Boğaziçi University
Physics Department

X-ray Scattering: The Duane-Hunt Displacement Law

Experiment 3

Student:
Sina Aktaş
2012203024

Partner:
Egemen Yüzbaşı

Instructor:
Veysi Erkcan Özcan

Assistant:
Saime Sarıkaya

Contents

1 Abstract

In this experiment, we measured Planck's constant with X-ray spectrometer with molybdenum target. First of all, we found calibration function via comparing theoretical and measured K_α and K_β ray's constructive interfere angles of molybdenum on NaCl Crystal. Then, we found highest energy of the Bremsstrahlung radiation with various analysis. We measured angle θ which will give us λ_{min} via Bragg's law. In the end, we plot qV(Energy) versus c/$\lambda_{min}(\nu max)$ graph and the slope gave us the Planck's constant with respect to Duane-Hunt Law.In addition, all uncertainties were taken into account.

In conclusion we found the Planck's constant as $7.38x10^{-34}Js \pm 0.079x10^{-34}$ and its χ^2 is equal to 92.

2 Introduction

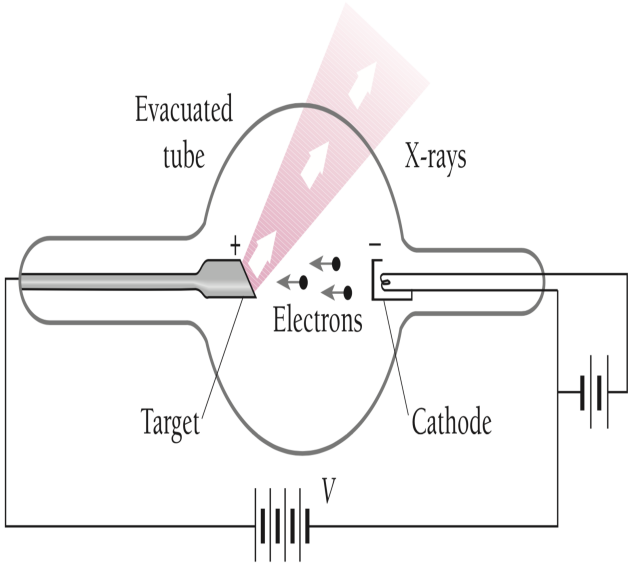
The photoelectric effect provides convincing evidence that photons of light can transfer energy to electrons. Is the inverse process also possible? That is, can part or all of the kinetic energy of a moving electron be converted into a photon? As it happens, the inverse photoelectric effect not only does occur but had been discovered before the work of Planck and Einstein.

2.1 History

In 1895 Wilhelm Roentgen found that a highly penetrating radiation of unknown nature is produced when fast electrons impinge on matter. These x-rays were soon found to travel in straight lines, to be unaffected by electric and magnetic fields, to pass readily through opaque materials, to cause phosphorescent substances to glow, and to expose photographic plates. The faster the original electrons, the more penetrating the resulting x-rays, and the greater the number of electrons, the greater the in- tensity of the x-ray beam.[1]

Max von Laue (1879–1960, Germany) developed the method of X-ray diffraction for the study of crystal structures, for which he received the 1914 Nobel Prize. Lawrence Bragg (1890 – 1971, England) developed the Bragg law for X-ray diffraction while he was a student at Cambridge University. He shared the 1915 Nobel Prize with his father, William Bragg, for their research on the use of X rays to determine crystal structures.[2]

2.2 Theory



,

Figure 1: Simple diagram of x-ray tube[1]

Figure 1 is a diagram of an x-ray tube. A cathode, heated by a filament through which an electric current is passed, supplies electrons by thermionic emission. The high potential difference V maintained between the cathode and a metallic target accelerates the electrons toward the latter. The face of the target is at an angle relative to the electron beam, and the x-rays that leave the target pass through the side of the tube. The tube is evacuated to permit the electrons to get to the target unimpeded. The kinetic energy of the electron is;

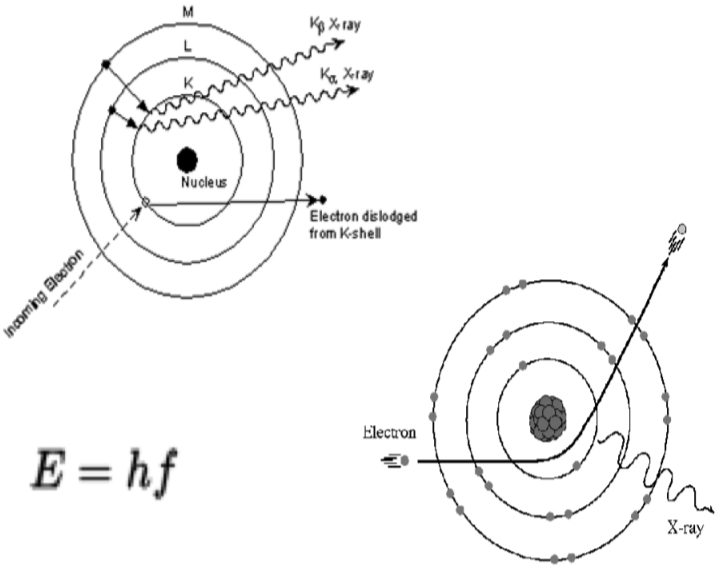
$$KE_{electron} = qV \tag{1}$$

When kinetic energy of electron reach qV, they hit to the anode. Then, some of them make an elastic collision and some of them degenerate electron from n=1 state then produce K_α and K_β ray probabilistically as in the Figure 2. Probability of K_α higher than K_β. Produced x-rays have various energies. And we have following equations known as Duane-Hunt Displacement Law;

$$q_e V = h\nu_{max} \tag{2}$$

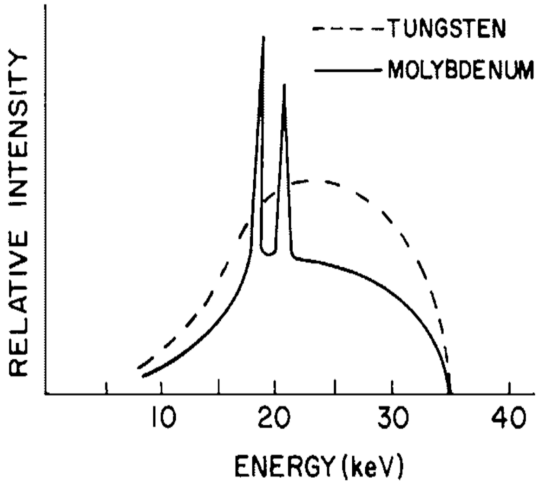
$$\lambda_{min} = \frac{hc}{q_e V} = \frac{1.240x10^{-6}}{V}[Vm] \qquad [1] \tag{3}$$

where h is Planck's constant, c is speed of light, q_e is charge of an electron and, V is voltage between anode and cathode.



,

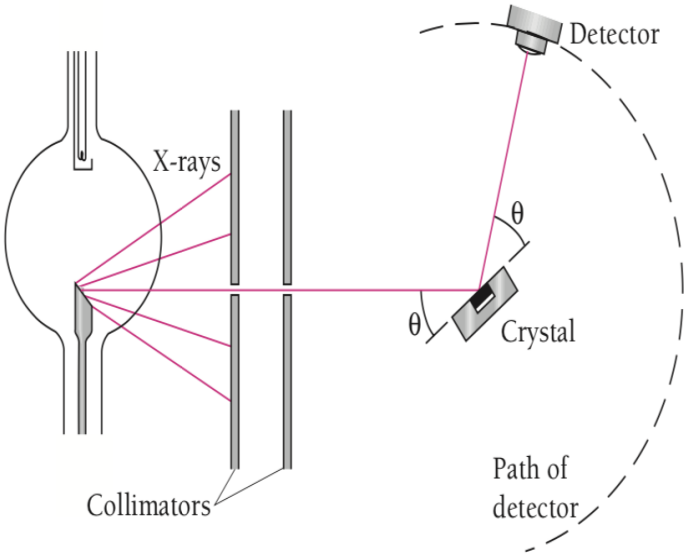
Figure 2: Bremsstrahlung radiation[4]



,

Figure 3: X-ray spectrum from molybdenum (solid line) and tungsten (dashed line) target x-ray tubes[3]

In x-ray production, temperature of anode increase incredibly. Mostly, tungsten and Molybdenum are used because of their high melting point in vacuum.In Figure 3, we see Bremsstrahlung radiation intensity versus energy of photon graph.Each element has characteristic K_α and K_β ray. The reason of why we can't see K_α and K_β peak of tungsten in Figure 3 is not adequate energy. The energy of K_α and K_β of tungsten is 59.3 and 67.2 respectively[4].



,

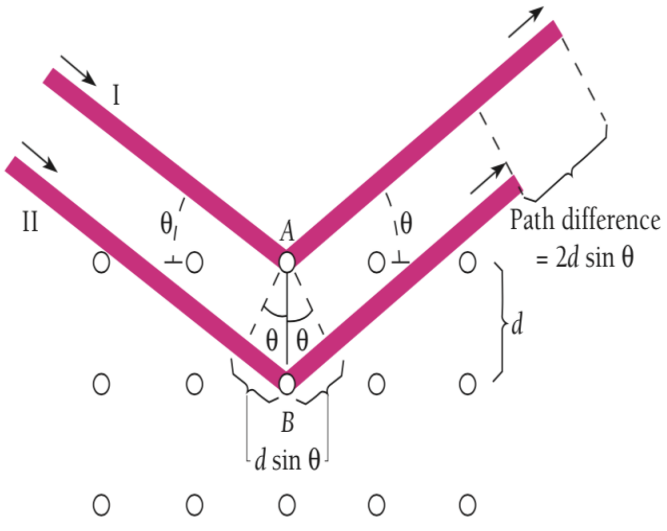
Figure 4: X-ray spectrometer.[1]

To determine x-ray wavelengths, x-ray spectrometer(Figure 4) is used. X-rays going towards the crystal in a straight line. Then they reflected from crystal to the detector. Since crystal act like a many slit, certain directions the scattered waves will constructively interfere with one another while in others they will destructively interfere.

The constructive interference happened when the path difference is equal to $m\lambda$(where m is positive integer). The equation which give us relation between θ and λ is titled Bragg's Law.

$$2dsin(\theta) = n\lambda \tag{4}$$

where d is distance between atoms of crystals.



,

Figure 5: X-ray scattering from a cubic crystal[1]

3 Experiment

3.1 Apparatus

1. Phywe X-Ray Unit
2. X-ray Tube with Molybdenum Target
3. Geiger Counter
4. Computer for Control the Unit
5. NaCl Crystal

3.2 Setup

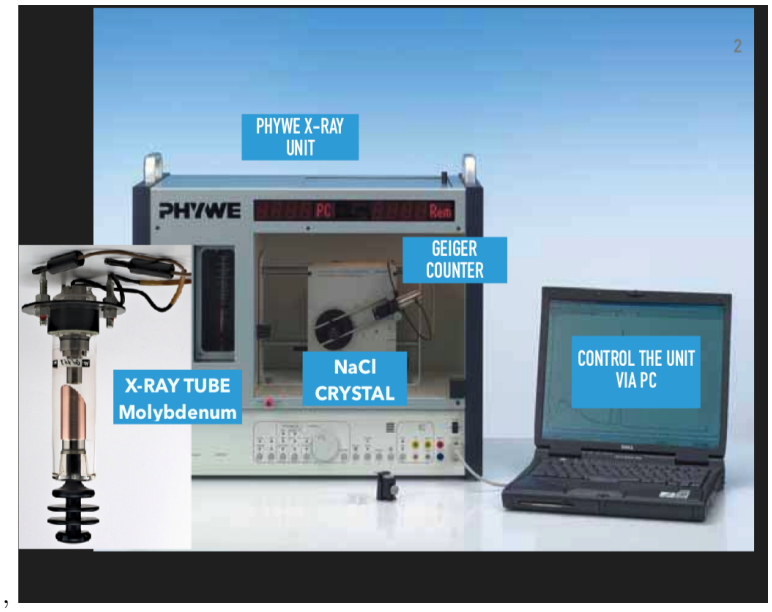


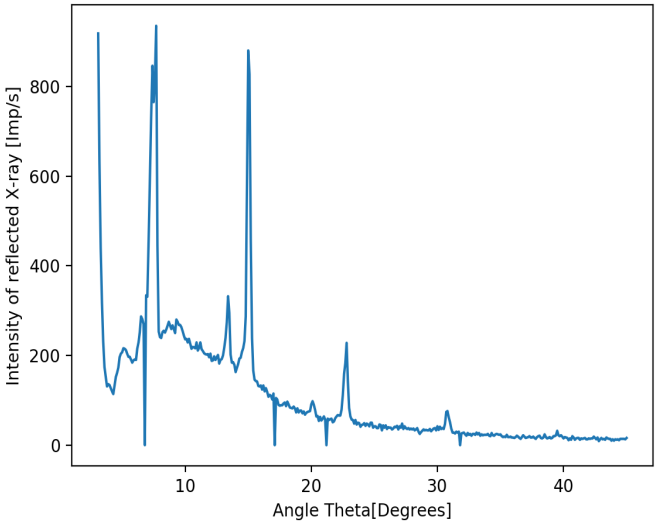
Figure 6: Phywe X-Ray scattering setup[4]

3.3 Procedure

1. Set voltage 35kV to take full spectrum. Measure peak point of K_α and K_β for finding calibration function.
2. Take spectrum of 15kV to 30kV by increasing 3kV separately
3. Make line fit to first increasing data and calculate x-line intersection point for each voltage. This give you θ of λ_{min}.
4. Calibrate this θ's with calibration function and calculate respective ν_{max} with formula 4 and ν=c/λ formula.
5. Plot qV versus ν graph and fit the data. Then slope of the line is equal to Planck's constant.

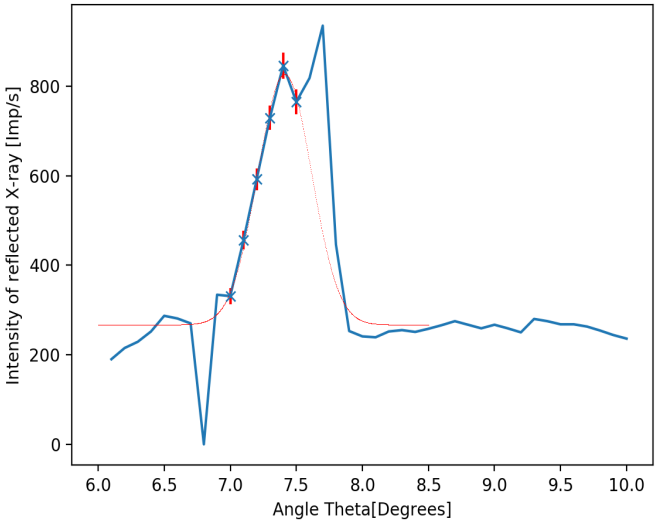
4 Data and Data Analysis

4.1 Calibration Function



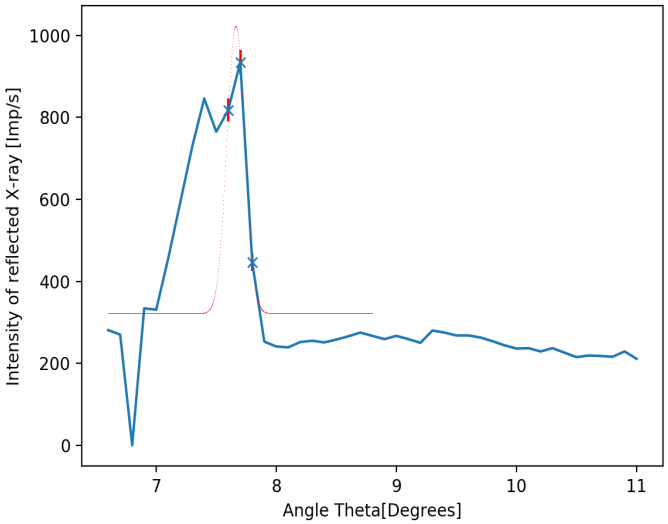
,

Figure 7: Angels vs Intensity of reflected X-ray graph for 35kV for Calibration



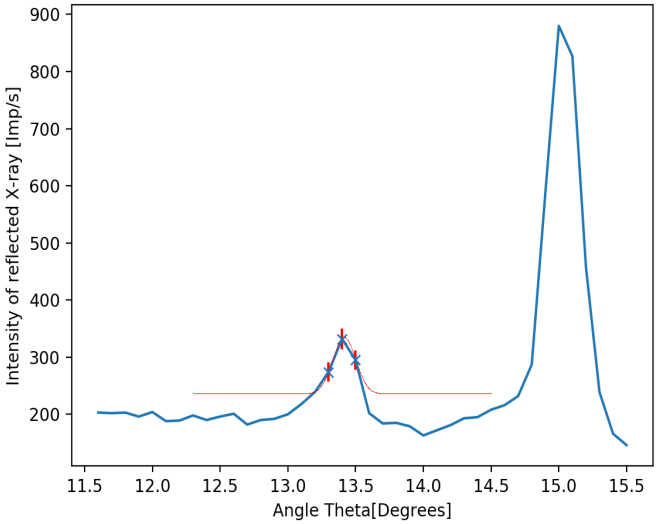
,

Figure 8: Gaussian fit to find more proper $K_{\beta 1}$ ray degree



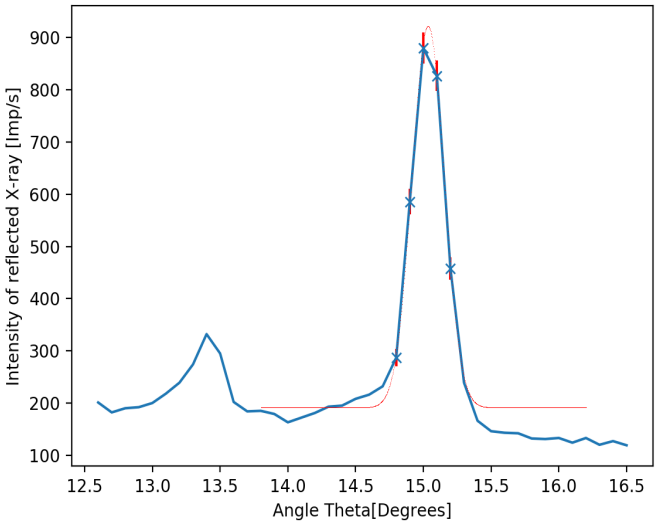
,

Figure 9: Gaussian fit to find more proper $K_{\alpha 1}$ ray degree



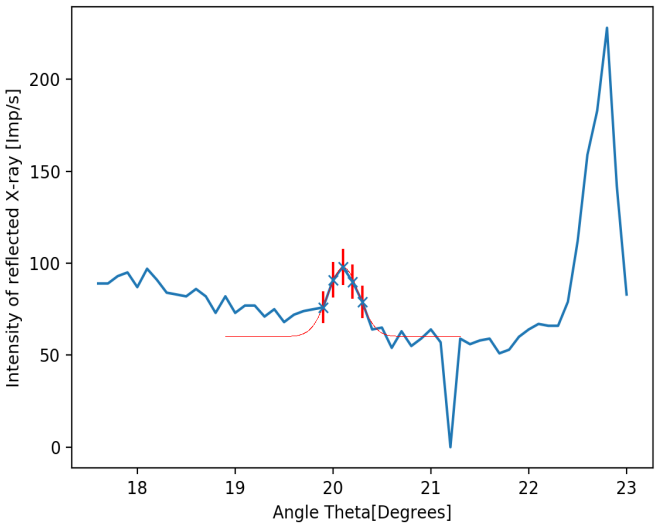
,

Figure 10: Gaussian fit to find more proper $K_{\beta 2}$ ray degree



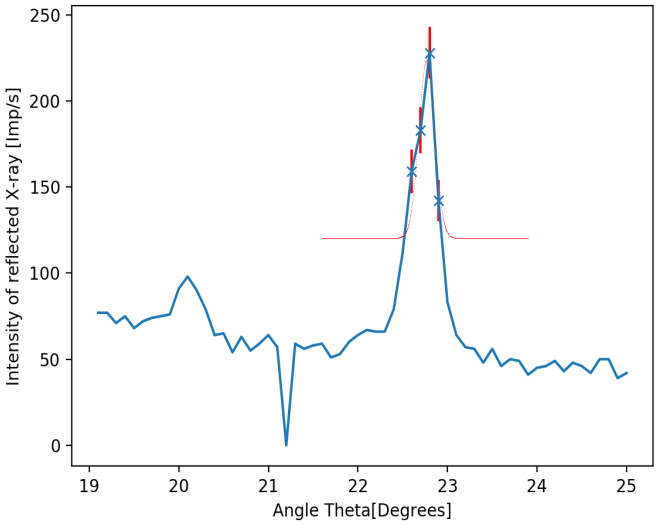
,

Figure 11: Gaussian fit to find more proper $K_{\alpha 2}$ ray degree



,

Figure 12: Gaussian fit to find more proper $K_{\beta 3}$ ray degree



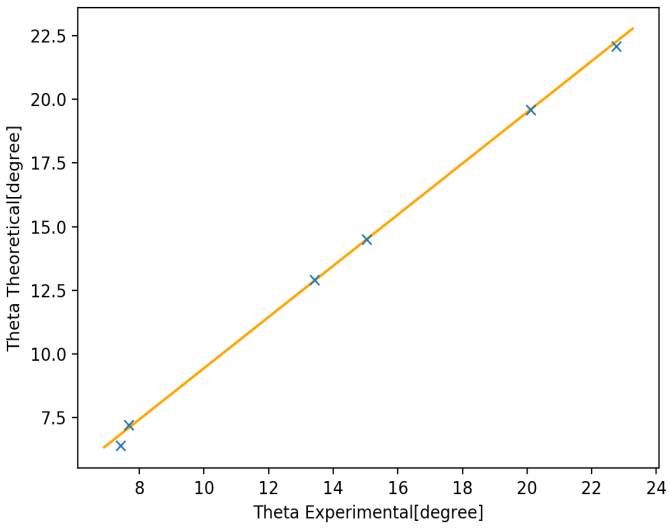
,

Figure 13: Gaussian fit to find more proper $K_{\alpha 3}$ ray degree

	$\theta_{th.}$	$\theta_{exp.}$	$\sigma_{\theta_{exp.}}$
$K_{\beta 1}$	6.41°	7.41°	0.01°
$K_{\alpha 1}$	7,20°	7.66°	0.01°
$K_{\beta 2}$	12.9°	13.42°	0.01°
$K_{\alpha 2}$	14.5°	15.03°	0.01°
$K_{\beta 3}$	19.6°	20.10°	0.01°
$K_{\alpha 3}$	22.1°	22.77°	0.03°

Table 1: measured peak of gauss fit and its error in degree

In table 1; theoretical θ values calculated by 2dsinθ=mλ formula. d is taken 2.83$\mathring{A}$ [5]. $\lambda_{k\alpha_1}$ taken 0.0709nm and $\lambda_{k\beta_1}$ taken 0.0632nm. m = 1,2,3[4]



,

Figure 14: Linear Fit to $\theta_{th.}$ versus $\theta_{exp.}$ graph

$$\theta_{th.} = m \cdot \theta_{exp.} + c \tag{5}$$

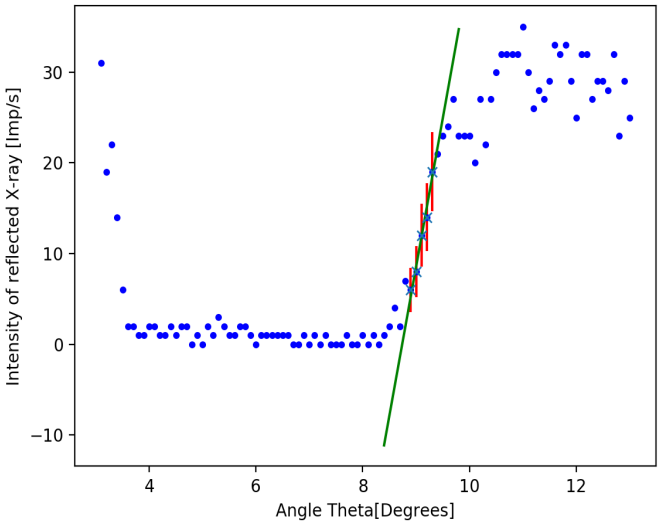
Where m is slope and c is constant. We will use these to calibrate coming experimental data. We calculate these value via Python and found; $m = 1.00 \pm 0.01$ and $c = -0.62° \pm 0.20°$. Change the dimension of c degree to radian. Then we get followin calibration formula, and calibrated uncertainty calculated with error propragation formula;

$$\theta_{calibrated}[radian] = (1.00 \pm 0.01) \cdot \theta_{exp.}[radian] - (0.011 \pm 0.0035)[radian] \tag{6}$$

$$\theta_{cal} = 1.00 \cdot \theta_{exp.} - 0.011 \tag{7}$$

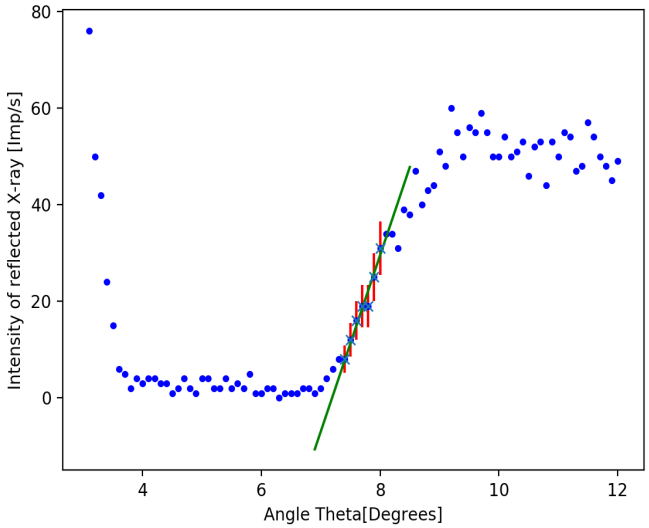
$$\sigma_{\theta cal.} = \sqrt{(0.01 \cdot \theta_{exp.})^2 + (1.00 \cdot \sigma_{\theta exp})^2 + (0.0035)^2} \tag{8}$$

4.2 Measuring ν_{max}



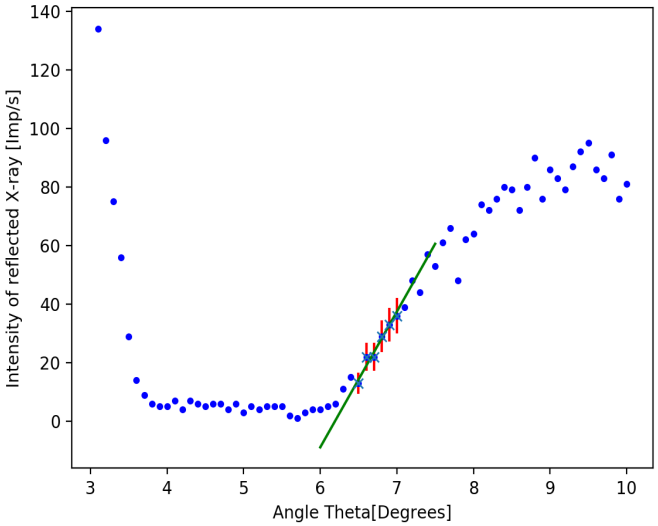
,

Figure 15: Linear Fit of Angels vs Intensity of reflected X-ray graph for 15 kV



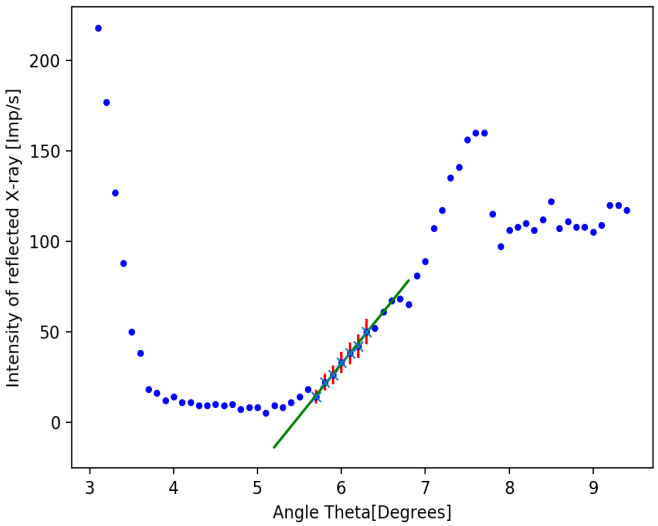
,

Figure 16: Linear Fit of Angels vs Intensity of reflected X-ray graph for 18 kV



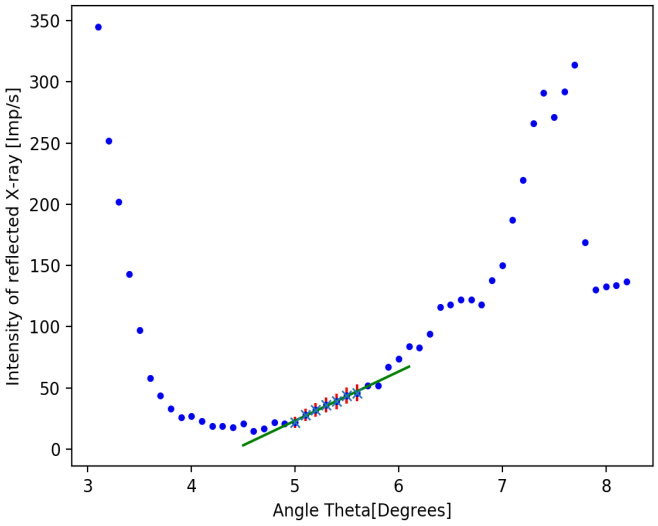
,

Figure 17: Linear Fit of Angels vs Intensity of reflected X-ray graph for 21 kV



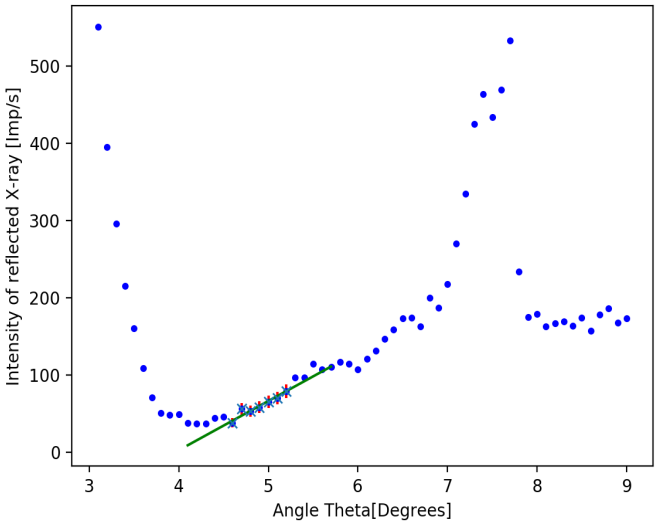
,

Figure 18: Linear Fit of Angels vs Intensity of reflected X-ray graph for 24 kV



,

Figure 19: Linear Fit of Angels vs Intensity of reflected X-ray graph for 27 kV



,

Figure 20: Linear Fit of Angels vs Intensity of reflected X-ray graph for 30 kV

θ[radian]	σ_θ[radian]	Frequency[$10^{18}Hz$]	$\sigma_{Frequency}$[$10^{18}Hz$]
0.152	0.019	3.48	0.44
0.126	0.017	4.23	0.57
0.108	0.017	4.91	0.77
0.0950	0.0054	5.58	0.31
0.0772	0.0060	6.87	0.53
0.0691	0.0151	7.67	1.68

Table 2: Not calibrated intersection angles in radian and calculated frequency with respect to Equation 11 and 12

Each linear fit function is like y = mx + c .Intersection points and its uncertainty of the graphs are calculated with following forumula and change degree to radian;

$$\theta = -\frac{c}{m} \tag{9}$$

$$\sigma_\theta = \sqrt{\left(\frac{\sigma_c}{m}\right)^2 + \left(\frac{c\sigma_m}{m^2}\right)^2} \tag{10}$$

where m is slope and $\sigma_m isitsuncertainty.$ C is constant of the fit function and σ_c is its uncertainty.
Calculation of frequency and its uncertainty via error propagation;

$$\nu = \frac{c}{2.d.sin\theta} \tag{11}$$

$$\sigma_\nu = \frac{c.cos\theta}{2.d.sin^2\theta} \cdot \sigma_\theta \tag{12}$$

where c is speed of light and taken 299 792 458 m/s[6]

θ[radian]	σ_θ[radian]	Frequency[$10^{18}Hz$]	$\sigma_{Frequency}$[$10^{18}Hz$]
0.164	0.020	3.25	0.39
0.136	0.018	3.89	0.49
0.119	0.017	4.46	0.65
0.106	0.006	5.01	0.30
0.0882	0.0070	6.02	0.47
0.0801	0.0156	6.62	1.28

Table 3: Calibrated θ and its uncertainty in radian calculated with Equation 7 and 8. After that, frequency and its uncertainty calculated with Equation 11 and 12

4.3 Measuring Planck's Constant

$$Energy = q \cdot V \tag{13}$$

Where V is applied voltage and q is charge of electron and taken 1.60217662 x 10^{-16}C[6]

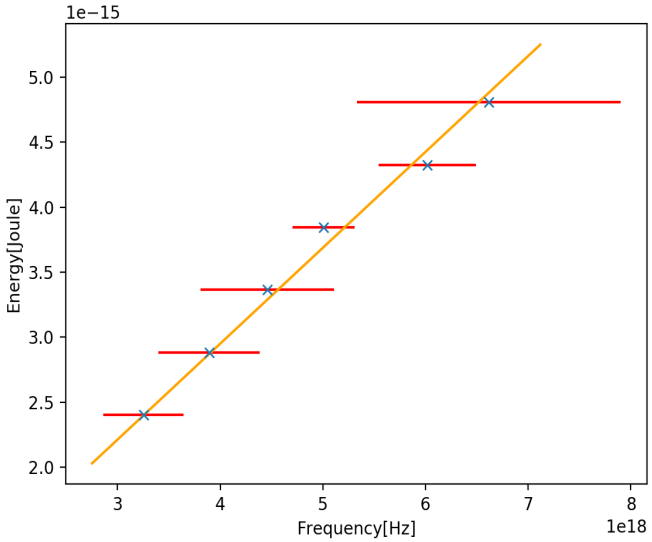


Figure 21: Linear Fitting with considering uncertainty of Frequency vs. Energy graph

The line in Figure 15 passing through origin, hence the equation of linear fit is E=mν. Where m is slope. The slope m is equal to Planck's constant(h). We measured Planck's constant as;

$$h_{measured} = 7.38x10^{-34}Js \pm 0.079x10^{-34}Js \tag{14}$$

5 Conclusion

This is the χ^2 of Planck's constant measurement which we found

$$\chi^2 = \left(\frac{7.38x10^{-34}Js - 6.62x10^{-34}Js}{0.079x10^{-34}Js} \right)^2 = 92 \tag{15}$$

if we don't use calibration then we will find Planck's constant as 6.54 x 10^{-34} Js $\pm$ 0.13 x 10^{-34} Js and the χ^2 will be

$$\chi^2 = \left(\frac{6.54x10^{-34}Js - 6.62x10^{-34}Js}{0.13x10^{-34}Js} \right)^2 = 0.38 \tag{16}$$

First of all, calibration has very important role in every experiment and it should be implemented.However, as seen in χ^2 values, calibration function falsify our accuracy but increase precision of measurement. This is why χ^2 was increased incredibly when we implement calibration. Why it was happened? Is it Luck? The Answer is absolutely Luck or making lucky mistake. In Figure 14, you can see slope is 1 but function has a negative constant, this is mean that there is problem with the angle, the written all angle is a little smaller than the actual angle. if the slope was higher or smaller than 1, our calibration function should have been wrong. For this reason, calibration function is correct. Probably we made mistake in line fit and it causes the interchange of intersection point on the x-axis. Our all line fits should have been more perpendicular.

Another problem is probably the collimator of the Phywe X-Ray Unit. For this reason, incident x-ray's not coming towards to NaCl crystal in a straight line, they coming scatteredly. Then, close wavelengths interfere each other. It is also a reason of having mistake in line fitting. This is why our line fit was not more perpendicular.

Also other possibility of error is Nacl crystal.It might slightly rippled. If all distance between atoms not exactly same and they have slight difference, we would also observe interfering of close wavelengths each other.

References

[1] Arthur Beiser - Concepts of Modern Physics (2003, McGraw Hill) page 68-74

[2] Kenneth S. Krane-Modern Physics-Wiley (2012) Page 73

[3] William R. Hendee, E. Russell Ritenour-Medical Imaging Physics-Wiley-Liss (2002) page 82

[4] PHYS 442 - SPRING 2019 (X-RAY EXPERIMENT)
https : *//d1b10bmlvqabco.cloudfront.net/attach/jruio0miog73c2/he2w4811edi5q4/jsw8ewq2dmna/*

[5] Advanced Physics Experiments, Erhan Gulmez, Bogazici University Publications, 1999,ISBN 975-518-129-6

[6] Constants
https : *//physics.nist.gov/cuu/Constants/*

[7] line fitting with orthogonal distance regression in data analysis (scipy.odr)
https : *//docs.scipy.org/doc/scipy/reference/odr.html*

Appendices

```
####Full Spectrum####
import numpy as np
import matplotlib.pyplot as plt
import pylab
import pandas

colnames = ['A1', 'B1']
data = pandas.read_excel("35_kv.xlsx", names=colnames)
icc = data.A1.tolist()
vcc = data.B1.tolist()
angle=np.array(icc)
intens=np.array(vcc)

plt.xlabel('Angle Theta[Degrees]')
plt.ylabel('Intensity of reflected X-ray [Imp/s]')
plt.plot(angle, intens)

plt.show()

#### One example of gauss fit####
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy import asarray as ar,exp
import pylab
import pandas

colnames = ['A1', 'B1']
data = pandas.read_excel("35_kv.xlsx", names=colnames)
icc = data.A1.tolist()
vcc = data.B1.tolist()

angle=icc[39:45] # arrange where we get th data
intens=vcc[39:45]
angle=np.array(angle)
intens=np.array(intens)

n = len(angle) #for begining point of parameter of fit
mean = sum(angle)/n
sigma = np.sqrt(sum((angle-mean)**2)/n)



def gaus(x,a,x0,sigma,c):
    return a*exp(-(x-x0)**2/(2*sigma**2)) +c

popt,pcov = curve_fit(gaus,angle,intens,p0=[1,mean,sigma,1.])
```

```
x_fit = np.linspace(angle[0]-1, angle[-1]+1, 1000)
y_fit = gaus(x_fit,*popt)

plt.errorbar(angle,intens,xerr=0,yerr=np.sqrt(intens),linestyle='None',marker
plt.xlabel('Angle Theta[Degrees]')
plt.ylabel('Intensity of reflected X-ray [Imp/s]')
plt.plot(icc[30:70], vcc[30:70])
plt.plot(x_fit,y_fit,'r,')

plt.show()


print(popt)
print(angle)
print(np.sqrt(np.diag(pcov)))

###calibration fit###
import numpy as np
import matplotlib.pyplot as plt
from scipy.odr import *

ex=[7.41047218,7.66174719,13.4154789,15.0341798 ,20.1043927,22.7669384]
exs=[0.0149985032,0.0000001,0.0000001,0.00240660654,0.005873,0.0297995619]
th=[6.41,7.2,12.9,14.5,19.6,22.1]
ths=[0.01,0.01,0.01,0.01,0.01,0.01]

def fit_func(p, x):
    m,c= p
    return m*x+c

linear = Model(fit_func)

data = RealData(ex, th, sx=exs, sy=ths)

odr= ODR(data, linear, beta0=[0., -1.])

out = odr.run()

out.pprint()

x_fit = np.linspace(ex[0]-0.5, ex[-1]+0.5, 1000)
y_fit = fit_func(out.beta, x_fit)

plt.errorbar(ex,th,xerr=exs,yerr=ths,linestyle='None',marker='x',ecolor='r')
plt.xlabel('Theta Experimental[degree]')
plt.ylabel('Theta Theoretical[degree]')
plt.plot(x_fit, y_fit,color='orange')

plt.show()

### One example of intersection ####
import numpy as np
```

```
import matplotlib.pyplot as plt
from scipy.odr import *
import pylab
import pandas

colnames = ['A1', 'B1']
data = pandas.read_excel("15_kv.xlsx",  names=colnames)
icc = data.A1.tolist()
vcc = data.B1.tolist()
angle=icc[58:63]  # arrange where we get th data
intens=vcc[58:63]
angle=np.array(angle)
intens=np.array(intens)



def fit_func(p, x):
    m ,c= p
    return m*x+c

linear = Model(fit_func)

data = RealData(angle, intens, sy=np.sqrt(angle))

odr= ODR(data, linear, beta0=[0., -1.])

out = odr.run()

out.pprint()

x_fit = np.linspace(angle[0]-0.5, angle[-1]+0.5, 1000)
y_fit = fit_func(out.beta, x_fit)

plt.errorbar(angle,intens,xerr=0,yerr=np.sqrt(intens),linestyle='None',marker
plt.xlabel('Angle Theta[Degrees]')
plt.ylabel('Intensity of reflected X-ray [Imp/s]')
plt.plot(icc[:100], vcc[:100],'b.')
plt.plot(x_fit,y_fit,'g-')

plt.show()

print('intercection of x-line[degree]',-out.beta[1]/out.beta[0])
print('sigma[degree]',np.sqrt((out.sd_beta[1]/out.beta[0])**2+
        ((out.beta[1]*out.sd_beta[0])/out.beta[0]**2)**2))
print('intercection of x-line[radian]',-(np.deg2rad(out.beta[1])/out.beta[0])
print('sigma[radian]',np.sqrt((np.deg2rad(out.sd_beta[1])/out.beta[0])**2+
        ((np.deg2rad(out.beta[1])*out.sd_beta[0])/out.beta[0]**2)**2))

#### Calibrate angles ####

import numpy as np
ang=np.array([0.15253511843098186,0.1255079976721513,0.10807048315498674,
```

```
        0.09497221757899951,0.07715505410475225,0.06912763757240625])
sig=np.array([0.019461359507793277,0.01703081042359836,0.01705287607527344,
        0.005361560486049449,0.005965762890487764,0.015148129426351 82])
cang= 1.*ang + 0.011
csig= np.sqrt((ang*0.011)**2+(1.*sig)**2+(0.0035)**2)

print('calibrated angle',cang)
print('calibrated sigma',csig)


print(299792458./(2.*2.83e-10*np.sin(ang)))
print((299792458.*np.cos(ang)*sig)/(2.*2.83e-10*(np.sin(ang)**2)))
print('CALIBRATED')
print(299792458./(2.*2.83e-10*np.sin(cang)))
print((299792458.*np.cos(cang)*csig)/(2.*2.83e-10*(np.sin(cang)**2)))

#### Energy vs Frequency fit####
import numpy as np
import matplotlib.pyplot as plt
from scipy.odr import *

frq=np.array([3.48593958e+18, 4.23129843e+18, 4.91069407e+18,
        5.58548331e+18,6.87180556e+18 ,7.66828950e+18])
sig=np.array([4.41302657e+17 ,5.71148155e+17 ,7.71859151e+17,
        3.14374199e+17, 5.30285193e+17 ,1.67769590e+18])
energy = 1.60217662e-16*np.array([15.,18.,21.,24.,27.,30.])

def fit_func(p, x):
    m= p
    return m*x

linear = Model(fit_func)

data = RealData(frq, energy, sx=sig)

odr= ODR(data, linear, beta0=[0.2])

out = odr.run()

out.pprint()

x_fit = np.linspace(frq[0]-0.5e+18, frq[-1]+0.5e+18, 1000)
y_fit = fit_func(out.beta, x_fit)

plt.errorbar(frq, energy, xerr=sig, linestyle='None', marker='x',ecolor='r')
plt.xlabel('Frequency[Hz]')
plt.ylabel('Energy[Joule]')
plt.show()
```