Daniel Opdahl

Photoelectric effect experiment

Introduction

In the early 1900’s, inspired by the idea that all energy is quantized, Albert Einstein postulated that light energy must also be quantized, meaning that light delivers its energy in “packets”. Physicists of the early 1900’s observed that when metallic surfaces were illuminated by light of a sufficient frequency, electrons are ejected from the metal. Linking these observations to his postulates, Einstein proposed that every material has a certain threshold of energy that it takes to eject an electron out of the material and eject it. Additionally, Einstein proposed that when electrons are ejected from a material, their energy after being ejected are proportional to the energy of the incident light. Since physicists of the time observed that a brighter light did not result in more ejected electrons, but different colors of light resulted in different amounts of ejected electrons, Einstein’s postulates implied that the energy of of the ejected electron is related not to the intensity of the light (how bright the light is), but rather the frequency of the light (the color).

When light hits a metal, if the light is of a certain frequency or above, electrons can be ejected from the metal. This phenomena occurs because, as Einstein hypothesized, the energy in light is quantized and is carried by quanta of energy called photons. As Einstein discovered, a photon’s energy is dependent on its frequency, having energy,

(1)

where is the energy of the photon, is Planck’s constant, and is the frequency of light. When an incoming photon is absorbed by an electron in a metal, the photon completely transfers all of its energy to the electron. If the energy of the photon is great enough (*i.e.*, if the frequency of the incoming photon is sufficiently large), the electron can gain enough energy to break the atomic forces that keep it bound in the material, and it can escape from the metal. The minimum energy necessary for an electron to escape from the surface of a given material is called the work function of the material. Subtracting the work function of the material from the energy of the photon, we can find the maximum kinetic energy of an ejected electron for an incident photon of given frequency, for a material of a given work function:

(2)

where is the remaining kinetic energy of the electron after it has escaped, and is the energy of the incoming photon.

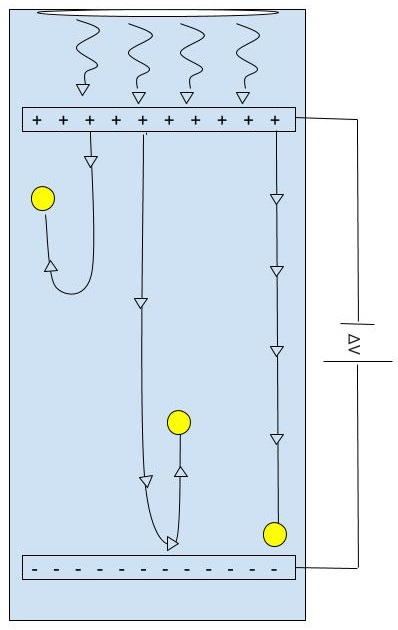
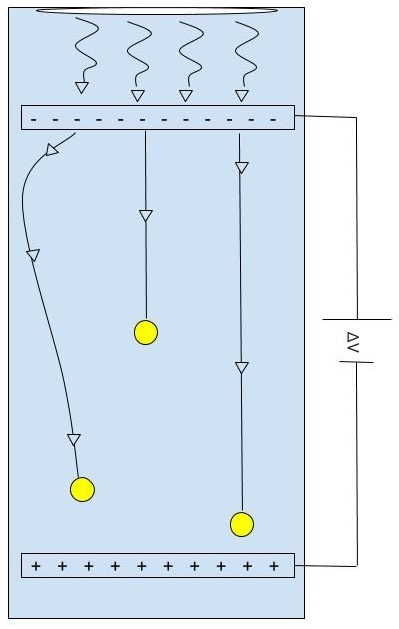


Figure 1: The phototube consists of two metal plates with an applied voltage across them, where the plate that has light incident on it is called the cathode and the plate that does not have light incident on it is called the anode. As incoming light of a certain frequency hits the cathode, photoelectrons are ejected from the cathode in all directions. Depending on potential difference between the plates and the kinetic energy of the photoelectrons, some photoelectrons have sufficient energy to cross the gap and form a current between the two plates. The top diagram represents a “forward bias” configuration, and the bottom diagram represents a “reverse bias” configuration.

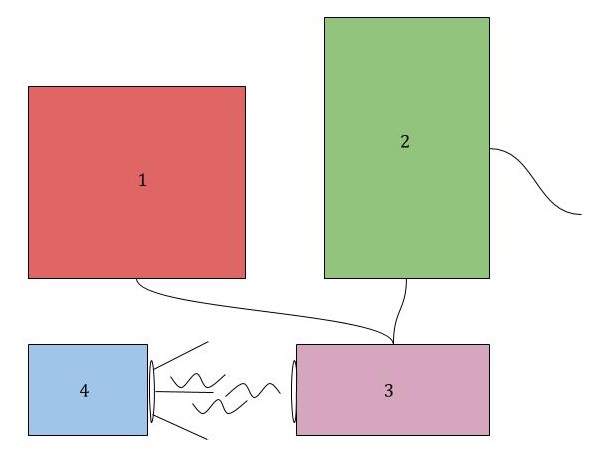
If two metal plates are set up in a phototube such that one plate has light shone on it, and the other plate is positioned in a direct line to the other metal plate, ejected electrons from the first plate can reach the second plate, as shown in Figure 1. If the two plates are connected electrically, the ejection of electrons from one plate to the other can cause a small current. The plate that has the incident light of a sufficient frequency shone on and ejects electrons is called the cathode (depicted on the left in Figure 1) and the plate that receives the ejected electrons, the “collector” plate, is called the anode (depicted on the right in Figure 1). If a voltage is applied between the two plates, it is possible to limit the amount of electrons that get to the anode. If a positive voltage is applied between the plates, such as in the top image of Figure 1, ejected electrons will be attracted to the anode and more electrons will cross the gap and thus the current will increase. When ejected electrons are attracted to the anode, it is referred to as a forward bias setup. If the voltage is increased, eventually we will reach a limit of a maximum number of electrons crossing the gap and making up the current. This maximum current is referred to as the “saturation current”. If there is no voltage applied between the plates, only the photoelectrons that have the a direct exiting direction towards the other plate and sufficient kinetic energy will be able to cross the gap and contribute to the current. If a negative voltage is applied, the ejected electrons will be repelled by the anode and fewer electrons will cross the gap, and thus the current will decrease. When ejected electrons are repelled by the anode, it is referred to as a reverse bias setup. In this case, only the electrons with a high enough kinetic energy will be able to cross the gap and overcome the repelling force of the negatively-charged anode. The greater the applied voltage in this case, the fewer photoelectrons will be attracted to the anode and the less the resulting current will be until no ejected photoelectrons have sufficient energy to cross the gap and the resulting current is 0 A. The voltage at which the resulting current is 0 A is referred to as the “stopping voltage” or the “retarding voltage”. Due to physical limitations, however, our apparatus always registers a certain negative current due to stray ejected photoelectrons coming off of the anode and colliding with the cathode, resulting in current flowing from anode to cathode. Fortunately, these reverse photoelectrons make up a constant current, so the voltage at which there is no more forward current and the reverse current is the only current measured is our stopping potential.

The work done on an electron by the electric potential as the electron crosses the gap between the two plates is equal to the charge of the electron times the applied voltage between the plates . If the voltage between the plates is set up in a reverse bias configuration and increased until the measured current is zero, the applied voltage is stopping all electrons from crossing the gap. This minimum voltage that stops all current between the two plates is known as the stopping voltage, . At this voltage, the kinetic energy of the most energetic electrons is equal to the work done on them:

(4)

Combining Equation 4 with Equation 2, we can define a relationship between the stopping potential and the frequency of light incident on the cathode.

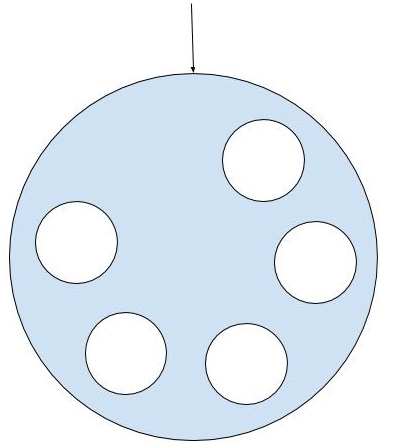
Setup and Methods



Revise diagram – write description in logical way: light shines on phototube.. etc

Figure 2: A depiction of the experimental setup. (1) Power supply controlling the voltage between the plates housed in (3). (2) A picoammeter reading the current created by ejected electrons in (3), connected to a laptop (not shown) that controls the power supply and reads and stores the data from the picoammeter. (3) The phototube that houses the two plates and has attached to it several filters for filtering incoming light down to one wavelength. (4) A mercury lamp that provides the incident light on the cathode housed in (3).

The first thing we did when preparing to collect data was power up the mercury lamp. The lamp required a bit of time to warm up and reach a sustained brightness, and we could not start taking data until we reached that sustained level of brightness. The next thing we did was ensure that the room in which we were collecting data was as dark as we could make it by drawing curtains around the area and shutting off all lights. Ensuring that the room was dark minimized the chances that stray light from outside sources could interfere with our experiment via stray photons entering the phototube and contributing to extraneous current. Eliminating external sources of light was necessary because we wanted to have light incident from the mercury lamp be the only source of light in our testing so that we could control for the intensity of the light at any given wavelength.



Label filters in this figure

Figure 3: A close up of the lens disk that housed 5 different filters used to filter light from the mercury lamp, only allowing a certain band of wavelength through.

It was important to allow only light of a certain wavelength and frequency in to the phototube so that we could control for the energies of incident photons. On the front of the phototube was a disk with several filters that would allow only light of a certain wavelength and frequency through, which are shown in Figure 3. The frequencies that the filters allowed through correspond to the frequencies that the peak intensities of the light that the mercury lamp gave off. We rotated the filter disk to place a desired filter over the opening to the phototube, being careful not to touch the filter lest we dirty it.

Next, we configured our power supply to control the potential between the two plates inside the phototube while a picoammeter was connected to the phototube to measure the current at each incremented voltage. A diagram of a cross section of the phototube is shown in Figure 1, which has already been discussed.

Finally, a computer running a MATLAB script, found in Appendix A, automated the data collection process by automatically incrementing the voltage applied by the power supply and taking measurements from the picoammeter of the resulting current inside the phototube. Starting by setting up the leads from the power supply to the phototube in a forward bias fashion (such that the cathode is negatively charged and the anode is positively charged), for each filter, we configured the MATLAB script to use the power supply to increase the applied potential across the plates from 0 V to 30 V, incrementing the voltage by 0.1 V, and waiting 1 second between increments. At each increment, the picoammeter would measure a value for the current at that voltage, and send that measured value to the computer to be recorded by the MATLAB script. After we had taken measurements from 0 V to 30 V in this way three times for each filter, we switched the leads on the power supply in order to reverse the voltage and gather data for the reverse bias configuration. Again, we used the MATLAB script found in Appendix A to configure the power supply and the picoammeter to take data automatically. However, this time the setup was configured to take data from 0 V to -2 V, with increments of 0.05V, and still a 1 second pause in between increments. We repeated taking data in this reverse bias position three times for each filter.

Results and Interpretation

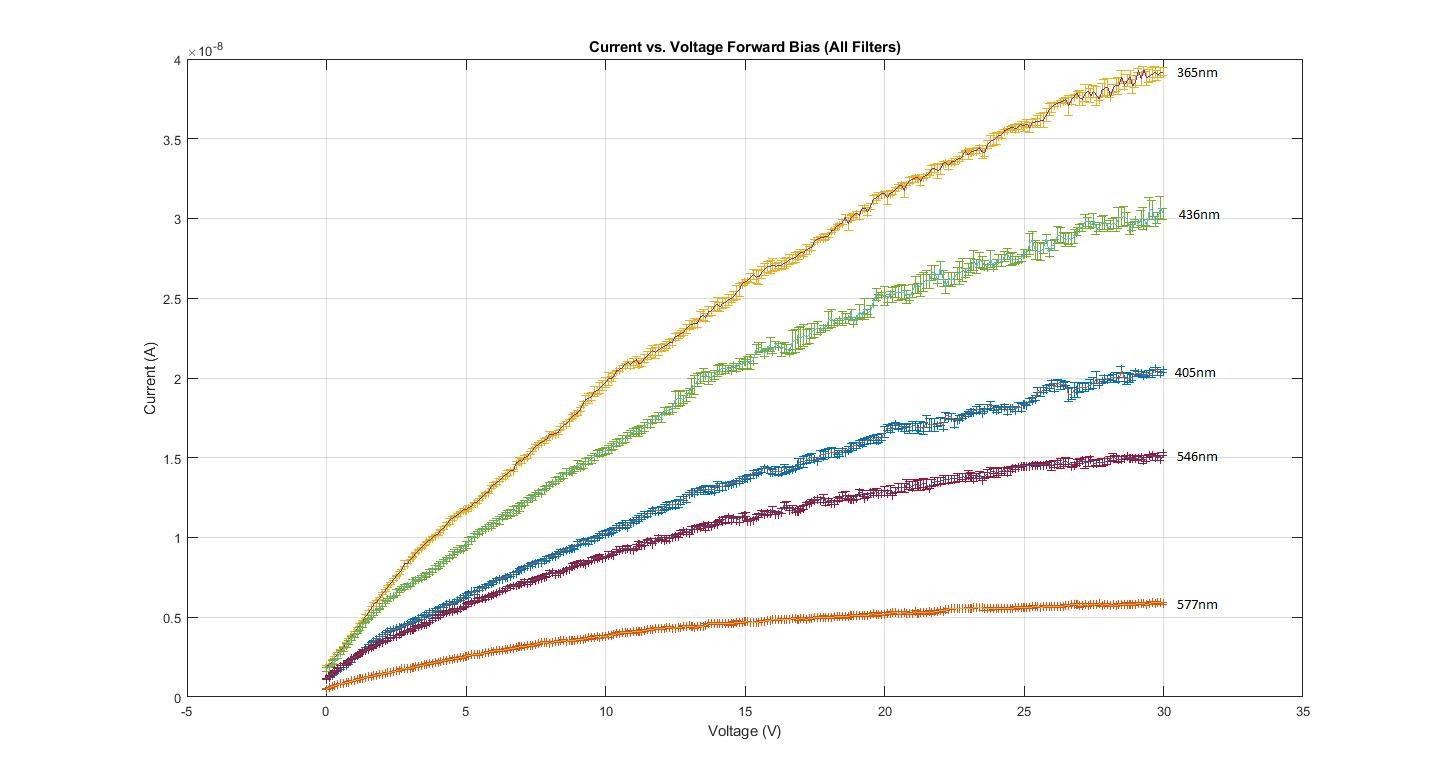


Figure 4: Data we gathered for each filter in the forward bias configuration from 0 V to 30 V. The trend of smaller wavelength light resulting in a larger current appears very clearly in our data. Shorter wavelength light means more energetic photons imparting their energy to photoelectrons, resulting in more energetic and faster photoelectrons. More energetic photoelectrons means that more photoelectrons will have sufficient energy to cross the gap, resulting in a higher current.

The trend of smaller wavelength light resulting in a larger current appears very clearly in our data. This is consistent with what we would expect, as shorter wavelengths mean higher frequency light which means more energetic photons imparting their energy to photoelectrons. Since the photoelectrons have more energy if the wavelength of incident light is shorter, more photoelectrons will have sufficient energy to cross the gap, resulting in a higher current. Each curve has uncertainty bars in both the voltage and the current, even though the voltage uncertainty bars are not visible. The uncertainty in both the voltage and the current come from the limitations of our data gathering apparatuses.

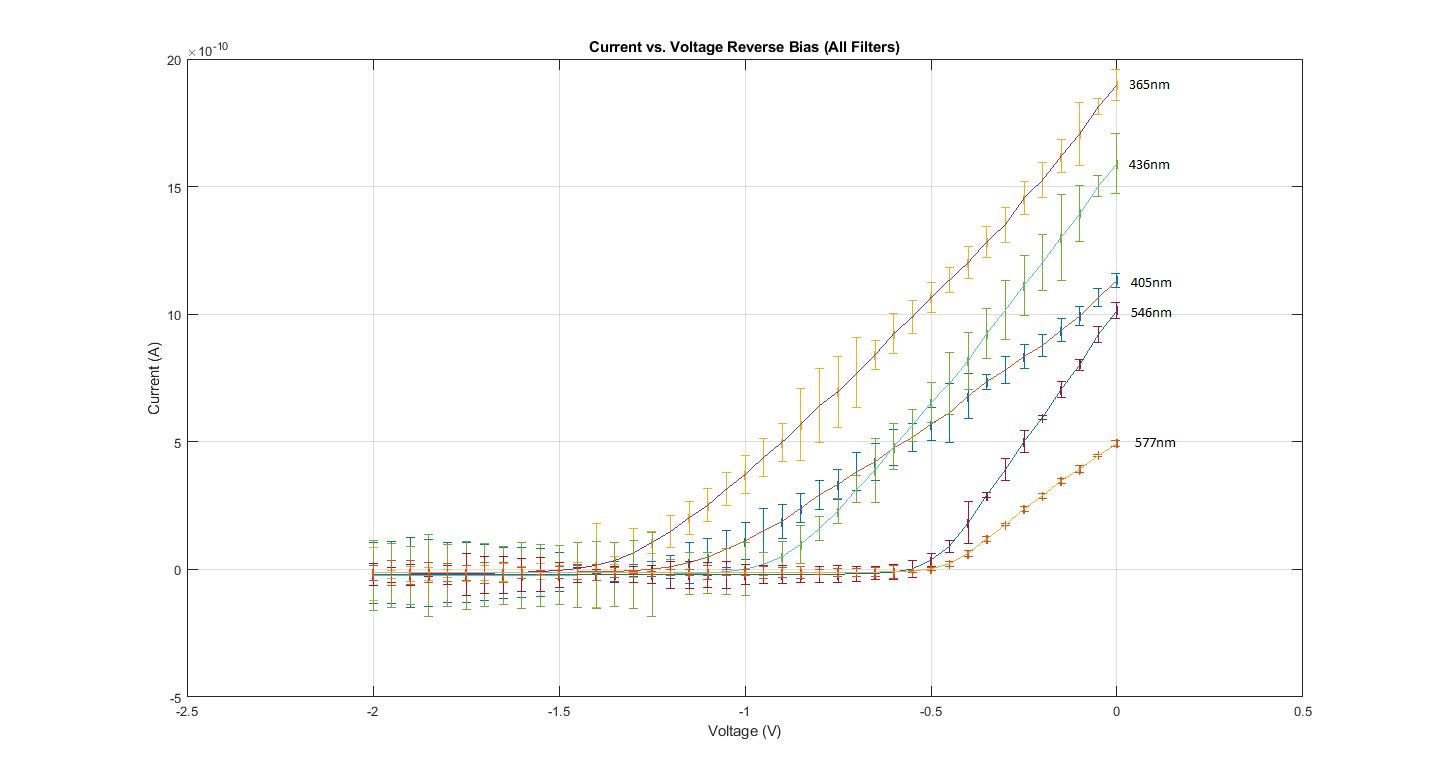


Figure 5: Data we gathered for each filter in the reverse bias configuration from 0V to -2V. We see that the knees in the curves follow a pattern of shorter wavelength corresponding to a greater negative voltage. This makes sense because more energetic photoelectrons require a greater applied potential to halt, and accordingly, the most energetic light generates the photoelectrons that require the most voltage to stop.

At first glance, one might conclude that our trend of shorter wavelengths resulting in a larger current does not hold with this data set given how the 405nm filter has a lower current than the 436nm filter at 0V. However, looking at the scale of the values for the measured currents in Figure 5 and Figure 4, we see that this discrepancy is temporary, because all filters generate nearly the same amount of current at 0 V relative to their generated currents at 30 V in Figure 4.

The data that Figure 5 displays is most useful in determining the stopping voltage. This can be accomplished through determining the voltage at which the rather constant decreasing curve in voltage stops for each filter and “levels out”, representing only the reverse current discussed previously. This point is referred to as the “knee” in the curve, the voltage at which the curve changes from decreasing to flat. When seeking to answer where the stopping potential is for each filter (*i.e.*, where the knees in the curves are) based on the data in Figure 5, we see that our data is consistent with what we would expect. More energetic photoelectrons require a greater applied potential to halt, and the most energetic light (the light with the shortest wavelength and consequentially the highest frequency) generates the photoelectrons that require the most voltage to stop. We see that the knees in the curves follow this pattern of shorter wavelength corresponding to a greater negative voltage. Each curve has uncertainty bars in both the voltage and the current, even though the voltage uncertainty bars are not visible. The uncertainty in both the voltage and the current come from the limitations of our data gathering apparatuses.

In order to determine the saturation current for each wavelength of light allowed in by each of the filters, we noticed that the current vs potential difference curves in Figure 4 did not obviously plateau at the maximum applied voltage. In order to quantitatively determine where the curves would plateau, and therefore determine at what value the saturation current was, we noticed that the curves looked to be exponential in shape. We used MATLAB’s exponential fit tool to find exponential functions that fitted each curve present in Figure 4.

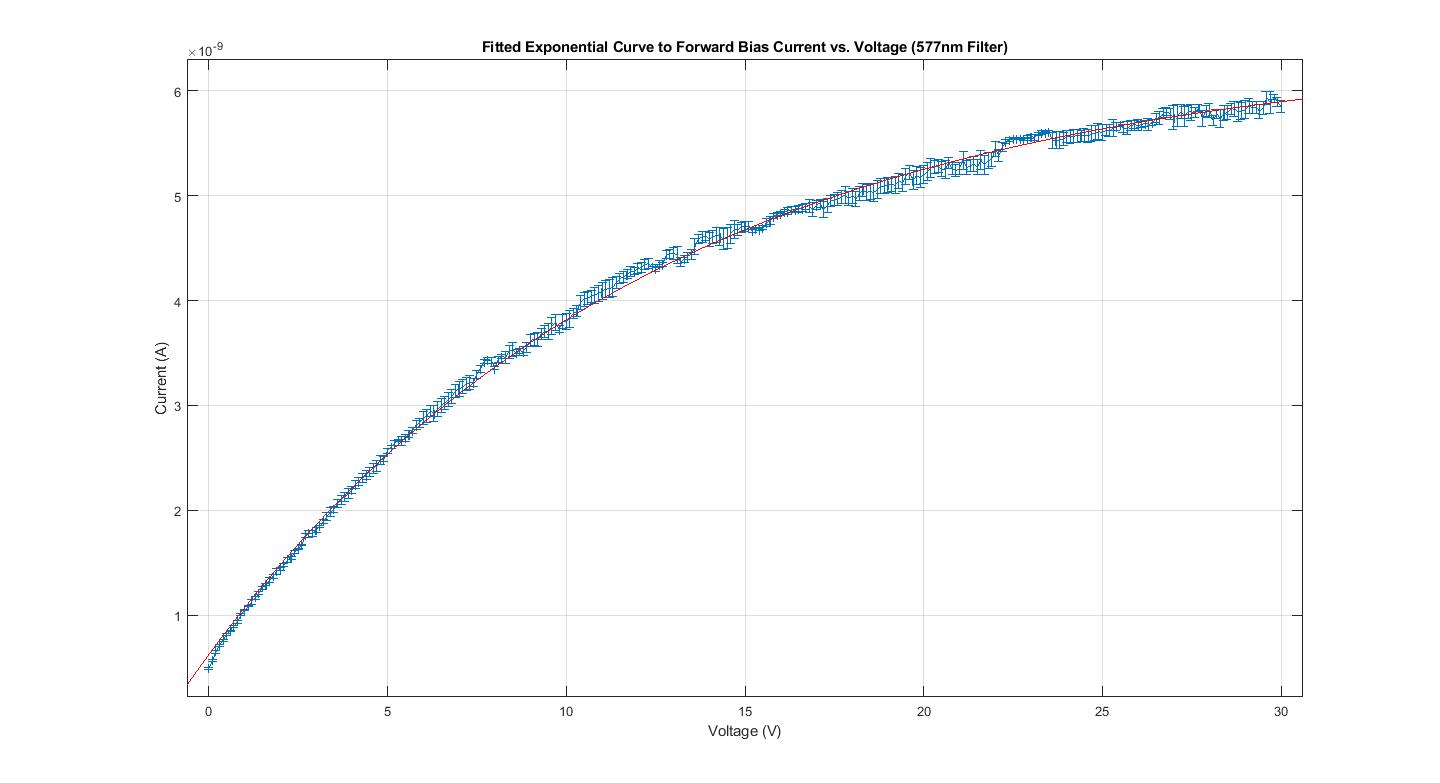


Figure 6: An example of a curve we fitted to the data gathered for the 577nm wavelength filter in the forward bias configuration. By fitting an exponential curve to each data set for each filter, we were able to use the equations of the exponential fits to find where the curve would plateau, and we were able to say that the value of the current at that plateau is the saturation current.

An example of a curve we fitted to the data is shown in Figure 6. By fitting an exponential curve to each data set for each filter, we were able to use the equations of the exponential fits to find where the curve would plateau, and we were able to say that the value of the current at that plateau is the saturation current.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Wavelength of light (nm) | 365 | 405 | 436 | 546 | 577 |
| Saturation Current (nA) | 55± 0.34 | 29±0.22 | 40±0.28 | 18.0±0.062 | 6.4 ±0.015 |

Table 1: The saturation currents for each of the wavelengths of light tested. A pattern emerges of shorter wavelengths of light yielding higher currents. This leads us to the conclusion that the energy of a photon does not depend on the brightness of the light, but rather its frequency and wavelength.

Table 1 shows the saturation currents for each of the wavelengths of light tested. As we saw in Figure 4, the same pattern emerges of shorter wavelengths of light yielding higher currents. This leads us to the conclusion that the energy of a photon does not depend on the brightness of the light, but rather its frequency and wavelength. We believe we can say this with confidence because we controlled for the intensity of emitted light from our mercury lamp by shutting out external sources of light and by allowing the mercury lamp to be fully warmed up before we started gathering data. Essentially, because we controlled for the intensity of light emitted from our light source, we can say that either the lamp emits light of shorter wavelengths at higher intensities and that is what causes the trend in our data, or the energy of a photon depends on its wavelength and not on the intensity of the light.

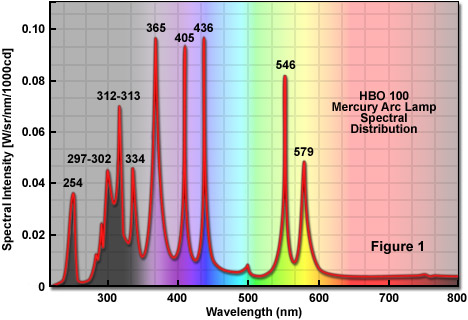


Figure 7: A graph displaying the spectral intensity of a mercury lamp. Note that although mercury lamps emit light of shorter wavelengths at greater intensities in general, we found that 405nm light has more energy than that of 436nm light.

When looking at Figure 7, which shows the spectral intensity of a mercury lamp, we do actually see that in general, mercury lamps emit light of shorter wavelengths at greater intensities. However, one can see that although the relative intensity of 405 nm light is less than that of 43 6nm light, we found that 405 nm light has more energy than that of 436 nm light. This supports the theory that it is not intensity that determines how much energy a photon carries, but rather it is frequency and wavelength that determine how much energy a photon carries.

In order to determine the stopping potential for each wavelength of light tested, we needed to determine where the “knee” was for each curve present in Figure 5. Since the knee of a curve isn’t well defined, we decided to use two methods to approximate the value of the knee, and therefore determine the value of the stopping potential for each current vs voltage curve.

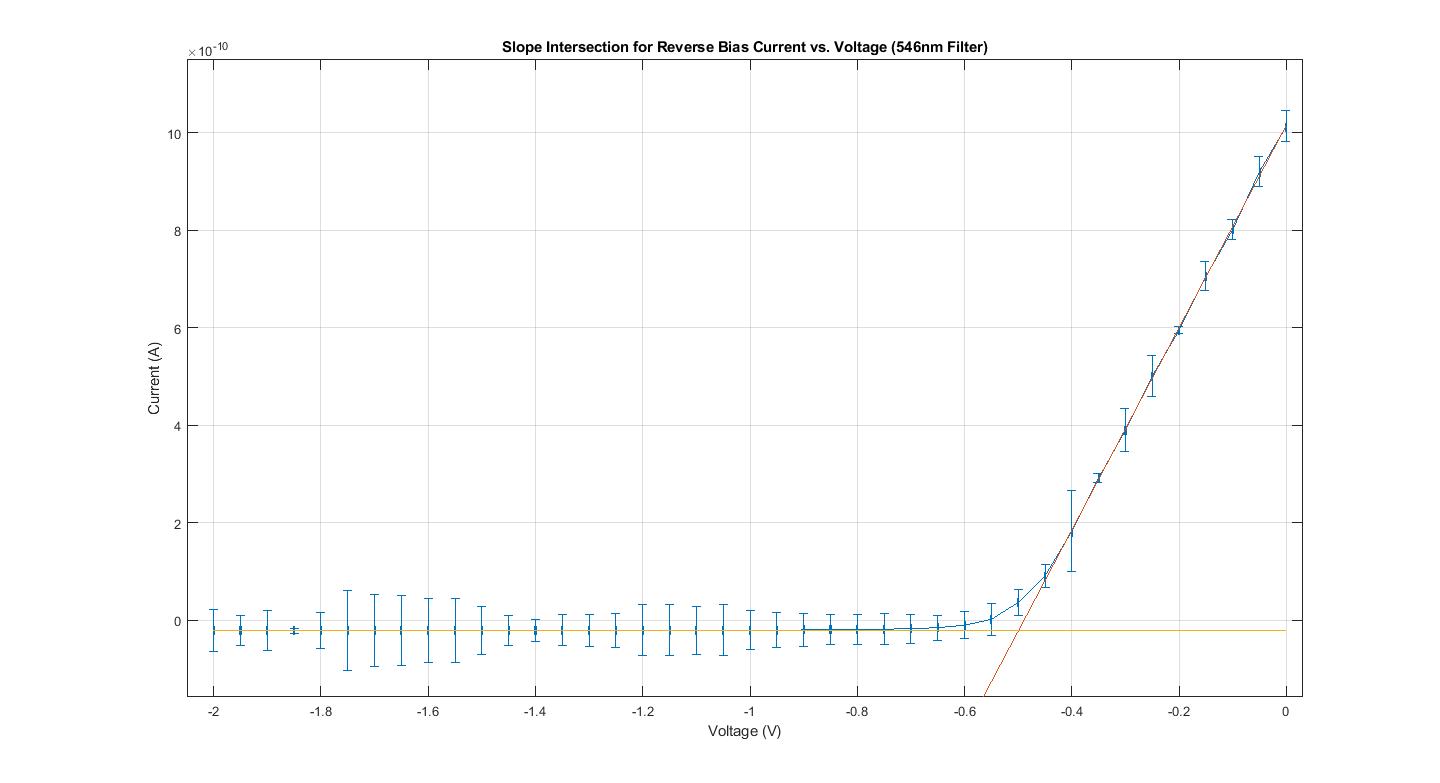


Figure 8: An example of our intersection method for the 546nm wavelength filter testing in the reverse bias configuration. Where the slopes of each of the two most linear parts of the graph were found and then the intersection of those two slopes was found. The voltage at which that intersection occurred was then said to be the stopping voltage.

The first method we used to determine the stopping potential was to fit the two linear parts of the graph before and after the knee and find the intersection of those two lines. An example of this method is shown in Figure 8. The voltage at which the intersection occurred is approximately -0.50 V. Using this method, we said that -0.50 V was the stopping voltage for this filter, for Method 1.

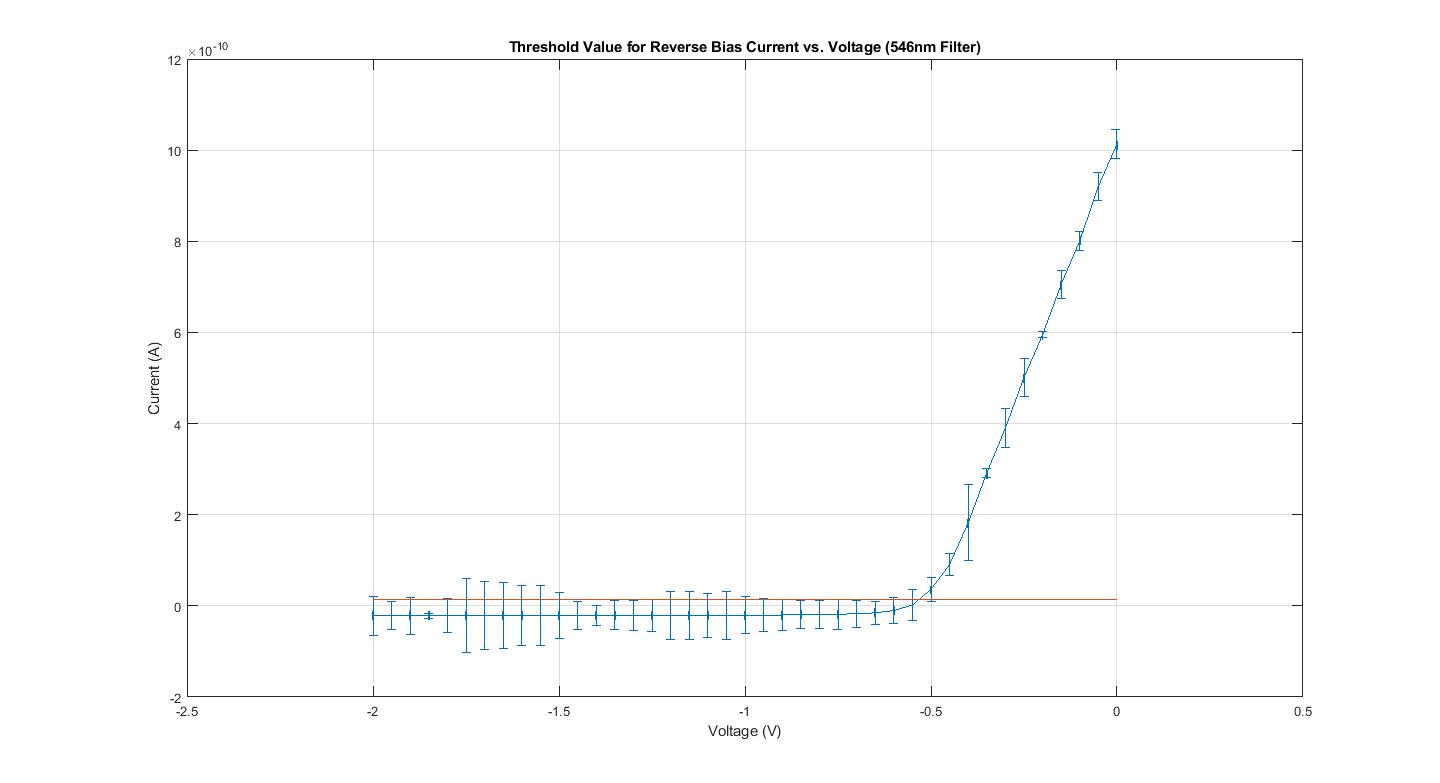


Figure 9: An example of our threshold method for the 546 nm wavelength filter, tested in the reverse bias configuration. We to used data points from the bottom plateau of each curve to determine a threshold value for zero forward current, and the data point whose value for current was above that threshold value and whose voltage was the most negative was said to be located at the stopping voltage.

The second method we used was to use data points of the bottom plateau of each curve to determine a threshold value for zero forward current. Thus, if a data point’s value was above that threshold value, we could say that it was where the current ceased, and the voltage at that data point would be our stopping potential. Figure 9 shows an example of this method. Using our method, the data point at -0.5 V would be the data point at which we would say that there is no more forward current, and thus -0.5 V is our stopping potential for this filter, for Method 2.

Since we determined the values of the stopping potentials by inspection for both methods, we elected to use a constant, estimated uncertainty for both the voltage and the current based on the uncertainties of the original data.

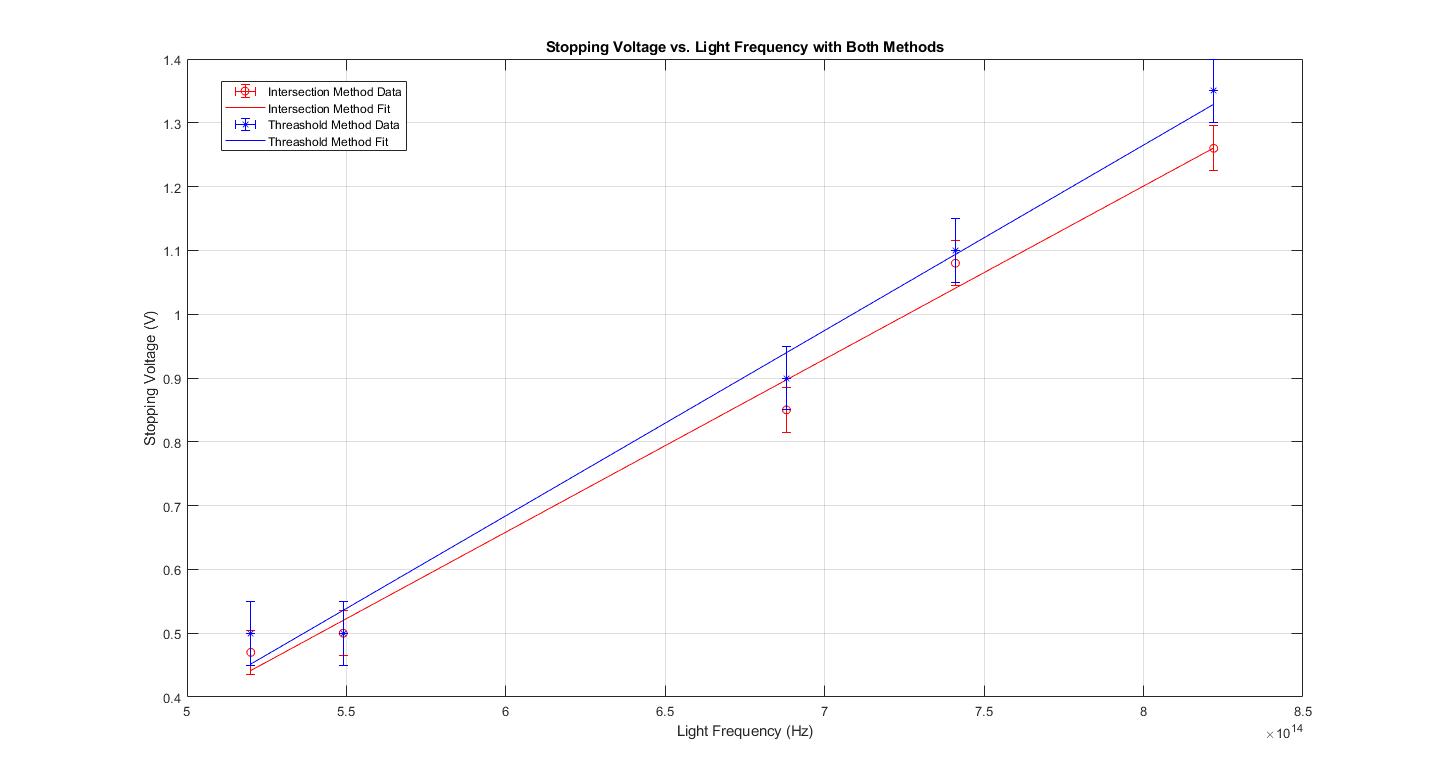


Figure 10: A plot of our data for absolute value of stopping potential versus frequency for each frequency of light tested along with a linear curve for both methods used to determine the stopping potential.

Figure 10 shows stopping potential versus frequency of light tested, and linear curve fits for both methods. We can see that our inference from Figure 5 is right. Since frequency and wavelength are inversely related, as frequency decreases (or as wavelength increases), the absolute value of the stopping potential increases.

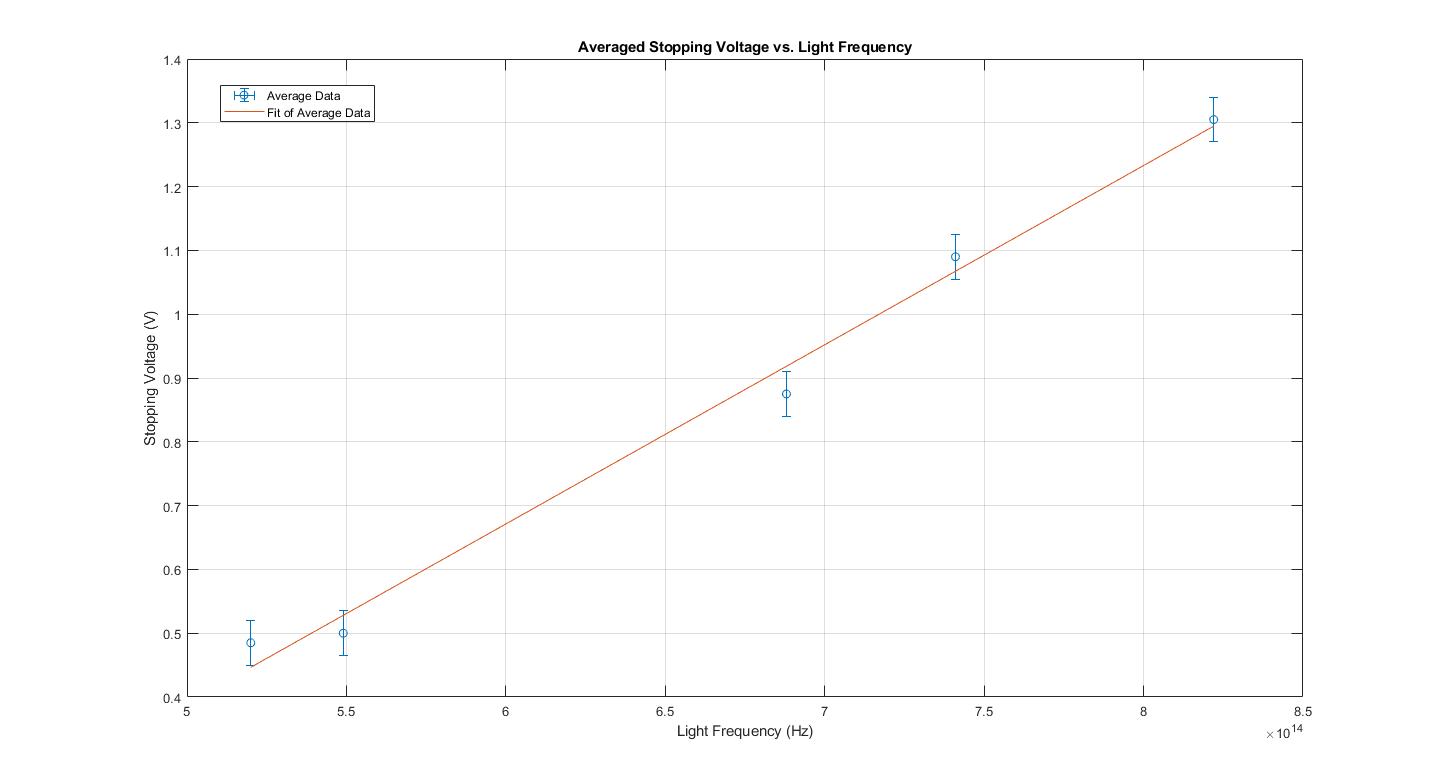


Figure 11: A plot of our averaged data for absolute value of stopping potential versus frequency for each frequency of light tested along with a linear curve representing the average of both methods used to determine the stopping potential.

Figure 11 averages the data from Figure 10 and propagates the uncertainties in the stopping voltage. Using Equation 5, we can produce an equation that models this averaged linear fit:

where is the work function of the material of the cathode and is Planck’s constant. Using this data, the value of can be computed from the slope of the linear fit. Additionally, the work function of the material used for the cathode can be determined by finding the linear fit’s y-intercept. We determined the slope of our averaged linear fit to be Vs. The expected value of is 3.4±0.3×10−15 Vs.

We were able to determine the value of the work function of the material used for the pates inside the phototube. Our experimentally determined value of the work function is 0.87 eV. Unfortunately, there is no metal with this value as its work function. The best estimate we can give for the metal is Rubidium, because it has the smallest work function of 2.26 eV, but even that value is over two times greater than our experimentally determined value. More analysis is needed.

Summary

The photoelectric experiment was performed by shining light of a particular wavelength on a plate, which caused electrons to be ejected from that plate. Inside the phototube, another plate was positioned on the line formed by the light source and the first plate. An electric potential was created between the two plates and was varied by incremental amounts for both forward and reverse bias. Electrons crossing the gap between the two plates for any given applied voltage would create a current and that current was measured and recorded. Using out gathered data for current for a particular voltage for each filter, we were able to determine the saturation current for each wavelength of light tested by fitting an exponential cure to our forward bias data and finding the horizontal asymptote of that curve. Additionally, we were able to determine the stopping potential for each wavelength tested in two ways. First, by finding the threshold value of the horizontal plateau of the data for reverse bias and finding the data point that was closest to that value. Second, by fitting lines to the relatively constant slopes present in the reverse bias graphs and finding the intersection of those two fitted lines. Averaging those values resulted in a measured value for the stopping potential of every wavelength tested, and by plotting the stopping potentials and frequencies (found from the wavelengths of light), we were able to estimate values for and the work function of the material used in the plates inside the phototube. Unfortunately, experimentally determined values of and the work function are inconsistent with known values and expected values.

Appendix A: MATLAB script used for data collection

%This script automates the data collection capabilities for the

%PASCO Photoelectric effect apparatus.

%Both the Keithley Picoammeter and the AMREL powersupply should be

%connected in series to the computer using the Agilent GPIB-to-USB cable, and the

%GPIB-to-GPIB cable.

%% Code exported from tmtool for Keithley Ammeter

% Find a VISA-GPIB object.

obj1 = instrfind('Type', 'visa-gpib', 'RsrcName', 'GPIB0::14::INSTR', 'Tag', '');

% Create the VISA-GPIB object if it does not exist

% otherwise use the object that was found.

if isempty(obj1)

obj1 = visa('AGILENT', 'GPIB0::14::INSTR');

else

fclose(obj1);

obj1 = obj1(1);

end

% Connect to instrument object, obj1.

fopen(obj1);

%% Code exported from tmtool for AMREL powersupply

% Find a VISA-GPIB object.

obj2 = instrfind('Type', 'visa-gpib', 'RsrcName', 'GPIB0::12::0::INSTR', 'Tag', '');

% Create the VISA-GPIB object if it does not exist

% otherwise use the object that was found.

if isempty(obj2)

obj2 = visa('AGILENT', 'GPIB0::12::0::INSTR');

else

fclose(obj2);

obj2 = obj2(1);

end

% Connect to instrument object, obj2.

fopen(obj2);

%% Automating Photoelectric effect data acquistion

%Prompts for the user

vstart = input('Input your desired initial value for the powersupply voltage (in V): ');

vend = input('Input your desired final value for the powersupply voltage(in V): ');

vstep = input('Input the step size for the powersupply voltage (in V): ');

n = (vend - vstart)/vstep + 1;

voltage = (vstart:vstep:vend)';

%initialize output data vector

current = zeros(n,1);

for i = 1:n

% Communicating with instrument object, obj2: Setting voltage

fprintf(obj2, horzcat('vset ', num2str(voltage(i))));

pause(1.0); %pause for 1 sec for current to change

% Communicating with instrument object, obj1: Reading current

tempcurrent = fscanf(obj1);

cutcurrent = tempcurrent(5:end);

current(i) = str2double(cutcurrent);

end

data = [voltage, current];

fprintf(obj2, 'vset 0');

volt\_list = data(:,1);

current\_list = data(:,2);

grid on

hold off

plot(volt\_list, current\_list, 'o');

Appendix B: MATLAB script used for data analysis

% Import Data

TITLE= 'Select the file with the data you want to bring into MATLAB';

[filename,filepath] = uigetfile('\*.\*', TITLE); %Prompts the user to select a data file

full\_filename = fullfile( filepath, filename );

[~,SheetNames] = xlsfinfo(full\_filename);

nSheets = length(SheetNames);

Data = [];

for ii=1:nSheets

Name = SheetNames{ii};

Data = [Data, xlsread(full\_filename, Name)];

end

forward\_voltage\_V577 = Data(:,1);

forward\_current\_A577 = Data(:,2);

reverse\_voltage\_V577 = Data(:,4);

reverse\_current\_A577 = Data(:,5);

forward\_voltage\_V546 = Data(:,6);

forward\_current\_A546 = Data(:,7);

reverse\_voltage\_V546 = Data(:,9);

reverse\_current\_A546 = Data(:,10);

forward\_voltage\_V436 = Data(:,11);

forward\_current\_A436 = Data(:,12);

reverse\_voltage\_V436 = Data(:,14);

reverse\_current\_A436 = Data(:,15);

forward\_voltage\_V405 = Data(:,16);

forward\_current\_A405 = Data(:,17);

reverse\_voltage\_V405 = Data(:,19);

reverse\_current\_A405 = Data(:,20);

forward\_voltage\_V365 = Data(:,21);

forward\_current\_A365 = Data(:,22);

reverse\_voltage\_V365 = Data(:,24);

reverse\_current\_A365 = Data(:,25);

%%%%% 577 analysis

%%% FORWARD

% Calculate average of three runs

firstrun577 = forward\_current\_A577(1:301);

secondrun577 = forward\_current\_A577(302:602);

thirdrun577 = forward\_current\_A577(603:903);

averagecurrent577 = [];

current\_standard\_dev\_577 = [];

for i = 1:301;

averagecurrent577i = (firstrun577(i) + secondrun577(i) + thirdrun577(i))/3;

current\_standard\_dev\_577(i) = std([firstrun577(i); secondrun577(i); thirdrun577(i)]);

averagecurrent577(i) = averagecurrent577i;

end

voltagetouse577 = forward\_voltage\_V577(1:301);

% Uncertainty in Voltage

voltage\_unc\_577 = 0.001\*ones(size(voltagetouse577));

% Uncertainty in Current

current\_standard\_error\_577 = current\_standard\_dev\_577/ sqrt(3);

% Find imax

figure

fitop\_577 = fitoptions('Method','NonlinearLeastSquares','StartPoint',[4e-8 8e-10 0.08]);

imax\_fittype\_577 = fittype('a-b\*exp(-c\*x)','options',fitop\_577);

imax\_577\_fit = fit(voltagetouse577,averagecurrent577',imax\_fittype\_577);

imax\_577\_fit\_coeff = coeffvalues(imax\_577\_fit);

errorbar(voltagetouse577, averagecurrent577',current\_standard\_error\_577,current\_standard\_error\_577,voltage\_unc\_577,voltage\_unc\_577);

hold on;

plot(imax\_577\_fit);

grid on;

imax\_577 = imax\_577\_fit\_coeff(1);

i\_max\_fit\_confint = confint(imax\_577\_fit, 0.68);

imax\_577\_unc = (i\_max\_fit\_confint(2,1)-i\_max\_fit\_confint(1,1))/2;

title("Fitted Exponential Curve to Forward Bias Current vs. Voltage (577nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

%%% REVERSE

% Calculate average of three runs

firstrun\_reverse\_577 = reverse\_current\_A577(1:41);

secondrun\_reverse\_577 = reverse\_current\_A577(42:82);

thirdrun\_reverse\_577 = reverse\_current\_A577(83:123);

averagecurrent\_reverse\_577 =[];

current\_reverse\_standard\_dev\_577 = [];

for i = 1:41;

averagecurrent\_reverse\_577i = (firstrun\_reverse\_577(i) + secondrun\_reverse\_577(i) + thirdrun\_reverse\_577(i)) / 3;

current\_reverse\_standard\_dev\_577(i) = std([firstrun577(i); secondrun577(i); thirdrun577(i)]);

averagecurrent\_reverse\_577(i) = averagecurrent\_reverse\_577i;

end

voltagetouse\_reverse\_577 = reverse\_voltage\_V577(1:41);

voltagetouse\_reverse\_577 = -voltagetouse\_reverse\_577;

% Uncertainty in Voltage

voltage\_unc\_reverse\_577 = 0.001\*ones(size(voltagetouse\_reverse\_577));

% Uncertainty in Current

current\_reverse\_standard\_error\_577 = current\_reverse\_standard\_dev\_577/ sqrt(3);

% Plot a figure with both data sets on one graph

figure

errorbar(voltagetouse577, averagecurrent577',current\_standard\_error\_577,current\_standard\_error\_577,voltage\_unc\_577,voltage\_unc\_577);

hold on;

errorbar(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577,current\_reverse\_standard\_error\_577,current\_reverse\_standard\_error\_577,voltage\_unc\_reverse\_577,voltage\_unc\_reverse\_577)

plot(voltagetouse577, averagecurrent577');

plot(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577);

grid on;

title("Current vs. Voltage (577nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the forward bias

figure

errorbar(voltagetouse577, averagecurrent577',current\_standard\_error\_577,current\_standard\_error\_577,voltage\_unc\_577,voltage\_unc\_577);

title("Forward Bias Current vs. Voltage (577nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the reverse bias

figure

errorbar(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577,current\_reverse\_standard\_error\_577,current\_reverse\_standard\_error\_577,voltage\_unc\_reverse\_577,voltage\_unc\_reverse\_577)

title("Reverse Bias Current vs. Voltage (577nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Find knee using plateau uncertaities and deviation from that value

% Calculate aveage data point

running\_total\_577 = 0;

h577 = averagecurrent\_reverse\_577(20:35);

for i = 1:6;

running\_total\_577 = h577(i) + running\_total\_577;

end

average\_threshold\_point\_577 = running\_total\_577 / 6;

% Calculate average error bar size

running\_total\_577 = 0;

g577 = current\_reverse\_standard\_error\_577(20:35);

for i = 1:6;

running\_total\_577 = g577(i)^2 + running\_total\_577;

end

uncertainty\_mean\_577 = sqrt(running\_total\_577)/6;

% Calculate standard deviation

standard\_dev\_577 = std(current\_reverse\_standard\_error\_577(20:35));

% Calculate threshold bar

threshold\_bar\_577 = sqrt(uncertainty\_mean\_577^2 + standard\_dev\_577^2);

% Calculate threshold

threshold\_577 = threshold\_bar\_577 + average\_threshold\_point\_577;

% Plot the threshold line

figure

errorbar(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577,current\_reverse\_standard\_error\_577,current\_reverse\_standard\_error\_577,voltage\_unc\_reverse\_577,voltage\_unc\_reverse\_577)

title("Threshold Value for Reverse Bias Current vs. Voltage (577nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

x = [-2 : 0.5 : 0];

Z = threshold\_577 \* ones(1, length(x));

plot(x, Z)

grid on;

% Find knee using intersection of flat slopes

% Calculating slope of top line

k577 = averagecurrent\_reverse\_577(1:9);

j577 = voltagetouse\_reverse\_577(1:9);

% Plot intersection of lines

figure

errorbar(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577,current\_reverse\_standard\_error\_577,current\_reverse\_standard\_error\_577,voltage\_unc\_reverse\_577,voltage\_unc\_reverse\_577)

title("Slope Intersection for Reverse Bias Current vs. Voltage (577nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

% Plot top line

c\_577 = polyfit(j577,k577',1);

xFit\_577 = linspace(-2, 0, 100);

yFit\_577 = polyval(c\_577, xFit\_577);

hold on;

plot(xFit\_577, yFit\_577);

grid on;

% Plot bottom line

x = [-2 : 0.5 : 0];

% Use already calculated value for average\_threshold\_point to get bottom

% line

Z\_577 = average\_threshold\_point\_577 \* ones(1, length(x));

plot(x, Z\_577)

grid on;

%%%%% 546 analysis

%%% FORWARD

% Calculate average of three runs

firstrun546 = forward\_current\_A546(1:301);

secondrun546 = forward\_current\_A546(302:602);

thirdrun546 = forward\_current\_A546(603:903);

averagecurrent546 = [];

current\_standard\_dev\_546 = [];

for i = 1:301;

averagecurrent546i = (firstrun546(i) + secondrun546(i) + thirdrun546(i))/3;

current\_standard\_dev\_546(i) = std([firstrun546(i); secondrun546(i); thirdrun546(i)]);

averagecurrent546(i) = averagecurrent546i;

end

voltagetouse546 = forward\_voltage\_V546(1:301);

% Uncertainty in Voltage

voltage\_unc\_546 = 0.001\*ones(size(voltagetouse546));

% Uncertainty in Current

current\_standard\_error\_546 = current\_standard\_dev\_546/ sqrt(3);

% Find imax

figure

fitop\_546 = fitoptions('Method','NonlinearLeastSquares','StartPoint',[6e-8 1e-10 0.06]);

imax\_fittype\_546 = fittype('a-b\*exp(-c\*x)','options',fitop\_546);

imax\_546\_fit = fit(voltagetouse546,averagecurrent546',imax\_fittype\_546);

imax\_546\_fit\_coeff = coeffvalues(imax\_546\_fit);

errorbar(voltagetouse546, averagecurrent546',current\_standard\_error\_546,current\_standard\_error\_546,voltage\_unc\_546,voltage\_unc\_546);

hold on;

plot(imax\_546\_fit);

grid on;

imax\_546 = imax\_546\_fit\_coeff(1);

i\_max\_fit\_confint = confint(imax\_546\_fit, 0.68);

imax\_546\_unc = (i\_max\_fit\_confint(2,1)-i\_max\_fit\_confint(1,1))/2;

title("Fitted Exponential Curve to Forward Bias Current vs. Voltage (546nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

%%% REVERSE

% Calculate average of three runs

firstrun\_reverse\_546 = reverse\_current\_A546(1:41);

secondrun\_reverse\_546 = reverse\_current\_A546(42:82);

thirdrun\_reverse\_546 = reverse\_current\_A546(83:123);

averagecurrent\_reverse\_546 =[];

current\_reverse\_standard\_dev\_546 = [];

for i = 1:41;

averagecurrent\_reverse\_546i = (firstrun\_reverse\_546(i) + secondrun\_reverse\_546(i) + thirdrun\_reverse\_546(i)) / 3;

current\_reverse\_standard\_dev\_546(i) = std([firstrun546(i); secondrun546(i); thirdrun546(i)]);

averagecurrent\_reverse\_546(i) = averagecurrent\_reverse\_546i;

end

voltagetouse\_reverse\_546 = reverse\_voltage\_V546(1:41);

voltagetouse\_reverse\_546 = -voltagetouse\_reverse\_546;

% Uncertainty in Voltage

voltage\_unc\_reverse\_546 = 0.001\*ones(size(voltagetouse\_reverse\_546));

% Uncertainty in Current

current\_reverse\_standard\_error\_546 = current\_reverse\_standard\_dev\_546/ sqrt(3);

% Plot a figure with both data sets on one graph

figure

errorbar(voltagetouse546, averagecurrent546',current\_standard\_error\_546,current\_standard\_error\_546,voltage\_unc\_546,voltage\_unc\_546);

hold on;

errorbar(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546,current\_reverse\_standard\_error\_546,current\_reverse\_standard\_error\_546,voltage\_unc\_reverse\_546,voltage\_unc\_reverse\_546)

plot(voltagetouse546, averagecurrent546');

grid on;

plot(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546);

title("Current vs. Voltage (546nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the forward bias

figure

errorbar(voltagetouse546, averagecurrent546',current\_standard\_error\_546,current\_standard\_error\_546,voltage\_unc\_546,voltage\_unc\_546);

title("Forward Bias Current vs. Voltage (546nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the reverse bias

figure

errorbar(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546,current\_reverse\_standard\_error\_546,current\_reverse\_standard\_error\_546,voltage\_unc\_reverse\_546,voltage\_unc\_reverse\_546)

title("Reverse Bias Current vs. Voltage (546nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Find knee using plateau uncertaities and deviation from that value

% Calculate aveage data point

running\_total\_546 = 0;

h546 = averagecurrent\_reverse\_546(35:40);

for i = 1:6;

running\_total\_546 = h546(i) + running\_total\_546;

end

average\_threshold\_point\_546 = running\_total\_546 / 6;

% Calculate average error bar size

running\_total\_546 = 0;

g546 = current\_reverse\_standard\_error\_546(35:40);

for i = 1:6;

running\_total\_546 = g546(i)^2 + running\_total\_546;

end

uncertainty\_mean\_546 = sqrt(running\_total\_546)/6;

% Calculate standard deviation

standard\_dev\_546 = std(current\_reverse\_standard\_error\_546(35:40));

% Calculate threshold bar

threshold\_bar\_546 = sqrt(uncertainty\_mean\_546^2 + standard\_dev\_546^2);

% Calculate threshold

threshold\_546 = threshold\_bar\_546 + average\_threshold\_point\_546;

% Plot the threshold line

figure

errorbar(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546,current\_reverse\_standard\_error\_546,current\_reverse\_standard\_error\_546,voltage\_unc\_reverse\_546,voltage\_unc\_reverse\_546)

title("Threshold Value for Reverse Bias Current vs. Voltage (546nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

x = [-2 : 0.5 : 0];

Z = threshold\_546 \* ones(1, length(x));

plot(x, Z)

grid on;

% Find knee using intersection of flat slopes

% Calculating slope of top line

k546 = averagecurrent\_reverse\_546(1:8);

j546 = voltagetouse\_reverse\_546(1:8);

% Plot intersection of lines

figure

errorbar(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546,current\_reverse\_standard\_error\_546,current\_reverse\_standard\_error\_546,voltage\_unc\_reverse\_546,voltage\_unc\_reverse\_546)

title("Slope Intersection for Reverse Bias Current vs. Voltage (546nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

% Plot top line

c\_546 = polyfit(j546,k546',1);

xFit\_546 = linspace(-2, 0, 100);

yFit\_546 = polyval(c\_546, xFit\_546);

hold on;

plot(xFit\_546, yFit\_546);

grid on;

% Plot bottom line

x = [-2 : 0.5 : 0];

% Use already calculated value for average\_threshold\_point to get bottom

% line

Z\_546 = average\_threshold\_point\_546 \* ones(1, length(x));

plot(x, Z\_546)

grid on;

%%%%% 436 analysis

%%% FORWARD

% Calculate average of three runs

firstrun436 = forward\_current\_A436(1:301);

secondrun436 = forward\_current\_A436(302:602);

thirdrun436 = forward\_current\_A436(603:903);

averagecurrent436 = [];

current\_standard\_dev\_436 = [];

for i = 1:301;

averagecurrent436i = (firstrun436(i) + secondrun436(i) + thirdrun436(i))/3;

current\_standard\_dev\_436(i) = std([firstrun436(i); secondrun436(i); thirdrun436(i)]);

averagecurrent436(i) = averagecurrent436i;

end

voltagetouse436 = forward\_voltage\_V436(1:301);

% Uncertainty in Voltage

voltage\_unc\_436 = 0.001\*ones(size(voltagetouse436));

% Uncertainty in Current

current\_standard\_error\_436 = current\_standard\_dev\_436/ sqrt(3);

% Find imax

figure

fitop\_436 = fitoptions('Method','NonlinearLeastSquares','StartPoint',[8e-8 1e-10 0.045]);

imax\_fittype\_436 = fittype('a-b\*exp(-c\*x)','options',fitop\_436);

imax\_436\_fit = fit(voltagetouse436,averagecurrent436',imax\_fittype\_436);

imax\_436\_fit\_coeff = coeffvalues(imax\_436\_fit);

errorbar(voltagetouse436, averagecurrent436',current\_standard\_error\_436,current\_standard\_error\_436,voltage\_unc\_436,voltage\_unc\_436);

hold on;

plot(imax\_436\_fit);

grid on;

imax\_436 = imax\_436\_fit\_coeff(1);

i\_max\_fit\_confint = confint(imax\_436\_fit, 0.68);

imax\_436\_unc = (i\_max\_fit\_confint(2,1)-i\_max\_fit\_confint(1,1))/2;

title("Fitted Exponential Curve to Forward Bias Current vs. Voltage (436nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

%%% REVERSE

% Calculate average of three runs

firstrun\_reverse\_436 = reverse\_current\_A436(1:41);

secondrun\_reverse\_436 = reverse\_current\_A436(42:82);

thirdrun\_reverse\_436 = reverse\_current\_A436(83:123);

averagecurrent\_reverse\_436 =[];

current\_reverse\_standard\_dev\_436 = [];

for i = 1:41;

averagecurrent\_reverse\_436i = (firstrun\_reverse\_436(i) + secondrun\_reverse\_436(i) + thirdrun\_reverse\_436(i)) / 3;

current\_reverse\_standard\_dev\_436(i) = std([firstrun436(i); secondrun436(i); thirdrun436(i)]);

averagecurrent\_reverse\_436(i) = averagecurrent\_reverse\_436i;

end

voltagetouse\_reverse\_436 = reverse\_voltage\_V436(1:41);

voltagetouse\_reverse\_436 = -voltagetouse\_reverse\_436;

% Uncertainty in Voltage

voltage\_unc\_reverse\_436 = 0.001\*ones(size(voltagetouse\_reverse\_436));

% Uncertainty in Current

current\_reverse\_standard\_error\_436 = current\_reverse\_standard\_dev\_436/ sqrt(3);

% Plot a figure with both data sets on one graph

figure

errorbar(voltagetouse436, averagecurrent436',current\_standard\_error\_436,current\_standard\_error\_436,voltage\_unc\_436,voltage\_unc\_436);

hold on;

errorbar(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436,current\_reverse\_standard\_error\_436,current\_reverse\_standard\_error\_436,voltage\_unc\_reverse\_436,voltage\_unc\_reverse\_436)

plot(voltagetouse436, averagecurrent436');

plot(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436);

grid on;

title("Current vs. Voltage (436nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the forward bias

figure

errorbar(voltagetouse436, averagecurrent436',current\_standard\_error\_436,current\_standard\_error\_436,voltage\_unc\_436,voltage\_unc\_436);

title("Forward Bias Current vs. Voltage (436nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the reverse bias

figure

errorbar(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436,current\_reverse\_standard\_error\_436,current\_reverse\_standard\_error\_436,voltage\_unc\_reverse\_436,voltage\_unc\_reverse\_436)

title("Reverse Bias Current vs. Voltage (436nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Find knee using plateau uncertaities and deviation from that value

% Calculate aveage data point

running\_total\_436 = 0;

h436 = averagecurrent\_reverse\_436(35:40);

for i = 1:6;

running\_total\_436 = h436(i) + running\_total\_436;

end

average\_threshold\_point\_436 = running\_total\_436 / 6;

% Calculate average error bar size

running\_total\_436 = 0;

g436 = current\_reverse\_standard\_error\_436(35:40);

for i = 1:6;

running\_total\_436 = g436(i)^2 + running\_total\_436;

end

uncertainty\_mean\_436 = sqrt(running\_total\_436)/6;

% Calculate standard deviation

standard\_dev\_436 = std(current\_reverse\_standard\_error\_436(35:40));

% Calculate threshold bar

threshold\_bar\_436 = sqrt(uncertainty\_mean\_436^2 + standard\_dev\_436^2);

% Calculate threshold

threshold\_436 = threshold\_bar\_436 + average\_threshold\_point\_436;

% Plot the threshold line

figure

errorbar(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436,current\_reverse\_standard\_error\_436,current\_reverse\_standard\_error\_436,voltage\_unc\_reverse\_436,voltage\_unc\_reverse\_436)

title("Threshold Value for Reverse Bias Current vs. Voltage (436nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

x = [-2 : 0.5 : 0];

Z = threshold\_436 \* ones(1, length(x));

plot(x, Z)

grid on;

% Find knee using intersection of flat slopes

% Calculating slope of top line

k436 = averagecurrent\_reverse\_436(1:11);

j436 = voltagetouse\_reverse\_436(1:11);

% Plot intersection of lines

figure

errorbar(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436,current\_reverse\_standard\_error\_436,current\_reverse\_standard\_error\_436,voltage\_unc\_reverse\_436,voltage\_unc\_reverse\_436)

title("Slope Intersection for Reverse Bias Current vs. Voltage (436nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

% Plot top line

c\_436 = polyfit(j436,k436',1);

xFit\_436 = linspace(-2, 0, 100);

yFit\_436 = polyval(c\_436, xFit\_436);

hold on;

plot(xFit\_436, yFit\_436);

grid on;

% Plot bottom line

x = [-2 : 0.5 : 0];

% Use already calculated value for average\_threshold\_point to get bottom

% line

Z\_436 = average\_threshold\_point\_436 \* ones(1, length(x));

plot(x, Z\_436)

grid on;

%%%%% 405 analysis

%%% FORWARD

% Calculate average of three runs

firstrun405 = forward\_current\_A405(1:301);

secondrun405 = forward\_current\_A405(302:602);

thirdrun405 = forward\_current\_A405(603:903);

averagecurrent405 = [];

current\_standard\_dev\_405 = [];

for i = 1:301;

averagecurrent405i = (firstrun405(i) + secondrun405(i) + thirdrun405(i))/3;

current\_standard\_dev\_405(i) = std([firstrun405(i); secondrun405(i); thirdrun405(i)]);

averagecurrent405(i) = averagecurrent405i;

end

voltagetouse405 = forward\_voltage\_V405(1:301);

% Uncertainty in Voltage

voltage\_unc\_405 = 0.001\*ones(size(voltagetouse405));

% Uncertainty in Current

current\_standard\_error\_405 = current\_standard\_dev\_405/ sqrt(3);

% Find imax

figure

fitop\_405 = fitoptions('Method','NonlinearLeastSquares','StartPoint',[4e-8 4e-10 0.04]);

imax\_fittype\_405 = fittype('a-b\*exp(-c\*x)','options',fitop\_405);

imax\_405\_fit = fit(voltagetouse405,averagecurrent405',imax\_fittype\_405);

imax\_405\_fit\_coeff = coeffvalues(imax\_405\_fit);

errorbar(voltagetouse405, averagecurrent405',current\_standard\_error\_405,current\_standard\_error\_405,voltage\_unc\_405,voltage\_unc\_405);

hold on;

plot(imax\_405\_fit);

grid on;

imax\_405 = imax\_405\_fit\_coeff(1);

i\_max\_fit\_confint = confint(imax\_405\_fit, 0.68);

imax\_405\_unc = (i\_max\_fit\_confint(2,1)-i\_max\_fit\_confint(1,1))/2;

title("Fitted Exponential Curve to Forward Bias Current vs. Voltage (405nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

%%% REVERSE

% Calculate average of three runs

firstrun\_reverse\_405 = reverse\_current\_A405(1:41);

secondrun\_reverse\_405 = reverse\_current\_A405(42:82);

thirdrun\_reverse\_405 = reverse\_current\_A405(83:123);

averagecurrent\_reverse\_405 =[];

current\_reverse\_standard\_dev\_405 = [];

for i = 1:41;

averagecurrent\_reverse\_405i = (firstrun\_reverse\_405(i) + secondrun\_reverse\_405(i) + thirdrun\_reverse\_405(i)) / 3;

current\_reverse\_standard\_dev\_405(i) = std([firstrun405(i); secondrun405(i); thirdrun405(i)]);

averagecurrent\_reverse\_405(i) = averagecurrent\_reverse\_405i;

end

voltagetouse\_reverse\_405 = reverse\_voltage\_V405(1:41);

voltagetouse\_reverse\_405 = -voltagetouse\_reverse\_405;

% Uncertainty in Voltage

voltage\_unc\_reverse\_405 = 0.001\*ones(size(voltagetouse\_reverse\_405));

% Uncertainty in Current

current\_reverse\_standard\_error\_405 = current\_reverse\_standard\_dev\_405/ sqrt(3);

% Plot a figure with both data sets on one graph

figure

errorbar(voltagetouse405, averagecurrent405',current\_standard\_error\_405,current\_standard\_error\_405,voltage\_unc\_405,voltage\_unc\_405);

hold on;

errorbar(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405,current\_reverse\_standard\_error\_405,current\_reverse\_standard\_error\_405,voltage\_unc\_reverse\_405,voltage\_unc\_reverse\_405)

plot(voltagetouse405, averagecurrent405');

plot(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405);

grid on;

title("Current vs. Voltage (405nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the forward bias

figure

errorbar(voltagetouse405, averagecurrent405',current\_standard\_error\_405,current\_standard\_error\_405,voltage\_unc\_405,voltage\_unc\_405);

title("Forward Bias Current vs. Voltage (405nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the reverse bias

figure

errorbar(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405,current\_reverse\_standard\_error\_405,current\_reverse\_standard\_error\_405,voltage\_unc\_reverse\_405,voltage\_unc\_reverse\_405)

title("Reverse Bias Current vs. Voltage (405nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Find knee using plateau uncertaities and deviation from that value

% Calculate aveage data point

running\_total\_405 = 0;

h405 = averagecurrent\_reverse\_405(35:40);

for i = 1:6;

running\_total\_405 = h405(i) + running\_total\_405;

end

average\_threshold\_point\_405 = running\_total\_405 / 6;

% Calculate average error bar size

running\_total\_405 = 0;

g405 = current\_reverse\_standard\_error\_405(35:40);

for i = 1:6;

running\_total\_405 = g405(i)^2 + running\_total\_405;

end

uncertainty\_mean\_405 = sqrt(running\_total\_405)/6;

% Calculate standard deviation

standard\_dev\_405 = std(current\_reverse\_standard\_error\_405(35:40));

% Calculate threshold bar

threshold\_bar\_405 = sqrt(uncertainty\_mean\_405^2 + standard\_dev\_405^2);

% Calculate threshold

threshold\_405 = threshold\_bar\_405 + average\_threshold\_point\_405;

% Plot the threshold line

figure

errorbar(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405,current\_reverse\_standard\_error\_405,current\_reverse\_standard\_error\_405,voltage\_unc\_reverse\_405,voltage\_unc\_reverse\_405)

title("Threshold Value for Reverse Bias Current vs. Voltage (405nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

x = [-2 : 0.5 : 0];

Z = threshold\_405 \* ones(1, length(x));

plot(x, Z)

grid on;

% Find knee using intersection of flat slopes

% Calculating slope of top line

k405 = averagecurrent\_reverse\_405(3:16);

j405 = voltagetouse\_reverse\_405(3:16);

% Plot intersection of lines

figure

errorbar(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405,current\_reverse\_standard\_error\_405,current\_reverse\_standard\_error\_405,voltage\_unc\_reverse\_405,voltage\_unc\_reverse\_405)

title("Slope Intersection for Reverse Bias Current vs. Voltage (405nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

% Plot top line

c\_405 = polyfit(j405,k405',1);

xFit\_405 = linspace(-2, 0, 100);

yFit\_405 = polyval(c\_405, xFit\_405);

hold on;

plot(xFit\_405, yFit\_405);

grid on;

% Plot bottom line

x = [-2 : 0.5 : 0];

% Use already calculated value for average\_threshold\_point to get bottom

% line

Z\_405 = average\_threshold\_point\_405 \* ones(1, length(x));

plot(x, Z\_405)

grid on;

%%%%% 365 analysis

%%% FORWARD

% Calculate average of three runs

firstrun365 = forward\_current\_A365(1:301);

secondrun365 = forward\_current\_A365(302:602);

thirdrun365 = forward\_current\_A365(603:903);

averagecurrent365 = [];

current\_standard\_dev\_365 = [];

for i = 1:301;

averagecurrent365i = (firstrun365(i) + secondrun365(i) + thirdrun365(i))/3;

current\_standard\_dev\_365(i) = std([firstrun365(i); secondrun365(i); thirdrun365(i)]);

averagecurrent365(i) = averagecurrent365i;

end

voltagetouse365 = forward\_voltage\_V365(1:301);

% Uncertainty in Voltage

voltage\_unc\_365 = 0.001\*ones(size(voltagetouse365));

% Uncertainty in Current

current\_standard\_error\_365 = current\_standard\_dev\_365/ sqrt(3);

% Find imax

figure

fitop\_365 = fitoptions('Method','NonlinearLeastSquares','StartPoint',[1e-8 4e-10 0.04]);

imax\_fittype\_365 = fittype('a-b\*exp(-c\*x)','options',fitop\_365);

imax\_365\_fit = fit(voltagetouse365,averagecurrent365',imax\_fittype\_365);

imax\_365\_fit\_coeff = coeffvalues(imax\_365\_fit);

errorbar(voltagetouse365, averagecurrent365',current\_standard\_error\_365,current\_standard\_error\_365,voltage\_unc\_365,voltage\_unc\_365);

hold on;

plot(imax\_365\_fit);

grid on;

imax\_365 = imax\_365\_fit\_coeff(1);

i\_max\_fit\_confint = confint(imax\_365\_fit, 0.68);

imax\_365\_unc = (i\_max\_fit\_confint(2,1)-i\_max\_fit\_confint(1,1))/2;

title("Fitted Exponential Curve to Forward Bias Current vs. Voltage (365nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

%%% REVERSE

% Calculate average of three runs

firstrun\_reverse\_365 = reverse\_current\_A365(1:41);

secondrun\_reverse\_365 = reverse\_current\_A365(42:82);

thirdrun\_reverse\_365 = reverse\_current\_A365(83:123);

averagecurrent\_reverse\_365 =[];

current\_reverse\_standard\_dev\_365 = [];

for i = 1:41;

averagecurrent\_reverse\_365i = (firstrun\_reverse\_365(i) + secondrun\_reverse\_365(i) + thirdrun\_reverse\_365(i)) / 3;

current\_reverse\_standard\_dev\_365(i) = std([firstrun365(i); secondrun365(i); thirdrun365(i)]);

averagecurrent\_reverse\_365(i) = averagecurrent\_reverse\_365i;

end

voltagetouse\_reverse\_365 = reverse\_voltage\_V365(1:41);

voltagetouse\_reverse\_365 = -voltagetouse\_reverse\_365;

% Uncertainty in Voltage

voltage\_unc\_reverse\_365 = 0.001\*ones(size(voltagetouse\_reverse\_365));

% Uncertainty in Current

current\_reverse\_standard\_error\_365 = current\_reverse\_standard\_dev\_365/ sqrt(3);

% Plot a figure with both data sets on one graph

figure

errorbar(voltagetouse365, averagecurrent365',current\_standard\_error\_365,current\_standard\_error\_365,voltage\_unc\_365,voltage\_unc\_365);

hold on;

errorbar(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365,current\_reverse\_standard\_error\_365,current\_reverse\_standard\_error\_365,voltage\_unc\_reverse\_365,voltage\_unc\_reverse\_365)

plot(voltagetouse365, averagecurrent365');

plot(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365);

grid on;

title("Current vs. Voltage (365nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the forward bias

figure

errorbar(voltagetouse365, averagecurrent365',current\_standard\_error\_365,current\_standard\_error\_365,voltage\_unc\_365,voltage\_unc\_365);

title("Forward Bias Current vs. Voltage (365nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Plot just the reverse bias

figure

errorbar(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365,current\_reverse\_standard\_error\_365,current\_reverse\_standard\_error\_365,voltage\_unc\_reverse\_365,voltage\_unc\_reverse\_365)

title("Reverse Bias Current vs. Voltage (365nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

% Find knee using plateau uncertaities and deviation from that value

% Calculate aveage data point

running\_total\_365 = 0;

h365 = averagecurrent\_reverse\_365(35:40);

for i = 1:6;

running\_total\_365 = h365(i) + running\_total\_365;

end

average\_threshold\_point\_365 = running\_total\_365 / 6;

% Calculate average error bar size

running\_total\_365 = 0;

g365 = current\_reverse\_standard\_error\_365(35:40);

for i = 1:6;

running\_total\_365 = g365(i)^2 + running\_total\_365;

end

uncertainty\_mean\_365 = sqrt(running\_total\_365)/6;

% Calculate standard deviation

standard\_dev\_365 = std(current\_reverse\_standard\_error\_365(35:40));

% Calculate threshold bar

threshold\_bar\_365 = sqrt(uncertainty\_mean\_365^2 + standard\_dev\_365^2);

% Calculate threshold

threshold\_365 = threshold\_bar\_365 + average\_threshold\_point\_365;

% Plot the threshold line

figure

errorbar(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365,current\_reverse\_standard\_error\_365,current\_reverse\_standard\_error\_365,voltage\_unc\_reverse\_365,voltage\_unc\_reverse\_365)

title("Threshold Value for Reverse Bias Current vs. Voltage (365nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

x = [-2 : 0.5 : 0];

Z = threshold\_365 \* ones(1, length(x));

plot(x, Z)

grid on;

% Find knee using intersection of flat slopes

% Calculating slope of top line

k365 = averagecurrent\_reverse\_365(5:21);

j365 = voltagetouse\_reverse\_365(5:21);

% Plot intersection of lines

figure

errorbar(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365,current\_reverse\_standard\_error\_365,current\_reverse\_standard\_error\_365,voltage\_unc\_reverse\_365,voltage\_unc\_reverse\_365)

title("Slope Intersection for Reverse Bias Current vs. Voltage (365nm Filter)")

xlabel('Voltage (V)')

ylabel('Current (A)')

hold on;

% Plot top line

c\_365 = polyfit(j365,k365',1);

xFit\_365 = linspace(-2, 0, 100);

yFit\_365 = polyval(c\_365, xFit\_365);

hold on;

plot(xFit\_365, yFit\_365);

grid on;

% Plot bottom line

x = [-2 : 0.5 : 0];

% Use already calculated value for average\_threshold\_point to get bottom

% line

Z\_365 = average\_threshold\_point\_365 \* ones(1, length(x));

plot(x, Z\_365)

grid on;

%%%%% Stopping voltage vs. frequency analysis

two\_line\_stop\_V = [0.47, 0.50, 0.85, 1.08, 1.26];

one\_line\_stop\_V = [0.5, 0.5, 0.9, 1.1, 1.35];

voltage\_unc = [.05, .05, .05, .05, .05]

average\_voltage\_unc = [.035, .035, .035, .035, .035]

frequency\_unc = [0, 0, 0, 0, 0]

% Average values from both methods

avg\_stop\_V = [0.485, 0.50, 0.875, 1.09, 1.305];

avg\_stop\_V\_top\_big\_bottom\_small = [0.535, 1.255];

avg\_stop\_V\_bottom\_big\_top\_small = [0.435, 1.355];

frequency = [5.20\*10^14, 5.49\*10^14, 6.88\*10^14, 7.41\*10^14, 8.22\*10^14];

frequency\_extremes = [5.20\*10^14, 8.22\*10^14];

figure

errorbar(frequency, avg\_stop\_V, average\_voltage\_unc, average\_voltage\_unc, frequency\_unc, frequency\_unc, 'o')

grid on;

hold on;

fit1 = polyfit(frequency,avg\_stop\_V,1);

yFit1 = polyval(fit1, frequency);

plot(frequency,yFit1)

title("Averaged Stopping Voltage vs. Light Frequency")

xlabel("Light Frequency (Hz)")

ylabel("Stopping Voltage (V)")

legend('Average Data', 'Fit of Average Data')

figure

errorbar(frequency, avg\_stop\_V, voltage\_unc, voltage\_unc, frequency\_unc, frequency\_unc, 'o')

grid on;

hold on;

fit1 = polyfit(frequency,avg\_stop\_V,1);

yFit1 = polyval(fit1, frequency);

plot(frequency,yFit1)

fit2 = polyfit(frequency\_extremes,avg\_stop\_V\_top\_big\_bottom\_small,1);

yFit2 = polyval(fit2, frequency\_extremes);

plot(frequency\_extremes,yFit2)

fit3 = polyfit(frequency\_extremes,avg\_stop\_V\_bottom\_big\_top\_small,1);

yFit3 = polyval(fit3, frequency\_extremes);

plot(frequency\_extremes,yFit3)

title("Averaged Stopping Voltage vs. Light Frequency with Minimum and Maximum Linear Fits")

xlabel("Light Frequency (Hz)")

ylabel("Stopping Voltage (V)")

legend('Average Data', 'Fit of Average Data', 'Smallest Work Function Fit', 'Largest Work Function Fit')

avg\_work\_func = -fit1(2)

avg\_slope = fit1(1)

large\_work\_func = -fit3(2)

small\_slope = fit3(1)

small\_work\_func = -fit2(2)

large\_slope = fit2(1)

hold off;

figure

errorbar(frequency, two\_line\_stop\_V, average\_voltage\_unc, average\_voltage\_unc, frequency\_unc, frequency\_unc, 'ro')

grid on;

hold on;

fit1 = polyfit(frequency,two\_line\_stop\_V,1);

xFit = linspace(0, 8.5e14, 100);

yFit1 = polyval(fit1, frequency);

plot(frequency,yFit1,'r')

xlabel("Light Frequency (Hz)")

ylabel("Stopping Voltage (V)")

errorbar(frequency, one\_line\_stop\_V, voltage\_unc, voltage\_unc, frequency\_unc, frequency\_unc, 'b\*')

hold on;

fit2 = polyfit(frequency,one\_line\_stop\_V,1);

xFit2 = linspace(0, 8.5e14, 100);

yFit2 = polyval(fit2, frequency);

plot(frequency,yFit2,'b')

title("Stopping Voltage vs. Light Frequency with Both Methods")

legend('Intersection Method Data','Intersection Method Fit','Threashold Method Data','Threashold Method Fit')

grid on;

% Plot a figure with both data sets on one graph for forward bias

figure

errorbar(voltagetouse405, averagecurrent405',current\_standard\_error\_405,current\_standard\_error\_405,voltage\_unc\_405,voltage\_unc\_405);

hold on;

plot(voltagetouse405, averagecurrent405');

errorbar(voltagetouse365, averagecurrent365',current\_standard\_error\_365,current\_standard\_error\_365,voltage\_unc\_365,voltage\_unc\_365);

hold on;

plot(voltagetouse405, averagecurrent365');

errorbar(voltagetouse436, averagecurrent436',current\_standard\_error\_436,current\_standard\_error\_436,voltage\_unc\_436,voltage\_unc\_436);

hold on;

plot(voltagetouse405, averagecurrent436');

errorbar(voltagetouse546, averagecurrent546',current\_standard\_error\_546,current\_standard\_error\_546,voltage\_unc\_546,voltage\_unc\_546);

hold on;

plot(voltagetouse405, averagecurrent546');

errorbar(voltagetouse577, averagecurrent577',current\_standard\_error\_577,current\_standard\_error\_577,voltage\_unc\_577,voltage\_unc\_577);

hold on;

plot(voltagetouse405, averagecurrent577');

grid on;

title("Current vs. Voltage Forward Bias (All Filters)")

xlabel('Voltage (V)')

ylabel('Current (A)')

%Plot a figure with both data sets on one graph for reverse bias

figure

errorbar(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405,current\_reverse\_standard\_error\_405,current\_reverse\_standard\_error\_405,voltage\_unc\_reverse\_405,voltage\_unc\_reverse\_405)

hold on;

plot(voltagetouse\_reverse\_405, averagecurrent\_reverse\_405);

errorbar(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365,current\_reverse\_standard\_error\_365,current\_reverse\_standard\_error\_365,voltage\_unc\_reverse\_365,voltage\_unc\_reverse\_365)

plot(voltagetouse\_reverse\_365, averagecurrent\_reverse\_365);

errorbar(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436,current\_reverse\_standard\_error\_436,current\_reverse\_standard\_error\_436,voltage\_unc\_reverse\_436,voltage\_unc\_reverse\_436)

plot(voltagetouse\_reverse\_436, averagecurrent\_reverse\_436);

errorbar(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546,current\_reverse\_standard\_error\_546,current\_reverse\_standard\_error\_546,voltage\_unc\_reverse\_546,voltage\_unc\_reverse\_546)

plot(voltagetouse\_reverse\_546, averagecurrent\_reverse\_546);

errorbar(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577,current\_reverse\_standard\_error\_577,current\_reverse\_standard\_error\_577,voltage\_unc\_reverse\_577,voltage\_unc\_reverse\_577)

plot(voltagetouse\_reverse\_577, averagecurrent\_reverse\_577);

title("Current vs. Voltage Reverse Bias (All Filters)")

xlabel('Voltage (V)')

ylabel('Current (A)')

grid on;

TODO:

* References
* Spell/grammar check
* Cross reference other lab handouts for relevant information
* More background / historical info?
* Readability of figures.
* Flickering of the lamp