

# Algoritmo di bisezione

## Algoritmo di Newton

Davide Peccioli

2 maggio 2021

Entrambi questi algoritmi sono utilizzati per calcolare con una buona approssimazione gli zeri di una funzione, quando questi non sono "banali", ovvero calcolabili con puri passaggi algebrici.

In particolar modo, il programma qui presentato prende in input una funzione qualsiasi e, dopo averne stampato il grafico e richiesto all'utente in quale intervallo vi sia **un solo** zero, calcola questo zero con due algoritmi, in due tempi separati.

Il codice commentato è presente su [GitHub](#)<sup>1</sup>.

## 1 Algoritmo di bisezione

L'algoritmo di bisezione, il primo eseguito dal programma, funziona nel seguente modo:

- l'utente sceglie un intervallo, tale per cui:
  - la funzione è continua e derivabile;
  - è presente un solo zero;
  - la funzione, agli estremi dell'intervallo, ha segno discorde;
- viene calcolato il punto medio dell'intervallo scelto;
- si stabilisce quale dei due intervalli venutisi a formare (*estremo sinistro-punto medio* o *punto medio-estremo destro*) rispetta i requisiti presentati nel punto precedente;
- l'algoritmo si ripete con questo "nuovo" intervallo.

Così facendo si isolerà con una precisione sempre maggiore l'intervallo in cui è presente lo zero. Non si giungerà *quasi* mai ad un risultato preciso, tale per cui la funzione calcolata nel punto sia esattamente 0, e pertanto è necessario porre una sensibilità al programma: nel programma qui presentato la sensibilità è impostata a  $1 \cdot 10^{-15}$ .

---

<sup>1</sup>[https://github.com/DavideP02/Scuola\\_Liceo\\_5\\_Informatica\\_Octave-AnalisiNumerica-Pecciolli5H/blob/main/analisi.m](https://github.com/DavideP02/Scuola_Liceo_5_Informatica_Octave-AnalisiNumerica-Pecciolli5H/blob/main/analisi.m)

## 2 Algoritmo di Newton

L'algoritmo di Newton (o delle tangenti), il secondo eseguito dal programma, funziona in modo analogo all'algoritmo di bisezione:

- l'utente sceglie un intervallo, che deve presentare le stesse caratteristiche dell'intervallo scelto per l'algoritmo di bisezione, ma in cui la derivata seconda della funzione deve essere continua e mantenere costante il suo segno;
- preso il punto estremo con ordinata di segno concorde con quello della derivata seconda, si traccia la retta tangente alla curva in quel punto;
- la proiezione dell'intersezione della retta con l'asse delle  $x$  sulla curva studiata rappresenterà il nuovo estremo dell'intervallo, che sostituirà quello preso in considerazione in precedenza;
- si itera l'algoritmo fino a che l'intersezione della retta tangente con l'asse delle  $x$  presenta un'immagine sufficientemente vicina allo zero (con precisione arbitrariamente scelta).

In questo modo, aggiungendo un'ipotesi sull'intervallo scelto, rispetto all'algoritmo di bisezione, è possibile diminuire fortemente il numero di iterazioni.

## 3 Funzioni studiate

Segue una tabella con indicati le funzioni studiate, gli intervalli considerati e il numero di iterazioni necessarie a ciascun algoritmo per trovare la soluzione, data una precisione di  $1 \cdot 10^{-15}$ .

| Funzione                               | Intervallo       | Bisezione | Newton |
|--|------------------|-----------|--------|
| $\arctan(x) - x^3$                     | $[-1, 2; -0, 6]$ | 47        | 6      |
| $\sin(x) - \sqrt{x} + 1$               | $[1, 5; 4]$      | 48        | 10     |
| $\log(x) - x^2 + 1$                    | $[0, 2; 0, 6]$   | 46        | 6      |
| $x^4 + x^3 - 1$                        | $[-1, 8; -1]$    | 46        | 6      |
| $(10 - x) \cdot e^{-10x} - x^{10} - 1$ | $[0; 0, 4]$      | 49        | 7      |

Come si può notare dalla tabella riportata, l'algoritmo di bisezione impiega un numero di iterazioni ben maggiore rispetto all'algoritmo di Newton, di circa il 600%.

Considerando che la precisione per gli entrambi gli algoritmi è la stessa, e che per comodità per entrambi gli algoritmi viene utilizzato lo stesso intervallo, a prima vista sembrerebbe che l'algoritmo di Newton sia il più efficiente senza ombra di dubbio. Sebbene sia effettivamente l'algoritmo più efficiente, è necessario considerare che utilizzando lo strumento matematico della derivazione, l'algoritmo di Newton impiega più risorse rispetto a quello di bisezione per ogni iterazione fatta.

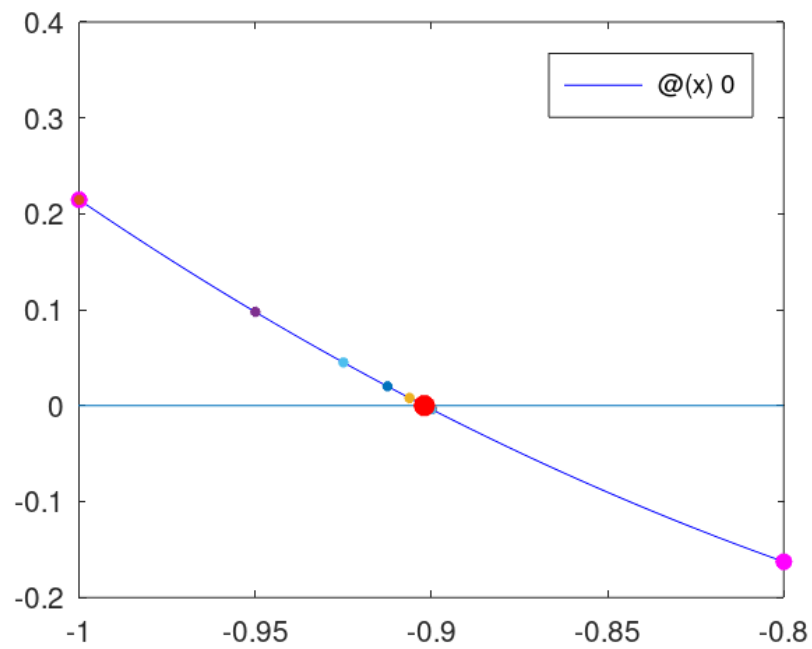


Figura 1: Punti considerati dall'algoritmo di bisezione per studiare la funzione  $\arctan(x) - x^3$

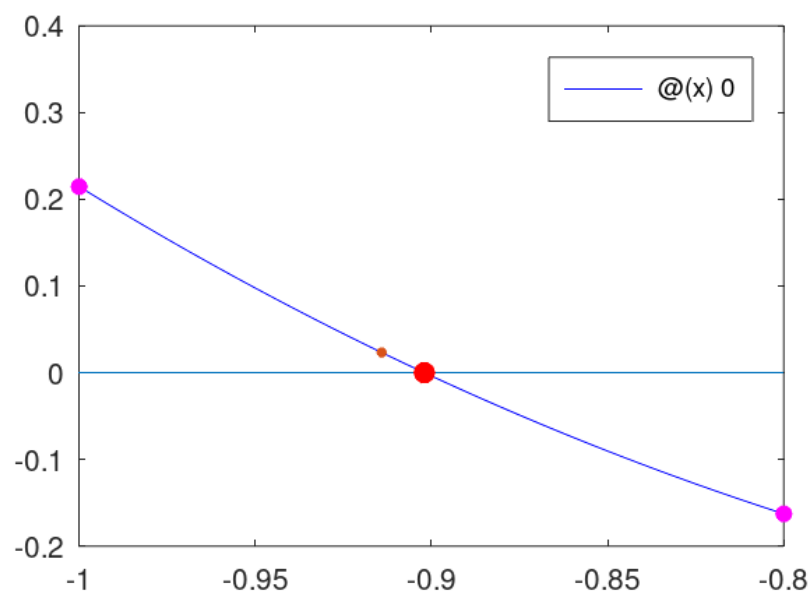


Figura 2: Punti considerati dall'algoritmo di Newton per studiare la funzione  $\arctan(x) - x^3$