Sabahattin Mert Daloglu

Monte Carlo Radiative Transfer

Description of my code architecture:

Function and inputs:

My code simulates radiative transfer with the use of Monte Carlo methods by defining rad_transfer function. The function has 4 inputs that could be changed by the user. These are the <u>number of photons</u>, <u>number of steps</u>, <u>optical depth</u> and <u>albedo</u>. The number of steps represents the number of intended interactions we chose for the photons to undergo. However, this number could be reduced by any of the following: if the photon is absorbed, or if the photon escapes the gas cloud which is defined to be 10 units away from the star.

Variables:

albedo_option: is a array that includes the possibility of either “scattering” or “absorbing” as its elements.

angle: is a numpy array which has 361 evenly spaced azimuthal angles(phi) from 0 to 2pi.

absorbed: an empty array to count for every photon that are absorbed
escaped: an empty array to count for every photon that escapes
total_scat_absorbed: is an initial value set to 0, keeps track of the number of scattering events before a photon is absorbed
total_scat_absorbed: is an initial value set to 0, keeps track of the number of scattering events before a photon escapes.
zero_scatter: is an initial value set to 0, keeps track of the number of zero scattering events.
scattered: is an empty array to count for every scattering event
scat_escaped: is an empty array that will be assigned to ‘scattered’ array if the specific photon escaped the gas cloud and we stop tracking it.
scat_absorbed: is an empty array that will be assigned to ‘scattered’ array if the specific photon is absorbed and we stop tracking it.
i,j: are the initial values for x and y coordinates of the photons where they start to travel radially. In this case, the star is located at (0,0) as a point mass.
location_x: is an empty array to keep track of the x values of a photon
location_y: is an empty array to keep track of the y values of a photon

Outline (conditions, loops, etc.):

The function has a nested for loop. The outer for loop iterates over the number of photons(n). The inner loop iterates over the number of steps(N). This means that the function tracks all the steps of a specific photon, until it escapes or is absorbed, and does the same thing for the next photon. This is not a true representation of the radiative transfer in terms of the motion of all the photons over time, since they should start traveling radially outward from the star at the same time. However, for the simplicity of not tracking all the photons at the same time, this method could still yield accurate results for the purpose of this project.
Within the first for loop my code defines a random variable z, to generate the random distance traveled by a given photon before its next interaction event. The given non-uniform

PDF was used with the transformation method we practiced in class. This distance is later used to calculate the x and y projection to convert polar coordinates into cartesian coordinates. In order to do this, a random azimuthal angle from the 'angle' array was chosen and assigned as 'theta'. After calculating the x and y values of the distance, they are added to location_x and location_y arrays respectively. Notice that this calculation of the initial distance occurs before any interaction. Thus, a photon travels some distance initially, right after it was emitted from the star, and before any interaction with the medium. Later the second for loop starts to iterate over the number of steps defined by the user. There are two general possibilities for a photon to interact with the medium. Using the numpy random choice function, a choice of scattering vs absorption was made between the elements of 'albedo_option' array. The probability of choosing scattering can be represented with the value of albedo. Thus, probabilities associated with each elements of the array are assigned to 'p' parameter of the numpy random choice function.

Within the first if statement, there exists a nested inner if/else statement to take account of the maximum distance from the star(10) beyond which the photon will be assumed to have escaped the gas cloud, and added to the 'escaped' array. If the radial distance of the photon(s) is smaller than the maximum distance, then a new random scattering direction as well as the distance is chosen. This condition will also add a count of 1 to 'scattered' array which will be later used for the analysis. It is important to mention that if the radial distance is greater than the maximum distance, 'scattered' array is assigned to 'scat_escaped' array to distinguish the difference between the number of scatterings for an escaped photon and the number of scatterings for an absorbed photon. The length of these arrays(scat_escaped, scat_absorbed) is later added to total_scat_escaped and total_scat_absorbed resepectively because we will be using the cumulative sum of the scatterings to find the average number of scattering for escaping/absorbed photons in the analysis section.

If the photon is absorbed, we still have to consider the possibility of radial distance being greater than the maximum distance. This is because the radial distance could have been very close to maximum before but was not exceeding and the photon underwent scattering event before getting absorbed. If the radial distance is greater than the maximum distance, the nested if statement would count this stoppage of this simulation as a result of the escaping event not the absorbing event although the choice was made to be 'absorbing'. If the radial distance is not greater than the maximum distance, then a count of one is added to the 'absorbed' array. Later, scattered array is assigned to scat_absorbed for the average scattering analysis as mentioned before. After this, the nested if statement adds a count of one to zero_scatter array if there exists no previous scatterings. This would mean that a photon is absorbed without any scattering event. Lastly, we have to check if the radial distance is still smaller than the maximum radius after a random scattering event. If so, coordinates are appended to the location arrays.

Sabahattin Mert Daloglu

Description of how I tested the following parameters with diagnostic plots of the impact of varying each of them:

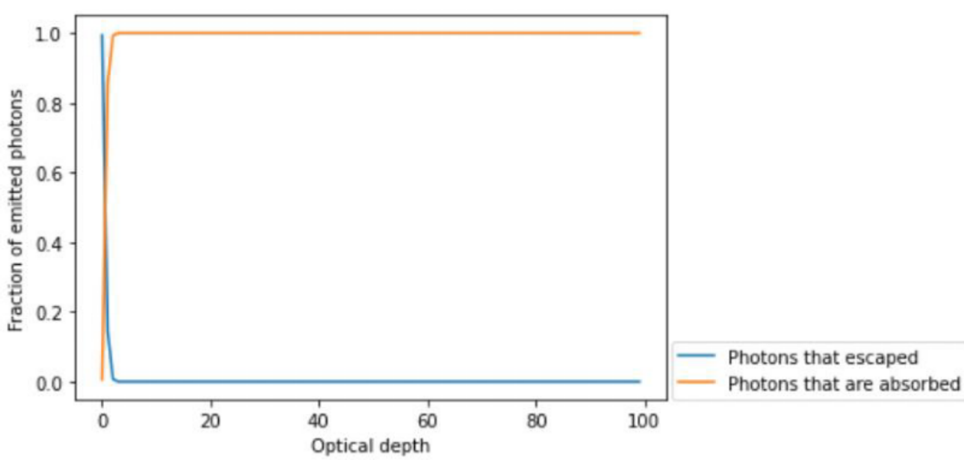
- Optical Depth
- Albedo
- Number of photons
- Extending simulation to 3D

To see the effects of changing the parameters above over a range of values, I defined a number of analysis functions which has a similar structure with the rad_transfer function without the plotting part and an extra for loop iterating over the parameter being varied.

<u>**Optical Depth**</u>

Testing the effects of the optical depth on the fraction of emitted photons that are absorbed/escaped:

To test the effects of the optical depth on the fraction of emitted photons that are absorbed/escaped, I defined analysis_optical function which has three user inputs as follow: start, stop, step. For convenience, parameters that are not tested are set to optimal values suggested by the project instructions. The number of photons and the number of steps are set to 1000 and albedo value is set to 0.95. Later, optical depth is changed from 0.01 to 100 and at each value the fraction of emitted photons that are escaped/absorbed is appended to 'escaped_photons' and 'absorbed_photons' arrays respectively. The value of the optical depth(T) in every iteration step is, also, appended to x array representing the x axis in the plot. Finally, two plots showing the optical dependence of the fraction of emitted photons that are absorbed/escaped are graphed.

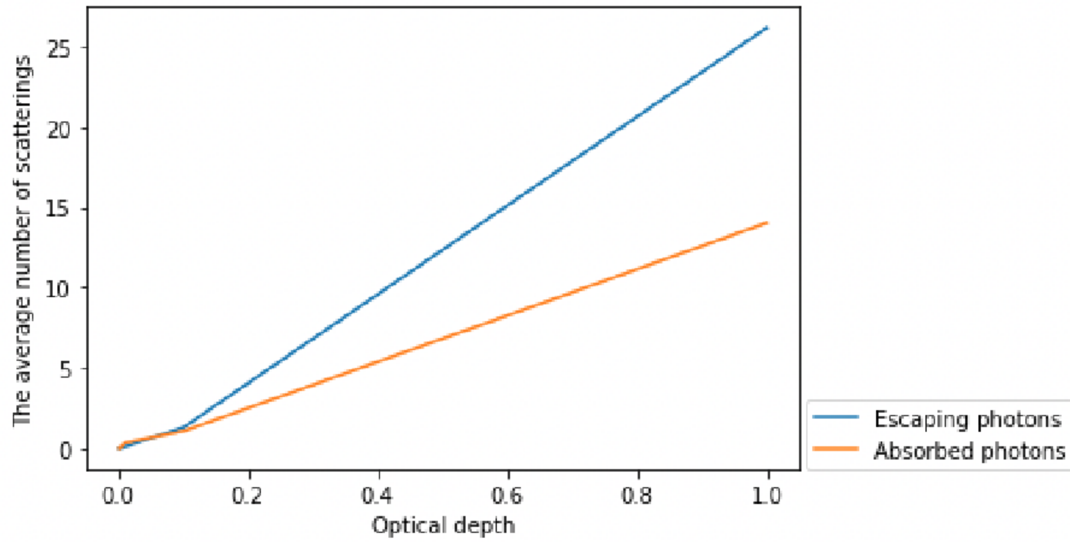


As expected, as we increase the optical depth, the fraction of emitted photons that are absorbed increases whereas the fraction of emitted photons that escaped the gas cloud decreases with the same rate. This is because a photon traveling in an optically thick medium with high optical depth, will either be quickly absorbed or undergo many scatterings before emerging. This increases the chance of being absorbed before escaping the gas cloud. On the other hand, a photon traveling in an optically thin medium, is very unlikely to be scattered or absorbed, hence has more chance to escape the gas cloud. A vertical line drawn anywhere on the graph would intersect two point, one from the absorbed photon line, the other from the escaped photon line. The fractional values of these two intersections would always add up to one, which explains the trend of these the two lines with the change of optical depth.

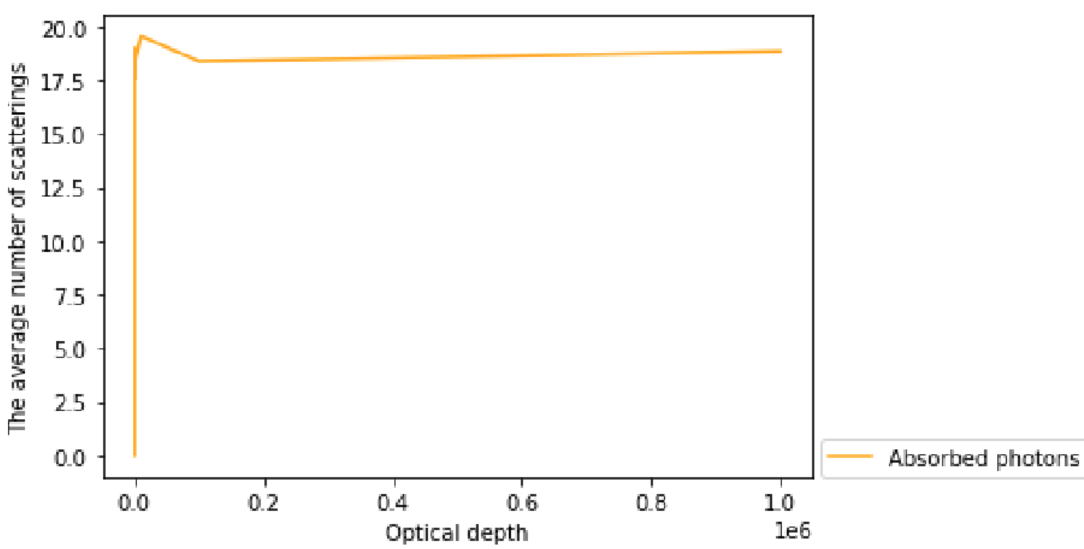
Testing the effects of the optical depth on the average number of scatterings for escaping photons and for absorbed photons:

Similar to the previous analysis function, another function named analysis_optical1 is defined. This time the average number of scatterings for escaping photons are tracked and recorded for values of optical depth between 1e-5 to 1e6 with steps increased by power of 10. In order to do this, an extra line of code under the first for loop is added. Hence, one can enter any range of numbers that will be transformed into the power of tens. By using this method, we can plot much wider range of optical depth, however with less precision since the points in between the power of tens will be smoothed out.

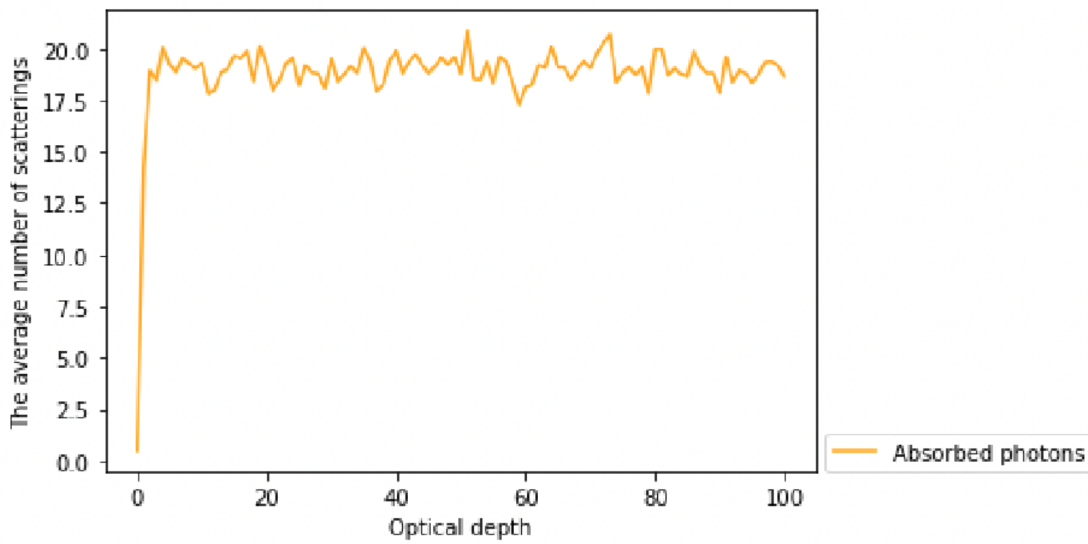
Furthermore, notice that the very last if statement makes sure that these values are appended to the predefined arrays only if length of the escaped and absorbed arrays are not zero. This is because while finding the average number of scatterings we use these lengths as the denominators and if they are zero we would get undefined results. Hence, it is important to append the average scattering values for absorbed and escaped photon under this condition. Two empty arrays ‘tse’ and ‘tsa’ for the average number of scatterings for escaping photons and for absorbed photons respectively are defined in the beginning of the function to keep track of the average values for each optical depth value. Furthermore, it is important to append the values of optical depth into the ‘x’ array under the last if condition since we want equal number of x and y values while plotting the graph. For instance, if the length of the escaped array is zero, then the code would omit appending the average value to the ‘tse’ array which is the y coordinate in the graph. If we do not have a y value for a point on the graph, then we should also omit recording the x value for that point. The following is the graph plotted for this analysis.



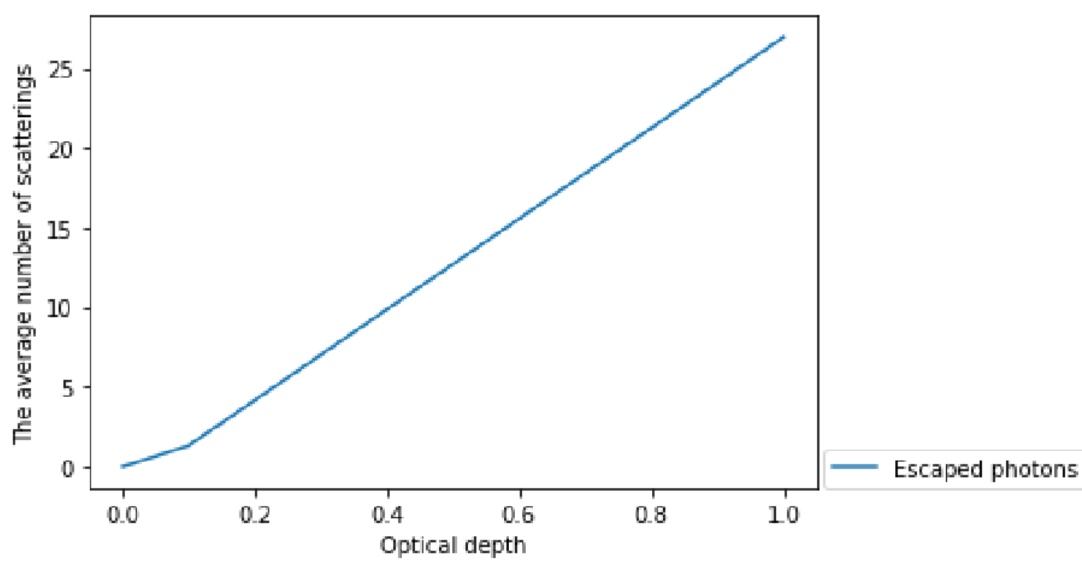
Notice that optical depth was varied between 1e-5 to 1e6. However, above graph plots the average values for optical depth between 1e-5 to 1. This is due to the last if condition in the 'analysis_optical1' function. As we increase the optical depth, there will be less number escaped photons according to our previous analysis. Thus, when the optical value is greater than 1, the length of the 'escaped' array is likely to become zero, which causes the if statement to omit appending values to 'tse', 'tsa', and 'x' arrays. However, this might not be the case for absorbed photons. To check our hypothesis, the average number of scatterings for escaping photons and absorbed photons are graphed separately. The graph below is for the average number of scatterings for absorbed photons only. Another function named 'analysis_optical2' is defined to omit escaped photons, by adjusting the last if condition and plotting array.



Hence, we can see that the average number of scatterings for absorbed photons is continuous for the entire range of optical depth values. Additionally, this graph seems to be overfitting the trend expected by varying the optical depth. Hence, adjusting the parameters of the function, by deleting the line T = 10**(T), and entering a range from 0.01 to 100 for the optical depth, the below graph is obtained. Although this graph has a narrower range of optical depth values, it accounts for more points in between steps, thus taking more time to plot. Both graphs seem to converge to the same average value.



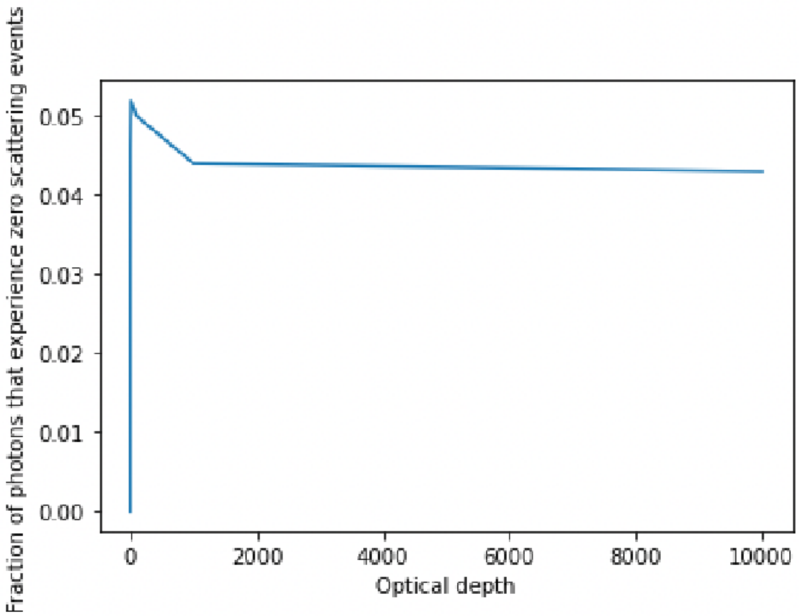
Next, the average number of scatterings for escaped photons only with the change in optical depth is plotted. As expected, the line is plotted within the range of 0 to 1. Although the entry range was 1e-5 to 1e5, when optical depth is assigned to values that are greater than 1, the length of the 'escaped' array becomes zero and thus the last if condition omits appending the average number scattering values.



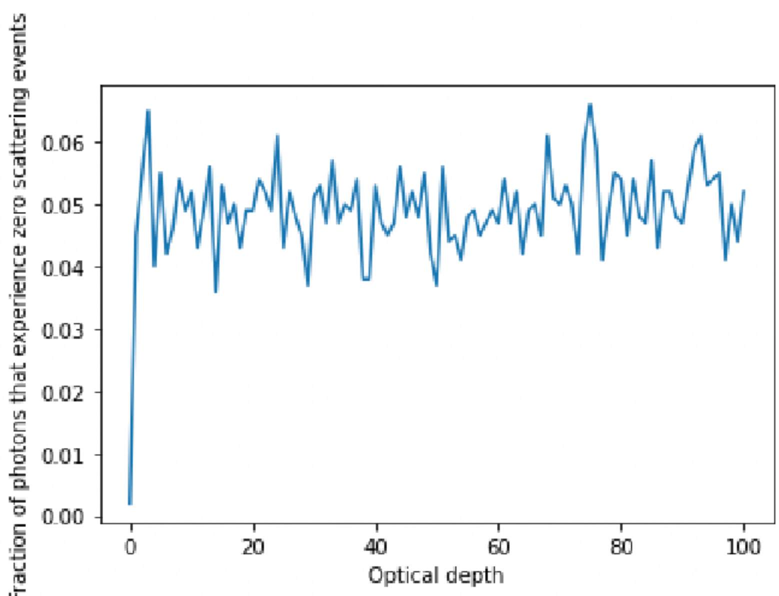
Sabahattin Mert Daloglu

Testing the effects of the optical depth on the fraction of photons that experience zero scattering events:

'analysis_optical4' function is defined to test the effect of the optical depth on the fraction of photons that experience zero scattering events. The graph below is plotted for the range 1e-3 to 1e4, with steps increase by power of tens.



To test the same effect on a narrower range with more points, the line T = 10**(T) was removed and the range values for the optical depth was adjusted from 0.01 to 100. The below graph is obtained:

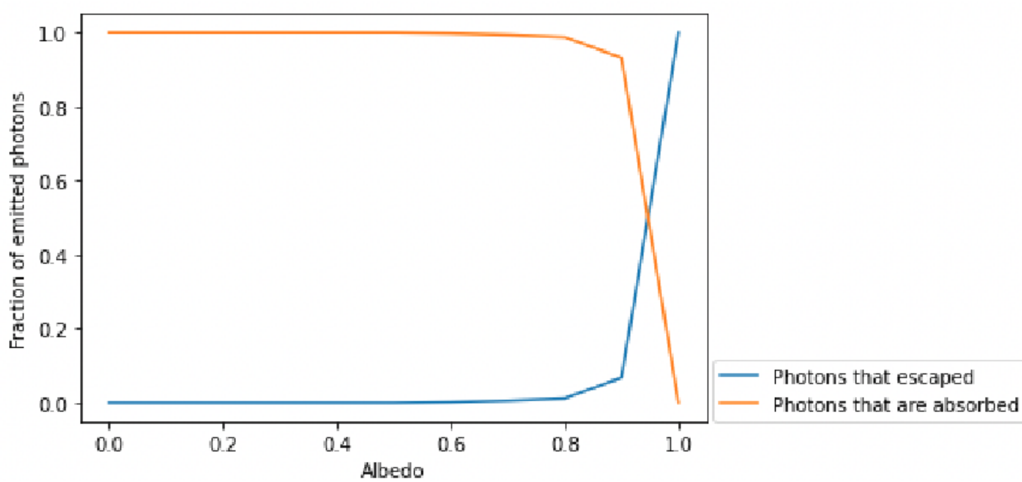


Overall, one can conclude that a photon traveling through an optically thick medium is more likely to get absorbed without scattering compared to a photon traveling through an optically thin medium. However, one should also consider the case where a photon travels through an optically thin medium without undergoing any scattering or absorbing events, but to detect this photon one should consider different range of maximum radius of the gas cloud. In my simulation, a photon has to be absorbed and have 0 scattering events before absorption to be counted under zero scattering event. Notice both graphs converge to the same fraction: around 0.045.

Albedo

Testing the effects of the albedo of the gas particle on the fraction of emitted photons that are absorbed/escaped:

To test the effects of the albedo of the gas particle on the fraction of emitted photons, 'analysis_albedo' function is defined. This function is similar to previous analysis functions in structure but this time the optical depth of the medium is set to 1 and the outer for loop iterates over the range of albedo values between 0 to 1 in steps of 0.1. The graph below is obtained by appending fraction of emitted photons that are escaped/absorbed into 'escaped_photons' and 'absorbed_photons' arrays accordingly. These values are then plotted against their according albedo values.

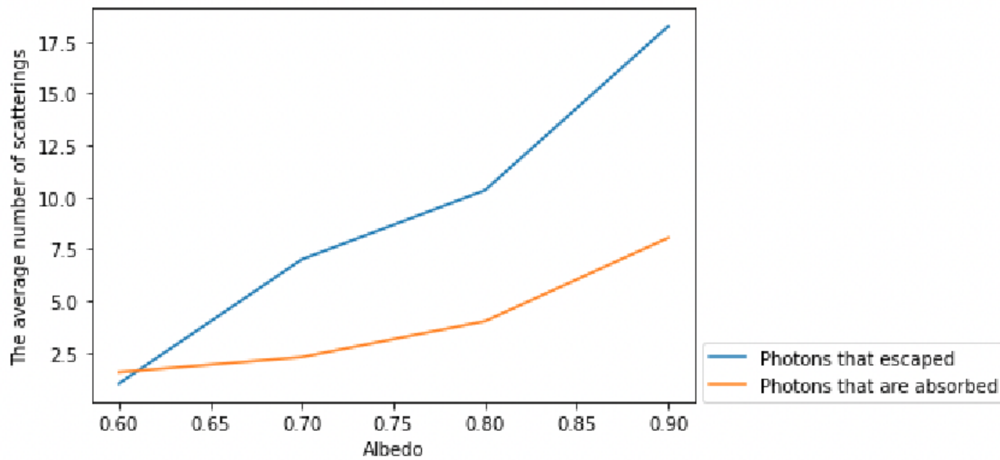


As we increase the albedo of the gas particle the fraction of the emitted photons that escape the gas cloud increases, whereas the fraction of the emitted photons that are absorbed decreases with the same rate. If we take a vertical line trace on this graph, the two points of intersection would add up to 1, hence the inverse relation of the escaped and absorbed photons make sense. Finally, to confirm the effect of different albedo values, one can consider the fraction of emitted photons that are absorbed when albedo is 1. This means that every time a photon interacts with the medium it will be scattered 100%. Hence, the fraction of photons emitted that are absorbed is 0 when the albedo of the gas particle is 1.

Sabahattin Mert Daloglu

Testing the effects of the albedo of the gas particle on the average number of scatterings for escaping photons and for absorbed photons:

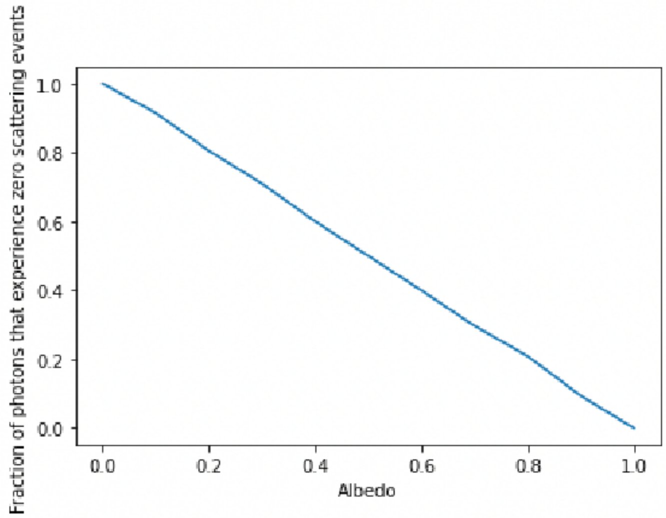
'anaylsis_albedo1' function is defined to test the effects of the albedo of the gas particle on the average number of scatterings for escaping photons and for absorbed photons. As before, there is an additional if statement added at the end of the function to make sure we are not dividing by 0 when calculating the average scattering. The graph below is plotted for range of albedo values between 0 to 1 in steps of 0.1.



From the graph above, one can interpret that as the albedo of the gas particle is increased the average number of scatterings for escaping photons and for absorbed photons increases. This interpretation is parallel to our understanding of the simulation since as we increase the albedo there will be more scattering events correspondingly.

Testing the effects of the albedo of the gas particle on the fraction of photons that experience zero scattering events:

'analysis_albedo3' function is defined to test the effects of the albedo of the gas particle on the fraction of photons that experience zero scattering events. The fraction of zero scattering events is appended to 'na_scatter' array for each iteration over albedo values varying between 0 to 1 in steps of 0.1. The graph below is plotted accordingly:

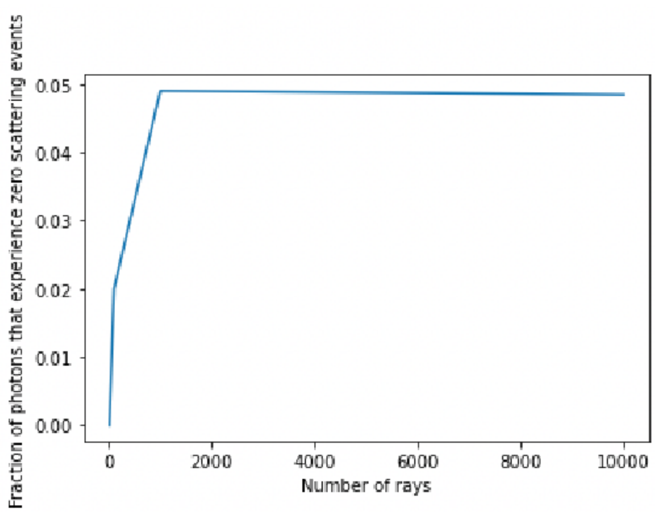


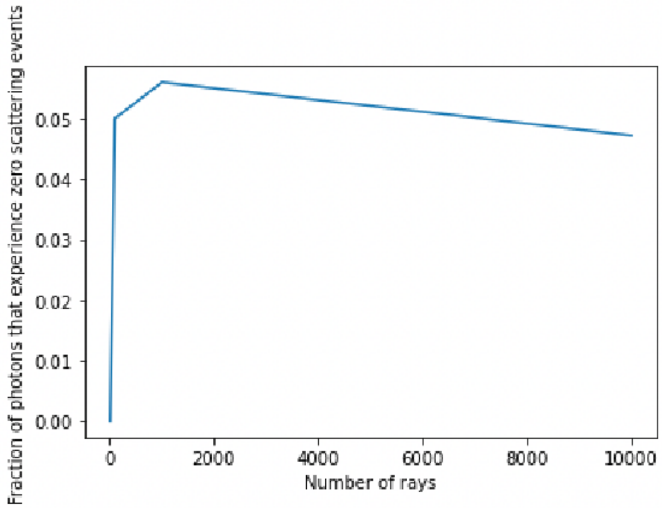
One can conclude that there exists an inverse relationship between the albedo of the gas particle and the fraction of photons that experience zero scattering events.

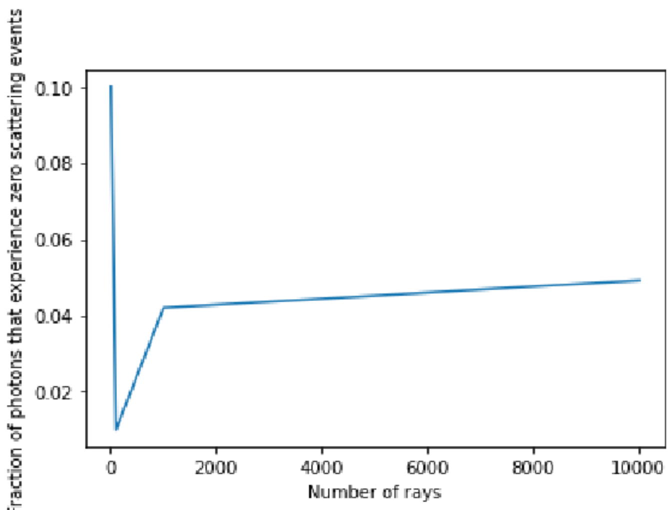
Number of rays

Testing the effects of the number of rays on the fraction of emitted photons that experience zero scattering events:

'analysis_rays' function is defined to test the effects of the number of rays/photons on the fraction of emitted photons that experience zero scattering events. Other parameters such as optical depth, albedo and number of steps is set to the optimal values(T=0.95, albedo=0.95, N=1000). The fraction of zero scattering events is appended to 'na_scatter' array for each iteration over number of rays varying between 1e1 to 1e4 in steps of power of 10. Interestingly, every time the plot of number of rays vs fraction of 0 scattering events is plotted, a different trend of line is observed. Below are some of the graphs obtained for the same parameters:





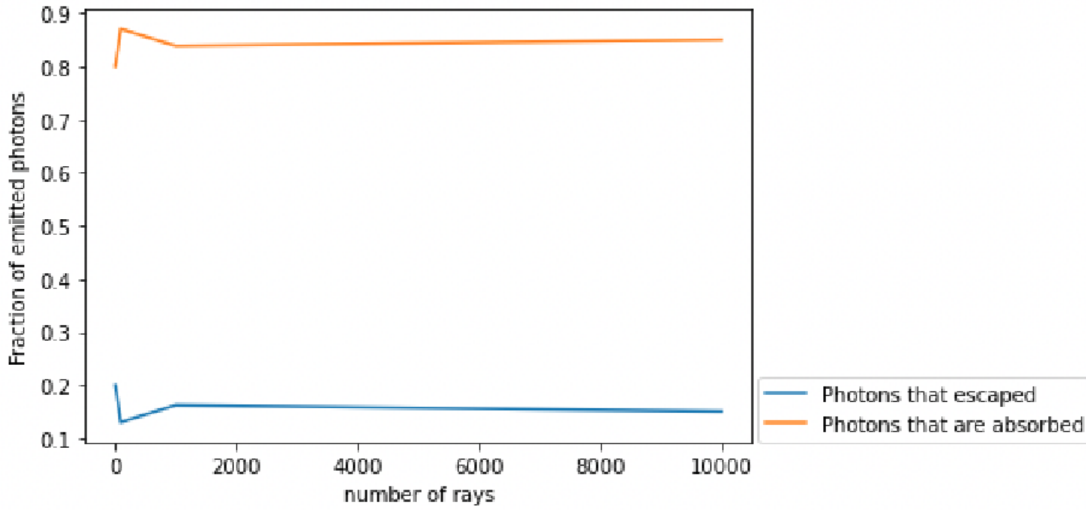


Although all the graphs above seem to be different due to randomness, for large number of photons the fraction of photons that experiences zero scattering events converges to 0.05 roughly.

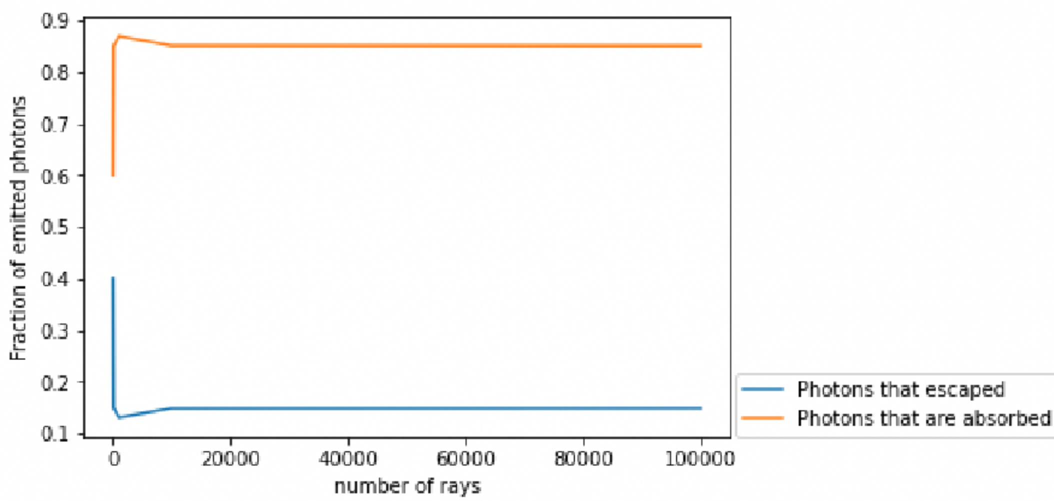
Testing the effects of the number of rays on the fraction of emitted photons that are absorbed/escaped:

'analysis_rays1' function is defined to test the effects of the number of rays/photons on the fraction of emitted photon that are absorbed/escaped. Similar to the previous analysis functions, the fraction of emitted photons that are escaped/absorbed is appended to 'escaped_photons', 'absorbed_photons' arrays respectively for each iteration over number of rays varying between 1e1 to 1e4 in steps of power of 10. The graphs below represent the convergence of the results:

Sabahattin Mert Daloglu



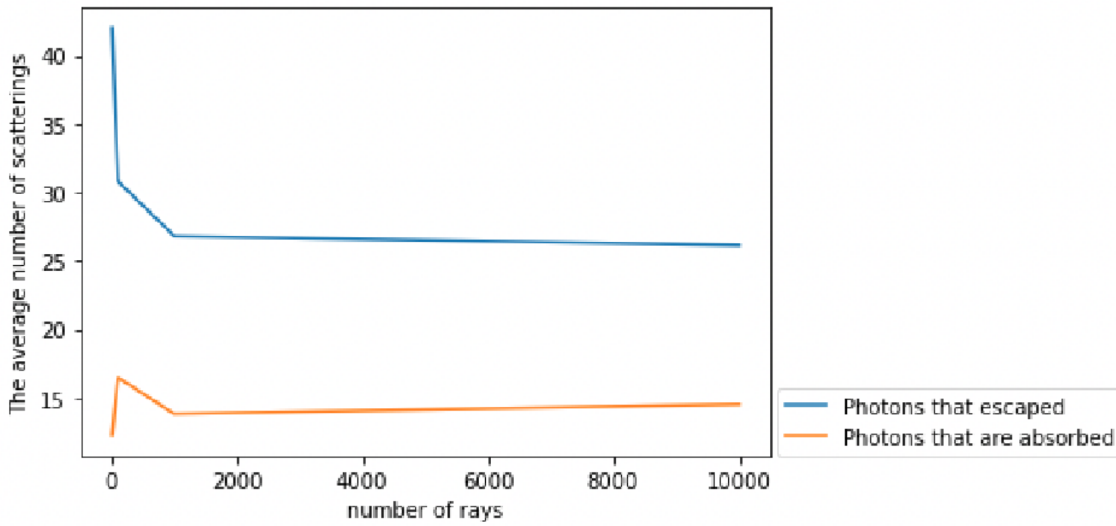
The range of number of rays is later changed to: [1e1,1e5,1] , the following result is obtained:

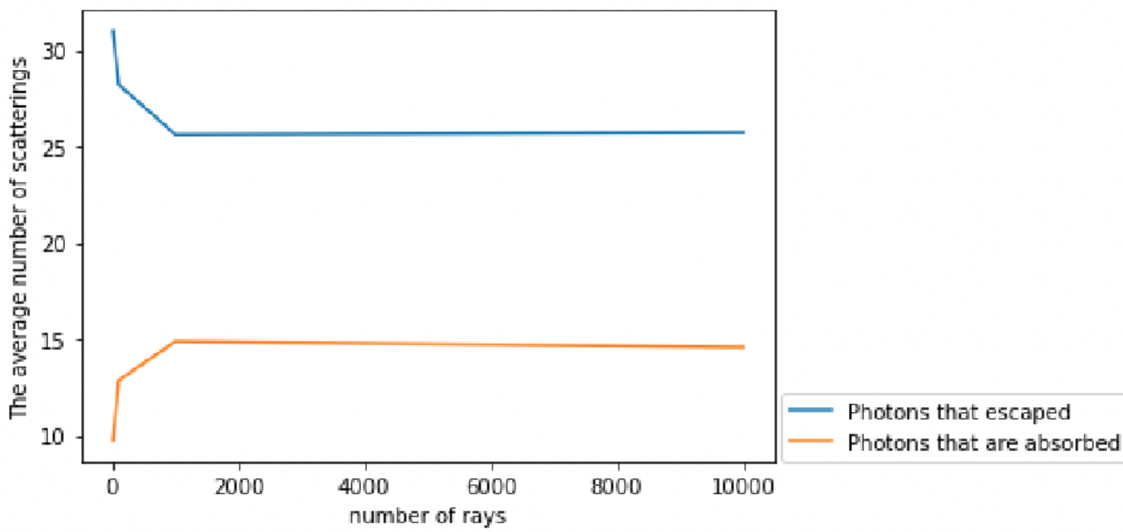


Hence, one can conclude that as the number of rays is increased the fraction of emitted photons that escaped converges to 0.15 whereas the fraction of emitted photons that are absorbed converges to 0.85.

Testing the effects of the number of rays on the average number of scatterings for escaping photons and for absorbed photons:

'analysis_rays2' function is defined to test the effects of the number of rays on the average number of scatterings for escaping photons and for absorbed photons. The average number of scatterings for escaping photons and for absorbed photons is appended to the 'tse' and 'tsa' arrays respectively for each iteration over number of rays varying between 1e1 to 1e5 in steps of power of 10. The graphs below represent the convergence of the results.(each plotted with the exact same parameters, difference is due to the random selection)





Hence, one can conclude that as the number of rays is increased, the average number of scatterings for escaping photons converges to a value between 26 to 27, whereas the average number of scattering for absorbed photons converges to 15.

Extending the simulation to 3D:

"rad_transfer3D" function is defined right after rad_transfer function, before starting the analysis of the parameters. The structure and the parameters of this 3D simulation function follows the same pattern as the 2D function, except an additional polar angle is defined as the variable 'theta'. Notice that in 2D function 'theta' is defined to be the azimuthal angle. However, in "rad_transfer3D", 'theta' is the polar angle and 'phi' is the azimuthal angle. As suggested by the instructions for the project, a random variable z1 is defined which is uniformly chosen. Later, the transformation method was used to generate a random theta angle. Furthermore, an additional empty array 'location_z' and the initial value 'k' is defined for the third dimension of the cartesian coordinates. By using randomly chosen distance and two spherical angles, spherical coordinates are transformed into cartesian coordinates. Finally, a 3D axes is enabled at the beginning of the function so that a 3D plot can be added to the figure. Below are the comparison of the analysis calculated by the 2D simulation and the 3D simulation with same parameters (n=100, N=100, T=1, albedo=0.95)

Dimensions	Fraction of emitted photons absorbed	Fraction of emitted photons escaped	Avg. number of scatterings for escaping photons.	Avg. number of scatterings for absorbed photons.	The fraction of photons that experience 0 scattering
2D	0.84	0.16	26.875	14.345	0.04
3D	0.44	0.56	9.821	4.955	0.08

Notice that these **key quantities** vary with every run due to the random choice of distance, angles, and albedo each time. One could make a plot of these quantities but there exists no independent variable as previous analysis. Previously, we varied the number of photons, the albedo, and the optical depth. However, this time we are keeping all these parameters constant and adding an additional dimension to our simulation. Nevertheless, one can observe some trends on the key quantities changing as 3D simulation is introduced, according to the table above(*same trends are observed for multiple runs):

- The fraction of emitted photons escaped the maximum radius is always greater in 3D than 2D. This is because in 3D motion photons are more likely to escape the radius of 10 units since there is an additional dimensional component 'k', which contributes to i^2+j^2+k^2<100 condition.
- Consequently, the fraction of emitted photons that are absorbed is always greater in 2D than 3D. This trend could be seen as the consequence of the previous trend since the fraction of emitted photons should add up to 1.
- One can also claim that since photons are less likely to spend more time within the maximum radius in 3D, there will be lower number of scatterings for both escaped and absorbed photons. Thus, average number of scatterings for escaping/absorbed photons is always smaller in 3D than 2D. This could also be seen as direct consequence of the first trend.
- The fraction of photons that experience zero scattering seems to be arbitrary and does not follow a trend when switching 2D simulation into 3D. Although in the table above zero scattering fraction for 3D simulation is double the fraction for 2D, this trend is not observed consecutively when the test is repeated multiple times.