

Laboratorio di Calcolo per Fisici, Nona esercitazione

Lo scopo della nona esercitazione di laboratorio è di fare pratica con gli argomenti appresi durante il corso in vista dell'esercitazione valutata. Vi consigliamo di utilizzare tale esercitazione per simulare un'esercitazione valutata, quindi usate libri di testo e appunti, ma lavorate autonomamente. L'esercizio consiste nello scrivere un programma che utilizzi il metodo Monte Carlo per calcolare l'area ottenuta come l'intersezione di due ellissi che sono l'una perpendicolare all'altra.

► Prima parte:

Si scriva un programma chiamato `hitmiss.c` per calcolare con il metodo Monte Carlo l'area A della superficie che si ottiene considerando l'intersezione delle superfici delle due ellissi individuate dalle seguenti equazioni:

$$\frac{x^2}{4} + y^2 = 1 \quad (1)$$

$$x^2 + \frac{y^2}{4} = 1 \quad (2)$$

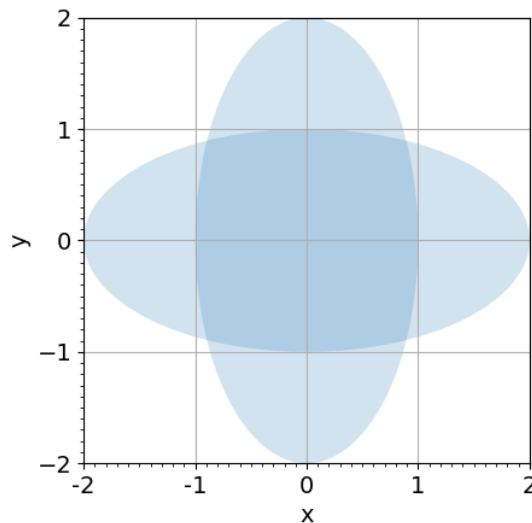


Figura 1: Grafico delle superfici delle due ellissi individuate dalle equazioni (1) e (2). La superficie che si ottiene come intersezione delle loro superfici coincide con la regione più scura.

Notare che le due ellissi hanno entrambe semiassi pari a 2 e 1 e sono ruotate una rispetto all'altra di 90 gradi come mostrato in Fig. 1, dove l'area A corrisponde alla regione più scura. Inoltre l'area A di tale superficie si può calcolare analiticamente e risulta essere pari a:

$$A_{ex} = 4 \arcsin\left(\frac{4}{5}\right)$$

Suggerimento: in C la funzione per calcolare $\arcsin(x)$ è `asin()` che è definita in `math.h`.
In particolare il programma dovrà:

1. Chiedere in input il numero N_p (intero positivo) di punti da generare per stimare A con il metodo Monte Carlo.
2. Calcolare l'area A come segue (metodo *hit and miss*):
 - Generare a caso N_p punti all'interno del quadrato di lato pari a 4 centrato anch'esso nell'origine degli assi, ovvero generare a caso N_p punti di coordinate (x, y) con $-2 < x < 2$ e $-2 < y < 2$.
 - Contare il numero di volte N_h che i punti di coordinate (x, y) cadono all'interno della superficie che si ottiene considerando l'intersezione delle superfici delle due ellissi definite dalle equazioni (1) e (2). *Suggerimento:* la condizione in C perché un punto di coordinate (x, y) sia in A si può scrivere `x*x/4 + y*y < 1 && x*x + y*y/4 < 1`

Una stima dell'area A sarà dunque fornita dalla seguente formula:

$$A = 16N_h/N_p$$

dove 16 è l'area del quadrato all'interno del quale si stanno generando gli N_p punti casuali di coordinate (x, y) .

3. Stampare su schermo il risultato dell'integrazione (ossia il valore di A) per $N_p = 10^7$ ed il valore analitico utilizzando 2 cifre dopo la virgola.

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

- `inserimento(...)`, che richiede l'inserimento di un numero intero positivo
- `genera_coordinata()`, che restituisce un numero casuale in virgola mobile compreso tra -2 e 2 .

► Seconda parte:

Copiate il file `hitmiss.c` in un nuovo file chiamato `hmsave.c` e modificalo in modo che i punti generati, che cadono all'interno di A , vengano salvati su di un file chiamato `punti.dat` tramite le istruzioni C `fopen`, `fprintf` e `fclose`. Generate $N_p = 1000$ punti e con python graficateli insieme alle due ellissi indicate nelle equazioni (1) e (2). Per disegnare due ellissi in python potete utilizzare il seguente script:

```
1 import matplotlib.pyplot as plt
2 from matplotlib.patches import Ellipse
3 a = plt.subplot(1,1,1, aspect='equal')
4 a.add_patch(Ellipse((0, 0), 4, 2, 0, fill=False))
5 a.add_patch(Ellipse((0, 0), 4, 2, 90, fill=False))
6 plt.xlabel('x')
7 plt.ylabel('y')
8 plt.grid()
9 plt.xlim(-2, 2)
10 plt.ylim(-2, 2)
```

```
11 plt.show()
```

Modificate quindi tale codice aggiungendo (tra `plt.ylim` e `plt.show`) le opportune istruzioni per graficare anche i punti contenuti nel file `punti.dat`. Salvate infine il grafico così ottenuto nel file `punti.png`.