Lab_12_Planck's_Constant

Paige Brady and Austin Jones

02-26-2020

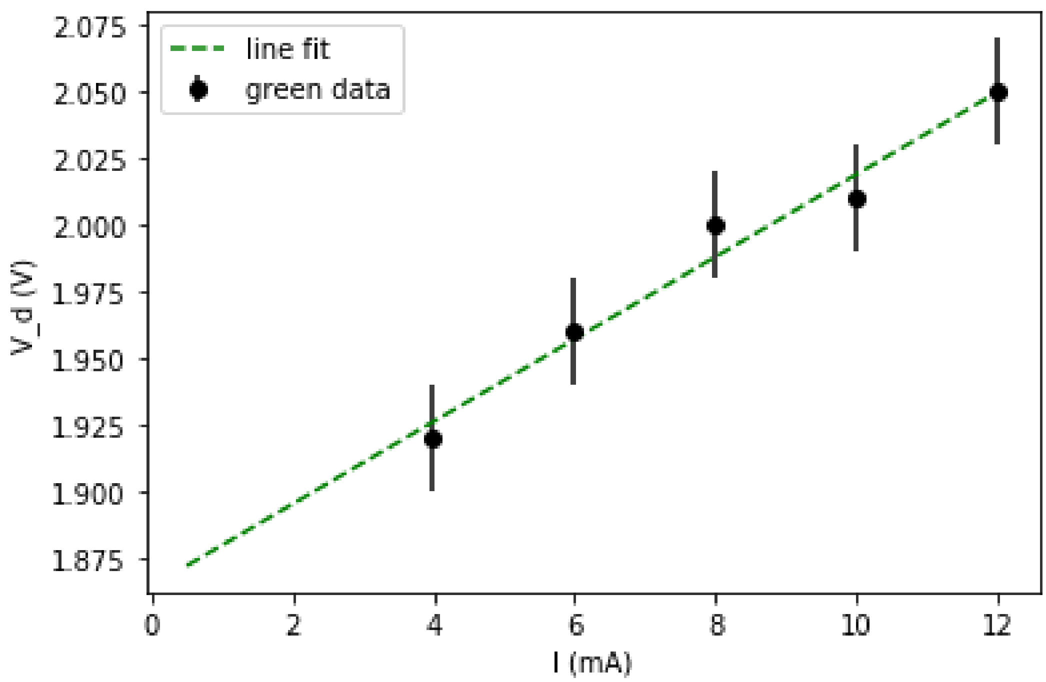
[36]:
```
%pylab inline
from scipy import optimize
```

```
Populating the interactive namespace from numpy and matplotlib
```

[108]:
```
def line_func(x, a, b):
    return x*a+b

#x values = I(mA) for green LED
x_green = np.array([3.99,5.99,8.00,10.00,12.02])

#y values = V_d(V) for green LED
y_green = np.array([1.92,1.96,2.00,2.01,2.05])

# Uncertainty on the measured voltages for each data point
y_unc = 0.02*np.ones(5)

# Uncertainty on the measured current
x_unc = 0.02*np.ones(5)

#Plotting the green data
plt.errorbar(x_green,y_green,yerr=y_unc,fmt="ko",label="green data")
guess_slope = 0.01
guess_int = 0.0
par1, cov1 = optimize.curve_fit(line_func,x_green,y_green,
p0=[guess_slope,guess_int],sigma=y_unc,absolute_sigma=True)

# Uncertainty in the y-intercept for green LED
g_int_var = np.diag(cov1)
g_int_unc = g_int_var[1]**0.5

#Fitted values of a and b
fit_a = par1[0]
fit_b = par1[1]

#Plotting the best fit line
xf = np.linspace(0.50,12.0,100)
yf = line_func(xf,fit_a,fit_b)
```

```python
plt.plot(xf, yf,"g--",label="line fit")
plt.xlabel("I (mA)")
plt.ylabel("V_d (V)")
plt.legend()

print("best fit value of a:  ", fit_a)
print("best fit value of b:  ", fit_b)
print("Uncertainty on measured voltages:",y_unc[0],"V")
print("Uncertainty on measured currents:",x_unc[0],"mA")
print("Uncertainty on y-int:",g_int_unc)

plt.show()
print("""Jupyter Notebook 12.1
Plot of voltage versus current data for green LED only considering measured
values for I > 2.0mA""")
```

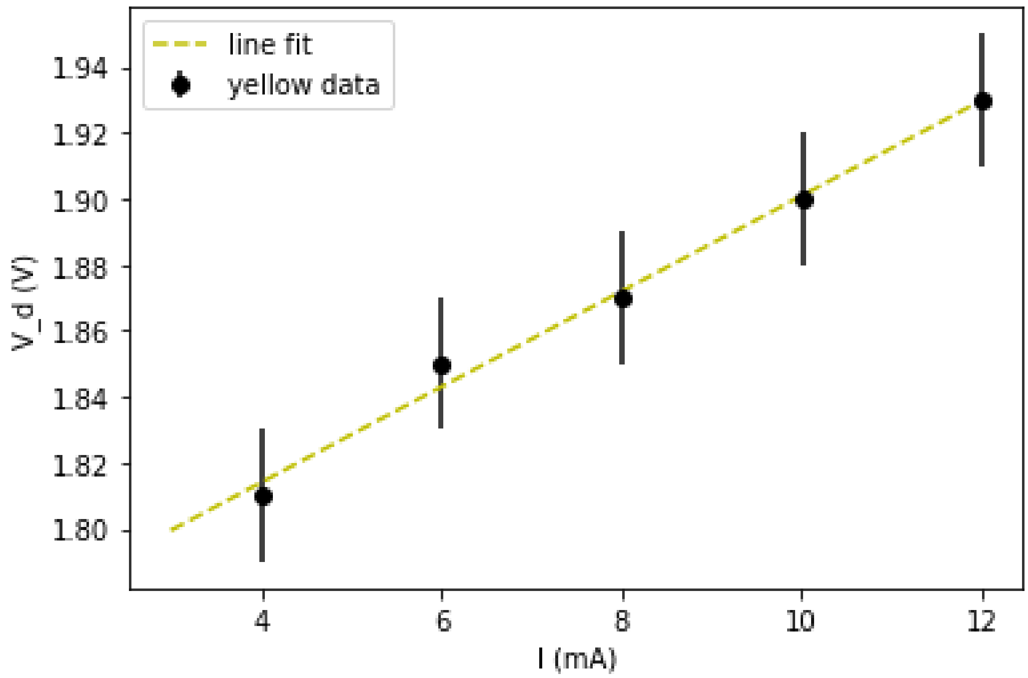
```
best fit value of a:   0.015446641927042558
best fit value of b:   1.8644268646028994
Uncertainty on measured voltages: 0.02 V
Uncertainty on measured currents: 0.02 mA
Uncertainty on y-int: 0.02674960921469231
```




```
Jupyter Notebook 12.1
Plot of voltage versus current data for green LED only considering measured
values for I > 2.0mA
```

[109]:
```python
#x values = I(mA) for yellow LED
x_yellow = np.array([4.00,6.00,8.01,10.01,12.00])

# y values = V_d(V) for yellow LED
y_yellow = np.array([1.81,1.85,1.87,1.90,1.93])

#Plotting the yellow data
plt.errorbar(x_yellow, y_yellow,yerr=y_unc,fmt="ko",label="yellow data")
guess_slope = 24.0
guess_int = 0.0
par2, cov2 = optimize.curve_fit(line_func, x_yellow, y_yellow,
 p0=[guess_slope,guess_int],sigma=y_unc,absolute_sigma=True)

# Uncertainty in y-intercept for yellow LED
y_int_var = np.diag(cov2)
y_int_unc = y_int_var[1]**0.5

#Fitted values of a and b
fit_a = par2[0]
fit_b = par2[1]

#Plotting the best fit line
xf = np.linspace(3.0,12.0,100)
yf = line_func(xf,fit_a,fit_b)
plt.plot(xf, yf,"y--",label="line fit")
plt.xlabel("I (mA)")
plt.ylabel("V_d (V)")
plt.legend()

print("best fit value of a:  ", fit_a)
print("best fit value of b:  ", fit_b)
print("Uncertainty on measured voltages:",y_unc[0],"V")
print("Uncertainty on measured currents:",x_unc[0],"mA")
print("Uncertainty on y-int:",y_int_unc)

plt.show()
print("""Jupyter Notebook 12.2
Plot of voltage versus current data for yellow LED only considering measured
values for I > 2.0mA""")
```

```
best fit value of a:   0.014491964507229227
best fit value of b:   1.756006315669177
Uncertainty on measured voltages: 0.02 V
Uncertainty on measured currents: 0.02 mA
Uncertainty on y-int: 0.026832784196086513
```

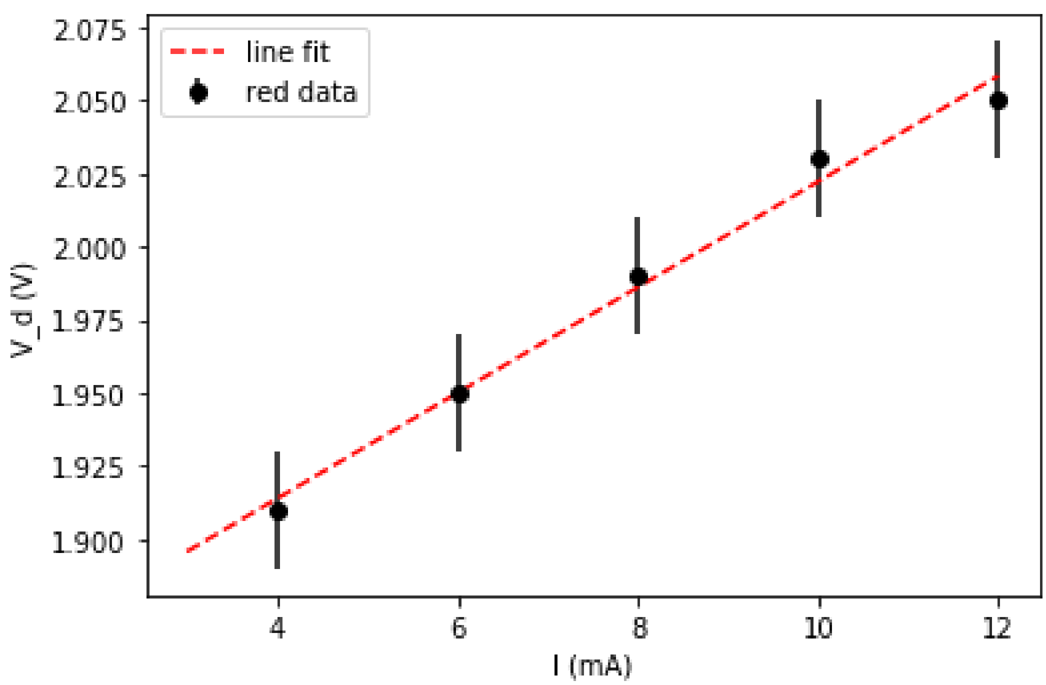


Jupyter Notebook 12.2
Plot of voltage versus current data for yellow LED only considering measured values for I > 2.0mA

[110]:
```
#x values = I(mA) for red LED
x_red = np.array([4.00,6.00,8.00,10.00,11.98])

# y values = V_d(V) for red LED
y_red = np.array([1.91,1.95,1.99,2.03,2.05])

#Plotting the red data
plt.errorbar(x_red, y_red,yerr=y_unc,fmt="ko",label="red data")
guess_slope = 24.0
guess_int = 0.0
par3, cov3 = optimize.curve_fit(line_func,x_red,y_red,p0=[guess_slope,guess_int],
sigma=y_unc,absolute_sigma=True)

# Uncertainty in y-intercept for red LED
r_int_var = np.diag(cov3)
r_int_unc = r_int_var[1]**0.5

#Fitted values of a and b
fit_a = par3[0]
fit_b = par3[1]
```

```python
#Plotting the best fit line
xf = np.linspace(3.0,12.0,100)
yf = line_func(xf,fit_a,fit_b)
plt.plot(xf, yf,"r--",label="line fit")
plt.xlabel("I (mA)")
plt.ylabel("V_d (V)")
plt.legend()

print("best fit value of a:  ", fit_a)
print("best fit value of b:  ", fit_b)
print("Uncertainty on measured voltages:",y_unc[0],"V")
print("Uncertainty on measured currents:",x_unc[0],"mA")
print("Uncertainty on y-int:",r_int_unc)

plt.show()
print("""Jupyter Notebook 12.3
Plot of voltage versus current data for red LED only considering measured
values for I > 2.0mA""")
```

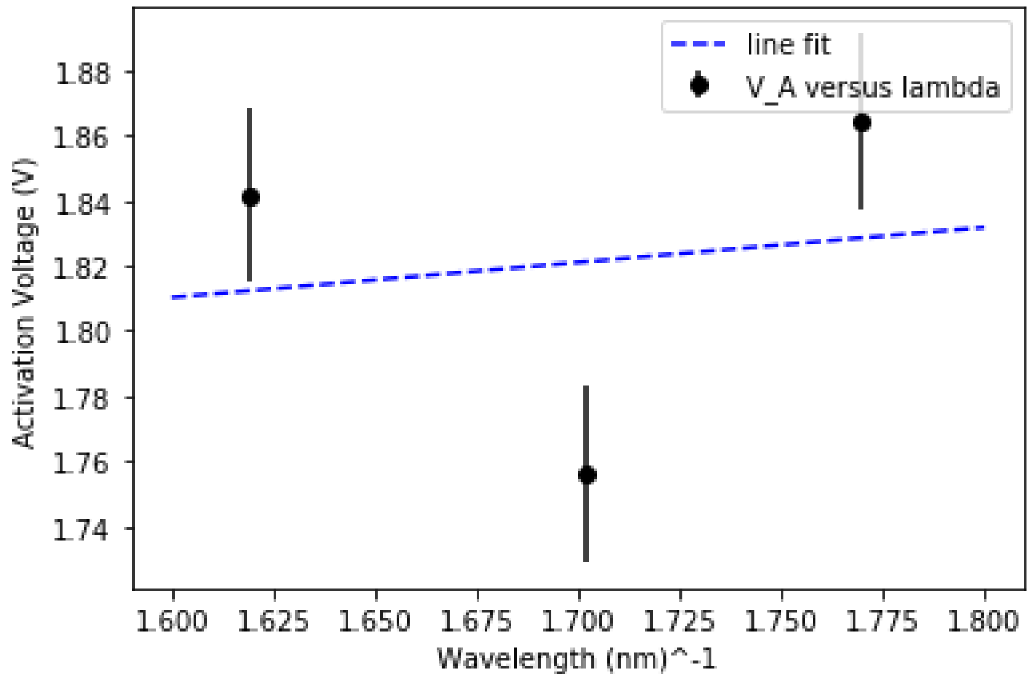
```
best fit value of a:   0.018040015896837472
best fit value of b:   1.8417520329143235
Uncertainty on measured voltages: 0.02 V
Uncertainty on measured currents: 0.02 mA
Uncertainty on y-int: 0.026868621024181166
```



Jupyter Notebook 12.3
Plot of voltage versus current data for red LED only considering measured values for I > 2.0mA

[111]:
```python
# Best fit values of V_A determined from previous plots:
active_voltage = np.array([par1[1],par2[1],par3[1]])

# 1/wavelength (nm)^-1
wavelength = 1000*np.array([1/565,1/587.5,1/617.5])

# Uncertainty
V_a_unc = np.array([g_int_unc,y_int_unc,r_int_unc])

# Plotting the error bars
plt.errorbar(wavelength,active_voltage,yerr=V_a_unc,fmt="ko",label="V_A versus␣
↪lambda")
plt.xlabel("Wavelength (nm)^-1")
plt.ylabel("Activation Voltage (V)")

# Parameters
guess_a = 1.0
guess_b = 0.0
par4, cov4 = optimize.curve_fit(line_func,wavelength,
active_voltage,p0=[guess_a,guess_b])

# Plotting the best fit lline
xf = np.linspace(1.6,1.8,100)
yf = line_func(xf,par4[0],par4[1])
plt.plot(xf, yf,"b--",label="line fit")
plt.legend()

# Theoretical value in eVμm
hc = 1.23984193

# % Error
err = (hc-par4[0])/hc*100

print("Theoretical slope hc:",hc)
print("Measured slope:",par4[0])
plt.show()
print("""Jupyter Notebook 12.4
Plot of activation voltage of each diode versus the 1/wavelength""")
print("Calculated error",err,"%")
```

```
Theoretical slope hc: 1.23984193
Measured slope: 0.10771373789681749
```




```
Jupyter Notebook 12.4
Plot of activation voltage of each diode versus the 1/wavelength
Calculated error 91.31230076266112 %
```

[43]:

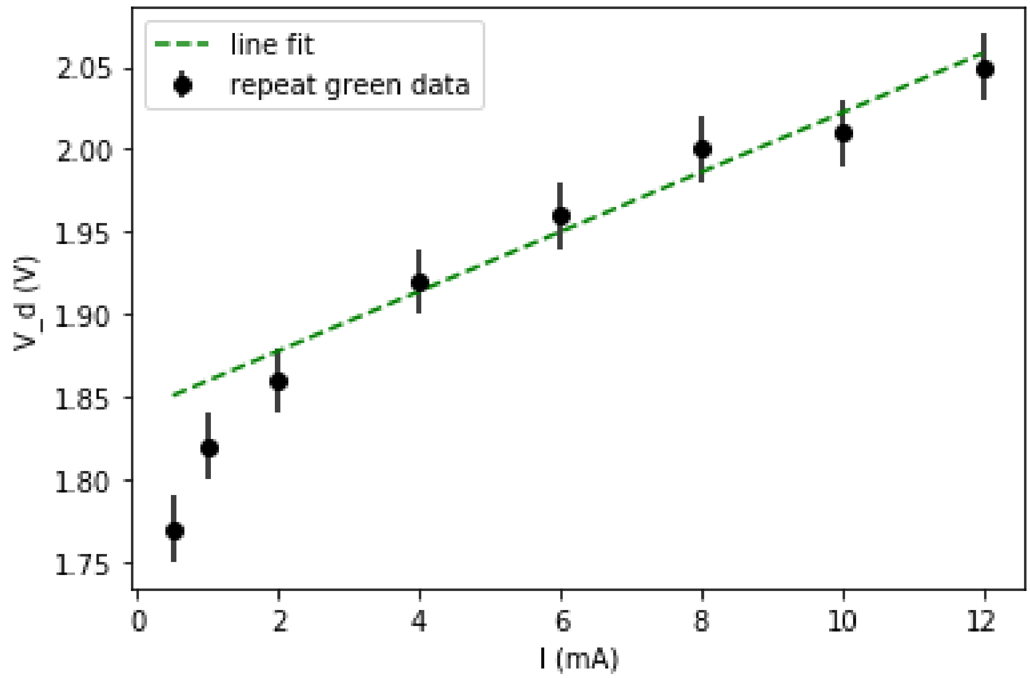
```
# Jupyter Notebook 12.5 - "Systematic Uncertainties" removing the requirement
# of I > 2.0 mA and repeating analysis for green, yellow and red LED's
```

[112]:

```
# Repeat analysis for green LED

#x values = I(mA) for green LED
g_x_repeat = np.array([0.5,1.01,1.99,3.99,5.99,8.00,10.00,12.02])

#y values = V_d(V) for green LED
g_y_repeat = np.array([1.77,1.82,1.86,1.92,1.96,2.00,2.01,2.05])

# Uncertainty on the measured current
x_unc_repeat = 0.02*np.ones(8)

# Uncertainty on the measured voltages for each data point
y_unc_repeat = 0.02*np.ones(8)

#Plotting the green data repeat
plt.errorbar(g_x_repeat,g_y_repeat,yerr=y_unc_repeat,fmt="ko",label="repeat␣
 ↪green data")
guess_slope = 0.01
```

```
guess_int = 0.0
par1_repeat,cov1_repeat=optimize.curve_fit(line_func,g_x_repeat,
g_y_repeat,p0=[guess_slope,guess_int],sigma=y_unc_repeat,absolute_sigma=True)

#Fitted values of a and b
fit_a_rep1 = par1_repeat[0]
fit_b_rep1 = par1_repeat[1]

# Uncertainty in the y-intercept for green LED
g_int_var_repeat = np.diag(cov1_repeat)
g_int_unc_repeat = g_int_var_repeat[1]**0.5

#Plotting the best fit line
xf = np.linspace(0.50,12.0,100)
yf = line_func(xf,fit_a,fit_b)
plt.plot(xf, yf,"g--",label="line fit")
plt.xlabel("I (mA)")
plt.ylabel("V_d (V)")
plt.legend()


print("best fit value of a:  ", fit_a_rep1)
print("best fit value of b:  ", fit_b_rep1)
print("Uncertainty on measured voltages:",y_unc[0],"V")
print("Uncertainty on measured currents:",x_unc[0],"mA")
print("Uncertainty in y-int:",g_int_unc_repeat)

plt.show()
print("""Jupyter Notebook 12.5 (green)
Repeat plot of voltage versus current data for green LED without restriction on␣
 ↪y-values
values for I > 2.0mA""")
```

```
best fit value of a:   0.022357613932060198
best fit value of b:   1.8021804742444227
Uncertainty on measured voltages: 0.02 V
Uncertainty on measured currents: 0.02 mA
Uncertainty in y-int: 0.01190350042072758
```

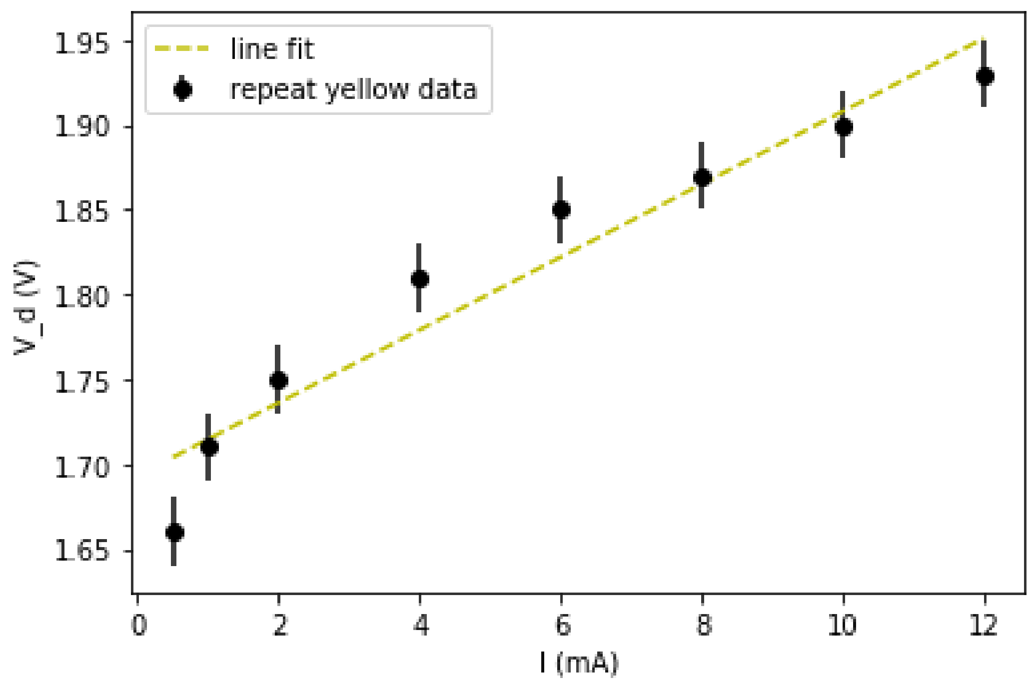


Jupyter Notebook 12.5 (green)
Repeat plot of voltage versus current data for green LED without restriction on y-values
values for I > 2.0mA

[107]:
```
# Repeat analysis for yellow LED

#x values = I(mA) for yellow LED
ye_x_repeat = np.array([0.5,1.02,2.00,4.00,6.00,8.01,10.01,12.00])

# y values = V_d(V) for yellow LED
ye_y_repeat = np.array([1.66,1.71,1.75,1.81,1.85,1.87,1.90,1.93])

#Plotting the yellow data
plt.errorbar(ye_x_repeat,ye_y_repeat,yerr=y_unc_repeat,fmt="ko",label="repeat␣
 ↪yellow data")
guess_slope = 0.01
guess_int = 0.0
par2_repeat,cov2_repeat=optimize.curve_fit(line_func,ye_x_repeat,
ye_y_repeat,p0=[guess_slope,guess_int],sigma=y_unc_repeat,absolute_sigma=True)

# Uncertainty in y-intercept for yellow LED
y_int_var_repeat = np.diag(cov2_repeat)
y_int_unc_repeat = y_int_var_repeat[1]**0.5
```

```
#Fitted values of a and b
fit_a_rep2 = par2_repeat[0]
fit_b_rep2 = par2_repeat[1]

#Plotting the best fit line
xf = np.linspace(0.50,12.0,100)
yf = line_func(xf,fit_a_rep2,fit_b_rep2)
plt.plot(xf, yf,"y--",label="line fit")
plt.xlabel("I (mA)")
plt.ylabel("V_d (V)")
plt.legend()

print("best fit value of a:  ", fit_a_rep2)
print("best fit value of b:  ", fit_b_rep2)
print("Uncertainty on measured voltages:",y_unc[0],"V")
print("Uncertainty on measured currents:",x_unc[0],"mA")
print("Uncertainty in y-int:",y_int_unc_repeat)

plt.show()
print("""Jupyter Notebook 12.5 (yellow)
Repeat plot of voltage versus current data for yellow LED without restriction on␣
 ↪y-values
values for I > 2.0mA""")
```

```
best fit value of a:   0.02148283228182466
best fit value of b:   1.6930796852778771
Uncertainty on measured voltages: 0.02 V
Uncertainty on measured currents: 0.02 mA
Uncertainty in y-int: 0.011919433457100614
```

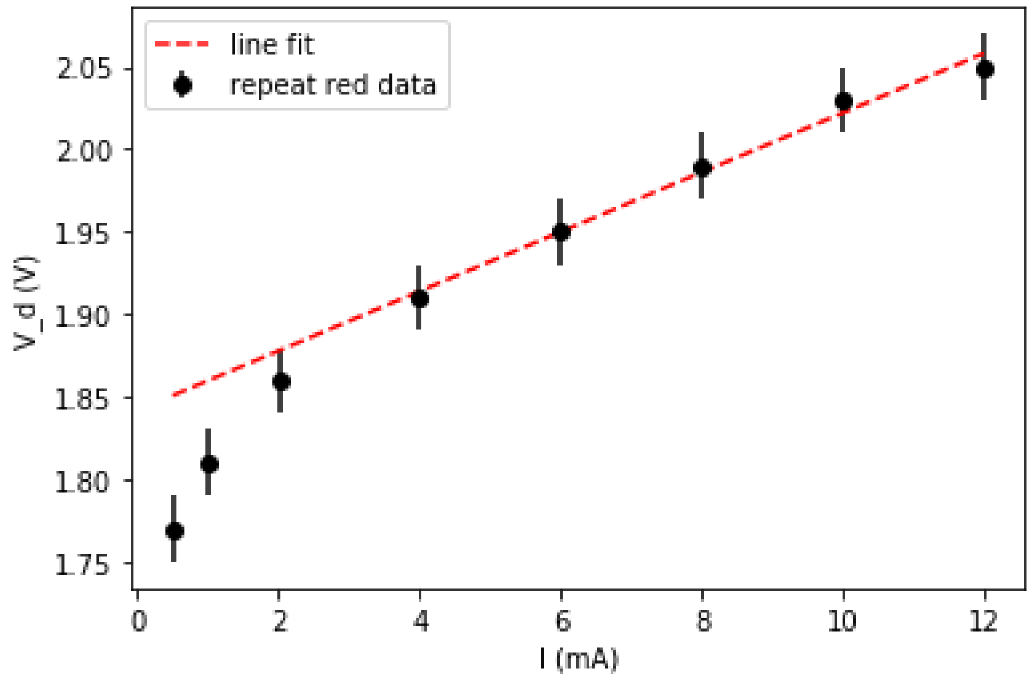


Jupyter Notebook 12.5 (yellow)
Repeat plot of voltage versus current data for yellow LED without restriction on y-values
values for I > 2.0mA

[113]:
```
# Repeat analysis for red LED

#x values = I(mA) for red LED
r_x_repeat = np.array([0.5,0.99,2.01,4.00,6.00,8.00,10.00,11.98])

# y values = V_d(V) for red LED
r_y_repeat = np.array([1.77,1.81,1.86,1.91,1.95,1.99,2.03,2.05])

#Plotting the red data
plt.errorbar(r_x_repeat,r_y_repeat,yerr=y_unc_repeat,fmt="ko",label="repeat red␣
 ↪data")
guess_slope = 0.01
guess_int = 0.0
par3_repeat,cov3_repeat=optimize.curve_fit(line_func,r_x_repeat,r_y_repeat,
p0=[guess_slope,guess_int],sigma=y_unc_repeat,absolute_sigma=True)

# Uncertainty in y-intercept for red LED
r_int_var_repeat = np.diag(cov3_repeat)
r_int_unc_repeat = r_int_var_repeat[1]**0.5
```

```python
#Fitted values of a and b
fit_a_rep3 = par3_repeat[0]
fit_b_rep3 = par3_repeat[1]

#Plotting the best fit line
xf = np.linspace(0.50,12.0,100)
yf = line_func(xf,fit_a,fit_b)
plt.plot(xf, yf,"r--",label="line fit")
plt.xlabel("I (mA)")
plt.ylabel("V_d (V)")
plt.legend()

print("best fit value of a:  ", fit_a_rep3)
print("best fit value of b:  ", fit_b_rep3)
print("Uncertainty on measured voltages:",y_unc[0],"V")
print("Uncertainty on measured currents:",x_unc[0],"mA")
print("Uncertainty in y-int:",r_int_unc_repeat)

plt.show()
print("""Jupyter Notebook 12.5 (red)
Repeat plot of voltage versus current data for red LED without restriction on␣
↪y-values
values for I > 2.0mA""")
```

```
best fit value of a:   0.023339233112374593
best fit value of b:   1.7944012680070354
Uncertainty on measured voltages: 0.02 V
Uncertainty on measured currents: 0.02 mA
Uncertainty in y-int: 0.011915011120724841
```

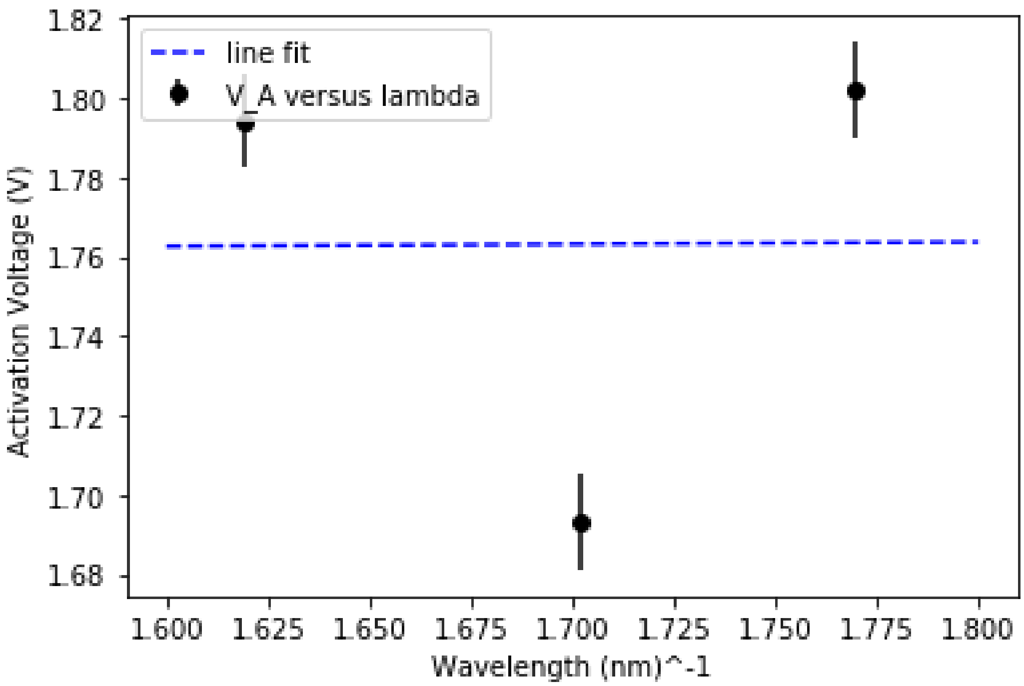



```
Jupyter Notebook 12.5 (red)
Repeat plot of voltage versus current data for red LED without restriction on
y-values
values for I > 2.0mA
```

[114]:
```python
# Best fit values of V_A determined from previous plots:
active_voltage = np.array([par1_repeat[1],par2_repeat[1],par3_repeat[1]])

# 1/wavelength
wavelength = 1000*np.array([1/565,1/587.5,1/617.5])

# Uncertainty
V_a_unc_repeat = np.array([g_int_unc_repeat,y_int_unc_repeat,r_int_unc_repeat])

# Plotting the error bars
plt.errorbar(wavelength,active_voltage,yerr=V_a_unc_repeat,fmt="ko",label="V_A␣
 ↪versus lambda")
plt.xlabel("Wavelength (nm)^-1")
plt.ylabel("Activation Voltage (V)")

# Parameters
guess_a = 1.0
guess_b = 0.0
par4_repeat,cov4_repeat=optimize.curve_fit(line_func,wavelength,
active_voltage,p0=[guess_a,guess_b])
```

```python
# Plotting the best fit lline
xf = np.linspace(1.6,1.8,100)
yf = line_func(xf,par4_repeat[0],par4_repeat[1])
plt.plot(xf, yf,"b--",label="line fit")
plt.legend()

# Theoretical value in eVμm
hc = 1.23984193

# % Error
err_repeat = (hc-par4_repeat[0])/hc*100

print("Theoretical slope hc:",hc)
print("Measured slope:",par4_repeat[0])
plt.show()
print("""Jupyter Notebook 12.5
Repeat plot of activation voltage of each diode versus the 1/wavelength without␣
↪restriction of I > 2.0 mA""")
print("Calculated error",err_repeat,"%","Remark: higher percentage error without␣
↪restriction")
```

```
Theoretical slope hc: 1.23984193
Measured slope: 0.005491612479598285
```



Jupyter Notebook 12.5
Repeat plot of activation voltage of each diode versus the 1/wavelength without restriction of I > 2.0 mA
Calculated error 99.55707156317916 % Remark: higher percentage error without restriction

```
[ ]:
```