TFY4235 Numerical Physics Assignment 3

Jostein Kløgetvedt

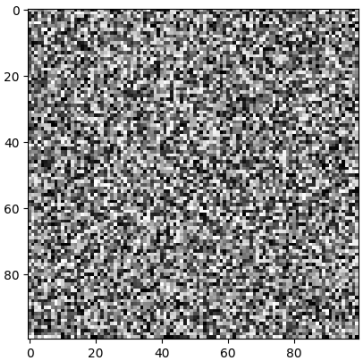
April 2022

Pseudo random number generator (PRNG)

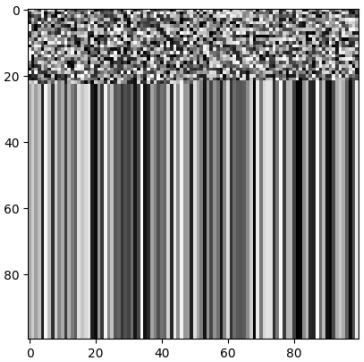
The two PRNG's that we shall investigate are the middle square method and the linear congruential method. The middle square method (MS) is generally a 'bad' generator since it often gets stuck on a number or end up in a short-cycled loop. This number may be a number other than zero, for instance 2500 and 100 for $n = 4$ as the n-digit input seed. To illustrate what we mean by a short-cycled loop will the seed 6758 after 69 steps end up in a loop where it alternates between the numbers 6100, 2100, 4100 and 8100. In order to generate a large sequence of pseudo-random numbers should one use a large seed since this often results in a higher period before the sequence starts repeating and the 'randomness' is broken. But also in this case does it often happen that the sequence ends in a fast convergence and it is generally difficult to handpick specific seeds that produce large periodic sequences. To illustrate this further will an input seed of $s_1 = 67348210$ produce a sequence with period longer than 1×10^4 while the seed $s_2 = 67348212$ ends in a short-cycled loop after 2000 numbers. Figure 1a and 1b show the diversity of the random numbers gathered with s_1 and s_2 by representing each number as a gray-scaled pixel in an image. It was in fact difficult to find a seed with so long period by only using a 8-digit input seed.

The linear congruential (LCG) method will, in contrast to the middle square method, not converge to a single number and does often have a larger period. The LCG method is in general very sensitive to the input parameters a, c and m and will in some cases simply not produce a pseudo-random sequence. For instance will $a = 1$ and $c = 1$ produce a modulo counter which is not random. The method is, however, not so sensitive to the seed value. If the parameters are properly chosen the period can have maximal length m, traversing all integers from 0 to $m - 1$ in a random sequence. To illustrate the sensitivity of the parameters will a value of $m_1 = 85726$ produce a long periodic sequence while $m_2 = 85722$ has a period of 12. Here, a=1234, c =342 and the seed is 8643. Figure 1c and 1d show the pixel-image for the different m's.

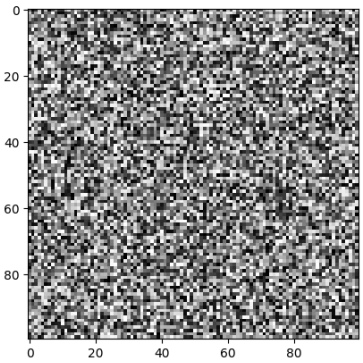
The LCG methods has some weaknesses. It was found that k-dimensional random numbers taken from this method does not fill the entire $k-$dimensional space, but rather lie on $(k-1)$-dimensional planes [1]. Figure 2b illustrates this where $k = 3$-dimensional numbers are plotted. Here the parameters are as before, with $m = m_1$. In comparison, figure 2a shows random points from the MS-algorithm, here with the seed s_2. A total of 6×10^4 numbers are drawn from the acquired sequence, which is high above the well-functioning interval for this MS-sequence ($\sim$2000). Here it is clear that the sequence of numbers does not fill the space at all since the sequence ends in an short-cycled loop as seen by 1b.



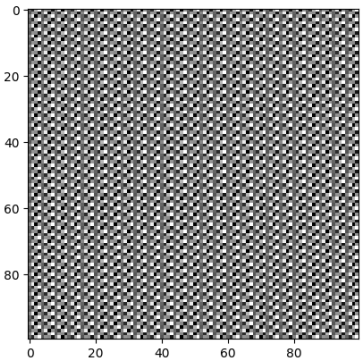
(a) MS $s_1 = 67348210$



(b) MS $s_2 = 67348212$

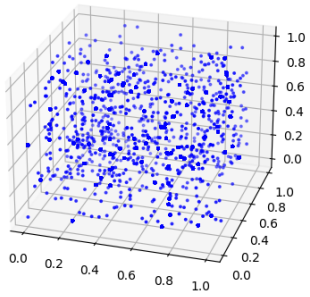


(c) LCG $m_1 = 85726$

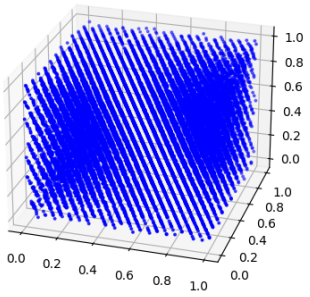


(d) LCG $m_2 = 85722$

Figure 1: Gray-scaled image of the sequences produced with MS and LCG for different parameters. The other parameters for LCG are a=1234, c =342 and the seed is 8643.



(a) MS $s_2 = 67348212$



(b) LCG $m_1 = 85726$

Figure 2: The figures show a 3d-plot of random numbers from the MS- and LCG-algorithm. For the MS, a seed of s_2 =67348212 is used while the LCG has parameters a =1234, c =342, m_1 =85726 and the seed 8643 is used.

Now, we compare the two algorithms with the built-in function *numpy.random.rand* used from the *Numpy* library in Python. A seed of 42 is used throughout this project. In order to compare do

we scale the random sequences such that every number is between 0 and 1 and use *numpy.histogram* such that we count the number of random values that lie within a certain range. The used parameters are as described above, with s_1 and m_1 for the MS- and LCG-algorithm respectively. In order to avoid fast convergence and an invalid sequence for the MS-algorithm is the number of random values chosen to be 1×10^4, which is below the estimated period. Figure 3 show the distributions of the sequences acquired from MS, LCG and Numpy. The plot show a uniform distribution with some random noise, which is what we expect. The three algorithms produce very similar results and it is in fact difficult to separate them. The mean value is 1 and the standard deviation is $\sim$0.1 for all three cases with very small deviations.

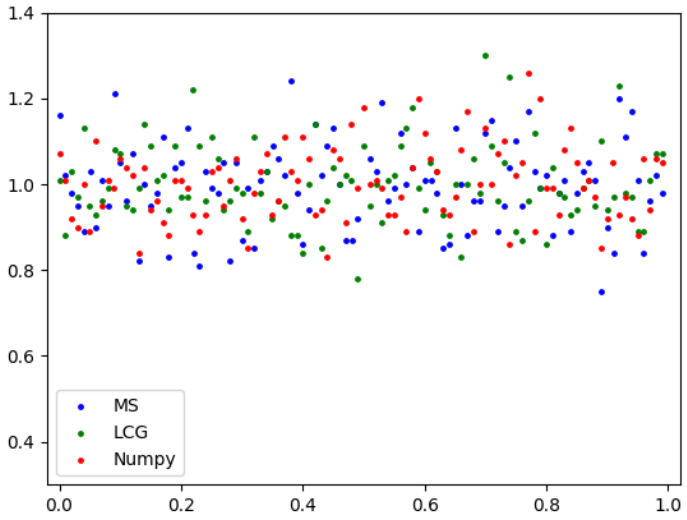
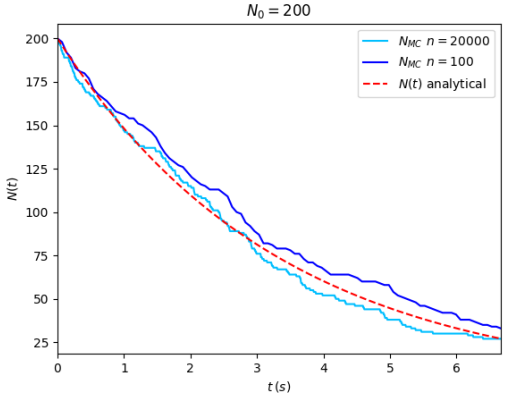


Figure 3: Distributions of the random sequences acquired from MS, with s_1, LCG, with m_1, and the built in function from Numpy with seed 42. The other parameters for LCG are a=1234, c =342 and the seed is 8643.

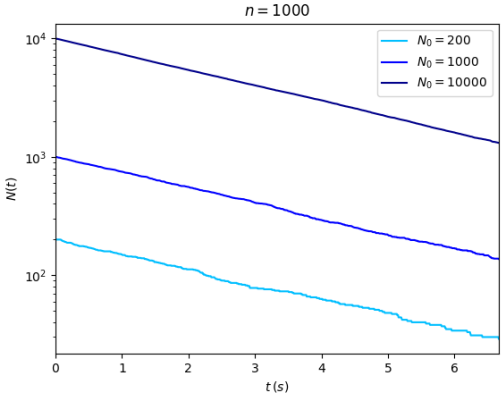
In conclusion may we say that the MS-algorithm is hard to work with since only a few seed's are appropriate and the period is generally short. LCG works better, but has a few shortcomings aswell. It may produce a very short sequence in some cases and only lie on $(k-1)$-dimensional planes. A high m reduces this error, in fact, a value of 2^{32} in combination with a large a and c is often used [1].

Nuclear decay

Nuclear decay is determined by the exponential law of decay, expressed as $N(t) = N_0 e^{-\lambda t}$, where the decay constant is given the value $\lambda = 0.3/\text{s}$. The mean life is thus $\tau = 3.3\text{s}$ and the half-life is $t_{1/2} = 2.3\text{s}$. In order to simulate do we need to discretize the $t-$axis like $t_k = k\Delta t$ where k is an integer and Δt is the step length. The axis stops at the end-time $T = n\Delta t$ where n is the number of discretized points along the grid. The nuclear decay is simulated with a Monte-Carlo (MC) algorithm where the probability of decay at each time-step is given by $P_{decay} = 1 - e^{-\lambda \Delta t}$. Figure 4 shows the obtained values from the MC-simulation to the end-time $T = 2\tau$. Figure 4a has initial value $N_0 = 200$ and the plot is shown for $n = 100$ and $n = 2 \times 10^4$ points along the time grid. The curves follow the analytical value $N(t)$, but contains some noise. The agreement is slightly better for the high resolution case. Now, keeping n=1000 constant and increasing the initial value to N_0 =200, 1000 and 1×10^4, we see from figure 4b better agreements for higher N_0. The analytical expression is an exponential function so we expect a straight line when the axes are scaled logarithmically, which is what we observe for the large initial value case. This might suggest that it is more efficient to increase N_0 rather than n to achieve lower deviation from the analytical curve.



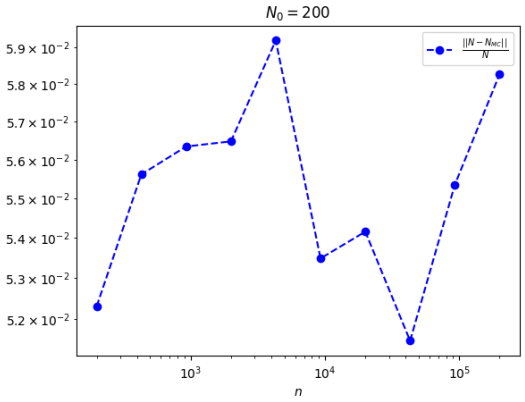
(a) MC result with low, $n = 100$, and high, $n = 2 \times 10^4$ resolution along the $t-$axis. The analytical function $N(t)$ is shown in red.



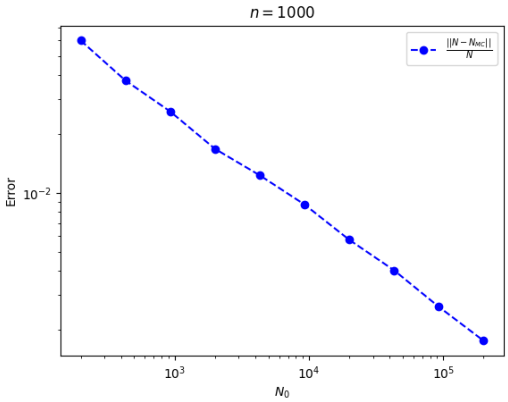
(b) MC result with initial values N_0 =200, 1000 and 10000. Used $n = 1000$ resolution along the $t-$axis. The $y-$axis is scaled logarithmically.

Figure 4: Decay of particle with $\lambda = 0.3/\text{s}$ till $T = 2\tau$.

We can quantify this observation further. Figure 5 show the relative error, calculated as $\frac{||N_{an} - N_{MC}||_F}{||N_{an}||_F}$, as a function of n and N_0. Here, N_{an} is a vector of the analytical particle decay at all time points and the same values of $\lambda = 0.3/\text{s}$ and $T = 2\tau$ is used. The shown results are the averaged values over 100 simulations per point. We see that increasing n does not yield better estimates while increasing N_0 slowly decreases the error.



(a) Averaged error as a function n.



(b) Averaged error as a function of N_0.

Figure 5: Plot of deviation from the analytical $N(t)$ keeping N_0 and n fixed separately. The end time is $T = 2\tau$.

Continuing, we may also investigate how the relative error changes as a function of time. In this case is the error only calculated at the end time T. We set T to increase from $0.1 \cdot \text{t}_{1/2}$ to $6 \cdot t_{1/2}$. This means that, given $N_0 = 2000$, the number of particles remaining after six half-lives is 31. Figure 6 shows the averaged results by doing the simulation 100 times for each T. A number of $n = 500$ points is used along the time-grid. The figure suggests that the error increases with T.

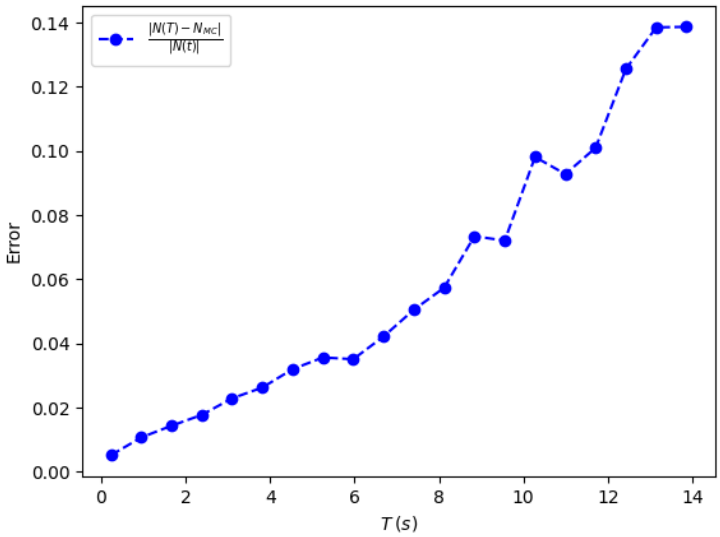


Figure 6: Relative error as a function of end time T. The initial value is $N_0 = 2000$ and a number of $n = 500$ points is used.

Figure 7 illustrates the distribution of N_{MC}-values obtained for a given time. The time is chosen to be the end-time $t = T = \tau$. The initial value is $N_0 = 5000$ and the resolution is $n = 500$ points. The distribution is found by performing the MC-simulation 5000 times and distributing the values with a histogram. We can clearly see that the distribution is centered around the analytical $N(\tau)$ value and has form of a Gaussian distribution. Taking the mean value of the N_{MC}-values we obtain a deviation of 2.6×10^{-4}. The standard deviation of the distribution is $\sigma = 33.7$.

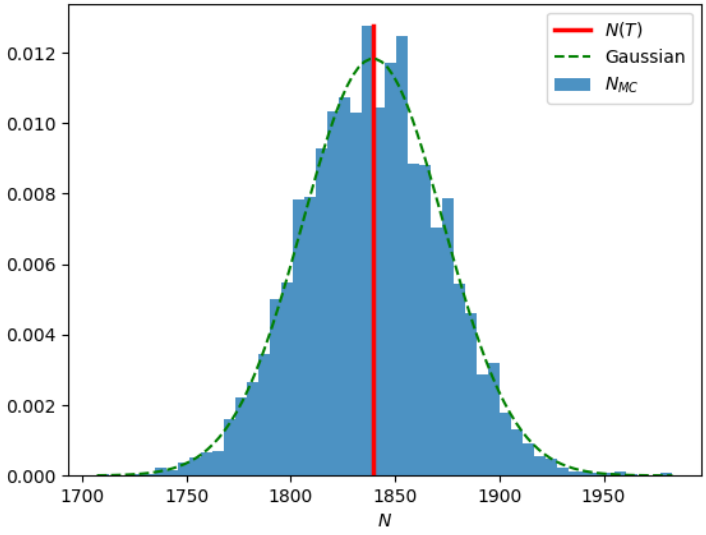


Figure 7: Distribution of N_{MC} gathered by performing the MC-decay 5000 times till the end-time $T = \tau$. The initial value is $N_0 = 5000$ and $n = 500$ points are used. The red line represents the analytical value $N(\tau)$ and the green dashed line is a Gaussian distribution with mean and standard deviation as for N_{MC}.

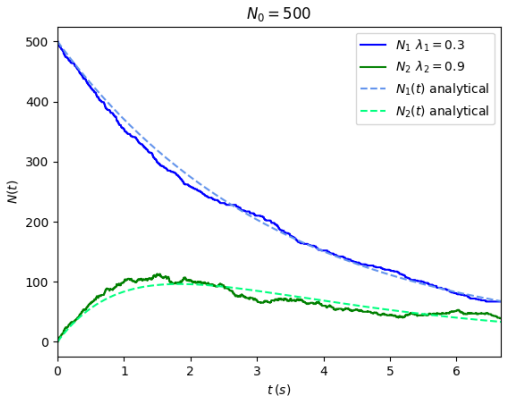
Secular equilibrium

We shall here investigate features of a 3-step decay chain with initial values $N_1(0) = N_0$ and $N_2(0) = 0$. We still consider the case where $\lambda_1 = 0.3/\text{s}$ and use $n = 4000$ points along the time-grid. To illustrate how the decay will look like for $N_1(t)$ and $N_2(t)$ as a function of time does figure 8a show the acquired results together with the analytical values for each particle type. The father-particle decays in a similar fashion as described above, but the $N_2(t)$ has a somewhat more complicated expression. The decay is given by

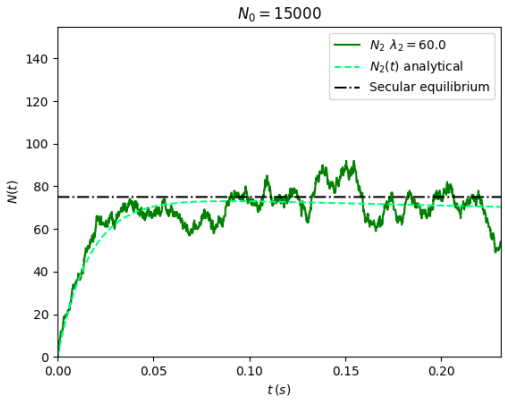
$$N_2(t) = \frac{\lambda_1 N_0}{\lambda_2 - \lambda_1} \left(e^{-\lambda_1 t} - e^{-\lambda_2 t} \right). \tag{1}$$

Here, the decay constant of the second particle is set to be $\lambda_2 = 3\lambda_1$ and we start with $N_0 = 500$ particles. There seems to be a good correlation between the analytical and the computed values.

For a particular high decay constant, $\lambda_2 \gg \lambda_1$, can the second particle in the decay chain reach a secular equilibrium. This happens after a long time, in the timescale of particle two, and the given value of N_2 in equilibrium is $N_2(t) = N_0 \frac{\lambda_1}{\lambda_2}$. Figure 8b shows an illustration of this phenomenon. Here we start out with $N_0 = 15000$ and have used $\lambda_2 = 200\lambda_1$ to satisfy the criterion. The mean life of particle two is thus $\tau = 0.017$s. The simulation is run to $T = 20t_{1/2}$ (half-life of particle two), which shows that the equilibrium is reached after a long time in particle two's timescale. Even though the analytical curve does not overlap the predicted secular equilibrium line, it is sufficiently close and can be regarded as an equilibrium since the curve is almost flat.



(a) Decay of N_1 and N_2 as a function of time.

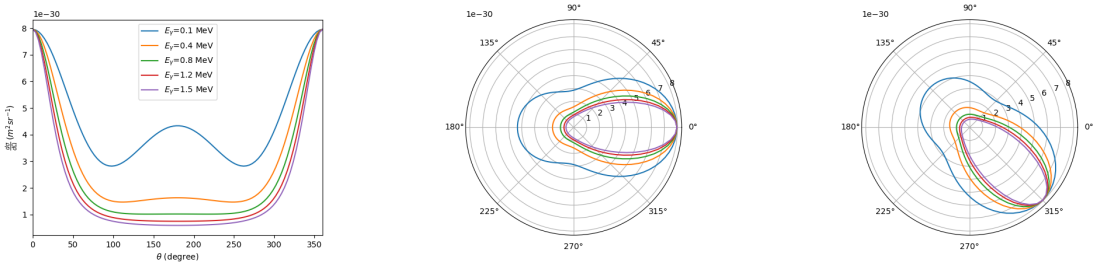


(b) Decay of N_2 with large decay constant.

Figure 8: Plot of the remaining number of particles in a 3-step decay chain. The resolution along the time-grid is $n = 4000$. The analytical curves are shown in comparison. A secular equilibrium is reached in 8b.

Klein-Nishina equation

During Compton scattering, where a photon transfers a portion of its energy to an electron, the differential scattering cross-section is described with the Klein-Nishina equation. The differential scattering cross-section $\frac{d\sigma}{d\Omega}$ is a function of both the incoming photon energy E_γ and the deflection angle θ. The deflection angle is the difference between the incoming direction and the outgoing, $\theta = \theta_1 - \theta_0$. Figure 9 shows how the differential scattering cross-section varies as a function of θ for different incoming energies. The direction of the incoming photon is set to be zero. If the incoming angle differs from zero, like in 9c where $\theta_0 = -\frac{\pi}{4}$, will the resulting differential scattering cross-section be rotated accordingly.



(a) Cartesian plot, $\theta_0 = 0$. (b) Polar plot, $\theta_0 = 0$. (c) Polar plot, $\theta_0 = -\frac{\pi}{4}$.

Figure 9: Plot of the differential cross-section as a function of the deflection angle according to the Klein-Nishina equation with different energies of the incoming photon.

The differential scattering cross-section may be regarded as a probability distribution for the deflection angle θ given E_γ. The probability distribution is made with the inverse CDF (cumulative distribution function) method. In particular, we want to map a uniform distribution to the Klein-Nishina distribution. For a given photon energy, $P(\theta) = \frac{d\sigma(\theta)}{d\Omega}$, $\theta \in [0, 2\pi]$ and the CDF is $F(\theta) = \int_0^\theta P(\theta)d\theta$. It can be shown that a distribution for θ is $\theta \sim F^{-1}(X)$, where $X \sim \text{Unif}[0, 1]$ [4]. In the implementation, the CDF is found by numerical integration and the inverse is found by with interpolation, *scipy.interpolate.interp1d*, using linear interpolation between a high resolution grid. Once the inverse CDF is found, a sample of randomly distributed angles is found by inserting uniformly distributed values, generated with *numpy.random.rand*, into the function. Figure 10 shows how the computed values is located relative to the Klein-Nishina distribution. A number of 1×10^5 datapoints are drawn from the made distribution. Each blue point represents the height of a bar in a histogram, scaled down to fit with the analytical curve. There seems to be a good agreement with the computed and the analytical values.

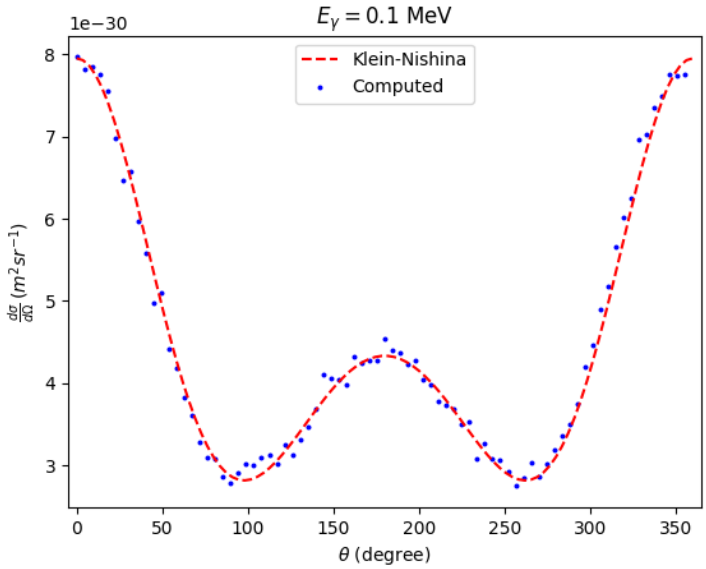


Figure 10: Illustration of a sample of 1×10^5 datapoints drawn from the Klein-Nishina probability distribution in comparison with the analytical Klein-Nishina distribution.

Photon transport in a medium

To simulate nuclear decay and hence photon transport in a medium do we make use of a gamma ray decay spectrum. A photon starts out with an initial energy E_0, which is one of the values $[0.135, 0.525, 0.615]$ MeV, with corresponding probabilities $p = [5\%, 80\%, 15\%]$. For each iteration

is the photon either scattered by Compton scattering or absorbed by the photoelectric effect. The probability to be absorbed by the photoelectric effect is given by

$$\begin{aligned} P_{absorbed} &= 1 - \exp\left(-\sigma_{PE}(E_\gamma, Z) \cdot l\right) \\ \text{where } \sigma_{PE}(E_\gamma, Z) &= 3 \times 10^{12} \frac{Z^4}{E_\gamma^{3.5}}. \end{aligned} \tag{2}$$

Here, E_γ is the photon energy, Z is the atomic number of the medium and l is the step-length the photon moves by for each iteration. σ_{PE} is an approximated expression for the cross-section for low energies [2]. We shall in this report consider a light, Z=3 (lithium), and heavy, Z=82 (lead), medium. If the photon undergoes Compton scattering will the energy be reduced and the direction may be altered. The energy is reduced by

$$E'_\gamma = \frac{E_\gamma}{1 + k(1 - \cos(\theta))}, \tag{3}$$

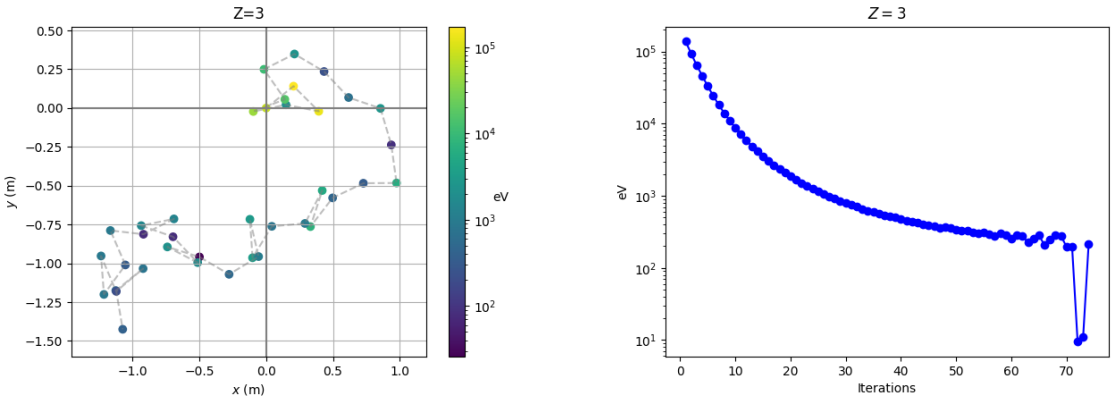
where θ is the deflection angle and k is the ratio between the incoming photon energy and the rest mass energy of the electron. The new direction is given by the Klein-Nishina differential cross-section, as mentioned earlier, which is dependent on the incoming energy of the photon as well as the incoming angle. Since it is computationally expensive to calculate a new inverse CDF for each energy and photon, do we make a database with different CDF's for different energies. From figure 9a do we see that the distribution changes mostly for energies of the order 1×10^5eV. Since the highest initial energy is $E_{0_3} = 0.615$MeV do we consider an array of energies in the range $[0.01, 0.615]$MeV and make an inverse CDF for each energy. For lower energies does the distribution remain constant and looks like a $1 - \sin^2(\theta)$ distribution. An incoming photon will thus use the probability distribution that has lowest difference between the incoming energy and the energy in the array.

The step-length l is chosen as $l = 1/\mu$ where μ is the attenuation coefficient in the given material. The attenuation coefficient depends on the photon energy so a different μ is used depending on Z and the initial energy of the photon E_0. Table 1 shows the used values for the attenuation coefficients corresponding to each initial energy E_{0_1}, E_{0_2} and E_{0_3} from the gamma decay spectrum. The obtained values are found in [3].

Table 1: Attenuation coefficients used in the simulation.

	μ_1 (m^{-1})	μ_2 (m^{-1})	μ_3 (m^{-1})
Z =3	6.883	4.022	3.577
Z =82	6264	182.2	140.9

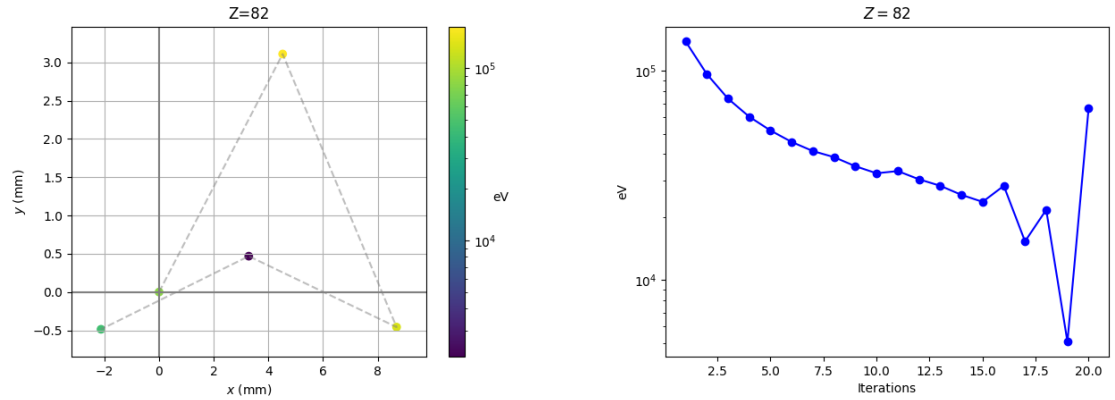
We start by looking at a simplified model where the step-length is constant, $l = 1/\mu$, and each photon starts at origin with the initial angle $\theta_0 = 0$. We aggregate the results from a number of $N_0 = 1 \times 10^5$ number of particles for both $Z = 3$ and $Z = 82$. Figure 11a shows the trajectory for a single photon propagating in lithium. The colors indicate the energy lost at each location. The energy lost is given by $E_{lost} = E_\gamma - E'_\gamma$. Figure 11b better illustrates how the energy lost is reduced exponentially for each collision with the medium. The plot shows the mean result from all the photons. As we can see, most of the energy is lost at the very beginning. The trajectory does not give us useful information other than that the photon might have a higher tendency to experience back- and sideways scattering for lower energies so the trajectory often ends with the photon being scattered in all directions back and forth in a small bubble. We note that the photon has incoming angle $\theta_0 = 0$ and that the first collision occurs at origin. Looking at statistical data, the mean distance travelled from emission to absorption is $r = 1.42$m with standard deviation $\sigma = 0.82$m. The mean $x-$coordinate of the absorption position of all photons is $x_{last} = 0.21$m while the $y_{last}-$coordinate is of the order 10^{-3}, i.e. approximately zero. These results originates from the fact that the initial angle is zero and that forward scattering occurs more frequently, especially for high energies.



(a) Trajectory of a single photon.

(b) Mean energy lost at each collision.

Figure 11: Plot of the trajectory and mean energy lost for a number of 1×10^5 photons in lithium.



(a) Trajectory of a single photon.

(b) Mean energy lost at each collision.

Figure 12: Plot of the trajectory and mean energy lost for a number of 1×10^5 photons in lead.

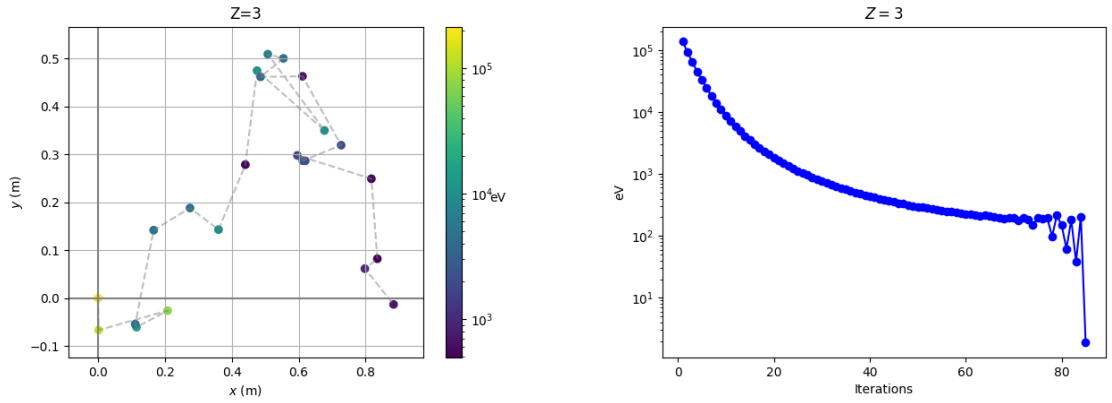
Figure 12 shows the results where lead is the given medium that the photons propagate through. The trajectory is similar to the first five points in figure 11a, only with smaller length scales. The same random sequence is used so the only thing that changes are the distances and that the absorption happens earlier. The mean distance travelled from emission to absorption is thus much lower. The computed value is $r = 1.3$cm with standard deviation $\sigma = 0.97$cm. The reason for this is that the step-length, $l = 1/\mu$, is in general much lower due to the high attenuation coefficient. The mean absorption position is also in this case such that x_{last} is much higher than y_{last} due to the initial angle being zero. The reason for why the absorption occurs earlier, i.e. the number of iterations decrease, is due to the fact that the photoelectric cross-section increases much faster with $Z = 82$ than for lower Z and thus will the photons in general be absorbed for a lower number of iterations. In other words, the photoelectric effect occurs more frequently among heavy atoms. Comparing the energy-lost plots can we also see that the lead case is an early cut-off of the lithium plot.

In this case, where the initial angle is set to zero, can we compare the simulation to the physical case where a beam of light penetrates into the medium. Light intensity is reduced according to Beer-Lambert's law $I(z) = I_0 e^{-\mu z}$. The mean distance is thus the inverse of the attenuation coefficient. In our case, where we use different μ's, can we compute the mean attenuation coefficient with their respective probabilities. The mean distance computed for lithium is $l_{lithium} = 0.24$m. This is close to the obtained value $x_{last} = 0.21$m. We can do the same calculations for lead and find a mean distance of $l_{lead} = 5.5$mm while $x_{last} = 4.8$mm. The attenuation coefficients used correspond to the initial energy, while the energy actually decreases during the transportation. Thus, the mean

distance calculated is in fact too high. A better model would perhaps use the attenuation coefficient as a function of the energy $\mu(E_\gamma)$, and hence l, to adjust this error.

We note that our discussion about average length travelled before absorption solely relies on our chosen parameter l. In the simulation, the photons will, on average, perform the same number of iterations because they are starting with the same initial energy, so the distance travelled is determined by l. In the literature, [5], it is determined that if the incoming photons have energy 500keV then a lead piece with thickness 1.35cm would yield 10% transmission. A similar calculation was done where we found $x_{lead} = 1.71$cm to be the mean position among the remaining, not yet absorbed photons, when there were 10% photons left. The acquired result seems to be a good fit considering the crude approximations done, with the step-length being constant and that either a Compton or a photoelectric interaction occur at every step. The physical value is lower due to other scattering processes that occurs, like pair- and triplet production. Thus, to obtain better estimates that fit with physical values does the step-length l have to be reduced.

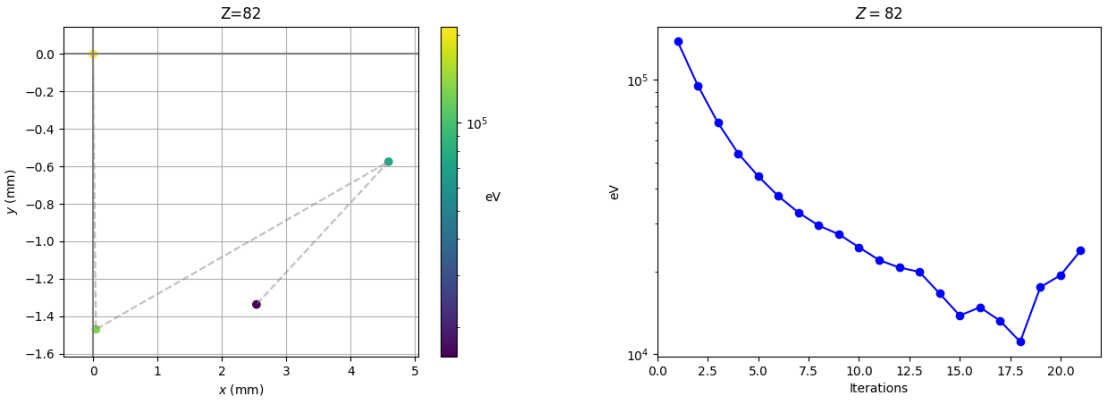
To better simulate photon transport due to nuclear decay do we need to consider the case where the initial angle is selected randomly. We generate the initial angle from a uniform distribution in the interval $[0, 2\pi]$. The step-size is also altered, and takes the new value for each iteration, $l = l_0(1 - \eta)$ where η is a random number between zero and one and $l_0 = 1/\mu$ is the previously used step-size. These modifications should be a better fit to reality since the length between each Compton scattering is in fact not constant and is to some extent a random variable. Performing the same calculations as before we obtain figure 13 and 14. Both the energy-lost plot and the trajectories have similar structures as before. Again can we see that the lead-trajectory is an early cut-off of the lithium plots, but with different lengths. The statistical data acquired from using the same number N_0 of photons is $r = 0.85$m with standard deviation $\sigma = 0.48$m for lithium, and $r = 7.3$mm with $\sigma = 5.6$mm for lead. In both cases are the mean absorption position for all photons approximately zero, which confirms that the initial angles are drawn randomly.



(a) Trajectory of a single photon.

(b) Mean energy lost at each collision.

Figure 13: Plot of the trajectory and mean energy lost for a number of 1×10^5 photons in lithium with the modified parameters.



(a) Trajectory of a single photon.

(b) Mean energy lost at each collision.

Figure 14: Plot of the trajectory and mean energy lost for a number of 1×10^5 photons in lead with the modified parameters.

The Monte-Carlo simulation illustrates how photons are attenuated in a gas and a solid medium. We see that there are huge differences, of order 10^2, in the average distance travelled to absorption. This is a well-known mechanism where lead often works as a shield against gamma rays. The numerical results is at least in the ball-park of measured values. The deviations come from crude approximations as that a Compton scattering occurs at every step while neglecting other scattering processes. A better model would also change μ and l as a function of the energy.

References

[1] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes The Art of Scientific Computing, third edition, (2007). Cambridge University.

[2] Fornalsky, Krzysztof W (2018). Simple empirical correction functions to cross sections of the photoelectric effect, Compton scattering, pair and triplet production for carbon radiation shields for intermediate and high photon energies. *Journal of Physics Communications.* 2(3):035038

[3] NIST Standard Reference Database 126. Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients from 1 keV to 20 MeV for Elements Z = 1 to 92 and 48 Additional Substances of Dosimetric Interest. URL: https://physics.nist.gov/PhysRefData/XrayMassCoef/tab3.html

[4] Zhao Jianhua, Lecture notes. Yunnan University of Finance and Economics, Department of Statistics. URL: https://www.ynufe.edu.cn/pub/tsxy/jhzhao/teach/CS/notes/lec3.pdf

[5] G. Nelson, R. Reilly. Gamma-Ray Interactions with Matter. URL: https://sgp.fas.org/othergov/doe/lanl/lib-www/la-pubs/00326397.pdf