Dust Scattering Polarization Modeling

Dissipatone Lucem Pulvis Magneticus Unus

Kenji Emerson[1] and Ramprasad Rao (Mentor)[2]

[1]University of Hawaii at Hilo
[2]SMA - Smithsoanian Institute

July 13, 2018

Abstract

Abstract!

1 Introduction

2 Dated Log

2.1 2018 June 21

Meeting Notes:

First off, the main introduction to the scope of the project was introduced.

Light scattered off of dust is polarized preferentially towards the plane shared by its longest side. In this case, dust is modeled as an oblate shaped particle of varying size.

Thus, the alignment of the polarization of light coming from a dust based object can reveal information about its rotation. There are two different influences towards the orientation of the dust.

Magnetic fields in the dust align the particles of the dust such that the plane shared by its longest side is perpendicular to the B-field. Consider the longest axis of a grain a dust to be the y-axis; the B-field would be in the positive or negative z-axis direction (the specifics towards the exact orientation of the B-field is beyond the scope).

Next, there are two similar tasks to be completed.

1. Create a function that generates a noisy (non-smooth) gaussian function given some input parameters. The gaussian function should be fully modifiable.
2. Create the reverse function: a function that, when given data points from a gaussian, can fit a gaussian function to it.

Both of these functions have been completed as of this date, written in Python. The gaussian equation implemented is similar to that on Wikipedia:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (1)$$

In order to share the functions and code work done with the Mentor, a Github repository has been created. It can be found at `https://github.com/psmd-iberutaru/Akamai_Internship`.

2.2 2018 June 22

Meeting Notes:

Modifications to the gaussian program is desired. In particular, the program should be somewhat more robust. In essence, the following modifications and tests to the fitting function is as follows.

1. Make the code more robust, account for any and all cases, and if it is the case that there is some unlikely error, make sure to throw an exception.
   - Using Monte Carlo methods, implement some as-

surance that the program data is real and complete.

– Change the noise of the program, allowing for the testing of varying noise values.

Other tasks include:

Allow for the finding of multiple gaussian functions. It is assumed that there will be a generator function for multiple gaussian functions too.

After these functions have been done, it is possible that this will soon evolve to curve fitting in 2D (where x,y are inputs for a z output).

A field trip on Monday, 2018 June 25, to Mauna Loa's Yuan-Tseh Lee Array (YTLA), a radio telescope, is planned. Departure is expected to be at 14:00, arrival back at the SMA facility is expected to be 19:00 at the latest.

2.3 2018 June 26

The gaussian fitting functions have been completed. They work accurately unless the separation between two gaussian functions is such that the peak of one is obscured in the body of the other (in the case of multi-gaussian fitting).

In the case of multi-gaussian fitting, the peaks of each gaussian (if not already obscured) is detected using `scipy` based peak finding algorithms. To prevent the intrusion of other gaussians, the domain that the data points for any one particular gaussian is determined by the `peak_widths` function, with the start and ending points of the gaussian being determined by the function's 2nd and 3rd returns in index space.

Research has also been done to also fit Bessel functions. Bessel functions are a class of functions that satisfy the Bessel ODE equation.

$$x^2 \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + x \frac{\mathrm{d}y}{\mathrm{d}x} + \left(x^2 + \alpha^2\right) y = 0 \tag{2}$$

For α being the order of the Bessel equation.

The solution(s) to Bessel's equations, the Bessel functions that we are interested in, come in one of two flavors.

Bessel functions of the first kind are used when x is finite and definable at the origin for either integer or positive α ($\{\alpha | \alpha \in \mathbb{Z} \vee \alpha > 0\}$) or if y diverges as x approaches 0 for negative non-integer α ($\{\alpha | \alpha \notin \mathbb{Z} \wedge \alpha < 0\}$).

$$J_\alpha(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\Gamma(k+\alpha+1)} \left(\frac{x}{2}\right)^{2k+\alpha} \tag{3}$$

Bessel functions of the second find are used when $x = 0$ is a singularity. In this case, the solutions are in a few forms.

For non-integer α ($\{\alpha | \alpha \notin \mathbb{Z}\}$):

$$Y_\alpha(x) = \frac{J_\alpha(x)\cos(\alpha\pi) - J_{-\alpha}(x)}{\sin(\alpha\pi)} \tag{4}$$

For an integer order n ($\{n | n \in \mathbb{Z}\}$):

$$Y_n(x) = \lim_{\alpha \to n} \left[Y_{\alpha(x)}\right] \tag{5}$$

It is, of course, the case that `scipy` already has a family of functions outlining the numerical implementation of these functions. The Bessel functions, and the generating/fitting functions are written using those family of equations as the basis of computation.

Meeting Notes:

We search for mostly the deformation of the polarization region.

Future tasks:

Bessel function and gaussian function fitting, where both are linearly combined, i.e. $y = aG(x, \mu, \sigma, h) + bB(x, \alpha, \mu, h)$.

2.4 2018 June 28

:

There has been some interesting trends or quirks found in the properties of Bessel functions of the first kind. These quirks prevent Python from plotting it very effectively, however, it is the case that there are workarounds.

These are by no mean always true, but, for a large range of values (to numerical accuracy and computational usefulness), they seem to be true.

For $\alpha \in \mathbb{Z}$:

$$J_{-\alpha}(x) = (-1)^\alpha J_\alpha(x)$$

This is a true relationship as it also proves that $J_\alpha(x)$ and $J_{-\alpha}(x)$ are not linearly independent.

For $\alpha \in \mathbb{Z}$:

$$J_\alpha(x) = \begin{cases} J_\alpha(-x) & \text{if } \alpha \in \mathbb{Z}_{\text{even}} \\ -J_\alpha(-x) & \text{if } \alpha \in \mathbb{Z}_{\text{odd}} \end{cases}$$

Thus Bessel functions of the first kind for integer order is an even function for even order, or an odd function for odd order.

For $\alpha \notin \mathbb{Z} \wedge \alpha \in \mathbb{R}$:

$$\mathrm{Re}\left(J_\alpha(x)\right) = \mathrm{Im}\left(J_\alpha(-x)\right)$$
$$\Downarrow$$
$$J_\alpha(-x) = J_\alpha(x)\mathrm{i}$$

For $\alpha = \frac{1}{2}k$ For: $\{k \in \mathbb{Z} | k \neq 0\}$:

$$\mathrm{Re}\left(J_\alpha(x\mathrm{i})\right) = \mathrm{Im}\left(J_\alpha(x\mathrm{i})\right)$$

Meeting Notes:

First, a small discussion was held about the Mentor's meeting times. In general, between the times of 08:30 and 10:30, he will be in a meeting and unavailable. Otherwise, if he is not present in his office, he may be downstairs of the SMA building in one of the labs. Emailing is preferred by both parties, however, if necessary, his cell phone number is recorded in contacts.

Based on the discussions with my Mentor, it does not seem that fractional order Bessel functions are needed. We can stick to integer Bessel functions ($\alpha \in \mathbb{Z}$).

In fact, so far, the main functions that are going to be used (i.e. periodic functions)

Basic trigonometric functions: $\sin(x)$, $\cos(x)$

2. Gaussian functions (see Equation 1 [Page 1]:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

3. Bessel functions of the first kind ($J_\alpha(x)$) of integer order ($\alpha \in \mathbb{Z}$), see Equation 3 [Page 2]:

$$J_\alpha(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\Gamma(k+\alpha+1)} \left(\frac{x}{2}\right)^{2k+\alpha}$$

4. Error functions $\operatorname{erf}(x)$:

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} \exp(-t^2)\, \mathrm{d}t = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp(-t^2)\, \mathrm{d}t$$

2.5 2018 July 9

Today, two main projects were finished up. A programable form of the equations of the magnetic fields in a 'hourglass' shape for a cylindrical coordinate system was completed. Although it was attempted to bring the two dimensional equations ($\vec{B} = f(r,z)$) into three dimensions via an invariant *phi* parameter ($\vec{B} = f(r,\phi,z)$), it is so far unsuccessful in visualization, but seems to be alright in terms of actual functionality.

The equations for the magnetic field is given by [**Ewertowski_Basu_2013**], where h is a free parameter and *not* Planck's constant.

$$B_r = \sum_{m=1}^{\infty} k_m \sqrt{\lambda_m} J_1(\sqrt{\lambda_m} r) \left[\operatorname{erfc}\left(\frac{\sqrt{\lambda_m}h}{2} - \frac{z}{h}\right) \exp\left(-\sqrt{\lambda_m} z\right) - \operatorname{erfc}\left(\frac{\sqrt{\lambda_m}h}{2} + \frac{z}{h}\right) \exp\left(\sqrt{\lambda_m} z\right)\right] \tag{6}$$

$$B_z = \sum_{m=1}^{\infty} k_m \sqrt{\lambda_m} J_0(\sqrt{\lambda_m} r) \left[\operatorname{erfc}\left(\frac{\sqrt{\lambda_m}h}{2} + \frac{z}{h}\right) \exp\left(\sqrt{\lambda_m} z\right) + \operatorname{erfc}\left(\frac{\sqrt{\lambda_m}h}{2} - \frac{z}{h}\right) \exp\left(-\sqrt{\lambda_m} z\right)\right] \tag{7}$$

For $a_{m,1}$ is the mth positive root of $J_1(x)$, whereas $a_{m,1} > a_{m+1,1}$, such that λ_m is given by, provided R the radius of the protoplanetary disk of the system simulation.

$$\lambda_m = \left(\frac{a_{m,1}}{R}\right)^2 \qquad \text{For: } m \in \mathbb{R}$$

The paper also defines the coefficient k_m as,

$$k_m \triangleq \frac{2h\pi^{\frac{3}{2}} \exp\left(\frac{h^2\lambda_m}{4}\right)}{cR^2\sqrt{\lambda_m}\left(J_2(\sqrt{\lambda_m}R)\right)^2} \int_0^R f(\xi) J_1(\sqrt{\lambda_m}\xi)\xi\, \mathrm{d}\xi$$

with some custom defined function $f(x)$; but, for all intents and purposes, we use k_m also as a parameter as does the paper.

2.6 2018 July 10

The presentation to the Akamai Internship program is due tomorrow. Recent things have came up about the motivation of this project and the purpose and contributions that magnetic fields make to star forming regions.

One of the big issues with star formation is that of the conservation of angular momentum. By the time a protoplanetary disk can form into a star, the total angular momentum of the system is estimated to be too great; possibly tearing the star apart. Thus, in order to stop the star from spinning too much, it is believed that magnetic breaking is involved (see Figure 1 [Page 4] for a mechanical demonstration, else see `https://en.wikipedia.org/wiki/Magnetic_braking`).

A second reason for the research into the magnetic field of protoplanetary regions (star forming regions) is because magnetic fields prevent the collapse of clouds, preventing star formation. For a gas cloud with less mass than $M_{B,cr}$, given by

$$\begin{aligned} M_{B,\mathrm{cr}} &= 0.12\frac{\Phi}{\sqrt{G}} \\ &\simeq 10^3 \left(\frac{B}{30\mu G}\right)\left(\frac{R}{pc}\right)^2 M_\odot \end{aligned}$$

For masses below this critical limit, $M_{\mathrm{cloud}} < M_{B,\mathrm{cr}}$, the magnetic field can support the cloud, preventing it from collapsing and turning it into a star.

2.7 2018 July 13

Today two or three main things have been completed. First off, the implementation of the linear combination of a circular function and the hourglass function has been completed.

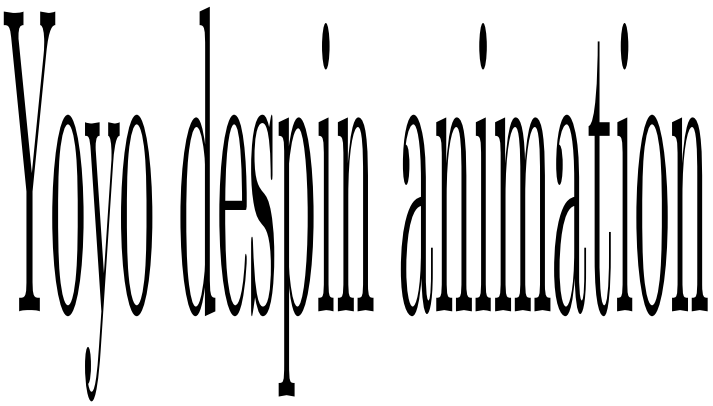

Figure 1: This is a small animation that demonstrates the lessening of a ship's angular momentum because of the objects on the tethers stealing the momentum. Ionized particles caught in protoplanetary stellar magnetic fields act similar to objects on tethers, slowing down the rotation of the protostar.

There has been the observation that this hybrid function is not symmetric with respect to the yz-plane projection. This is explained by the circular function being rotationally symmetric along the x-axis, while the hourglass function is symmetric along the z-axis. These competing symmetries lead to the asymmetry of a reflection across the xz-plane. The reflective symmetry is restored if the contribution of the circular function is 0, and the rotational symmetry along the x-axis for the circular function is restored if the contribution of the hourglass function is 0.

Secondly, an implementation of a function to return the angle of polarization given two Stoke numbers/parameters (see `https://en.wikipedia.org/wiki/Stokes_parameters`). In particular, assuming the definition of the polarization ellipse as given in the article, the functional form is given as,

$$\phi = \frac{1}{2}\arctan\left(\frac{U}{Q}\right) \tag{8}$$

for the Stoke parameters of $S_1 = Q$, $S_2 = U$. Currently, this is actually implemented via the arctan2 function.

Next, in order to compute a line integral of a sightline through a three dimensional object like a gaseous cloud, a function was written to both determine the bounds for the regions of the sightline is within the cloud and not, and to also compte the integral. The cloud acts only as the bounds for the integration constrained by the path of the sightline. The medium integrating over is a field. As of current, a single parameter field is working. It is not known how this function will work with multi-parameter return functions. It does not seem like a difficult fix. It might be best to generalize the function (but it is too early to tell). The cloud, field function, and sightline are all modifiable according to user specifications.

A small thing to note as of right now, the programs are not very user-friendly. Some effort will need to be put into making this entire codebase into something much more user-friendly. Perhaps not as large as a themed module, but enough to be rather useful. In fact, some of the older functions like Bessel fitting and the gaussian fitting functional groups are considered to be obsolete because of their shoddy implementations.