

pylab1

January 25, 2020

0.1 PyLab 1

Jeff Shen | 1004911526

Stacy Ossipov | 1004877779

27 January 2020

```
[1]: # imports
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.optimize import curve_fit
import pandas as pd
```

```
[2]: # read data (using dataframe because it's easier)
data = pd.read_csv('data2.csv')

# convert current and uncertainty from mA to A
data.current /= 1000
data.uncurrent /= 1000
```

```
[3]: # reduced chi squared
def rcs(pred, target, uncertainty, n_params):
    return np.square((pred - target) / uncertainty).sum() / (pred.size -
    ↪n_params)

# fit function
def model(x, a, b):
    return a*x + b

# fit function where line passes through 0
def model2(x, a):
    return a*x
```

```
[4]: # making plot bigger
plt.figure(figsize=(16, 12))
# initializing array to store output
params = np.zeros((4, 8))
```

```

# loop over all resistors
for i in np.unique(data.resistorid):
    # filter data for this resistor
    current = data[data.resistorid == i].current
    voltage = data[data.resistorid == i].voltage
    ucurrent = data[data.resistorid == i].ucurrent
    uvoltage = data[data.resistorid == i].uvoltage

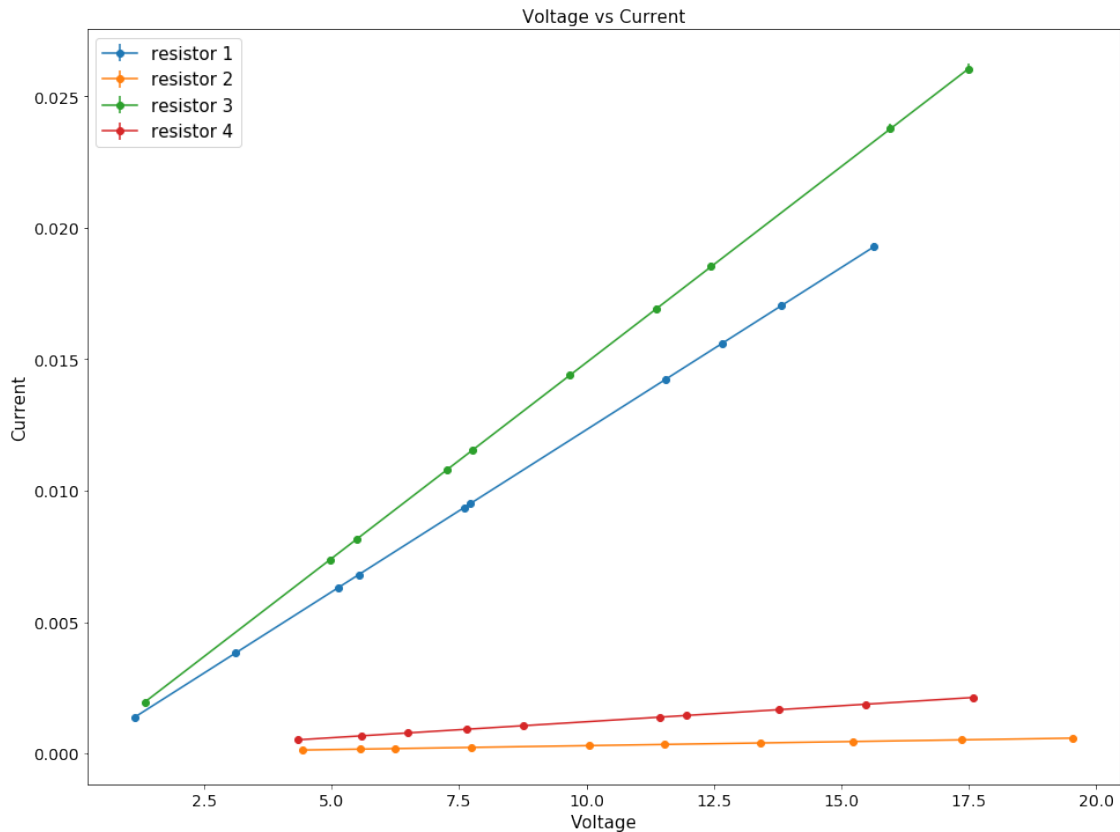
    # fit curve
    popt, pcov = curve_fit(f=model, xdata=voltage, ydata=current,
→sigma=ucurrent, absolute_sigma=True, p0=(1, 1))
    # plot results of fit with error bars
    plt.errorbar(voltage, model(voltage, *popt), yerr=ucurrent, marker='o',
→ls='-', label=f'resistor {i}')

    # second fit, forcing line to pass through 0. not plotting, just getting
→answers for questions.
    popt2, pcov2 = curve_fit(f=model2, xdata=voltage, ydata=current,
→sigma=ucurrent, absolute_sigma=True, p0=(1))

    # store output parameters in array
    params[i-1] = (i, popt[1], np.sqrt(np.diag(pcov)[1]), rcs(model(voltage,
→*popt), current, ucurrent, 2), 1/popt[0], np.sqrt(np.diag(pcov)[0])/np.
→square(popt[0]), 1/popt2[0], np.sqrt(np.diag(pcov2)[0])/np.square(popt2[0]))

# plot stuff
plt.legend(fontsize=15);
plt.title('Voltage vs Current', fontsize=15)
plt.xlabel('Voltage', fontsize=15)
plt.ylabel('Current', fontsize=15)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.savefig('plot.png')

```



```
[5]: # put parameters into dataframe because it looks nicer
params = pd.DataFrame(params)
# add labels to columns
params.columns = ['resistor_id', 'fit intercept', 'intercept error', 'reduced_
↳chi squared', 'resistance', 'resistance error', 'resistance (forced through_
↳0)', 'resistance error (forced through 0)']
params
```

```
[5]:
```

	resistor_id	fit intercept	intercept error	reduced chi squared \
0	1.0	-1.174579e-05	0.000012	0.115313
1	2.0	5.383516e-07	0.000001	1.070705
2	3.0	-2.253287e-05	0.000018	0.268037
3	4.0	-5.938997e-06	0.000005	0.202627

	resistance	resistance error	resistance (forced through 0) \
0	811.053336	2.592638	812.727682
1	33454.156824	178.852048	33386.250926
2	671.273711	2.150820	673.111334
3	8232.651096	46.704861	8280.273887

```
resistance error (forced through 0)
```

```

0          1.927563
1          79.184394
2          1.596440
3          19.638552

```

```

[6]: # calculate resistances and uncertainties
data['resistance'] = data.voltage / data.current
data['uresistance'] = np.sqrt(np.square(data.uvoltage / data.voltage) + np.
    ↳square(data.ucurrent / data.current)) * data.resistance
data

```

```

[6]:      current      ucurrent  voltage  uvoltage  resistorid  resistance \
0    0.001373  1.029750e-05    1.122  0.002805         1    817.188638
1    0.003810  2.857500e-05    3.110  0.007775         1    816.272966
2    0.006300  4.725000e-05    5.120  0.012800         1    812.698413
3    0.006820  5.115000e-05    5.530  0.013825         1    810.850440
4    0.009360  7.020000e-05    7.600  0.019000         1    811.965812
5    0.009490  7.117500e-05    7.720  0.019300         1    813.487882
6    0.014220  1.066500e-04   11.560  0.028900         1    812.939522
7    0.015570  1.167750e-04   12.670  0.031675         1    813.744380
8    0.017070  1.280250e-04   13.830  0.034575         1    810.193322
9    0.019360  1.452000e-04   15.640  0.039100         1    807.851240
10   0.000131  9.825000e-07    4.420  0.011050         2   33740.458015
11   0.000170  1.275000e-06    5.570  0.013925         2   32764.705882
12   0.000187  1.402500e-06    6.230  0.015575         2   33315.508021
13   0.000232  1.740000e-06    7.740  0.019350         2   33362.068966
14   0.000301  2.257500e-06   10.060  0.025150         2   33421.926910
15   0.000345  2.587500e-06   11.540  0.028850         2   33449.275362
16   0.000401  3.007500e-06   13.410  0.033525         2   33441.396509
17   0.000456  3.420000e-06   15.230  0.038075         2   33399.122807
18   0.000518  3.885000e-06   17.370  0.043425         2   33532.818533
19   0.000585  4.387500e-06   19.550  0.048875         2   33418.803419
20   0.001960  1.470000e-05    1.330  0.003325         3    678.571429
21   0.007340  5.505000e-05    4.960  0.012400         3    675.749319
22   0.008220  6.165000e-05    5.500  0.013750         3    669.099757
23   0.010770  8.077500e-05    7.270  0.018175         3    675.023213
24   0.011500  8.625000e-05    7.760  0.019400         3    674.782609
25   0.014350  1.076250e-04    9.670  0.024175         3    673.867596
26   0.016890  1.266750e-04   11.380  0.028450         3    673.771462
27   0.018550  1.391250e-04   12.450  0.031125         3    671.159030
28   0.023800  1.785000e-04   15.970  0.039925         3    671.008403
29   0.026200  1.965000e-04   17.500  0.043750         3    667.938931
30   0.000518  3.885000e-06    4.340  0.010850         4   8378.378378
31   0.000675  5.062500e-06    5.580  0.013950         4   8266.666667
32   0.000785  5.887500e-06    6.490  0.016225         4   8267.515924
33   0.000926  6.945000e-06    7.650  0.019125         4   8261.339093
34   0.001060  7.950000e-06    8.770  0.021925         4   8273.584906

```

35	0.001382	1.036500e-05	11.430	0.028575	4	8270.622287
36	0.001448	1.086000e-05	11.970	0.029925	4	8266.574586
37	0.001666	1.249500e-05	13.780	0.034450	4	8271.308523
38	0.001869	1.401750e-05	15.480	0.038700	4	8282.504013
39	0.002130	1.597500e-05	17.600	0.044000	4	8262.910798

	uresistance
0	6.460443
1	6.453204
2	6.424945
3	6.410336
4	6.419153
5	6.431186
6	6.426851
7	6.433214
8	6.405141
9	6.386625
10	266.741742
11	259.027744
12	263.382217
13	263.750313
14	264.223532
15	264.439741
16	264.377453
17	264.043250
18	265.100207
19	264.198839
20	5.364578
21	5.342267
22	5.289698
23	5.336527
24	5.334625
25	5.327391
26	5.326631
27	5.305978
28	5.304787
29	5.280521
30	66.236897
31	65.353738
32	65.360452
33	65.311620
34	65.408432
35	65.385010
36	65.353010
37	65.390435
38	65.478944
39	65.324046

[]: