FYS2150
Lab Report: Elasticity

Nicholas Karlsen
(Dated: March 31, 2018)

A study on different methods to determine the Young's modulus of a brass rod.

I. INTRODUCTION

II. THEORY

III. EXPERIMENTAL PROCEDURE

IV. RESULTS

V. DISCUSSION

VI. CONCLUSION

CODE

All of the code used to produce this report. Anything noteworthy should already be mentioned in the main body of the paper.

scripts/lab_data.py

```python
from numpy import *
import FYS2150lib as fys


def weights():
    # Mass of weights measured with balance
    m_a_balance = 500.1e-3
    m_b_balance = 1000.3e-3
    m_c_balance = 2000.5e-3

    # Mass of reference weights
    m_reference = array([0.5, 1.0, 2.0])
    m_reference_balance = array([500.0e-3, 999.9e-3, 2000.1e-3])  # Weighed

    a, b, da, db = fys.linfit(m_reference, m_reference_balance)

    m_a = (m_a_balance - b) / a
    m_b = (m_b_balance - b) / a
    m_c = (m_c_balance - b) / a

    return m_a, m_b, m_c
```

scripts/FYS2150lib.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
A collection of commonly used functions in FYS2150.
author: Nicholas Karlsen
"""
import numpy as np

```

```

def stddev(x):
    """
    Finds the standard deviation, and standard deviation of
    a 1D array of data x.
    See. Eqn D. Page 24 squires
    """
    n = len(x)
    sigma = np.sqrt((np.sum(x**2) - 1.0 / n * np.sum(x)**2) / (n - 1))
    sigma_m = np.sqrt((np.sum(x**2) - 1.0 / n * np.sum(x)**2) / (n * (n - 1)))

    return sigma, sigma_m


def linfit(x, y):
    """
    Finds the line of best-fit in the form y=mx+c given two
    1D arrays x and y.
    """
    n = np.size(y)
    D = np.sum(x**2) - (1.0 / n) * np.sum(x)**2
    E = np.sum(x * y) - (1.0 / n) * np.sum(x) * np.sum(y)
    F = np.sum(y**2) - (1.0 / n) * np.sum(y)**2

    dm = np.sqrt(1.0 / (n - 2) * (D * F - E**2) / D**2)
    dc = np.sqrt(1.0 / (n - 2) * (float(D) / n + np.mean(x)) *
                 ((D * F - E**2) / (D**2)))
    m = float(E) / D
    c = np.mean(y) - m * np.mean(x)

    return m, c, dm, dc
```