

# Andiamoci vicino: l'importanza dell'approssimazione in matematica (e nella realtà)

Stefano FINZI VITA - Davide PASSARO

# Elogio dell'approssimazione

- Nelle cose che si fanno si punta a ottenere il risultato perfetto, ma non sempre è possibile. Spesso è sufficiente avvicinarsi all'obiettivo. La questione è casomai capire quanto vicini ci siamo arrivati.
- Non significa accontentarsi, ma valutare accettabili gli effetti di tale scostamento di risultato. È il concetto di **approssimazione della soluzione**, non una sconfitta, anzi spesso l'unica strategia vincente.
- In matematica è una strada fondamentale, che consente di passare senza traumi **dalla teoria alla pratica, dal continuo al discreto, dalle formule a programmi efficienti per il computer**.
- Oggi vedremo alcune questioni interessanti legate all'approssimazione: il calcolo delle cifre decimali di numeri importanti, degli zeri di una funzione e delle aree di regioni piane.
- Per capire questi metodi saranno utili brevi codici di Python da eseguire all'interno di un apposito Notebook Colab.

# Indice

## I parte (**Stefano, slides**)

- Introduzione: la matematica è una *scienza esatta*?
- L'**analisi numerica**: ruolo e obiettivi
- La rappresentazione dei numeri in **virgola mobile** e l'**algebra dei numeri macchina**
- Primi esempi: il calcolo approssimato di  $\pi$ ,  $e$ ,  $\sqrt{2}$
- La **ricerca degli zeri** in generale. Metodi: **bisezione**, **Newton**, **secanti**

## II parte (**Davide, notebook**)

- L'importante aiuto di **Python** con i Notebook Jupyter di Colab
- **Il calcolo delle aree** mediante l'approssimazione di integrali.  
Metodi: **rettangoli**, **trapezi** e **Montecarlo**
- Una **proposta didattica per il quinto anno** delle superiori

# Matematica: una scienza esatta?



Definizione ottocentesca:

*Una scienza esatta è quella in grado di rispondere a qualsiasi domanda con rigore e risultati esatti, esprimibili in modo analitico e oggettivo.*

È la concezione deterministica della *meccanica classica*. Si cita in proposito il **demone di Laplace**: se esistesse un'intelligenza tale da conoscere in un certo istante di tempo la posizione e le condizioni di moto di ogni singola particella dell'Universo e tutte le leggi della fisica che lo governano, allora, per questo essere, il demone appunto, sarebbe perfettamente possibile conoscere l'esatta condizione di ogni particella nel futuro e, allo stesso tempo, ricostruire con esattezza ogni evento del passato.

Era l'idea formulata da **Laplace**, puramente teorica, per l'impossibilità concreta di effettuare una tale mole di calcoli.

Oggi grazie ai super computer e l'AI si potrebbe riformulare?

## Matematica: una scienza esatta? (segue)

In realtà le leggi della *meccanica quantistica* (introdotte nel Novecento) ci dicono che un tale demone non potrà mai esistere: le particelle subatomiche non si comportano come una pallina da tennis, ma seguono leggi governate dal caso, ci sono fenomeni che avvengono con una certa probabilità, è la fisica del possibile, dell'indeterminatezza.

Quindi oggi una definizione moderna più corretta sarebbe:

*Una scienza esatta è quella che, per il suo rigore metodologico (metodo scientifico), è in grado di produrre risultati e predizioni con un'accurata espressione quantitativa.*

**F. Enriques:** "Le scoperte scientifiche non sono la verità assoluta, ma ci avvicinano sempre di più ad essa."



# La soluzione

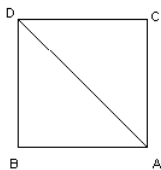
Veniamo dunque ai problemi della matematica. In genere per ogni problema gli sforzi si fanno per trovare 'la soluzione', possibilmente attraverso un'espressione che ce la fornisca in modo esplicito.

La questione è però complessa, e richiede alcuni passi fondamentali.

- **Esistenza:** si può dimostrare che la soluzione esista?
- **Unicità:** se esiste ce n'è una sola? Potrebbero esserne più di una, o addirittura infinite.
- **Formula:** anche se è unica, occorre trovare un modo di calcolarla.
- Se troviamo la formula, potrebbe non bastare per ottenerne il valore numericamente 'esatto'.

## Un esempio elementare: la radice di 2

- Il problema più banale che viene in mente è quello della misura della diagonale del quadrato unitario.
- Il teorema di Pitagora ci risolve tutte le questioni precedenti, fornendoci anche la formula finale: il risultato è  $\sqrt{2}$ , un numero irrazionale però, quindi non rappresentabile nei calcoli in forma decimale finita: possiamo cioè solo approssimarlo con la precisione desiderata.



La questione non sta però nel fatto che la soluzione sia irrazionale. Potremmo avere lo stesso problema anche se fosse razionale, magari pure decimale finito, ma, e questo è meno facile da capire, semplicemente non rappresentabile sul nostro computer.

## Risolvere equazioni

In molti problemi dunque quello che serve non è tanto avere la formula della soluzione esatta: **serve un metodo efficiente per approssimarla.**

- Se parliamo di approssimazione nasce immediatamente il concetto di **errore**. Possiamo ad esempio dire che  $\pi = 3.14$ , se ci accontentiamo di un errore dell'ordine di un centesimo. Parleremo quindi anche di **ordine di approssimazione**.
- Ma c'è di più: spesso il metodo di approssimazione equivale ad un algoritmo iterativo, a volte perfino infinito, che ci fornirebbe cioè la soluzione cercata solo al limite. Ecco allora che ci serve una **stima dell'errore** che commettiamo ad ogni iterazione, per poter decidere quando fermarci: parleremo allora di **criterio di arresto**.
- L'**analisi numerica** è il ramo della matematica che si occupa di fornire appunto dei metodi di approssimazione per calcolare le soluzioni dei problemi con la **precisione** desiderata, dimostrandone la convergenza, e possibilmente valutandone l'**efficienza** del calcolo.



# La rappresentazione dei numeri e i numeri macchina

Per parlare di approssimazione e implementare algoritmi su di un computer, diventa essenziale partire dalla rappresentazione dei numeri.

- Occasione per riparlare dei reali, delle notazioni scientifiche (decimale, esponenziale, in virgola mobile), dei cambi di base, sottolineando le differenze ai fini dell'utilizzo all'interno di un computer.
- Ci avventureremo così nel sorprendente mondo dei **numeri macchina**, cioè di quelli effettivamente rappresentabili su di un computer: un **insieme 'colabrodo'** in confronto al rassicurante continuo dei reali.
- **L'algebra dei numeri macchina è molto diversa da quella dei reali:** identità che diventano false, proprietà che non valgono più, operazioni 'pericolose' ed effetti disastrosi anche di piccole perturbazioni.
- Per capirlo meglio partiremo da un **'toy computer'**, cioè un modello giocattolo dove è possibile lavorare con carta e penna in base 10, ragionando sulla **solitudine dello zero e degli altri numeri interi**.

## La rappresentazione dei numeri: il caso continuo

Ogni numero reale può essere rappresentato come un numero decimale illimitato, periodico o no, in notazione posizionale:

$$x = n.a_1a_2 \dots a_k \dots = n + \frac{a_1}{10} + \frac{a_2}{10^2} + \dots + \frac{a_k}{10^k} + \dots$$

dove  $n \in \mathbb{Z}$  (parte intera),  $a_i = 0, 1, \dots, 9$ . In particolare:

- **periodici**  $\rightarrow$  **razionali** ( $\mathbb{Q}$ ):

Es.:  $2(.00000\dots)$ ,  $3.5(00000\dots)$ ,  $0.\bar{6}$ ,  $7.02\bar{428}$

- **non periodici**  $\rightarrow$  **irrazionali** ( $\mathbb{R} \setminus \mathbb{Q}$ ):

Es.:  $\sqrt{2}$ ,  $\sqrt{3}$ ,  $\pi$ ,  $e$

Ogni numero reale  $x$  può essere approssimato (con precisione arbitraria) mediante un numero razionale  $\bar{x}$  [**densità di  $\mathbb{Q}$  in  $\mathbb{R}$** ] e **la retta reale non ha buchi** [**continuità dei numeri reali**].

Se  $\bar{x}$  approssima  $x$ , definiremo:

- **errore assoluto:**  $e_A(x) = |x - \bar{x}|$
- **errore relativo:**  $e_R(x) = \frac{|x - \bar{x}|}{|\bar{x}|} = \frac{e_A}{|\bar{x}|}$
- **arrotondamento per difetto** di  $x$  a  $k$  cifre:  $x_{(k)} = n.a_1a_2 \dots a_k$
- **arrotondamento per eccesso** di  $x$  a  $k$  cifre:  
 $x^{(k)} = n.a_1a_2 \dots (a_k + 1) = x_{(k)} + 10^{-k}$
- **arrotondamento di  $x$  alla  $k$ -ma cifra :**  
 $fl_k(x) = x_{(k)}$  se  $a_{k+1} < 5$ ;  $fl_k(x) = x^{(k)}$  se  $a_{k+1} \geq 5$
- $x_{(k)} \leq x < x^{(k)}$ ,  
 $|x - x_{(k)}| < 10^{-k}$ ,  $|x - x^{(k)}| < 10^{-k}$ ,  
 $|x - fl_k(x)| < 0.5 * 10^{-k}$
- **Domanda:** se approssimiamo 10000 con 9999, e 1 con 0.99, quale delle due approssimazioni vi sembra la migliore?

## Virgola mobile

Per operare con i numeri decimali è più comodo rappresentarli in **virgola mobile** (*floating point*), una forma normalizzata che ne metta in risalto l'ordine di grandezza (le cifre significative):

$$x = \pm m \cdot 10^z = (\pm m, z), \quad 0.1 \leq m < 1, \quad z \in \mathbb{Z}$$

con  $m =$  **mantissa**,  $z =$  **esponente**.

Per  $x \neq 0$ , la corrispondenza  $x \rightarrow (m, z)$  è **biunivoca**<sup>(\*)</sup>. Esempi:

$$9 \rightarrow (0.9, 1)$$

$$-123.81471 \rightarrow (-0.12381471, 3)$$

$$0.0000715 \rightarrow (0.715, -4).$$

---

(\*) L'unica eccezione sarebbero i numeri 9-periodici, però identificabili con i rispettivi equivalenti 0-periodici. Ad esempio  $0.\overline{9} = 1$ ,  $3.4\overline{9} = 3.5$ , ecc. (del resto nessuna divisione tra interi genera un numero 9-periodico...)

# La memoria del computer e i numeri macchina

- I computer elaborano le informazioni (istruzioni e dati) in **formato digitale**. L'unità è il **bit** (**binary digit**) che può valere solo 0 o 1. Le celle di memoria sono i **byte**, sequenze di 8 bit. Un numero verrà rappresentato **in base due**, usando in genere 2, 4 o 8 byte.
- I reali saranno memorizzati in **virgola mobile**, con un **numero fissato di cifre per la mantissa e l'esponente**.
- Se  $\mathcal{A}$  è l'insieme dei numeri rappresentabili, con  $k$  cifre per la mantissa e  $j$  per l'esponente, non apparterranno ad  $\mathcal{A}$  i numeri:
  - ▶ irrazionali (decimali illimitati non periodici)
  - ▶ decimali periodici (periodo diverso da zero)
  - ▶ decimali finiti con più di  $k$  cifre significative per la mantissa
  - ▶ in modulo maggiori di  $M$  (massimo numero macchina) [**overflow**]
  - ▶ in modulo minori di  $m$  (minimo numero macchina positivo, o **zero macchina**) [**underflow**]

## Toy computer

Il nostro insieme  $\mathcal{A}$  è dunque un colabrodo, e tutti i numeri che non sono in  $\mathcal{A}$  vanno approssimati con il più vicino numero macchina. Per capire meglio ragioniamo su di un esempio semplificato che però riproduce fedelmente il tipo di limitazioni del caso reale e le sue conseguenze.

Supponiamo che il nostro computer giocattolo sia in grado di trattare solo numeri (positivi) con TRE cifre per la mantissa e UNA per l'esponente (in base 10 per comodità). Quindi se  $x \in \mathcal{A}$  e  $x \neq 0$ :

$$x = 0.a_1a_2a_3 \cdot 10^{\pm e} = (a_1a_2a_3, \pm e), \quad a_1 \neq 0.$$

- **Esercizio 1.** In questo caso chi sono  $M$  (il massimo numero macchina) e  $m$  (lo zero macchina, il più piccolo numero maggiore di zero rappresentabile)?

## Toy computer

Il nostro insieme  $\mathcal{A}$  è dunque un colabrodo, e tutti i numeri che non sono in  $\mathcal{A}$  vanno approssimati con il più vicino numero macchina. Per capire meglio ragioniamo su di un esempio semplificato che però riproduce fedelmente il tipo di limitazioni del caso reale e le sue conseguenze.

Supponiamo che il nostro computer giocattolo sia in grado di trattare solo numeri (positivi) con TRE cifre per la mantissa e UNA per l'esponente (in base 10 per comodità). Quindi se  $x \in \mathcal{A}$  e  $x \neq 0$ :

$$x = 0.a_1a_2a_3 \cdot 10^{\pm e} = (a_1a_2a_3, \pm e), \quad a_1 \neq 0.$$

- **Esercizio 1.** In questo caso chi sono  $M$  (il massimo numero macchina) e  $m$  (lo zero macchina, il più piccolo numero maggiore di zero rappresentabile)?

- **Soluzione:**

$$M = 0.999 \cdot 10^9 = (999, 9) \text{ (999 milioni)}$$

$$m = 10^{-10} = 0.1 \cdot 10^{-9} = (1, -9).$$

- **Esercizio 2.** Quali dei seguenti numeri appartengono ad  $\mathcal{A}$  e quali no? Per ognuno scrivere la sua rappresentazione come numero macchina (arrotondando quando necessario):

0.000715, 12.7,  $\sqrt{2}$ ,  $\frac{1}{3}$ , 39210,  $10^{-8}$ , 217.5,  $10^9$ , 700.000.000,  $10^{-11}$



- **Esercizio 2.** Quali dei seguenti numeri appartengono ad  $\mathcal{A}$  e quali no? Per ognuno scrivere la sua rappresentazione come numero macchina (arrotondando quando necessario):

$$0.000715, 12.7, \sqrt{2}, \frac{1}{3}, 39210, 10^{-8}, 217.5, 10^9, 700.000.000, 10^{-11}$$

- **Appartengono ad  $\mathcal{A}$ :**

$$0.000715 = (715, -3)$$

$$12.7 = (127, 2)$$

$$10^{-8} = (100, -7)$$

$$700.000.000 = (700, 9)$$

- **Non appartengono ad  $\mathcal{A}$  (tra parentesi il valore per arrotondamento):**

$$\sqrt{2} \simeq (141, 1)$$

$$1/3 \simeq (333, 0)$$

$$39210 \simeq (392, 5)$$

$$217.5 \simeq (218, 3)$$

$$10^9 \simeq M$$

$$10^{-11} \simeq 0$$

L'insieme  $\mathcal{A}$  non è chiuso rispetto alle quattro operazioni.

- **Esercizio 3.** Eseguire i seguenti calcoli. Per ognuno scrivere la rappresentazione dei numeri in ingresso (appartenenti ad  $\mathcal{A}$ ) e quella del risultato rappresentabile in  $\mathcal{A}$ , ragionando sulla sua correttezza:

$$823 + 195, \quad 13.4 - 2.75, \quad 4 \times 5 \cdot 10^8, \quad 1000/32.$$

L'insieme  $\mathcal{A}$  non è chiuso rispetto alle quattro operazioni.

- **Esercizio 3.** Eseguire i seguenti calcoli. Per ognuno scrivere la rappresentazione dei numeri in ingresso (appartenenti ad  $\mathcal{A}$ ) e quella del risultato rappresentabile in  $\mathcal{A}$ , ragionando sulla sua correttezza:

$$823 + 195, \quad 13.4 - 2.75, \quad 4 \times 5 \cdot 10^8, \quad 1000/32.$$

- **Soluzione:**

$$823 + 195 = (823, 3) + (195, 3) = (102, 4) \neq 1018$$

(si è usato l'arrotondamento);

$$13.4 - 2.75 = (134, 2) - (275, 1) = (107, 2) \neq 10.65$$

$$4 \times 5 \cdot 10^8 = (400, 1) \times (500, 9) = M \neq 2 \cdot 10^9 \text{ (overflow);}$$

$$1000/32 = (100, 4)/(320, 2) = (312, 2) \neq 31.25$$

In  $\mathcal{A}$  non valgono molte proprietà elementari (associativa, distributiva).

- **Esercizio 4.** Controllare che per  $a = 1$ ,  $b = 0.005$   $c = 0.007$

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

per  $a = 3$ ,  $b = 815$  e  $c = 521$

$$a \times (b - c) \neq a \times b - a \times c$$

In entrambi i casi qual è la sequenza più corretta e perché?

In  $\mathcal{A}$  non valgono molte proprietà elementari (associativa, distributiva).

- **Esercizio 4.** Controllare che per  $a = 1$ ,  $b = 0.005$   $c = 0.007$

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

per  $a = 3$ ,  $b = 815$  e  $c = 521$

$$a \times (b - c) \neq a \times b - a \times c$$

In entrambi i casi qual è la sequenza più corretta e perché?

- **Soluzione:**

$$(a + b) + c = (1 + 0.005) + 0.007 \simeq 1 + 0.007 \simeq 1$$

$$a + (b + c) = 1 + 0.012 \simeq \mathbf{1.01} \quad \text{Valore esatto: } \mathbf{1.012}$$

$$a \times (b - c) = 3 \times (815 - 521) = 3 \times 294 = \mathbf{882} \text{ (valore esatto)}$$

$$a \times b - a \times c \simeq 2440 - 1560 = \mathbf{880}$$

## L'elemento neutro della somma non è unico

$$x + y = x \quad \nRightarrow \quad y = 0$$

Lo **zero macchina** è il primo numero rappresentabile a destra dello zero, quindi ci dà un'idea della *solitudine del numero zero*, e grazie alla rappresentazione in virgola mobile può essere molto piccolo: le cifre significative possono infatti scorrere verso destra giocando sull'esponente.

Chiameremo invece **precisione macchina** il più piccolo numero positivo  $\varepsilon$  tale che  $1 + \varepsilon > 1$ . Essa misura la *solitudine del numero uno*, cioè la taglia dei buchi più grandi del nostro 'colabrodo', indicando lo spazio vuoto tra uno e il successivo numero rappresentabile: poiché stavolta non possiamo perdere la prima cifra significativa relativa all'unità, agire sull'esponente non aiuta. In genere, se  $b$  indica la base utilizzata e  $k$  il numero di cifre a disposizione per la mantissa, vale  $\varepsilon = b^{1-k}$ .

Nel nostro toy computer  $\varepsilon = 10^{1-3} = 10^{-2} = 0.01 = (1, -1)$ ; quindi:  $1 + 0.01 = 1.01 = (101, 1)$ , mentre  $1 + 0.009 = 1 = (100, 1)$  !

## Propagazione degli errori

Quando  $x$  e  $y$  sono due numeri positivi molto vicini tra loro, anche se gli errori relativi  $e_R(x)$ ,  $e_R(y)$  sono piccoli può risultare  $e_R(x - y)$  molto grande. In altre parole **la sottrazione di due numeri quasi uguali provoca la perdita di cifre significative** del risultato, compromettendo i nostri calcoli.

- **Esercizio 5** Considerate le due espressioni numeriche equivalenti:

$$A_n = \sqrt{n+1} - \sqrt{n}, \quad B_n = \frac{1}{\sqrt{n+1} + \sqrt{n}}.$$

Quale formula conviene usare, ad es. per  $n = 90$ ?

## Propagazione degli errori

Quando  $x$  e  $y$  sono due numeri positivi molto vicini tra loro, anche se gli errori relativi  $e_R(x)$ ,  $e_R(y)$  sono piccoli può risultare  $e_R(x - y)$  molto grande. In altre parole **la sottrazione di due numeri quasi uguali provoca la perdita di cifre significative** del risultato, compromettendo i nostri calcoli.

- **Esercizio 5** Considerate le due espressioni numeriche equivalenti:

$$A_n = \sqrt{n+1} - \sqrt{n}, \quad B_n = \frac{1}{\sqrt{n+1} + \sqrt{n}}.$$

Quale formula conviene usare, ad es. per  $n = 90$ ?

- **Soluzione:**

$$\mathbf{A}_{90} = (954, 1) - (949, 1) = (500, -1) = \mathbf{0.05};$$

$$\mathbf{B}_{90} = (100, 1) / ((954, 1) + (949, 1)) = (526, -1) = \mathbf{0.0526}$$

(valore corretto alle prime 3 cifre significative)



## Equazioni di secondo grado

Tutti ricordano come si risolve un'equazione di secondo grado:

$$ax^2 + bx + c = 0 .$$

Ecco lo pseudocodice di una possibile procedura per la sua risoluzione (limitandoci al **caso delle soluzioni reali distinte**):

- ① Leggi  $a, b, c$
- ② Calcola  $\Delta = b^2 - 4ac$
- ③ Se  $\Delta > 0$ , poni:  $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$ ,  $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$ .

Tutto bene, se non che abbiamo scoperto che certe operazioni sui numeri macchina sono potenzialmente pericolose, in particolare le sottrazioni tra quantità quasi uguali possono portare alla cancellazione di cifre significative del risultato.

**Esercizio 6.** Nel nostro toy computer assumiamo  $a = -0.005$ ,  $b = c = 1$ . Calcoliamo in  $\mathcal{A}$  le due soluzioni con l'algoritmo precedente e il loro errore relativo:

**Esercizio 6.** Nel nostro toy computer assumiamo  $a = -0.005$ ,  $b = c = 1$ . Calcoliamo in  $\mathcal{A}$  le due soluzioni con l'algoritmo precedente e il loro errore relativo:

- **Soluzione:** Soluzioni esatte dell'equazione:

$$x_1 = 200.99504\dots, \quad x_2 = -0.99504\dots$$

Con le formule tradizionali in  $\mathcal{A}$  si ottiene:

$$\Delta = 1.02, \sqrt{\Delta} = 1.00995\dots = 1.01 \in \mathcal{A} \text{ (per arrotondamento),}$$

$$x_1 = \frac{-2.01}{-0.01} = 201, x_2 = \frac{0.01}{-0.01} = -1;$$

quindi

$$x_1 = 201, \quad e_R(x_1) = 2 \cdot 10^{-5},$$

$$x_2 = -1, \quad e_R(x_2) = 5 \cdot 10^{-3}.$$

Però ci sono altre formule, matematicamente equivalenti. Vale infatti:

$$\frac{-b \pm \sqrt{\Delta}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \frac{-b \mp \sqrt{\Delta}}{-b \mp \sqrt{\Delta}} = \frac{2c}{-b \mp \sqrt{\Delta}}$$

(al '+' nella prima formula corrisponde il '-' nella seconda, e viceversa)  
Ricalcoliamo in  $\mathcal{A}$  le due soluzioni dell'esercizio precedente con le nuove formule e confrontiamo i risultati. Ora si ottiene

$$\begin{aligned} x_1 &= 200, \quad e_R(x_1) = 5 \cdot 10^{-3}, \\ x_2 &= -0.995, \quad e_R(x_2) = 4 \cdot 10^{-5}. \end{aligned}$$

Notate che i risultati migliori si ottengono con le prime formule per  $x_1$ , con le seconde per  $x_2$ . Questo suggerisce che un algoritmo più affidabile dovrebbe ricorrere a una formula o all'altra in base al segno di  $b$ .

Ecco un possibile frammento di pseudocodice:

- ① se  $b > 0$  allora  $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$  ;  $x_2 = \frac{2c}{-b - \sqrt{\Delta}}$  ;
- ② altrimenti  $x_1 = \frac{2c}{-b + \sqrt{\Delta}}$  ;  $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$  .

## Partiamo dai soliti noti

Per parlare di specifiche tecniche numeriche di approssimazione, cominciamo dal calcolo approssimato, nel miglior modo possibile, di tre vecchie conoscenze, tre numeri irrazionali di cui possiamo quindi solo determinare un certo numero di cifre decimali corrette:

$$\pi, \quad e, \quad \sqrt{2}$$

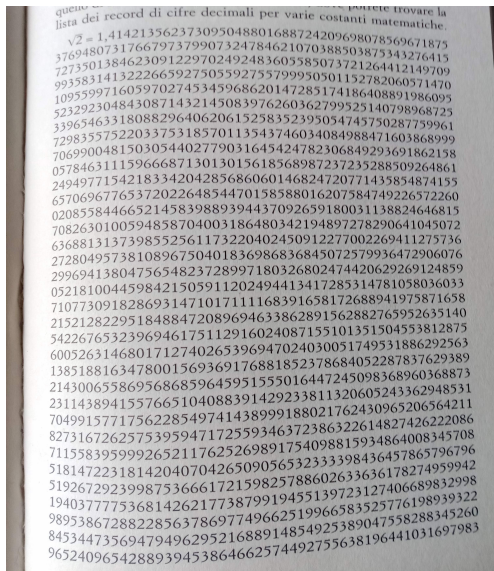
Qual è l'utilità del calcolo di tante cifre decimali?

- La propagazione degli errori ci spinge ad una maggiore precisione.
- Può servire a studiarne la distribuzione statistica ( $\sqrt{2}$ ) o per es. a testare rapidità e affidabilità dei grandi computer ( $\pi$ ).
- Gli approcci possibili sono tanti, e i risultati anche molto differenti.
- Per approfondimenti:

P. Greco, *Storia di pi greco*, Carocci

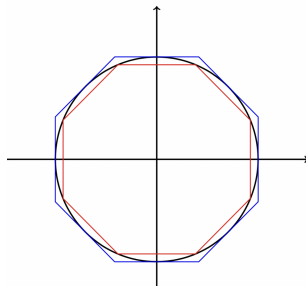
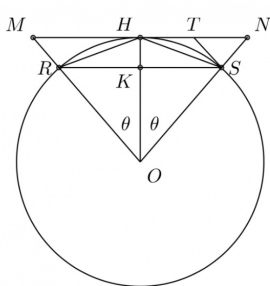
B. Rittaud, *La favolosa storia della radice quadrata di due*, Boringhieri

## Solo l'inizio



## Il calcolo di $\pi$

Il metodo più famoso per la sua approssimazione si deve ad Archimede, ma ne esistono diverse varianti. Poiché  $\pi$  coincide col valore della lunghezza della semicirconferenza di raggio unitario, possiamo dunque approssimarlo per difetto attraverso le lunghezze dei semiperimetri dei poligoni regolari inscritti al cerchio unitario, e per eccesso attraverso quelle relative ai poligoni circoscritti, lunghezze ricavabili in modo ricorsivo partendo da quelle dei quadrati (inscritto e circoscritto), raddoppiandone ad ogni iterazione i lati.



## Le formule

Detti rispettivamente  $l_n$  e  $L_n$  le misure dei lati dei poligoni inscritti e circoscritti aventi  $2^n$  lati, attraverso il Teorema di Pitagora e un po' di geometria elementare si giunge alle seguenti formule iterative:

$$l_2 = \sqrt{2}, \quad l_{n+1} = \sqrt{2 - \sqrt{4 - l_n^2}}, \quad L_n = \frac{2l_n}{\sqrt{4 - l_n^2}}, \quad n = 2, 3, \dots$$

Quindi:

$$2^{n-1}l_n < \pi < 2^{n-1}L_n$$

con una precisione che dovrebbe aumentare al crescere di  $n$ .



## Il calcolo di $\pi$ (2)

Purtroppo le cose non vanno così bene, e con questo metodo non riusciamo ad ottenere più di 8 cifre corrette (per  $n = 15$ , quindi un poligono di circa 32700 lati), poi le due successioni iniziano ad oscillare e ad allontanarsi dal valore corretto. Cosa succede?

Al crescere di  $n$  infatti  $l_n \rightarrow 0$ , e quindi  $\sqrt{4 - l_n^2} \rightarrow 2$ , con cancellazione di cifre significative nella sottrazione di due numeri quasi uguali. Metodo quindi da scartare? No, basta osservare che vale l'identità:

$$\sqrt{2 - \sqrt{4 - l_n^2}} = \frac{l_n}{\sqrt{2 + \sqrt{4 - l_n^2}}}$$

per ottenere una formula ricorsiva stabile (senza sottrazioni pericolose), in grado di fornirci un risultato corretto di  $\pi$  fino alla precisione voluta. Per esempio per  $n = 26$  si ottengono 15 cifre decimali esatte.

## Il calcolo di $e$

Una possibile definizione del numero di Nepero  $e$  è quella di limite della successione

$$a_n = \left(1 + \frac{1}{n}\right)^n ;$$

si dimostra infatti che la successione è monotona crescente e limitata superiormente, quindi ammette limite. Si potrebbe quindi pensare di approssimare  $e$  alla precisione desiderata scegliendo un numero di  $n$  sufficientemente grande.

Anche in questo caso la teoria non trova corrispondenze nel calcolo numerico. Calcolando i valori della successione per esempio per  $n$  uguale alle potenze di 10, si nota che da un certo punto in poi i valori smettono di crescere, oscillano, per poi stabilizzarsi sul numero 1 !

Cosa succede? Ormai dovrebbe essere chiaro: per  $n$  molto grande,  $1/n$  diventa molto vicino a zero, e ricordando la 'solitudine dei numeri interi', non stupisce che  $1 + 1/n = 1$ , così come le sue potenze.

## Il calcolo di $e$ (2)

Anche in questo caso per fortuna ci sono alternative, come la successione:

$$s_n = \sum_{k=0}^n a_k,$$

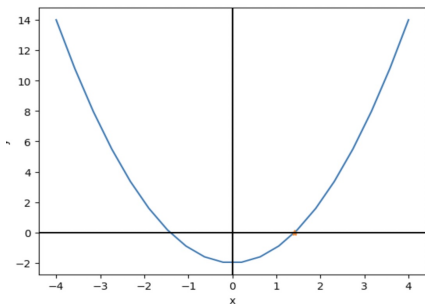
dove  $a_k$  è definita per ricorrenza da

$$a_0 = 1, \quad a_{k+1} = \frac{a_k}{k+1}.$$

È facile verificare che  $a_k = \frac{1}{k!}$ , e si può dimostrare che la sequenza converge proprio ad  $e$  (di fatto è la serie di Taylor della funzione  $e^x$  calcolata per  $x = 1$ ). Qui non entriamo nella dimostrazione, che richiede diversi prerequisiti. Ci interessa solo mostrare come formule matematiche teoricamente equivalenti forniscano numericamente risultati molto diversi. In questo caso la sequenza si avvicina rapidamente a  $e$  (già per  $n = 14$  avremo un errore dell'ordine di  $10^{-12}$ ).

## Il calcolo di $\sqrt{2}$

Ci sono molti modi di calcolare la radice di due. Un modo possibile è quello di considerarla come una soluzione dell'equazione  $x^2 - 2 = 0$ , cioè come uno zero della parabola  $f(x) = x^2 - 2$ . Graficamente stiamo parlando dell'intersezione della parabola col semiasse positivo delle  $x$ .



**Attenzione:** la formula risolutiva delle equazioni di secondo grado ci restituisce in questo caso semplicemente il valore simbolico di  $\sqrt{2}$ , ma noi vogliamo calcolarne le cifre decimali!

## Una possibile strada

Per ottenere un metodo numerico convergente ci viene in soccorso l'analisi matematica, col teorema di **Bolzano-Weierstrass**:

**Teorema.** Se  $f$  è continua nell'intervallo  $[a, b]$  e  $f(a) * f(b) < 0$ , allora esiste un valore  $c \in (a, b)$  tale che  $f(c) = 0$ .

Nel nostro caso possiamo prendere:

$$[a, b] = [0, 2], \quad f(0) = -2 < 0, \quad f(2) = 2 > 0,$$

quindi il teorema è soddisfatto e  $c$  sarà proprio il valore cercato.

## Il metodo di bisezione (dicotomia)

L'idea è semplice: il Teorema ci dice che la nostra soluzione è tra  $a$  e  $b$ . Suddividiamo l'intervallo iniziale in due parti uguali attraverso il punto medio  $c = (a + b)/2$ , calcoliamo  $f(c)$ , e in base al segno di questo potremo individuare in quale delle due metà è ancora contenuto lo zero, quindi con una precisione migliore. Ripetendo il procedimento possiamo costruire una successione di intervalli  $(a_n, b_n)$  di ampiezza ogni volta dimezzata contenenti sempre lo zero cercato.

Al passo iniziale l'errore commesso era pari a  $a$  e  $b$ , al secondo passo sarà:

$$err_0 = \frac{b-a}{2},$$

$$err_1 = \frac{b_1-a_1}{2} = \frac{b_0-a_0}{4},$$

... all' $n$ -mo:

$$err_n = \frac{b_n-a_n}{2} = \frac{b_0-a_0}{2^{n+1}}.$$

Potremo fermarci appena l'errore sarà inferiore alla tolleranza desiderata:

$$\frac{b_0 - a_0}{2^{n+1}} \leq TOL \Rightarrow \log_2\left(\frac{b_0 - a_0}{TOL}\right) \leq n + 1$$

cioè quando  $n \geq \log_2\left(\frac{b_0-a_0}{TOL}\right) - 1$  (N.B.  $n$  è indipendente da  $f!$ )

## Calcoliamo $\sqrt{2}$ con la bisezione

Consideriamo la sequenza dei punti medi successivi a partire dall'intervallo iniziale  $[0, 2]$ :

$$c_1 = 1 \quad f(c_1) = -1 \quad \Rightarrow \quad [a_1, b_1] = [1, 2]$$

$$c_2 = 1.5 \quad f(c_2) = 0.25 \quad \Rightarrow \quad [a_2, b_2] = [1, 1.5]$$

$$c_3 = 1.25 \quad f(c_3) = -0.4375 \quad \Rightarrow \quad [a_3, b_3] = [1.25, 1.5]$$

$$c_4 = 1.375 \quad f(c_4) = -0.109375 \quad \Rightarrow \quad [a_4, b_4] = [1.375, 1.5]$$

$$c_5 = 1.4375 \quad f(c_5) = 0.06640625 \quad \Rightarrow \quad [a_5, b_5] = [1.375, 1.4375]$$

$$c_6 = 1.40625 \quad f(c_6) = -0.0224609375 \quad \Rightarrow \quad [a_6, b_6] = [1.40625, 1.4375]$$

Cosa notiamo? Dopo 6 iterazioni abbiamo individuato solo la prima cifra decimale. La sequenza  $c_n$  non è monotona, ma fornisce in modo arbitrario stime per eccesso o per difetto del numero. Insomma, è un metodo sicuro (lo zero cercato non può sfuggire, e sarà sempre meglio approssimato), ma molto lento.

## Alcuni vecchi metodi

- Un metodo molto antico (IV secolo) ha viaggiato dalla Cina all'India e poi tramite gli Arabi ha raggiunto l'Occidente, ed è quello che è sopravvissuto fino alla meta del XX secolo, cioè fino all'avvento dei computer. Era lungo e complicato e per fortuna oggi si è dimenticato.
- Un'altra idea la dobbiamo a Nepero, l'inventore dei logaritmi. Le famose **tavole logaritmiche** consentivano il calcolo delle radici: poiché  $\log \sqrt{a} = \frac{\log a}{2}$ , si cerca sulla tavola il valore del logaritmo di  $a$ , lo si divide per due e si cerca sulla tavola il numero il cui logaritmo corrisponde al risultato, quello sarà  $\sqrt{a}$ .  
(È interessante scoprire che per creare le tavole logaritmiche le radici quadrate siano state a loro volta un aiuto prezioso)



## L'algoritmo di Erone di Alessandria (I sec. d.C.)

In realtà la strada migliore era anche la più antica di tutte, il cosiddetto **Metodo di Erone** (detto pure dei Babilonesi), a cui si può arrivare in vari modi: dalla geometria, dall'algebra, dall'analisi.

Assegnato un dato iniziale  $x_0$  (possibilmente una stima per eccesso del risultato atteso), ripetere per  $N$  volte (si tratta quindi di una successione per ricorrenza):

$$x_{n+1} = \frac{1}{2} \left\{ x_n + \frac{2}{x_n} \right\}$$

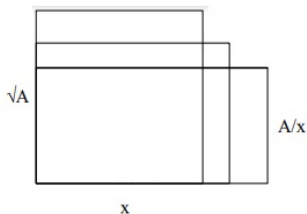
Proviamo a capirlo.

## Idea geometrica: rettangoli che tendono a un quadrato

$\sqrt{2}$  è la misura del lato del quadrato di area 2; partendo da un rettangolo di base  $x_0 > \sqrt{2}$  e altezza  $2/x_0$  (quindi di area 2), generiamo una sequenza di rettangoli equivalenti ma di dimensioni sempre più vicine tra loro, in modo da avvicinarci al quadrato obiettivo. Come nuova base  $x_1$  prendiamo la media aritmetica di  $x_0$  e  $2/x_0$  (un numero quindi compreso tra di essi).

$$x_1 = \frac{1}{2} \left\{ x_0 + \frac{2}{x_0} \right\}$$

La nuova altezza sarà dunque  $2/x_1$ . Per esempio se  $x_0 = 2$ , allora  $2/x_0 = 1$  e  $x_1 = \frac{3}{2} = 1.5$ , e così via, da cui la formula.



## Idea algebrica

Partiamo dall'equazione  $x^2 = 2$ ; allora

$$x = \frac{2}{x} \Rightarrow 2x = x + \frac{2}{x} \Rightarrow x = \frac{1}{2} \left( x + \frac{2}{x} \right) \quad (*)$$

L'algoritmo si ottiene trasformando l'ultima equazione (soddisfatta da  $\sqrt{2}$ ) in un metodo iterativo innescato da un valore iniziale. In modo equivalente potremmo osservare che per qualunque  $x \neq 0$  vale

$$\sqrt{2} = \sqrt{x \cdot \frac{2}{x}}$$

e nell'approssimare la media geometrica  $\sqrt{pq}$  di due numeri  $p$  e  $q$  con la media aritmetica  $\frac{p+q}{2}$ .

N.B. Funziona sempre? Se ci fossimo fermati al primo passo di (\*) prendendo come metodo iterativo  $x_{n+1} = \frac{2}{x_n}$ , che avremmo ottenuto?

# Facciamo agire Erone

Partiamo da  $x_0 = 2$  e calcoliamo anche stavolta le prime sei iterazioni:

$$x_1 = 1.5$$

$$x_2 = 1.4166666666666665$$

$$x_3 = 1.4142156862745097$$

$$x_4 = 1.4142135623746899$$

$$x_5 = 1.414213562373095$$

$$x_6 = 1.414213562373095$$

Otteniamo una precisione di oltre  $10^{-10}$ , con solo operazioni razionali.

## Erone: perché funziona

Non è difficilissimo provare alcune proprietà del nostro algoritmo:

- Se  $x_0 > \sqrt{2}$  allora vale sempre  $x_n > 0$
- $x_n^2 > 2$  per ogni  $n$  (tutti **valori per eccesso**)
- $x_n > x_{n+1}$  per ogni  $n$  (**la sequenza decresce** quindi verso un limite  $L$ )
- L'equazione  $L = \frac{1}{2} \left\{ L + \frac{2}{L} \right\}$  ha per soluzione proprio  $\sqrt{2}$
- Vale:  $x_n - \sqrt{2} < \frac{x_0 - \sqrt{2}}{2^n}$  (l'errore si dimezza almeno ad ogni passo)
- La **convergenza** è **quadratica**: l'errore si riduce ad ogni iterazione come il quadrato dell'errore precedente! In pratica il numero delle cifre significative corrette di  $\sqrt{2}$  raddoppia ad ogni iterazione.
- Vale anche:  $x_{n+1} - \sqrt{2} < x_n - x_{n+1}$  (l'errore ad ogni passo è minore della differenza delle ultime iterazioni: un ottimo **criterio di arresto!**)

## Erone ci risolve il calcolo delle radici quadrate, e non solo

È facile estendere l'algoritmo al calcolo di una qualunque **radice quadrata di un numero  $\alpha$** :

$$x_{n+1} = \frac{1}{2} \left\{ x_n + \frac{\alpha}{x_n} \right\}$$

ma anche per esempio alla sua **radice cubica**:

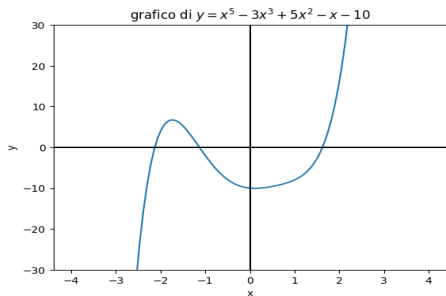
$$x_{n+1} = \frac{1}{3} \left\{ 2x_n + \frac{\alpha}{x_n^2} \right\}$$

(generalizzando l'idea geometrica, sostituendo quadrato e rettangoli con cubo e parallelepipedi), in generale alla **radice di qualunque ordine intero**.

Una tecnica così antica ma così efficiente è sopravvissuta all'avvento dei computer, anzi è stata valorizzata da questi, fornendo l'algoritmo ideale per il calcolo delle radici.

## Bisezione: caso generale

Supponiamo di voler calcolare le soluzioni  $f(x) = 0$ , con  $f$  generica funzione assegnata, per esempio un polinomio di grado elevato. Il problema è la loro localizzazione, cioè la determinazione di un intervallo che li contenga con valori di segno opposto agli estremi. Come individuarle? Esaminando il segno di  $f$  nei nodi di una partizione di passo  $h$  possiamo individuare gli intervalli dove vale il Teorema di Bolzano e applicare ad essi la bisezione. La taglia di  $h$  sarà importante (se troppo grande potremmo perdere alcune soluzioni).



Intervallo	soluz. approx.
-2.204 -2.040	-2.190
-1.224 -1.061	-1.129
1.551 1.714	1.623

## Il metodo di Newton (o delle tangenti)

C'è un altro metodo iterativo per il calcolo degli zeri di una funzione  $f$  che ci offre l'analisi numerica: data un'approssimazione iniziale  $x_0$ , otterremo  $x_1$  come l'intersezione con l'asse delle ascisse della retta tangente ad  $f$  per il punto  $(x_0, f(x_0))$ , e così di seguito. Cerchiamone la formula. La retta tangente in questione ha equazione

$$r(x) = f'(x_0)(x - x_0) + f(x_0)$$

Allora imponendo  $r(x) = 0$  otteniamo facilmente

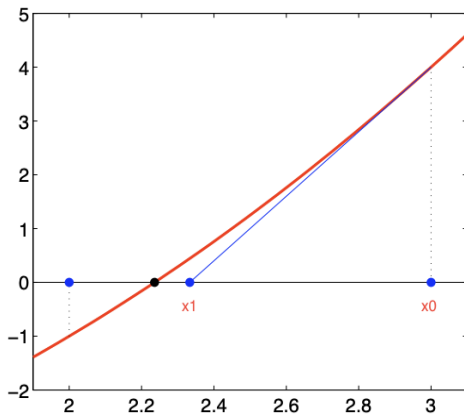
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

per cui la formula generale sarà:

$$\text{dato } x_0, \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geq 0.$$

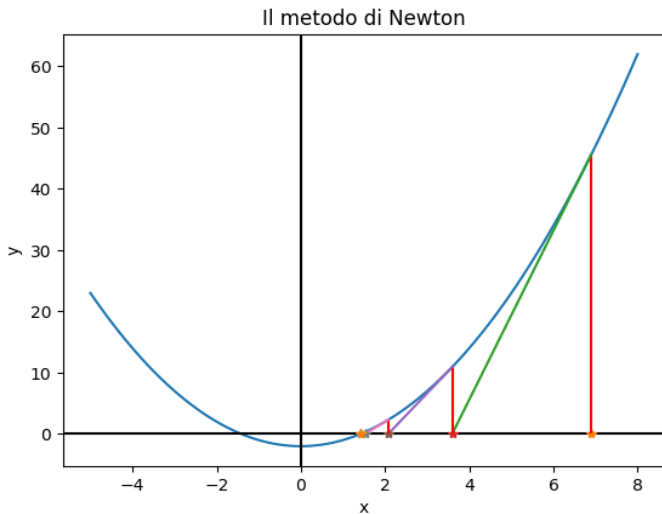


## Vediamo graficamente



In rosso il grafico di  $f$ ,  $x_0$  è il punto a destra, in azzurro è tracciata la tangente con la sua intersezione  $x_1$  con l'asse  $x$  molto più vicina alla radice cercata. Quindi si riparte da  $x_1$ .

# Vediamo graficamente



## Applichiamolo al calcolo di $\sqrt{2}$

Prima di discuterne le proprietà, vediamo come funziona nel nostro caso.

Se  $f(x) = x^2 - 2$ , allora  $f'(x) = 2x$  e il metodo diventa

$$x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n} = \frac{2x_n^2 - x_n^2 + 2}{2x_n} = \frac{1}{2} \left\{ x_n + \frac{2}{x_n} \right\}$$

Ma allora lo conosciamo già!

**Erone è il metodo di Newton applicato alla parabola.**

Abbiamo già visto la sua efficienza. Ma si può anche migliorare: sui computer si vogliono evitare le divisioni (richiedono più tempo delle moltiplicazioni). Prendiamo  $f(x) = 2 - \frac{1}{x^2}$  e cerchiamo il suo zero ( $\frac{\sqrt{2}}{2}$ ) con Newton. Allora  $f'(x) = \frac{2}{x^3}$ , e con pochi calcoli:

$$x_{n+1} = \frac{3x_n}{2} - x_n^3$$

quindi **nessuna divisione!** (in base due dividere per 2 equivale solo a spostare la virgola)

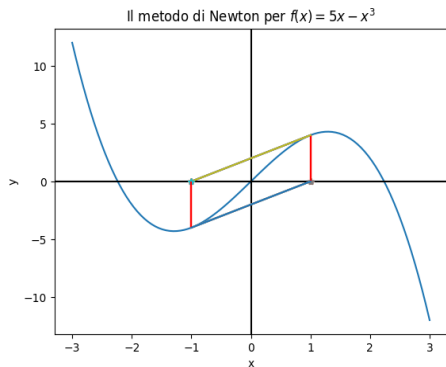
## Alcune riflessioni su Newton

Per il calcolo della radice di 2 tutto bene. Funziona sempre così?

- È chiaro che il metodo è utilizzabile solo per funzioni derivabili (per bisezione bastava la continuità).
- Per innescare Newton serve una buona stima iniziale della radice cercata (per bisezione tutto avveniva all'interno dell'intervallo di partenza).
- Serve che la derivata non si annulli nelle iterate (divisione per zero).
- Anche se in qualche iterata capita che  $f'(x_n) \simeq 0$  possono esserci seri problemi. Capite perché?

## Possibili problemi per Newton

- Se la derivata in qualche iterata ha valori molto vicini allo zero la tangente sarà quasi parallela all'asse  $x$ , e quindi la nuova iterata potrebbe schizzare molto lontano dalla precedente.
- Potrebbero succedere pure cose strane come nel seguente esempio: sia  $f(x) = 5x - x^3$ . Le tre radici sono 0 e i valori  $\pm\sqrt{5}$ . Se proviamo a lanciare Newton con  $x_0 = 1$  ecco quello che succede:



```

1 -1.0
2 1.0
3 -1.0
4 1.0
5 -1.0
6 1.0
7 -1.0
8 1.0
9 -1.0
10 1.0

```

# Proprietà di convergenza

- Si può dimostrare che il metodo converge sempre se  $x_0$  è **sufficientemente** vicino ad  $\alpha$ . (In teoria OK, ma in pratica?)
- Se il metodo converge allora lo fa in modo molto veloce (**convergenza quadratica**).
- Se la funzione  $f$  è convessa e crescente, e  $x_0 \geq \alpha$ , allora la sequenza converge decrescendo ad  $\alpha$ .

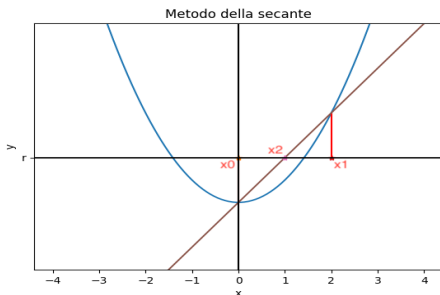
Con Python, aiutandosi con la grafica, è facile mostrare diverse situazioni particolari.

## Il metodo delle secanti

Se non abbiamo a disposizione la derivata di  $f$ , o se è complicato calcolarla, possiamo sostituirla con il rapporto incrementale  $\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$  ottenendo:

$$x_{n+1} = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})}$$

Di fatto si calcola l'intersezione con l'asse  $x$  della retta secante al grafico di  $f$  relativamente ai punti delle ultime due iterate. Il metodo risulta più lento di Newton, ma è un'alternativa valida.



# Conclusioni

Abbiamo visto diversi metodi per ottenere in teoria le stesse cose. Quale scegliere in generale?

La filosofia di base è la seguente: **si sceglie il veicolo in base al percorso.**  
Per raggiungere il più presto possibile una località, quale mezzo scelgo?





## Conclusioni

Abbiamo visto diversi metodi per ottenere in teoria le stesse cose. Quale scegliere in generale?

La filosofia di base è la seguente: **si sceglie il veicolo in base al percorso.** Per raggiungere il più presto possibile una località, quale mezzo scelgo?



**Dipende dai collegamenti e dai costi.** In pista va bene la Ferrari, ma in autostrada userò la berlina (ho i limiti di velocità e poi risparmio). In una strada provinciale piena di curve meglio l'utilitaria. In un percorso di montagna solo la Mountain bike mi può aiutare.

Nella numerica è lo stesso, dipende dai dati del problema: occorre studiare bene la regolarità delle funzioni in gioco, il costo delle operazioni, gli eventuali ostacoli. **Approssimare richiede abilità!**

# La parola a Davide (attraverso un Notebook Colab)

Link al Notebook:

[https://colab.research.google.com/drive/15tnFzDC-cTmhQn3\\_-a1tL-yX52Ld-pYY?usp=sharing](https://colab.research.google.com/drive/15tnFzDC-cTmhQn3_-a1tL-yX52Ld-pYY?usp=sharing)