KHIK Data Correction

November 2, 2018

In [43]:
```
#import!
import pandas
import numpy
import scipy.constants
from scipy import optimize
from bokeh.plotting import figure, output_notebook, show

output_notebook(hide_banner=True)
```

In [2]:
```
#pull and define data

raw = pandas.read_csv('Data Only.csv')

V = raw.iloc[:,0].values
I_365 = raw.iloc[:,1].values
I_405 = raw.iloc[:,2].values
I_436 = raw.iloc[:,3].values
I_486 = raw.iloc[:,4].values
I_546 = raw.iloc[:,5].values
I_577 = raw.iloc[:,6].values
I_589 = raw.iloc[:,7].values
I_656 = raw.iloc[:,8].values
```

In [3]:
```
#plot uncorrected data

p=figure(title="Uncorrected Data", x_axis_label='Voltage (V)', y_axis_label='Current (

p.circle(V, I_365, fill_color="white", size=4)
p.triangle(V, I_405, fill_color="blue", size=4)
p.square(V, I_436, fill_color="green", size=4)
p.square_cross(V, I_486, fill_color="orange", size=4)
p.diamond(V, I_546, fill_color="yellow", size=4)
p.circle_cross(V, I_577, fill_color="black", size=4)
p.diamond_cross(V, I_589, fill_color="red", size=4)
p.inverted_triangle(V, I_656, fill_color="purple", size=4)

show(p)
```

In [7]:
```
#Here I create some empty arrays so that I can define them later.
#We are skipping 656 because the mercury lamp does not give off that wavelength

f_365 = numpy.zeros(len(V))
f_405 = numpy.zeros(len(V))
f_436 = numpy.zeros(len(V))
f_486 = numpy.zeros(len(V))
f_546 = numpy.zeros(len(V))
f_577 = numpy.zeros(len(V))
f_589 = numpy.zeros(len(V))

In_365 = numpy.zeros(len(V))
In_405 = numpy.zeros(len(V))
In_436 = numpy.zeros(len(V))
In_486 = numpy.zeros(len(V))
In_546 = numpy.zeros(len(V))
In_577 = numpy.zeros(len(V))
In_589 = numpy.zeros(len(V))
```

In [11]:
```
#finding f, the ratio of current at +V and -V for -V<V_c

i = 0
#V[i] = 0 at i = 400
while i<len(V):
    if 800-i < len(V):
        if V[i] < -1.605:
            f_365[i] = I_365[i]/I_365[800-i]
        if V[i] < -1.25:
            f_405[i] = I_405[i]/I_405[800-i]
        if V[i] < -1.05:
            f_436[i] = I_436[i]/I_436[800-i]
        if V[i] < -0.785:
            f_486[i] = I_486[i]/I_486[800-i]
        if V[i] < -0.525:
            f_546[i] = I_546[i]/I_546[800-i]
        if V[i] < -0.415:
            f_577[i] = I_577[i]/I_577[800-i]
        if V[i] < -0.375 :
            f_589[i] = I_589[i]/I_589[800-i]

    i += 1
```

In [12]:
```
#plot the coefficients to see if they've been calculated correctly.

p=figure(title="Correction Coefficients", x_axis_label='Voltage (V)', y_axis_label='Cu

p.circle(V, f_365, fill_color="white", size=4)
p.triangle(V, f_405, fill_color="blue", size=4)
```

```
p.square(V, f_436, fill_color="green", size=4)
p.square_cross(V, f_486, fill_color="orange", size=4)
p.diamond(V, f_546, fill_color="yellow", size=4)
p.circle_cross(V, f_577, fill_color="black", size=4)
p.diamond_cross(V, f_589, fill_color="red", size=4)

show(p)
```

In [13]:
```
#In is the new current data after adjusting for back-flow current.

i = 0

while i<len(V):
    if 800-i < len(V):
        In_365[i] = I_365[i]-f_365[i]*I_365[800-i]
        In_405[i] = I_405[i]-f_405[i]*I_405[800-i]
        In_436[i] = I_436[i]-f_436[i]*I_436[800-i]
        In_486[i] = I_486[i]-f_486[i]*I_486[800-i]
        In_546[i] = I_546[i]-f_546[i]*I_546[800-i]
        In_577[i] = I_577[i]-f_577[i]*I_577[800-i]
        In_589[i] = I_589[i]-f_589[i]*I_589[800-i]
    else:
        In_365[i] = I_365[i]
        In_405[i] = I_405[i]
        In_436[i] = I_436[i]
        In_486[i] = I_486[i]
        In_546[i] = I_546[i]
        In_577[i] = I_577[i]
        In_589[i] = I_589[i]
    i += 1
```

In [54]:
```
#Plot the corrected data!

p=figure(title="Corrected Data", x_axis_label='Voltage (V)', y_axis_label='Current (A

p.circle(V, In_365, fill_color="white", size=4)
p.triangle(V, In_405, fill_color="blue", size=4)
p.square(V, In_436, fill_color="green", size=4)
p.square_cross(V, In_486, fill_color="orange", size=4)
p.diamond(V, In_546, fill_color="yellow", size=4)
p.circle_cross(V, In_577, fill_color="black", size=4)
p.diamond_cross(V, In_589, fill_color="red", size=4)

show(p)
#At this point I moved to MatLab to do the curve fitting.
```