Dust Scattering in Protoplanetary Disks

Daniel F. Devlin[1],
[1]Department of Astronomy, University of Virginia, 530 McCormick Road, Charlottesville, VA 22904-4325, USA

16 May 2022

ABSTRACT
Using the Monte Carlo radiative transfer method, the following code, written in C++ and building on the Athena++ framework, adds a program for dust scattering simulations. We then compare the output of the code to a tested dust simulation code written in FORTRAN. For this comparison we use a problem generator which simulates a protoplanetary disc of varying density which surrounds a binary star system with the stars having different luminosities. We then discuss the issues with our code and how we might correct the code and improve on it.

Key words: Monte Carlo radiative transfer, dust scattering, protoplanetary disc

1 INTRODUCTION

Even in the most remote intergalactic regions of the universe, the vacuum of space is not truly empty, as there is always some amount of matter moving through space. Thus, as light travels through space, it is likely it will eventually interact with matter. Within galaxies, their can be a large amount of dust particles for light to interact with as it travels outward from objects within the galaxy. Due to the vast distances which light must cover to reach our telescopes, it is possible that light which we observe was scattered by dust at some point in its journey to Earth. When light interacts with a medium, it can either be absorbed by the medium or scattered out into space. Dust scattering is extremely important to take into account when making astronomical observations as dust scattering will impact the light which we are observing. In order to better understand observations of astronomical objects, we want to know how our observation has been impacted by dust scattering. Multiple factors determine the extent that dust scattering impacts an observation, such as concentration of dust between the telescope and the source, the distance to the object and the wavelength of the light.

Protoplanetary discs are accretion discs around young stars, or protostars which are made up of mostly dust and gas, so it is critical to understand the impact dust scattering has on observations of protoplanetary discs. Protoplanetary discs are formed by the gravitational collapse of a huge gas cloud parsecs across in size, forming a protostar. The leftover dust and gas cloud, falls inward and accelerates forming a rotating a disc, as the conservation of angular momentum requires the dust continue to rotate, but the constant collisions of the dust cause a single direction of rotation to dominate, leading to the formation of a rotating accretion disc. The energy from the gravitational collapse along with the protostar's radiation heats the disc, causing the dust to emit typically in the infrared or sub millimeter wavelengths which we can observe with ground-based telescopes and arrays such as ALMA, the Atacama Large Millimeter Array in northern Chile (Williams & Cieza 2011). Planets can then form in the disc by dust particles clumping together through collisions until the clump is large enough to have a gravitational field capable of attracting nearby particles. Protoplanetary discs are mainly made up of dust orbiting around the proto star, thus we can learn both about the disc and the star by comparing the observations of the protoplanetary disc with models and simulations which use dust scattering, as light passing through the disc will be scattered by the dust.

Astronomers developed a variety of methods to analyze scattered light, such as making simulations of dust scattering and comparing the simulations to observations. However these simulations are quite complicated due to the large variety of shapes and sizes of dust and thus there are a variety of ways in which light interacts with dust particles. It is impossible to accurately guess how an individual photon will interact with a particle of dust, but by using a very large number of photons randomly interacting with dust, astronomers can make an accurate simulation of dust scattering on large scales. The Monte Carlo radiative transfer method is commonly used by astronomers to create dust scattering simulations which are not overly complicated but can create useful models for real dust scattering (Whitney 2011). The Monte Carlo method tracks the progress of a photon with randomized properties through a specified medium and accounts for absorption and scattering given the parameters of the simulation, and records when and where the photon leaves the medium. This process is then repeated for a specified number photons. The complicated nature of the dust scattering lies within the parameters of the absorption, scattering and randomized parameters of the photon. Since scattering depends on the polarization, wavelength and direction of impact of the photon along with the size and shape of the dust grain, this process can become quite complicated (Robitaille 2011). The way in which the light is scattered can provide insight into the properties of the individual dust particles, such as their size and composition. By using the monte carlo dust scattering simulation below, we can predict what observations of protoplanetary discs should look like, and learn about the early solar system we are observing and in turn learn about the formation of our own solar system and other solar systems across the Milky Way galaxy.

The following outlines how the attached code uses Monte Carlo scattering to model dust scattering in a protoplanetary disk using the Athena++ grid framework Stone et al. (2020).

2 METHODS

Generally, the Monte Carlo radiation transfer method simulates the lives of photon packets as they travel through a medium and are scattered, absorbed or leave the medium. The simulated photons packets are given a total energy, along with a direction of travel and a polarization. By giving the packets a total energy, the specific intensity, I_v can by modeled by the following equation where dE_v is the radiant energy, or the photon packet energy, passing through a surface area dA at an angle θ perpendicular to the surface of solid angle $d\sigma$ with frequency dv in time dt:

$$I_v = \frac{dE_v}{\cos\Theta dA dt dv d\sigma}\,, \tag{1}$$

These simulated photon packets, or just photons, interact according to the attributes of their initialization along with the scattering and absorption cross sections. These cross sections are defined by the energy per second per frequency per solid angle that is lost by either scattering or absorption. The energy lost is equal to the specific intensity, I_v multiplied by the cross section σ, with the cross section in units of area. The number of photons scattered per second can be modeled by the differential equation:

$$dI_v = -I_v n\sigma dl\,, \tag{2}$$

This equation is solved by:

$$I_v(l) = I_v(0)e^{-n\sigma l}\,, \tag{3}$$

This equation gives us the fraction of photons scattered or absorbed per unit length l. The faction of scattered photons thus scales by a factor of $e^{-n\sigma}$, so we refer to $n\sigma$ as the absorption coefficient. The inverse of the absorption coefficient gives us the average distance traveled by the photons in between interactions with the particles in the medium, which is called the mean free path of the photons and is equal to $1/n\sigma$. We can find the probability that a photon does interact over a length dl by multiplying dl by the absorption coefficient. To find the probability of an interaction not occurring over a length L, we can divide L into N sections each of length dl so the probability of no interaction occurring per length dl would be $1 - n\sigma dl$ where $dl = L/N$. To find the probability of an interaction not occurring for the full length L we set $N = \infty$ and raise $1 - n\sigma dl$ to the Nth power giving us:

$$P(L) = 1 - n\sigma L/N^N = e^{-n\sigma L}\,, \tag{4}$$

Where P is the probability of a photon not interacting over a length L. We can simplify this relation by replacing $n\sigma L$ with τ, the optical depth, which gives the number of mean free paths needed to cross a distance L. More generally, we can express the optical depth as:

$$\tau = \int_0^L n\sigma ds\,, \tag{5}$$

When a photon does interact with a particle, it can either be absorbed or scattered. The probability of being aborbed versus scattered is determined by the albedo of the system, which is defined by the equation:

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

Here, n is the number density and σ is the cross section, with the subscript s for scatters and a for absorbers. If a photon is absorbed it is destroyed in our simulation, . If a photon is scattered, it will travel in a different direction which is given by the phase function of the particle which scatters the photon. The phase function is given by the following equation:

$$\int_{-1}^{1} P(\mu)d\mu = 1\,, \tag{7}$$

The phase function for scattering by dust and electrons can be approximated as:

$$P(\mu) = \frac{3}{8}(1+\mu^2)\,, \tag{8}$$

With $\mu = \cos(\chi)$ where χ is the scattering angle. Isotropic scattering is modeled by randomly sampling sampling an angle ϕ from 0 to 2π and an angle μ from -1 to 1, meaning the photon is randomly given a dirction out of 4π steraidians. This is carried out by the following equations:

$$\phi = 2\pi\xi, \quad \mu = 2\xi - 1\,, \tag{9}$$

Here, ξ is a random number between 0 and 1.

The Monte Carlo radiative transfer method tracks the lives of photons as they travel through a specified medium which has a boundary. When a photon reaches this boundary, it is recorded and destroyed. To better visualize the simulations, photons are placed in groups or bins depending on where they hit the boundary and the number of photons which hit each section of the border is counted, or the bins are filled by the photons which hit the specified region on the boundary which is assigned to the each bin. We split the boundary into section based on the angles μ and ϕ. Here, μ is shorthand for $\cos\theta$. We create bins using μ and ϕ so that the solid angle for each bin is equal in surface area. For our simulation, we created 64 bins by splitting both μ and ϕ into 8 equal sections.

We use the Monte Carlo radiative transpher method to create and track photons through their life in the simulation. However, in order to best model dust scattering, we use a scattering matrix which dictates how each photon will be scattered given the initial conditions of the photon. The scattering matrix we use follows Chandrasekhar (1960) and we chose to use this way of calculating the scattering matrix as it allows for multiple different types of scattering, although we will be focusing on dust scattering here. To understand the scattering matrix, it is critical to understand elliptical photon polarization and the stokes parameters.

Elliptical polarization is named as such because the electric field vector of polarized radiation will trace out an ellipse if looked at in a fixed plane that is perpendicular and intersecting the direction the wave is travelling. Essentially, if an elliptically polarized wave was travelling directly out of the page, the tip of the polarization vector would trace an ellipse on the page. Now let us look at how we describe polarization with equations. The electric field can be expressed as a combination of two electric fields, one in the x direction and one in the y direction describing its polarization as:

$$E_x = \varepsilon_1 \cos(\omega t - \phi_1), \quad E_y = \varepsilon_2 \cos(\omega t - \phi_2)\,, \tag{10}$$

If we want to look at these components with respect to their principal axes x' and y', we can generalize the above equations by tilting the x and y axis by an angle X so that they can be written as:

$$E'_x = \varepsilon_0 \cos(\beta)\cos(\omega t), \quad E'_y = -\varepsilon_0 sin(\beta) sin(\omega t)\,, \tag{11}$$

Where β is between $-\pi/2$ and $\pi/2$. Arranging the equations in this way makes it easier to see the magnitude, which is $\varepsilon\|\cos(\beta)\|$ respectively. We can see that dividing E'_x and E'_y by their respective magnitudes, squaring and adding them together will give us 1, meaning these equations are normalized. Whether β is negative or positive will determine the direction in which the ellipse is traced out on the plane perpendicular to the direction of travel, with positive β values leading to a clockwise tracing and negative β resulting in counter clockwise tracing. Radiation with positive β values is thus called right-handed elliptical polarization and left-handed for negative β.

If we now want to relate our equations for the electric field back to the equations defining the principal axes of the ellipse, we rotate the X and Y axes by an angle χ which gives us:

$$E_x = \varepsilon_0(\cos\beta\cos\chi\cos\omega t + sin\beta sin\chi sin\omega t)\,, \tag{12}$$

$$E_y = \varepsilon_0(sin\beta\cos\chi\cos\omega t - sin\beta\cos\chi sin\omega t)\,, \tag{13}$$

In order to make the above equations equations to look like the previous equations, we set the following equal:

$$\varepsilon_1 \cos\phi_1 = \varepsilon_0 \cos\beta\cos\chi\,, \tag{14}$$

$$\varepsilon_1 sin\phi_1 = \varepsilon_0 sin\beta sin\chi\,, \tag{15}$$

$$\varepsilon_2 \cos\phi_2 = \varepsilon_0 \cos\beta sin\chi\,, \tag{16}$$

$$\varepsilon_2 sin\phi_2 = -\varepsilon_0 sin\beta\cos\chi\,, \tag{17}$$

To simplify these equations and make them easier to work with, we solve for ε_0, β and χ This will yield the Stoke parameters which are defined by:

$$I = \varepsilon_1^2 + \varepsilon_2^2 = \varepsilon_0^2\,, \tag{18}$$

$$Q = \varepsilon_1^2 - \varepsilon_2^2 = \varepsilon_0^2 \cos 2\beta\cos 2\chi\,, \tag{19}$$

$$U = 2\varepsilon_1\varepsilon_2\cos(\phi_1 - \phi_2) = \varepsilon_0^2\cos 2\beta sin2\chi\,, \tag{20}$$

$$V = 2\varepsilon_1\varepsilon_2 sin(\phi_1 - \phi_2) = \varepsilon_0^2 sin2\chi\,, \tag{21}$$

These four parameters translate to specific properties of the polarized photon. I is proportional to the total energy flux, and the four stokes parameters are typically normalized so that I is exactly equal to the intensity, as this can make things simpler. V is the circularity parameter which is the ratio of the principal axes of the ellipse, meaning that negative V corresponds to left-handed polarization and positive V gives right-handed polarization. Additionally, V = 0 means it is linearly polarized. Q and U together specify the orientation of the ellipse relative to the x-axis, assuming the direction of travel is in the z direction. Thus, for circular polarization, Q=U=0.

The parameters I, Q, U and V make up the stokes vector, S. If a photon has a stokes vector S' and scatters into the direction (θ, ϕ), the photon's stokes vector will change due to the scattering to a new stokes vector, S, given by the equation:

$$S = L(\pi - i_2)RL(-i_1)S'\,, \tag{22}$$

Here, L is a Mueller matrix which rotates to and from the observers frame.

$$L(\psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\psi & sin2\psi & 0 \\ 0 & -sin2\psi & \cos 2\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Here, R is the scattering probability matrix in the frame of the particle with respect to the direction of the photon. The matrix R has elements:

$$R(\Theta) = \begin{bmatrix} P_1 & P_2 & 0 & 0 \\ P_2 & P_1 & 0 & 0 \\ 0 & 0 & P_3 & -P_4 \\ 0 & 0 & P_4 & P_3 \end{bmatrix}$$

For dust scattering, the matrix R is made up of the the following elements, where Θ is the scattering angle measured from the incident photon direction:

$$P_1 = \frac{1 - g^2}{(1 + g^2 - 2g\cos\Theta)^{3/2}}\,, \tag{23}$$

$$P_2 = -p_1 P_1 \frac{1 - \cos^2\Theta}{1 + \cos^2\Theta}\,, \tag{24}$$

$$P_3 = P_1 \frac{2\cos\Theta}{1 + \cos^2\Theta}\,, \tag{25}$$

$$P_4 = -p_c P_1 \frac{1 - \cos^2\Theta_f}{1 + \cos^2\Theta_f}\,, \tag{26}$$

In the above matrix elements, g is the scattering symetry parameter which ranges from 0 to 1, *pl* is the peak linear

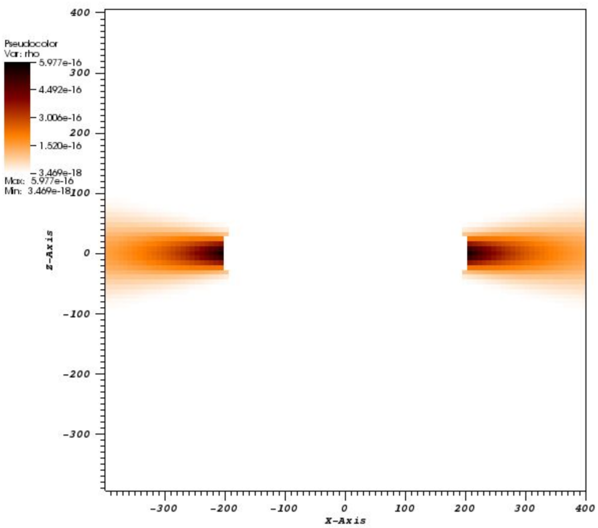

Figure 1. The above plot shows the density distribution of the protoplanetary disc from an edge on view, meaning that the protoplanetry disc is aligned on the x-axis, and we are looking at the edge of the disc. The density is highest at the inner most ring of the disc, seen above in black, and the density decreases outwardly in both the x and y directions, which is typical for protoplantary discs.

polarization, pc is the peak circular polarization and $\Theta_f = \Theta(1 + 3.13s \exp[-7\Theta/\pi])$.

3 DUST SCATTERING SIMULATION AND COMPARISON

3.1 The simulation

Using the Monte Carlo radiative transfer method, we added a dust scattering module to the Athena++ code using the equations described above, and ran this code using a problem generator for a simulation of a binary star system with stars of different luminosities which are encircled by a protoplanetary disc.

3.2 Comparison to FORTRAN code output

In order to test whether the dust scatter program was running correctly, we ran the same simulation using a FORTRAN code that has a dust scattering program that has been tested and is confirmed to give the proper outputs. We then made a program to graph the outputs of the Athena++ and FORTRAN outputs on the same plot, below are the results. The Athena++ code output is shown by blue circles and the FORTRAN code is shown by red diamonds.

By comparing the output of the Athena++ code to the output of the FORTRAN code, we see there is a noticeable discrepancy in every graph, meaning that something is going wrong in the Athena++ code. The discrepancy tends to decrease as $\cos\theta$ approaches 1, which might imply that the more scattering that occurs the less accurate the Athena++ code is, since at values of $\cos\theta$ about equal to 1, the disc is most dense. This could be a consequence of the Athena++ code having a bug for photons which are not scattered, which would be more common at lower values of $\cos\theta$.

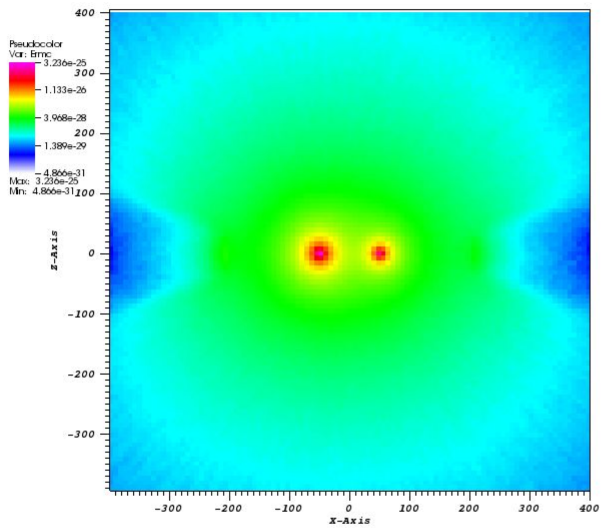

Figure 2. The above plot shows the energy density distribution for the output of the Athena++ code, from an edge on view, which is the same orientation as the density distribution in figure 1. The two bright point sources near the center are the two stars which is where all the photons are initialized and travel outward from one of the two stars. The energy density is lowest on the outside of the protoplanetary disc, with the minimum on the same axis as the disc, which is expected, as photons traveling directly through the disc have the highest chance of losing energy and being absorbed or scattered into another direction.

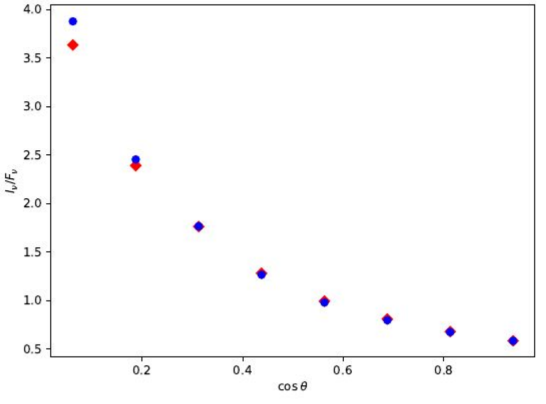

Figure 3. The above plot shows the flux of both the Athena++ code and the fortran code plotted against the polar angle. The two outputs have an almost perfect agreement for $\cos\theta$ greater than 0.3, but the Athena++ code's intensity is higher than it should be for low values of $\cos\theta$, and the discrepancy worsens as $\cos\theta$ approaches 0.

3.3 Correcting the code

To correct the code, we need to pinpoint what exactly the issues are with the code. We could start by using different

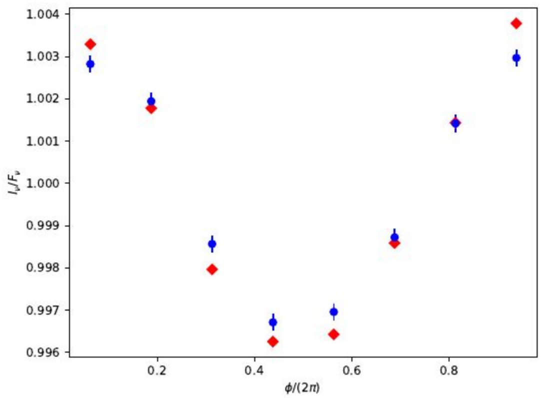

Figure 4. The above plot shows the flux of both the Athena++ code and the fortran code plotted against the azimuthal angle. The Athena++ code output has a larger amplitude of fluctuation than the FORTRAN output, meaning the Athena++ code has highs that are too high and lows that are too low.

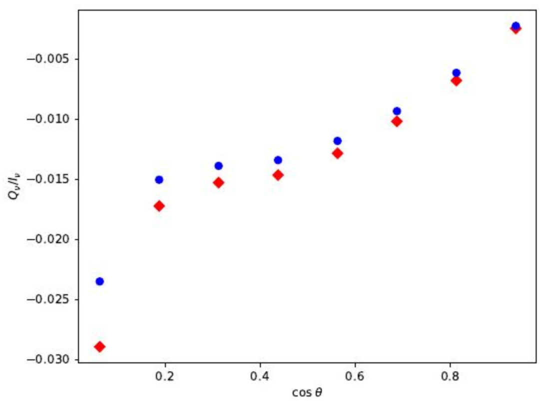

Figure 5. The above plot shows the parameter q, which is equal Q divided by I, for both the Athena++ code and the fortran code plotted against the polar angle. Similar to figure 3, the agreement between the FORTRAN and Athena++ code is best when $\cos\theta$ is close to 1 and gets worse as $\cos\theta$ approaches 0.

problem generators that create simpler dust scattering simulations, such as having all the photons initialized with the same direction and polarization, then sending them through a uniform plane of dust. By graphing the output of this much simpler simulation, it could be easier to see how exactly the Athena++ code and the FORTRAN code are differing. In addition, it could be helpful to add some parameters such as only letting each photon scatter once to simplify the simulation.

3.4 Adding to the code

Once the discrepancy is found and corrected, we could improve on this code by adding a frequency dependence which the code would take into account and make the simulation more realistic. Since you will always find objects emitting photons with a spectrum of frequencies in nature, it would be beneficial to make this program be able to model effect that different frequencies of the photons has on how they are

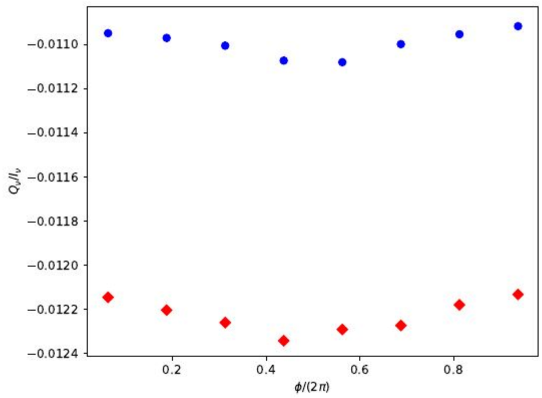

Figure 6. The above plot shows the parameter q, which is equal Q divided by I, for both the Athena++ code and the FORTRAN code plotted against the azimuthal angle. Although the Athena++ output seems to have a q value that is far too high, the two outputs follow a very similar trend, meaning the Athena++ code could be accurately calculating q but is missing a constant somewhere which is creating the discrepancy.

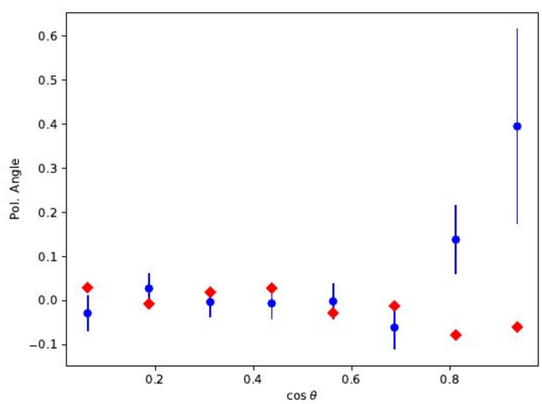

Figure 7. The above plot shows the polarization angle, which is the direction the net polarization of the photons is pointing, for both the Athena++ code and the fortran code plotted against the polar angle. There is a discrepancy throughout, however the discrepancy becomes much worse as $\cos\theta$ approaches 1.

scattered. For example, we could make the simulation take the temperature of the two stars as a parameter and then create a realistic spectrum for each star based on its given temperature.

3.5 Conclusion

Our code utilizes the Monte Carlo raidative tranfer method to simulate randomized photons which then travel and scatter off dust based on the above equations. Although our dust scattering module did not exactly match the tested FORTRAN code, our code could be debugged and used for a dust scattering module for the Athena++ grid framework, which could yield useful results in creating simulations for a variety of different astronomical systems. Simulations are great tools to help understand how the universe works and how astronomical systems might appear when viewed from Earth.

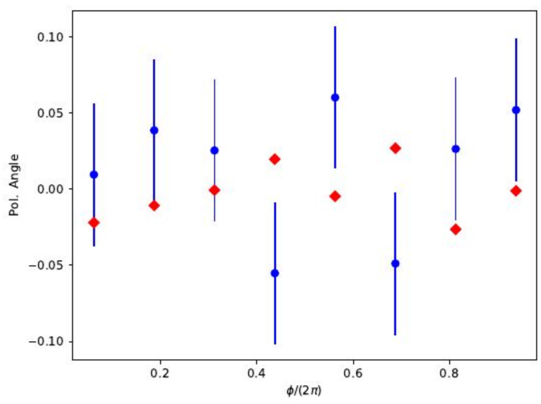

Figure 8. The above plot shows the polarization angle, which is the direction the net polarization of the photons is pointing, for both the Athena++ code and the fortran code plotted against the azimuthal angle. The discrepancy between the FORTRAN and Athena++ outputs is large throughout, however the two outputs seem to have a somewhat similar trend.

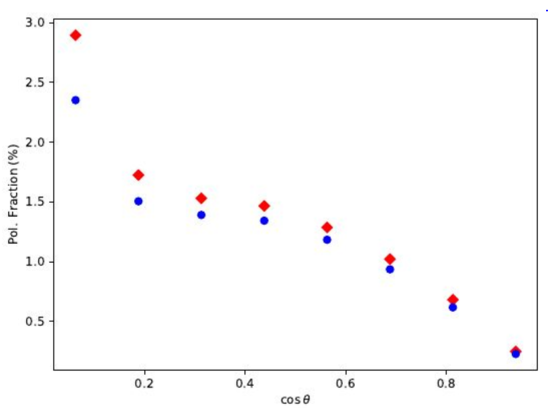

Figure 9. The above plot shows the fraction of photons which are polarized, for both the Athena++ code and the fortran code plotted against the polar angle. Mirroring figure 3 and figure 5, the Athena++ output has a larger discrepancy as $\cos\theta$ approaches 0.

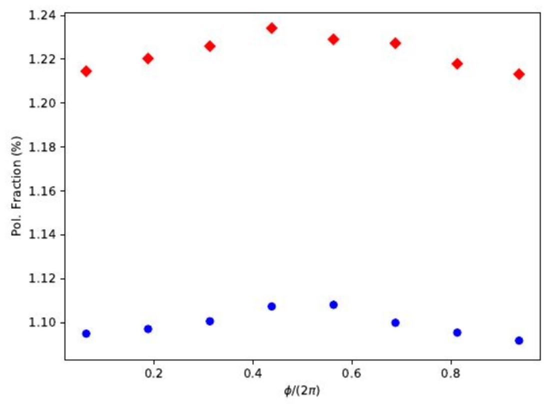

Figure 10. The above plot shows the fraction of photons which are polarized, for both the Athena++ code and the fortran code plotted against the azimuthal angle. Similar to figure 8, although both outputs follow a similar trend, they differ by a constant at each point.

REFERENCES

Robitaille T. P., 2011, A&A, 536, A79
Stone J. M., Tomida K., White C. J., Felker K. G., 2020, ApJS, 249, 4
Whitney B. A., 2011, Bulletin of the Astronomical Society of India, 39, 101
Williams J. P., Cieza L. A., 2011, ARA&A, 49, 67