Monte Carlo Simulation of Radiation Transport-RADIATION AND DOSIMETRY,FYS4711 20H

Shailendra Bhandari*

November 25, 2020

Abstract

Monte Carlo Method (MCM) is widely used for solving the radiation transport problem. This technique is based on the simulation of history of particle which account for the energy transfer mechanism. Monte Carlo simulation needs a voluminous quantity of random numbers in the range [0,1] and the probability density function.Applications range from solving problems in theoretical physics to predicting trends in financial investments.This report briefly describes the Monte Carlo simulation of random number generator and the transport of radiation through matter (water) with only Comptom Scattering interaction between the energy range of 200 keV and 2 MeV.

I Introduction

The Monte Carlo Method (MCM) is like playing a game of chance, using randomized mathematical methods to determine the estimates of numerical quantities. MCM is widely used in advanced mathematical and statistical techniques which can solve the variety of problems including optimization and numerical integration problems. These algorithms works by cleverly sampling from a distribution to simulate the workings of a system. Applications of MCM range from solving problems in theoretical physics to predicting trends in modern financial investments.This report briefly describes the Monte Carlo simulation of random number generator and the transport of radiation through matter (water) with only Comptom Scattering interaction between the energy range of 200 keV and 2 MeV.

II Theory

II.1 Random Number

Generating a random number is a prerequisite to use any of the Monte Carlo Methods in computer programming. Random num-

*shailendra.bhandari@fys.uio.no Department of Physics, University of Oslo,Blindern,Norway.

bers are sampled from the variables to perform integration and simulations. Hence they are of primary requisite. The random numbers need to be distributed uniformly within the interval of [0,1].

II.2 The Monte Carlo Simulation of Transport of Radiation through matter

Transport of radiation is necessary in many fields such as radiation detection, radiation dosimetry and treatment planning. Solving the transport problems for photons can be performed by means of deterministic and Monte Carlo algorithms. The Monte Carlo method interpret a particle transport process as a statistical process where the large number of random histories of individual particles are simulated by the computer and averaged, to obtain the estimates of mean fluxes.

The transport of radiation through matter is governed mainly by the attenuation coefficients or by the cross sections. The of linear attenuation coefficient describe the statistical nature of the radiation penetration in matter.

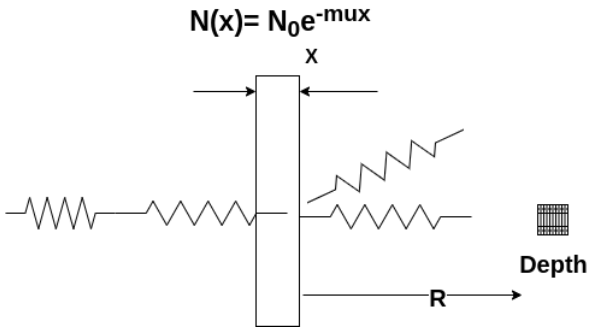

Figure 1. A normally incident photon in a matter with thickness x.

The probability of a normally incident photon in Fig 1 reaching the depth x in a material without interaction is

$$P(x) = e^{-\mu x} \tag{1}$$

Also, the probability that the first interaction of an incident photon will take place at a depth between x and $x + dx$ is

$$P_1(x)dx = P(x)\mu dx \tag{2}$$

Where P(x) is the probability that it will reach the depth x and μdx is the probability that it will reach in depth dx.

The cumulative probability is that a normally incident photon will interact before reaching a depth x.

$$P_c(x) = \int_0^x P_1(x)dx = \mu \int_0^x e^{-\mu x}dx \tag{3}$$

After the photon has travelled a pathlength, another task is to know the photon interaction. Different interaction phenomenon like Photoelectric effect, Rayleigh, Compton Scattering, or Pair production might happen. For any of the given interaction has certain probability proportional to its cross-section.

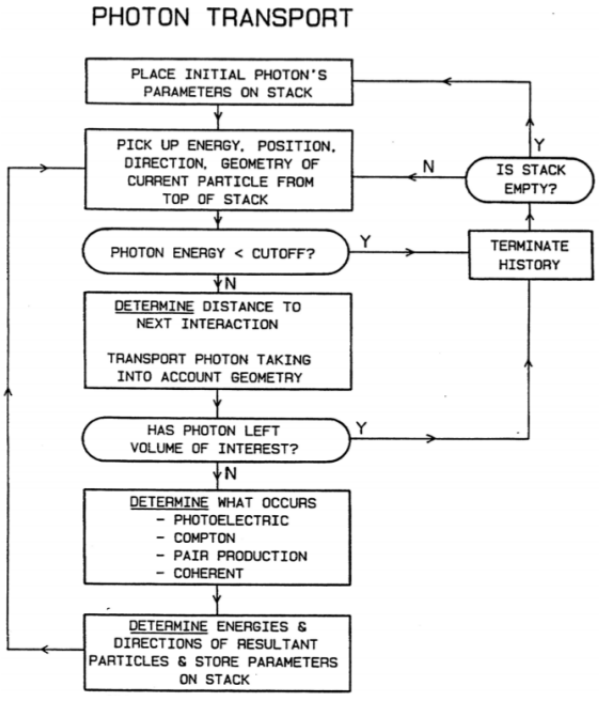

Figure 2. "Flow Chart of Monte Carlo simulation of photon transport.At each of these points one makes use of detailed knowledge of the physical processes involved in photon transport and, by sampling from an appropriate probability distribution, determines the parameters of the event". [1]

III Part-I: Results and Discussion of the Exercises

Random Number Generator and Random Number

First of all, a 1000 random numbers sampled from a uniform distribution between 0 and 1 is generated. The normal Gaussian distribution with a mean (μ) 0 and standard deviation (σ) of 1 is applied to the same random numbers. The histograms for this case is as shown in 3.

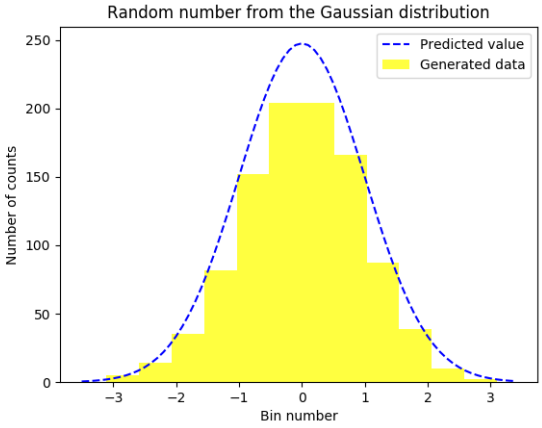

Figure 3. A histogram showing the standard normal distribution of 1000 random sample with $N(0,1)$,12 bins.

For 1000 particles with 100 steps for each particles when simulating with mean ($\mu = 0$) and the variance ($\sigma = 1$) for arbitrary particles; the histogram is as like in Figure 4. A normal Gaussian fit is also added in the histogram showing that it follows the standard normal distribution.

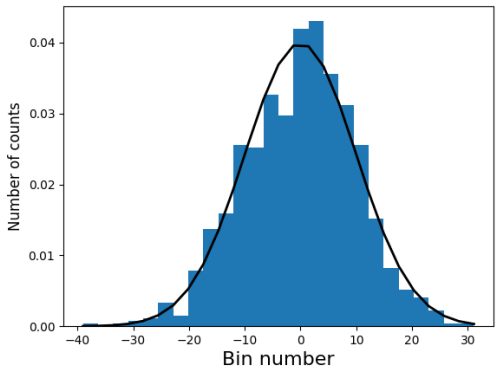

Figure 4. A histogram showing the standard normal distribution of 1000 particles with 100 steps for each particle,12 bins.

Figure 5 is the plot of the position of the 10 arbitrary particles versus the collision number.

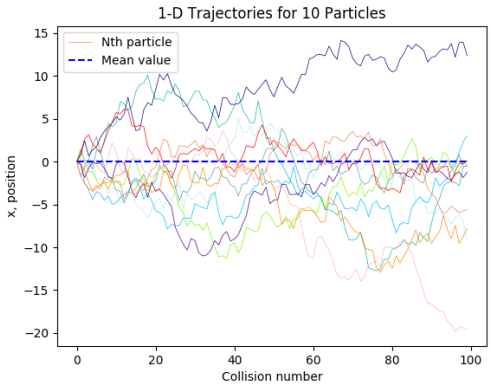

Figure 5. Plot of the position(y) vs the collision number(x) for 10 arbitrary particles.

Random walk in 1-D including energy loss.

The final distribution of 1000 particles along x-axis with 100 steps is illustrated on the figure 6. If comparing histogram 6 with the histogram from exercise 2 histogram 3, the distribution is centered by 0. This result is as per the expectations as we know that the modification in the form of energy does not change the step length of the particle after a certain collision between them.

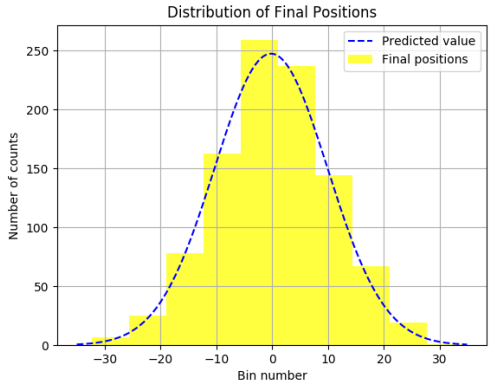

Figure 6. Distribution of final positions for 1000 particles,introducing energy and energy loss,12 bins.

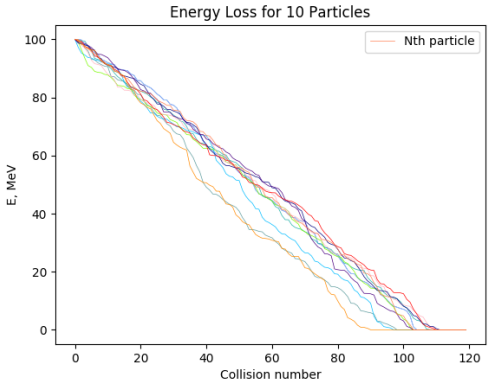

Figure 7. Energy loss of the 10 particles versus the collision number.

Figure 7 shows the gradual decrease in energy from 100 Mev to 0 for 10 particles. Also it can be seen that the average number of collision required for this energy loss phenomenon is about 100 collisions.

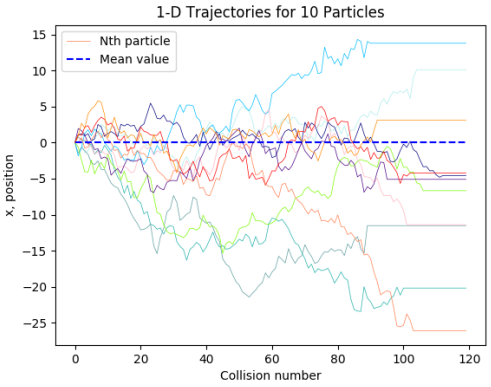

Figure 8. Distribution of position of the 10 random particles versus the collision number with energy loss.

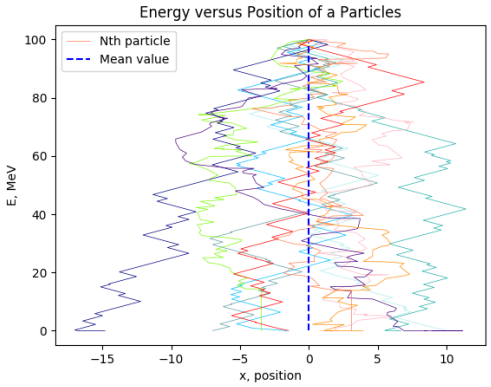

Figure 9. Energy loss for 10 particles versus the x-position.

Figure 8 manifest the position of the 10 particles plotted as a function of the number of collision. And figure 9 demonstrate the energy deduction of 10 arbitrary particles as a function of their x-position.

IV Part-II: Results and Discussion of the Exercises

The following simulations are for photons in water. The number of electrons per volume unit in water is $nV = 3.4310^{23} cm^{-3}$. We assume that the only interaction taking place is Compton scattering. The minimum and maximum photon energy is 50 and 2000 keV, respectively.

Creation of Attenuation Coefficient Function The total Klein-Nishina cross-section per electron (σ_e) for the Compton effect is given by the relation [2].

$$\sigma_e = 2\pi r^2 \left\{ \frac{1+\alpha}{\alpha^2} \left[\frac{2(1+\alpha)}{1+2\alpha} - \frac{ln(1+2\alpha)}{\alpha} \right] + \frac{ln(1+2\alpha)}{2\alpha} - \frac{1+3\alpha}{(1+2\alpha)^2} \right\} \tag{4}$$

where $\alpha = \frac{h\nu}{m_o c^2}$, in which $h\nu$ is expressed in MeV and $m_o c^2 = 0.511 MeV$. The relationship between the electron scattering cross-section and the attenuation coefficient(μ) is $\mu = n_v \times \sigma$. Equation 4 is used to calculate the μ value for energy 200keV and 2 MeV as Compton electronic cross-section is energy dependent function. The μ values are 0.138 $cm^{-1}$and 0.050 cm^{-1}respectively for 200 keV and 2 MeV. These μ values are in good agreement with the published value 0.137 $cm^{-1}$and 0.049 cm^{-1}. The plot of μ as a function of photon energy is shown in Figure 11.

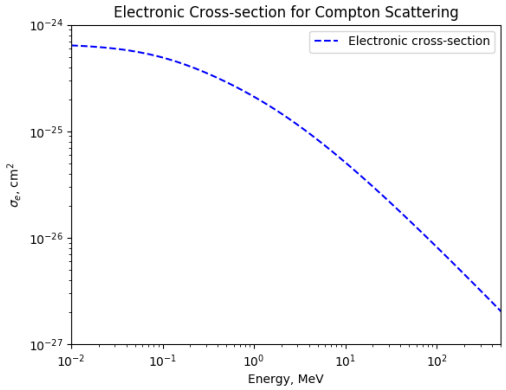

Figure 10. Energy dependence of the electronic cross-section for the Compton Scattering in water.

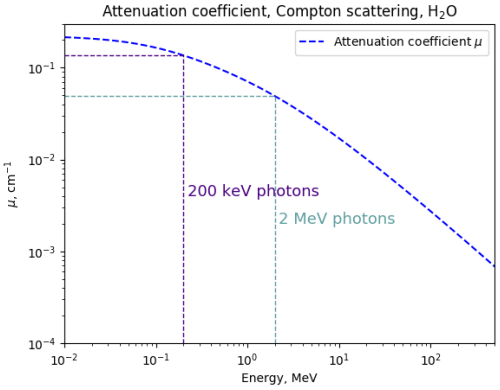

Figure 11. The plot of μ as a function of photon energy, energy loss of the 10 particles.

Probability Distribution

Using the same μ value as defined above the depth of the interaction is simulated by making the probability distribution function (PDF) and Cumulative Probability distribution function (CPD). The plot of PDF and CPD for 200 keV and 2 MeV photons is shown respectively in Figures[12, 13]. The blue lines in the plots is for 200 keV photons whereas the red is for 2 MeV.

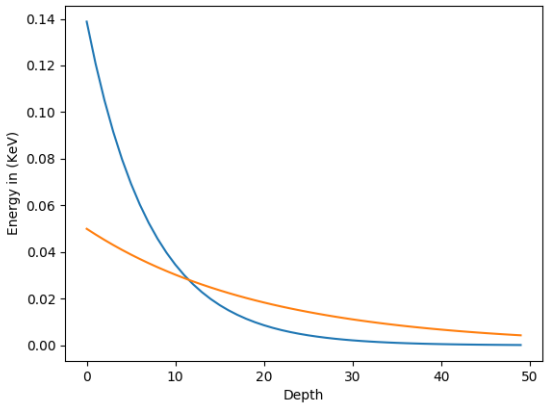

Figure 12. Plot of PDF for simulating the depth of interaction at 200Kev and 2 Mev Photons.

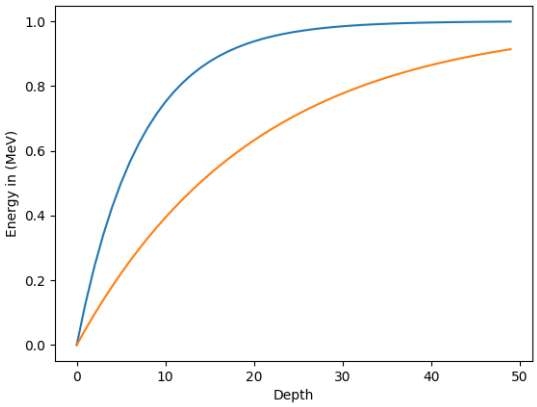

Figure 13. Plot of CPD for simulating the depth of interaction at 200Kev and 2 Mev Photons

The CPD in the above plot 12 fulfill the requirements because $P_c(0) = 0$ and $P_c(\infty) = 1$. We can see clearly in the figure that the both the lines for 200kEv and 2 MeV begins from zero and ends almost at 1.

Sampling photon steplengths using the inverse transform The same μ function as mentioned above is used to sample the photon steplength ΔS by inverse transform method for 200keV and 2 MeV photons.The data is presented in histogram in Figures [14,15] with probability density function for 200 keV and 2 MeV photons.The path of the histogram and the plots in figure12 are almost similar for 200 Kev but some complication in 2 MeV photons.

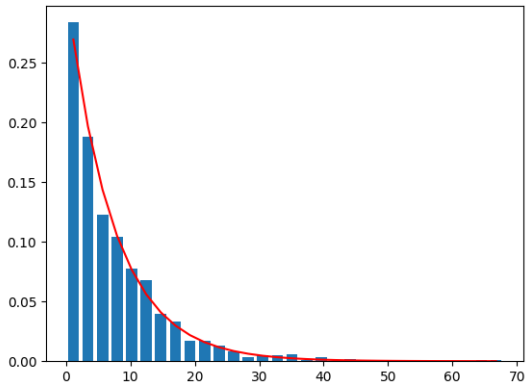

Figure 14. Histogram for 1000 photons simulated for steplength (ΔS) using inverse transform method for 200 keV photons.

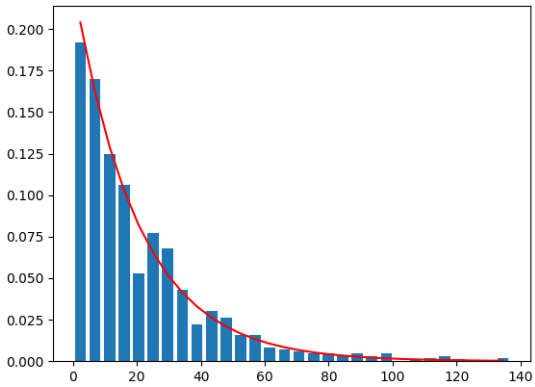

Figure 15. Histogram for 1000 photons simulated for steplength(ΔS) using inverse transform method for 2 MeV photons.

The probability that a free flight will be determined when its length is between s and ΔS is $P(s) = \sum e^{\sum s} ds$. To check the simulations are correct or not we now can compare the simulated mean value and the expected mean values.The simulated mean value m_s is

$$m_s = \int_0^\infty sp(s)ds = \int_0^\infty p\mu e^{-\mu s} ds = 1/\mu \tag{5}$$

The calculated m_s values of 200 keV and 2 MeV photons are 7.40 and 21.01 cm respectively.Also, the $1/\mu$ values for respective photons of 200 keV and 2 MeV are 7.20 and 20.00 respectively. As by equation 5 the calculated m_s values are nearly equal to $1/\mu$ values. Hence we can say that tha path length for these photons are valid.

Normalized Differential Scattering Cross-section

The differential Compton cross-section for the interaction of the energy of the incident photon($h\nu$), scattered photon($h\nu'$ (with respect to the solid angle $d\Omega$) is given by the relation [2].

$$\frac{d\sigma}{d\Omega} = \frac{{r_0}^2}{2}\left[\frac{h\nu}{h\nu'} + \frac{h\nu'}{h\nu} - sin^2\theta\right] \tag{6}$$

where the energy of the scattered photon is given by

$$h\nu = \frac{h\nu}{1 + \frac{h\nu}{mc^2}(1 - cos\theta)}. \tag{7}$$

The differential solid angle is represented by $d\Omega$=$2\pi sin\theta d\theta$ assuming all azimuthal angles are equally probable. This cross-section can be expressed as cross-section based on polar angle θ given by the relation:

$$\frac{d\sigma}{d\theta} = \pi {r_0}^2\left[\frac{h\nu}{h\nu'} + \frac{h\nu'}{h\nu} - sin^2\theta\right] sin\theta. \tag{8}$$

The normalized differential cross-section for 200kev and 2 MeV photons is computed and the results are presented in Figures respectively.

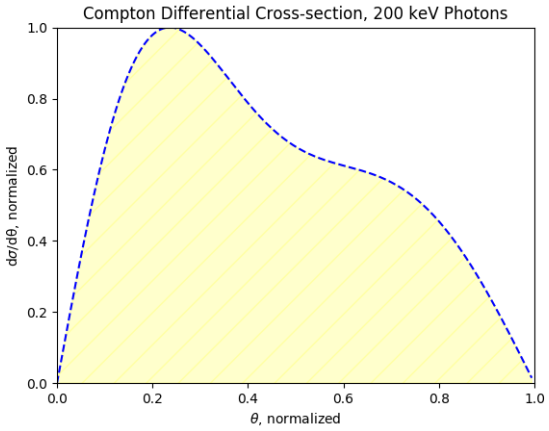

Figure 16. Plot of PDF for simulating the depth of interaction at 200Kev and 2 Mev Photons.

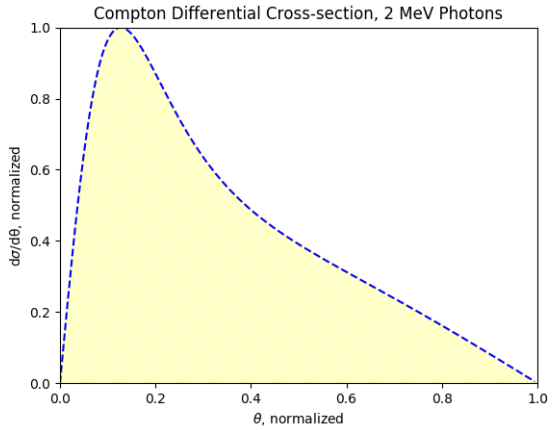

Figure 17. Plot of CPD for simulating the depth of interaction at 200Kev and 2 Mev Photons

Photon Trajetory Simulation

Using recurrence transformation formula for the scattered photon's direction of motion, expressed in coordinates of laboratory system is given by [3]:

$$cos\theta_{n+1} = sin\theta_s cos\phi_s sin\theta_n + cos\theta_s cos\theta_n \tag{9}$$

Also,

$$sin(\theta_{n+1} - \phi_n) = -\frac{sin\theta_s sin\phi_s}{sin\theta_{n+1}} \tag{10}$$

$$cos(\theta_{n+1} - \phi_n) = \frac{cos\theta_s - cos\theta_n cos\theta n + 1}{sin\theta_n sin\theta_{n+1}} \tag{11}$$

where: θ_n and ϕ_n are the polar and the azimuthal angles, respectively, before and the n-th scattering and θ_{n+1} and ϕ_{n+1} after. P_{n+1} is the path length travelled between the n-th and the (n+1)th interaction and the relationship between the spatial coordinates (x_n, y_n, x_n) and $(x_{n+1}, y_{n+1}, x_{n+1})$ of these interaction is given in the laboratory system by:

$$x_{n+1} = x_n + P_{n+1} sin\theta_{n+1} cos\theta_{n+1} \tag{12}$$

$$y_{n+1} = y_n + P_{n+1} sin\theta_{n+1} cos\theta_{n+1} \tag{13}$$

$$z_{n+1} = z_n + P_{n+1} cos\theta_{n+1} \tag{14}$$

where p_{n+1} is the sample step length. Azimuthal symmetry is taken into account and the new angle ϕ_{n+1} is sampled from the normal distribution$[0, 2\pi]$. The photons are generated randomly on the $10cm \times 10cm$ surface in the xy-plane. During the process, thus generated mean energy is then plotted as a function of distance in figures 18, 19.

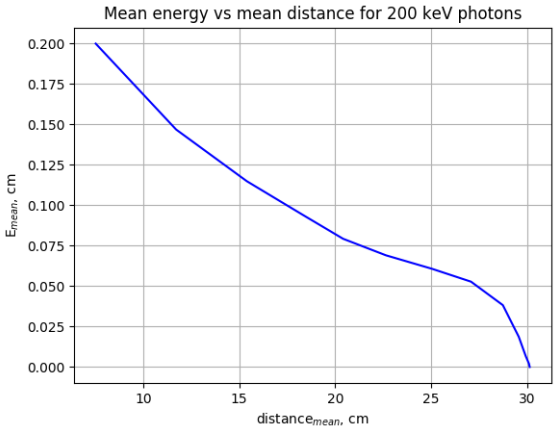

Figure 18. Mean energy of 200 KeV photon as a function of photons generating surface.

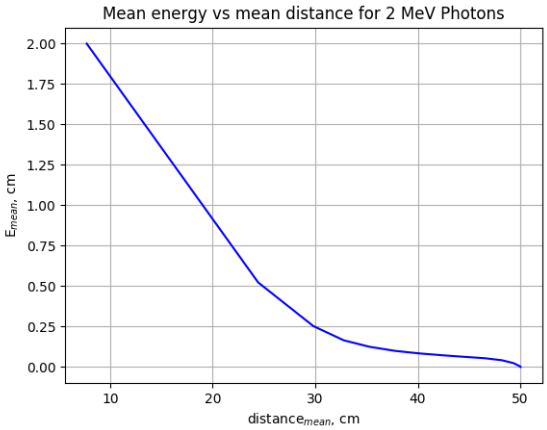

Figure 19. Mean energy of 2 MeV photon as a function of photons generating surface

Also, the x-y projection of the trajectories are illustrated in the figures 20, 21.

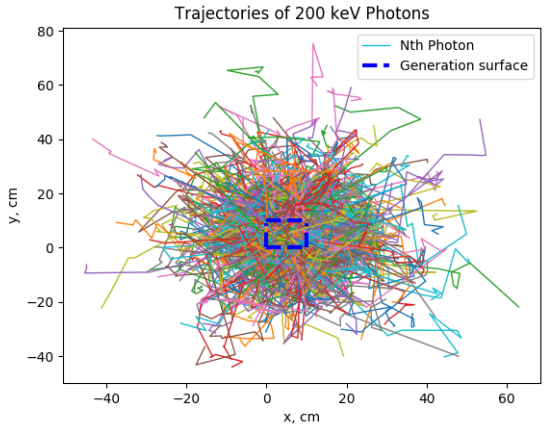

Figure 20. x-y projection of the trajectories for 1000 200 keV photons.

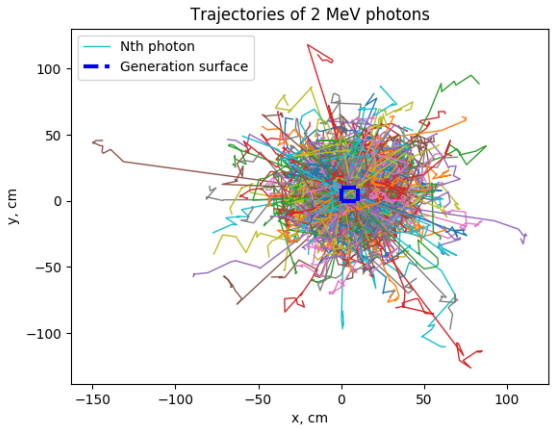

Figure 21. x-y projection of the trajectories for 1000 2 MeV photons.

V Conclusion

The Monte Carlo method is employed with high level programming language(python+numpy) to solve the above exercises. The random sampled are generated from the uniform and standard normal distribution. With in the same distribution function, 1000 particles were simulated with position and some energy at each steps for each particle. The histograms for both cases were generated and analyzed the results between two. The result is convincing in both cases that the distribution is centered to mean 0. For the second part of exercise, photons are simulated in water between the energy range of 200 keV to 2 MeV considering Compton scattering. The main goal is to calculate the attenuation coefficient functions implementing electronic cross-section for Compton scattering, finding its probability distribution, and sampling photons using inverse transform method. In addition to that, the maximum $d\sigma/d\theta$ was calculated numerically and normalized the function with this value. The final task was to track the photon trajectory using recurrence transformation formula.

References

[1] Christensen, R. (1987). The Dosimetry of Ionizing Radiation, Volume 1 edited by Kenneth R. Kase, Bengt E. Bjärngard, and F. H. Attix . Medical Physics, 14(3), 426–427. https://doi.org/10.1118/1.596062

[2] Attix, F. H. (1986). Introduction to Radiological Physics and Radiation Dosimetry. Introduction to Radiological Physics and Radiation Dosimetry.pg-132 https://doi.org/10.1002/9783527617135

[3] Persliden, J. (1983). A Monte Carlo program for photon transport using analogue sampling of scattering angle in coherent and incoherent scattering processes. Computer Programs in Biomedicine, 17(1–2), 115–128. https://doi.org/10.1016/0010-468X(83)90032-6

Programming Codes(Python+numpy)

Part I

1. Know your random number generator. Generate a list of 1000 random numbers sampled from a) a uniform distribution between 0 and 1 and b) a normal (Gaussian) distribution with a mean (μ) of zero and standard deviation(σ) of 1. Plot the distributions as histograms with a suitable bin size.Are the distributions as expected?

```
1
2 # example of effect of size on monte carlo sample
3 from numpy.random import normal
4 from matplotlib import pyplot
5 # define the distribution
6 mu = 0
7 sigma = 1
8 # generate monte carlo samples of differing size
9 sizes = [1000]
10 for i in range(len(sizes)):
11         # generate sample
12         sample = normal(mu, sigma, sizes[i])
13         # plot histogram of sample
14         #pyplot.subplot(2, 2, i+1)
15         pyplot.subplot(1, 1, i+1)
16         pyplot.hist(sample, bins=50)
17         pyplot.title('%d samples' % sizes[i])
18         pyplot.xticks([])
19 # show the plot
20 pyplot.show()
```

Figure 22. Python code for Exercise 1.

2. Random walk in 1dimension.A 'particle' is to move in one dimension where the step length Δs follows a normal distribution with μ =0and σ =1. Simulate 1000 particles with 100 steps('collisions')for each particle. Save the position at each collision for each particle. Plot the position(y) vs the collision number(x) for 10 arbitrary particles.Plot the full 1D position distribution(for all particles) as a histogram.Is the distribution as expected, qualitatively speaking?

```
1
2
3 # example making a random collision and showing that the distribution after a number of
4 # steps is a gaussian
5 import numpy as np
6 #matplotlib inline
7 import matplotlib.pyplot as plt
8 from scipy import stats # has lots of distribution functions in it
9 from math import erfc   # complimentary error function
10
11 # randomly walk nsteps and return the x value
12 # starting at x=0
13 # each step has zero mean and a variance of 1
14 def walkn(nsteps):  # random walk using a normal distribution for step sizes
15     r = stats.norm.rvs(size=nsteps)  # normal distribution mean=0 sigma =1
16     # r is a vector values randomly generated with a normal distribution
17     return sum(r)  # the sum of the entire vector!
18
19 # colide npart numbers of particles  nsteps and return a vector of x positions
20 # the function that gives us a randomly generated position is walkn
21 def npart_walkn(npart,nsteps):
22     xvec = np.zeros(0)
23     for i in range(npart):
24         x = walkn(nsteps)  # a single random collision value
25         xvec = np.append(xvec,x)  # append each random collision to the vector
26     return xvec
27
28 nsteps = 100 # number of steps
29 npart = 1000 # number of particles
30 # fill a vector with npart collides each with n nsteps
31 xvec = npart_walkn(npart,nsteps)
32 # plot the histogram, i.e., measured distribution of final positions
33 #   after n steps of collisions
34 n, bins, patches =plt.hist(xvec,bins='auto', density=True)
35 # this retuns as n the number of values counted, bins is the x values of the bins,
36 # patches is for plotting the histogram
37 plt.xlabel("Collision Number",fontsize=16)
38 plt.ylabel("Position",fontsize=16)  # probability!
39
40 # a gaussian probability density distribution, this is a function!
41 mygaus = stats.norm(0.0, np.sqrt(nsteps))  # should scale with sqrt(nsteps)
42 y = mygaus.pdf(bins)  # evaluate the function at the bin locations
43 plt.plot(bins,y,"k", lw=2 )  #plot the expected density distribution as a black line
44 plt.show()
```

Figure 23. Python code for Exercise 2.

Codes for exercise 3 is simple modifications of exercise 2.

Part II

The following simulations are for photons in water. The number of electrons per volume unit in water is $n_V = 3.431023\ cm^{-3}$. We assume that the only interaction taking place is Compton scattering. The minimum and maximum photon energy is 50 and 2000 keV, respectively.

4.Create attenuation coefficient function. Implement the electronic cross section for Compton scattering in your code (define it as an energy-dependent function) using eq. 7.15 in Attix. Express σ in $[cm^2]$.Compare with values in figure 7.6 in Attix. Calculate the attenuation coefficient μ (again as a function) from σ and n_V. Make a plot of μ (in $[cm^{-1}]$) as a function of photon energy. Calculate explicitly values of μ for200 keV and 2

MeV photons. Compare to published values of $0.137cm^{-1}$and $0.049cm^{-1}$, respectively. If you find large differences($>$ 10 percentage), you may want to check your code before you proceed....!

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import math
from statistics import stdev
#---------------------------------- Task 4 -------------------------------------

nv = 3.43e23

Emax = 2000
Emin = 50

x = np.logspace(-2, 3, 100)

alpha = x/0.511

sigma_e = 2*math.pi*(2.818e-13)**2*((1+alpha)/(alpha**2)*(2*(1+alpha)/(1+2*alpha)-np.log(1+2*alpha)/
alpha)+np.log(1+2*alpha)/alpha/2-(1+3*alpha)/((1+2*alpha)**2))

sigma_O_mass = 6.022e23*8/16*sigma_e
sigma_H_mass = 6.022e23*1/1*sigma_e

sigma_tot = 2/18*sigma_H_mass+16/18*sigma_O_mass


plt.plot(x,sigma_e,'b--',label='Electronic cross-section')
plt.xlabel('Energy, MeV')
plt.ylabel('$\sigma_e$, cm$^2$')
plt.title('Electronic Cross-section for Compton Scattering')
plt.xscale("log")
plt.yscale("log")
#plt.grid(True)
plt.xlim(0.01,500)
plt.ylim(1e-27,1e-24)
plt.legend()
plt.show()

plt.plot(x,sigma_tot,'b--',label='Attenuation coefficient $\mu$')
plt.xlabel('Energy, MeV')
plt.ylabel('$\mu$, cm$^{-1}$')
plt.title('Attenuation coefficient, Compton scattering, H$_2$O')
plt.xscale("log")
plt.yscale("log")
plt.plot([2,2],[0.0001,0.049],"--",color="cadetblue", linewidth=1)
plt.plot([0.0001,2],[0.049,0.049],"--",color="cadetblue", linewidth=1)
plt.text(2.2, 0.002, r'2 MeV photons', fontsize=13,color="cadetblue")
plt.plot([0.2,0.2],[0.0001,0.137],"--",color="indigo", linewidth=1)
plt.plot([0.0001,0.2],[0.137,0.137],"--",color="indigo", linewidth=1)
plt.text(0.22, 0.004, r'200 keV photons', fontsize=13,color="indigo")
#plt.grid(True)
plt.xlim(0.01,500)
plt.ylim(0.0001,0.3)
plt.legend()
plt.show()
```

Figure 24. Python code for plotting the μ value as a function of photon energy

5 Probability distributions. Make a probability distribution function (PDF) and cumulative distribution function (CPD) for simulating the depth of interaction. Use μ defined under 4. Plot the PDF and CPD for 200 keV and 2 MeV photons.

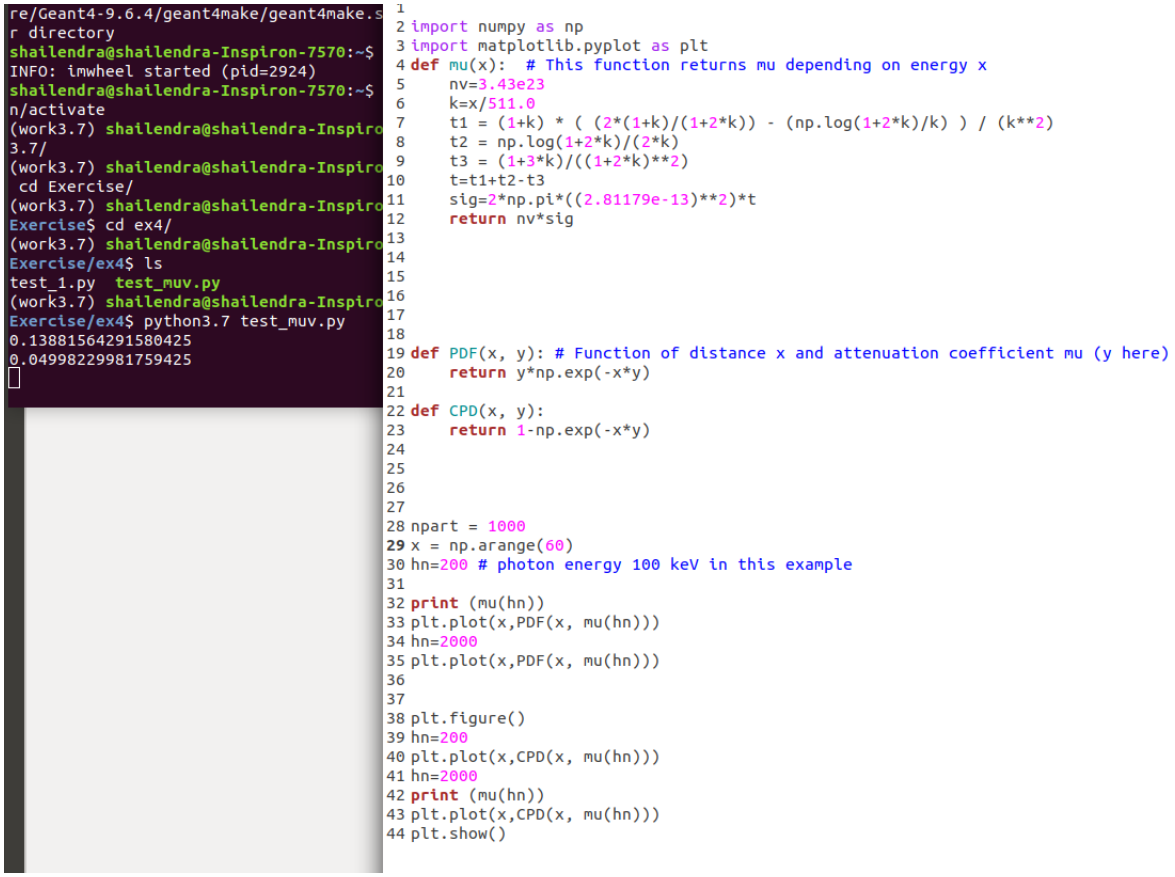

Figure 25. Modification of code 4 for the calculation of μ value. The same code employs for developing the plots of PDF and CPDF in exercise 5.

6 Sampling photon steplengths using the inverse transform. Sample photon steplength Δs using the inverse transform method for 200 keV and 2 MeV photons. Use the μ function from above. Simulate 1000 photons. Present the data as histograms and compare with respective PDFs.Are the computational (MC) and analytical ($e^{-\mu x}$)depth distributions similar? Also calculate mean MC steplength and compare to mean free path for photons. [Additional task for those with particular drive/interest: sample photon steplengths using the rejection technique(see also 8.) and compare with results from inverse transform.]

```python
import numpy as np
import matplotlib.pyplot as plt

def mu(x):
    nv=3.43e23
    k=x/511.0
    t1 = (1+k) * ( (2*(1+k)/(1+2*k)) - (np.log(1+2*k)/k) ) / (k**2)
    t2 = np.log(1+2*k)/(2*k)
    t3 = (1+3*k)/((1+2*k)**2)
    t=t1+t2-t3
    sig=2*np.pi*((2.81179e-13)**2)*t
    return nv*sig

def iCPD(x, y):
    ran=np.random.rand(y)
    return -np.log(1-ran)/mu(x)

def PDF(x, y):
    return y*np.exp(-x*y)

hn=2000 # in keV
npart=1000 # number of photons
vec=iCPD(hn, npart)

nbin=30
hist=np.histogram(vec, bins=nbin)

yhist=hist[0]/np.sum(hist[0])
xhist=hist[1]
x1=xhist[0:nbin]
x2=xhist[1:nbin+1]
xhist=(x1+x2)/2.0
wid=(xhist[nbin-1]-xhist[0])/(1.25*nbin)
plt.figure()
plt.bar(xhist, yhist, width=wid)

y=PDF(xhist, mu(hn))/np.sum(PDF(xhist, mu(hn))) # data must be normalized to unit area
plt.plot(xhist,y, 'r-')
print(np.mean(vec))
print(1/mu(hn))
plt.show()


#For Mean and 1/mu value 200 keV
#7.401194524856777
#7.203799074766589



#For Mean and 1/mu value 2 MeV
#21.018238589602923
#20.007082580221535


```

Figure 26. Python code for Exercise 6.

7. Normalized differential scattering cross section. Define the differential cross section $\frac{d\sigma}{d\theta}$ for Compton scattering (eq. 7.13) as a function. Remember to include the 'sin '-term from the solid angle element (slide 40, MC lecture). Further hint: employ Compton's formula (slide 39, $h\nu'$ ' = ...) directly into this implementa-

tion, as it links the scattered photon energy to the scattering angle. Find the maximum of $\frac{d\sigma}{d\theta}$ numerically and normalize the function with this value. Plot the normalized $\frac{d\sigma}{d\theta}$ as a function of for 200 keV and 2 MeV photons. For rest of the exercises the codes are simple modifications of these codes. So, they are not mention here.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import math
from statistics import stdev

#------------------------------------------ Task 7 ------------------------------------------
theta = np.arange(180)

Einit = 0.2

Escat = Einit/(1+Einit/0.511*(1-np.cos(theta*math.pi/180)))

dsigma_dtheta = math.pi*(2.818e-13)**2*(Escat/Einit)**2*(Einit/Escat+Escat/Einit-np.sin(theta*math.pi/180)**2)*np.sin(theta*math.pi/180)

plt.plot(theta/180,dsigma_dtheta/max(dsigma_dtheta),'--',color="blue",label='Normalization of Compton cross section')
plt.fill_between(theta/180,dsigma_dtheta/max(dsigma_dtheta), color='yellow', alpha=0.2, hatch='/')
plt.xlabel('$\u03B8$, normalized')
plt.ylabel('d$\sigma$/d\u03B8, normalized')
plt.title('Compton Differential Cross-section, 200 keV Photons')
plt.xlim(0,1)
plt.ylim(0,1)
plt.show()

print(max(dsigma_dtheta))

```

Figure 27. Python code for Exercise 7.

9. Photon trajectory simulation. Use the 'recurrence transformation formula' (see paper by Persliden or MC lecture notes slide 44-45) to simulate photon trajectories. Employ code you have developed above. Remember to also include sampling of the azimuthal angle. Save x, y and z position and energy for each photon and interaction. Plot example trajectories (2D or 3D), e.g. for 100 photons. Estimate the mean photon energy as a function of depth z (employ more than 1000 photons).Perform for both 200 keV and 2 MeV photons.