

Esercitazione N°4

Luca Zepponi

19 gennaio 2023

1. DESCRIZIONE DEL PROGRAMMA

Il programma contiene una classe che gestisce lo studio di funzione. Tale classe presenta le seguenti caratteristiche:

- possiede un costruttore che permette di inserire un puntatore ad una funzione matematica definita come funzione C++;
- un metodo per il calcolo della funzione in un punto inserito dall'utente;
- un metodo per calcolare la derivata prima della funzione in un punto inserito dall'utente;
- un metodo per il calcolo dell'integrale definito in un intervallo i cui estremi sono inseriti da tastiera;
- un metodo che stima il massimo della funzione e della derivata prima;
- un metodo per la rappresentazione grafica della funzione e della derivata prima.

L'applicazione realizzata insieme alla classe ha anche un menù che permette di scegliere all'utente varie opzioni:

- 1 - calcolare il valore della funzione definito in un punto;
- 2 - il valore della derivata prima in un punto;
- 3 - il valore dell'integrale definito su un intervallo contenuto nell'insieme di definizione della funzione;
- 4 - visualizzare la funzione;

5 - visualizzare la derivata prima;

0 - uscire dal programma.

All'interno della classe sono presenti tre attributi:

- `pFun pFunz;` contiene il puntatore della funzione;
- `float inferiore;` contiene l'estremo inferiore del dominio della funzione;
- `float superiore;` contiene l'estremo superiore della funzione.

Occorre notare che il tipo di dato `pfun` non è predefinito in C++, ma è stato definito utilizzando

```
using pFun = double (*)(double);.
```

La funzione da studiare è stata creata utilizzando una funzione C++ globale:

```
double funX(double x){ return 2 * x - 1 ; };
```

Le costanti che sono necessarie al corretto funzionamento del programma invece sono:

- `#define NUM_PARTIZIONE 1e5`: per partizionare l'intervallo di definizione;
- `#define ASSEY 40`: per stampare l'asse y .

Le librerie utilizzate sono:

- `<iostream>`: per gli input e gli output;
- `<cmath>`: per utilizzare funzioni matematiche per definire la funzione.

I vari output sono stati messi in modo da obbligare il testo ad andare a capo prima di svilupparsi troppo in lunghezza.

1.1. DESCRIZIONE DELLE SINGOLE FUNZIONI

Il programma riportato nella sezione 2 inizia creando un oggetto della classe **funzione** prendendo in input il nome della funzione **funX** e i due estremi del dominio.

Vengono poi dichiarate le variabili che servono al **main** che verranno usate nel ciclo **do-while** che segue. È stato scelto di usare tale ciclo per poter eseguire almeno una volta tutti i codici in esso contenuto e, nel caso la condizione del **while** dovesse risultare vera, poterlo rieseguire all'occorrenza.

Dopo la stampa del menù viene chiesto all'utente di scegliere l'opzione da eseguire e viene fatto un controllo sul carattere in modo da essere sicuri che esista una scelta corrispondente. In caso in cui il carattere inserito non è fra quelli presenti nel menù, viene lanciato un messaggio di errore e si obbliga a rimettere un altro carattere. La variabile **scelta** è di tipo **char** per permettere di escludere anche di eventuali lettere e non limitare il controllo ai soli numeri.

Se la lettera inserita è una di quelle consentite, il programma entra nel corrispondente **case** dello **switch**. Ognuno dei quali si occupa di eseguire l'opzione corrispondente del menù. Ciascun **case** viene spiegato durante la sua esecuzione con un messaggio stampato sul terminale. L'unica attenzione va data al **case '3'** perché esso richiede l'inserimento da tastiera degli estremi di integrazione e controlla che l'estremo inferiore abbia un valore minore dell'estremo superiore, in caso contrario viene stampato un messaggio che richiede un nuovo inserimento degli estremi.

Il metodo **incremento** si occupa di calcolare il valore dell'incremento che verrà poi usato nei metodi successivi. Tale metodo non prende in input nulla e restituisce un **double**.

Il metodo **valore** si occupa di calcolare il valore della funzione nel punto che riceve in input. Se tale valore dovesse essere esterno all'intervallo di definizione, viene restituito un messaggio e il valore 0.

Il metodo **derivataPunto** prende in input un **double** e, dopo aver verificato che tale numero appartenga all'intervallo di definizione della funzione, valuta la derivata prima nel punto inserito. Come per il metodo **valore**, il metodo **derivataPunto** restituisce un messaggio e il valore 0 se il punto inserito non appartiene all'intervallo di definizione. Per il calcolo della derivata è stato utilizzato il rapporto incrementale [1]

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

dove h è l'incremento della funzione.

Il metodo **integrale** prende in input due valori **double** che servono come estremi di integrazione. Se, per esempio, l'estremo superiore di integrazione è maggiore dell'estremo superiore del dominio, allora l'estremo superiore di integrazione viene riassegnato e prende il valore dell'altro estremo. Una cosa analoga viene fatta con l'estremo inferiore. Una volta assicurati che l'integrale venga calcolato solo nei punti in cui la funzione è definita, si procede al suo calcolo tramite la regola dei trapezi. È stato suddiviso l'asse delle x in un numero sufficientemente alto di intervalli

$$x_{\text{inf}} = x_0 < x_1 < \dots < x_k = x_{\text{sup}}$$

in modo da approssimare localmente la funzione ad una funzione lineare in modo da demandare il calcolo dell'integrale alla più semplice somma di aree di trapezi. Si ottiene quindi la formula [2]

$$\int_a^b f(x) \, dx \approx \sum_{i=0}^k \frac{(f(i) + f(i+h)) \cdot h}{2},$$

dove

- h è l'altezza del trapezio: la partizione dell'asse x è omogenea, quindi ciascun sottointervallo $[x_i, x_{i+1}]$ ha sempre lunghezza pari ad h ;
- $f(i)$ e $f(i+h)$ sono le basi maggiori e minori del trapezio.

Il metodo **massimo** si occupa di calcolare il massimo della funzione o della sua derivata prima. Tale massimo in realtà non è il reale massimo assoluto, ma solo un'approssimazione che servirà al metodo **disegnaFunzione**. Per evitare di riscrivere due metodi simili, **massimo** prende in input un dato di tipo **int** che non viene dato dall'utente. Se l'intero in input è 1, si ricerca il massimo della retta, mentre se è 2, si cerca il massimo della derivata prima. Tali interi vengono passati in input nel metodo **disegnaFunzione** richiamato nel **main**.

Per scrivere il metodo **disegnaFunzione** è stata utilizzata la logica del metodo **disegnaFunzione** scritto nell'Esercitazione N°2. Successivamente è stato modificato il tipo di dato in input utile all'assegnamento a **yFun** di **funX** o **derivataPunto** e parte della sintassi nel disegno del punto della funzione. Esiste un'ulteriore differenza che merita una spiegazione un po' più dettagliata. Per la normalizzazione dell'asse y è stato necessario dividere per

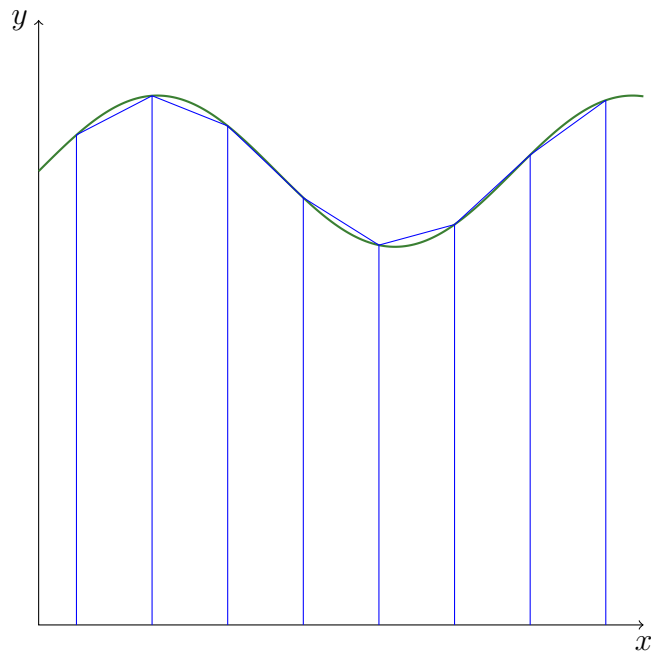


Figura 1: Regola del trapezio.

il massimo valore che la funzione assume. Questo causa una rappresentazione errata del suo grafico nel caso in cui la funzione inserita sia identicamente nulla o, nel grafico della derivata, se la funzione è costante. Una volta notato questo è stato aggiunto nel metodo la seguenti linea di codice:

```
else if ((max == 0) && (y == ASSEY/2)) cout << '*' ;.
```

In questo modo, anche se la funzione (o la sua derivata) è identicamente nulla, viene stampato il grafico in modo corretto.

2. CODICE SORGENTE

```
1 // Esercitazione 4
2 // Studente: Luca Zepponi
3 //
4 // Data una funzione predefinita, il programma calcola:
5 // - il valore della funzione in un punto;
6 // - il valore della derivata in un punto;
7 // - l'integrale definito in un intervallo della funzione utilizzando il
8 // metodo dei trapezi;
```

```

9 // Inoltre , vengono anche fornite le rappresentazioni grafiche normalizzate
10 // rispetto l'asse y della funzione e della sua derivata prima.
11 //
12
13 #include <iostream>
14 #include <cmath>
15
16 using namespace std;
17
18 using pFun = double (*)(double);
19
20 #define NUM_PARTIZIONE 1e5
21 #define ASSEY 40 // Doppio dell'ampiezza asse y
22
23 // Definizione funzione
24 double funX(double x){ return 2 * x - 1 ; };
25
26
27
28 // Definizione classe "funzione"
29 class funzione{
30     pFun pFunz;
31     float inferiore ;
32     float superiore;
33
34 public:
35     //costruttore
36     funzione(pFun fun, float i, float s):pFunz(fun),superiore(s), inferiore (i){};
37
38     double incremento();
39
40     // metodi per il calcolo
41     double valore(double x);
42     double derivataPunto(double x);
43     double integrale(double i, double s);
44
45     // Metodo per il massimo
46     double massimo(int i);
47
48     // Stampa grafico funzione
49     void disegnaFunzione(int i);
50 };
51

```

```

52
53
54 int main(){
55     setlocale (LC_ALL, "");
56
57     // Definisco fun1
58     funzione fun1(funX, -24, 24);
59
60     // Dichiarazione variabili
61     // Variabile per il menù
62     char scelta;
63     // Variabile per immagazzinare i punti
64     double punto;
65     // Variabili per gli estremi dell'integrale
66     double eInf = 0.0;
67     double eSup = 0.0;
68     int ordine = 0;
69
70     // menù
71     do {
72         cout << "%-----%" << endl;
73         cout << "menù:" << endl;
74         cout << "1_ _Valore_della_funzione_in_un_punto;" << endl;
75         cout << "2_ _Valore_della_derivata_in_un_punto;" << endl;
76         cout << "3_ _Valore_dell'integrale_definito_in_un_intervallo;" << endl;
77         cout << "4_ _Visualizzazione_grafica_della_funzione;" << endl;
78         cout << "5_ _Visualizzazione_grafica_della_derivata;" << endl;
79         cout << "0_ _Esci." << endl;
80         cout << "%-----%" << endl;
81
82         cout << "\n";
83
84         cout << "Inserisci_la_scelta:_";
85         cin >> scelta;
86
87         while ((scelta != '0') && (scelta != '1') && (scelta != '2') &&
88             (scelta != '3') && (scelta != '4') && (scelta != '5')) {
89             cout << "ATTENZIONE!" << endl;
90             cout << "Scelta_non_valida,_riprovare." << endl;
91
92             cout << "\n";
93
94             cout << "Inserisci_la_scelta:_";

```

```

95     cin >> scelta;
96 }
97
98 // Opzione
99 switch (scelta) {
100     case '1':
101         cout << "Inserisci il punto dove vuoi calcolare la funzione: x=";
102         cin >> punto;
103         cout << "Il valore della funzione nel punto di ascissa" << endl;
104         cout << "\tx=" << punto << endl;
105         cout << "è" << endl;
106         cout << "\tf(x)=" << fun1.valore(punto) << "." << endl;
107         cout << "\n";
108     break;
109     case '2':
110         cout << "Inserisci il punto dove vuoi calcolare la derivata prima:";
111         cout << "x=";
112         cin >> punto;
113         cout << "Il valore della derivata prima nel punto di ascissa" << endl;
114         cout << "\tx=" << punto << endl;
115         cout << "è" << endl;
116         cout << "\tf'(x)=" << fun1.derivataPunto(punto) << "." << endl;
117         cout << "\n";
118     break;
119     case '3':
120         do {
121             cout << "Inserisci estremo inferiore:";
122             cin >> eInf;
123             cout << "Inserisci estremo superiore:";
124             cin >> eSup;
125
126             if (eInf > eSup) {
127                 cout << "Attenzione!" << endl;
128                 cout << "L'estremo inferiore è maggiore dell'estremo superiore.\n";
129                 ordine = 0;
130             } else ordine = 1;
131         } while (ordine == 0);
132         cout << "L'integrale è " << fun1.integrale(eInf, eSup) << endl;
133         cout << "\n";
134     break;
135     case '4':
136         cout << "Il grafico della funzione è:" << endl;
137         fun1.disegnaFunzione(1);

```



```

138         cout << "\n";
139     break;
140     case '5':
141         cout << "Il grafico della derivata prima della funzione è:" << endl;
142         fun1.disegnaFunzione(2);
143         cout << "\n";
144     break;
145     case '0':
146         cout << "Termino rogramma" << endl;
147     break;
148     default:
149         cout << "Nessuna scelta disponibile." << endl;
150         cout << "\n";
151     break;
152 }
153 } while (scelta != '0');
154 }
155
156
157
158 // Metodo per il calcolo dell'incremento
159 double funzione::incremento(){
160     return (superiore - inferiore)/(NUM_PARTIZIONE);
161 }
162
163
164
165 // Metodo per il calcolo del valore della funzione nel punto x
166 double funzione::valore(double x){
167     // Controllo punto in intervallo
168     if ((x > inferiore) && (x < superiore)) return (pFunz(x));
169     else {
170         cout << "Il punto non appartiene all'intervallo inserito." << endl;
171         return 0;
172     }
173 }
174
175
176
177 // metodo per il calcolo della derivata nel punto x
178 double funzione::derivataPunto(double x){
179     // Calcolo incremento inc
180     double inc = incremento();

```

```

181
182 // Controllo punto in intervallo
183 // Restituisco valore derivata
184 if((x>=inferiore) && (x<=superiore)) return((funX(x + inc)-funX(x))/inc);
185 else {
186     cout << "Il_punto_non_appartiene_all'intervallo_inserito." << endl;
187     // Derivata nulla
188     return 0;
189 }
190 }
191
192
193
194 // Metodo per il calcolo dell'integrale definito della funzione nell'intervallo
195 // [i, s]
196 double funzione::integrale(double i, double s){
197     // Inizializzazione variabile
198     double inc = incremento();
199     double integ = 0.0;
200
201     // Controllo estremi di integrazione
202     if (s > superiore) s = superiore;
203     if (i < inferiore) i = inferiore;
204
205     // Calcolo valore integrale
206     while (i < s) {
207         // Incremento variabile integ
208         integ += ((funX(i + inc) + funX(i)) * inc)/2;
209         // i -> s
210         i += inc;
211     }
212     return integ;
213 }
214
215
216
217 // Metodo per il calcolo del massimo
218 double funzione::massimo(int i){
219     double funzValutata;
220
221     // La funzione va in moduo per non dover calcolare anche il minimo
222     double max = 0;
223

```

```

224 // Cerco il massimo della funzione in modulo
225 for (double t = inferiore; t <= superiore; t++) {
226     if (i == 1) funzValutata = abs(funX(t));
227     else if (i == 2) funzValutata = abs(derivataPunto(t));
228     if (funzValutata > max) max = funzValutata;
229 }
230 return max;
231 }
232
233
234
235 // Disegna grafico della funzione
236 void funzione::disegnaFunzione(int i) {
237     double yFun;
238
239     // Calcolo massimo funzione
240     double max = massimo(i);
241
242     // Mi sposto sull'asse y
243     for (int y = ASSEY+2; y >= 0; y--)
244     {
245         // Mi sposto sull'asse x
246         for (double x = round(inferiore); x <= round(superiore); x++)
247         {
248             // funX o derivata?
249             if (i == 1) yFun = funX(x);
250             else if (i == 2) yFun = derivataPunto(x);
251
252             // Disegna punto funzione
253             if ((max != 0) &&
254                 (y == (round(yFun/max*(ASSEY/2)) + ASSEY/2))) cout << '*';
255             // se si trova sull'asse delle y a quota ASSEY+2, inserisci y
256             else if ((x == 0) && (y == (ASSEY + 2))) cout << 'y';
257             // se si trova sull'asse delle y a quota ASSEY+1, inserisci "^"
258             else if ((x == 0) && (y == (ASSEY + 1))) cout << "^";
259             // se si trova alla fine dell'asse delle x, inserisci -> x
260             else if ((y == ASSEY/2) && (x == round(superiore))) cout << ">x";
261             // Se max == 0, la funzione è identicamente nulla
262             else if ((max == 0) && (y == ASSEY/2)) cout << '*';
263             // se si trova al centro del grafico, inserisci +
264             else if ((y == ASSEY/2) && (x == 0)) cout << '+';
265             // se si trova sull'asse delle x, inserisci -
266             else if (y == ASSEY/2) cout << '-';

```

```

267     // se si trova sull'asse delle y, inserisci |
268     else if (x == 0) cout << '|';
269     // Se non c'è nulla da rappresentare, inserisci spazio vuoto
270     else cout << ' ';
271 }
272 // Fine y-esima riga
273 cout << '\n';
274 }
275 }

```

3. RISULTATO

Di seguito vengono riportate le schermate della console con tre esempi di funzionamento, rispettivamente con le funzioni

$$f(x) = 2x - 1, \quad (1)$$

$$f(x) = 1, \quad (2)$$

$$f(x) = x^2. \quad (3)$$

```

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 1
Inserisci il punto dove vuoi calcolare la funzione: x = 1
Il valore della funzione nel punto di ascissa
    x = 1
è
    f(x) = 1.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 2
Inserisci il punto dove vuoi calcolare la derivata prima: x = 2
Il valore della derivata prima nel punto di ascissa
    x = 2
è
    f'(x) = 2.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 3
Inserisci estremo inferiore: 0
Inserisci estremo superiore: 1
L'integrale è 0.000320102
%-----%

```

Figura 2: Schermata della consolle con la funzione (1).


```
%-----%  
Menu:  
1 - Valore della funzione in un punto;  
2 - Valore della derivata in un punto;  
3 - Valore dell'integrale definito in un intervallo;  
4 - Visualizzazione grafica della funzione;  
5 - Visualizzazione grafica della derivata;  
0 - Esci.  
%-----%  
  
Inserisci la scelta: 0  
Termine programma.
```

Figura 5: Schermata della console con la funzione (1).


```

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 1
Inserisci il punto dove vuoi calcolare la funzione: x = 9
Il valore della funzione nel punto di ascissa
      x = 9
è
      f(x) = 1.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 2
Inserisci il punto dove vuoi calcolare la derivata prima: x = -4
Il valore della derivata prima nel punto di ascissa
      x = -4
è
      f'(x) = 0.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 3
Inserisci estremo inferiore: -8
Inserisci estremo superiore: 7
L'integrale è 15.0005

```

Figura 6: Schermata della consolle con la funzione (2).

[illegible]

Figura 7: Schermata della consolle con la funzione (2).


```
%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 0
Termine programma.
```

Figura 9: Schermata della consolle con la funzione (2).

```

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 1
Inserisci il punto dove vuoi calcolare la funzione: x = -99
Il valore della funzione nel punto di ascissa
    x = -99
è
    f(x) = Il punto non appartiene all'intervallo inserito.
0.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 1
Inserisci il punto dove vuoi calcolare la funzione: x = 2
Il valore della funzione nel punto di ascissa
    x = 2
è
    f(x) = 4.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 2
Inserisci il punto dove vuoi calcolare la derivata prima: x = 70
Il valore della derivata prima nel punto di ascissa
    x = 70
è
    f'(x) = Il punto non appartiene all'intervallo inserito.
0.

```

Figura 10: Schermata della console con la funzione (3).

```

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 2
Inserisci il punto dove vuoi calcolare la derivata prima: x = 7
Il valore della derivata prima nel punto di ascissa
      x = 7
è
      f' (x) = 14.0005.

%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 3
Inserisci estremo inferiore: 17
Inserisci estremo superiore: 2
Attenzione!
L'estremo inferiore è maggiore dell'estremo superiore.
Inserisci estremo inferiore: 2
Inserisci estremo superiore: 17
L'integrale è 1635.14

```

Figura 11: Schermata della consolle con la funzione (3).

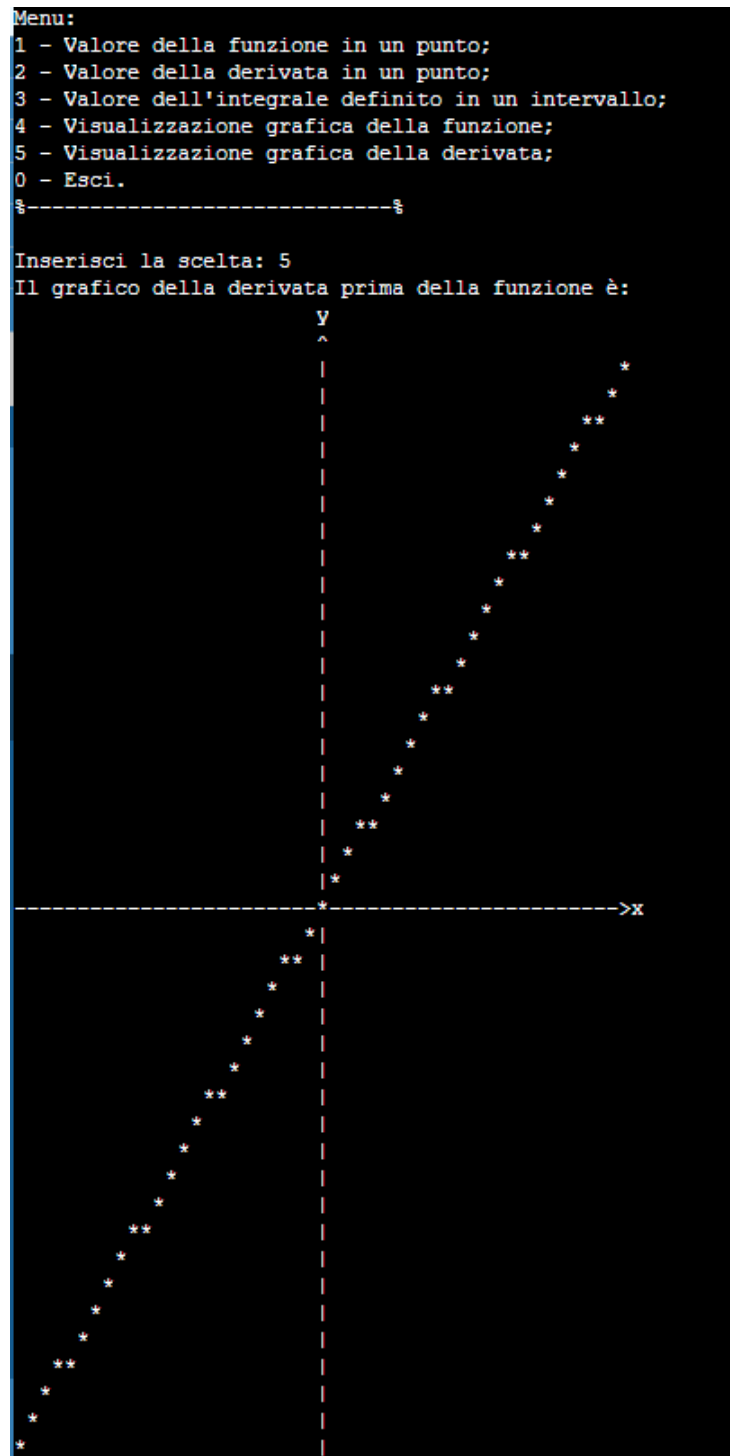


Figura 13: Schermata della consolle con la funzione (3).


```
%-----%
Menu:
1 - Valore della funzione in un punto;
2 - Valore della derivata in un punto;
3 - Valore dell'integrale definito in un intervallo;
4 - Visualizzazione grafica della funzione;
5 - Visualizzazione grafica della derivata;
0 - Esci.
%-----%

Inserisci la scelta: 0
Termine programma.
```

Figura 14: Schermata della consolle con la funzione (3).

4. OSSERVAZIONI E CONCLUSIONI

1. Una prima miglione che si potrebbe fare è inserire la possibilità di visualizzare a schermo la funzione che si vuole studiare e il suo dominio.
2. Come spiegato anche sopra, il metodo per la ricerca del massimo non è esatta e, in alcune situazioni particolarmente patologiche, potrebbe portare ad una visualizzazione non perfettamente normalizzata.
3. un altro possibile miglioramento si può avere nella definizione dell'estremo superiore e inferiore del dominio della funzione. Nel codice riportato sopra, essi vengono definiti quando viene creato l'oggetto `fun1`. Si potrebbero implementare metodi per la lettura e la modifica di tali valori.
4. Come mostrato nella foto 11 a pagina 22, se si inserisce un punto da tastiera non appartenente al dominio della funzione, il messaggio di errore presenta delle carenze estetiche.

5. RIFERIMENTI

- [1] Wikipedia. *Derivata* — *Wikipedia, L'enciclopedia libera*. URL: `%5Curl%7B//it.wikipedia.org/w/index.php?title=Derivata&oldid=130407292%7D`. (ultimo accesso: 19/01/2023).
- [2] Wikipedia. *Regola del trapezio* — *Wikipedia, L'enciclopedia libera*. URL: `%5Curl%7B//it.wikipedia.org/w/index.php?title=Regola_del_trapezio&oldid=130398155%7D`. (ultimo accesso: 19/01/2023).