresentazione dei numeri nel calcolatore. Altrimenti si dice **instabile**.

- La cancellazione numerica genera delle formule instabili.
- Per evitare i problemi legati alla cancellazione numerica occorre trasformare le formule in altre numericamente più stabili.

- La stabilità è un concetto legato all'algoritmo usato per risolvere un determinato problema.

Esempii di Algoritmo Instabile

Esempio 1

$\sqrt{x+\delta} - \sqrt{x}$ per $\delta \to 0$

Razionalizzando:

$$\sqrt{x+\delta} - \sqrt{x} = \frac{(\sqrt{x+\delta} - \sqrt{x})(\sqrt{x+\delta} + \sqrt{x})}{\sqrt{x+\delta} + \sqrt{x}} = \frac{\delta}{\sqrt{x+\delta} + \sqrt{x}}$$

Formula risolutiva delle equazioni di secondo grado

$ax^2 + bx + c = 0, \quad a \neq 0$
$x^2 + \frac{b}{a}x + \frac{c}{a} = 0$
con $\frac{b}{a} = 2p, \quad \frac{c}{a} = -q$
$x^2 + 2px - q = 0$

$x_{1,2} = -p \pm \sqrt{p^2 + q}$

Potenzialmente instabile per $p \gg q > 0$, a causa della sottrazione di due numeri quasi uguali (cancellazione numerica).

Soluzione stabile: razionalizzare la formula.

$x_1 = -p + \sqrt{p^2+q} \cdot \frac{(p+\sqrt{p^2+q})}{(p+\sqrt{p^2+q})} = -p + \frac{q}{p+\sqrt{p^2+q}}$

Successione Ricorrente

Si vuole calcolare la seguente successione di integrali definiti:

$$I_n = \frac{1}{e}\int_0^1 x^n e^x dx, \quad n \geq 0$$

Dove:

$$I_0 = \frac{1}{e}\int_0^1 e^x dx = 1 - \frac{1}{e} \approx 0.632121$$

Per $n \geq 1$, integrando per parti:

$$I_n = \frac{1}{e}\left[x^n e^x\Big|_0^1 - n\int_0^1 x^{n-1}e^x dx\right]$$

Si ottiene la formula ricorsiva:

$$I_n = 1 - nI_{n-1}, \quad n \geq 1$$

Si noti che $0 < I_n < 1$.

Instabilità della Formula Ricorsiva

Implementando la formula $I_n = 1 - nI_{n-1}$ per $n = 2, \ldots, 25$, i valori calcolati mostrano un comportamento instabile.

Analisi dell'Errore

Nel calcolatore:

$$(I_n + \epsilon_n) = 1 - n(I_{n-1} + \epsilon_{n-1})$$

Sottraendo la relazione teorica $I_n = 1 - nI_{n-1}$:

$$\epsilon_n = -n\epsilon_{n-1}$$

Per induzione:

$$|\epsilon_n| = n!|\epsilon_0|$$

Il fattore $n!$ amplifica l'errore iniziale su I_0. Per esempio, nel calcolo di I_{20}:

$$\epsilon_{20} = 20!\epsilon_0 \approx 2.7 \times 10^2 \epsilon_0$$

Formula Alternativa Stabile

Una successione ricorrente alternativa può essere calcolata all'indietro, partendo da un'approssimazione di I_m:

$$I_{n-1} = \frac{1}{n}(1 - I_n), \quad n = m, m-1, \ldots, m-k+1$$

Questa formula è **stabile**, poiché l'errore diminuisce a ogni passo.

Smorzamento dell'Errore

L'errore al passo $n-1$ è:

$$\epsilon_{n-1} = \frac{-1}{n}\epsilon_n$$

Iterando:

$$|\epsilon_{m-1}| = \frac{|\epsilon_m|}{m}, \quad |\epsilon_{m-2}| = \frac{|\epsilon_m|}{m(m-1)}, \quad \ldots, \quad |\epsilon_{m-k}| = \frac{|\epsilon_m|}{m(m-1)\ldots(m-k+1)}$$

La produttoria al denominatore **riduce rapidamente** l'errore iniziale!

Per esempio, calcolando I_{25} partendo da $I_{40} = 0.5$, l'errore iniziale $|\epsilon_{40}| < 0.5$ viene abbattuto di un fattore:

$$40 \cdot 39 \cdots 27 \cdot 26 \approx 5.26 \times 10^{22}$$

Condizionamento di un Problema

Definizione

Un problema è **mal condizionato** se piccole variazioni nei dati producono grandi variazioni nei risultati. Se invece il problema è **ben condizionato**, gli errori nei dati iniziali restano contenuti nei risultati.

Il malcondizionamento è **indipendente** dall'algoritmo scelto: se un problema è mal condizionato, nessun algoritmo potrà fornire una soluzione accurata.

Esempio

Consideriamo il sistema lineare:

$$\begin{cases} x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

La soluzione esatta è $x = 1, y = 1$.

Se perturbiamo il coefficiente della x nella prima equazione di 0.01:

$$\begin{cases} 1.01x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

La nuova soluzione diventa $x \approx -0.1111$, $y \approx 2.1122$.

L'errore relativo:

$$err_x = \frac{|1 + 0.1111|}{1} \approx 1.1111, \quad err_y = \frac{|1 - 2.1122|}{1} \approx 1.1122$$

Entrambi superiori al **100%**, dimostrando il malcondizionamento del problema.

Numero di Condizionamento

Il **numero di condizionamento** misura la sensibilità di un problema rispetto a variazioni nei dati iniziali.

Per la valutazione di una funzione $f(x)$ in un punto:

$$y = f(x)$$

Se x è perturbato di Δx, per il teorema di Lagrange:

$$f(x + \Delta x) - f(x) = \Delta x f'(\xi)$$

L'errore relativo:

$$\left|\frac{\Delta y}{y}\right| = \left|\frac{\Delta x f'(\xi)}{y}\right| = \left|\frac{\Delta x}{x}\right| \cdot \left|\frac{x f'(x)}{y}\right|$$

Definiamo quindi il **numero di condizionamento**:

$$K(f,x) = \left| \frac{xf'(x)}{y} \right|$$

Esempio

Per la funzione:

$$f(x) = \sqrt{1-x^2}$$

Il numero di condizionamento è:

$$K(f,x) = \frac{x^2}{1-x^2}$$

Il problema peggiora quanto più x si avvicina a 1:

x	$K(f,x)$
$1-10^{-6}$	4.99999×10^5
$1-10^{-12}$	5.00011×10^{11}
$1-10^{-15}$	5.00399×10^{14}

Più x è vicino a 1, maggiore è il numero di condizionamento, indicando un **problema mal condizionato**.

Calcolo degli Zeri di una Funzione

Introduzione

Problema: trovare gli zeri di una funzione $f(x)$, ovvero i punti in cui $f(x) = 0$, con $f : [a,b] \to \mathbb{R}$ continua.

Zeri: $\alpha \in [a,b]$ tale che $f(\alpha) = 0$.

Esempi:
$f(x) = (x-1)^2 = 0$
$f(x) = \cos(\log(\frac{1}{x})) = 0$

Molteplicità di uno zero: α è *zero semplice* se $f(\alpha) = 0$ e $f'(\alpha) \neq 0$. La molteplicità di uno zero è l'indice della prima derivata non nulla in α.

Esempi:
$f(x) = \cos(x) - 1 + \frac{x^2}{2} + \frac{x^5}{5} = 0$
$f'(x) = -\sin(x) + x + x^4$
$f'(0) = 0$ $f''(x) = -\cos(x) + 1 + 4x^3$
$f''(0) = 0$
$f'''(x) = \sin(x) + 12x^2$
$f'''(0) = 0$
$f^{(4)}(x) = \cos(x) + 24x$
$f^{(4)}(0) = 1 \neq 0 \Rightarrow$ zero di molteplicità 4

Esistenza dello zero - Teorema di Bolzano

Teorema di Bolzano:

$f : [a,b] \to \mathbb{R}$ continua. Se $f(a) \cdot f(b) < 0$, allora $\exists \alpha \in [a,b]$ tale che $f(\alpha) = 0$.

Unicità dello zero

Se $f(x)$ è **monotona in senso stretto** su $[a,b]$ e $\exists \alpha \in [a,b]$ tale che $f(\alpha) = 0$, allora α è unico (condizione sufficiente).

Nota: la monotonia in senso stretto è definita come $f'(x) > 0$ o $f'(x) < 0$ per ogni $x \in [a,b]$.

Esempio:

$f(x) = 2x^2 + \log(x) - \dfrac{1}{x}$

Esistenza:

So che fra 0.5 e 1 c'è uno zero. Per il teorema degli zeri, se $f(0.5) \cdot f(1) < 0$ allora c'è uno zero.

$f(0.5) = 2 \cdot 0.5^2 + \log(0.5) - 2 = -2.1931$
$f(1) = 2 \cdot 1^2 + \log(1) - 1 = 1$

Quindi $\exists \alpha \in [0.5, 1]$ tale che $f(\alpha) = 0$.

Unicità:

$f'(x) = 4x + \dfrac{1}{x} - \dfrac{1}{x^2} > 0$ per $x \in [0.5, 1]$. Quindi lo zero è unico, poiché la funzione è monotona in senso stretto.

```
>> f = @(x) 2*x.^2 + log(x) - 1./x
f =

@(x) 2 * x .^ 2 + log (x) - 1 ./ x

>> f(0.5)
ans = -2.1931
>> f(1)
ans = 1
>> fplot(f, [0.5, 1]); hold on; plot([0.1 1], [0 0], 'k-');
>> format long
>> fzero(f, 0.8)
ans = 0.832233982809322
```

Metodi per il Calcolo degli Zeri

I metodi numerici per il calcolo degli zeri di una funzione sono **iterativi**.

Definizione: Un **metodo iterativo** è una procedura che genera una successione $\{x_k\}_{k \geq 0}$ a partire da uno o più valori iniziali e quindi a partire da uno o più termini precedenti.

Definizione: Un metodo iterativo è **convergente** ad α se $\lim\limits_{k \to \infty} x_k = \alpha$ o equivalentemente $\lim\limits_{k \to \infty} |x_k - \alpha| = 0$.
$|x_k - \alpha| = \varepsilon_k$ è l'errore al passo k.

Definizione: Un metodo iterativo è **localmente convergente** ad α se $\exists \delta > 0$ raggio di convergenza tale che $\forall x_0 \in B(\alpha, \delta) = [\alpha - \delta, \alpha + \delta]$ si ha che $\lim\limits_{k \to \infty} x_k = \alpha$.

Definizione: si dice **ordine di convergenza** p di un metodo iterativo convergente (tale che $x_k \to \alpha$) se esiste una costante $C > 0$ tale che $\lim\limits_{k \to \infty} \dfrac{|\varepsilon_{k+1}|}{|\varepsilon_k|^p} = C$ (con $p \geq 1, p \in \mathbb{R}$).

Se $p = 1$ il metodo è detto **lineare**, se $p = 2$ è detto **quadratico**. Per $p > 1$ il metodo è detto **superlineare**.

Se $\lim_{k\to\infty} \frac{|\varepsilon_{k+1}|}{|\varepsilon_k|} = 0$ avremo convergenza **superlineare**.

Per $p = 1$, togliendo il limite ho che:

$$\frac{|\varepsilon_{k+1}|}{|\varepsilon_k|} \sim C \Rightarrow |\varepsilon_{k+1}| \sim C \cdot |\varepsilon_k|$$

C è la **costante asintotica di riduzione dell'errore**. Per $p = 1$, C deve essere minore di 1 affinché il metodo sia convergente.

Per $p = 2$, togliendo il limite ho che:

$$\frac{|\varepsilon_{k+1}|}{|\varepsilon_k|^2} \sim C \Rightarrow |\varepsilon_{k+1}| \sim C \cdot |\varepsilon_k|^2$$

In questo caso, C non determina la convergenza del metodo, poiché ε_k tende a 0 più velocemente. ???