Monte Carlo Radiative Transport for Astrophysical Applications

Zack Andalman,[1]★
[1]Yale University, New Haven, CT 06520, USA

4 May 2022

ABSTRACT

Radiative transport is propagation of electromagnetic radiation through a medium. As a beam of radiation propagates, it loses energy to absorption, gains energy from emission, and redistributes energy by scattering. These processes are described mathematically by the radiative transport equation (RTE). In idealized media, the RTE can be solved analytically, but for realistic media the RTE can only be solved using numerical methods. A common approach is to solve the RTE using Monte Carlo (MC) methods to sample the probability distribution functions which determine interaction lengths, scattering angles, and absorption rates. In this work, we develop a MC radiative transport code in Python from scratch with an eye towards applications in astrophysics. The code uses a three-dimensional Cartesian grid to allow arbitrary density distributions and includes Thompson scattering and linear polarization effects. We test on a simple exponentially decaying density distribution and compute interaction and polarization statistics. We run our simulation on the Grace Cluster at Yale University. The code is available on GitHub. On average, the photons scatter twice at a distance of 2 cm from the center of the density distribution. As the photons interact, they become polarized with polarization angles peaking at multiples of $\pi/6$. In future work, we will develop this code to model real astrophysical systems.

1 INTRODUCTION

Radiative transport is the propagation of electromagnetic radiation through a medium. As a beam of radiation propagates, it loses energy to absorption, gains energy from emission, and redistributes energy by scattering. These processes are described mathematically by the radiative transport equation (RTE).

$$\frac{1}{c}\frac{\partial}{\partial t}I_\nu + \hat{\Omega}\cdot\nabla I_\nu + (\kappa_{\nu,s}+\kappa_{\nu,a})\rho I_\nu = j_\nu\rho + \frac{1}{4\pi}\kappa_{\nu,s}\rho\int_\Omega I_\nu \mathrm{d}\Omega \quad (1)$$

where c is the speed of light, I_ν is the spectral radiance, $\hat{\Omega}$ is the direction of propagation, $\kappa_{\nu,s}$ is the scattering opacity, $\kappa_{\nu,a}$ is the absorption opacity, j_ν is the emission coefficient, and ρ is the mass density. The first two terms on the left-hand side represent the material derivative of the intensity, the third term on the left-hand side represents the loss in intensity due to scattering and absorption, the first term on the right-hand side represents the gain in intensity due to emission, and the second term on the right-hand side represents the gain in intensity due to radiation scattered from other directions.

Radiative transport has applications in a wide range of fields including optics, atmospheric science, medicine, computer graphics, and astrophysics. Within astrophysics, radiative transport has been used to model the interstellar medium, nebulae, molecular clouds, galaxies, and accretion disks. In idealized media (e.g. no scattering, isotropic density distribution) the RTE can be solved analytically, but for realistic media the RTE can only be solved using numerical methods. A common approach is to solve the RTE using Monte Carlo (MC) methods to sample to probability distribution functions which determine interaction lengths, scattering angles, and absorption rates (Noebauer & Sim 2019).

In this work, we develop a MC radiative transport code in Python which is available on GitHub. Our code takes advantage of NumPy arrays to evolve many photons simultaneously and reproduce expected statistical distributions with good signal-to-noise ratio.

2 METHODS

2.1 Sampling from Probability Distribution Functions

Suppose that we have a quantity $q \in [a, b]$ with a probability distribution function $P(q)$. To sample q over its probability distribution function, we set the cumulative probability distribution function $\psi(q)$ equal to a random number ξ sampled uniformly over a range $[0, 1]$ and solve for q.

$$\xi = \int_a^q P(q')\mathrm{d}q' = \psi(q) \quad (2)$$

In cases where Equation 2 cannot be evaluated analytically, we use the rejection method. In the rejection method, we sample q from uniform distributions and compute $P(q)$. We also sample P_{test} from a uniform distribution. If $P_{\text{test}} < P(q)$, we continue with the current value of q. Otherwise, we repeat the procedure.

2.2 Photon Packets and Interactions

The specific intensity of the radiation field is defined as the radiant energy $\mathrm{d}E_\nu$ per unit surface area $\mathrm{d}A$ per unit time $\mathrm{d}t$ per unit frequency $\mathrm{d}\nu$ per solid angle $\mathrm{d}\Omega$ at an angle θ to the surface normal.

$$\mathrm{d}I_\nu \equiv \frac{\mathrm{d}E_\nu}{\cos\theta \mathrm{d}A\mathrm{d}t\mathrm{d}\nu\mathrm{d}\Omega} \quad (3)$$

We track the evolution of photon packets representing a differential radiant energy $\mathrm{d}E_\nu$.

In the code, each photon packet $\boldsymbol{\gamma}_i$ is represented by twelve quantities.

$$\boldsymbol{\gamma}_i = (i, t, x, y, z, \cos\theta, \phi, \tau_{\text{run}}, \tau, I, Q, U) \quad (4)$$

where i is an ID, t is the time, $\mathbf{r} = (x, y, z)$ is the position of the photon in Cartesian coordinates, $\cos\theta$ and ϕ are the photon direction

of travel in spherical polar coordinates, $\tau_{\rm run}$ and τ are the running and interaction optical depth respectively, and $\mathbf{S} = (I, Q, U)$ is the Stokes vector. The Stokes vector is described in detail in Section 2.4.

The probability that a photon travels an optical depth τ without an interaction is $P(\tau) = e^{-\tau}$. Therefore, the probability of scattering after travelling an optical depth τ is $P(\tau) = 1 - e^{-\tau}$. Sampling from the cumulative probability distribution function (Section 2.1), the optical depth is

$$\tau = -\ln(1-\xi) = -\ln\xi$$

After each photon packet interaction, we sample a new interaction optical depth and initialize the running optical depth to zero. As the photon moves through the grid, we update the running optical depth.

$$\tau_{\rm run} = \tau_{\rm run} + \int_0^L n\sigma \mathrm{d}l \tag{5}$$

where n is the number density of scatters and absorbers and σ is the interaction cross section. When the running optical depth exceeds the interaction optical depth, we trigger an interaction. When an interaction occurs, we compare a random number to the albedo a. If the random number is smaller than the albedo, then the event is a scattering event. The albedo is given by

$$a = \frac{n_s\sigma_s}{n_s\sigma_s + n_a\sigma_a} \tag{6}$$

where n_s and n_a are the number densities of scatterers and absorbers respectively and σ_s and σ_a are the cross sections of scatterers and absorbers respectively. For absorption events, we stop evolving the photon packet. For scattering events, we reset the optical depths and update the scattering angle and Stokes vector according to the procedure described in Section 2.4.

The code produces NumPy array (npy) files as the simulation evolves. At each interaction, the code logs all photon packet data as well as the type of interaction. The code similarly outputs to the log if a packet reaches the edge of the grid or if a packet fails to reach the edge of the grid or be absorbed in a user-specified maximum number of iterations.

2.3 Evolving Photon Packets on a Cartesian Grid

We evolve photon packets on a Cartesian grid with constant density within each cell using the method of Hubber et al. (2015). The advantage of the grid-based approach is that the integral in Equation 5 reduces to multiplication because the integrand is constant within each cell. However, globally we can model arbitrary density distributions as long as they are discritized onto the grid.

Consider an photon packet initially located at position $\mathbf{r}_0$ and travelling in direction $\hat{\mathbf{n}}$. Then the position of the photon packet is

$$\mathbf{r} = \mathbf{r}_0 + t\hat{\mathbf{n}} \tag{7}$$

The distance between the packet and a given point $\mathbf{r}_i$ is

$$d_i^2 = (t\hat{\mathbf{n}} - \mathbf{p}_i)\cdot(t\hat{\mathbf{n}} - \mathbf{p}_i) = t^2 + p_i^2 - 2t\mathbf{p}\cdot\hat{\mathbf{n}}$$

where $\mathbf{p}_i \equiv \mathbf{r}_i - \mathbf{r}_0$ is the position relative to the original photon packet. The photon packet is initially contained in the cell whose centerpoint is the closest to $\mathbf{r}_0$. The photon packet crosses into a neighboring cell when it is equidistant between the centerpoints of the original cell and an adjacent cell. If $\mathbf{r}_i$ is the centerpoint of the original cell and $\mathbf{r}_j$ is the centerpoint of an adjacent cell, then the time of crossing is given by setting $d_i^2 = d_j^2$.

$$t^2 + p_i^2 - 2t\mathbf{p}\cdot\hat{\mathbf{n}} = t^2 + p_j^2 - 2t\mathbf{p}\cdot\hat{\mathbf{n}} \tag{8}$$

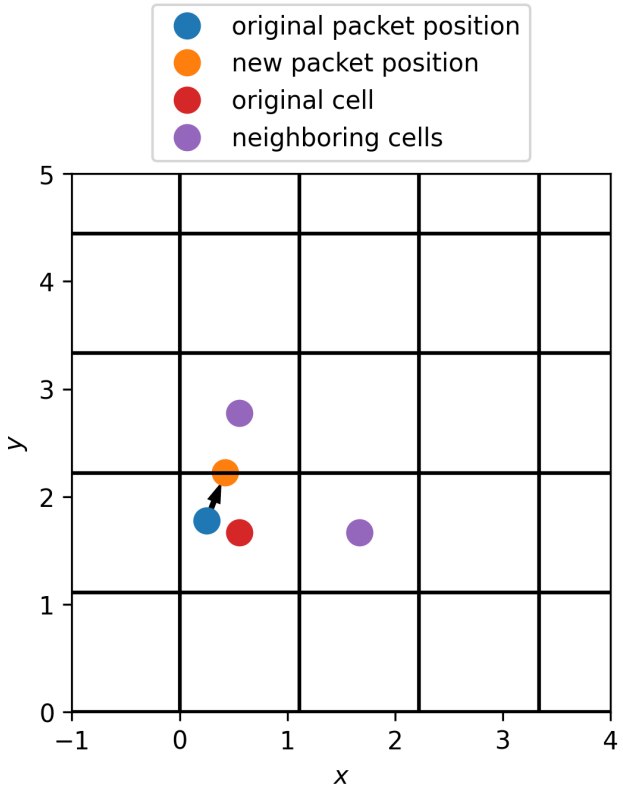

Figure 1. Visualization of the algorithm for tracing photon packets on a two-dimensional Cartesian grid. The original and new packet positions are depicted by the blue and orange dots respectively. The centerpoints of the original and neighboring cells are depicted by the red and purple dots respectively.

Solving for t, we obtain

$$t = \frac{p_j^2 - p_i^2}{2(\mathbf{p}_j - \mathbf{p}_i)\cdot\hat{\mathbf{n}}} \tag{9}$$

The packet will enter the neighboring cell which has the smallest crossing time. We ignore cells located behind the packet, which give a negative crossing time in Equation 9. When computing the new position of the packet, we add a small numerical factor $\epsilon = 10^{-6}$ to avoid the packet being located exactly on the face between cells. We visualize our method for tracing photon packets in Figure 1.

2.4 Thompson Scattering and Polarization

Our code models Thompson scattering, the elastic scattering of electromagnetic radiation by a free charged particle as described by classical electromagnetism. Thompson scattering is an approximation of Compton scattering, which is the dominant scattering process in many astrophysical systems. In Thompson scattering, the kinetic energy of the scatterer and the frequency of the photon do not change as a result of the scattering. The Thompson cross section $\sigma_{\rm T}$ is a constant. The number density of electrons n_e is related to the density ρ and the mass fraction of hydrogen X in the medium.

$$n_e = \frac{\rho}{2m_p}(X+1) \tag{10}$$

where m_p is the mass of the proton.

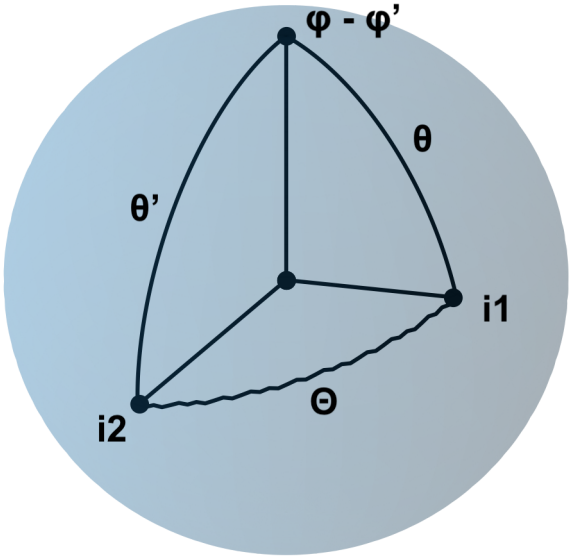

Figure 2. Visualization of the scattering geometry. (θ, ϕ) and (θ', ϕ') are the pointings of the incoming and outgoing photon packets respectively, Θ is the scattering angle, and i_1 and i_2 are as shown and can be derived from the spherical law of cosines.

The intensity and polarization of a photon packet are completely described by the Stokes parameters I, Q, U, V. I measures the total intensity of the beam, Q measures the flux with linear polarization parallel or perpendicular to the z-axis, U measures the flux with linear polarization 45° away from those axes, and V measures the flux with circular polarization. The Stokes parameters are often combined into a vector $\mathbf{S} = (I, Q, U, V)$ known as the Stokes vector. The Stokes vector for unpolarized light is $\mathbf{S} = (1, 0, 0, 0)$. The effect of any optical system on the polarization of light can be determined by applying Mueller calculus to the input Stokes vector to obtain the output Stokes vector.

Following Chandrasekhar (1960), the output Stokes vector $\mathbf{S}'$ is related to the input Stokes vector $\mathbf{S}$ by

$$\mathbf{S}' = L(\pi - i_2)R(\Theta)L(-i_1)\mathbf{S} \tag{11}$$

where $R(\Theta)$ is the scattering matrix describing the scattering probability to scatter at angle Θ in the frame of the particle, $L(\psi)$ is the rotation matrix for the polarization basis

$$L(\psi) \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\psi & \sin 2\psi & 0 \\ 0 & -\sin 2\psi & \cos 2\psi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

and i_1 and i_2 are angles related to the scattering geometry as shown in Figure 2. For Thompson scattering, the scattering matrix is

$$R(\Theta) = \frac{3}{4}\begin{pmatrix} \cos^2\Theta + 1 & \cos^2\Theta - 1 & 0 & 0 \\ \cos^2\Theta - 1 & \cos^2\Theta + 1 & 0 & 0 \\ 0 & 0 & 2\cos\Theta & 0 \\ 0 & 0 & 0 & 2\cos\Theta \end{pmatrix} \tag{13}$$

Evaluating Equation 11 explicitly, we find

$$\begin{aligned} I' = \frac{3}{4}\big[&(\cos^2\Theta + 1)I + (\cos^2\Theta - 1)\cos(2i_1)Q \\ &- (\cos^2\Theta - 1)\sin(2i_1)U\big] \end{aligned} \tag{14}$$

$$\begin{aligned} Q' = \frac{3}{4}\big[&(\cos^2\Theta - 1)\cos(2i_2)I \\ &+ (\cos^2\Theta + 1)\cos(2i_1)\cos(2i_2)Q \\ &- 2\cos\Theta\sin(2i_1)\sin(2i_2)Q \\ &- (\cos^2\Theta + 1)\sin(2i_1)\cos(2i_2)U \\ &- 2\cos\Theta\cos(2i_1)\sin(2i_2)U\big] \end{aligned} \tag{15}$$

$$\begin{aligned} U' = \frac{3}{4}\big[&(\cos^2\Theta - 1)\sin(2i_2)I \\ &+ (\cos^2\Theta + 1)\cos(2i_1)\sin(2i_2)Q \\ &+ 2\cos\Theta\sin(2i_1)\cos(2i_2)Q \\ &- (\cos^2\Theta + 1)\sin(2i_1)\sin(2i_2)U \\ &+ 2\cos\Theta\cos(2i_1)\cos(2i_2)U\big] \end{aligned} \tag{16}$$

$$V' = \frac{3}{2}\cos\Theta V \tag{17}$$

Because the Stokes parameter V is independent of the other Stokes parameters, we don't evolve V and construct Stokes vectors with only three components.

At each scattering event, we sample the angle i_1 from a uniform distribution. Then, we sample Θ from the scattering matrix $R(\Theta)$. Given the pointing of the incoming photon packet (θ, ϕ), the scattering angle Θ, and the angle i_1, we can compute the pointing of the outgoing photon packet (θ', ϕ') and the angle i_2 using spherical trigonometry. By the spherical law of cosines, we have

$$\cos\theta' = \cos\theta\cos\Theta + \sin\theta\sin\Theta\cos i_1 \tag{18}$$

By the spherical law of sines, we have

$$\sin i_2 = \frac{\sin\theta\sin i_1}{\sin\theta'}, \quad \sin(\phi - \phi') = \frac{\sin\Theta\sin i_1}{\sin\theta'} \tag{19}$$

Finally, we determine the new Stokes vector $\mathbf{S}'$ using Equation 11. Because the Stokes parameter V is independent of the other Stokes parameters for Thompson scattering, we do not evolve V in our simulation and our Stokes vectors only have three components.

We use the rejection method to sample Θ from $R(\Theta)$. First, we sample $\cos\Theta$ from a uniform distribution over $[-1, 1]$ and compute I' using Equation 14. Then we sample I'_{test} from a uniform distribution over $[0, 3/2]$, noting that $3/2$ is the maximum value of I' (corresponding to scattering angle $\Theta = 0$). If $I'_{\text{test}} < I'$, we continue with scattering angle Θ. Otherwise, we repeat the procedure. The efficiency of the rejection method is 66%. For the case of unpolarized light, the phase function is

$$P(\Theta) = \frac{3}{4}(\cos^2\Theta + 1)$$

In Figure 3, we show that our method reproduces the analytical phase function for unpolarized light. This method was used by previous works including Mason et al. (2013).

2.5 Simulation

We test our code on a simple density distribution which drops off exponentially with radius.

$$\rho(\mathbf{r}) = 300\ \text{g/cm}^3 e^{-r} \tag{20}$$

The grid is an 8 cm cube centered on the origin with 200 cells in each dimension for a total of 8×10^6 cells. We set the hydrogen mass

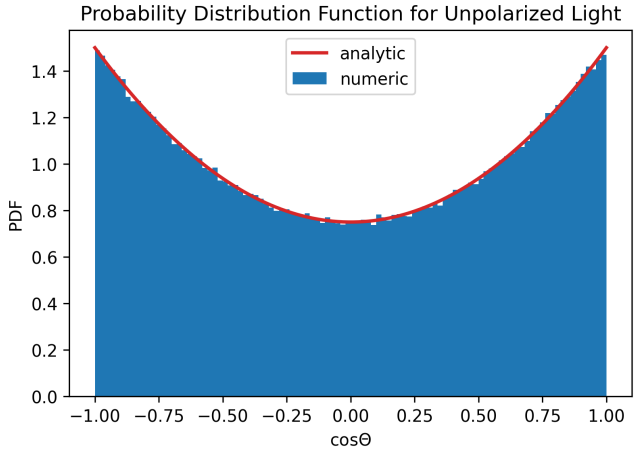

Figure 3. The phase function for unpolarized light as a function of $\cos\Theta$. In blue, we plot a normalized histogram of $\cos\Theta$ for 10^5 photon packets. In red, we plot the analytical phase function.

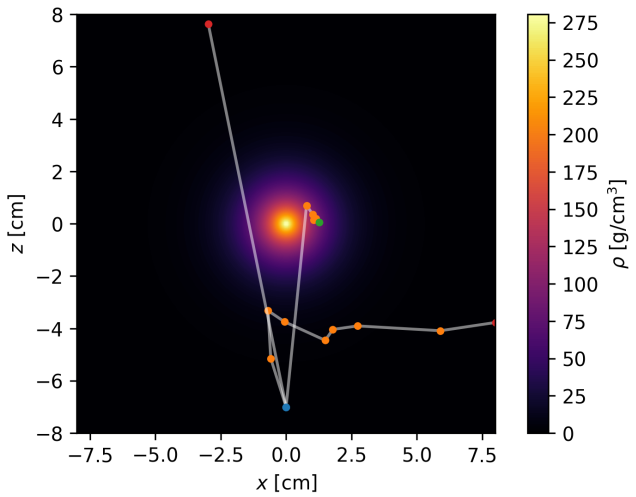

Figure 4. Density distribution and three example photons projected onto the xz-plane. The path of the photons is shown in white. Blue, orange, red, and green dots represent start, scatter, terminate, and absorb events respectively.

fraction $X = 0.7$ and the albedo $a = 0.9$. We source $N = 10^6$ photons at time $t = 0$ from the point $(0, 0, -7)$ with angular probability distribution functions

$$P(\cos\theta) = \frac{1}{2}\max(0, \cos\theta), \quad P(\phi) = \frac{1}{2\pi} \tag{21}$$

Sampling the cumulative probability distribution functions (Section 2.1), the initial angles are

$$\cos\theta = \sqrt{\xi}, \quad \phi = 2\pi\xi \tag{22}$$

The photons are initially unpolarized with Stokes vectors $\mathbf{S} = (1, 0, 0, 0)$. The simulation is run on the Grace cluster at Yale University as ten parallel jobs each consisting of 10^5 photons. In Figure 4, we depict the density distribution and the paths of three example photons from the source. One photon reaches the edge of the grid without interacting, a second photon reaches the edge of the grid after several scattering events, and a third photon is absorbed after several scattering events.

3 RESULTS

3.1 Post-Processing

As described in Section 2.2, our code logs photon packet data at each event. To determine photon data for a particular photon $\boldsymbol{\gamma}_i$ at a particular time t_0, we first filter events by the photon ID. Second, we determine the index of t_0 in the list of time coordinates for all events involving the photon. If the most recent event was an absorption event, we ignore the photon. Otherwise, we take the direction of travel and Stokes vector of the photon from the values at the last event. The position of the photon is

$$\mathbf{r}(t_0) = \mathbf{r}(t_{\text{event}}) + \hat{n}_{\text{event}}(t - t_{\text{event}}) \tag{23}$$

where the "event" subscript indicates a quantity taken from the most recent event. In Figure 5, we visualize the positions of 1000 randomly selected photons colored by intensity at three snapshots. Some photons do not interact and continue to move radially outward from the source while other photons scatter, resulting in changing intensities and still other photons are absorbed and disappear.

3.2 Interaction Statistics

Over the course of the simulation, 21.4% of the photons were absorbed and the rest made it to the edge of the grid. Although all photons were sourced moving in the positive z direction, by the end of the simulation 49.5% of photons point in the negative z direction. In Figure 6, we plot a histogram of the number of scatterings for each photon. The distribution of number of scatterings is approximately Poissonian. On average, each photon scattered 1.91 times with a standard deviation of 2.49. The maximum number of scatterings for a photon is 76. In Figure 7, we plot the interaction frequency as a function of time. The frequency of interactions peaks when the photons from the source pass through the center of the density distribution. As photons move away from the center of the distribution, the frequency of interactions decays to zero. In Figure 8, we plot a histogram of the interaction radius for each interaction normalized by the surface area of a sphere at the radius. We find that the number of interactions per area peaks around 2 cm from the center of the density distribution.

3.3 Polarization Statistics

We compute the Stokes vector for each photon packet at the end of the simulation. From the Stokes parameters, we can derive the degree of linear polarization p and the linear polarization angle α.

$$p = \frac{\sqrt{Q^2 + U^2}}{I}, \quad \alpha = \frac{1}{2}\arctan\left(\frac{U}{Q}\right) \tag{24}$$

In Figure 9, we plot a histogram of p and α for each photon. Both histograms have a large bin at zero due to photons which did not interact and therefore maintain their original unpolarized Stokes vectors. The degree of polarization peaks at $p = 0.75$ and the polarization angle peaks at multiples of $\pi/6$. The mean degree of polarization is $\langle p \rangle = 0.589$, indicating that over time the photons become more polarized.

The mean intensity by the end of the simulation $\langle I \rangle = 1.79$ is greater than the initial intensity. This makes sense because each scattering, the intensity can increase by up to a factor two, so over many interactions some fraction of photons will reach exponentially high intensities and drive up the mean. By contrast, the median

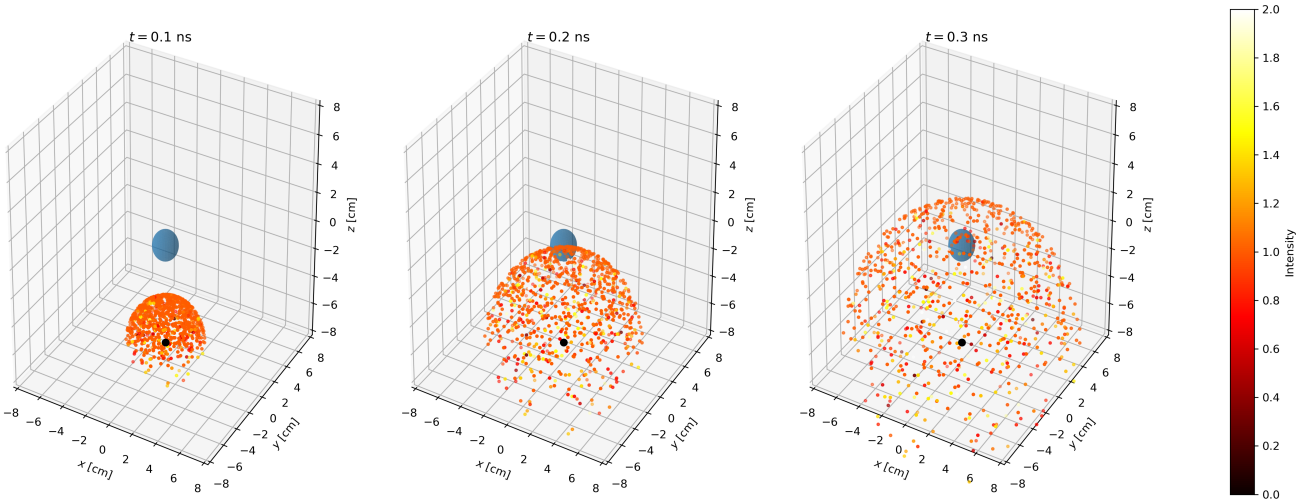

Figure 5. The positions of 1000 randomly selected photons at three snapshots in time in the simulation. The photons are colored by intensity. The density distribution is represented by a blue sphere at the origin of radius one. The source is represented as a black dot.

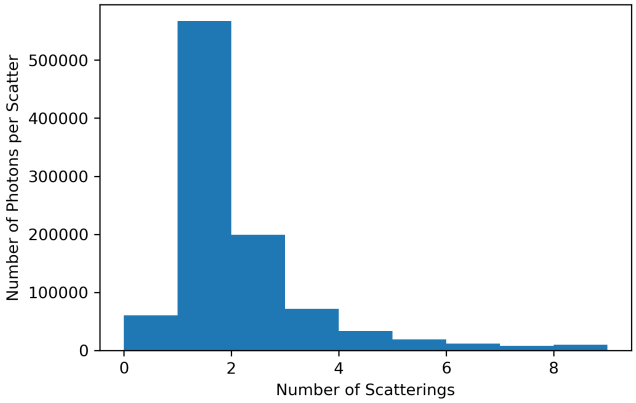

Figure 6. Histogram of the number of scatterings for each photon. The distribution is approximately Poissonian.

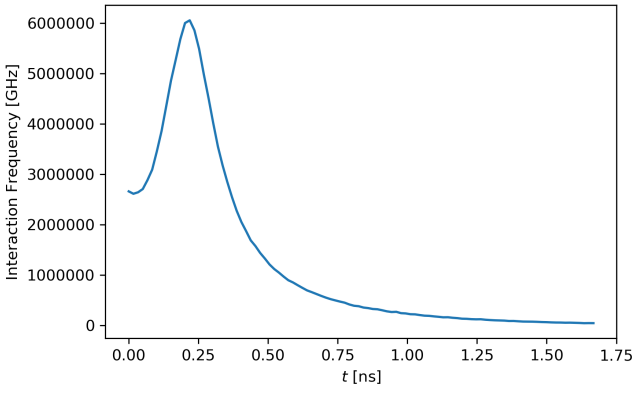

Figure 7. The interaction frequency as a function of time. The frequency peaks when the photons from the source pass through the center of the density distribution.

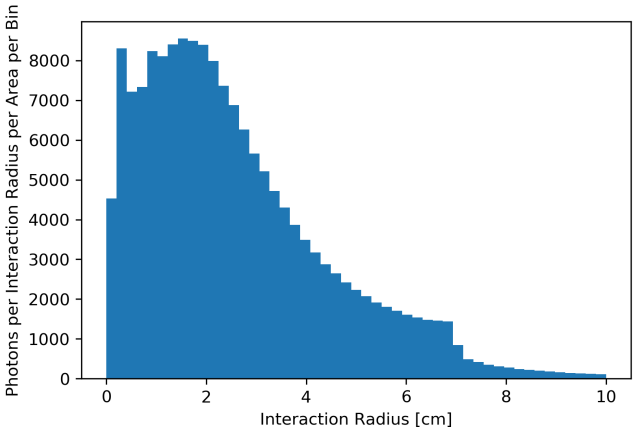

Figure 8. Histogram of the radius of interaction for each interaction normalized by the surface area of a sphere at that radius.

intensity is $\langle I \rangle_{\text{median}} = 1.04$, indicating that on averages photons neither gain nor loose intensity.

4 CONCLUSIONS

In this work, we develop a MC raditive transport code from scratch in Python. The code uses a structured grid to allow arbitrary density distributions and accounts for anisotropic scattering and polarization. We test the code on a simple density distribution and compute interaction and polarization statistics. About 20% of the photons are ultimately absorbed while the remaining photons reach the edge of the simulation box. We find that on average, the photons scatter twice at a distance of 2 cm from the center of the density distribution. As the photons interact, they become polarized with polarization angles peaking at multiples of $\pi/6$.

There are improvements which would allow the code to capture a wider range of astrophysical systems with greater fidelity. Thompson scattering is the low-energy limit of Compton scattering; that is, the

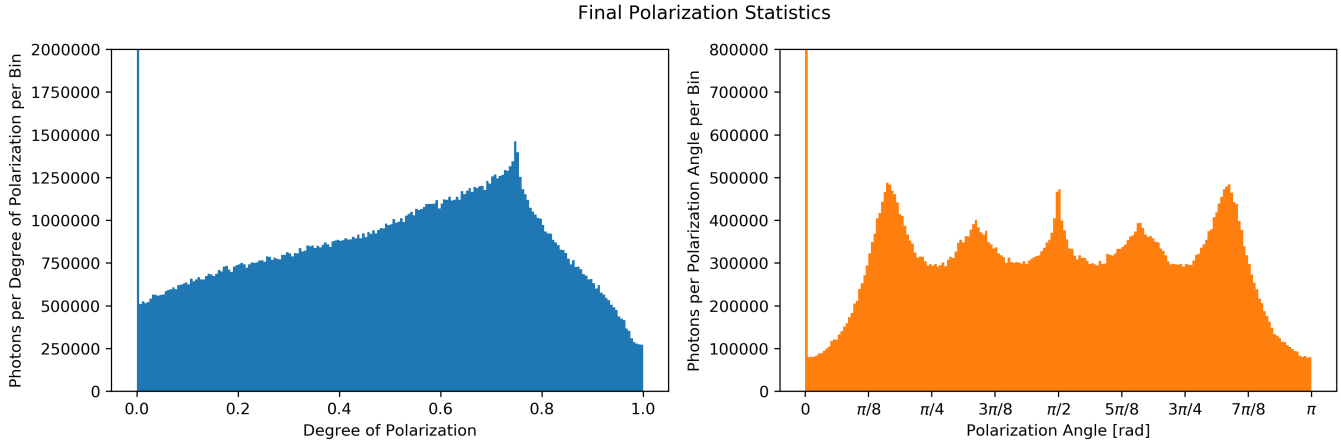

Figure 9. On the left panel, a histogram of the degree polarization for each photon. On the right, a histogram of the polarization angle for each photon. Both histograms have a large bin at zero due to photons which did not interact and therefore maintain their original unpolarized Stokes vectors. The degree of polarization peaks at $p = 0.75$ and the polarization angle peaks at multiples of $\pi/6$.

limit in which the energy of the photon is much less than the rest mass energy of the scattering electron. To accurately capture the evolution of high-energy photons which are common in astrophysical systems, we could fully implement Compton scattering. In Compton scattering, the energy of the outgoing photon is different than the energy of the incoming photon, so we would need to evolve energy and we could produce a spectrum. The scattering matrix also depends on the energy of the scattering electron, the we would need to numerically integrate the scattering matrix over the electron energy probability distribution function $f(E)$.

$$R(\Theta) = \int_{E_{\rm min}}^{\infty} \mathrm{d}E f(E) R_E(\Theta) \tag{25}$$

where E is the electron energy in units of the rest mass energy $m_e c^2$. For a thermal plasma, the distribution function can be modelled as a relativistic Maxwellian distribution.

$$f(E) = \frac{y}{4\pi K_2(y)} e^{-yE}, \quad y \equiv \frac{m_e c^2}{kT_e} \tag{26}$$

where T_e is the electron temperature, m_e is the electron mass, and K_2 is the MacDonald function. The full expression for the Compton scattering matrix is provided in Poutanen & Vilhu (1993).

Another improvement would be to implement image construction in our post-processing to facilitate comparison with observational data. To construct an image, we would need to chose a point in the simulation to represent the observer and chose a plane to represent the image plane. Then, we could project all photons moving towards the observer onto the image plane to produce a two-dimensional image of the simulation from the perspective of the observer. However, in general photon packets do not move towards any particular point in our simulation. To create photon packets which move towards the observer, we could emit photons towards the observer at each scattering event and weight the Stokes vector using the scattering phase function.

REFERENCES

Chandrasekhar S., 1960, Radiative transfer
Hubber D. A., Ercolano B., Dale J., 2015, Monthly Notices of the Royal Astronomical Society, 456, 756
Mason J. P., Patel M. R., Lewis S. R., 2013, Icarus, 223, 1
Noebauer U. M., Sim S. A., 2019, Living Reviews in Computational Astrophysics, 5, 1
Poutanen J., Vilhu O., 1993, A&A, 275, 337

This paper has been typeset from a TEX/LATEX file prepared by the author.