

# Studio Sperimentale delle Proprietà di una Molla

Claudiu Mihail Pantiru, Pietro Pennestrì, Andrea Quirini

## *Report di Laboratorio*

### Sommario

L'esperienza qui descritta, svolta nel laboratorio di Fisica di Ingegneria Elettronica in data 27/04/2016, ha lo scopo di determinare sperimentalmente le proprietà dinamiche di una molla. In particolare, utilizzando un metodo statico e un metodo dinamico vengono stimate la costante elastica  $k$ , la massa  $m_0$  della molla e le rispettive incertezze.



## INDICE

1	Descrizione dell'apparato Sperimentale	2
2	Parte I - Misura della costante elastica $k$ della molla con il metodo statico	2
3	Parte II - Determinazione della costante elastica $k$ di una molla: metodo dinamico	6
4	Parte III	11
5	Appendice	13
6	Codice	13
6.1	Elaborazione Tabella 1	13
6.2	Elaborazione Tabella 2	16
6.3	Elaborazione Tabella 3	19

## INTRODUZIONE

L'esperienza si compone di tre parti:

- **Parte I** Determinazione della costante di rigidità  $k$  della molla mediante metodo statico.
- **Parte II** Determinazione della costante di rigidità  $k$  e della massa  $m_0$  della molla e del piattello mediante metodo dinamico.
- **Parte III** Determinazione del coefficiente  $k$  mediante un metodo grafico.

- 
- *Claudiu Mihail Pantiru, nr. 1705328,  
E-mail: pantiru.1705328@studenti.uniroma1.it,*
  - *Pietro Pennestrì, nr. 1694905,  
E-mail: pennestrì.1694905@studenti.uniroma1.it,*
  - *Andrea Quirini, nr. 1693547,  
E-mail: quirini.1693547@studenti.uniroma1.it,  
Sapienza - Università di Roma.*

L'elaborazione matematica ed il *post-processing* dei dati raccolti è stata eseguita mediante apposite procedure in linguaggio Python 2.7. In particolare sono state utilizzate le librerie NumPy, per i calcoli statistici e matplotlib per i grafici.

## 1 DESCRIZIONE DELL'APPARATO SPERIMENTALE

L'apparato sperimentale, come mostrato in Figura 1, è costituito da una molla fissata da un'estremità ad un telaio. Sull'altro capo della molla si possono collegare delle masse note  $m_i$ , per valutarne la costante di rigidità  $k$ .

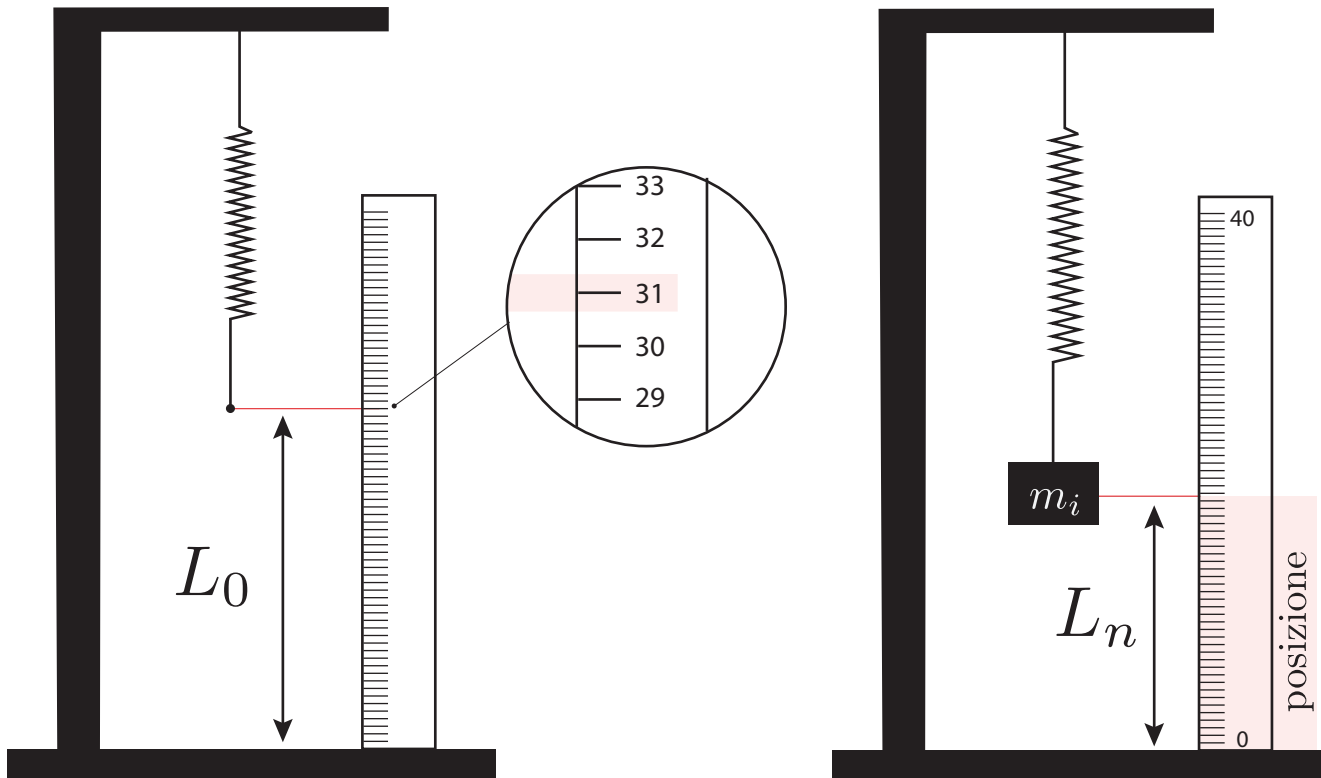


Figura 1: Apparato Sperimentale

Nel corso dell'esperienza viene adottata la seguente convenzione:

- $L_0$  indica la distanza tra la base del telaio e l'estremità libera della molla;
- $L_n$  indica la distanza tra la base del telaio e la massa  $m_i$  agganciata alla molla.

## 2 PARTE I - MISURA DELLA COSTANTE ELASTICA $k$ DELLA MOLLA CON IL METODO STATICO

La legge di Hooke, che ipotizza una relazione lineare tra la deformazione della molla e l'entità delle forze applicate all'estremità, viene qui impiegata per la stima della rigidità della molla medesima.

Nel nostro caso le forze applicate sono le forze peso che agiscono sulle masse che via via vengono aggiunte o tolte dall'estremità.

Vengono quindi descritte, nell'ordine in cui si susseguono, le operazioni compiute ed i risultati delle elaborazioni sono riportati in forma tabulare e grafica.

### Task I-1

Viene misurata la posizione  $L_0 = 31.1\text{cm}$  del piattello vuoto;

**Task I-2, I-3, I-4, I-5**

Si aggiunge al piattello vuoto un dischetto alla volta di massa  $m_m = 81.4g$ , fino ad un massimo di dieci dischetti e si misura (fase di andata) la posizione del piattello. I dischetti vengono quindi rimossi uno ad uno e si misura (fase di ritorno) la posizione del piattello.

Indicata con  $L_n$  la generica posizione del piattello con la presenza della n-esima massa, definiamo le variazioni

$$\Delta L_n = L_n - L_0 \quad (n = 1, 2, \dots, 10) \quad (1)$$

$$\Delta L_s = L_{n+1} - L_n \quad (n = 1, 2, \dots, 10) \quad (2)$$

Le misurazioni ottenute sono riportate nelle Tabelle 1 e 2:

Tabella 1: Posizione della molla  $L_n$  in funzione della massa

N.Disco	Massa kg	1. Posizione $L_n$ mm		2. Posizione $L_n$ mm		3. Posizione $L_n$ mm		Media mm	Errore mm
		Andata	Ritorno	Andata	Ritorno	Andata	Ritorno		
0	0.0	310.00	310.00	310.00	309.00	310.00	310.00	309.83	0.50
1	0.0814	302.00	301.00	303.00	302.00	301.00	302.00	301.83	1.00
2	0.1628	286.00	288.00	287.00	288.00	288.00	287.00	287.33	1.00
3	0.2442	266.00	267.00	266.00	266.00	267.00	266.00	266.33	0.50
4	0.3256	245.00	244.00	245.00	244.00	245.00	244.00	244.50	0.50
5	0.407	223.00	223.00	223.00	223.00	223.00	223.00	223.00	0.00
6	0.4884	201.00	202.00	201.00	201.00	201.00	201.00	201.17	0.50
7	0.5698	181.00	180.00	180.00	180.00	181.00	180.00	180.33	0.50
8	0.6512	158.00	159.00	159.00	158.00	156.00	157.00	157.83	1.50
9	0.7326	137.00	138.00	137.00	137.00	137.00	136.00	137.00	1.00
10	0.814	116.00	116.00	117.00	117.00	116.00	117.00	116.50	0.50

Tabella 2: Allungamento della molla  $\Delta L_s$  in funzione della molla

N.Disco	Massa kg	1. Allungamento $\Delta L_s$ mm		2. Allungamento $\Delta L_s$ mm		3. Allungamento $\Delta L_s$ mm	
		Andata	Ritorno	Andata	Ritorno	Andata	Ritorno
0	0.0	0.00	0.00	0.00	0.00	0.00	0.00
1	0.0814	8.00	10.00	7.00	7.00	9.00	9.00
2	0.1628	16.00	13.00	16.00	14.00	14.00	15.00
3	0.2442	20.00	21.00	21.00	22.00	20.00	21.00
4	0.3256	21.00	23.00	21.00	22.00	22.00	20.00
5	0.407	22.00	21.00	22.00	21.00	23.00	23.00
6	0.4884	22.00	22.00	22.00	22.00	21.00	21.00
7	0.5698	20.00	22.00	21.00	21.00	21.00	21.00
8	0.6512	23.00	21.00	21.00	22.00	22.00	22.00
9	0.7326	21.00	21.00	22.00	21.00	21.00	22.00
10	0.814	21.00	22.00	20.00	20.00	20.00	21.00
<b>Media</b>		17.64	17.82	17.55	17.45	17.55	17.73

**Task I-6**

La Figura 2 riporta lo spostamento relativo  $\Delta L_s$  in funzione della massa totale  $m_n$

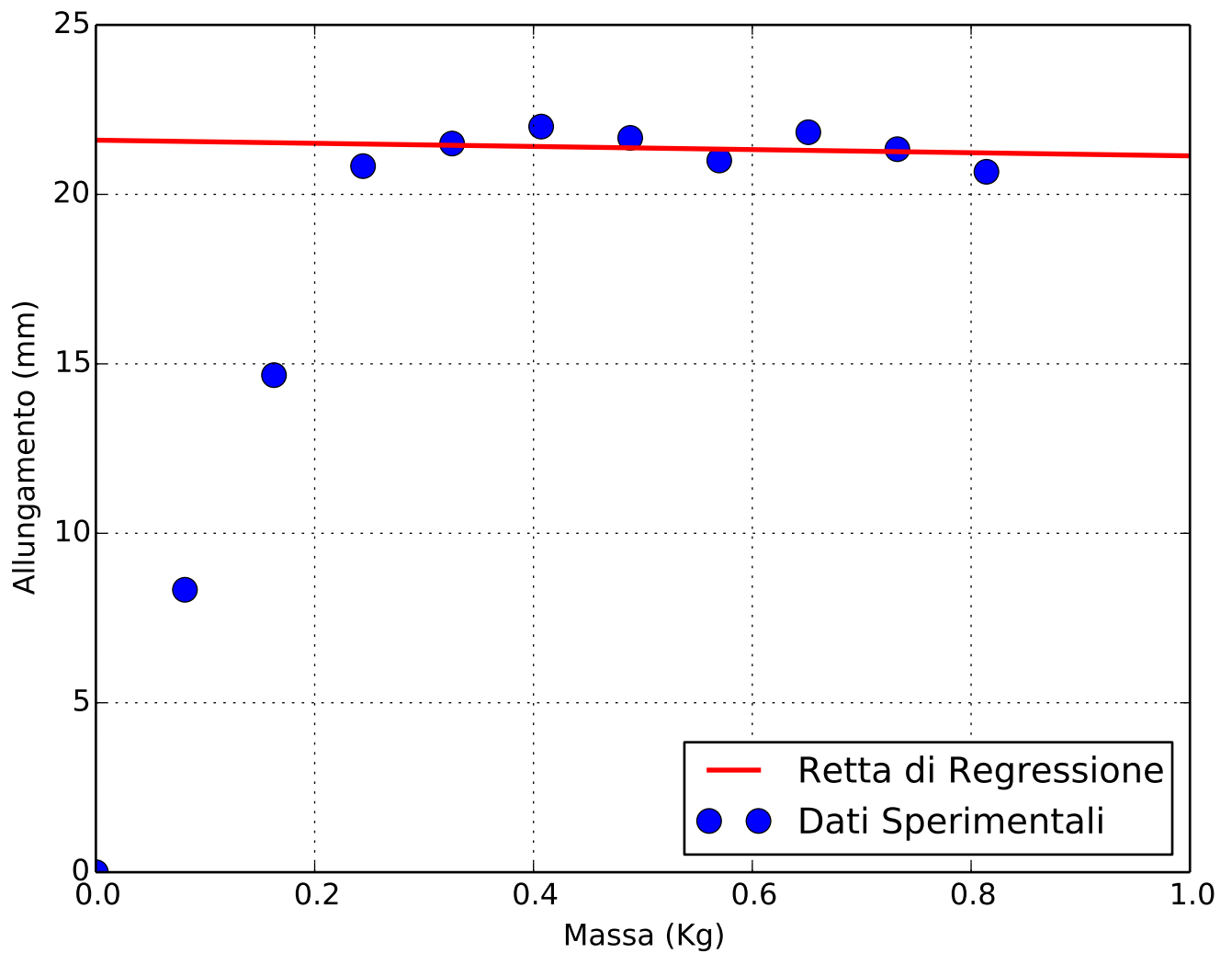


Figura 2: Spostamento relativo  $\Delta L_s$  in funzione della massa totale  $m_n$

La retta non sar  orizzontale per i seguenti motivi:

- Posizione iniziale della molla non perfettamente verticale;
- Errori di misura;
- non esatta corrispondenza tra legge di Hooke e comportamento reale.

#### Task I-7

La Figura 3 riporta la posizione  $L_n$  in funzione della massa totale  $m_n$  (senza errore), ovvero

$$L_n = L_0 + \frac{g}{k} m_n . \quad (3)$$

La precedente relazione pu  essere riscritta nella forma linearizzata

$$y = A + Bx \quad (4)$$

dove:

$$x = m_n \quad (5)$$

$$A = L_0 \quad (6)$$

$$B = \frac{g}{k} \quad (7)$$

$$y = L_n \quad (8)$$

Pertanto, la costante di rigidità  $k$  della molla risulta essere:

$$k = \left| \frac{g}{B} \right| \quad (9)$$

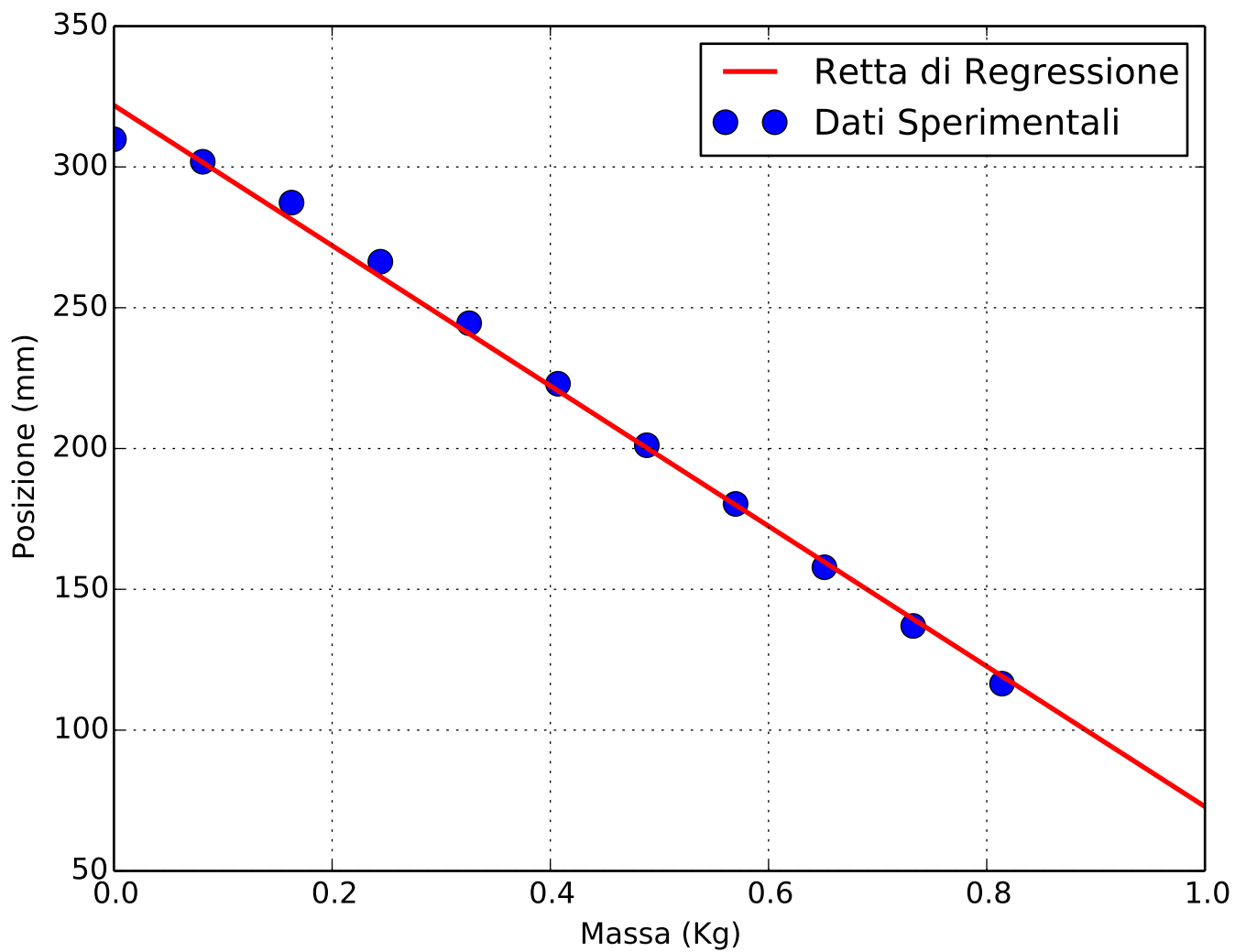


Figura 3: Spostamento relativo  $L_n$  in funzione della massa totale  $m_n$

Dai risultati della regressione di Figura 3 si ottiene

$$A = 321.87 \pm 2.95$$

$$B = -249.03 \pm 6.13$$

**Task I-8**

In definitiva il valore della rigidità  $k$  della molla, stimato con il **metodo statico** (v. equazione (9)), risulta essere:

$$k = 0.0393925136408 \approx 0.04 \frac{\text{N}}{\text{mm}}$$

### 3 PARTE II - DETERMINAZIONE DELLA COSTANTE ELASTICA $k$ DI UNA MOLLA: METODO DINAMICO

Conoscendo la massa  $m$  attaccata alla molla è possibile ottenerne la rigidità  $k$  misurando il periodo  $T$  delle oscillazioni libere della massa medesima.

Il periodo  $T$  delle oscillazioni è legato a massa  $m$  e rigidità  $k$  secondo le relazioni:

$$T = \frac{2\pi}{\omega_n} \quad (10a)$$

$$T = 2\pi\sqrt{\frac{m}{k}} \quad (10b)$$

Grazie a queste ultime si riesce a determinare mediante il **metodo dinamico** la costante  $k$  della molla. Tale metodo prevede che vengano acquisiti sperimentalmente, utilizzando il conometro, i tempi di oscillazioni  $T_1, T_2, \dots, T_n$  della molla con all'estremità  $m_1, m_2, \dots, m_n$ . Nell'analisi sperimentale si dovrà tener conto anche della massa distribuita della molla e di quella del piattello. La somma di queste ultime viene indicata con  $m_0$ . In definitiva, la relazione che lega i periodi  $T_i$  di oscillazione misurati, le masse  $m_i$  attaccate alla molla e la rigidità  $k$  risulta essere:

$$T_i = 2\pi\sqrt{\frac{m_0}{k} + \frac{m_i}{k}} \quad (i = 1, 2, \dots, n) \quad (11)$$

Per l'analisi degli errori è opportuno riscrivere la precedente relazione nella forma linearizzata

$$y = A + Bx \quad (12)$$

dove

$$x = m \quad (13)$$

$$y = T^2 \quad (14)$$

$$A = 4\pi^2 \frac{m_0}{k} \quad (15)$$

$$B = 4\pi^2 \frac{1}{k} \quad (16)$$

La relazione (12) può essere interpretata quale retta di *regressione* i cui parametri  $A$  e  $B$  vengono stimati sperimentalmente sulla base delle misure ripetute dei periodi di oscillazione  $T_1, T_2, \dots, T_n$ , quando all'estremità della molla vi siano attaccate le masse  $m_1, m_2, \dots, m_n$ .

Grazie a tale interpretazione, possiamo affermare che la rigidità  $k$  e la rispettiva incertezza  $\pm\Delta k$ , risultano essere:

$$k = \frac{k_{\max} + k_{\min}}{2} = 2\pi^2 \left( \frac{B_{\max} + B_{\min}}{B_{\max} B_{\min}} \right) \quad (17)$$

$$\Delta k = \frac{k_{\max} - k_{\min}}{2} = 2\pi^2 \left( \frac{B_{\max} - B_{\min}}{B_{\max} B_{\min}} \right) \quad (18)$$

con  $B_{\max}$  e  $B_{\min}$ , rispettivamente, i valori massimo e minimo della retta di regressione.

Tabella 3: Periodo di oscillazione della molla in funzione della massa

N.Disco	Massa kg	Periodo $T_{10}$ s			Periodo $T$ s	$T^2$ s <sup>2</sup>	Errore $\Delta T_{10}$ s	Errore $\Delta T$ s	$(T + \Delta T)^2$ s	$(T - \Delta T)^2$ s
		Periodo I	Periodo II	Periodo III						
9	0.814	9.16	9.20	9.18	0.92	0.84	0.02	0.00	0.85	0.84
8	0.7326	9.00	8.90	8.80	0.89	0.79	0.10	0.01	0.81	0.77
7	0.6512	8.80	8.75	8.70	0.88	0.77	0.05	0.01	0.77	0.76
6	0.5698	8.30	8.12	8.20	0.82	0.67	0.09	0.01	0.69	0.66
5	0.4884	7.66	7.63	7.72	0.77	0.59	0.04	0.00	0.60	0.58
4	0.407	7.02	7.10	7.00	0.70	0.50	0.05	0.00	0.50	0.49
3	0.3256	6.20	6.57	6.30	0.64	0.40	0.19	0.02	0.43	0.38
2	0.2442	5.72	5.68	5.65	0.57	0.32	0.03	0.00	0.33	0.32
1	0.1628	4.82	4.72	4.53	0.47	0.22	0.15	0.01	0.23	0.21

Il valore di  $m_0$  si otterrà dalla (15)

$$m_0 = \frac{Ak}{4\pi^2} \quad (19)$$

### Task II-1, II-2, II-3

Sono stati gradualmente posizionati sul piattello tutti i dischetti a disposizione. Il sistema veniva quindi fatto oscillare liberamente nella direzione verticale, dopo aver spostato il piattello di qualche centimetro dalla posizione di equilibrio.

È stata quindi misurata la durata  $T_{10}$  di 10 oscillazioni consecutive complete e compilata la Tabella 3.

### Task II-4, II-5

Sulla base dei risultati acquisiti, sono stati ottenuti i grafici delle Figure 4 -6.

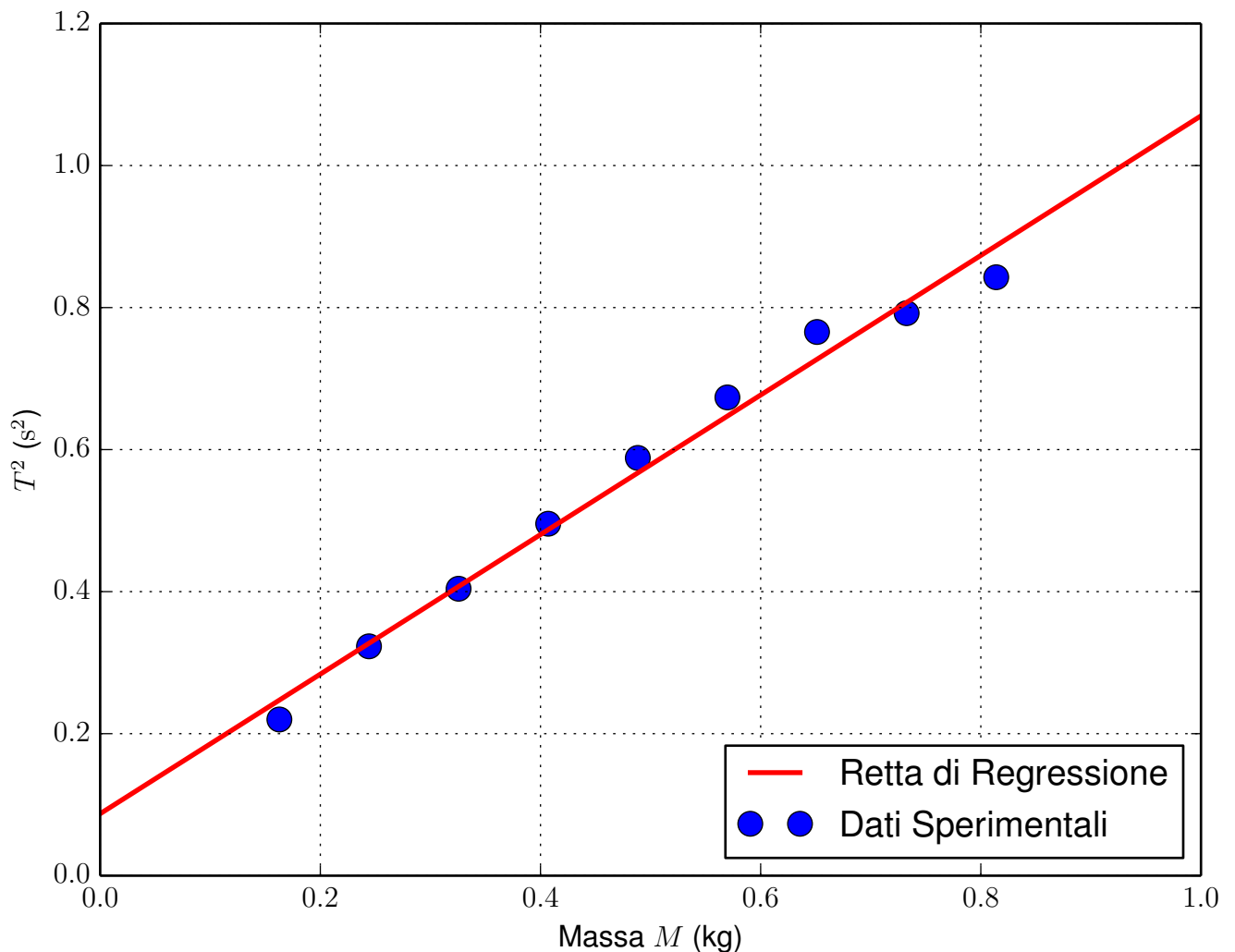


Figura 4: Quadrato dei periodi di oscillazione in funzione della massa

I coefficienti della retta di regressione  $y = A + Bx$ , rappresentata nel grafico di Figura 4, risultano essere:

$$A = 0.0872765740741$$

$$B = 0.982663231413$$

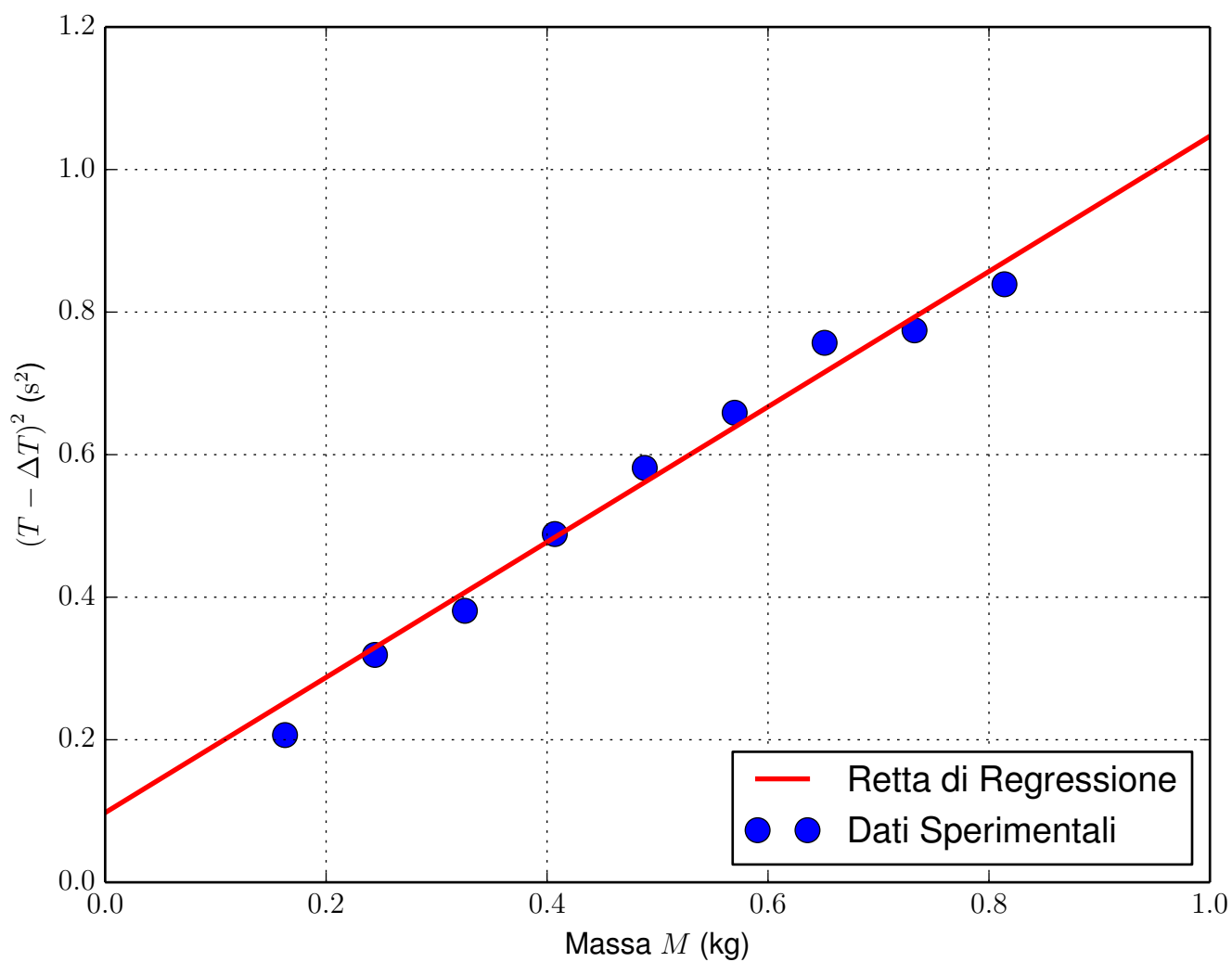


Figura 5: Quadrato dei periodi di oscillazione in funzione della massa

I coefficienti della retta di regressione  $y = A + Bx$ , rappresentata nel grafico di Figura 5, risultano essere:

$$A = 0.0976882632275$$

$$B = 0.949156896344$$

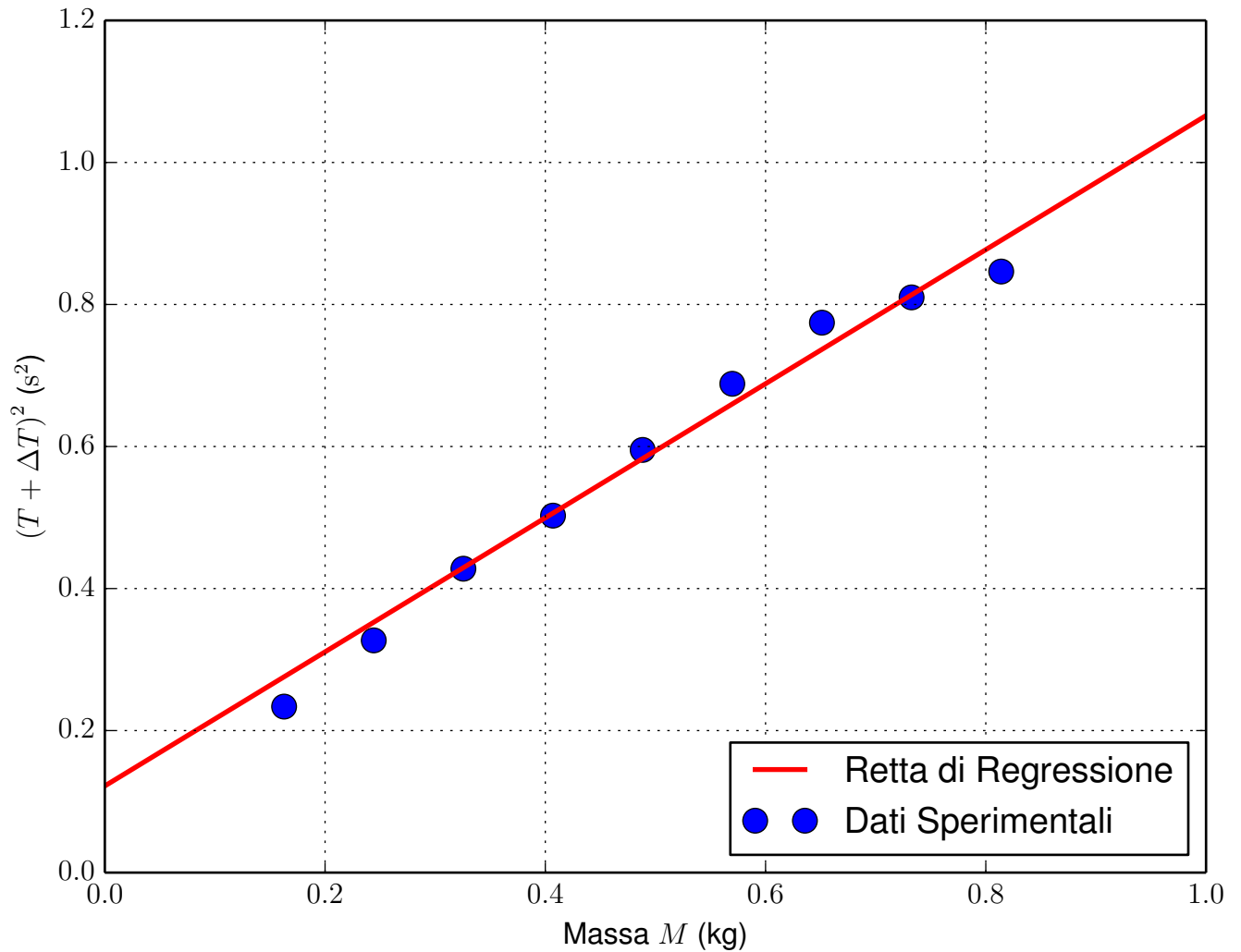


Figura 6: Quadrato dei periodi di oscillazione in funzione della massa

I coefficienti della retta di regressione  $y = A + Bx$ , rappresentata nel grafico di Figura 6, risultano essere:

$$A = 0.122043056878$$

$$B = 0.943966868654$$

#### Task II-6, II-7

Dai risultati ottenuti

$$B_{\max} = 0.949156896344$$

$$B_{\min} = 0.943966868654$$

per cui, in virtù delle formule (17) e (18), i valori di  $k$  e  $\Delta k$  risultano essere:

$$k = 0.0417074837394 \approx 0.04 \frac{\text{N}}{\text{mm}}$$

$$\Delta k = 0.00011434170311 \approx 0.00 \frac{\text{N}}{\text{mm}}$$

Utilizzando la formula (19) il valore di  $m_0$  risulta essere:

$$m_0 = 0.0922044629677 \approx 0.09\text{kg}$$

#### 4 PARTE III

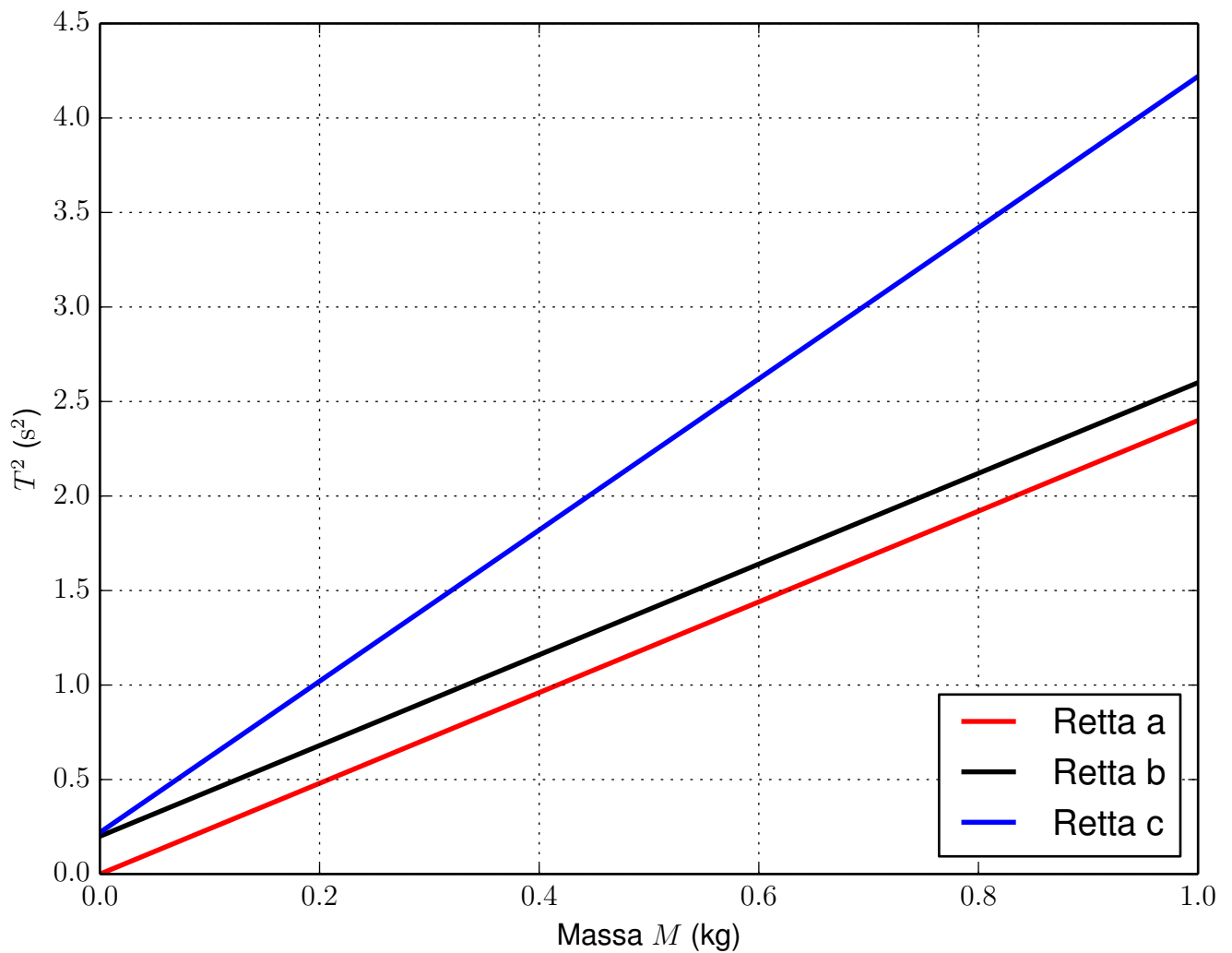


Figura 7:  $T^2$  in funzione della massa del corpo oscillante

##### Task III-1

Osservando la Figura 7, si nota che:

- La retta  $a$  contiene l'origine;
- la retta  $b$  non contiene l'origine, ma è parallela alla retta  $a$ ;
- la retta  $c$  ha un coefficiente angolare  $B$  maggiore delle altre due;
- la retta  $c$  non contiene l'origine.

##### Task III-2

Dall'analisi dei grafici si ottengono i valori di  $k$  e  $m_0$

- Utilizzando la retta  $a$  si ottiene:

$$k = \frac{4\pi^2}{B} = \frac{4\pi^2}{2.4} \approx 16.44 \frac{N}{m} \quad (20)$$

$$m_0 = 0 \quad (21)$$

- Utilizzando la retta  $b$  si ottiene:

$$k = \frac{4\pi^2}{B} = \frac{4\pi^2}{2.4} \approx 16.44 \frac{N}{m} \quad (22)$$

$$m_0 = \frac{Ak}{4\pi^2} = \frac{0.2 \cdot 16.44}{4\pi^2} \approx 0.08 \text{ kg} \quad (23)$$

- Utilizzando la retta  $c$  si ottiene:

$$k = \frac{4\pi^2}{B} = \frac{4\pi^2}{4} \approx 9.86 \frac{N}{m} \quad (24)$$

$$m_0 = \frac{Ak}{4\pi^2} = \frac{0.22 \cdot 9.86}{4\pi^2} \approx 0.05 \text{ kg} \quad (25)$$

## 5 APPENDICE

In questa appendice vengono riportate le espressioni di calcolo dei valori estremi dei parametri della retta di regressione

$$y = A + Bx . \quad (26)$$

Le incertezze  $\Delta A$  e  $\Delta B$  sui parametri della retta si ottengono dalle espressioni:

$$S = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - 2}}$$

$$\Delta B = \frac{S}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad (27a)$$

$$\Delta A = S \sqrt{\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2}} \quad (27b)$$

essendo,

$$\hat{y}_i = Bx_i + A \quad (28)$$

il generico  $\hat{y}_i$  valore atteso di  $y_i$ .

In definitiva, dalle incertezze (27) si ottengono i valori estremi del coefficiente angolare  $B$  e del termine noto  $A$  della retta di regressione

$$B_{\max} = B + \Delta B \quad (29a)$$

$$B_{\min} = B - \Delta B \quad (29b)$$

$$A_{\max} = A + \Delta A \quad (29c)$$

$$A_{\min} = A - \Delta A \quad (29d)$$

## 6 CODICE

### 6.1 Elaborazione Tabella 1

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Apr 23 10:42:12 2016
4
5 @author: pietro
6 """
7 import numpy as np
8 import matplotlib.pyplot as plt
9
10 plt.close("all")
11
12 def cm2mm(list_name):
13     converted_list=[]
14     for element in list_name:
15         element=element*10
16         converted_list.append(element)
17     return converted_list
18 n=2
19 def fprint(a,case):

```

```

20     if case==1:
21         a=round(a,n)
22         formato="{:10.}" + str(n) + "f}"
23         nice_a= formato.format(a)
24         return nice_a
25     if case==0:
26         a=round(a,n)
27         a=np.absolute(a)
28         formato="{:10.}" + str(n) + "f}"
29         nice_a= formato.format(a)
30         return nice_a
31
32
33
34 def linear_reg_coef_error(x,y,m,q):
35     n=len(x)
36     x_medium=np.average(x)
37     y_expec=range(0,n,1)
38     i=0
39     while i<n:
40         y_expec[i]=m*x[i]+q
41         i=i+1
42     x_sum=0
43     j=0
44     while j<n:
45         x_sum=x_sum+(x[j]-x_medium)**2
46         j=j+1
47
48     x_sum1=0
49     s=0
50     while s<n:
51         x_sum1=x_sum1+(x[s])**2
52         s=s+1
53     y_sum=0
54     k=0
55     while k<n:
56         y_sum=y_sum+(y[k]-y_expec[k])**2
57         k=k+1
58     delta_m=((y_sum)/((n-2)*x_sum))**0.5
59     delta_q=((y_sum*x_sum1)/((n-2)*n*x_sum))**0.5
60     error=[delta_m, delta_q]
61     return error
62
63 massa=[]
64 spostamento1_andata=[]
65 spostamento1_ritorno=[]
66 spostamento2_andata=[]
67 spostamento2_ritorno=[]
68 spostamento3_andata=[]
69 spostamento3_ritorno=[]
70
71 def singolo_disco(n):
72     esperienza=range(0,6,1)
73     esperienza[0]=spostamento1_andata[n]
74     esperienza[1]=spostamento1_ritorno[n]
75     esperienza[2]=spostamento2_andata[n]
76     esperienza[3]=spostamento2_ritorno[n]
77     esperienza[4]=spostamento3_andata[n]
78     esperienza[5]=spostamento3_ritorno[n]
79     return esperienza
80
81 f = open("tab1.csv","r")
82
83 #get lines from file
84 linelist=[]

```

```

85 for line in f:
86     linelist.append(line)
87 linelist.pop(0) #remove first line from file
88 for line in linelist:
89     line = line.split(",")
90     massa.append(float(line[1]))
91     spostamento1_andata.append(float(line[2]))
92     spostamento1_ritorno.append(float(line[3]))
93     spostamento2_andata.append(float(line[4]))
94     spostamento2_ritorno.append(float(line[5]))
95     spostamento3_andata.append(float(line[6]))
96     spostamento3_ritorno.append(float(line[7]))
97 #count = len(linelist)
98 f.close()
99 #trasforma i dati inseriti da cm in mm
100 spostamento1_andata= cm2mm(spostamento1_andata)
101 spostamento2_andata= cm2mm(spostamento2_andata)
102 spostamento3_andata= cm2mm(spostamento3_andata)
103 spostamento1_ritorno= cm2mm(spostamento1_ritorno)
104 spostamento2_ritorno= cm2mm(spostamento2_ritorno)
105 spostamento3_ritorno= cm2mm(spostamento3_ritorno)
106
107 disco0=singolo_disco(0)
108 disco1=singolo_disco(1)
109 disco2=singolo_disco(2)
110 disco3=singolo_disco(3)
111 disco4=singolo_disco(4)
112 disco5=singolo_disco(5)
113 disco6=singolo_disco(6)
114 disco7=singolo_disco(7)
115 disco8=singolo_disco(8)
116 disco9=singolo_disco(9)
117 disco10=singolo_disco(10)
118 #calcolo delle medie
119 disco0_media=np.average(disco0)
120 disco1_media=np.average(disco1)
121 disco2_media=np.average(disco2)
122 disco3_media=np.average(disco3)
123 disco4_media=np.average(disco4)
124 disco5_media=np.average(disco5)
125 disco6_media=np.average(disco6)
126 disco7_media=np.average(disco7)
127 disco8_media=np.average(disco8)
128 disco9_media=np.average(disco9)
129 disco10_media=np.average(disco10)
130 disco_medie=[disco0_media , disco1_media , disco2_media , disco3_media , disco4_media , disco5_media ,
    ↪ disco6_media , disco7_media , disco8_media , disco9_media , disco10_media ]
131 # calcolo errori
132 disco0_errore=(np.amax(disco0)-np.amin(disco0))/2
133 disco1_errore=(np.amax(disco1)-np.amin(disco1))/2
134 disco2_errore=(np.amax(disco2)-np.amin(disco2))/2
135 disco3_errore=(np.amax(disco3)-np.amin(disco3))/2
136 disco4_errore=(np.amax(disco4)-np.amin(disco4))/2
137 disco5_errore=(np.amax(disco5)-np.amin(disco5))/2
138 disco6_errore=(np.amax(disco6)-np.amin(disco6))/2
139 disco7_errore=(np.amax(disco7)-np.amin(disco7))/2
140 disco8_errore=(np.amax(disco8)-np.amin(disco8))/2
141 disco9_errore=(np.amax(disco9)-np.amin(disco9))/2
142 disco10_errore=(np.amax(disco10)-np.amin(disco10))/2
143 disco_errori=[disco0_errore , disco1_errore , disco2_errore , disco3_errore , disco4_errore ,
    ↪ disco5_errore , disco6_errore , disco7_errore , disco8_errore , disco9_errore , disco10_errore ]
144 #genera file tex per la tab1
145 file_tex=open("tab1.tex","wb")
146 i=0
147 while i<11:

```

```

148 file_tex.write(str(i)+"&"+str(massa[i])+"&"+fprint(spostamento1_andata[i],1)+"&"+fprint(
↪ spostamento1_ritorno[i],1)+"&"+fprint(spostamento2_andata[i],1)+"&"+fprint(
↪ spostamento2_ritorno[i],1)+"&"+fprint(spostamento3_andata[i],1)+"&"+fprint(
↪ spostamento3_ritorno[i],1)+"&"+fprint(disco_medie[i],1)+"&"+fprint(disco_errori[i],1)+
↪ chr(92)+chr(92)+" \n")
149 i=i+1
150 file_tex.close()
151
152 #genera grafico
153 x=massa
154 y=disco_medie
155 dati_sperimental_i , =plt.plot(x, y, "o", color='b', markersize=10 )
156 m,q=np.polyfit(x, y, 1)
157 plt.grid()
158 plt.xlabel('Massa (Kg)')
159 plt.ylabel('Posizione (mm)')
160 #plt.errorbar(x, y, yerr=20, fmt='o')
161 t = np.arange(0, 1.2, 0.2)
162 #retta di regressione
163 regressione , = plt.plot(t, m*t+q,color='r', linewidth=2.0)
164 regressione_legend = plt.legend([regressione , dati_sperimental_i], ["Retta di Regressione", "Dati
↪ Sperimental_i"])
165 plt.savefig("graf1.pdf")
166 plt.show()
167
168
169 errori_reg=linear_reg_coef_error(x,y,m,q)
170 delta_m=errori_reg[0]
171 m_max=m+delta_m
172 m_min=m-delta_m
173 delta_q=errori_reg[1]
174 q_max=m+delta_q
175 q_min=m-delta_q
176 k=np.absolute(9.81/m)
177
178 #genera file risultati regressione lineare
179 file_tex1=open("regressione_tab1.tex", "wb")
180 file_tex1.write(chr(92)+"begin{align*} \n")
181 file_tex1.write("A &"+fprint(q,1)+"\pm "+fprint(delta_q,1)+chr(92)+chr(92)+" \n")
182 file_tex1.write("B &"+fprint(m,1)+"\pm "+fprint(delta_m,1)+" \n")
183 file_tex1.write("\end{align*} \n")
184 file_tex1.close()
185
186 #genera file k
187 file_tex2=open("coef_rig.tex", "wb")
188 file_tex2.write("$$k="+str(k)+chr(92)+"approx "+fprint(k,1)+" "+chr(92)+" \dfrac{\mathrm{N}}{\}
↪ \mathrm{mm}} $$")
189 file_tex2.close()

```

code/tab1/tab1.py

## 6.2 Elaborazione Tabella 2

```

1 from __future__ import division
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 plt.close("all")
6
7 def cm2mm(list_name):
8     converted_list=[]
9     for element in list_name:
10         element=element*10
11         converted_list.append(element)
12     return converted_list
13 n=2

```

```

14 def fprint(a, case):
15     if case==1:
16         a=round(a,n)
17         formato="{:10.}" + str(n) + "f}"
18         nice_a= formato.format(a)
19         return nice_a
20     if case==0:
21         a=round(a,n)
22         a=np.absolute(a)
23         formato="{:10.}" + str(n) + "f}"
24         nice_a= formato.format(a)
25         return nice_a
26 # http://www.chem.utoronto.ca/coursenotes/analsci/stats/ErrRegr.html
27 def linear_reg_coef_error(x,y,m,q):
28     n=len(x)
29     x_medium=np.average(x)
30     y_expec=range(0,n,1)
31     i=0
32     while i<n:
33         y_expec[i]=m*x[i]+q
34         i=i+1
35     x_sum=0
36     j=0
37     while j<n:
38         x_sum=x_sum+(x[j]-x_medium)**2
39         j=j+1
40
41     x_sum1=0
42     s=0
43     while s<n:
44         x_sum1=x_sum1+(x[s])**2
45         s=s+1
46     y_sum=0
47     k=0
48     while k<n:
49         y_sum=y_sum+(y[k]-y_expec[k])**2
50         k=k+1
51     delta_m=((y_sum)/((n-2)*x_sum))**0.5
52     delta_q=((y_sum*x_sum1)/((n-2)*n*x_sum))**0.5
53     error=[delta_m, delta_q]
54     return error
55
56
57 massa=[]
58 allungamento1_andata=[]
59 allungamento1_ritorno=[]
60 allungamento2_andata=[]
61 allungamento2_ritorno=[]
62 allungamento3_andata=[]
63 allungamento3_ritorno=[]
64
65 def singolo_disco(n):
66     esperienza=range(0,6,1)
67     esperienza[0]=allungamento1_andata[n]
68     esperienza[1]=allungamento1_ritorno[n]
69     esperienza[2]=allungamento2_andata[n]
70     esperienza[3]=allungamento2_ritorno[n]
71     esperienza[4]=allungamento3_andata[n]
72     esperienza[5]=allungamento3_ritorno[n]
73     return esperienza
74
75 f = open("tab2.csv", "r")
76
77 #get lines from file
78 linelist=[]

```

```

79 for line in f:
80     linelist.append(line)
81 linelist.pop(0) #remove first line from file
82 for line in linelist:
83     line = line.split(",")
84     massa.append(float(line[1]))
85     allungamento1_andata.append(float(line[2]))
86     allungamento1_ritorno.append(float(line[3]))
87     allungamento2_andata.append(float(line[4]))
88     allungamento2_ritorno.append(float(line[5]))
89     allungamento3_andata.append(float(line[6]))
90     allungamento3_ritorno.append(float(line[7]))
91 #count = len(linelist)
92 f.close()
93 #trasforma i dati inseriti da cm in mm
94 allungamento1_andata= cm2mm(allungamento1_andata)
95 allungamento2_andata= cm2mm(allungamento2_andata)
96 allungamento3_andata= cm2mm(allungamento3_andata)
97 allungamento1_ritorno= cm2mm(allungamento1_ritorno)
98 allungamento2_ritorno= cm2mm(allungamento2_ritorno)
99 allungamento3_ritorno= cm2mm(allungamento3_ritorno)
100
101 allungamento1_andata_media=np.average(allungamento1_andata)
102 allungamento2_andata_media=np.average(allungamento2_andata)
103 allungamento3_andata_media=np.average(allungamento3_andata)
104 allungamento1_ritorno_media=np.average(allungamento1_ritorno)
105 allungamento2_ritorno_media=np.average(allungamento2_ritorno)
106 allungamento3_ritorno_media=np.average(allungamento3_ritorno)
107
108
109 disco0=singolo_disco(0)
110 disco1=singolo_disco(1)
111 disco2=singolo_disco(2)
112 disco3=singolo_disco(3)
113 disco4=singolo_disco(4)
114 disco5=singolo_disco(5)
115 disco6=singolo_disco(6)
116 disco7=singolo_disco(7)
117 disco8=singolo_disco(8)
118 disco9=singolo_disco(9)
119 disco10=singolo_disco(10)
120
121 #calcolo delle medie
122 disco0_media=np.average(disco0)
123 disco1_media=np.average(disco1)
124 disco2_media=np.average(disco2)
125 disco3_media=np.average(disco3)
126 disco4_media=np.average(disco4)
127 disco5_media=np.average(disco5)
128 disco6_media=np.average(disco6)
129 disco7_media=np.average(disco7)
130 disco8_media=np.average(disco8)
131 disco9_media=np.average(disco9)
132 disco10_media=np.average(disco10)
133 disco_medie=[disco0_media , disco1_media , disco2_media , disco3_media , disco4_media , disco5_media ,
    ↪ disco6_media , disco7_media , disco8_media , disco9_media , disco10_media ]
134
135 #genera file tex tab2.tex
136 file_tex=open("tab2.tex","wb")
137 i=0
138 while i<11:
139     file_tex.write(str(i)+"&"+str(massa[i])+"&"+fprint(allungamento1_andata[i],1)+"&"+fprint(
    ↪ allungamento1_ritorno[i],1)+"&"+fprint(allungamento2_andata[i],1)+"&"+fprint(
    ↪ allungamento2_ritorno[i],1)+"&"+fprint(allungamento3_andata[i],1)+"&"+fprint(
    ↪ allungamento3_ritorno[i],1)+chr(92)+chr(92)+" \n")

```

```

140     i=i+1
141
142 file_tex.write("\hline \n")
143 file_tex.write("\hline \n")
144 file_tex.write("& "+chr(92)+"textbf{Media}&" +fprint(allungamento1_andata_media,1)+"&" +fprint(
    ↪ allungamento1_ritorno_media,1)+"&" +fprint(allungamento2_andata_media,1)+"&" +fprint(
    ↪ allungamento2_ritorno_media,1)+"&" +fprint(allungamento3_andata_media,1)+"&" +fprint(
    ↪ allungamento3_ritorno_media,1)+chr(92)+chr(92)+" \n")
145 file_tex.write("\hline \n")
146 file_tex.close()
147
148
149 #regressione lineare
150 x=massa
151 y=disco_medie
152 dati_sperimentali, =plt.plot(x, y, "o", markersize=10 )
153 x1=[massa[3],massa[4],massa[5],massa[6],massa[7],massa[8],massa[9],massa[10]]
154 y1=[disco3_media,disco4_media,disco5_media,disco6_media,disco7_media,disco8_media,disco9_media,
    ↪ disco10_media]
155 m,q=np.polyfit(x1, y1, 1)
156 t = np.arange(0, 1.2, 0.2)
157 #retta di regressione
158 regressione, = plt.plot(t, m*t+q,color='r', linewidth=2.0)
159 plt.grid()
160 plt.xlabel('Massa (Kg)')
161 plt.ylabel('Allungamento (mm)')
162 regressione_legend = plt.legend([regressione, dati_sperimentali], ["Retta di Regressione","Dati
    ↪ Sperimentali"],loc=4)
163 plt.savefig("graf2.pdf")
164 plt.show()
165 print m,q
166 print linear_reg_coef_error(x1,y1,m,q)

```

code/tab2/tab2.py

### 6.3 Elaborazione Tabella 3

```

1 from __future__ import division
2 import numpy as np
3 import matplotlib.pyplot as plt
4 plt.rcParams['text.usetex'] = True
5 plt.rcParams['text.latex.unicode'] = True
6
7 plt.close('all') #close all plots
8 n=2
9 def fprint(a,case):
10     if case==1:
11         a=round(a,n)
12         formato="{:10.}" +str(n)+"f}"
13         nice_a= formato.format(a)
14         return nice_a
15     if case==0:
16         a=round(a,n)
17         a=np.absolute(a)
18         formato="{:10.}" +str(n)+"f}"
19         nice_a= formato.format(a)
20         return nice_a
21 # http://www.chem.utoronto.ca/coursenotes/analsci/stats/ErrRegr.html
22 def linear_reg_coef_error(x,y,m,q):
23     n=len(x)
24     x_medium=np.average(x)
25     y_expec=range(0,n,1)
26     i=0
27     while i<n:
28         y_expec[i]=m*x[i]+q
29         i=i+1

```

```

30 x_sum=0
31 j=0
32 while j<n:
33     x_sum=x_sum+(x[j]-x_medium)**2
34     j=j+1
35
36 x_sum1=0
37 s=0
38 while s<n:
39     x_sum1=x_sum1+(x[s])**2
40     s=s+1
41 y_sum=0
42 k=0
43 while k<n:
44     y_sum=y_sum+(y[k]-y_expec[k])**2
45     k=k+1
46 delta_m=((y_sum)/((n-2)*x_sum))*0.5
47 delta_q=((y_sum*x_sum1)/((n-2)*n*x_sum))*0.5
48 error=[delta_m, delta_q]
49 return error
50
51
52 massa=[]
53 periodo1=[]
54 periodo2=[]
55 periodo3=[]
56
57 def singolo_disco(n):
58     esperienza=range(0,3,1)
59     esperienza[0]=periodo1[n]
60     esperienza[1]=periodo2[n]
61     esperienza[2]=periodo3[n]
62     return esperienza
63
64 def graph(formula, x_range):
65     x = np.array(x_range)
66     y = eval(formula)
67     plt.plot(x, y, linewidth=3)
68     plt.show()
69
70 f = open("tab3.csv", "r")
71
72 #get lines from file
73 linelist=[]
74 for line in f:
75     linelist.append(line)
76 linelist.pop(0) #remove first line from file
77 for line in linelist:
78     line = line.split(",")
79     massa.append(float(line[1]))
80     periodo1.append(float(line[2]))
81     periodo2.append(float(line[3]))
82     periodo3.append(float(line[4]))
83 #count = len(linelist)
84 f.close()
85
86 #allungamento1_andata_media=np.average(allungamento1_andata)
87 #allungamento2_andata_media=np.average(allungamento2_andata)
88 #allungamento3_andata_media=np.average(allungamento3_andata)
89 #allungamento1_ritorno_media=np.average(allungamento1_ritorno)
90 #allungamento2_ritorno_media=np.average(allungamento2_ritorno)
91 #allungamento3_ritorno_media=np.average(allungamento3_ritorno)
92
93 periodi_disco0=singolo_disco(0)
94 periodi_disco1=singolo_disco(1)

```

```

95 periodi_disco2=singolo_disco(2)
96 periodi_disco3=singolo_disco(3)
97 periodi_disco4=singolo_disco(4)
98 periodi_disco5=singolo_disco(5)
99 periodi_disco6=singolo_disco(6)
100 periodi_disco7=singolo_disco(7)
101 periodi_disco8=singolo_disco(8)
102 #periodi_disco9=singolo_disco(9)
103
104 periodo_disco0_max=np.amax(periodi_disco0)
105 periodo_disco1_max=np.amax(periodi_disco1)
106 periodo_disco2_max=np.amax(periodi_disco2)
107 periodo_disco3_max=np.amax(periodi_disco3)
108 periodo_disco4_max=np.amax(periodi_disco4)
109 periodo_disco5_max=np.amax(periodi_disco5)
110 periodo_disco6_max=np.amax(periodi_disco6)
111 periodo_disco7_max=np.amax(periodi_disco7)
112 periodo_disco8_max=np.amax(periodi_disco8)
113 #periodo_disco9_max=np.amax(periodi_disco9)
114 periodi_max=[periodo_disco0_max, periodo_disco1_max, periodo_disco2_max, periodo_disco3_max,
    ↪ periodo_disco4_max, periodo_disco5_max, periodo_disco6_max, periodo_disco7_max,
    ↪ periodo_disco8_max]
115
116 periodo_disco0_min=np.amin(periodi_disco0)
117 periodo_disco1_min=np.amin(periodi_disco1)
118 periodo_disco2_min=np.amin(periodi_disco2)
119 periodo_disco3_min=np.amin(periodi_disco3)
120 periodo_disco4_min=np.amin(periodi_disco4)
121 periodo_disco5_min=np.amin(periodi_disco5)
122 periodo_disco6_min=np.amin(periodi_disco6)
123 periodo_disco7_min=np.amin(periodi_disco7)
124 periodo_disco8_min=np.amin(periodi_disco8)
125 #periodo_disco9_min=np.amin(periodi_disco9)
126 periodi_min=[periodo_disco0_min, periodo_disco1_min, periodo_disco2_min, periodo_disco3_min,
    ↪ periodo_disco4_min, periodo_disco5_min, periodo_disco6_min, periodo_disco7_min,
    ↪ periodo_disco8_min]
127 #calcolo delle medie
128 periodo_disco0=np.average(periodi_disco0)
129 periodo_disco1=np.average(periodi_disco1)
130 periodo_disco2=np.average(periodi_disco2)
131 periodo_disco3=np.average(periodi_disco3)
132 periodo_disco4=np.average(periodi_disco4)
133 periodo_disco5=np.average(periodi_disco5)
134 periodo_disco6=np.average(periodi_disco6)
135 periodo_disco7=np.average(periodi_disco7)
136 periodo_disco8=np.average(periodi_disco8)
137 #periodo_disco9=np.average(periodi_disco9)
138
139 periodi10=[periodo_disco0, periodo_disco1, periodo_disco2, periodo_disco3, periodo_disco4,
    ↪ periodo_disco5, periodo_disco6, periodo_disco7, periodo_disco8]
140
141 periodi=range(0,9,1)
142 t=0
143 while t<len(periodi):
144     periodi[t]=periodi10[t]/10
145     t=t+1
146
147 periodi_quadrato=range(0,9,1)
148 i=0
149 while i<len(periodi_quadrato):
150     periodi_quadrato[i]=periodi[i]**2
151     i=i+1
152
153 periodi_errori_10=range(0,9,1)
154 k=0

```

```

155
156 while k<len(periodi_errori_10):
157     periodi_errori_10[k]=(periodi_max[k]-periodi_min[k])/2
158     k=k+1
159
160 periodi_errori=range(0,9,1)
161 k=0
162
163 while k<len(periodi_errori):
164     periodi_errori[k]=(periodi_max[k]-periodi_min[k])/20
165     k=k+1
166
167 periodi_plus_errori=range(0,9,1)
168 periodi_plus_errori1=range(0,8,1)
169 j=0
170 while j<len(periodi_plus_errori):
171     periodi_plus_errori[j]=(periodi[j]+periodi_errori[j])**2
172     j=j+1
173 j=0
174 while j<len(periodi_plus_errori1):#viene eliminato l'ultimo valore per la regressione
175     periodi_plus_errori1[j]=(periodi[j]+periodi_errori[j])**2
176     j=j+1
177
178 periodi_minus_errori=range(0,9,1)
179 periodi_minus_errori1=range(0,8,1)
180 j=0
181 while j<len(periodi_minus_errori):
182     periodi_minus_errori[j]=(periodi[j]-periodi_errori[j])**2
183     j=j+1
184 j=0
185 while j<len(periodi_minus_errori1): #viene eliminato l'ultimo valore per la regressione
186     periodi_minus_errori1[j]=(periodi[j]-periodi_errori[j])**2
187     j=j+1
188 j=0
189 massa1=range(0,8,1)#viene eliminato l'ultimo valore per la regressione
190 while j<len(massa1):
191     massa1[j]=massa[j]
192     j=j+1
193
194
195 #genera file tex tab3.tex
196 file_tex=open("tab3.tex","wb")
197 i=0
198 while i<9:
199     v=9-i
200     file_tex.write(str(v)+"&"+str(massa[i])+"&"+fprint(periodo1[i],1)+"&"+fprint(periodo2[i],1)
↪ +"&"+fprint(periodo3[i],1)+"&"+fprint(periodi[i],1)+"&"+fprint(periodi_quadrato[i],1)+"&
↪ "+fprint(periodi_errori_10[i],1)+"&"+fprint(periodi_errori[i],1)+"&"+fprint(
↪ periodi_plus_errori[i],1)+"&"+fprint(periodi_minus_errori[i],1)+chr(92)+chr(92)+" \n")
201     i=i+1
202 file_tex.close()
203
204 #grafico 3.1
205 plt.figure()
206 dati_sperimentalil1, = plt.plot(massa, periodi_minus_errori, "o", markersize=10) #(x,y)
207 m1,q1=np.polyfit(massa1, periodi_minus_errori1, 1)
208 t = np.arange(0, 1.2, 0.2)
209 regressione1, = plt.plot(t, m1*t+q1,color='r', linewidth=2.0)
210 plt.grid()
211 plt.xlabel(r'Massa $M$ (kg)')
212 plt.ylabel(r'$\left(T- \Delta T \right)^2$ ($\mathrm{s}^2$)')
213 regressione_legend1 = plt.legend([regressione1, dati_sperimentalil1], ["Retta di Regressione",
↪ Dati Sperimentalil1"],loc=4)
214 plt.savefig('grafico31.pdf')
215 plt.show()

```

```

216 #grafico 3.2
217 plt.figure()
218 dati_sperimental2, = plt.plot(massa, periodi_plus_errori, "o", markersize=10 ) #(x,y)
219 m2,q2=np.polyfit(massa1, periodi_plus_errori1, 1)
220 t = np.arange(0, 1.2, 0.2)
221 regressione2, = plt.plot(t, m2*t+q2,color='r', linewidth=2.0)
222 plt.grid()
223 plt.xlabel(r'Massa $M$ (kg)')
224 plt.ylabel(r'$\left(T+ \Delta T \right)^2$ ($\mathrm{s}^2$)')
225 regressione_legend2 = plt.legend([regressione2, dati_sperimental2], ["Retta di Regressione",
↪ "Dati Sperimentali"],loc=4)
226 plt.savefig('grafico32.pdf')
227 plt.show()
228
229 #grafico 3.3
230 plt.figure()
231 dati_sperimental3, =plt.plot(massa, periodi_quadrato, "o", markersize=10 ) #(x,y)
232 m,q=np.polyfit(massa, periodi_quadrato,1)
233 t = np.arange(0, 1.2, 0.2)
234 regressione, = plt.plot(t, m*t+q,color='r', linewidth=2.0)
235 plt.grid()
236 plt.xlabel(r'Massa $M$ (kg)')
237 plt.ylabel(r'$T^2$ ($\mathrm{s}^2$)')
238 regressione_legend3 = plt.legend([regressione, dati_sperimental3], ["Retta di Regressione",
↪ "Dati Sperimentali"],loc=4)
239 plt.savefig('grafico33.pdf')
240 plt.show()
241
242
243 #genera file csv per grafico 3.1
244 csvfile1=open("grafico31.csv","wb")
245 i=0
246 while i<len(massa):
247     csvfile1.write(str(massa[i])+",""+str(periodi_minus_errori[i])+" \n")
248     i=i+1
249 csvfile1.close()
250
251 #genera file csv per grafico 3.2
252 csvfile2=open("grafico32.csv","wb")
253 i=0
254 while i<len(massa):
255     csvfile2.write(str(massa[i])+",""+str(periodi_plus_errori[i])+" \n")
256     i=i+1
257 csvfile2.close()
258
259
260 print m1
261 print m2
262 pi=np.pi
263 k=(2*pi**2)*((m1+m2)/(m1*m2))
264 # Trasforma k da N/m a N/mm
265 k=k/1000
266 deltak=np.absolute((2*pi**2)*((m1-m2)/(m1*m2)))
267 deltak=deltak/1000
268
269 #genera file k e delta k
270 file_k=open("filek.tex","wb")
271 file_k.write(chr(92)+"begin{align}")
272 file_k.write("k&="+str(k)+chr(92)+"approx "+fprint(k,1)+"\dfrac{\mathrm{N}}{\mathrm{mm}}"+chr(92)+"
↪ nonumber "+chr(92)+chr(92))
273 file_k.write("\Delta k&="+str(deltak)+chr(92)+"approx"+fprint(deltak,1)+"\dfrac{\mathrm{N}}{\mathrm{mm}}
↪ "+chr(92)+"nonumber ")
274 file_k.write(chr(92)+"end{align}")
275 file_k.close()
276

```

```

277 print k
278 print deltak
279 print q
280 print q1
281 print q2
282 m0 = (q*k*1000)/(4*pi**2)
283
284 b=[m1,m2]
285 b_max=np.amax(b)
286 b_min=np.amin(b)
287
288 #genera file b
289
290 file_b=open("fileb.tex","wb")
291 file_b.write(chr(92)+"begin{align}")
292 file_b.write("B_{\max}="+str(b_max)+chr(92)+"nonumber"+chr(92)+chr(92))
293 file_b.write("B_{\min}="+str(b_min)+chr(92)+"nonumber")
294 file_b.write(chr(92)+"end{align}")
295 file_b.close()
296
297 #genera file m_0
298
299 file_m=open("filem.tex","wb")
300 file_m.write(chr(92)+"begin{equation*}")
301 file_m.write("m_0="+str(m0)+chr(92)+"approx "+fprint(m0,1)+"\mathrm{kg}")
302 file_m.write(chr(92)+"end{equation*}")
303 file_m.close()
304
305 #genera file regressione
306 file_regressione1=open("regressione31.tex","wb")
307 file_regressione1.write(chr(92)+"begin{align}")
308 file_regressione1.write("A&="+str(q1)+chr(92)+"nonumber "+chr(92)+chr(92))
309 file_regressione1.write("B&="+str(m1)+chr(92)+"nonumber ")
310 file_regressione1.write(chr(92)+"end{align}")
311 file_regressione1.close()
312
313 file_regressione2=open("regressione32.tex","wb")
314 file_regressione2.write(chr(92)+"begin{align}")
315 file_regressione2.write("A&="+str(q2)+chr(92)+"nonumber "+chr(92)+chr(92))
316 file_regressione2.write("B&="+str(m2)+chr(92)+"nonumber ")
317 file_regressione2.write(chr(92)+"end{align}")
318 file_regressione2.close()
319
320
321 file_regressione3=open("regressione33.tex","wb")
322 file_regressione3.write(chr(92)+"begin{align}")
323 file_regressione3.write("A&="+str(q)+chr(92)+"nonumber "+chr(92)+chr(92))
324 file_regressione3.write("B&="+str(m)+chr(92)+"nonumber ")
325 file_regressione3.write(chr(92)+"end{align}")
326 file_regressione3.close()
327 #
328 #
329 ##regressione lineare
330 #x=massa
331 #y=disco_medie
332 #plt.plot(x, y, "o", markersize=10 )
333 #x1=[massa[3],massa[4],massa[5],massa[6],massa[7],massa[8],massa[9],massa[10]]
334 #y1=[disco3_media,disco4_media,disco5_media,disco6_media,disco7_media,disco8_media,disco9_media
    ↵ ,disco10_media]
335 #m,q=np.polyfit(x1, y1, 1)
336 #graph(str(m)+'*x'+str(q), range(0, 2))
337 #plt.grid()
338 #plt.show()
339 #print m
340 #print q

```

```
341 #print linear_reg_coef_error(x1,y1,m,q)
```

code/tab3/tab3.py