How good is dust emission as a tracer of star forming regions in molecular clouds?

Tomas James
Student ID: 1158976

Supervisor: Dr. P. C. Clark

April 9, 2016

Abstract

*A data reduction pipeline to analyse Herschel images of early star forming regions to probe the dust emission characteristics of the data was written in the Python programming language. Initially a simple scenario of a spherical molecular cloud was simulated using RADMC-3D (***RADMC-3D***), with bespoke input files generated using the Python programming language. This simulation was performed across the 3 SPIRE (***SPIRE***) wavebands centered on 250 µm, 350 µm and 500 µm as well as the 3 PACS (***PACS***) wavebands centered on 72 µm, 103 µm and 167 µm. This data was then 'degraded' by accounting for the instrument's transmission curve to better simulate an object observed by Herschel. This data was then subsequently analysed by plotting Spectral Energy Distributions (SEDs) on a pixel by pixel basis to recover the column density N and temperature T at each pixel. The goodness-of-fit was assessed using a χ^2 test. 2-dimensional maps of these regions was then constructed and the recovered values of N and T compared to the initial input values of density and temperature. Finally this machinery was applied to data from the SPH simulation Arepo to emulate real Herschel data.*

Contents

I. Introduction

I. Molecular clouds

A molecular cloud is a dense region of the Interstellar Medium (ISM), composed primarily of molecular Hydrogren (H_2) at temperatures $< 20\ K$ (**dustopacity**). Despite the cloud being primarily a gas of H_2, a small (by mass) dust component is also present. This dust mass component is approximately 1% the gas mass according to **noise** however despite this minimal presense dust remains a crucial component in the evolution of the cloud and the subsequent star forming process.

/Users/tomasjames/Documents/University/Project/ZiggyStarDust/Report/Final/img/rho.jpg

Figure 1: *An IRAS image showing the Rho Opiuchi cloud complex (***rho***).*

Figure **??** shows the Rho Ophiuchi cloud complex as imaged by the Infra-red Astronomical Satellite (IRAS). Rho Opiuchi is the nearest active star forming molecular cloud complex to the Milky Way[1]. As a result of this, Rho Ophiuchi provides an unprecedented oppurtunity to study the sites of early star formation at high resolution. This is in spite of the large uncertainty in estimates of the distance to Rho Ophiuchi (**rho-dist**).

Much like the ISM, molecular clouds are composed of gas and cosmic dust, however a molecular cloud differs from the ISM in that significantly greater densities are found within a molecular cloud than that of the ISM. The dust density in a molecular cloud is thought to be around $10^5\, g\, cm^{-3}$, whilst the dust density in the surrounding ISM is thought to be around $10^2\, g\, cm^{-3}$.

The molecular cloud is also colder than the surrounding ISM: the temperature is approximately $15\ K$ in the ISM whilst the temperature in the molecular cloud is approximately $10\ K$. The warmer ISM bathes the cooler cloud, resulting in a temperature gradient; the ISM heats the cloud from the outside in, resulting in a temperature that increases with distance.

Figure **??** illustrates the intense infra-red emission within molecular clouds owing to the active star forming occurring within them. It also illustrates the turbulence and chaos within the cloud, with both filamentary structure and (potentially) magnetic fields running through the cloud itself. These filaments are regions of higher density, and as **evo-mol** states "One fundamental characteristic of molecular clouds is that they are not, as has sometimes been assumed, isolated billiard balls moving about independently in space, but instead are just dense condensations in more widely distributed, mostly atomic gas. Although molecular clouds may often appear to have sharp boundaries, these boundaries do not represent the edge of the matter distribution but just rapid transitions from the molecular gas to the surrounding atomic gas, which is distributed in extended envelopes that typically have comparable mass (Blitz 1988, 1991)". These filaments act as preferential sites for star formation (**filaments**).

These star forming regions within molecular clouds originate from locations of gravitational instability that lead to subsequent collapse (**jeans**). A stable cloud (or portion of a cloud) is in hydrostatic equilibrium, such that the force due to gravity is balanced by the force due to gaseous pressure from the gas within the cloud. This equilibrium state can also be regarded as a virialised state such that $2K + U = 0$, where K is the total kinetic energy of the system and U is the total potential energy of the system. The cloud begins to collapse when the gravitational force is greater than the force due to the hydrostatic pressure. This collapse continues unimpinged until such a time that another opposing force can halt the collapse. The initial collapse is triggered when a region within the molecular cloud has a mass that exceedes the Jeans' mass given by Equation I (**lecture**).

[1] **rho-dist** estimates the range of distances as being between 125pc-165pc - a range of almost 40%.

This is known as the Jeans Criterion.

$$M_J = \left(\frac{5kT}{GM}\right)^{\frac{3}{2}} \left(\frac{3}{4\pi\rho}\right)^{\frac{1}{2}} \tag{1}$$

Within Equation I, k is the Stefan-Boltzmann constant, T is the temperature of the region, G is the Universal Gravitational Constant, M is the mass of Hydrogen in the region and ρ is the density of the region prior to collapse. The collapse of an M_J mass region occurs according to the free-fall time, t_{ff} described in Equation I (**krumholz**).

$$t_{ff} = \sqrt{\frac{3\pi}{32G\rho}} \tag{2}$$

Variables within Equation I are as they were defined in Equation I.

Interestingly the onset of collapse is also the onset of gravitational binding. According to **bound** the majority of a molecular cloud remains gravitationally unbound, further reinforcing the idea stated in **evo-mol** that molecular clouds are diffuse, transiest structures. As **bound** explain, the idea that a molecular cloud is gravitationally bound

II. Star Formation

The star forming process is one of the most important processes in the cosmos, as star formation (and the subsequent evolution of the star) poses an unequivocal factor driving the evolution of the Universe forward.

Stars are essentially chemical foundaries, acting as the primary source of elements heavier than ^{2}He in the Universe. The production of ^{1}H is thought to have occured approximately 380,0000 years after the Big Bang (**peebles**), when the primordial quark-gluon plasma had cooled sufficiently to allow protons and neutrons (hereafter p and n) to bind and form atoms. ^{2}He, according to **bbc** also formed at this time however the fundamental particles required to form elements were produced within the first few seconds after the Big Bang. Whilst further production of ^{2}He can occur via the first fusion process, the proton-proton (pp) reaction described by Equation 3 (**synthesis**) is the primary source of ^{2}He in stars. This is in stark contrast to ^{1}H, which cannot be produced via a fusion process.

$$^1\mathrm{H} + {}^1\mathrm{H} \longrightarrow {}^2\mathrm{He} + \mathrm{e}^+ + \mathrm{v_e} + 0.42\,\mathrm{MeV} \tag{3}$$

Despite the lack of ^{1}H production, ^{1}H is still the most abundant element in the Universe, amounting to roughly 72% by mass according to **abundance** This relative prevalence of ^{1}H is known as a direct result of observations of the 21cm line parencite21cm. The 21cm line originates from the hyperfine splitting that occurs when the only electron in ^{1}H in the 1s orbital spin flips to become antiparellel with the spin of the nucleus. This produces the characteristic photon emission at $\lambda = 21cm$.

Fusion

The fusion process occuring within stars adapts and evolves with the star itself. After the star has undergone its initial fusion process described in Equation 3, it begins to cool.

^{56}Fe cannot be fused to create any heavier elements as this fusion requires more energy to fuse than the fusion would produce, thus acting as a net loss of energy. As a direct result of this, fusion stops within a star at ^{56}Fe.

III. Dust emission

Dust is a relatively small fraction of the total ISM mass, estimated as being only 1% according to **noise** Whilst the dust mass accounts for such a small mass component, it still presents an important role in forming stars. The large (by mass) gaseous component does not radiate in the infra-red, and is thought to not interact

Dust radiates as a modified black body approximated by Equation **??** (**noise**).

$$S_\nu = N\Omega\kappa_0 \left(\frac{\nu}{\nu_0}\right)^\beta B_\nu(T) \tag{4}$$

In Equation 4 S_ν is the flux density, N is the column density, Ω is the solid angle subtended by the beam, κ_0 is a reference dust opacity, ν is the frequency of the image, ν_0 is a reference frequency at which the reference opacity κ_0 was evaluated at, β is the dust spectral index and $B_\nu(T)$ is the frequency dependent Planck function. In terms of intensity:

$$I_\nu = \frac{N}{R_{dust-gas}}\mu m_p \left(\frac{\nu}{\nu_0}\right)^\beta B_\nu(T) \tag{5}$$

All variables in Equation 5 are as defined in Equation 4, with the addition of $R_{dust-gas}$. $R_{dust-gas}$ is the dust-to-gas ratio, here approximated as being $\frac{1}{100}$. Throughout this project, dust emission was quantified in terms of intensity rather than flux density, as intensity is independent of distance to the source of emission.

Dust properties such as the spectral index β and temperature T clearly have an effect on dust emission according to Equations 4 and 5. These parameters therefore effect the ability to detect radiation from within a molecular cloud.

IV. RADMC-3D

RADMC-3D is a 3-dimensional Monte-Carlo radiative transfer and ray tracing code developed by Cornelis Dullemond and written in Fortran 90 for radiative transfer and emission astrophysics in dusty environments (**RADMC-3D**). One of the newest additions to RADMC-3D is a Python wrapper, `radmc3dPy`, that allows RADMC-3D objects to be handled in Python.

RADMC-3D is able to compute dust emission intensity as well as Spectral Energy Distributions (SED) and spectra for a number of different continua. This project focussed only on dust continuum, neglecting gas. The dust emission intensity can be computed using 2 different methods: the thermal Monte-Carlo simulation, or an image ray trace. All computations require input files for RADMC-3D to read and use in its simulations. RADMC-3D 's Python port, `radmc3dPy`, is able to write these input files, however for this project a custom script was written such that file creation was independent of `radmc3dPy`.

V. Input files

As stated in Section IV, RADMC-3D requires a number of user defined input files in order to operate. These input files act as parameter spaces, defining boundary conditions along with step sizes for each given input files. Figure **??** shows a workflow of RADMC-3D's operation, including all files required in order to operate.

Figure 2: *A schematic representation of the RADMC-3D workflow.*

As Figure 2 illustrates, all input files to RADMC-3D have extension `.inp` whilst all output files have extension `.out`. Any intermediate files, such as those directly calculated by RADMC-3D, have extension `.dat`. *Most* input files' initial lines - the header information - contain quantitative information regarding the number of pixels/cells in the file, the pixel/cell width along with other identifiers that inform RADMC-3D whether to look for binary files, the number of dust species being considered and so on.

`amr_grid.inp` is the file used in order to create a spatial domain within the model. It does this by defining the start and end points of pixels in 3 dimensions: x, y and z. These definitions produce pixels, each of fixed width defined by the distance between pixel start and end points. RADMC-3D also has the ability to perform Adaptive Mesh Refinement, such that regions of interest can be split into more pixels (and have a smaller pixel width and therefore simulated higher resolutions) than, for example, regions of the ISM that do not require a higher resolution. Any further files that setup physical quantities draw directly from the structure of `amr_grid.inp`. They essentially 'insert 'a value of a physical quantity into each cell, therefore producing another parameter space.

dust_density.inp is the file that gives RADMC-3D its density information. It has the same structure as amr_grid.inp, however in place of the start and end pixel points that are found in amr_grid.inp, dust_density.inp defines the dust density (in *cgs* units) at each pixel. This therefore creates a density parameter space with which to work from. Similarly, dust_temperature.dat draws upon the same idea, however instead of defining dust densities, dust_temperature.dat defines the dust temperature (in *K*) at each pixel point. These 3 files therefore create 3 parameter spaces: spatial, density and temperature.

Another crucial file is the wavelength_micron.inp file. This input file defines the wavelengths over which RADMC-3D should compute. By default, wavelength_micron.inp runs from 0.1 µm to 1000 µm with 150 points. Dust emission intensity can then be determined at any wavelength between these two bounds. In reality however, computing dust emission intensity over such a wide wavelength range is not always necessary (or practical). RADMc-3D thus allows the user to define a camera_wavelength_micron.inp file. This file is designed to emulate the wavelengths available to a given telescope or filter.

VI. Thermal Monte-Carlo dust emission

The thermal Monte-Carlo simulation is able to compute dust temperatures using the technique described in **b&w** from user-defined dust densities, providing a source of flux (and therefore photons) exists. This source of flux can either be through a star in the image space or an Interstellar Radiation Field. The process used to compute the dust temperature, according to **manual** is as follows. The total luminosity of the source is first split into *nphot* photon packets. The code then proceeds to 'fire 'photons out from the flux source into the image space. These photons can then either scatter off of dust grains or be absorbed by one, thus immediately being re-emitted at a different wavelength and direction as defined in **b&w** Under scattering/re-emission the photon will travel through the image space as defined in amr_grid.inp - see Section V - entering different adjoining pixels as it does so. This increases the temperature of that pixel, again according to **b&w** This process of 'ping-ponging 'throughout the image space is continued until the photon packet leaves the image space. At this point, a new photon packet is 'fired 'and the process is repeated using the previously determined dust temperatures until no further photon packets remain. The dust temperature at this point is therefore the solution.

Naturally the more pixels an image space comprises then the longer the simulation takes to compute owing to the large number of pixels and therefore collisions encountered. Likewise the greater the dust density at a given point the longer the simulation takes to run owing to a similar effect. Normally, RADMC-3D would write the calculated dust temperatures to dust_temperature.dat however as the thermal Monte-Carlo simulation was not used in this project, RADMC-3D was supplied with idealised dust temperatures in the form of a pre-determined dust_temperature.dat file.

VII. Ray tracing dust emission

Conversely the image ray trace reads in both user-defined dust temperatures and dust densities, thus bypassing the computation of the dust temperatures as would have been the case in the thermal Monte-Carlo simulation. Again, according to **manual** the ray-trace follows, by default, a first-order integration where the source function, $S = j_\nu / \alpha$ and opacity (and therefore extinction α) are constant across the ray. This is described by Equation **??**.

$$I_{result} = I_{start}e^{-\tau} + (1 + e^{-\tau})S \tag{6}$$

??

I_{result} is the dust emission intensity of the cell, I_{start} is the dust emission intensity of the cell prior to the ray entering it and τ is the optical depth along the path of the ray through the cell. The optical depth τ is defined as $\tau = \alpha s$ where α is as defined earlier and s is the distance the ray travels through the cell. The option to perform second-order integration is also made possible by varying the source function S and extinction across the ray. Specifically S and α are fixed at the ingress and egress points of the ray. These points act as boundary conditions such that the variation in S amd α can then be linearly interpolated across the path of the ray through the cell.

The result, however, is the same in both cases: a ḿapóf the dust emission intensity in the image plane.

II. Methodology

I. Model setup

RADMC-3D includes a number of basic example scripts to familiarise the user with the software's operation and capabilities. As a result, they provide an excellent introductory exercise to using RADMC-3D. Examples of 1D and 2D thermal Monte-Carlo simulations were run in this regard, generating Figures 3 and 4.

Figure 3: *The RADMC-3D output for dust emission intensity computed using thermal Monte-Carlo simulation in 1D.*

Figure 4: *The RADMC-3D output for dust emission intensity computed using thermal Monte-Carlo simulation in 2D.*

In order to begin operation, RADMC-3D requires a number of user defined inputs that define parameter spaces. RADMC-3D takes these user defined inputs from a number of input files that have subtle differences depending on their intended uses. The examples were used to understand how RADMC-3D generated these files, as well as how the code used them and for what purpose. These examples were basic, consisting of both 1D and 2D sphere projections with user defined stars at their centres that ran thermal Monte-Carlo simulations to build dust temperatures.

To begin building 3D models outside of the example scripts, specific input files needed to be created. As stated, a custom Python script was written to handle this file creation. Traditionally, `radmc3dPy` would handle file creation at the model setup phase, however given that RADMC-3D is designed to be run from the command line it was determined that reducing the number of dependencies would allow greater compatibility with different machines. This is critical if the code was to be run on a machine that may not have `radmc3dPy` installed, such as a supercomputer for larger computations. This also resulted in some of `radmc3dPy's` model setup commands being made redundant. For example, `radmc3dPy.setup.problemSetupDust` sets up the model using input parameters that are parsed to a model file. This phase was handled solely by `datafilegen.py`, so this command was redundant. Instead, required parameters such as the number of pixels in each dimension was given to RADMC-3D when calling it from the command line using `os.system(radmc3d image loadlambda)`. Again, the image ray-trace could be initiated through Python via the `os.system(radmc3d image)` command. The `loadlambda` keyword informs RADMC-3D that it should use the `camera_wavelength_micron.inp` module.

All code was version controlled using GitHub and stored in a private repository throughout development. The repository containing all code can be found at http://www.github.com/tomasjames/ZiggyStarDust.