

Lorenzo Rossi, matr. 20031485  
Università degli Studi del Piemonte Orientale  
Dipartimento di Scienze ed Innovazione Tecnologica  
Corso di Laurea Magistrale in Informatica  
Anno Accademico 2020/2021

## **Relazione per il progetto del corso di Laboratorio di Calcolo**

Luglio 2021

# 1 Introduzione

Si vuole realizzare un programma in linguaggio **C** con cui valutare il calcolo di un integrale definito di una funzione mediante la generazione di numeri casuali. Il calcolo del valore dell'integrale tramite metodo Monte Carlo avviene attraverso la stima delle aree: vengono generate  $n$  coppie di valori casuali  $(x, y)$  all'interno di un rettangolo che ha come base l'intervallo di integrazione e come altezza la distanza tra  $y = 0$  ed il massimo dell'integranda nell'intervallo di integrazione. Conoscendo il valore dell'area del rettangolo ed il numero di punti che cadono tra il grafico dell'integranda e l'asse delle ascisse, è possibile stimare il valore dell'integrale definito con una proporzione.

Una volta scelto un dominio  $[a, b]$  ed una funzione limitata in  $[a, b]$ , è possibile stimarne l'integrale definito in quel dominio sapendo che:

$$\int_a^b f(x) dx \approx \frac{N_{acc}}{N_{tot}} M(b - a)$$

dove  $N_{acc}$  è il numero di punti generati randomicamente che cadono sotto al grafico di  $f(x)$  e che, quindi, vengono “accettati” (come suggerisce il pedice), mentre  $N_{tot}$  è il numero totale di punti generati.  $M$  è il massimo valore che  $f(x)$  assume in  $[a, b]$ , ed infine  $M(b - a)$  è l'area del rettangolo in cui vengono generati casualmente i punti per valutare l'integrale.

Si è scelto di valutare l'integrale definito della funzione

$$f(x) = |\sin(x)|$$

## 2 Struttura del programma

### 2.1 Funzioni

**double\* setRange():** Questa funzione permette all'utente di impostare il dominio di integrazione. L'utente può scegliere un qualsiasi intervallo all'interno di (0, 10) (estremi esclusi). La funzione restituisce un array bidimensionale che contiene in posizione 0 l'estremo inferiore ed in posizione 1 l'estremo superiore dell'intervallo di integrazione impostato.

**void setSeed():** Questa funzione non prevede interazioni con l'utente e viene chiamata prima di generare valori casuali tramite la funzione **rand()**. Il suo unico scopo è quello di impostare il *seed* per la generazione di numeri casuali, restituito tramite una chiamata interna alla funzione **time()**, che estrae un seme basandosi sulla data e sull'ora della macchina su cui il programma esegue (più precisamente sulla distanza temporale in secondi dalla mezzanotte del 1° gennaio 1970, detta *epoch* nei sistemi Unix-like). Utilizzando la funzione **time()** come seed ed essendo il programma di impossibile utilizzo multiplo in meno di un secondo, è ragionevole pensare che due esecuzioni consecutive dello stesso programma si basino su due seed differenti.

**size\_t setN():** Questa funzione permette all'utente di scegliere il numero intero di punti da generare all'interno dell'area del rettangolo menzionata nell'Introduzione 1, restituendolo per poterlo utilizzare all'interno del programma. Un numero maggiore di punti generati casualmente permette una valutazione più accurata dell'integrale.

**double getRandomDoubleFromRange(double min, double max):** Questa funzione viene chiamata quando sia necessario generare un valore reale (rappresentato come **double**) all'interno di un range, definito dall'utente od opportunamente calcolato. Restituisce il **double** estratto casualmente.

**double f(double x, double\* y\_max):** Questa funzione può rappresentare una qualsiasi funzione matematica ad una variabile. Dato il parametro **double x**, restituisce il valore della funzione di cui è fornita l'implementazione all'interno dello scope stesso di **f()**. Il secondo parametro è un puntatore a **double** che viene passato per riferimento per poter aggiornare, ad ogni valutazione di  $f(x)$ , quale sia il massimo di  $f(x)$  raggiunto finora. Tale valore sarà utilizzato come altezza del rettangolo in cui è contenuto il grafico di  $f(x)$  per poter stimare accuratamente l'integrale definito.

## 2.2 Flowchart

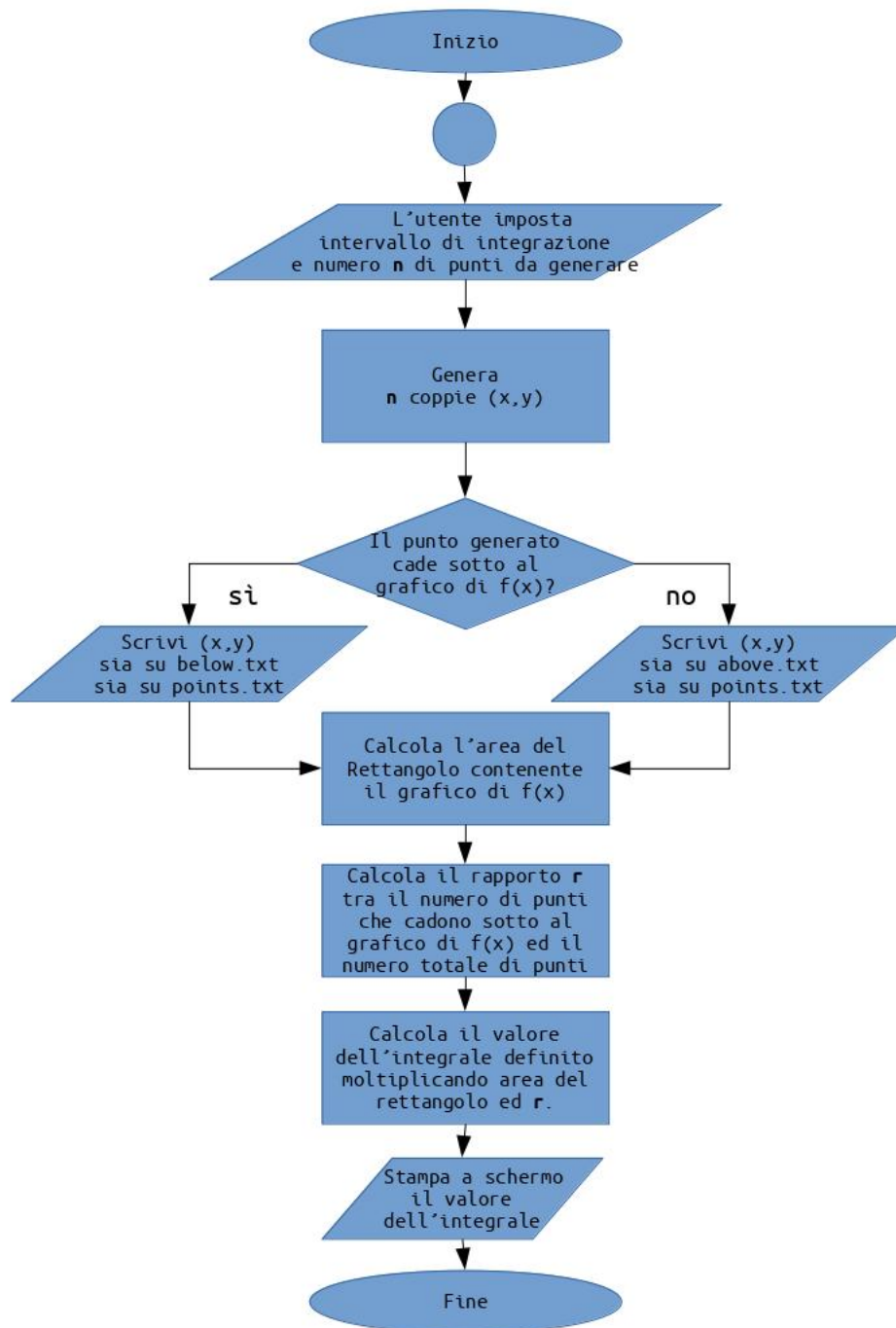


Figura 1: Flowchart del programma

## 2.3 Compilazione

Dato che il programma si serve della libreria `math.h` per poter chiamare funzioni matematiche (tra cui `sin()` e `fabs()`, che sono state utilizzate nell'implementazione), è necessario compilare il codice istruendo il linker con l'attributo `-lm`, lanciando dunque un comando seguente da terminale:

```
gcc "sorgente" -lm -o "eseguibile".
```

## 3 Risultati

Per verificare il corretto funzionamento del programma, sono stati fatti due esperimenti. In entrambi sono stati mantenuti fissi l'intervallo di integrazione  $[0.001, 9.999]$  e la variabilità del seed per la generazione di numeri pseudocasuali.

Per il **primo esperimento** sono stati utilizzati tre diversi valori di  $n$ , rispettivamente  $10^3$ ,  $10^4$  e  $10^5$ . I rispettivi grafici disegnati con **GNUPlot** si possono apprezzare nelle figure 2, 3 e 4. I grafici sono stati realizzati singolarmente con il comando `plot [0:10] 'above.txt' with points, 'below.txt' with points, abs(sin(x))`.

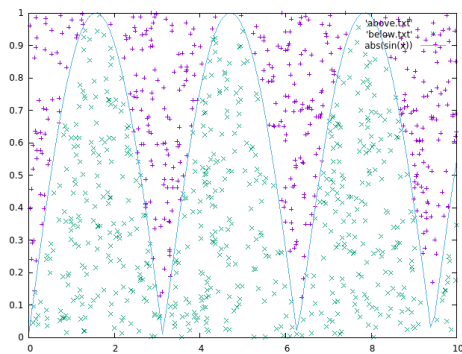


Figura 2:  $n = 10^3$

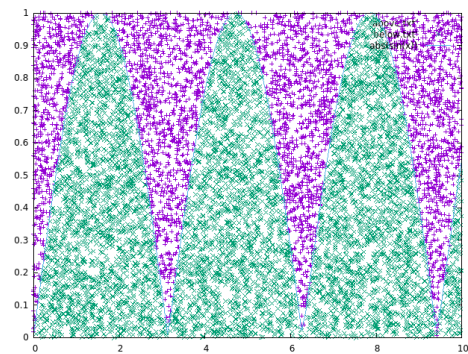


Figura 3:  $n = 10^4$

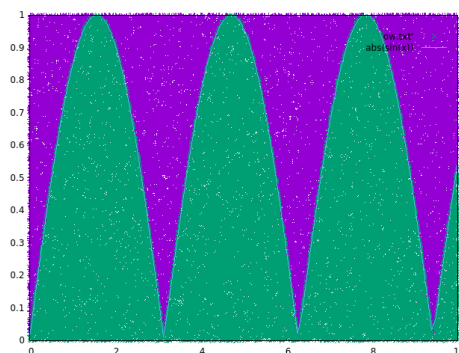


Figura 4:  $n = 10^5$ .

In tutte e tre le immagini, i punti verdi sono evidentemente quelli “accettati” che cadono sotto al grafico di  $f(x)$ , raccolti durante l’esecuzione del programma nel file `below.txt`. I punti viola, invece, sono raccolti nel file `above.txt` e, sommati ai punti verdi, rappresentano la totalità dei punti generati, anch’essi raccolti nel file `points_a_xmin_b_xmax_n_N.txt` (dove `xmin`, `xmax` e `N` sono i valori scelti dall’utente rispettivamente per  $a$ ,  $b$  ed  $n$ ). Si è scelto di scrivere i punti estratti su più files per una visualizzazione più chiara dei punti “accettati”, rispetto ai punti totali, nei grafici GNUPlot.

Da tre iterazioni successive con  $n$  ogni volta incrementato di un fattore 10, si ottengono i seguenti valori per l’integrale definito:

$n = 10^3$	$n = 10^4$	$n = 10^5$	“valore esatto”*
5.928793	6.257748	6.167066	6.16038

Si nota come, con l’aumentare del numero di punti estratti, diminuisca lo scarto tra il valore esatto dell’integrale e quello stimato mediante metodo Monte Carlo.

Per il **secondo esperimento**, invece, è stato calcolato 5 volte l’integrale mantenendo fisso  $n = 10^5$  e sono stati ottenuti i seguenti risultati:

1	2	3	4	5	“valore esatto”*
6.170866	6.151769	6.187762	6.163167	6.129074	6.16038

I valori dell’integrale così calcolati portano ad una deviazione standard  $\sigma = 0.01960409915$ , grazie a cui è possibile verificare che i valori stimati sono tutti ragionevoli. Trattando  $\sigma$  come incertezza sul singolo valore, si ha infatti che il valore esatto cade all’interno di ognuno degli intervalli di incertezza dei valori calcolati.

(\*calcolato con **WolframAlpha**).