Pendolo fisico

Giosué Aiello, Domenico Fenili, Francesco Sermi

14 Novembre 2023

Indice

1 Scopo dell'esperienza

Lo scopo di questa esperienza è quello di misurare il periodo di un pendolo fisico in funzione della distanza del perno di rotazione dal centro di massa.

2 Cenni teorici

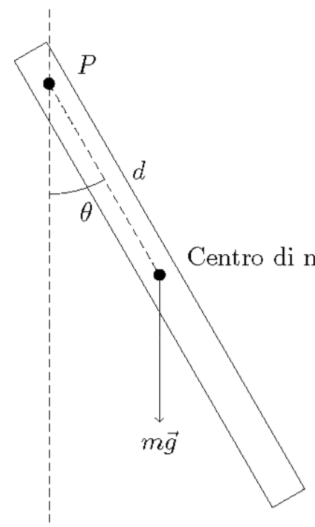


Figura 1: Schema del nostro apparato sperimentale

Un oggetto fissato ad un punto di sospensione P (che dista d dal centro di massa) e soggetto alla gravità costituisce un pendolo fisico. Se questo corpo viene spostato di un angolo θ dalla sua posizione di equilibrio, il momento torcente della forza di gravità (rispetto al punto di sospensione P) vale:

$$\tau = -mgd \sin\theta \tag{1}$$

che, per $\theta << 10° - 15°$ possiamo esprimere $sin(\theta)$ utilizzando la formula di espansione in serie di Taylor al primo ordine:

$$\sin\theta = \sum_{n=0}^{+\infty} \frac{(-1)^n}{(2n+1)!}\theta^{2n+1} = \theta + o(\theta^3) \approx \theta$$

Pertanto possiamo riscrivere il momento torcente della forza di gravità come:

$$\tau = -mgd\theta$$

E per la seconda equazione cardinale si ha che:

$$\tau = \frac{dL}{dt} \tag{2}$$

e sapendo che il momento angolare di un pendolo fisico risulta essere pari a $L = I\omega$ e $\omega = \frac{d\theta}{dt}$ si ha che:

$$\tau = \frac{dL}{dt} = I\frac{d}{dt}\left(\frac{d\theta}{dt}\right) = I\frac{d^2\theta}{dt^2}$$

Combinando la (1) e la (2):

$$I\frac{d^2\theta}{dt^2} = -mgd\theta \implies \frac{d^2\theta}{dt^2} + \frac{mgd}{I}\theta = 0 \tag{3}$$

Siamo dinanzi ad un'equazione differenziale di secondo ordine a coefficienti costanti omogenea di un moto armonico con pulsazione e periodo di oscillazione dati da:

$$\omega_0 = \sqrt{\frac{mgd}{I}} \quad T_0 = 2\pi\sqrt{\frac{I}{mgd}}$$

Utilizzando il teorema degli assi paralleli, possiamo concludere che il momento di inerzia dell'oggetto fisico risulta essere:

$$I = I_{cm} + md^2 = \frac{ml^2}{12} + md^2$$

Possiamo quindi riscrivere la formula nella seguente maniera:

$$T(d) = \sqrt{\frac{m(l^2 + d^2)}{mgd}} = \sqrt{\frac{\frac{l^2}{12} + d^2}{gd}} \tag{4}$$

3 Apparato sperimentale e strumenti

- Strumenti
  - Metro a nastro, risoluzione 0.1 cm;
  - Calibro ventesimale, risoluzione 0.05 mm;
  - Cronometro, risoluzione 0.01 s.
- Materiali
  - Asta metallica forata;
  - Un supporto di sospensione;

4 Descrizione delle misure

–Completare

Per fare le misurazioni

Numero prova	$\tau(s)$ ±0.01	$d(cm)$ ±0.1
1	16.09	
2	15.90	
3	15.73	
4	15.93	95.0
5	15.89	
6	15.67	
7	16.00	

Numero prova	τ (s) ±0.01	d (cm) ±0.1
1	15.31	
2	15.42	
3	15.30	
4	15.56	85.0
5	15.29	
6	15.50	
7	15.53	

Numero prova	τ (s) ±0.01	d (cm) ±0.1
1	15.75	
2	15.66	
3	15.73	
4	15.61	75.0
5	15.67	
6	15.80	
7	15.67	

Numero prova	τ (s) ± 0.01	d (cm) ± 0.1
1	15.75	
2	15.66	
3	15.73	
4	15.61	75.0
5	15.67	
6	15.80	
7	15.67	

Numero prova	τ (s) ± 0.01	d (cm) ± 0.1
1	15.75	
2	15.66	
3	15.73	
4	15.61	75.0
5	15.67	
6	15.80	
7	15.67	

Numero prova	τ (s) ± 0.01	d (cm) ± 0.1
1	15.75	
2	15.66	
3	15.73	
4	15.61	75.0
5	15.67	
6	15.80	
7	15.67	

5 Analisi dei dati

Per minimizzare le incertezze sul periodo di oscillazione (che **non** coincidono con la risoluzione del cronometro o al tempo di reazione di un individuo medio), abbiamo considerato i vari valori di τ misurati ad una certa distanza dal centro di massa e ne abbiamo calcolato il valor medio e la deviazione standard, che abbiamo poi diviso per il numero di oscillazioni fissato che abbiamo misurato. (inserire tabella)
Successivamente, i valori medi e i valori della deviazione standard sono stati utilizzati per effettuare il fit lineare in Python utilizzando il parametro l come parametro libero, che si può osservare qua sotto:

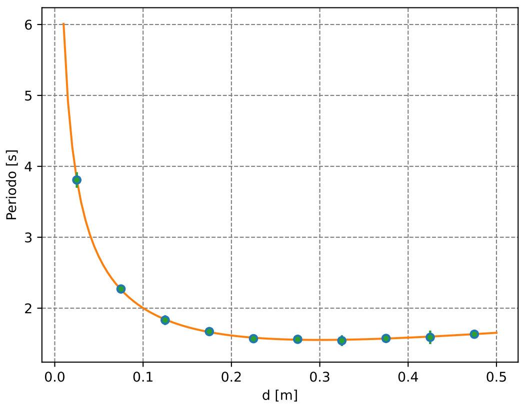


Figura 2: Grafico del fit ottenuto con *Scipy*

La libreria *Scipy* ci ha restituito che il miglior fit del nostro grafico risulta essere il valore $\hat{l} = 1.037m \pm 0.002m$ e possiamo notare che se calcoliamo l'errore associato:

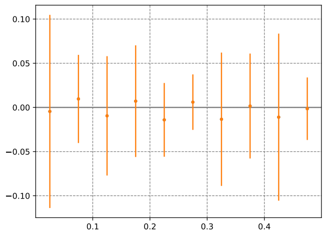
$$\frac{l - \hat{l}}{\sigma_{\hat{l}}} = \frac{(\,1.05 - 1.037\,)\,\mathrm{m}}{0.002\,\mathrm{m}} = 6.5\sigma_{\hat{l}}$$

Si osserva che, confrontandolo tramite l'incertezza associata a $\hat{l}$, la lunghezza effettiva dell'asta si discosta significativamente dal valore del best-fit ottenuto con la funzione `curve_fit` di Scipy. Inoltre, facendo il grafico dei residui r_i, dove:

$$r_i = T_i - 2\pi\sqrt{\frac{\frac{\hat{l}}{12} + d_i^2}{gd_i}} \tag{5}$$

dove T_i è il valore medio dei τ calcolato dalla tabella $i - esima$ presenta nella sezione superiore e, consequenzialmente, r_i è il suo residuo rispetto al modello teorico che noi cerchiamo di dimostrare:

si nota che il grafico di molti di questi valori presenta un valore


0.10
0.05
0.00
−0.05
−0.10
0.1
0.2
0.3
0.4

```python
import numpy as np
import math
from matplotlib import pyplot as plt
from scipy.optimize import curve_fit
data = np.loadtxt(fname="dati.txt", dtype=np.float64)
class Number:
    def __init__(self, arr):
        self.x = arr
        self.n = 7
        self.average_value = 0
    def media(self):
        sum = 0
        for x in self.x:
            sum = sum + x
        self.average_value = sum/len(self.x)
        return(self.average_value)
    def deviazione_standard(self):
        sum = 0
        if self.average_value != 0:
            for x in self.x:
                sum = sum + pow(x - self.average_value, 2)
        self.deviazione = math.sqrt(sum * (len(self.x)))
        return(self.deviazione)
T = np.ones(len(data))
sigma_T = np.ones(len(data))
for el in range(0, len(data)):
    arr = Number(data[el])
    T[el] = arr.media()
    sigma_T[el] = arr.deviazione_standard()
T = T/10
sigma_T = sigma_T/10
# Dati---mettete le vostre misure!
# Qui potete anche leggere i dati da file, usando il metodo np.loadtxt(),
# se lo trovate comodo.
L = np.array([0.95, 0.85, 0.75, 0.65, 0.55, 0.45, 0.35, 0.25, 0.15, 0.05])
d = abs(L - 0.525)
sigma_d = np.full(d.shape, 0.002)
# T = np.array([15.9, 15.4, 15.7, 18.3, 38.1, 22.7, 16.7, 15.61, 15.7, 16.3])
# T = T/10
# sigma_T = np.array([0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.06, 0.1, 0.1])
# sigma_T = sigma_T/10
# Definizione dell'accelerazione di gravita'.
g = 9.81
def period_model(d, l):
    """Modello per il periodo del pendolo.
    """
    return 2.0 * np.pi * np.sqrt((l**2.0 / 12.0 + d**2.0) / (g * d))
plt.figure("Periodo")
# Scatter plot dei dati.
plt.errorbar(d, T, sigma_T, sigma_d, fmt="o")
# Fit---notate che questo e' un fit ad un solo parametro.
popt, pcov = curve_fit(period_model, d, T, sigma=sigma_T)
l_hat = popt[0]
sigma_l = np.sqrt(pcov[0, 0])
# Confrontate i parametri di best fit con la vostra misura diretta!
print(l_hat, sigma_l)
# Grafico del modello di best-fit.
x = np.linspace(0.01, 0.5, 100)
plt.plot(x, period_model(x, l_hat))
plt.errorbar(d, T, yerr=sigma_T, xerr=sigma_d, fmt='.')
plt.xlabel("d [m]")
plt.ylabel("Periodo [s]")
plt.grid(which="both", ls="dashed", color="gray")
r = T - period_model(d, l=1.05)
sigma_r = sigma_T
plt.savefig("massa_raggio.pdf")
plt.show()
plt.plot(d, r, linestyle='', marker='.')
plt.axhline(y = 0, color = 'gray', linestyle = '-')
plt.errorbar(d, r, sigma_r, sigma_d, fmt=".")
plt.show()
```