Julia Frothingham
PHY 240

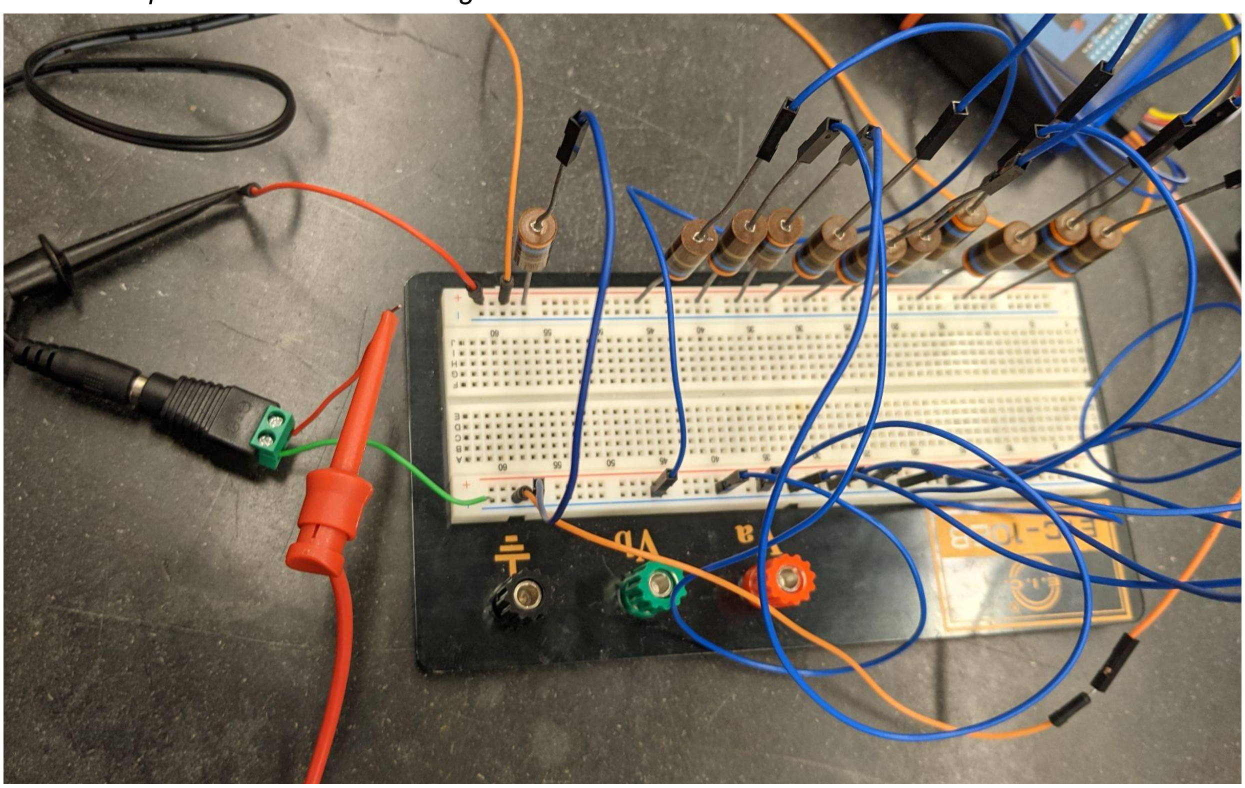
Lab 3 - Measurements and analysis

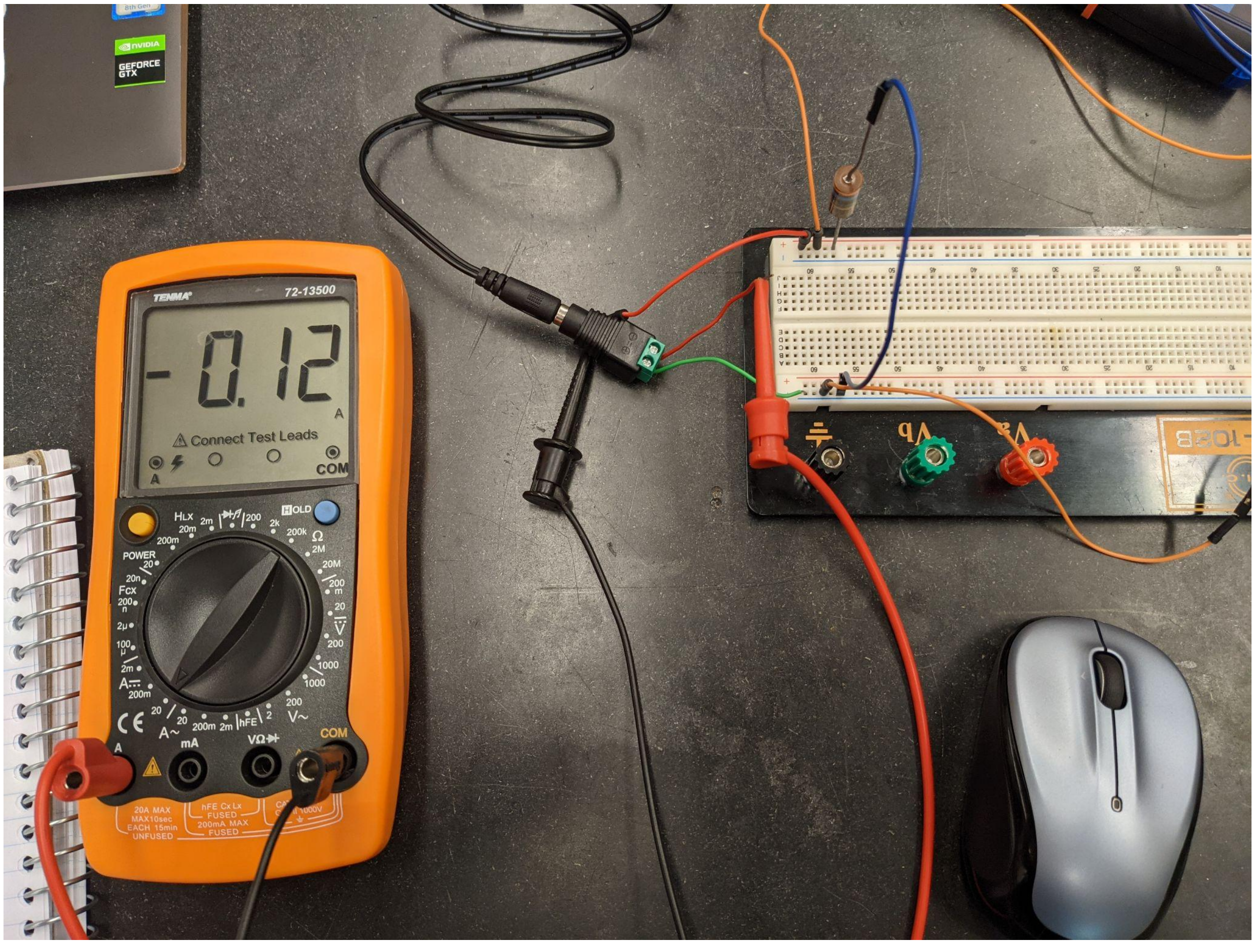
Part 1: Measuring the V vs I curve of a power supply

As outlined in the submitted experimental/circuit design, the voltage across the power supply was measured using the Voltmeter function in Scopy (orange wires in Figure 1), while the current through the power supply was measured using the DMM (red and black probes in Figure 1). Voltage and current measurements were taken after changing the current by a controlled amount. The current was varied by adding a number of resistors of the same resistance value to the circuit in parallel, creating a current divider.

In the submitted experimental design, measurements were to be taken after adding resistors to the circuit - so, starting with no resistors and increasing the number of resistors. In practice, it was more efficient to build the circuit with all the intended resistors and then take measurements after *removing* one resistor at a time. This was because it was difficult to attach the large, 2W-rated resistors to the protoboard and to the jumper wires connecting them to ground. Removing the resistors turned out to be easier and less disruptive to the rest of the constructed circuit.

Figure 1. Circuits used to measure internal resistance of a 5V power supply. The number of resistors in parallel in the circuit ranged from 0 to 10.





The current data was collected with an estimated uncertainty of 1%, as the third digit of the DMM display fluctuated by about +/- 1 for each measurement.

Figure 2. Collected I_s and V_s data.

Number of resistors	Measured I_s (A)	Measured V_s (A)
10	1.14	4.243
9	1.03	4.276
8	0.92	4.293
7	0.82	4.360
6	0.71	4.393
5	0.60	4.460
4	0.49	4.560
3	0.37	4.610

2	0.25	4.694
1	0.12	4.777
0	0	4.994

To determine the internal resistance of the power supply, the current through the power supply and the voltage across the power supply need to be plotted against each other. By Ohm's Law, the internal resistance of the power supply is equal to the voltage across the power supply, V_s, divided by the current through the power supply, I_s. With I_s plotted on the x-axis and V_s plotted on the y-axis, this means that the internal resistance is equal to the slope of a line approximating the relationship between I_s and V_s.

Python was used to calculate the linear regression of the data, and overlay it on such a plot.

Figure 3. Python code for linear regression and plotting.

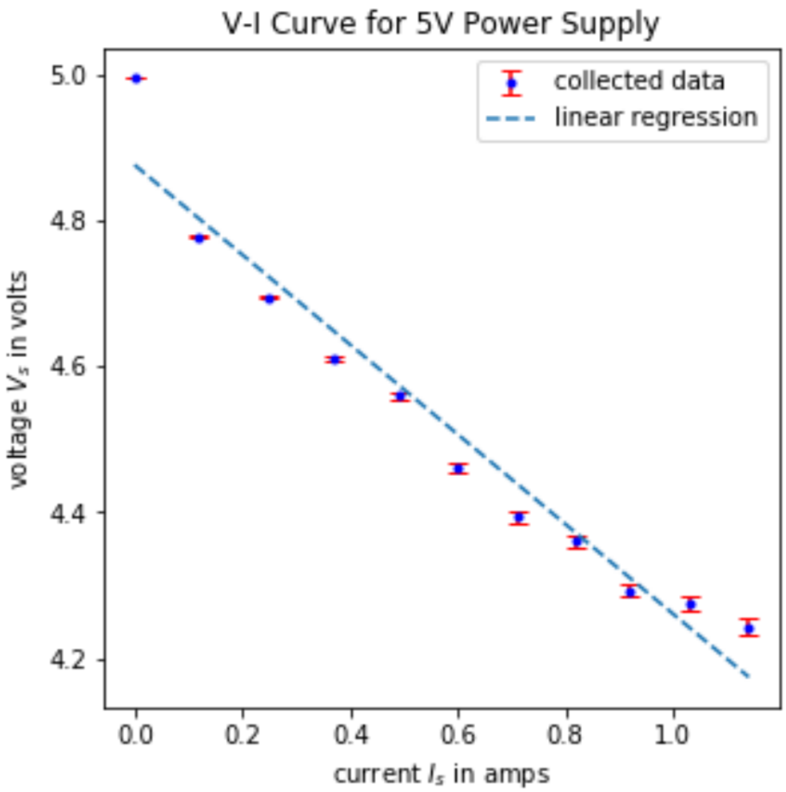
```
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
```

```
# collected data
current = np.array([0, 0.12, 0.25, 0.37, 0.49, 0.6, 0.71, 0.82, 0.92, 1.03, 1.14])
voltage = np.array([4.994, 4.777, 4.694, 4.610, 4.560, 4.460, 4.393,
                    4.360, 4.293, 4.276, 4.243])

# estimated uncertainty of 1%
delta_i = 0.01 * current

# calculate linear regression
slope, intercept, r_value, p_value, std_err = stats.linregress(current, voltage)
line = (slope*current)+intercept
print('voltage =',round(slope,3), '* current +', round(intercept,3))
print('r value:', round(r_value,3))
print('p value:', '%.2e'%p_value)
print('standard error:', round(std_err,3))
```

```
voltage = -0.615 * current + 4.875
r value: -0.975
p value: 3.36e-07
standard error: 0.047
```

```
# plot data with errorbars and linear regression
plt.figure(figsize=(5,5))
plt.errorbar(current, voltage, yerr=delta_i, fmt='b.', ecolor='r',
             capsize=4, label='collected data')
plt.errorbar(current, line, fmt='--', label='linear regression')
plt.title('V-I Curve for 5V Power Supply')
plt.xlabel('current $I_s$ in amps')
plt.ylabel('voltage $V_s$ in volts')
plt.legend()
```

Figure 4. V-I curve for 5V power supply.



From this linear regression, it was determined that the internal resistance of the power supply is approximately 0.62Ω, the absolute value of the (negative) slope of the plotted line.

The uncertainty on the measured values of current and voltage were estimated to be 1%. So the total uncertainty on the internal resistance should be 100*sqrt($0.01^2 + 0.01^2 + 0.01^2$), or approximately 1.7%.

So the internal resistance of the 5V power supply = 0.62Ω ± 1.7%

Part 2: Measuring the DMM's internal resistance

As outlined in the submitted experimental/circuit design, the voltage across the DMM was measured in a circuit with a known resistor, forming a voltage divider. Using the voltage divider equation, the internal resistance R_m of the meter can be calculated from the known resistance R_x in the circuit, the known 5V source voltage, and the measured voltage V_m across the meter. To get a more accurate value of R_m, measurements were taken with several different values of R_x.

In the submitted design, Scopy was used to measure the voltage across the meter. This turned out to be redundant and also unreliable, as issues grounding Scopy made it difficult to get stable voltage readings. Instead, the DMM itself was used to measure voltages.

Additionally, the submitted design used resistors ranging from 200Ω to 1kΩ. They were chosen to avoid loading, and to remain within safe power specifications. However, when attempting to measure V_m, the readings remained steady at about 5V, when there should have been a noticeable decrease in voltage. This was because the resistors used in the circuit were too low-resistance. In comparison to R_m, R_x was almost nothing. So instead, combinations of 2.2MΩ resistors in series were used to create R_x values ranging from 2.2MΩ to 12.2MΩ. These gave more successful results.

Figure 5. Circuit used to measure internal resistance of DMM. The number of resistors in series in the circuit ranged from 1 to 6.

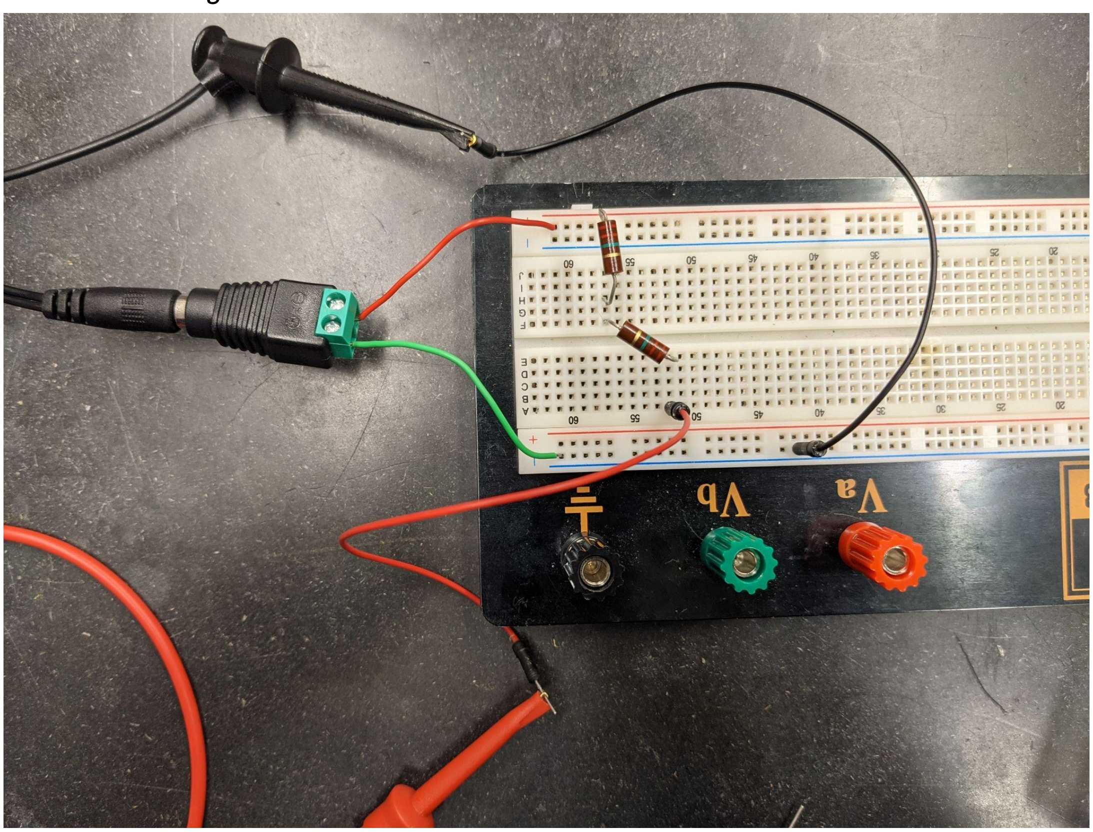


Figure 6. Collected V_m data.

Resistance (MΩ)	Measured V_m (V)
2.2	4.07
4.4	3.43
6.6	2.96
8.8	2.62
10.0	2.34
12.2	2.13

After collecting the data, Python was used to calculate the internal resistance six times, once for each resistance and voltage pair, then determine an average internal resistance with uncertainty.

Figure 7. Python code for R_m calculation

```
# function to calculate Rm from input Rx and Vm
def calc_rm(rx,vm):
    rm = (vm * rx) / (5 - vm)
    return rm
```

```
# arrays to store collected data and calculated values
Rx_values = np.array([2.2,4.4,6.6,8.8,10.0,12.2])
Vm_values = np.array([4.07,3.43,2.96,2.62,2.34,2.13])
Rm_values = np.zeros(6)

# calculate Rm from data
i = 0
while i<6:
    current_rx = Rx_values[i]
    current_vm = Vm_values[i]
    Rm_values[i] = round(calc_rm(current_rx,current_vm),2)
    i+=1

avg_Rm = round(np.average(Rm_values),2)
```

```
# uncertainty calculations
Vm_uncertainty = 0.01  # estimated uncertainty of 1% on Vm
Rx_uncertainty = 0.05  # resistor tolerance of 5%
Rm_uncertainty = round(np.sqrt((Rx_uncertainty**2)
                              +(Vm_uncertainty**2)
                              +(Vm_uncertainty**2)),3)
```

```
# the results
print('all Rm values:',Rm_values)
print('average Rm value:',avg_Rm, '+/-', 100*Rm_uncertainty,'%')
```

```
all Rm values: [9.63 9.61 9.58 9.69 8.8  9.05]
average Rm value: 9.39 +/- 5.2 %
```

So the internal resistance of the DMM = 9.39MΩ ± 5.2%