

IL MIO CAPOLAVORO

Programma per il Grafico di una Funzione e della sua Derivata, con
Calcolo degli Zeri tramite Metodi Numerici

Antonella Spera 5A

Il mio capolavoro nasce dall'unione tra le competenze di informatica, acquisite durante l'anno scolastico, e quelle di matematica, applicate alla realizzazione di un programma in linguaggio Java.

Il cuore del progetto è proprio questo: partendo da una funzione continua, il programma ne disegna il grafico, calcola e visualizza anche la sua derivata e, soprattutto, trova gli zeri della funzione, cioè i punti in cui essa si annulla.

Per farlo ho implementato due metodi numerici molto utilizzati in matematica: il **metodo di bisezione**, che procede per restringimenti successivi dell'intervallo, e il **metodo delle tangenti**, conosciuto anche come metodo di Newton-Raphson, che usa la derivata per avvicinarsi rapidamente allo zero.

Per costruire il programma ho utilizzato la programmazione orientata agli oggetti, un approccio, che abbiamo studiato durante quest'anno scolastico.

Questo progetto mi ha permesso di collegare concretamente teoria e pratica. Ho potuto vedere come un'idea teorica possa diventare un algoritmo funzionante, rafforzando la mia comprensione e rendendo l'apprendimento più efficace e stimolante.

IL TEOREMA DEGLI ZERI E IL METODO DI BISEZIONE

Il metodo di bisezione si basa sul Teorema degli Zeri, noto anche come Teorema di Bolzano.

Questo teorema è uno dei risultati fondamentali dell'analisi matematica e riguarda le funzioni continue.

Cosa afferma il teorema degli Zeri:

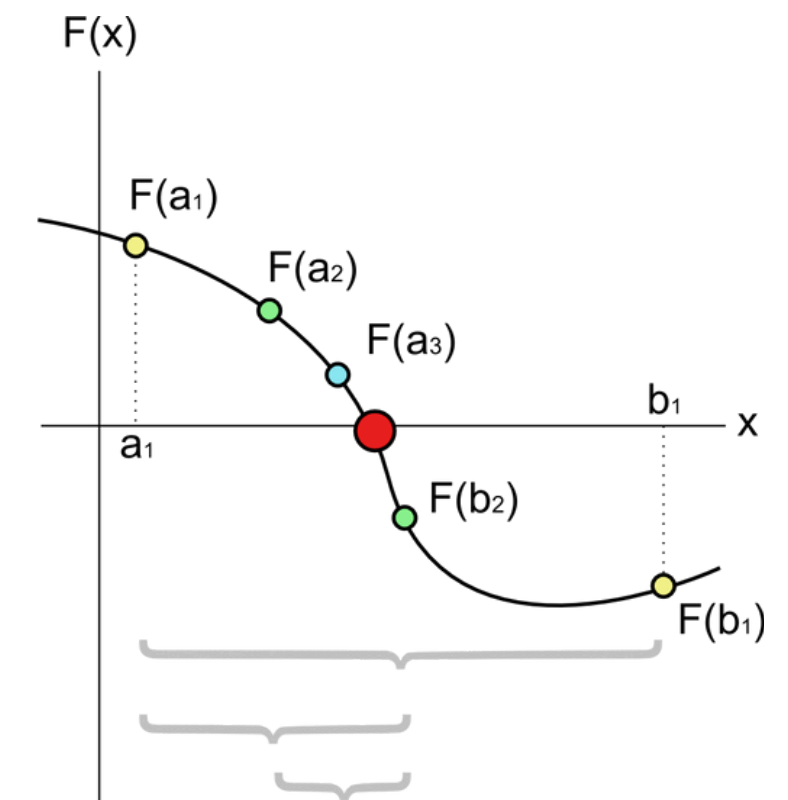
Sia f una **funzione continua** su un **intervallo chiuso** $[a,b]$. Se $f(a) \cdot f(b) < 0$, cioè se la funzione assume valori di segno opposto agli estremi dell'intervallo, allora esiste almeno un punto $c \in (a,b)$ tale che **$f(c) = 0$**

Quindi, la funzione si annulla almeno una volta nell'intervallo $[a,b]$.

Questo teorema non ci dice dove si trovi esattamente lo zero, ma garantisce l'esistenza di almeno uno zero nell'intervallo, purché la funzione sia continua e cambi segno.

Funzionamento del metodo:

1. Si verifica che $f(a) \cdot f(b) < 0$
2. Si calcola il punto medio $m = (a+b)/2$
3. Si valuta $f(m)$
4. Se $f(m) = 0$, si è trovato lo zero.
5. Altrimenti, si sceglie il sottointervallo $[a,m]$ o $[m,b]$ dove la funzione cambia segno.
6. Si ripete il procedimento fino a ottenere un'approssimazione sufficientemente precisa dello zero.



IL METODO DELLE TANGENTI O DI NEWTON-RAPHSON

Il Metodo delle Tangenti, noto anche come Metodo di Newton-Raphson, è un algoritmo numerico per trovare gli zeri di una funzione, cioè i punti x in cui $f(x)=0$. È un metodo rapido ed efficiente, ma che richiede alcune condizioni per funzionare correttamente.

Il metodo si fonda su un concetto geometrico e analitico: se una funzione è continua e derivabile in un intervallo, e se la derivata non si annulla, è possibile usare la tangente al grafico della funzione per avvicinarsi a uno zero.

Cosa afferma il teorema:

Se f è una funzione continua e derivabile in un intervallo contenente uno zero reale α , e se la derivata prima non si annulla in tale intervallo ($f'(x) \neq 0$), allora scegliendo un punto iniziale x_0 sufficientemente vicino a α , la successione x_n definita dalla formula del metodo converge a α .

Il metodo delle tangenti è molto più rapido rispetto alla bisezione, ma è più sensibile alla scelta del punto iniziale.

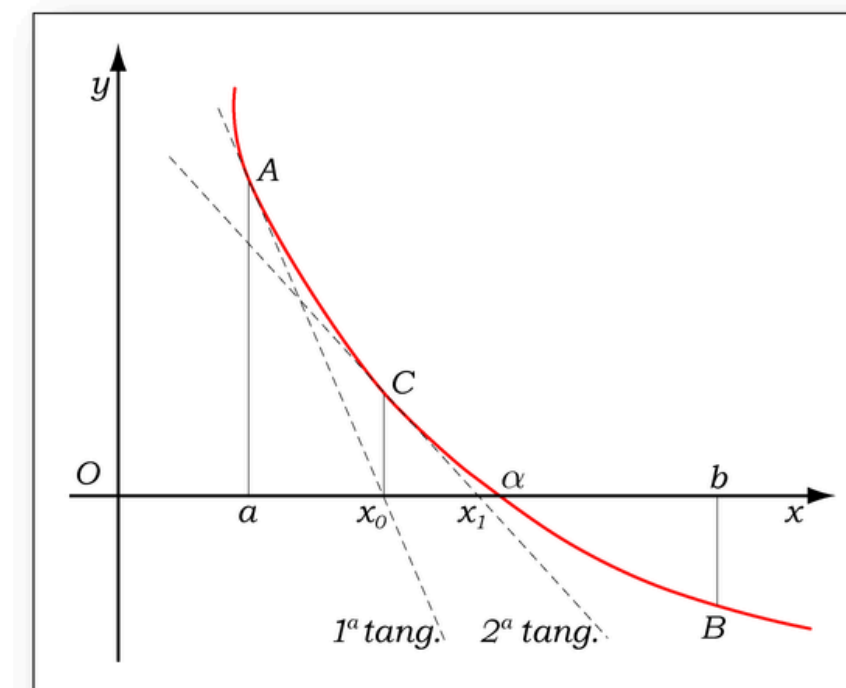
Funzionamento del metodo:

1. Si parte da un punto iniziale x_0 , che si ipotizza vicino allo zero della funzione.
2. Si calcola la tangente al grafico della funzione nel punto $(x_0, f(x_0))$.

La tangente ha come coefficiente angolare la derivata $f'(x_0)$.

3. Si trova il punto in cui questa tangente interseca l'asse delle ascisse (cioè dove $y=0$): questo punto si chiama x_1 .

4. Si ripete lo stesso procedimento partendo da x_1 , ottenendo x_2 , e così via:
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



COMPETENZE ACQUISITE

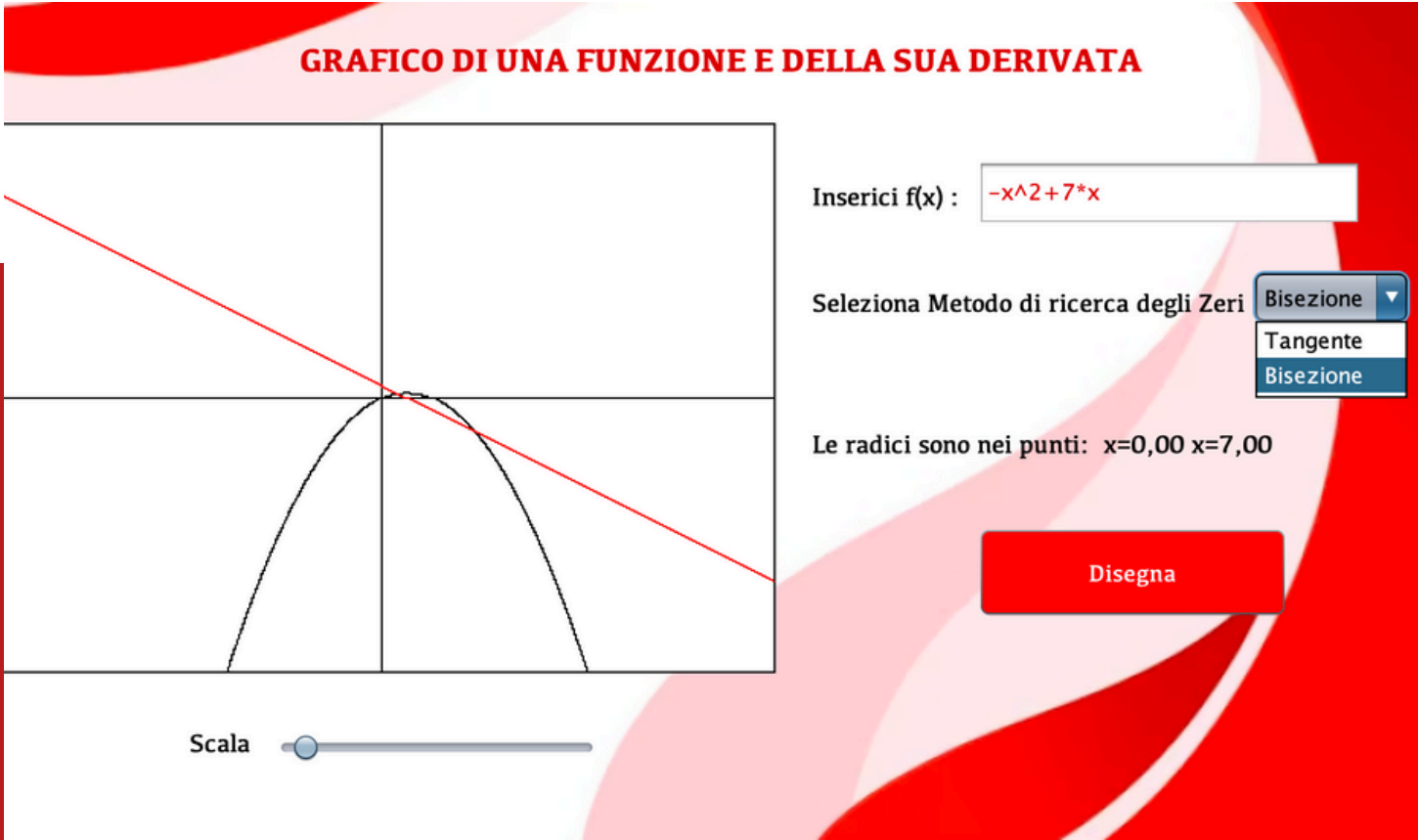
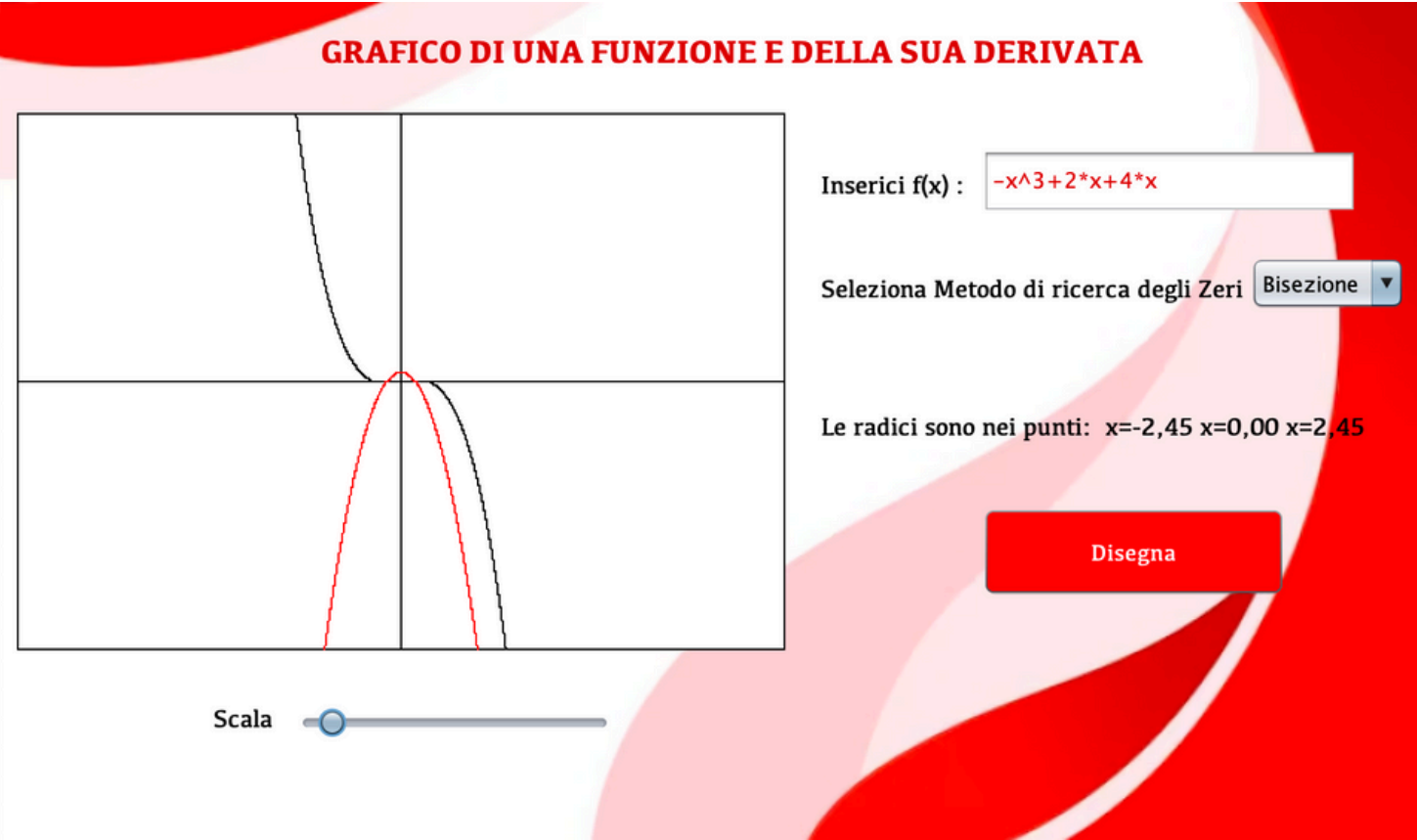
Durante la realizzazione di questo progetto ho sviluppato sia competenze tecniche che trasversali.

Dal punto di vista informatico, ho approfondito la programmazione in Java e in particolare la programmazione orientata agli oggetti, imparando a suddividere il codice in classi e metodi ben organizzati e riutilizzabili. Ho anche imparato a progettare un'interfaccia grafica semplice ma funzionale, che rappresentasse visivamente le funzioni e i risultati dei calcoli.

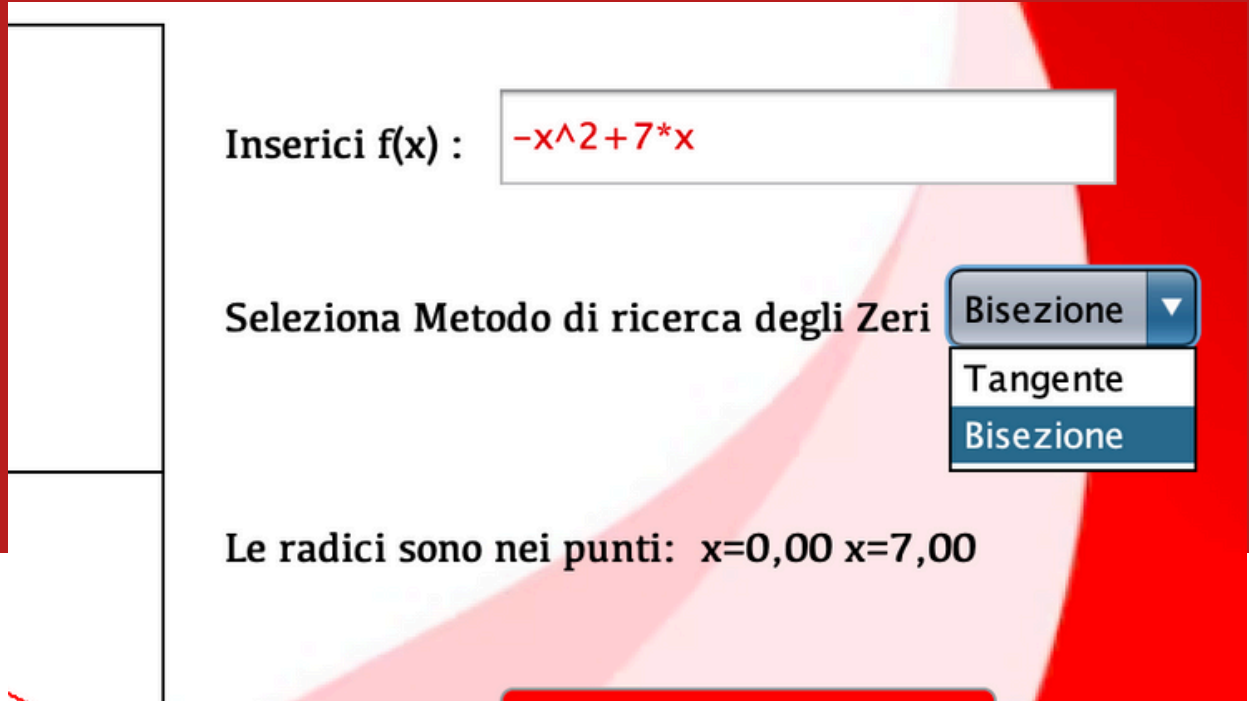
Sul piano matematico, ho applicato concetti come la continuità, la derivabilità e i metodi numerici per la risoluzione di equazioni.

Vedere come queste nozioni potessero essere tradotte in codice mi ha aiutato a comprenderle meglio e a vederne l'utilità pratica.

Inoltre, ho migliorato la mia capacità di problem solving, imparando a trovare soluzioni quando qualcosa non funzionava, a cercare errori nel codice e a ragionare in modo logico e struttura



Scelta del metodo, stesso obiettivo: ricerca Zeri



```
// Metodo delle tangenti per trovare uno zero della funzione
public double metodoTangenti(Function f, double a, double b, double precisione, double h2)
{
    if ((f.calculate(a) * f.calculate(b)) >= 0) { return 0; }
    double x0 = (a + b) / 2;
    boolean trovato = false;
    while (trovato==false) {
        // Calcola la derivata numerica (rapporto incrementale) in x0
        double m = (f.calculate(x0 + h2) - f.calculate(x0)) / h2;
        if (Math.abs(m) < precisione) {
            trovato = true;
            return m;
        } else {
            // Calcola la nuova approssimazione x1 usando il metodo di Newton
            double x1 = x0 - (f.calculate(x0) / m);
            // Controlla le condizioni di precisione per fermare il ciclo
            if (Math.abs(f.calculate(x1)) <= precisione || Math.abs(x1 - x0) < precisione)
                trovato = true;
            return x1;
        } else {
            // Aggiorna x0 per la prossima iterazione
            x0 = x1;
        }
    }
}
```

```
private double bisection(Function f, double a, double b, double precision)
{
    double p;
    while (Math.abs(b - a) > precision) {
        p = (a + b) / 2.0;
        double result = f.calculate(p);

        if (Math.abs(result) < precision) {
            return p;
        }

        if (f.calculate(a) * result < 0) {
            b = p;
        } else {
            a = p;
        }
    }
    return (a + b) / 2.0;
}
```

CONCLUSIONE

Questo progetto è stato per me molto più di un semplice compito scolastico. Mi ha dato la possibilità di mettere alla prova le mie conoscenze, ma soprattutto di trasformare teoria e formule in qualcosa di concreto e funzionante. È stato stimolante vedere come la matematica, spesso percepita come astratta, potesse diventare qualcosa di dinamico e visibile sullo schermo, grazie alla programmazione.

Realizzare questo capolavoro mi ha fatto capire quanto sia importante avere una visione interdisciplinare dello studio. Informatica e matematica, si rafforzano a vicenda: la programmazione rende la matematica più concreta, mentre la matematica fornisce alla programmazione un solido fondamento logico.