

Dust Scattering Polarization Modeling

Dissipatone Lucem Pulvis Magneticus Unus

Kenji Emerson¹ and Ramprasad Rao (Mentor)²

¹University of Hawaii at Hilo
²SMA - Smithsonian Institute

June 27, 2018

Abstract

Abstract!

1 INTRODUCTION

2 DATED LOG

2.1 2018 JUNE 21

eters. 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.

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.

Create a function that generates a noisy (non-smooth) gaussian function given some input param-

Both of these functions have been completed as of this date, written in Python.

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.

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 assurance 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).

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.

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.

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 + \alpha^2) y = 0 \quad (1)$$

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 ($\{\alpha | \alpha \notin \mathbb{Z} \wedge \alpha < 0\}$).

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

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

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

$$Y_\alpha(x) = \frac{J_\alpha(x) \cos(\alpha\pi) - J_{-\alpha}(x)}{\sin(\alpha\pi)} \quad (3)$$

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

$$Y_n(x) = \lim_{\alpha \rightarrow n} [Y_\alpha(x)] \quad (4)$$