

# Rappresentazione dei dati reali e teoria degli errori

Corso: Laboratorio Matematico-Informatico

Titolare del Corso: Nicoletta Del Buono

---

---

---

---

---

---

---

---

## Analisi dell'errore: introduzione

- L'analisi dell'errore è un argomento peculiare in ogni processo numerico
- Discuteremo principalmente della rappresentazione dei numeri reali sul calcolatore
  - errori di arrotondamento non dovuti ai dati, ma imputabili alla capacità finita del calcolatore di memorizzarli
  - a cui si sommano errori di misurazione

---

---

---

---

---

---

---

---

## Analisi dell'errore: introduzione

- L'ambiente numerico è un sistema contaminato da errori→
  - **importante** saper controllare e limitare la loro propagazione
- Stabilità degli algoritmi utilizzati nella risoluzione di diversi problemi→
  - algoritmi matematicamente equivalenti possono non esserlo dal punto di vista numerico

---

---

---

---

---

---

---

---

## Rappresentazione dei dati reali

- Sorgenti di errore
- Rappresentazione dei numeri reali in una base assegnata
- Rappresentazione approssimata nelle unità aritmetiche dei computer
- Errori di arrotondamento e di troncamento
- Propagazione degli errori in un algoritmo numerico progettato per risolvere un dato problema

---

---

---

---

---

---

---

---

## Sorgenti di Errore

I risultati numerici possono essere influenzati da diversi tipi di errori:

- semplificazioni introdotte in un modello
- errore nei dati
- errori di rappresentazione (arrotondamenti nei dati e nei calcoli)
- errori analitici

---

---

---

---

---

---

---

---

## Semplificazioni introdotte nel modello

- Sono errori dovuti a varie semplificazioni effettuate per descrivere il problema da risolvere:
  - ipotesi di linearità del modello matematico
  - trascurabilità di alcune grandezze fisiche
  - ecc.
- Adeguatezza del modello

---

---

---

---

---

---

---

---

## Errori nei dati

- I dati di un problema sono il risultato di misurazioni che possono essere influenzate da due tipi di errore:
- **errori sistematici**: dipendono dalla sensibilità dello strumento di misurazione
- **errori random**: sono dovuti al verificarsi in concomitanza di eventi imprevedibili
- **Una stima degli errori nei dati è importante nel calcolo numerico**

---

---

---

---

---

---

---

---

## Errore di rappresentazione

- Gli errori di arrotondamento nei dati e nei calcoli sono gli errori introdotti nella **rappresentazione dei numeri** all'interno di un computer
- A causa della capacità di memoria **finita** di un calcolatore non è possibile rappresentare gli **infiniti** numeri reali nè quei numeri che contengono (nella base utilizzata) un numero infinito di cifre

---

---

---

---

---

---

---

---

## Errore di rappresentazione

- I numeri di macchina sono quei numeri reali che un calcolatore può rappresentare esattamente
- L'insieme dei numeri di macchina (che indicheremo con  $\Phi$ ) è un insieme di **cardinalità finita**
- Ogni numero reale non appartenente ad  $\Phi$  verrà opportunamente approssimato da un numero di macchina
  - l'errore che ne consegue è detto **ERRORE DI RAPPRESENTAZIONE**

---

---

---

---

---

---

---

---

## Errore analitico

- Un ulteriore errore si introduce quando un procedimento infinito è approssimato mediante un procedimento finito
  - Esempio: Operazioni di passaggio al limite vengono approssimate con le operazioni che richiedono un numero finito di passi.
- Ogni algoritmo implementabile al calcolatore dovrà comunque prevedere un numero finito di operazioni
  - l'errore che ne consegue è detto **ERRORE ANALITICO**

---

---

---

---

---

---

---

---

## Errore computazionale

- L' errore di rappresentazione e quello analitico costituiscono l'**errore computazionale**
- Nello studio degli algoritmi numerici si cerca di ridurre al massimo questo tipo di errore, cercando però di conservare :
  - **l'affidabilità**
  - **l'efficienza**

---

---

---

---

---

---

---

---

## Rappresentazione dei numeri reali

I problemi da affrontare sono:

- la rappresentazione dei numeri reali (*infiniti*) in un calcolatore (*memoria finita*).
- l'ambiguità di rappresentazione di un numero reale nelle notazioni scientifiche.

Per esempio il numero **12,47**:

$$12,47 = 1,247 \times 10^1 = 0,1247 \times 10^2 = 1247 \times 10^{-2}$$

---

---

---

---

---

---

---

---

## Teorema di Esistenza ed Unicità di rappresentazione in una base

- **Teorema:** Sia  $B \in \mathbb{N}$ ,  $B > 1$ ,  $\forall x \in \mathbb{P} \setminus \{0\}$ , si può rappresentare univocamente in base  $B$  nella seguente forma:

$$x = \pm (d_0 B^0 + d_1 B^{-1} + \dots + d_k B^{-k} + \dots) B^p$$

- il segno dipende dalla positività o negatività di  $x$
- $p \in \mathbb{Z}$  e le cifre  $d_i$  verificano le seguenti proprietà

- 1)  $0 \leq d_i < B$
- 2)  $d_0$  è una cifra diversa da zero
- 3) le  $d_i$  non sono definitivamente uguali a  $B-1$

---

---

---

---

---

---

---

---

## Rappresentazione in base $B$ (forma normalizzata)

**DEF:** Dato un numero reale diverso da zero  $x$  l'espressione

$$\begin{aligned} x &= \pm (d_0 B^0 + d_1 B^{-1} + \dots + d_k B^{-k} \dots) B^p = \\ &= \pm B^p \sum_{i=0}^{\infty} d_i B^{-i} \end{aligned}$$

- si dice **rappresentazione in base  $B$**  del numero  $x$ ,
- il numero  $p$  si dice **esponente** o **caratteristica**
- I valori  $d_i$  sono le **cifre della rappresentazione**
- $d_0 d_1 d_2 d_3 \dots d_k \dots$  si dice **mantissa** del numero

---

---

---

---

---

---

---

---

## Rappresentazione in base $B$

- La **notazione posizionale** (o **forma normalizzata**) si può utilizzare scegliendo una qualunque base  $B$  (numero intero)
- L'ambiguità di rappresentazione di un numero reale viene risolta (la **notazione posizionale è unica**)
- L'uso della notazione normalizzata è utile quando si hanno a disposizione solo un numero finito di cifre per rappresentare un numero (ovvero ciò che avviene nel calcolatore)

---

---

---

---

---

---

---

---

## Rappresentazione in base: esempio

- Esempio: la base che utilizziamo per rappresentare i numeri reali è  $B=10$ 
  - $d_i$  utilizzabili sono 0,1,2,3,4,5,6,7,8,9
  - Rappresentiamo il numero  $x = 102.56$ 
    - $x = 1 \times 10^2 + 0 \times 10^1 + 2 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$
  - La forma normalizzata del numero è:
    - $x = (1.0256) \times 10^2$
    - =  $1 \times 10^0 + 0 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3} + 6 \times 10^{-4}) \times 10^2$

---

---

---

---

---

---

---

---

## Rappresentazione in base: esempio

- La base utilizzata dai calcolatori è  $B=2$  (sistema binario)
  - $d_i$  utilizzabili sono 0,1
  - Rappresentiamo il numero  $y = 0.0101$ 
    - $y = 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$
  - La forma normalizzata del numero è:
    - $y = (1.01) \times 2^{-2}$

---

---

---

---

---

---

---

---

## Rappresentazione floating point

- Un numero reale  $x$  diverso da zero può essere rappresentato (in generale) con infinite cifre nella mantissa.
- I numeri reali sono infiniti

---

---

---

---

---

---

---

---

## Rappresentazione floating point

- Poiché i registri delle unità aritmetiche ed i dispositivi di memoria di un calcolatore consentono l'allocazione di un **numero finito** di cifre binarie, non si possono, in generale, memorizzare tutte le cifre presenti nella rappresentazione normale del numero.
- Risulta importante stabilire dei criteri che permettano di rappresentare nel calcolatore i numeri reali in modo da commettere il minimo errore possibile

---

---

---

---

---

---

---

---

## Rappresentazione floating point

- La rappresentazione di un numero all'interno di un calcolatore viene effettuata assegnando delle posizioni di memoria per il segno, per l'esponente e per la mantissa del numero.
- La rappresentazione di un numero di può schematizzare come

segno   esponente   Mantissa

---

---

---

---

---

---

---

---

## Numeri di macchina

- L'insieme dei numeri rappresentabili nel precedente modo su un determinato calcolatore sono detti **numeri di macchina**
- Formalmente possiamo definire l'insieme dei numeri di macchina nel seguente modo:

---

---

---

---

---

---

---

---

## Numeri di Macchina

**DEF** L'insieme dei numeri di macchina, con  $t+1$  cifre significative, base  $B$ , e range  $[-m, M]$  è dato da:

$\Phi(B, t, -m, M) =$

$$\{x \in \mathbb{R}, x = \pm B^p \sum_{i=0}^t d_i B^{-i}\} \cup \{0, \text{Inf}, -\text{Inf}, \text{NaN}\}$$

- 1)  $B > 1, t \geq 0$
- 2)  $0 \leq d_i < B-1$
- 3) la cifra  $d_0$  è diversa da zero
- 4)  $-m < p < M$

---

---

---

---

---

---

---

---

## Numeri di Macchina

• **NaN** acronimo di **Not a Number**, compare come risultato di forma indeterminate

- Inf - Inf
- Inf / Inf
- 0 / 0
- 0 x Inf
- $1^{\text{Inf}}$

---

---

---

---

---

---

---

---

## Numeri di macchina

• Riassumendo, la rappresentazione in virgola mobile di un numero di macchina in una base  $B$  e con  $t+1$  cifre significative è del tipo

$$x = \pm B^p \times d_0.d_1d_2d_3\dots d_t$$

• con esponente  $p$  tale che:

$$-m \leq p \leq M \quad \text{con } m, M > 0$$

---

---

---

---

---

---

---

---

## Numeri di macchina

- Lo zero viene assegnato per definizione all'insieme dei numeri di macchina perchè sfugge alla rappresentazione in base normalizzata
- Normalmente lo zero viene rappresentato con mantissa nulla ed esponente  $-m$

---

---

---

---

---

---

---

---

## Numeri di macchina: esempio

- Esempio 1: consideriamo l'insieme dei numeri di macchina  $F(2,2,-2,3)$  ( $B=2, t=2, m=2, M=3$ ) e rappresentiamo solo i numeri positivi
  - generico numero di macchina:
    - $x = (1.d_1d_2) 2^p, p \in (-2,3)$
  - il più piccolo numero di macchina:
    - $1.00 \times 2^{-1} (= 1/2)$
  - Il più grande numero di macchina:
    - $1.11 \times 2^2 (= 7)$
  - Quali sono gli altri elementi dell'insieme  $F(2,2,2,3)$  ?

---

---

---

---

---

---

---

---

## Numeri di macchina

- I calcolatori usano un sistema di rappresentazione in base 2 o 16
  - Base 2: ciascun campo di memoria coincide con un BIT
  - Base 16: ciascun campo di memoria è costituito da 4 BIT

---

---

---

---

---

---

---

---

## Numeri di macchina

- Un calcolatore che abbia che seguenti caratteristiche:
  - Mantissa:  $r$  campi di memoria, ciascuno dei quali può assumere  $B$  differenti configurazioni;
  - Esponente: un campo di memoria che può assumere  $m+M+1$  differenti configurazioni
  - Segno: un campo di memoria che può assumere due diverse configurazioni
- rappresenta esattamente tutti i numeri di macchina aventi la precedente rappresentazione

---

---

---

---

---

---

---

---

## Standard IEEE

- L'ampiezza dello spazio dedicato all'esponente ed alla mantissa e le modalità della loro memorizzazione, varia da calcolatore a calcolatore e ne rappresenta una caratteristica fondamentale.
- Una configurazione molto diffusa è quella stabilita dallo standard dell'IEEE (Institute of Electrical and Electronic Engineers)
  - singola e doppia precisione

---

---

---

---

---

---

---

---

## Standard IEEE

- Lo standard IEEE assegna un'area
- di 32 posizioni per la *semplice precisione*
  - di 64 per la *doppia precisione*
- così distribuite:
- 1 posizione per il segno
  - 8 o 11 posizioni per l'esponente
  - 23 o 52 posizioni per la mantissa

---

---

---

---

---

---

---

---

## Standard IEEE

- Nella semplice precisione, le 8 posizioni binarie dedicate all'esponente consentono di rappresentare numeri da 0 a  $2^8-1=255$ .
- In realtà gli estremi 0 e 255 sono riservati ad altri scopi e quindi i numeri utili sono compresi tra 1 e 254.
- Se  $p$  è l'esponente della rappresentazione normale di  $x$ , nell'area in oggetto verrà allocato il numero  $q = p+127$ .
- Di conseguenza  $p$  può variare tra -126 e 127.

---

---

---

---

---

---

---

---

## Standard IEEE

- Per quanto riguarda la mantissa della rappresentazione floating point di  $x$ , poiché la parte intera è sempre 1, nello spazio ad essa dedicato verrà memorizzata solo la sua parte decimale.
- Poiché in base 2 la mantissa più piccola è 1.00000...0 e la più grande è 1.11111...1 ( $<2$ ), il più piccolo numero positivo rappresentabile è  $2^{-126}$  mentre il più grande è quasi  $2^{128}$ . Analogamente nel caso della doppia precisione

---

---

---

---

---

---

---

---

## Standard IEEE

- Se durante le operazioni si va al di sotto o al di sopra di tali valori, l'unità aritmetica segnala rispettivamente errore di **underflow** o **overflow**.
- Lo zero è rappresentato dalla sequenza di 32 o 64 zeri.

---

---

---

---

---

---

---

---

## Errori di underflow e di overflow

- Se l'esponente  $p$  con cui si memorizza un numero è più piccolo del più piccolo numero macchina, cioè  $p < -m$  allora l'elaboratore segnala un messaggio di **underflow**
- Se  $p > M$ , allora il numero è più grande del più grande numero di macchina e in questo caso si verifica un errore di **overflow**

---

---

---

---

---

---

---

---

## Conversione di un numero reale in un numero di macchina

- Consideriamo un numero reale  $x$ 
$$x = \pm (d_0 . d_1 d_2 \dots d_t \dots) B^p$$
- Se  $-m < p < M$  e  $d_i = 0$  per ogni  $i > t$  allora  $x$  è già un numero di macchina
- Se  $p < -m$  allora  $x$  è convertito in zero ed il calcolatore segnala un errore di underflow
- Se  $p > M$  allora il calcolatore segnala un errore di overflow

---

---

---

---

---

---

---

---

## Conversione di un numero reale in un numero di macchina

- Se le cifre della mantissa di  $x$  sono in numero superiore a  $t$  dobbiamo operare qualche approssimazione per la sua memorizzazione nel calcolatore
- Tra i vari modi di operare ci sono 2 strategie:
  - **troncamento** (  $tr(x)$  )
  - **Arrotondamento** (  $arr(x)$  )

---

---

---

---

---

---

---

---

## Troncamento

- Il **troncamento** consiste semplicemente nel trascurare tutte le cifre decimali della mantissa successive alla  $t$ -esima

- Consideriamo il nostro numero reale

$$x = \pm B^p \times d_0.d_1d_2d_3\dots d_t d_{t+1}d_{t+2}\dots$$

- il troncamento all  $t$ -esima cifra è:

$$x^* = tr(x) = \pm B^p \times d_0.d_1d_2d_3\dots d_t$$

---

---

---

---

---

---

---

---

## Troncamento

- La memorizzazione nei registri sarà:

$\pm$	$p$	$d_1, \dots, d_t$
-------	-----	-------------------

- l'errore assoluto commesso è:

$$|x - x^*| = |x - tr(x)| \leq B^{p-t}$$

---

---

---

---

---

---

---

---

## Arrotondamento

- L'arrotondamento di un numero  $x$  invece è definito nel seguente modo:

$$x^* = arr(x) = B^p \times d_0.d_1\dots d_t^*$$

$$d_t^* = \begin{cases} d_t + 1 & \text{se } d_{t+1} \geq B/2 \\ d_t & \text{se } d_{t+1} < B/2 \end{cases}$$

---

---

---

---

---

---

---

---

## Arrotondamento

- La memorizzazione nei registri sarà:

$\pm$	$p$	$d_1, \dots, d_t^*$
-------	-----	---------------------

- l'errore commesso :

$$|x - x^*| = |x - \text{arr}(x)| \leq \frac{1}{2} B^{p-t}$$

---

---

---

---

---

---

---

---

## Floating point

- I numeri  $\text{tr}(x)$  e  $\text{arr}(x)$  sono detti entrambi rappresentazioni in **virgola mobile (floating point)** del numero  $x$ .
- La funzione che associa al numero reale  $x$  il numero di macchina  $\text{tr}(x)$  oppure  $\text{arr}(x)$  viene indicata genericamente con  $\text{fl}(x)$  (**floating point** del numero reale  $x$ ).

---

---

---

---

---

---

---

---

## Errore Assoluto e Relativo

- Se  $x$  è un numero reale e  $x^* = \text{fl}(x)$  è la sua rappresentazione di macchina chiameremo:

- Errore assoluto:**

$$E_a = |x - \text{fl}(x)|$$

- Errore relativo** (se  $x$  è diverso da zero):

$$E_r = |x - \text{fl}(x)| / |x|$$

---

---

---

---

---

---

---

---

## Maggiorazioni dell'errore assoluto

Risulta che:

$$|x - fl(x)| \leq \rho B^{p-t}$$

con

$$\rho = \begin{cases} 1 & \text{se } fl(x) = tr(x) \\ 1/2 & \text{se } fl(x) = arr(x) \end{cases}$$

- L'errore dipende non solo dalla lunghezza  $t$  della mantissa ma anche da  $p$ , cioè dall'ordine di grandezza del numero  $x$

---

---

---

---

---

---

---

---

## Maggiorazioni dell'errore relativo

Risulta che:

$$\frac{|x - fl(x)|}{|x|} \leq \varepsilon$$

con

$$\varepsilon = \begin{cases} B^{-t+1} & \text{se } fl(x) = tr(x) \\ \frac{1}{2} B^{-t+1} & \text{se } fl(x) = arr(x) \end{cases}$$

è detto **precisione di macchina** o **zero di macchina**

---

---

---

---

---

---

---

---

## Precisione di macchina

- Nei calcolatori la base più comunemente usata è la base 2 e le cifre sono indicate con 0 ed 1.
- La precisione di macchina o epsilon di macchina o unità di arrotondamento è data da

$$\varepsilon = \rho 2^{-t}$$

$\varepsilon$  è un parametro caratteristico del calcolatore

---

---

---

---

---

---

---

---

## Precisione di macchina

- La relazione che lega un numero reale  $x$  alla sua rappresentazione di macchina  $fl(x)$  è:

$$fl(x) = x(1 + u)$$

- essendo  $|u|$  minore o uguale della precisione di macchina.

---

---

---

---

---

---

---

---

## Precisione di macchina

- La conoscenza della precisione di macchina è essenziale per un controllo della propagazione dell'errore nell'esecuzione dei vari algoritmi che vedremo nel corso.
- L'epsilon di macchina si può calcolare con il seguente algoritmo.

---

---

---

---

---

---

---

---

## Algoritmo per il calcolo della precisione di macchina

```
e=1
i=0
while 1+e > 1
    e=e/2
    i=i+1
end
print (i-1,2*e)
```

---

---

---

---

---

---

---

---

## Epsilon di macchina

I valori in output  $i-1$  e  $2^*e$  sono rispettivamente

- il numero delle cifre decimali della mantissa
- l'epsilon di macchina.
- La doppia precisione dello standard IEEE fornisce il valore:

$$\varepsilon = 2^{-52} = 2.22044 \dots \cdot 10^{-16}$$

---

---

---

---

---

---

---

---

## Operazioni di macchina

- In generale non è garantito che una operazione aritmetica su due numeri di macchina fornisca come risultato un numero di macchina.
- I numeri di macchina non sono un insieme chiuso rispetto alle operazioni aritmetiche.

---

---

---

---

---

---

---

---

## Operazioni di macchina

- Per poter realizzare la naturale ed importante proprietà di chiusura di una operazione in un certo insieme risulta importante definire delle operazioni di macchina che permettano di realizzare tale proprietà.
- Indichiamo con  $\mathcal{F}$  l'insieme dei numeri di macchina, e con  $*$  una delle quattro operazioni aritmetiche (+, -, x, /)

---

---

---

---

---

---

---

---

## Operazioni di macchina

- la corrispondente operazioni di macchina deve essere:

$$x \otimes y = (x * y)(1 + u), \quad |u| \leq \varepsilon$$

- per ogni  $x, y$  in  $\mathcal{F}$  tali che il numero ottenuto dall'operazione di macchina non dia luogo ad overflow o underflow

---

---

---

---

---

---

---

---

## Operazioni di macchina

- Si dimostra che:

$$x \otimes y = tr(x * y)$$

$$x \otimes y = arr(x * y)$$

- verificano la precedente condizione quindi danno luogo ad operazioni di macchina.

---

---

---

---

---

---

---

---

## Operazioni di macchina

- **In altre parole una operazione di macchina coincide con l'approssimazione floating del risultato dell'operazione esatta**

- Una singola operazione tra numeri di macchina da luogo ad un errore.

Quindi se il risultato di una operazione è il dato di una operazione successiva, l'errore sul dato si propagerà nel risultato finale

---

---

---

---

---

---

---

---

## Aritmetica finita

- Le operazioni di macchina definite come:

$$x \otimes y = fl(x * y)$$

- danno luogo alla **aritmetica di macchina** o **aritmetica finita**

Per le operazioni di macchina non valgono alcune proprietà, come l'associatività dell'addizione e della moltiplicazione o la distributività della moltiplicazione rispetto l'addizione

---

---

---

---

---

---

---

---

## Sensibilità delle operazioni di macchina

- Una singola operazione tra numeri di macchina da luogo ad un errore relativo dell'ordine della precisione di macchina.
- Quando però il risultato di una operazione è dato di input di una operazione successiva, l'errore si propaga sul risultato finale
- La misura della propagazione dell'errore dipenderà da quanto l'operazione è **sensibile** alle perturbazioni sui dati

---

---

---

---

---

---

---

---

## Problema ben condizionato

- Un problema si dice **ben posto** o **ben condizionato** se introducendo una perturbazione sui dati di input, il risultato viene perturbato di una quantità dello stesso ordine di grandezza delle perturbazioni sui dati.

---

---

---

---

---

---

---

---

## Sensibilità dell'operazione somma

- Consideriamo due numeri reali  $x$  e  $y$  e siano  $fl(x)$  e  $fl(y)$  le loro rappresentazioni di macchina
- Calcoliamo l'errore relativo che viene commesso dall'elaboratore quando calcola  $x+y$ :

$$E_r = \frac{|(fl(x) \oplus fl(y)) - (x+y)|}{|(x+y)|} \leq 3 \max(|u_x|, |u_y|, |u|)$$

con  $|u_x|, |u_y|, |u| < \epsilon$

---

---

---

---

---

---

---

---

## Sensibilità dell'operazione differenza

- Consideriamo due numeri reali  $x$  e  $y$  e siano  $fl(x)$  e  $fl(y)$  le loro rappresentazioni di macchina
- Calcoliamo l'errore relativo che viene commesso dall'elaboratore quando calcola  $x-y$ :

$$E_r = \frac{|(fl(x) \ominus fl(y)) - (x-y)|}{|(x-y)|} \leq |u^*| (1 + |x|/|x-y| + |y|/|x-y|)$$

con  $|u^*| \leq \epsilon$

---

---

---

---

---

---

---

---

## Sensibilità dell'operazione differenza

- Se  $x$  e  $y$  sono due numeri aventi lo stesso segno allora l'errore relativo può essere molto grande quanto più  $x$  e  $y$  sono vicini.
- Questo fenomeno prende il nome di **cancellazione di cifre significative**

---

---

---

---

---

---

---

---

## Propagazione dell'errore

- Algoritmo è una sequenza ordinata di operazioni elementari con le quali si ottiene la soluzione di un determinato problema.
- Ogni operazione della sequenza verrà eseguita come operazione di macchina dando quindi origine a due componenti di errore:
- **approssimazione delle operazioni di macchina**
- **propagazione dell'errore accumulato.**

---

---

---

---

---

---

---

---

## Propagazione dell'errore

- La prima componente di errore in un algoritmo, dovuta alle approssimazioni delle operazioni di macchina, non supera l'unità di arrotondamento
- La seconda componente di errore è la più importante e la più "pericolosa" ed è dovuta alla propagazione dell'errore accumulato fino al passo precedente .
- Essa dipende esclusivamente dalla natura **stabile** o **instabile** dell'operazione che si compie al passo considerato.

---

---

---

---

---

---

---

---

## Applicazione della teoria degli errori

- Come applicazione della teoria svolta, supponiamo di voler calcolare il valore del polinomio:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

- Il calcolo del valore del polinomio  $p(x)$  in un dato punto  $x_0$  può essere fatto in due modi:
- **attraverso lo schema naturale**
- **attraverso l'algoritmo di Horner**

---

---

---

---

---

---

---

---

## Schema naturale per il calcolo di $p(x)$

Dati  $x_0, a_0, a_1, \dots, a_n$

$s=1;$

$p=a_0;$

**for**  $i=1,2,\dots,n$

$s=s*x_0;$

$p=p+a_j*s$

**end**

**print** ( $p$ )

Il costo computazionale del precedente algoritmo è di  $n$  somme e  $2n$  moltiplicazioni

---

---

---

---

---

---

---

---

## Algoritmo di Horner

Il metodo di Horner si basa sul fatto che il polinomio  $p(x)$  può essere scritto nella seguente forma:

$$p(x) = (\dots((a_n x + a_{n-1})x + a_{n-2})x + \dots + a_1)x + a_0$$

ALGORITMO: Dati:  $x_0, a_0, \dots, a_n$

$p=a_n$

**for**  $i=1,2,\dots,n$

$p=p*x_0+a_{n-i}$

**end**

**print**  $p$

---

---

---

---

---

---

---

---

## Algoritmo di Horner

I vantaggi dell'algoritmo di Horner sono:

- minore costo computazionale, infatti algoritmo richiede il calcolo di  $n$  somme ed  $n$  moltiplicazioni
- maggiore stabilità dello schema naturale per piccoli valori del punto  $x$

---

---

---

---

---

---

---

---