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 17, 2016

Abstract

A data analysis pipeline to analyse synthetic 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 (Dullemond, Juhasz, Pohl, Sereshti, Shetty, Peters, Commercon, and Flock n.d.), with bespoke input files generated using the Python programming language. This simulation was performed across the 3 SPIRE (Griffin et al. 2010) wavebands centered on 250 μm, 350 μm and 500 μm as well as the 3 PACS (Poglitsch and Altieri 2009) 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

Chapter 1

Introduction

1.1. Molecular clouds

A molecular cloud is a dense region of the Interstellar Medium (ISM), composed primarily of molecular Hydrogren (H_2) at temperatures $< 20\ K$ (Hildebrand 1983). 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 Shetty et al. (2009), however despite this minimal presense dust remains a crucial component in the evolution of the cloud and the subsequent star forming process.

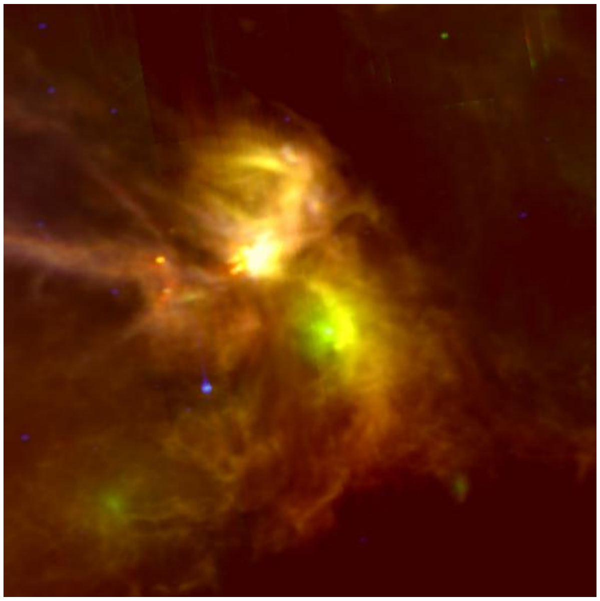


Figure 1.1: *An IRAS image showing the Rho Opiuchi cloud complex (Caltech n.d.).*

Figure 1.1 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 (Mamajek 2008).

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 from the centre of the cloud.

Figure 1.1 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 Larson (1994) 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 (Murphy and von Schill 2014).

[1] Mamajek (2008) estimates the range of distances as being between 125pc-165pc - a range of almost 40%.

These star forming regions within molecular clouds originate from locations of gravitational instability that lead to subsequent collapse (Jeans 1902). 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 1.1 (Pettini 2011). This is known as the Jeans Criterion.

$$M_J = \left(\frac{5kT}{GM}\right)^{\frac{3}{2}} \left(\frac{3}{4\pi\rho}\right)^{\frac{1}{2}} \tag{1.1}$$

Within Equation 1.1, 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 1.2 (Krumholz 2011).

$$t_{ff} = \sqrt{\frac{3\pi}{32G\rho}} \tag{1.2}$$

Variables within Equation 1.2 are as they were defined in Equation 1.1.

Interestingly the onset of collapse is also the onset of gravitational binding. According to Dobbs, Burkert, and Pringle (2011), the majority of a molecular cloud remains gravitationally unbound, further reinforcing the idea stated in Larson (1994) that molecular clouds are diffuse, transiest structures.

1.2. Dust emission

Dust is a relatively small fraction of the total ISM mass, estimated as being only 1% according to Shetty et al. (2009). Whilst the dust mass accounts for such a small mass component, it still presents an important role in forming stars. The sources of flux embedded deep within the cloud radiate across a wide range of wavelengths, however the dust surrounding the source has the effect of absorbing and then re-radiating photons at a different wavelength. As such, these sources of flux (be they prestellar cores, protostars or pre-main sequence stars) are occluded by the dust enveloping them.

Dust radiates as a modified black body approximated by Equation 1.3 (Shetty et al. 2009).

$$S_\nu = N\Omega\kappa_0 \left(\frac{\nu}{\nu_0}\right)^\beta B_\nu(T) \tag{1.3}$$

In Equation 1.3 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:

$$I_\nu = \frac{N}{R_{dust-gas}} \mu m_p \kappa_0 \left(\frac{\nu}{\nu_0}\right)^\beta B_\nu(T) \tag{1.4}$$

All variables in Equation 1.4 are as defined in Equation 1.3, 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 1.3 and 1.4. These parameters therefore effect the ability to detect radiation from within a molecular cloud.

1.3. Herchel Space Observatory

The Herschel Space Observatory was an ESA mission launched in 2009 with the primary objective of imaging cold regions of the Universe (Garcia-Lario et al. n.d.) and (ESA n.d.). Herschel had a number of objectives. As ESA (n.d.) states, these objectives included "targeted observations of star formation and both young and old stars, to reveal the physical and chemical processes in the early and later phases of a stars life." To achieve this, Herschel was launched with 3 different instruments onboard: Photodetector Array Camera and Spectrometer: PACS (Poglitsch and Altieri 2009), Spectral and Photometric Imaging Receiver: SPIRE (Griffin et al. 2010) and Heterodyne Instrument for the Far Infrared: HiFi (Helmich 2011). One of these instruments, SPIRE, was designed to probe the sites of early star formation deep within molecular clouds. This was achieved by imaging solely in the infra-red and sub-mm regimes. Herschel itself, across all instruments and functional modes, is sensitive to a range of wavelengths quoted by Garcia-Lario et al. (n.d.) as $\sim$ 55 µm to $\sim$ 672 µm. Both PACS and SPIRE were capable of being run in "Parallel mode" - effective simultaneous operation according to Garcia-Lario et al. (n.d.). SPIRE itself had an photometry wavelength range $\sim$ 200 µm to $\sim$ 670 µm whilst PACS was sensitive to a photometry wavelength range $\sim$ 60 µm to $\sim$ 210 µm. This therefore produces effectively continuous coverage of the wavelength range $\sim$ 60 µm to $\sim$ 670 µm.[2]

Tables 1.1 and 1.2 illustrates the passbands for each of the filters within SPIRE and PACS.

Filter	Passband (μm)
PSW	199.4540 – 298.5657
PMW	281.6949 – 424.7548
PLW	391.4346 – 690.8139

Table 1.1: *The passbands (for extended source) for the SPIRE instrument (Service n.d.).*

Filter	Passband (μm)
Blue	55.6706 – 97.7403
Green	79.1026 – 135.1498
Red	117.7762 – 243.6431

Table 1.2: *The passbands for the PACS instrument (Service n.d.).*

In order to achieve the effective wavelength range described in Tables 1.1 and 1.2, both SPIRE and PACS detectors required extensive cooling to a temperature of $0.3K$ via ^{3}He sorption coolers (Garcia-Lario et al. n.d.). Again according to Garcia-Lario et al. (n.d.) the image focal plane required cooling to $1.7K$, again achieved using 2160 litres of Helium cryogen. To minimise any thermal radiation from the observatory itself, the spacecraft was radiatively cooled to a temperature of $85K$ (Garcia-Lario et al. n.d.). These temperatures allowed effective detection (via a bolometric detector in the case of SPIRE and a photometer in the case of PACS) of infra-red and sub-mm photons.

1.4. 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 (Dullemond, Juhasz, Pohl, Sereshti, Shetty, Peters, Commercon, and Flock n.d.).

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` (Juhasz n.d.), is able to write these input files, however for this project a custom script was written such that file creation was independent of `radmc3dPy`.

1.4.1. `radmc3dPy`

Whilst RADMC-3D is written in Fortran 90, a Python wrapper, `radmc3dPy` allows Python to handle RADMC-3D objects.

[2] Further coverage is achieved down to 55 µm using HiFi however this is only achievable for spectroscopic applications.

1.4.2. Input files

As stated in Section 1.4, 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.

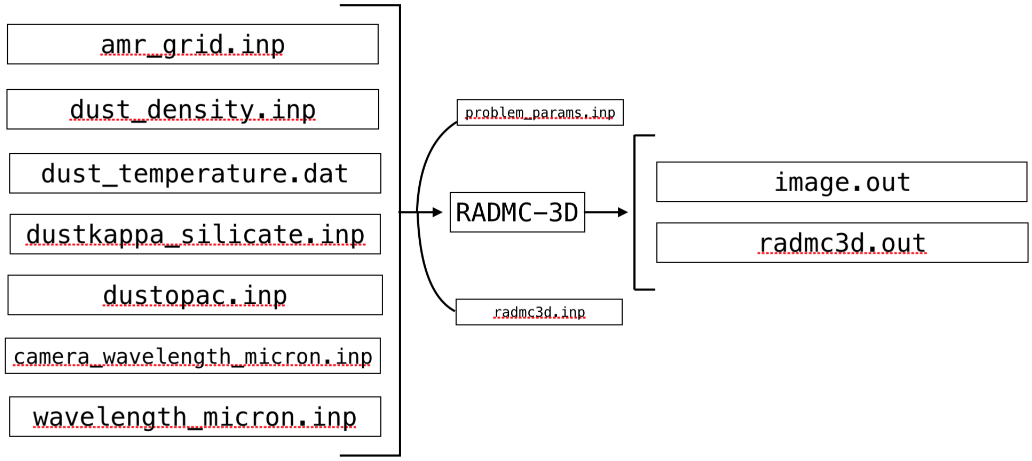


Figure 1.2: *A schematic representation of the RADMC-3D workflow.*

As Figure 1.2 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.

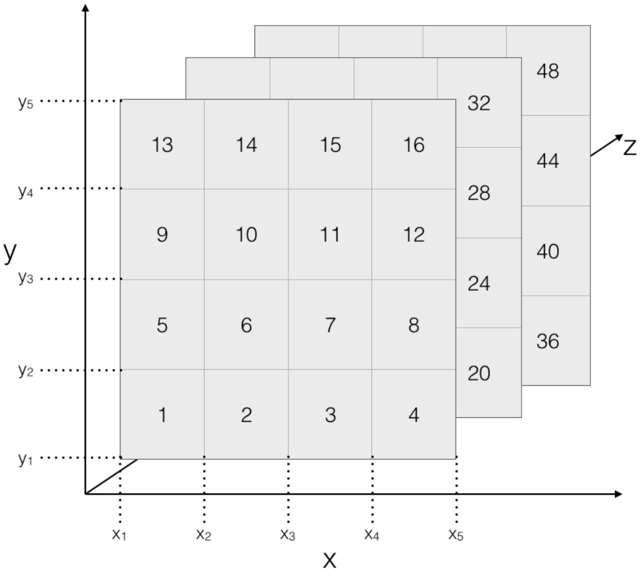


Figure 1.3: *A pictorial representation of the structure of* `amr_grid.inp` *for a 4x4x3 grid. The index of a cell, denoted by the number contained within it, is the position of that cell in* `amr_grid.inp`*.*

`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. This structure is illustrated in Figure 1.4.2.

The index of each cell in Figure 1.4.2 is that cell's position in `amr_grid.inp`. For example,

`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 constrict the range of wavelengths over which RADMC-3D computes, thus emulating the wavelengths available to a given telescope or filter. In this project camera_wavelength_micron.inp was used in order to allow RADMC-3D to compute dust emission intensity in wavelength bands corresponding to those found on both SPIRE (Griffin et al. 2010) and PACS (Poglitsch and Altieri 2009).

1.4.3. Thermal Monte-Carlo dust emission

?? The thermal Monte-Carlo simulation is able to compute dust temperatures using the technique described in Bjorkman and Wood (2001) 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 Dullemond, Juhasz, Pohl, Sereshti, Shetty, Peters, Commercon, Flock, and Min (n.d.), 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 Bjorkman and Wood (2001). Under scattering/re-emission the photon will travel through the image space as defined in amr_grid.inp - see Section 1.4.2 - entering different adjoining pixels as it does so. This increases the temperature of that pixel, again according to Bjorkman and Wood (2001). 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.

1.4.4. 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 Dullemond, Juhasz, Pohl, Sereshti, Shetty, Peters, Commercon, Flock, and Min (n.d.), 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 **??**.

$$I_{result} = I_{start} e^{-\tau} + (1 + e^{-\tau}) S \tag{1.5}$$

??

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 Thermal Monte-Carlo simulation and ray-trace: an `image.out` file that contains the dust emission intensity at each cell in the image plane. RADMc-3D is also capable of accounting for scattering of photons off of dust grains however this project focussed solely upon absorption and re-emission only.

Chapter 2

Methodology

2.1. Model setup

In order to begin developing analysis tools a "test-bed" was required. This test-bed comprised a simple, spherical cloud of uniform density and temperature placed within a medium of greater temperature and lower density. This effectively simulated a cold and dense molecular cloud bathed in a warmer, diffuse ISM. Prior to this however it was necessary to investigate the simulation software's (RADMC-3D) operation. This was achieved using example scripts bundled with the code.

2.1.1. Example scripts

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 2.1 and 2.2.

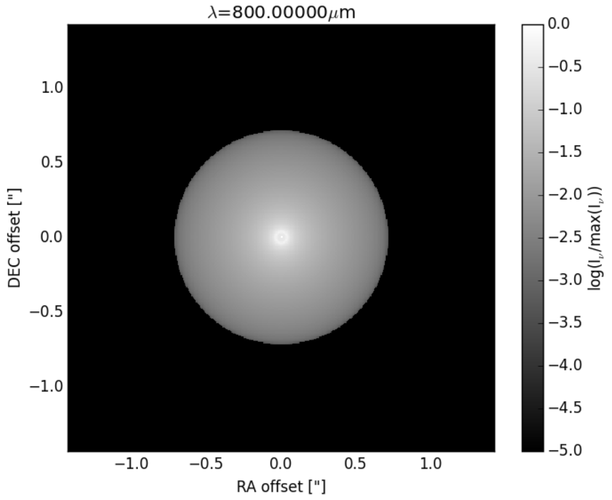


Figure 2.1: *The RADMC-3D output for dust emission intensity computed using thermal Monte-Carlo simulation in 1D.*

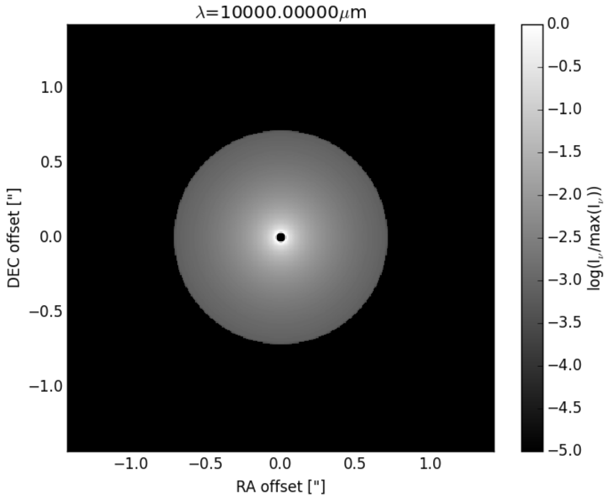


Figure 2.2: *The RADMC-3D output for dust emission intensity computed using thermal Monte-Carlo simulation in 2D.*

In order to begin operation, RADMC-3D's example scripts were analysed and run. The examples were used to understand how RADMC-3D generated the input files required (and as discussed in Section 1.4.2) 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.

These examples ran thermal Monte-Carlo simulations. The first stage in this computation after writing the input files from the user inputs was to build dust temperatures. This was achieved using the process described in Section **??**. Once dust temperatures had been computed, RADMC-3D then performed a ray-trace in order to build the dust emission intensity. This was again achieved using the method described in Section **??**.

2.1.2. Modelling a cloud

To begin building 3D models outside of the example scripts, specific input files needed to be created. As stated, a custom Python script `datafilegen.py` 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.

RADMC-3D then performs its raytrace using the method discussed in Section **??** and outputs the computed intensities to an `image.out` file. This image is a 2D image with image dimensions in the x-y plane corresponding to the initial x and y image dimensions supplied to RADMC-3D. This file contains the dust emission intensity values at each pixel within that plane. This intensity is, however, an idealised intensity that does not account for any ineffeciences. It was assumed that the source is imaged in the absence of any occluding material along the line of sight between the observer and the source, and thus exctinction was neglected.

By supplying `datafilegen.py` with the mass of the cloud, along with the number density of dust both inside and outside of the cloud, it calculates the expected radius of the cloud using a simple mass-density-volume relation. By then supplying the number of image dimensions ($128 \times 128 \times 128$) and the image bounds ($2r$ where r is the radius of the cloud)

It was necessary however in order to simulate data as imaged by Herschel to account for the frequency dependent transmission of Herschel's filters.

2.1.3. Simulating Herschel

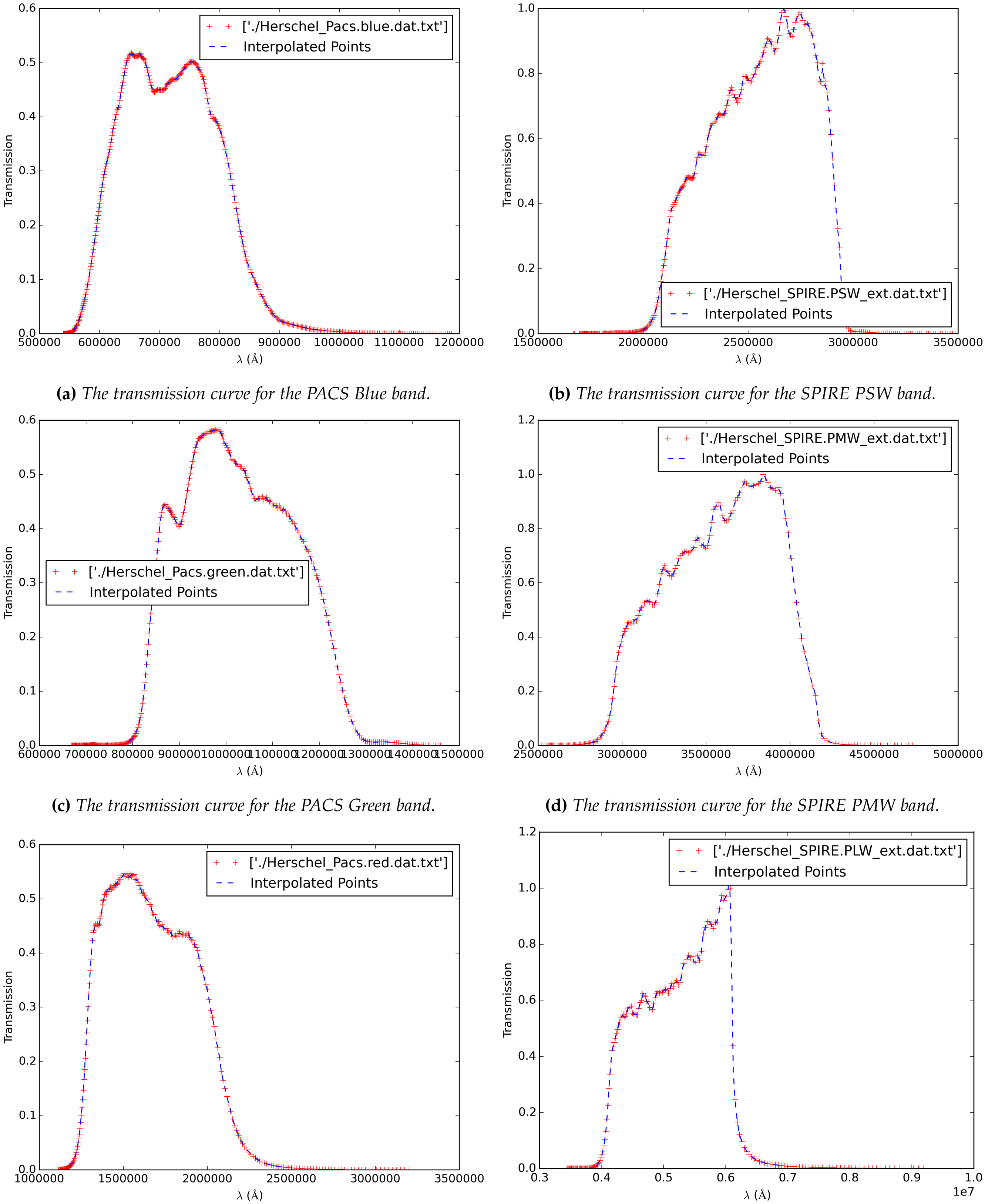
No telescope or observatory can produce perfect images or data; components within the telescope, the observatory or the surroundings all degrade the data quality. In real world applications maximising the SNR is crucial to data quality, but the principles of this will not be touched on here. Furthermore SNR (and therefore integration time) is not accounted for in the simulations or data analysis - images are plotted as if the source is integrated for sufficient time such that the fictional "detector" responds and detects all photons from the source incident upon it as they were emitted from the dust.

Naturally this is an idealised assumption; not all photons that fall on a detector are detected. Detectors have a given responsivity that is a measure of how many of the incident photons do not contribute to the signal. Furthermore, filters are essential to telescope operation as they discriminate wanted and unwanted frequencies to ensure only frequencies within the passband of interest are transmitted to the detector. This introduces a large inefficiency in that not all incident photons are transmitted owing to reflection off of the filter itself, as well as absorption within the filter. This is frequency dependent and thus produces a "transmission curve".

2.1.4. Transmission

This transmission curve varies depending on the filter under consideration and therefore acts as a measure of characterisation. As SPIRE's PSW, PMW and PLW bands along with PACS' Blue, Green and Red filers were considered for this project their transmission data was required. This data was downloaded from Service (n.d.) and subsequently plotted using Python. Figure 2.1.4 shows the transmission curves as a function of frequency for each band.

In order to begin accounting for the transmission effects it was necessary to multiply each datapoint at a given frequency/wavelength by the transmission coefficient at that frequency/wavelength. However the data downloaded from Service



(a) *The transmission curve for the PACS Blue band.*

(b) *The transmission curve for the SPIRE PSW band.*

(c) *The transmission curve for the PACS Green band.*

(d) *The transmission curve for the SPIRE PMW band.*

(e) *The transmission curve for the PACS Red band.*

(f) *The transmission curve for the SPIRE PLW band.*

(n.d.) did not have linearly spaced intervals, resulting in the transmission coefficients and freqencies supplied not matching with those used in the simulation (specifically the data in `camera_wavelength_micron.inp` file). To account for this, Python's `interp1d` function was used. This function would construct linearly spaced intermediate points between the existing datapoints supplied using a 'cubic' routine[1]. The result was a reconstruction of the original data with a user-defined number of points between upper and lower bounds. By choosing these upper and lower bounds to match the upper and lower bounds of `camera_wavelength_micron.inp` (with the same number of wavelength points) the transmission curve could be reconstructed to match the temporal data supplied to the simulation. Figure 2.1.4 illustrates this. The red $+$ datapoints indicate the original data supplied from Service (n.d.), whilst the blue $--$ datapoints indicate the reconstructed, interpolated data. The interpolation is handled by `datafilegen.py` at the input file creation phase.

2.1.4.1 Composite imaging

In reality images are not taken at one discrete wavelength. Instead, images are a composite of wavelengths across a broad range defined by the telescope's filter. To illustrate, consider the SPIRE PSW band. As Table 1.1 illustrates, the PSW band is sensitive to wavelengths in the range $199.4540\,\mu\text{m} - -298.5657$. This means that an image using the PSW band is actually a composite image comprising a weighted average of the wavelengths imaged across.

2.2. Appendix

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.

[1] Other routines to interpolate were available, however the cubic routine provided accurate results so alternatives were not exploreed.

Bibliography

Bjorkman J. E. and Wood K. (2001). "Radiative Equilibrium and Temperature Correction in Monte Carlo Radiation Transfer". 554, pp. 615–623. DOI: 10.1086/321336. eprint: astro-ph/0103249.

Caltech. *IRAS Image of Rho Ophiuchi*. File: rho.jpg. URL: http://coolcosmos.ipac.caltech.edu/image_galleries/IRAS/roph.html.

Dobbs C. L., Burkert A., and Pringle J. E. (2011). "Why are most molecular clouds not gravitationally bound?" 413, pp. 2935–2942. DOI: 10.1111/j.1365-2966.2011.18371.x. arXiv: 1101.3414.

Dullemond C., Juhasz A., Pohl A., Sereshti F., Shetty R., Peters T., Commercon B., and Flock M. *RADMC-3D: A multi-purpose radiative transfer tool*. Version 0.38.

Dullemond C., Juhasz A., Pohl A., Sereshti F., Shetty R., Peters T., Commercon B., Flock M., and Min M. *Manual for RADMC-3D*. English. Version Version 0.39. Universität Heidelberg. 174 pp.

ESA. *Herschel Fact Sheet*. Fact-sheet. URL: http://esamultimedia.esa.int/docs/herschel/Herschel-Factsheet.pdf.

Garcia-Lario P. et al. *Herschel Observer's Manual*. English. Version Version 5.0.3. ESA. 124 pp.

Griffin M. J. et al. (2010). "The Herschel-SPIRE instrument and its in-flight performance". 518, L3, p. L3. DOI: 10.1051/0004-6361/201014519. arXiv: 1005.5123 [astro-ph.IM].

Helmich F. P. (2011). "Herschel HIFI - the Heterodyne Instrument for the Far-Infrared". *EAS Publications Series*. Ed. by M. Röllig et al. Vol. 52. EAS Publications Series, pp. 15–20. DOI: 10.1051/eas/1152003.

Hildebrand R. H. (1983). "The Determination of Cloud Masses and Dust Characteristics from Submillimetre Thermal Emission". 24, p. 267.

Jeans J. H. (1902). "The Stability of a Spherical Nebula". *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 199.312-320, pp. 1–53. ISSN: 0264-3952. DOI: 10.1098/rsta.1902.0012. eprint: http://rsta.royalsocietypublishing.org/content/199/312-320/1.full.pdf. URL: http://rsta.royalsocietypublishing.org/content/199/312-320/1.

Juhasz A. *radmc3dPy*. Version 0.25.

Krumholz M. R. (2011). "Star Formation in Molecular Clouds". *American Institute of Physics Conference Series*. Ed. by E. Telles, R. Dupke, and D. Lazzaro. Vol. 1386. American Institute of Physics Conference Series, pp. 9–57. DOI: 10.1063/1.3636038. arXiv: 1101.5172.

Larson R. B. (1994). "The Structure and Content of Molecular Clouds 25 Years of Molecular Radioastronomy: Proceedings of a Conference Held at Schloss Ringberg, Tegernsee, Germany 14–16 April 1993". Ed. by T. L. Wilson and K. J. Johnston. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. The evolution of molecular clouds, pp. 13–28. ISBN: 9783540490357. DOI: 10.1007/3540586210_2. URL: http://dx.doi.org/10.1007/3540586210_2.

Mamajek E. E. (2008). "On the distance to the Ophiuchus star-forming region". *Astronomische Nachrichten* 329, p. 10. DOI: 10.1002/asna.200710827. arXiv: 0709.0505.

Murphy D. and von Schill L. (2014). "Filamentary Structure in Molecular Clouds". Filaments 2014. URL: https://science.nrao.edu/science/meetings/2014/filamentary-structure.

Pettini M. (2011). "Structure and Evolution of Stars". University Lecture.

Poglitsch A. and Altieri B. (2009). "The PACS Instrument". *EAS Publications Series*. Ed. by L. Pagani and M. Gerin. Vol. 34. EAS Publications Series, pp. 43–62. DOI: 10.1051/eas:0934004.

Service S. F. P. *Filer Profile Service: An experiment about filter standardization in the VO*. Online database comprising values and data for filter profiling. URL: http://svo2.cab.inta-csic.es/svo/theory/fps3/index.php?mode=browse&gname=Herschel&gname2=Pacs.

Shetty R. et al. (2009). "The Effect of Line-of-Sight Temperature Variation and Noise on Dust Continuum Observations". 696, pp. 2234–2251. DOI: 10.1088/0004-637X/696/2/2234. arXiv: 0902.3477 [astro-ph.GA].