

Laboratorio di Calcolo, Canale Pet-Z

Prova valutata 09/01/2025

Nome: _____ Cognome: _____
Matricola: _____ ☐ Ritirata/o

Lo scopo della prova è scrivere un programma per calcolare l'area sottostante una funzione con il metodo *hit & miss*.

1. Il tempo a disposizione è di 3 ore. Sono ammessi libri di testo, prontuari, appunti. Non si può parlare con nessuno, utilizzare cellulari/tablet/laptop, pena l'annullamento del compito.
2. Il programma va scritto e salvato esclusivamente sul computer del laboratorio, a cui si deve accedere utilizzando come username **studente** e come password **informatica**
3. **Tutti i file vanno salvati in una cartella chiamata LCSR11_NOME_COGNOME nella home directory**, dove NOME e COGNOME indicano rispettivamente il tuo nome e cognome. Ad esempio lo studente *Nicolò De Rossi* deve creare una cartella chiamata LCSR11_NICOLO_DEROSSII contenente tutti i file specificati nel testo. **Tutto ciò che non si trova all'interno della cartella suddetta non verrà valutato.**
4. Consegnare il presente testo indicando nome, cognome e numero di matricola (vedi sopra), barrando la casella "Ritirata/o" se ci si vuole ritirare.

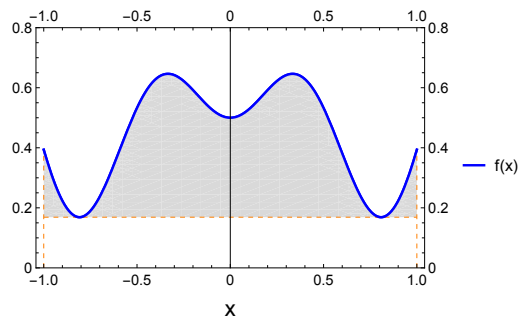
► **Esercizio in C:** Si vuole scrivere un programma `hitmiss.c` che utilizzi il metodo *hit & miss* per calcolare l'area sottostante la funzione

$$f(x) = \sin(3x) \cos(3x) \tanh(x) + 0.5 \quad (1)$$

nell'intervallo $[-1, 1]$. Siano y_{\min} e y_{\max} rispettivamente i valori minimo e massimo che tale funzione assume nell'intervallo considerato. Si scriva un programma per calcolare, l'area A della superficie compresa tra la funzione $f(x)$ e la retta $y = y_{\min}$ nell'intervallo $[-1, 1]$.

Per calcolare l'area A , bisogna generare casualmente in modo uniforme N_p punti nel rettangolo individuato dai punti $(-1, y_{\min})$, $(-1, y_{\max})$, $(1, y_{\max})$ e $(1, y_{\min})$ e contare il numero di volte N_h che tali punti si trovano nell'area grigia mostrata in figura. Una stima dell'area A sarà fornita dalla seguente formula:

$$A = 2(y_{\max} - y_{\min})N_h/N_p \quad (2)$$



Suggerimento: in C la funzione per calcolare $\tanh(x)$ è `tanh()`, definita in `math.h`.

Il programma dovrà:

1. Definire tramite una direttiva del precompilatore il numero di punti $N_f = 10000$ per i quali calcolare la funzione $f(x)$ e il numero $N_p = 100000$ (10^5) di punti da generare.
2. Identificare numericamente il valore massimo e minimo della funzione nell'intervallo nel modo seguente:
 - Calcolare il valore della funzione $f(x)$ in N_f punti nell'intervallo $[-1, 1]$ e memorizzare questi valori in array bidimensionale chiamato `valori` e costituito da N_f righe e 2 colonne, dove nella prima colonna verranno memorizzati gli N_f valori di x generati e nella seconda colonna i corrispondenti valori della funzione $f(x)$.
 - Utilizzando i dati contenuti nell'array `valori`, stimare il valore minimo y_{\min} e massimo y_{\max} assunto dalla funzione $f(x)$ nell'intervallo $[-1, 1]$.
3. Calcolare l'area A con il metodo *hit & miss*, ovvero:
 - Generare a caso N_p di punti all'interno del rettangolo suddetto.
 - Contare il numero di volte N_h in cui tali punti cadono all'interno dell'area grigia mostrata in figura.
 - Calcolare l'area A tramite l'eq. (2).
4. Stampare su schermo il valore dell'area A così ottenuto con 5 cifre dopo la virgola, per $N_p = 10^5$.

Nello scrivere il programma si richiede che vengano implementate le seguenti funzioni:

- `func(...)` che restituisce il valore della funzione $f(x)$ calcolata per il valore passato come argomento
- `riempi(...)`, che richiede come argomento l'array `valori` e memorizza in tale array N_f valori della funzione $f(x)$ nell'intervallo $[-1, 1]$ come spiegato in precedenza.
- `maxMin(...)` che richiede come argomento l'array `valori` e determina il valore massimo e minimo della funzione $f(x)$ utilizzando tale array. La funzione deve utilizzare due puntatori passati come argomenti per restituire i valori del massimo e del minimo.
- `integrale(...)`, che calcola l'area A con il metodo *hit & miss* illustrato in precedenza. Tale funzione prende come argomenti i valori y_{\min} e y_{\max} calcolati in precedenza con la funzione `maxMin()`.

Una volta verificato il corretto funzionamento del programma, modificarlo come segue: invece di fissare il valore di `Np` tramite il pre-compilatore, definire una variabile in modo che il programma calcoli l'integrale per valori `Np` pari a 2^{10} , 2^{11} , ..., 2^{20} , e scriva in un file di nome `integrale.dat` il numero di punti generati e il corrispondente valore dell'area A ottenuto. Quest'ultimo va scritto con quattro cifre dopo la virgola. Ci devono quindi essere due valori per ciascuna riga del file.

► **Esercizio in Python:** Creare uno script python chiamato `area.py` per leggere i dati dal file `integrale.dat`, e riporti su un grafico l'andamento dell'area A in funzione del numero di punti. Il grafico dovrà riportare una legenda ed opportuni label per gli assi e deve essere salvato con il nome di `area.png`