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

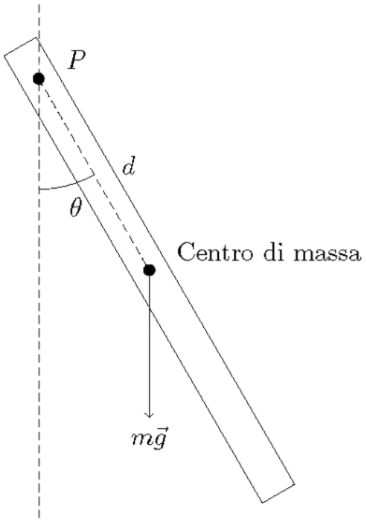

Figura 1: Schema del nostro apparato sperimentale

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

In primis, abbiamo misurato la lunghezza dell'asta $l = (105,0 \pm 0.1)cm$ con il metro a nastro. Successivamente abbiamo misurato la distanza dal centro di massa della nostra asta dei vari buchi in cui avremmo inserito il perno di rotazione con il metro a nastro, visto che le incertezze derivanti dalla misurazione col metro erano, in relazione alle grandezze da noi misurate, molto piccole e non è stato necessario misurarle col calibro.
Successivamente abbiamo iniziato a misurare i periodi di oscillazione τ misurando quanto occorreva al pendolo a compiere 10 oscillazioni, per poter ridurre l'errore sul periodo di oscillazione effettivo (perché l'incertezza sulla misura delle oscillazioni **non** è il tempo di reazione di un essere umano). Queste misurazioni sono state effettuato con il cronometro, sono state ripetute 7 volte per ogni foro.
L'unica accortezza che abbiamo deciso di seguire per ogni misura è stato quello di far partire il pendolo da un angolo per cui era ragionevole supporre valida l'approssimazione dei piccoli angoli mostrata nei "Cenni teorici", cercando di mantenere invariato l'angolo fra la verticale e il punto iniziale in cui si trovava l'asta.

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 divisi per il numero di oscillazioni che il pendolo aveva effettuato durante la misurazione (che abbiamo riportato nella tabella qua accanto).

Valore medio	Deviazione standard
1.59	0.09
1.54	0.08
1.57	0.04
1.83	0.07
3.8	0.1
2.27	0.05
1.67	0.06
1.56	0.03
1.57	0.06
1.63	0.05

Tabella 1: Tabella dei valori medi

Successivamente, questi sono stati utilizzati per effettuare il fit lineare in Python utilizzando il parametro l come parametro libero, di cui abbiamo fatto il grafico utilizzando il valore di *best-fit* $\hat{l}$ che abbiamo

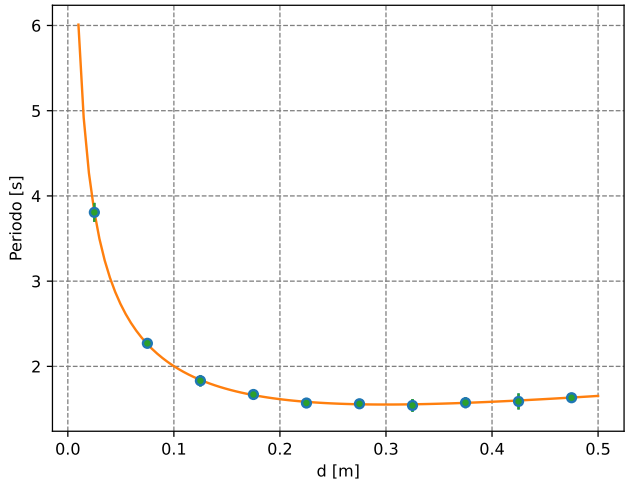

Figura 2: Grafico del fit ottenuto con `scipy`

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

$$\frac{l - \hat{l}}{\sqrt{\sigma_{\hat{l}}^2 + \sigma_l^2}} = \frac{(1.05 - 1.037)\,\mathrm{m}}{(0.002 + 0.01)\,\mathrm{m}} \approx 1.3$$

Pertanto il valore del *best-fit* $\hat{l}$ ha una buona precisione, visto che, rispetto al valore da noi atteso, dista solamente $1.3\sigma_{l-\hat{l}}$.
Considerando i residui r_i, dove:

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

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, ne abbiamo disegnato il grafico:

Si osserva che i residui rispetto al modello oscillano tutti rispetto allo zero, pertanto possiamo considerarli come "buoni" perché il fatto che fluttuino in questa maniera ci assicura che non erano presenti degli errori sistematici dovuti al nostro apparato sperimentale oppure che questi abbiano influito superficialmente nelle nostre misure.

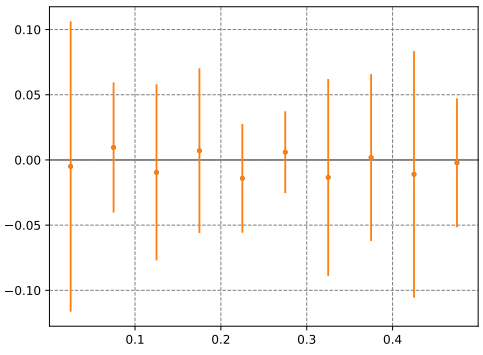

Figura 3: Grafico dei residui

Per quanto riguarda l'elaborazione di questi dati, abbiamo utilizzato il codice scritto in Python presente nella prossima pagina, che tramite una serie di librerie (fra i quali si annovera *Scipy*, *Matplotlib* e *Numpy*) attraverso cui è stato possibile svolgere le operazioni di fit lineare con parametro libero, il calcolo della deviazione standard e del valore medio.
Si cerca di dare una breve spiegazione di questo codice:

```
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)
```

All'inizio del file si includono tutte le librerie necessarie per l'interpolazione dei dati (`numpy` e `math`), per la creazione dei file (la classe `pyplot` dalla libreria `matplotlib`) e per effettuare l'analisi tramite fit lineare (la funzione `curve_fit` della libreria `scipy`). Successivamente, abbiamo aperto i dati da noi misurati tramite la funzione `open` di *Python* e abbiamo definito una classe chiamata `Number` con i metodi `media` e `deviazione_standard` che effettuava tutti i calcoli necessari per calcolare il valore medio e la deviazione standard dell'array passato al costruttore in cui erano presenti tutte le misurazioni di τ ad una certa distanza dal centro di massa.

Il blocco di codice qua accanto va a definire il vettore `T` e il vettore `sigma_T` (che contiene le incertezze sugli elementi di T) e iteriamo all'interno del file `data.txt` (che contiene sulle righe n misurazioni dell'oscillazione del pendolo ad una certa distanza d) per andare a "riempire" i due vettori: ogni riga del file viene presa e passata alla classe `Number` su cui vengono eseguite l'operazione di calcolo del valor medio e, successivamente, la misura della deviazione standard della media.

Successivamente definiamo il vettore L che contiene la distanza dei vari fori da un'estremità dell'asta che abbiamo usato nella riga successiva per definire il vettore d che effettivamente contiene la distanza del perno di rotazione dal centro di massa del sistema (e abbiamo avuto l'accortezza di porre le incertezze, come si può vedere nella riga successiva, 0.002 m ad ogni cella del vettore `sigma_d`, che contiene le incertezze legate ad ogni elemento del vettore `d`, il doppio dell'incertezza perché, ogni elemento di L, è stato sottratto a 0.525 m a cui era assegnata l'incertezza di 0.001 m).

Tutti questi vettori vengono presi e passati alla funzione `curve_fit()` di `scipy` che si occupa di fare il fit lineare della funzione e le istruzioni successive usano le varie funzioni della libreria `matplotlib` per disegnare i grafici "Distanza-Periodo" e il grafico dei residui (con i relativi errori sulle ascisse e sulle ordinate).

```
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
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)
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")
popt, pcov = curve_fit(period_model, d, T,
    sigma=sigma_T)
l_hat = popt[0]
sigma_l = np.sqrt(pcov[0, 0])
print(l_hat, sigma_l)
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()

```