

Pendolo fisico

Giosué Aiello, Domenico Fenili, Francesco Sermi

14 Novembre 2023

Indice

1	Scopo dell'esperienza	3
2	Cenni teorici	3
3	Apparato sperimentale e strumenti	4
4	Descrizione delle misure	4
5	Analisi dei dati	5

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

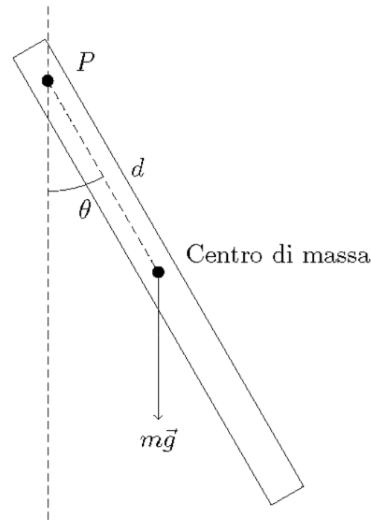


Figura 1: Schema del nostro apparato sperimentale

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:

$$\tau = -mgd \sin \theta \quad (1)$$

che, per $\theta \ll 10^\circ - 15^\circ$ 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:

$$\tau = \frac{dL}{dt} \quad (2)$$

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):

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

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:

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

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	95.0
2	15.90	
3	15.73	
4	15.93	
5	15.89	
6	15.67	
7	16.00	

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

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

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

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

Numero prova	τ (s) ± 0.01	d (cm) ± 0.1
1	15.75	75.0
2	15.66	
3	15.73	
4	15.61	
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:

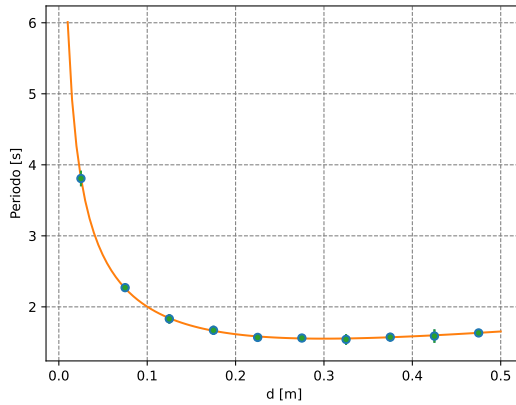


Figura 2: Grafico del fit ottenuto con *Scipy*

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:

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

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:

$$r_i = T_i - 2\pi\sqrt{\frac{\hat{l}}{12} + d_i^2} \quad (5)$$

```

1 import numpy as np
2 import math
3 from matplotlib import pyplot as plt
4 from scipy.optimize import curve_fit
5 data = np.loadtxt(fname="dati.txt", dtype=np.float64)
6 class Number:
7     def __init__(self, arr):
8         self.x = arr
9         self.n = 7
10        self.average_value = 0
11    def media(self):
12        sum = 0
13        for x in self.x:
14            sum = sum + x
15        self.average_value = sum/len(self.x)
16        return(self.average_value)
17    def deviazione_standard(self):
18        sum = 0
19        if self.average_value != 0:
20            for x in self.x:
21                sum = sum + pow(x - self.average_value, 2)
22        self.deviazione = math.sqrt(sum * (len(self.x)))
23        return(self.deviazione)
24 T = np.ones(len(data))
25 sigma_T = np.ones(len(data))
26 for el in range(0, len(data)):
27     arr = Number(data[el])
28     T[el] = arr.media()
29     sigma_T[el] = arr.deviazione_standard()
30 T = T/10
31 sigma_T = sigma_T/10
32 # Dati---mettete le vostre misure!
33 # Qui potete anche leggere i dati da file, usando il metodo np.loadtxt(),
34 # se lo trovate comodo.
35 L = np.array([0.95, 0.85, 0.75, 0.65, 0.55, 0.45, 0.35, 0.25, 0.15, 0.05])
36 d = abs(L - 0.525)
37 sigma_d = np.full(d.shape, 0.002)
38 # T = np.array([15.9, 15.4, 15.7, 18.3, 38.1, 22.7, 16.7, 15.61, 15.7, 16.3])
39 # T = T/10
40 # sigma_T = np.array([0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.06, 0.1, 0.1])
41 # sigma_T = sigma_T/10
42 # Definizione dell'accelerazione di gravita'.
43 g = 9.81
44 def period_model(d, l):
45     """Modello per il periodo del pendolo.
46     """
47     return 2.0 * np.pi * np.sqrt((l**2.0 / 12.0 + d**2.0) / (g * d))
48 plt.figure("Periodo")
49 # Scatter plot dei dati.
50 plt.errorbar(d, T, sigma_T, fmt="o")
51 # Fit---notate che questo e' un fit ad un solo parametro.
52 popt, pcov = curve_fit(period_model, d, T, sigma=sigma_T)
53 l_hat = popt[0]
54 sigma_l = np.sqrt(pcov[0, 0])
55 # Confrontate i parametri di best fit con la vostra misura diretta!
56 print(l_hat, sigma_l)
57 # Grafico del modello di best-fit.
58 x = np.linspace(0.01, 0.5, 100)
59 plt.plot(x, period_model(x, l_hat))
60 plt.errorbar(d, T, yerr=sigma_T, xerr=sigma_d, fmt='.')
61 plt.xlabel("d [m]")
62 plt.ylabel("Periodo [s]")
63 plt.grid(which="both", ls="dashed", color="gray")
64 r = T - period_model(d, l=1.05)
65 sigma_r = sigma_T
66 plt.savefig("massa_raggio.pdf")
67 plt.show()
68 plt.plot(d, r, linestyle='', marker='.')
69 plt.axhline(y = 0, color = 'gray', linestyle = '-')
70 plt.errorbar(d, r, sigma_r, sigma_d, fmt=".")
71 plt.show()
72

```