

Generating X-rays through Channeling Radiation

Edgar A. Mercado

Supervisor: Tanaji Sen

Summer Internships in Science and Technology (SIST)
Program at Fermilab

University of Illinois at Urbana-Champaign

August 7, 2014



ABSTRACT:

Channeling radiation is a sufficient way of producing X-rays. The goal of this project is to develop a compact X-ray source. Because of the delays of the installments of equipment, there was no real data to record. Instead, simulations for this experiment were created via Python script and will be used to help analyze and find data within the plots of the real experiment. These simulations will contribute to a successful run once all of the devices and software are ready to conduct the experiment.

1. INTRODUCTION:

For the summer of 2014 at Fermi National Laboratory, the project I contributed to consisted of writing out simulations in Python script for data collecting and reading from the experiment itself. The experiment entails shooting an electron bunch through a crystal and producing an X-ray source through channeling radiation (CR). Although some of the devices along the beam line were not installed this summer, I was still able to stay busy and learn a lot from this project. The experiment is being held in the ASTA building with a beam line that will produce the X-rays that we will analyze once the experiment is conducted.

The end result of this experiment will be to develop a compact X-ray radiation source through CR. Kumakhov theoretically predicted CR in 1974 (from Ref. [1]). Terhune and Pantell also experimentally observed CR in 1975 [2]. CR is emitted by relativistic charged particles passing through a sheet of positive charge in a crystal. The radiation that is produced by the oscillating electron bunch through the crystal is called Channeling Radiation. The CR peaks are an order of magnitude above the bremsstrahlung background radiation. Channeling radiation provides advantages over other sources of radiation when producing X-ray signals.

Although the main focus of this experiment is to produce X-rays through channeling radiation, I was not able to conduct the real experiment due to the delay of installment in the ASTA beam line. However, I was given plenty of work and assignments while I was here at Fermilab. I wrote many scripts that will contribute to further work on this project. A majority of the scripts I wrote were simulations. The structure of these simulations will be used to analyze the real data once the real experiment is conducted.

2. MATERIALS and METHODS:

a. In order to conduct this experiment, we need a 30-50 MeV electron beam. This beam is located in the ASTA building. The components of this beam

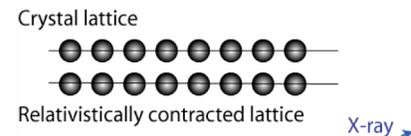


Figure 1. Mechanism of channeling radiation production.

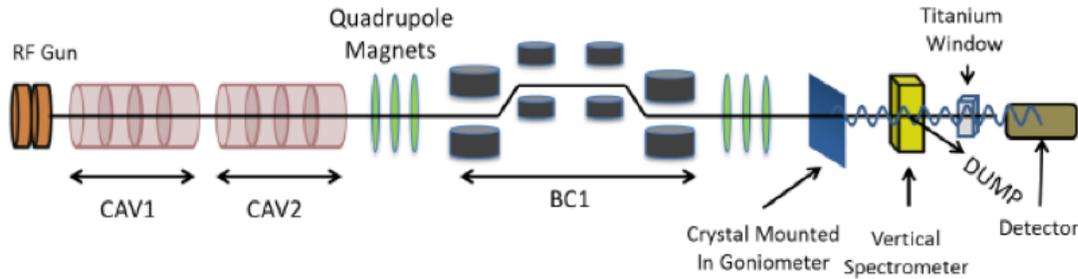


Figure 2. ASTA beam line components and structure.

include the photoinjector, a $\sim 40\mu\text{m}$ diamond crystal, a goniometer, and an X-ray detector. These devices are components that were significant to my part in this experiment. There are several other devices within the beam line that I do not mention. The start of the beam line occurs at the RF-gun that produces low energy electron bunches.

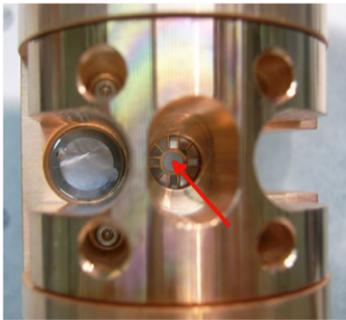


Figure 3. Targets for the production of bremsstrahlung (10mm on the left) and channeling radiation (3mm in the center). The direction of the electron beam is seen as the red arrow.

horizontal and vertical axis. It also translates along the horizontal axis in order to position the crystal correctly in the right angle to produce the best quality X-rays. My original assignment was to create a script that will communicate with the control system and tell the goniometer to rotate and translate a specific amount of steps. Due to the delay in the installments of the devices, this could not be done.

b. Since I was not able to conduct and run scripts using the goniometer I was assigned to write Python scripts that will analyze data form the X-ray detector. The detector will produce spectrums,

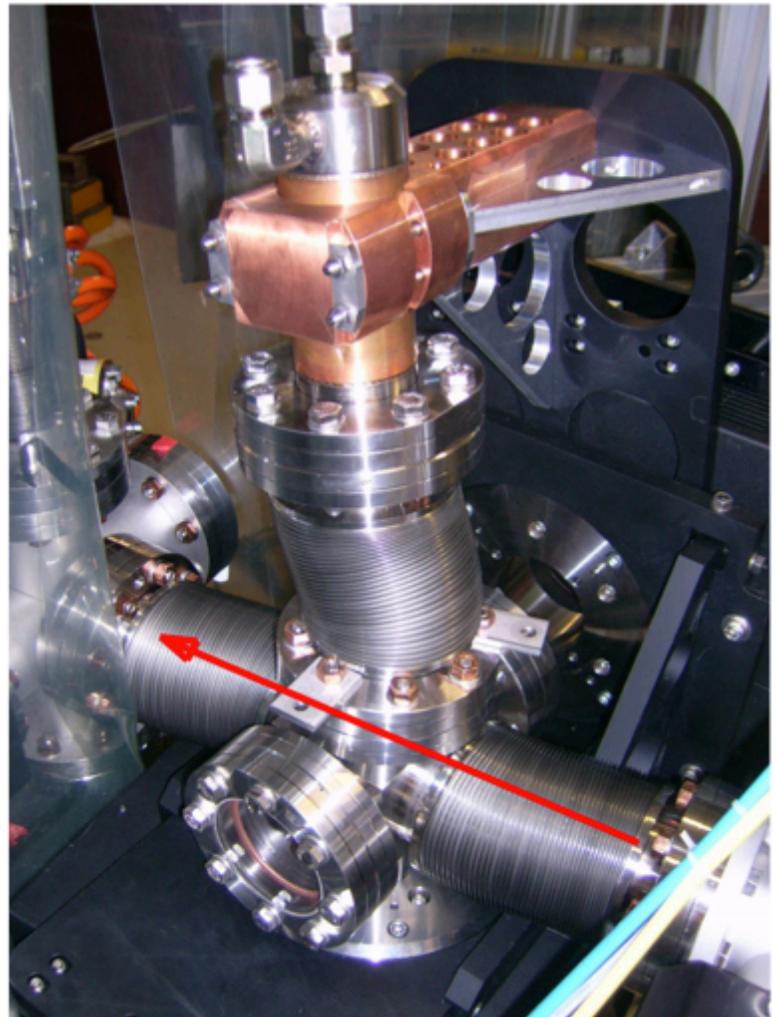


Figure 4. Goniometer located in the ASTA beam line. The direction of the electron beam is indicated by the red arrow.

which my simulations are designed to read and write out data that is found most significant amongst the plots.

Using previous data and my supervisor's expertise I was able to write out multiple scripts throughout the summer and learn a lot about scripting as well as channeling radiation. I was able to find create several different ways to produce such plots (right) and find the significant values on the plots for data analysis.

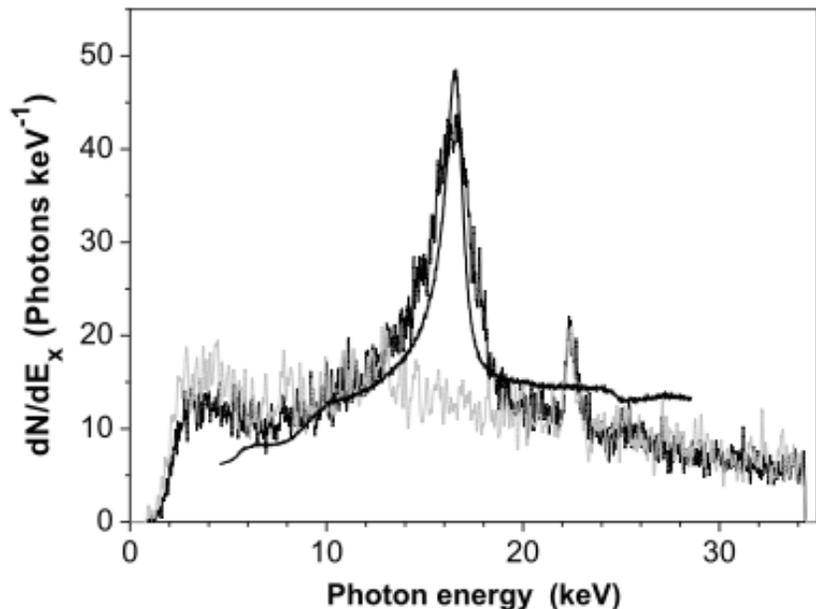
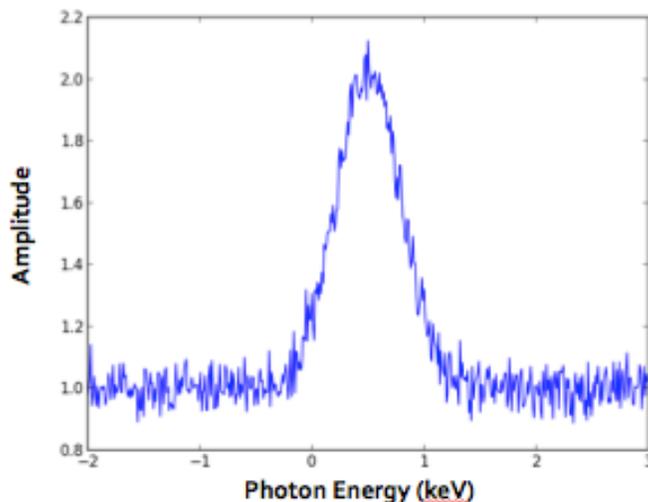


Figure 5. Measured X-ray signal. Conducted experiment in 2007. [3]

Some of these values included peak values and their indices, full width at half maximum of the peaks to display the size of the signals, total bunch charge in a current plot, and creating a script that will find multiple peaks in a plot from a simulated X-ray signal. The end value that we want from these simulations is the normalized photon intensity. Which is the peak value of the X-ray signal divided by the total bunch charge. While creating these simulations, I received plenty of help from Google and Todd Seiss, a student from Princeton University who was also working under Tanaji Sen.

3. RESULTS:

a. Using python script I was able to produce the following plots. The first



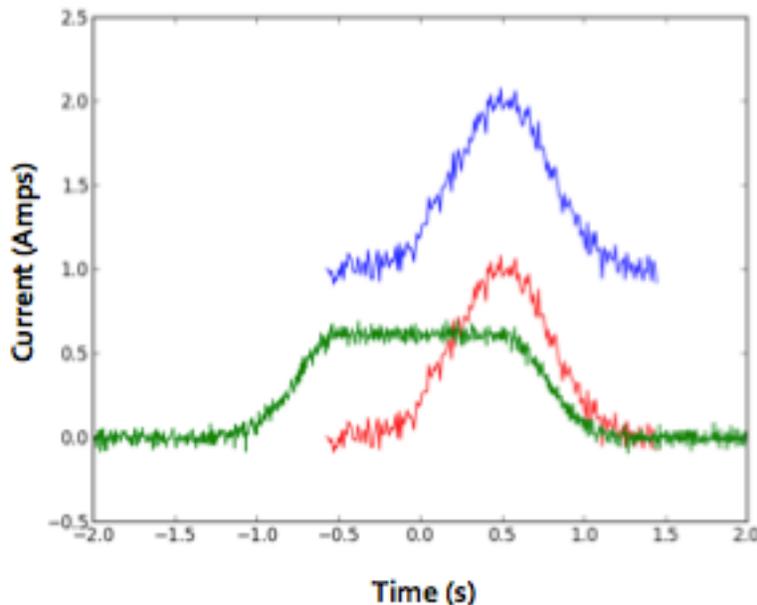
plot is a Gaussian distribution with noise that represents the X-ray signal after being detected and produced through channeling radiation from the signal. The values on the graph axis are not true values and are set values that are determined within the script itself.

b. The second plot is of the original distribution

and the corrected signal. Taking the average of the background, which is the average of the tail values of the distribution, and subtracting it from the original points of data formalize the corrected signal. The corrected signal is plotted in the color red.

c. The third plot is mainly of the current that is being plotted as the electron bunch passes through the crystal and produces a current. The value that is needed from this plot is the integrated value of the

function which is declared as the total bunch charge. In this plot I also include the original and corrected signals of the X-ray. Notice the axis values are changed in the third plot due to the change of data. We want to pay attention to the current in this plot.

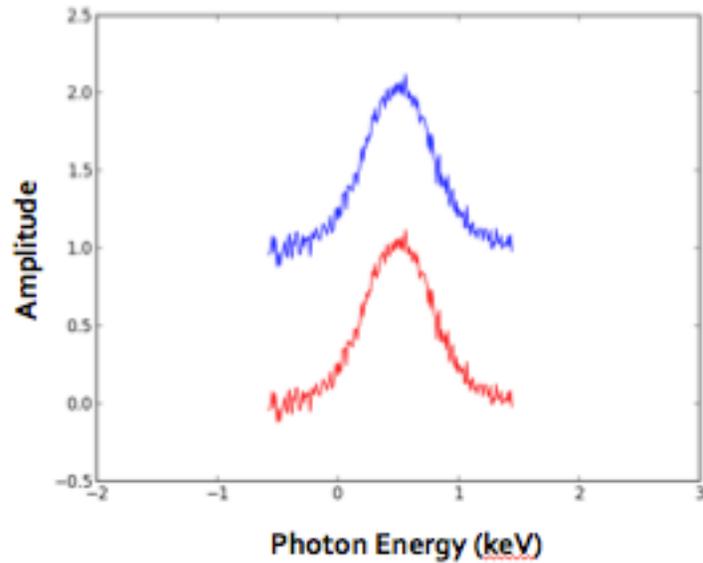


the current in this plot.

d. In the fourth plot, I find the full width at half maximum (FWHM) value. In order to do this, I found the absolute max value of the plot and then found the half max value. After finding the half max value, I scanned the plot for the spot where the value passes that half max value. Once I found those two spots I marked the indices of those values along the plot. Once I recorded those values, I was able to

subtract the indices from each other and that gave me the FWHM value. This is a standard form of use to determine the size and quality of the signal. The more narrow the peak, the better quality the signal is. I also use the FWHM formula to determine distances between peaks in the following plot.

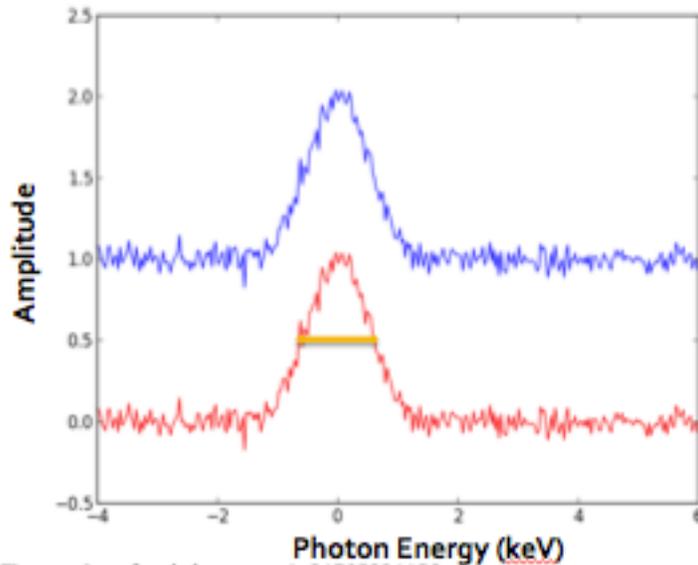
e. The fifth and final plot I created through simulation was to find multiple peaks and the peak values and their indices. This simulation was the most



difficult because there are so many ways this could be done but I had to make sure I covered every base and made sure I was able to find every peak value no matter what the plot looked like.

For this example, I used the sum of two Gaussian distributions with noise and was able to find the peak values that way. I first found the absolute max value within the plot and scanned the plot to the left. If I found a peak that was significant, meaning the peak value was greater than the value of half max of the previous peak, then I would mark that as a local max value.

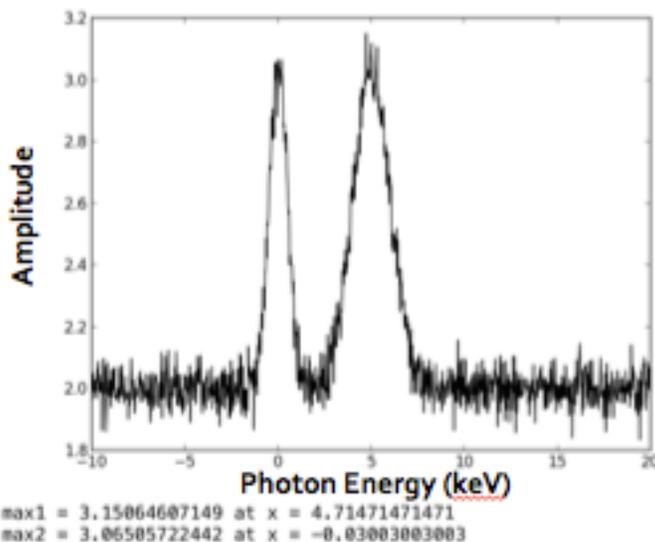
After finish scanning the left of the absolute max value, I scanned the plot on the right for a local max value. I scanned until there were no more significant max values. I printed out the max values and their indices.



The peak value(s) are: 1.21795221156
The width at half maximum is... 1.08108108108

This was important because when scanning the X-ray signal plot, we would like to find all significant peak values.

These were all the plots that I worked on throughout the summer while understanding the methods of how the experiment itself will be conducted. I used Python script and saved all my files.



4. CONCLUSION:

All of these simulations and python script will be submitted to my supervisor and the structure of the scripts will contribute to further work with the experiment. Once the experiment is up and running they can use my script to help analyze their data and help find the values they are looking for. In the end, this experiment's goal is to develop a compact X-ray source.

Future work will include testing various crystals to examine what happens when using other material. We may also use different energies and lengths of the beam line to experiment what is the best method to develop a sufficient X-ray source. This project will benefit Hospitals, Homeland security, Material Sciences, and various other fields of work.

5. APPENDICES:

Here are some examples of my script:

5.1.

```
#find multiple peak values
#edited 7/31/14
#Written by Anthony Mercado

import numpy as np
from scipy.optimize import leastsq
import matplotlib.pyplot as plt
import random

#Setting up data
rand = [random.random() for _ in range(100000)]
num = random.random()*0.1
#mu = numpy.average(rand)
#sigma = numpy.std(rand)

def norm(rand, mean, sig):
    norm = (1.0+num)*np.exp(-(mean - rand)**2) / (2 * sig**2) + 1.0
    return norm

mean1 = 0.0
mean2 = 5.0
#mean3 = 5.0
std1 = 0.5
std2 = 1.0
#std3 = 0.75

x = np.linspace(-10, 20, 1000)
y_real = []

#Creating noise
for val in x:
    normy = norm(val, mean1, std1) + norm(val, mean2, std2)
    y_real.append(np.random.normal(normy, .05))

#Get imin and imax, the indices of the point where y drops
#beneath half its peak value (above the noise) around the peak
```

```

def getFWHMi(y):
    #Compute the background level
    N = 10
    for i in range(0, len(x)):
        if x[i] > -(N-2)*std1:
            break
    avg1 = abs(np.average(y_real[0:i]))

    for j in range(0, len(x)):
        if x[j] > (N+2)*std1:
            break
    avg2 = abs(np.average(y_real[j-1:]))
    avg = abs((avg1+avg2)/2)

    #Compute the FWHM of the peak above the background
    half_max = (max(y)/2.0) + avg/2.0
    iOfMax = y.index(max(y))
    iscan = [iOfMax - i for i in range(0, iOfMax)]
    for i in iscan:
        if y[i] <= half_max:
            imin = i
            break
    iscan = range(iOfMax, len(y))
    for i in iscan:
        if y[i] <= half_max:
            imax = i
            break
    return imin, imax

#Gets the indices of the first N maxima of y, assuming that each maxima occur
#more than the width at half maximum away from each other
def getNMaxima(y, N):
    iOfMaxArray = []
    for _ in range(0, N):
        imin, imax = getFWHMi(y)
        iOfMaxArray.append(y.index(max(y[imin:imax])))
        y = np.append(y_real[0:imin], y_real[imax:])
        y = y.tolist()
    return iOfMaxArray

[iOfMax1, iOfMax2] = getNMaxima(y_real, 2)
print "max1 =", y_real[iOfMax1], "at x =", x[iOfMax1]
print "max2 =", y_real[iOfMax2], "at x =", x[iOfMax2]

#Plot
plt.plot(x, y_real, 'k')

```

```
#Show me the plot
plt.show()
```

```
#end
```

5.2.

```
#double gaussian
# and finding width at half maximum
#edited 7/29/14
#Written by Anthony Mercado
```

```
import numpy as np
from scipy.optimize import leastsq
import matplotlib.pyplot as plt
import random
```

```
#Setting up data
rand = [random.random() for _ in range(100000)]
num = random.random()*0.1
#mu = numpy.average(rand)
#sigma = numpy.std(rand)
```

```
def norm(rand, mean, sig):
    norm = (1.0+num)*np.exp(-(mean - rand)**2) / (2 * sig**2)+1.0
    return norm
```

```
mean1 = 0.0
mean2 = 5.0
#mean3 = 5.0
std1 = 0.5
std2 = 2.0
#std3 = 0.75
```

```
x = np.linspace(-10, 20, 1000)
y_real = []
```

```
#Creating noise
for val in x:
    normy = norm(val, mean1, std1)# + norm(val, mean2, std2)
    y_real.append(np.random.normal(normy, .05))
```

```
#AVG of BOTH Tails - one value
#N = 'user input'
N = 10
for i in range(0, len(x)):
```

```

        if x[i] > -(N-2)*std1:
            break
    avg1 = abs(np.average(y_real[0:i]))

    for j in range(0, len(x)):
        if x[j] > (N+2)*std1:
            break
    avg2 = abs(np.average(y_real[j-1:]))
    #print avg1
    #print avg2

    avg = abs((avg1+avg2)/2)

    #Remove tails
    x = x[i:j]
    y_real = y_real[i:j]

    #Finding width at half maximum
    half_max = (max(y_real)/2.0) + 0.5

    minSet = False
    maxSet = False
    for i in range(0, len(y_real)):
        if (y_real[i] >= half_max) and not minSet:
            imin = i
            minSet = True
        if (y_real[i] <= half_max) and not maxSet and minSet:
            imax = i
            maxSet = True
    #print y_real

    #Finding peak values
    max = max((y_real-avg))

    #Descending array
    y_new = sorted((y_real-avg), reverse = True)

    #Finding local peak values

    #Plot
    plt.plot(x, y_real)
    plt.plot(x, (y_real-avg), 'r')

    #Print
    print "The peak value(s) are: ", max

```

```

print half_max
#print imin, imax
#print y_real[imin], y_real[imax]
#print avg
print "The width at half maximum is... ", x[imax]-x[imin]
#print (y_real-avg)
print y_new

#Show me the plot
plt.show()

#end

```

6. ACKNOWLEDGEMENTS:

I would like to thank my supervisor Tanaji Sen very much for investing his time in me and guiding me throughout the summer on this project. I greatly appreciate what he did because he took the time to mentor me when he absolutely does not have to take on a student. I would also like to thank fellow co-workers in the group: Philippe Piot, Daniel Mihalcea, and Todd Seiss, for assisting me in understanding the material and the task at hand. Without them, I would not have been able to keep up the good work that was being done.

I would like to acknowledge Ms. Dianne Engram and Ms. Linda Diepholz for giving me this wonderful opportunity and accepting me into this program this summer. I understand there were plenty of qualified students for this position and I am honored to have been one of the chosen few. I would also like to thank Ms. Sandra Charles for keeping in touch with me from the TARGET program I participated in, in 2012, and referring me to the SIST program. I want to thank Dr. Davenport for his knowledge and for making this program possible. Without him, the opportunity to experience this internship would not be possible.

Last but not least, I would like to acknowledge the rest of the Fermilab staff for their tremendous work in such a successful program, and also the Department of Energy for funding this program. Thank you greatly for this amazing experience.

7. REFERENCES:

- 7.1. C. A. Brau, *Inst. of Rad. Phys.*, 1 (2013); M. A. Kumakhov, *Phys. Lett. A* 57, 17 (1976).
- 7.2. C. A. Brau, *Inst. of Rad. Phys.*, 1 (2013); R. W. Terhune and R. H. Pantell, *Appl. Phys. Lett* 30, 265 (1997).
- 7.3. W. Wagner, B. Azadegan, M. Sobiella, J. Steiner, K. Zeil, and J. Pawelke, *An intense channeling radiation source*. *Nucl. Instrum. Meth.* B266, 331 (2008).