Franck-Hertz Analysis

February 21, 2019

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

        output_notebook(hide_banner=True)
```

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

        raw = pandas.read_csv('Data\Data (Week 2).csv')

        V = raw.iloc[:,0].values
        I_433 = raw.iloc[:,1].values
        I_447 = raw.iloc[:,2].values
        I_462 = raw.iloc[:,3].values
        I_476 = raw.iloc[:,4].values
        E = raw.iloc[:,5].values
```

```
In [ ]: #flip current data to conventional current

        CI_433 = numpy.zeros(len(V));
        CI_447 = numpy.zeros(len(V));
        CI_462 = numpy.zeros(len(V));
        CI_476 = numpy.zeros(len(V));

        i=0;
        while i<len(V):
            CI_433[i] = -I_433[i];
            CI_447[i] = -I_447[i];
            CI_462[i] = -I_462[i];
            CI_476[i] = -I_476[i];
            i+=1;
```

```
In [ ]: #plot!
```

```
p=figure(title="Current vs Energy", x_axis_label='Energy (eV)', y_axis_label='Current

p.circle(V, CI_433, line_color="white", size=4, legend='433K')
#p.triangle(V, CI_447, line_color="blue", fill_color="blue", size=3, legend='447K')
#p.square(V, CI_462, line_color="green", fill_color="green", size=1, legend='462K')
p.square_cross(V, CI_476, line_color="orange", fill_color="orange", size=1, legend='47

p.legend.location='top_left'
show(p)
```

```
In [ ]: #set up finding the slope for minima vs n

        n3 = [3,4,5,6,7,8]
        n36 = [3,4,5,6]
        n5 = [5,6,7,8]

        M433 = [14.10,19.14,24.19,29.31]
        M447 = [14.01,19.03,24.08,29.08,34.13,39.13]
        M462 = [14.11,19.13,24.03,28.94,33.92,38.87]
        M476 = [24.26,29.33,34.19,39.01]
```

```
In [ ]: #plot!

        p=figure(title="Energy vs Number of Collisions", x_axis_label='Number of Collisions', y

        p.circle(n36, M433, line_color="white", size=5, legend='433K')
        p.triangle(n3, M447, line_color="blue", fill_color="blue", size=5, legend='447K')
        p.square(n3, M462, line_color="green", fill_color="green", size=5, legend='462K')
        p.square_cross(n5, M476, line_color="orange", fill_color="orange", size=5, legend='476K

        p.legend.location='top_left'
        show(p)
```

```
In [ ]: #find curve fit

        f433=numpy.polyfit(n36,M433,1)
        f447=numpy.polyfit(n3,M447,1)
        f462=numpy.polyfit(n3,M462,1)
        f476=numpy.polyfit(n5,M476,1)

        #print curve fit

        print(f433)
        print(f447)
        print(f462)
        print(f476)

        #set up arrays
```

```
x=[0,1,2,3,4,5,6,7,8,9];
y433=numpy.zeros(len(x));
y447=numpy.zeros(len(x));
y462=numpy.zeros(len(x));
y476=numpy.zeros(len(x));
```

In [ ]:
```
#define arrays
i=0;
while i<len(x):
    y433[i]=f433[0]*x[i]+f433[1];
    y447[i]=f447[0]*x[i]+f447[1];
    y462[i]=f462[0]*x[i]+f462[1];
    y476[i]=f476[0]*x[i]+f476[1];
    i+=1;
```

In [ ]:
```
#now plot with the lines!
#Make sure to change the temperature in the title, data, and fit, and make sure the co

p=figure(title="Energy vs Number of Collisions: Curve Fit (476K)", x_axis_label='Number

p.circle(n5, M476, line_color="white", size=5)
p.line(x,y476, line_color="blue", line_width=1)

show(p)
```

In [ ]:
```
Ea_433=f433[0]/2+f433[1]
Ea_447=f447[0]/2+f447[1]
Ea_462=f462[0]/2+f462[1]
Ea_476=f476[0]/2+f476[1]

print(Ea_433)
print(Ea_447)
print(Ea_462)
print(Ea_476)
```

In [ ]:
```
#This seems incorrect - maybe use DeltaE instead of E?

n4 = [4,5,6,7,8]
n46 = [4,5,6]
n6 = [6,7,8]

DM433 = [5.04,5.05,5.14]
DM447 = [5.01,5.05,5,5.05,5]
DM462 = [5.02,4.9,4.91,4.96,4.95]
DM476 = [5.07,4.86,5.1]

Df433=numpy.polyfit(n46,DM433,1)
Df447=numpy.polyfit(n4,DM447,1)
Df462=numpy.polyfit(n4,DM462,1)
```

```
Df476=numpy.polyfit(n6,DM476,1)

Dy433=numpy.zeros(len(x));
Dy447=numpy.zeros(len(x));
Dy462=numpy.zeros(len(x));
Dy476=numpy.zeros(len(x));
```

In [ ]:
```
#define arrays
i=0;
while i<len(x):
    Dy433[i]=Df433[0]*x[i]+Df433[1];
    Dy447[i]=Df447[0]*x[i]+Df447[1];
    Dy462[i]=Df462[0]*x[i]+Df462[1];
    Dy476[i]=Df476[0]*x[i]+Df476[1];
    i+=1;
```

In [ ]:
```
#now plot with the lines!
#Make sure to change the temperature in the title, data, and fit, and make sure the co

p=figure(title="Change in Energy vs Number of Collisions: Curve Fit (476K)", x_axis_lal

p.circle(n6, DM476, line_color="white", size=5)
p.line(x,Dy476, line_color="blue", line_width=1)

show(p)
```

In [ ]:
```
DEa_433=Df433[0]/2+Df433[1]
DEa_447=Df447[0]/2+Df447[1]
DEa_462=Df462[0]/2+Df462[1]
DEa_476=Df476[0]/2+Df476[1]

print(DEa_433)
print(DEa_447)
print(DEa_462)
print(DEa_476)
```

In [ ]:
```
#answers in m

lambda_433 = (.008/(2*DEa_433))*Df433[0]
lambda_447 = (.008/(2*DEa_447))*Df447[0]
lambda_462 = (.008/(2*DEa_462))*Df462[0]
lambda_476 = (.008/(2*DEa_476))*Df476[0]

print(lambda_433)
print(lambda_447)
print(lambda_462)
print(lambda_476)
```

In [ ]:
```
#answers in m^-2
```

```
k = 1.38065E-23

sigma_433 = k*433/(8.7*10**(9-3110/433)*lambda_433)
sigma_447 = k*447/(8.7*10**(9-3110/447)*lambda_447)
sigma_462 = k*462/(8.7*10**(9-3110/462)*lambda_462)
sigma_476 = k*476/(8.7*10**(9-3110/476)*lambda_476)

print(sigma_433)
print(sigma_447)
print(sigma_462)
print(sigma_476)
```

In [ ]:
```
#defining uncertainties

uI=10**-14
uV=10**-2
uT=1
uL=10**-3

uE=uV

uLambda_433 = (uL**2*((1/(2*DEa_433))*Df433[0])**2+0.008*uE**2*(((1/(2*DEa_433))*Df433
uLambda_447 = (uL**2*((1/(2*DEa_447))*Df433[0])**2+0.008*uE**2*(((1/(2*DEa_447))*Df447
uLambda_462 = (uL**2*((1/(2*DEa_433))*Df462[0])**2+0.008*uE**2*(((1/(2*DEa_462))*Df462
uLambda_476 = (uL**2*((1/(2*DEa_476))*Df476[0])**2+0.008*uE**2*(((1/(2*DEa_476))*Df433

uS_433 = (uLambda_433**2*(k*433/(lambda_433**2*8.7*10**(9-3110/433)))**2+uT**2*(1.149*
uS_447 = (uLambda_447**2*(k*447/(lambda_447**2*8.7*10**(9-3110/447)))**2+uT**2*(1.149*
uS_462 = (uLambda_462**2*(k*462/(lambda_462**2*8.7*10**(9-3110/462)))**2+uT**2*(1.149*
uS_476 = (uLambda_476**2*(k*476/(lambda_476**2*8.7*10**(9-3110/476)))**2+uT**2*(1.149*
```

In [ ]:
```
print(uLambda_433)
print(uLambda_447)
print(uLambda_462)
print(uLambda_476)

print(uS_433)
print(uS_447)
print(uS_462)
print(uS_476)
```

In [ ]:
```
1/(2*DEa_433)
```

In [ ]: