

note-sul-terzo-laboratorio-telematico-del_270420.pptx

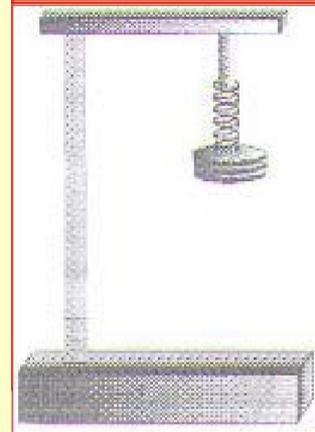


Nel link viene descritta una esperienza di tipo manuale con una molla elicoidale in acciaio, collegata ad un estremo ad un supporto fisso e all'altro estremo collegata con dei carichi attaccati aventi diversa massa .

Lo scopo è la misura di **g**

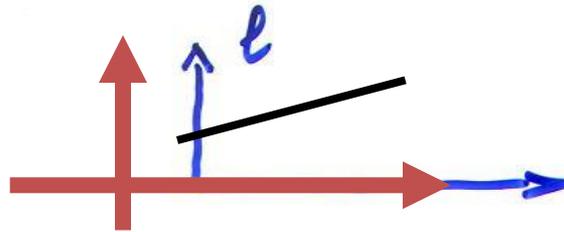
Strategia:

- Studio degli allungamenti: L vs. la massa (Regime statico: K_s);
- Studio delle oscillazioni: T^2 vs. la massa (Regime dinamico: K_d);
- Controllo delle sistematiche tramite il confronto tra K_s e K_d ;
- Superato il test di compatibilità tra K_s e K_d , si può procedere a:
 - Ricavare la migliore stima per K usando entrambe le misure della costante elastica e la media pesata;
 - Dalle misure statiche e dinamiche si può misurare g e confrontarlo con il valore atteso.



Suggerimenti strategici per le misurazioni:

- In generale, si suggerisce di **ripetere più volte le singole misure** (per esempio 20 volte sia quelle dell'allungamento, sia quelle del periodo, a parità di massa "appesa").
- Nella misura del periodo di oscillazione viene suggerito di **misurare la durata di più oscillazioni**, per esempio 10.

STATICAMENTE: allungamentoSTATICO

$$l = l_0 + (g / K) m$$

$$y = a + b x$$

Termine noto a , $\sigma(a)$ \rightarrow l_0 , $\sigma(l_0)$

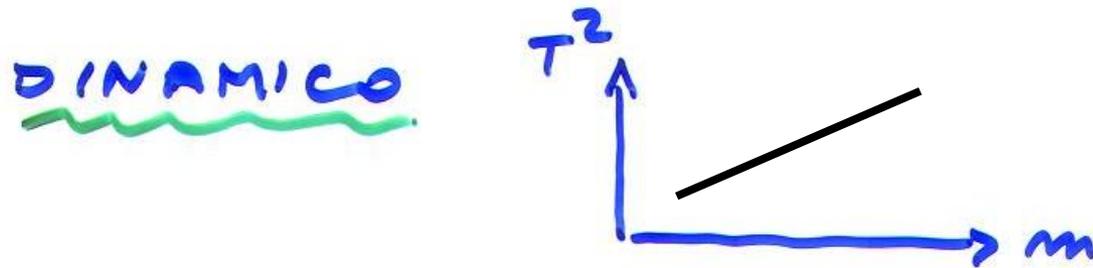
Slope b , $\sigma(b)$ \rightarrow K , $\sigma(k)$

$$\sigma^2(g/k) = (d (g/K) / dK)^2 \sigma^2(k) = (g^2 / k^4) \sigma^2(k)$$

$$\rightarrow \sigma(k) = (K^2 / g) \sigma(g/k)$$

\rightarrow "eliminare" misure nella prima tone
non lineare prima di fare il "fit"

DINAMICAMENTE: oscillazione



$$T = 2\pi \sqrt{\frac{(m+m_0)}{k}}$$

$$T^2 = 4\pi^2 \cdot \frac{m+m_0}{k} = \frac{4\pi^2}{k} \cdot m + \frac{4\pi^2 m_0}{k}$$

$$y = b x + a$$

$$\sigma^2\left(\frac{4\pi^2}{k}\right) = \left(-\frac{4\pi^2}{k^2}\right)^2 \sigma^2(k) \rightarrow \sigma(k) = \frac{k^2}{4\pi^2} \cdot \sigma\left(\frac{4\pi^2}{k}\right)$$

Determinazione dell'accelerazione di gravità: "g"

- Allungamento: migliore retta

$$\rightarrow \text{slope1} = (g / k) \pm \sigma(\text{slope1})$$

- Oscillazione: migliore retta

$$\rightarrow \text{slope2} = (4 \pi^2 / k) \pm \sigma(\text{slope2})$$

$$\frac{\text{slope1}}{\text{slope2}} = \frac{\left(\frac{g}{K}\right)}{\left(\frac{4\pi^2}{K}\right)} = \frac{g}{4\pi^2} \Rightarrow g = (4\pi^2) \left(\frac{\text{slope1}}{\text{slope2}}\right)$$

$$\dots \left(\frac{\sigma(g)}{g}\right) = \sqrt{\left(\frac{\sigma(\text{slope1})}{\text{slope1}}\right)^2 + \left(\frac{\sigma(\text{slope2})}{\text{slope2}}\right)^2}$$

→ Sommaro sui minimi quadrati [1/2]

$$y = a + bx$$

con solo σ_y
costante per ogni y

Termine
noto

$$a = \frac{(\sum x^2)(\sum y) - (\sum x)(\sum xy)}{N(\sum x^2) - (\sum x)^2} = \frac{\langle x^2 \rangle \langle y \rangle - \langle x \rangle \langle xy \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

Coefficiente
angolare
"SLOPE"

$$b = \frac{N(\sum xy) - (\sum x)(\sum y)}{N(\sum x^2) - (\sum x)^2} = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

$$\Delta = N(\sum x^2) - (\sum x)^2 = N^2 (\langle x^2 \rangle - \langle x \rangle^2)$$

[2/2]

$$\sigma_a = \sigma_y \sqrt{\frac{\sum x^2}{\Delta}} = \frac{\sigma_y}{\sqrt{N}} \cdot \sqrt{\frac{\langle x^2 \rangle}{(\langle x^2 \rangle - \langle x \rangle^2)}}$$

$$\sigma_b = \sigma_y \sqrt{\frac{N}{\Delta}} = \frac{\sigma_y}{\sqrt{N}} \cdot \frac{1}{\sqrt{(\langle x^2 \rangle - \langle x \rangle^2)}}$$

$$\sigma_{(a,b)} = - \frac{\sigma^2(y) \cdot \sum_{i=1}^N x_i}{N \cdot \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} = - \frac{\sigma^2(y)}{N} \cdot \frac{\langle x \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

$$|\sigma_{(a,b)}| < \sigma_a \cdot \sigma_b$$

$$\sigma(y = a + bx) = \sqrt{x^2 \cdot \sigma_b^2 + \sigma_a^2 + 2x \sigma_{(a,b)}}$$

Esperienza: molla simulata (Proff. A. Messina e F. Santanastasio)

Si tratta di seguire gli stessi passi dell'esperienza della molla condotta in maniera manuale usando però dati simulati tramite codice **Python**.

Tramite una **classe di python** vengono generati dei numeri casuali (**misure**) secondo il modello di fisica della molla appesa, riproducendo le caratteristiche degli strumenti e la precisione di ogni misura.

I valori numerici dei parametri sono regolati in base alle esperienze vere svolte in laboratorio degli anni passati.

In particolare:

- **SimMolla.py** : e' la classe che contiene i metodi;
- **SimulatoreMolla.ipynb** : e' lo **jupyter notebook** con le istruzioni per eseguire il simulatore e con qualche esempio per gestire i dati.

Si possono semplicemente trascrivere i risultati della misura su di un foglio (come si farebbe in laboratorio) per poi procedere "off-line" alla loro analisi.

Alternativamente, si possono usare direttamente le funzioni di Python per raccogliere dati in tabelle, vettori, etc.. su cui effettuare l'analisi.



```
In [2]: #plots will be shown inline
        %matplotlib inline

        import numpy as np
        import matplotlib.pyplot as plt
        import matplotlib.mlab as mlab
        from scipy import stats
        import pandas as pd
        import math
        from SimMolla import SimMolla
```

Istruzioni sull'uso del simulatore di una molla

```
In [4]: # Crea l'oggetto "simulatore della molla"  
sim = SimMolla()
```

```
# Istruzioni sull'uso del simulatore
```

massa

```
# MisuraMassa(numero pesetti, numero di misure ripetute) ← ≤ 10  
# unità di misura = grammi [g], digit = 0.1 g  
# l'output è un numpy array  
# es. misura la massa di 3 pesetti, 7 misure ripetute  
m = sim.MisuraMassa(3,7)
```

Lunghezza
di
equilibrio

```
# MisuraLunghezza(numero pesetti, numero di misure ripetute) ← ≤ 10  
# unità di misura = millimetri [mm], digit = 1 mm  
# l'output è un numpy array  
# es. misura la lunghezza della molla con 4 pesetti appesi, 6 misure ripetute  
xeq = sim.MisuraLunghezza(4,6)
```

Durata
di n
oscillazioni

```
# MisuraPeriodo(numero pesetti, numero oscillazioni, numero di misure ripetute) ← ≤ 40  
# unità di misura = secondi [s], digit = 0.01 s  
# l'output è un numpy array  
# es. misura il periodo di 4 oscillazioni (tempo totale) con 8 pesetti appesi, 15 misure ripetute  
T = sim.MisuraPeriodo(8,4,15)
```

```
[240.2 239.8 239.8 239.9 239.8 240.2 239.7] ---- unita' di misura = g  
[109. 111. 109. 110. 109. 109.] ---- unita' di misura = mm  
[2.94 2.91 2.92 2.92 2.94 2.95 3.01 2.94 2.99 2.87 2.99 2.96 3. 2.95  
2.91] ---- unita' di misura = s
```

```
In [11]: sim = SimMolla()
         xeq = sim.MisuraLunghezza(4,10)
         np.savetxt('out_measure_xeq.txt', xeq, fmt='%.1f')
```

```
[110. 109. 109. 110. 110. 109. 110. 109. 109. 110.] ---- unita' di misura = mm
```

```
In [4]: sim = SimMolla()
        T = sim.MisuraPeriodo(8,4,15)
```

```
[2.89 2.92 3.06 2.99 3.02 3.01 2.95 2.94 3.01 2.99 2.87 2.98 2.97 2.91
 2.94] ---- unita' di misura = s
```

Possibili valori con cui confrontare la misura sperimentale di g:

	$g \text{ (m/s}^2\text{)}$
Giove	24.79
Terra	9.807
Marte	3.711
Mercurio	3.697
Io	1.79
Luna	1.62
Europa	1.315
Callisto	1.236
Cerere	0.27

Download Anaconda-Navigator

- Link per scaricare l'ultima edizione di Anaconda:
<https://www.anaconda.com/distribution/#download-section>
- Selezionare il sistema operativo utilizzato tra i tre disponibili:

Scaricare I pacchetti relativi alla versione di Python 3.7

Windows | macOS | Linux

Anaconda 2020.02 for Windows Installer

Python 3.7 version	Python 2.7 version
Download	Download
64-Bit Graphical Installer (466 MB) 32-Bit Graphical Installer (423 MB)	64-Bit Graphical Installer (413 MB) 32-Bit Graphical Installer (356 MB)

Per sistemi Windows il pacchetto è disponibile sia per sistemi con architettura a 32-Bit e 64-Bit.

Dopo aver scaricato il pacchetto corrispondente al sistema operativo, è possibile installare Anaconda-navigator seguendo i passaggi mostrati ai seguenti link.

- Windows:

<https://docs.anaconda.com/anaconda/install/windows/>

- Linux:

<https://docs.anaconda.com/anaconda/install/linux/>

- MacOS:

<https://docs.anaconda.com/anaconda/install/mac-os/>

Per I Mac ci sono due diversi metodi di installazione a seconda che si sia scaricato il pacchetto grafico o a linea di comando.

Terminata l'installazione, tramite Anaconda-navigator si possono lanciare altri software come Spyder e Jupiter-notebook, utili per editare ed eseguire gli script in python.

Fortemente suggerito:

Procedere con l'installazione del software

ANACONDA 2020.02

sul proprio PC **durante questo fine settimana**

Lunedì 27/4/20 riceverete il link per scaricare i due file da inserire in una cartella visibile da parte di

JUPYTER notebook

PS

In caso di problemi di installazione potete contattare il Dr. Federico radiconi via mail:

federico.radiconi@roma1.infn.it