Lab report 1: Estimating a spring constant

Written by Jonathan Tybirk, s216136

Purpose

In this experiment, I will attempt to estimate the spring constant of a specific spring with Hooke's law $F_S = kx$. It states that length displacement of the spring x will scale with the force pulling on it F_S by the spring constant k.

By hanging an iOLab plus additional weight from a spring, I can measure the force pulling on the it. I will then measure the length of the spring with a ruler, and find the relation between the two by applying linear regression.

Experimental design

The iOLab is suspended by the spring attached to a stand while additional weights hang beneath the iOLab, as seen in figure 1. As the iOLab and weights pull on the spring, the spring pulls back with force F_s which is measured by the iOLab.

As we are interested only in how the length of the spring scales with F_S, I will for simplicity measure the whole spring as x. This will affect the intercept of the linear regression, but not the slope, which is the k value we are interested in.

To vary the force applied to the spring, I added to the force pulling on the spring by attaching weights of mass w under the iOLab. For $w = 0, 20, 40, 60, 80, \text{and } 100 \text{ grams}$, I measured F_S ten times and x once.

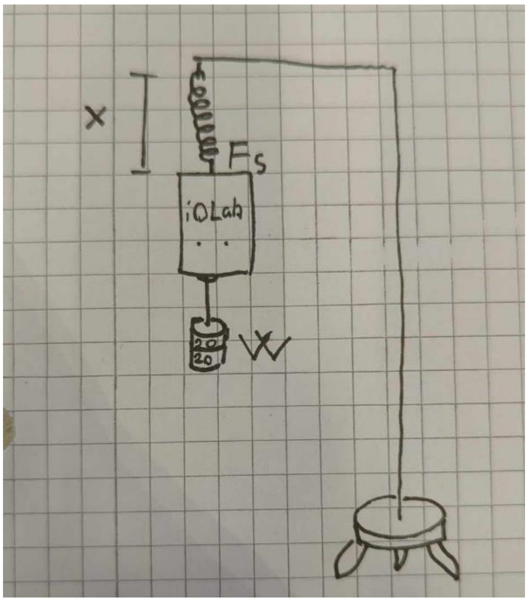


Figure 1: sketch of the experiment

Data

x was measured with a ruler. As this was done freehand with some parallax, I estimate an error of ± 0.5 cm for each measurement.

For each weight, the iOLab measured the average pulling force over 1 second. I did ten of these 1-second measurements per weight and computed the mean, standard deviation, and standard error of the mean, which is used for the uncertainty. Figure 2 shows how one such computation was done with Python.

```
w2 = 20
x2 = unc.ufloat(23.0, 0.5)

data2 = np.array([-2.387, -2.389, -2.386, -2.393, -2.388, -2.390, -2.396, -2.390, -2.387, -2.392])

F2 = unc.ufloat(np.mean(data2), np.std(data2) / np.sqrt(len(data2)))

display(w2, x2, F2)
[114]  ✓ 0.0s
...  20
...  23.0+/-0.5
...  -2.3898+/-0.0009359487165437905
```

Figure 2: computations of the measured data when $w = 20$ grams

I did not estimate the uncertainty of w, as the added weight is not directly relevant to the computations but merely lets me measure x at different pulling forces. Whether the weights are exactly in increments of 20 grams does not matter for the regression.

Data table

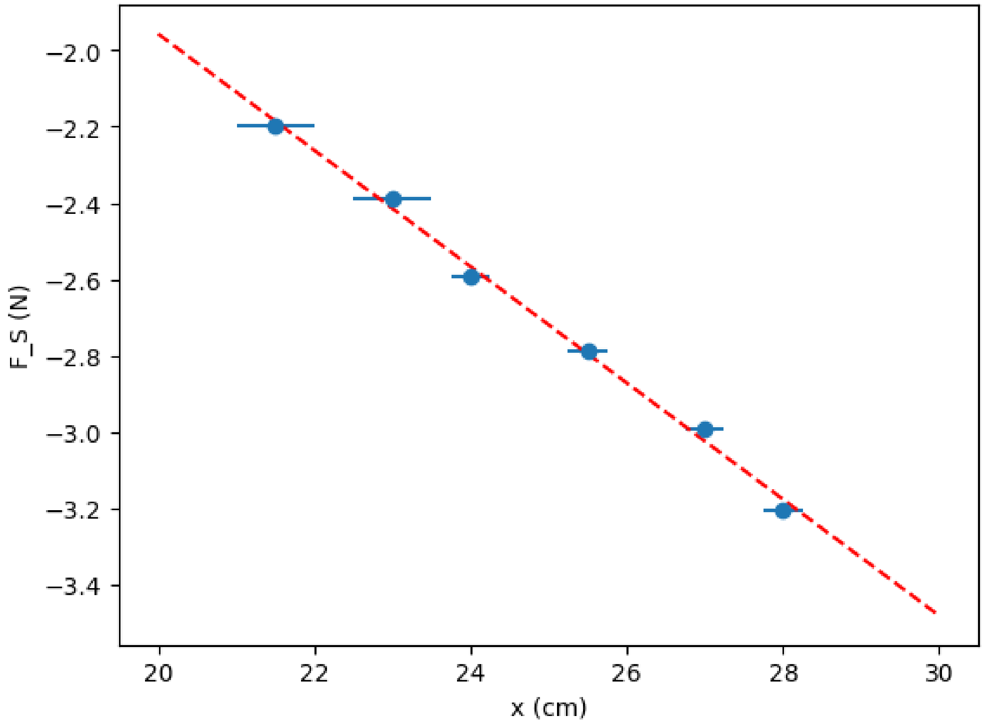
$\boldsymbol{w}$ (g)	$\boldsymbol{F_S}$ (N)	$\boldsymbol{x}$ (cm)
0	-2.1976 ± 0.00025	21.5 ± 0.5
20	-2.3898 ± 0.00093	23.0 ± 0.5
40	-2.5913 ± 0.0042	24.0 ± 0.5
60	-2.7881 ± 0.00048	25.0 ± 0.5
80	-2.9943 ± 0.0020	27.0 ± 0.5
100	-3.2039 ± 0.00033	28.0 ± 0.5

Analysis

If the data indeed follows Hooke's law, a linear regression on the data should fit all six datapoints well. In Python, I computed the best fit for a linear model $y = ax + b$, where y is F_S, a is k, and x is x. Converting to SI units, the parameters came out to $k = (-15.3 \pm 0.5)\,\frac{\mathrm{N}}{\mathrm{m}}$ and $b = (1.09 \pm 0.12)\ \mathrm{N}$.

The intercept is nonzero, but this is expected as I used the full length of the spring for x instead of the length displacement.

Beneath, the datapoints are plotted along with their uncertainties in blue and the regression line in red. The horizontal error bar represents the uncertainty of x. The vertical $\mathrm{F_S}$ standard error of the mean is too small to see.



Conclusion

I determined the spring constant of the specific spring to be $k = (-15.3 \pm 0.5)\,\frac{\mathrm{N}}{\mathrm{m}}$