

SCHOOL OF PHYSICS AND ASTRONOMY

YEAR 4 PROJECT DISSERTATION

SESSION 2015-2016

Name:	Tomas James
Student Number:	1158976
Degree Programme:	MPhys Physics with Astronomy
Project Title:	How good is dust emission as a tracer of star forming regions in molecular clouds?
Supervisor:	Dr P C Clark
Primary Assessor:	Dr N Peretto
Second Assessor:	Prof B Sathyaprakash

Declaration:

I have read and understand Appendix 2 in the Student Handbook: "Some advice on the avoidance of plagiarism".

I hereby declare that the attached dissertation is exclusively my own work, that no part of the work has previously been submitted for assessment, and that I have not knowingly allowed it to be copied by another person.

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

Word Count (excluding Bibliography and Appendix)¹: 10842

May 8, 2016

¹According to <http://app.uio.no/ifi/texcount/online.php>

Abstract

We present an investigation probing whether dust emission in molecular clouds can be used as a surrogate marker of the star forming regions within these clouds. To simulate a molecular cloud we use synthetic Herschel data from the moving-mesh code Arepo (Springel 2010). Computing dust emission intensity via the ray-tracing code RADMC-3D (Dullemond et al. n.d.[b]) in each of 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 75 μm , 110 μm and 170 μm) produces synthetic, composite observations. We ‘degrade’ the images by accounting for the given filter’s transmission curve, in order to better simulate true Herschel data. These observations are reduced to Spectral Energy Distributions (SEDs) on a pixel-by-pixel basis and a modified blackbody fitted to the points, estimating the best fit N and T at each pixel via a χ^2 minimisation test. These parameters, along with the Arepo input parameters, are used to produce 2-dimensional maps of both N and T. We also analyse these maps using dendograms to probe levels of structure, giving another measure of comparison. We compare both maps with each other (as well as their accompanying dendograms) and determine that whilst the χ^2 minimised maps retain their structure, the observed biases stated in Shetty et al. (2009a) and Shetty et al. (2009b) and Kelly et al. (2012) are evident such that the χ^2 test results in underestimates of N and overestimates of T. We observe the same effects as Shetty et al. (2009a) in that the single fit T used here is an inaccurate assumption, given the line of sight temperature variations. We also observe that 3 gravitationally bound cores are recovered from the χ^2 N data, whilst 25 gravitationally bound cores are recovered from the data derived N. The 3 bound cores from the χ^2 routine are contained within the data derived N cores. We observe that the core masses from the χ^2 map are significantly lower than the core masses from the data derived N, as is expected given the underestimate of N in the χ^2 minimisation.

Contents

1	Introduction	5
1.1	Molecular clouds	5
1.1.1	Molecular cloud composition	6
1.1.2	Star forming conditions	6
1.2	Dust emission	8
1.2.1	The β - T degeneracy	9
1.3	Herchel Space Observatory	9
1.4	RADMC-3D	10
1.4.1	Input files	10
1.4.2	Ray tracing dust emission	12
2	Methodology	13
2.1	Modelling routine	13
2.1.1	Modelling a cloud	13
2.1.2	Building dust intensities	13
2.1.3	Simulating Herschel	13
2.1.4	Accounting for filter transmission	14
2.1.5	Composite imaging	14
2.2	Analysis routine	16
2.2.1	Spectral energy distributions (SEDs)	16
2.2.2	Parameter estimation	17
2.3	Image construction	18
2.3.1	2D Maps of N and T	18
2.3.2	Dendrogram analysis	19
3	Results	21
3.1	Isothermal spherical cloud	21
3.2	Arepo data	25
3.3	Line of sight temperature variations	29
3.4	Core extraction	30
4	Discussion	32
5	Conclusion	33
6	Future work	34
7	Acknowledgements	35
8	Appendix	36

List of Figures

1.1	An IRAS image showing the Rho Ophiuchi cloud complex, the closest cloud complex to the Solar System (Caltech n.d.). Rho Ophiuchi is an example of a molecular cloud that is actively forming stars.	5
1.2	A plot illustrating how the dust spectral index β varies the emergent dust SED whilst both N and T remain fixed. This is based off of a plot found in Shetty et al. (2009a).	8
1.3	A figure illustrating a basic molecular cloud, with regions of higher density denoted by darker regions. Solid black lines indicate an incoming photon, whilst a dashed line indicates the line of sight.	9
1.4	A schematic representation of the RADMC-3D workflow.	11
1.5	A pictorial representation of the structure of <code>amr_grid.inp</code> for a 4x4x3 grid. The index of a cell, denoted by the number contained within it, is the position of that cell in <code>amr_grid.inp</code> . This image is based off of an image in Dullemond et al. (n.d.[a]).	11
2.1	The transmission curves for each of the SPIRE PSW, PMW and PLW bands along with the PACS Blue, Red and Green bands.	15
2.2	The emergent image (before transmission effects) of a raytrace on a $1M_{\text{Sun}}$ spherical, 3D, isothermal ($T = 10K$) cloud.	16
2.3	The emergent image (after transmission effects) of a raytrace on a $1M_{\text{Sun}}$ spherical, 3D, isothermal ($T = 10K$) cloud.	16
2.4	A figure showing the continuous SED sampled from <code>image_trans.out</code> , along with the SED after convolving with the transmission curve (right hand panel). Also illustrated is the discrete point that Herschel would see were it to observe this source in its PSW band.	17
2.5	An idealised molecular cloud containing some substructure defined by sequentially darker regions. Below, the dendrogramic representation of the cloud (astrodendro 2016).	19
3.1	The RADMC-3D intensity outputs for the isothermal source at $T = 10K$ imaged in all 6 passbands considered.	21
3.2	The recovered values of N from the χ^2 minimisation routine.	22
3.3	The reconstructed values of N from the initial input data.	22
3.4	The recovered values of T from the χ^2 minimisation routine.	22
3.5	The reconstructed values of T from the initial input data.	22
3.6	A representation of how the χ^2 derived N values compare to the initial data input N	23
3.7	A representation of how the χ^2 derived T values compare to the initial data input T	23
3.8	A plot showing how the ratio of $T_{\text{input}}/T_{\chi^2}$ varies with the input T	23
3.9	The RADMC-3D intensity outputs for the Arepo data imaged in all 6 passbands considered.	25
3.10	The recovered values of N from the χ^2 minimisation routine.	26
3.11	The reconstructed values of N from the initial input data.	26
3.12	The recovered values of T from the χ^2 minimisation routine.	26
3.13	The reconstructed values of T from the initial input data.	26
3.14	A representation of how the χ^2 derived N values compare to the initial data input N	27
3.15	A representation of how the χ^2 derived T values compare to the initial data input T	27
3.16	A plot showing how the ratio of $T_{\text{input}}/T_{\chi^2}$ varies with the data derived N	28
3.17	A plot showing the relationship between the χ^2 recovered values of N and T	29
3.18	The line of sight temperature variations in the isothermal sphere case as a function of the data derived N and T	29

3.19	The line of sight temperature variations in the Arepo simulation case as a function of the data derived N and T	29
3.20	The χ^2 recovered N . The contours represent leaves of the dendrogram: a red contour represents an unbound core, whilst a green contour represents a bound core. The lower graph shows the dendrogram. In this case, it displays the structure of the leaves (cores).	31
3.21	The data derived N . The contours represent leaves of the dendrogram: a red contour represents an unbound core, whilst a green contour represents a bound core. The lower graph shows the dendrogram. In this case, it displays the structure of the leaves (cores).	31

List of Tables

1.1	A table showing the different phases of the ISM, along with estimates of that phase's temperature, T and number density, n (Ferrière 2001).	6
1.2	The passbands (for extended source) for the SPIRE instrument (SVO Filter Profile Service n.d.).	10
1.3	The passbands for the PACS instrument (SVO Filter Profile Service n.d.).	10
2.1	A table showing the values of $\Delta\chi^2$ for fit parameters in varying M dimensions. Bold parameters are those used in this study (Press, Flannery, and Teukolsky 1986)	18
3.1	A table showing the masses of cores extracting from the χ^2 N map and the data derived N map.	31

Chapter 1

Introduction

1.1. MOLECULAR CLOUDS

A molecular cloud is a cold ($< 20K$) and dense region of the Interstellar Medium (ISM), composed primarily of molecular Hydrogen (H_2) (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. (2009a). Whilst the larger molecular gas component within the cloud constitutes the material responsible for star formation, the small dust mass component remains a crucial element in the evolution of the cloud and the star forming process.

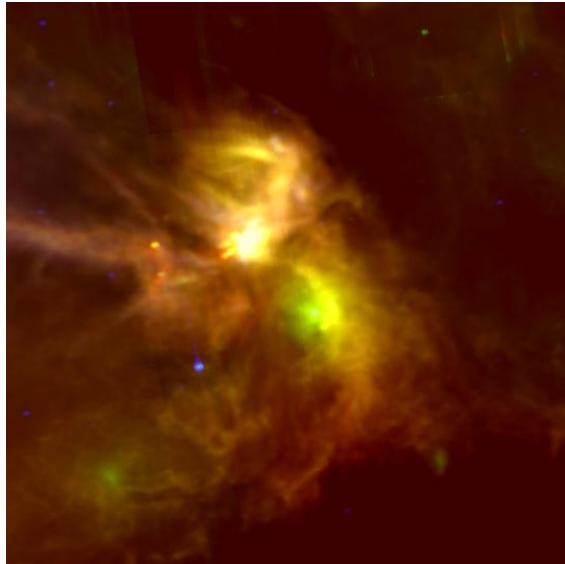


Figure 1.1: An IRAS image showing the Rho Ophiuchi cloud complex, the closest cloud complex to the Solar System (Caltech n.d.). Rho Ophiuchi is an example of a molecular cloud that is actively forming stars.

constrained in its relation to the IMF. Therefore, an alternative method of observing pre-Main Sequence objects, along with investigating the conditions within a molecular cloud at these points, is required to begin constraining both the initial conditions and the CMF. Given that dust occludes these sites, could the dust itself prove a useful tracer of the activity that it enshroudes?

An example of a molecular cloud is shown in Figure 1.1. The subject of Figure 1.1 is the Rho Ophiuchi cloud complex as imaged by the Infra-red Astronomical Satellite (IRAS). Rho Ophiuchi is the nearest active star forming molecular cloud complex to the Solar System¹. Crucially, Rho Ophiuchi is an example of a ‘stellar nursery’: a molecular cloud that is actively forming stars. As a result of this, Rho Ophiuchi provides an opportunity to study these 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). Figure 1.1 does illustrate, however, that the early stages of star formation aren’t directly observable owing to their occlusion by dust in the molecular cloud. This means that the star-forming process prior to the pre-Main Sequence phase - where the star blows away its surroundings material - has yet to be directly observed, merely theorised (Hayashi 1966). Consequently, the initial conditions for star formation to occur have yet to be constrained. Furthermore the only measure of mass distribution in the stellar environment is the Initial Mass Function (Salpeter 1955), which describes the mass at which a star enters the Main-Sequence - a measure of the mass distribution prior to the Main-Sequence (the Core Mass Function) when the stellar object remains unobserved is poorly

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

1.1.1. Molecular cloud composition

Much like the ISM, molecular clouds are composed of gas and cosmic dust. Gas in the ISM is thought to be in a number of ‘phases’, with each phase being a different chemical composition as well as a different density and temperature. The different gaseous phases of the ISM, along with estimates of their temperature T and number density n are shown in Table 1.1.

Component	$T(K)$	$n(cm^{-3})$
Molecular	10 – 20	$10^2 – 10^6$
Cold atomic	50 – 100	20 – 50
Warm atomic	6000 – 10000	0.2 – 0.5
Warm ionized	~ 8000	0.2 – 0.5
Hot ionised	$\sim 10^6$	~ 0.0065

Table 1.1: A table showing the different phases of the ISM, along with estimates of that phase’s temperature, T and number density, n (Ferrière 2001).

Much like gas in the interstellar medium, interstellar dust’s chemical composition varies, with theories predicting both silicate and carbonaceous species co-existing; this project assumed all dust to be of one species, and to be silicate based. As a molecular cloud is a condensation of the ISM, the composition of both remain similar, however a molecular cloud differs from the ISM in a number of key ways. Greater densities are found within a molecular cloud than that of the ISM. The gas density in the core of a molecular cloud is thought to be around $1 \times 10^4 cm^{-3}$ according to Pineda et al. (2010), whilst the gas density in the surrounding ISM is thought to be around $10^2 cm^{-3}$ as Table 1.1 shows. Assuming the dust mass to be 1% that of the gas mass then this produces dust densities of $10^2 cm^{-3}$ in dense regions of the cloud and $1 cm^{-3}$ in the surrounding ISM.

The dust densities may be greater in a molecular cloud than the ISM, but the temperature is observed to obey a different paradigm. Hildebrand (1983) estimates that a molecular cloud has a temperature $< 20K$, significantly cooler than the surrounding ISM as shown by Table 1.1. The warmer ISM results 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 throughout the cloud itself. These filaments are regions of higher density, but as Larson (1994) highlights, the ‘edges’ and transitions in molecular clouds are not strict boundaries as they may appear, but are rather molecular gas transitioning into the surrounding atomic gas. This poses an interesting question: if filaments are dense condensations of gas and dust, then do they act as preferential star formation sites? To answer this question, it is useful to review the conditions within the cloud itself that could instigate the star forming process.

1.1.2. Star forming conditions

Jeans (1902) first demonstrated that a gas of given density and mass will begin to collapse if the gas pressure is overwhelmed by the gravitational force of the gas. This is applicable to star formation, given that 99% of a molecular cloud, by mass, is gas. As a result, star forming regions within molecular clouds originate from locations of gravitational instability that lead to subsequent collapse. 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 $2T + \Omega = 0$, where T is the total thermal energy of the system and Ω is the total gravitational 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 unimpeded 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 exceeds the Jeans' mass given by Equation 1.1. This is known as the Jeans Criterion.

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

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 gas (in this case H_2) in the region and ρ is the density of the region prior to collapse. The radius of the region required to collapse is given in Equation 1.2.

$$R_J = \left(\frac{15kT}{4\pi G \mu m_p \rho} \right)^{1/2} \quad (1.2)$$

k is the Boltzmann constant, whilst G is the Universal Gravitational Constant. μ is the mean molecular weight and m_p is the mass of the proton. The collapse of an M_J mass region occurs according to the free-fall time, t_{ff} described in Equation 1.3 (Jeans 1902).

$$t_{ff} = \sqrt{\frac{3\pi}{32G\rho}} \quad (1.3)$$

Variables within Equation 1.3 are as they were defined in Equation 1.1. As Equation 1.1 illustrates, the larger the temperature T and the lower the gas density ρ , then the greater the Jeans' mass becomes, making it harder for the region to begin collapse. The temperature within the cloud itself exists as both a gas temperature and a dust temperature. Glover and Clark (2012) estimates that the gas within the cloud does not heat the dust within the cloud until the number density n exceeds $2 \times 10^4 \text{ cm}^{-3}$. This is defined as 'thermal coupling'. The dust itself reaches thermal equilibrium when the dust heating rate per unit volume Γ_{ext} equals the radiative cooling rate Λ_{dust} (Clark, Glover, and Klessen 2012):

$$\Gamma_{ext} - \Lambda_{dust} = 0 \quad (1.4)$$

The heating mechanism of the dust is primarily through absorption of radiation, either from the Interstellar Radiation Field (a field of electromagnetic radiation in the interstellar medium) or from sources of radiation within the cloud (a protostar, for example). A mathematical treatment as defined in Clark, Glover, and Klessen (2012) can be found in the Appendix. This study used an ISRF in the Arepo data but not in the isothermal sphere case discussed later (it was assumed that the dust grains in both the cloud and ISM were in thermal equilibrium such that Equation 1.4 was satisfied).

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, transient structures. The onset of gravitational binding is also not a singular process. A region of a molecular cloud may begin to collapse under the Jeans criterion in Equation 1.1, however this clump may then sub-fragment into smaller clumps that also begin collapsing in a similar manner. Whether a given clump is gravitationally bound or not is a measure of whether that clump is a prestellar core or not, according to André, Basu, and Inutsuka (2009). André, Basu, and Inutsuka (2009) also states that the majority of prestellar cores remain starless - even so, even fewer cores will evolve to become protostellar cores and begin the route to the Main-Sequence. Those prestellar cores that do not evolve to form protostellar cores are thought to disperse before they're ever able to form stars.

The conditions that harbour these star forming regions are difficult to observe directly because of their obscuration by the enveloping dust. Gas is responsible for the collapse of the medium (via the Jeans Criterion in Equation 1.1) and formation of the star itself, however dust absorbs the emission from these collapsing regions, essentially masking them. The dust itself will, however, re-radiate the absorbed radiation incident upon it at a sub-mm/IR wavelength.

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. (2009a). 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 (longer) wavelength. As such, these sources of flux (be they prestellar cores or protostars) are occluded by the dust enveloping them.

According to Hildebrand (1983), dust radiates as a modified black body in the far infra-red regime (i.e. from $\lambda = 60 \mu\text{m}$) that can approximated using Equation 1.5.

$$S_\nu = N_{H_2} \Omega \kappa_0 \left(\frac{\nu}{\nu_0} \right)^\beta B_\nu(T) \quad (1.5)$$

In Equation 1.5 S_ν is the flux density, N_{H_2} is the column density of molecular Hydrogen, 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 (for silicate and carbonaceous grains, Draine and Lee (1984) approximates $\beta \sim 2$) and $B_\nu(T)$ is the frequency dependent Planck function. An assumption made here, as stated by Kelly et al. (2012), is that the dust is optically thin such that $\tau_\nu = N\kappa_\nu \ll 1$. The term $\kappa_0(\frac{\nu}{\nu_0})^\beta$ is the Hildebrand dust opacity law κ_ν given in Hildebrand (1983). In terms of intensity this equation becomes:

$$I_\nu = \frac{N_{H_2}}{R_{dust-gas}} \mu m_p \kappa_0 \left(\frac{\nu}{\nu_0} \right)^\beta B_\nu(T) \quad (1.6)$$

All variables in Equation 1.6 are as defined in Equation 1.5, with the addition of $R_{dust-gas}$. $R_{dust-gas}$ is the dust-to-gas ratio, here approximated as being 100. We observe that $\frac{N_{H_2}}{R_{dust-gas}}$ is the column density of dust along the line of sight, N . It is this parameter (along with T along the line of sight) that are the focii of this project. Furthermore 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.

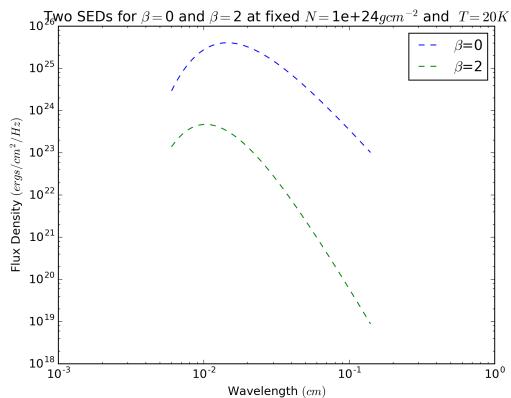


Figure 1.2: A plot illustrating how the dust spectral index β varies the emergent dust SED whilst both N and T remain fixed. This is based off of a plot found in Shetty et al. (2009a).

the right. Furthermore, the greater number of boundaries present greater opportunity for the photon to reflect/scatter. This means that the photon is not absorbed by the dust grain and therefore does not heat it. The larger number of boundaries through the central line of sight means that the material in regions of higher density are not heated as much

Dust properties such as the spectral index β , temperature T and column density N clearly have an effect on dust emission according to Equations 1.5 and 1.6. Importantly, both Equations 1.5 and 1.6 describe dust emission at a single temperature T . Two emergent SEDs described by Equation 1.5 at fixed single N and T are shown in Figure 1.2. These properties, however, are experimentally difficult to measure and Kelly et al. (2012) states that most observational attempts to determine T and β rely on a χ^2 SED fitting routine. This routine does, however, suffer from drawbacks when fitting for a single temperature T . To illustrate, consider Figure 1.3.

Figure 1.3 shows a basic molecular cloud with sub-structure (and therefore increasing density) illustrated by the consecutively darker regions. A circle indicates the point at which a photon, or indeed the line of sight, passes from a region of one density to a region of different density. By using the line of sight through the centre of the cloud, it is observed that this line of sight intersects a larger number of boundaries than the line of sight on the right that only passes through one region. This means that a larger N is observed through the line of sight in the middle of the cloud than the line of sight on

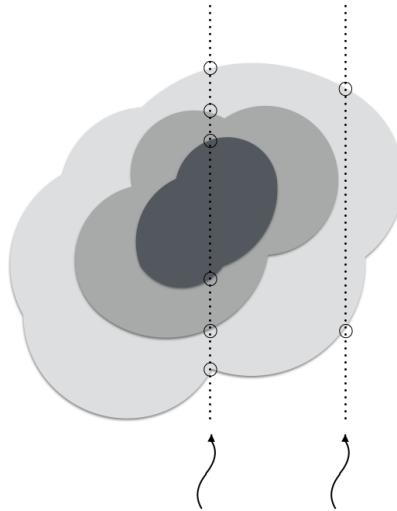


Figure 1.3: A figure illustrating a basic molecular cloud, with regions of higher density denoted by darker red regions. Solid black lines indicate an incoming photon, whilst a dashed line indicates the line of sight.

as the material at lower density, for example the region intersected by the right hand line of sight. This has the effect of introducing more temperature variations along the line of sight when N is large in comparison to low N . Therefore, single T fits (in the case of a χ^2 fit) in regions of low N recover accurate T , whereas single T fits in regions of high N are overestimated owing to the line of sight temperature variations. This effect is observed in studies by Shetty et al. (2009a) and Shetty et al. (2009b) and Kelly et al. (2012) by fitting for β - T .

1.2.1. The β - T degeneracy

As Equation 1.6 shows, dust does not radiate as a perfect blackbody. This is accounted for by multiplying the Planck function (i.e. the function that describes a body that radiates as a perfect blackbody) by the dust opacity term $\kappa = \kappa_0(\frac{v}{v_0})^\beta$. Two key variables here are the dust spectral index β and the dust temperature T . These two properties suffer from a degeneracy (Keene et al. 1980) such that any χ^2 test applied to determining the values of β and T may underestimate one parameter whilst overestimating another.

As Shetty et al. (2009a) states, a fitting routine for β and T is prone to overestimates owing to the assumption that temperature is constant along the line of sight. For an isothermal source this assumption may be well placed however in the case of a complex structure such as a molecular cloud shown in Figure 1.1 this assumption is inaccurate. Equations 1.5 and 1.6 describe dust emission from an isothermal source, i.e. one in which the dust is at one discrete temperature. Molecular clouds are not isothermal and have a range of temperatures along the line of sight, and as Shetty et al. (2009a) highlights, the SEDs describing the dust emission in these instances are actually combinations of the individual SEDs from each individual temperature source. Observing these sources directly accounts for this, and one such way to observe a molecular cloud in reality is via the Herschel Space Observatory.

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.; ESA n.d.). Herschel had a number of objectives, including observing sites of early star formation to understand the initial conditions that establishes star formation itself (ESA n.d.). 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 $\sim 55\text{ }\mu\text{m}$ to $\sim 672\text{ }\mu\text{m}$. 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 $\sim 200\text{ }\mu\text{m}$ to $\sim 670\text{ }\mu\text{m}$ whilst PACS was sensitive to a photometry wavelength range $\sim 60\text{ }\mu\text{m}$ to $\sim 210\text{ }\mu\text{m}$. This therefore produces effectively continuous coverage of the wavelength range $\sim 60\text{ }\mu\text{m}$ to $\sim 670\text{ }\mu\text{m}$.²

Tables 1.2 and 1.3 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 1.2: The passbands (for extended source) for the SPIRE instrument (SVO Filter Profile Service n.d.).

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

Table 1.3: The passbands for the PACS instrument (SVO Filter Profile Service n.d.).

In order to achieve the effective wavelength range described in Tables 1.2 and 1.3, both SPIRE and PACS detectors required extensive cooling to a temperature of 0.3K via ^3He 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. As this project did not use real Herschel data, a method of not only simulating dust emission intensity but also generating realistic images was required. This was achieved by using RADMC-3D.

1.4. RADMC-3D

RADMC-3D is a 3-dimensional Monte-Carlo radiative transfer and ray tracing code developed by Cornelis Dullemond for radiative transfer and emission astrophysics in dusty environments (Dullemond et al. n.d.[b]).

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. Gas emits radiation in lines rather than continua, and therefore an SED cannot be used to recover physical gas properties directly. Furthermore, the dust and gas temperatures remain uncoupled at the densities considered in this project meaning that their temperatures differ, resulting in their emission being in different wavelength ranges. The dust emission intensity can be computed using 2 different methods: the thermal Monte-Carlo simulation, or an image ray trace. All computations, regardless of whether a Monte Carlo simulation or raytrace, require input files for RADMC-3D to read and use in its simulations.

1.4.1. 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 file. Figure 1.4 shows

²Further coverage is achieved down to $55\text{ }\mu\text{m}$ using HiFi however this is only achievable for spectroscopic applications.

the workflow used in this project, including all files that are required in order to operate RADMC-3D.

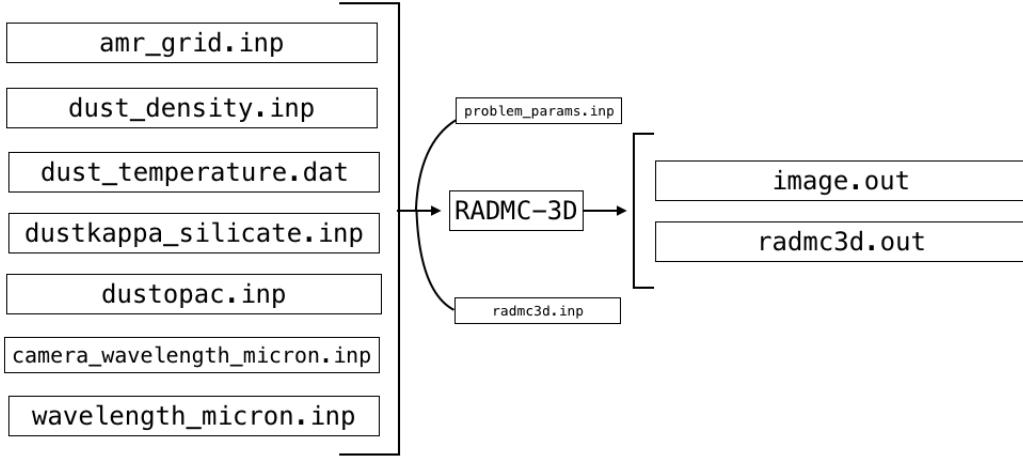


Figure 1.4: A schematic representation of the RADMC-3D workflow.

As Figure 1.4 illustrates, all input files to RADMC-3D have extension `.inp` whilst all output files have extension `.out`. Any file that RADMC-3D is able to modify, 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 as well as the number of datapoints in the file.

`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 1.5.

The index of each cell in Figure 1.5 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.

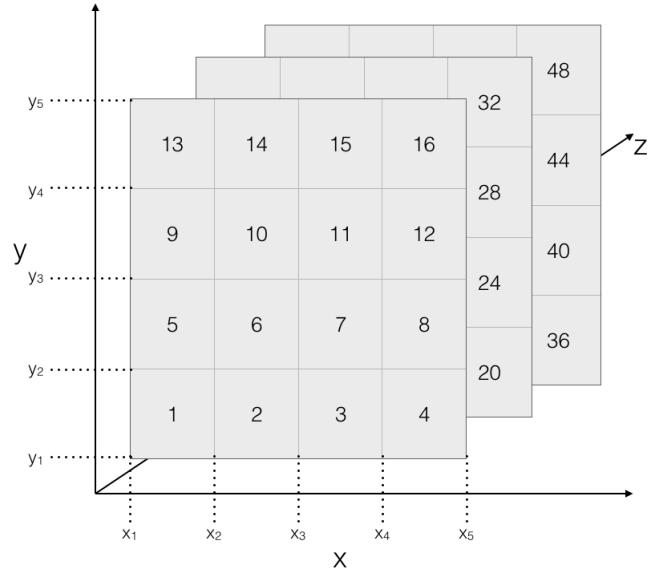


Figure 1.5: A pictorial representation of the structure of `amr_grid.inp` for a $4 \times 4 \times 3$ grid. The index of a cell, denoted by the number contained within it, is the position of that cell in `amr_grid.inp`. This image is based off of an image in Dullemond et al. (n.d.[a]).

Another crucial file is the `wavelength_micron.inp` file. This input file defines the wavelengths over which RADMC-3D should determine dust emission intensity. By default, `wavelength_micron.inp` runs from $0.1\text{ }\mu\text{m}$ to $1000\text{ }\mu\text{m}$ 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).

1.4.2. Ray tracing dust emission

Whilst RADMC-3D is capable of performing a full thermal Monte-Carlo simulation to determine dust temperature, it is also capable of an image ray trace that reads in both user-defined dust temperatures and dust densities, thus bypassing the computation of the dust temperatures. This project made use of the image raytrace rather than the Monte-Carlo simulation. In the case of the isothermal sphere, temperatures are user generated from idealised estimates of temperature, whilst temperatures are generated from a thermal Monte Carlo simulation in the case of the Arepo data. According to Dullemond et al. (n.d.[a]), 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 1.7.

$$I_{\text{result}} = I_{\text{start}}e^{-\tau} + (1 + e^{-\tau})S \quad (1.7)$$

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 and 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. MODELLING ROUTINE

This project uses a χ^2 routine to determine N and T from synthetic observations. This process is the same as that used to derive N , T or even β in ‘real’ observations (Shetty et al. 2009a; Shetty et al. 2009b). As mentioned in Section 1.2.1, a single T χ^2 routine overestimates N and T owing to the line of sight temperature variations. As a result, an assessment of how well the χ^2 results compare to the input quantities in the absence of line of sight temperature variations can be achieved with a simple, isothermal cloud of uniform density and temperature placed within a medium of greater temperature and lower density. This effectively simulates a cold and dense molecular cloud surrounded by a warmer, diffuse ISM.

2.1.1. Modelling a cloud

Modelling an isothermal cloud can be split into 3 distinct stages:

1. Generate input files to create parameter spaces for RADMC-3D
2. Perform ray-trace on these files, at each of the SPIRE and PACS bands, to build dust emission intensities
3. Account for the filter transmission to ‘degrade’ the intensities, and reduce these datapoints to one discrete, effective wavelength as SPIRE/PACS would observe

2.1.2. Building dust intensities

By supplying RADMC-3D with initial input files to build parameter spaces, the code can then perform its raytrace using the method discussed in Section 1.4.2. The output, `image.out`, 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 inefficiencies. It was necessary therefore 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. It was also 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 extinction was neglected.

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 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. Accounting for filter transmission

Whilst the detector’s filter permits only photons of the required frequencies to the detector by reflecting the unwanted photons, it also has the effect of reducing the intensity of the photons that are permitted through it. This can be described via a transmission curve, which varies depending on the filter under consideration. The transmission curve itself describes the transmission coefficient at each wavelength/frequency in the band, which is simply $\frac{I_f}{I_i}$ where I_f is the intensity of the incoming wave and I_i is the intensity of the wave prior to entering the filter. Figure 2.1 shows the transmission curves as a function of frequency for each of the 6 bands considered in this project.

Figure 2.1 illustrates this. The red + datapoints indicate the original data supplied from SVO Filter Profile Service (n.d.), whilst the blue — datapoints indicate the reconstructed, interpolated data. Figure 2.1 also illustrates that transmission is not constant across each filter. The interaction of the incoming waves with the filter material itself is frequency dependent, thus producing the curve.

To account for the transmission coefficients across the filter, multiplying each datapoint at a given frequency/wavelength by the transmission coefficient at that frequency/wavelength obtains the intensity of the photon after transmission. However the transmission data downloaded from SVO Filter Profile Service (n.d.) is not linearly spaced, resulting in the transmission coefficients and frequencies not matching with those used in the simulation (specifically the data in `camera_wavelength_micron.inp` file). To account for this, cubic interpolation was used¹ to produce 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 can be reconstructed to match the temporal data supplied to the simulation.

2.1.5. 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 1.2 illustrates, the PSW band is sensitive to wavelengths in the range $199.4540\text{ }\mu\text{m} - 298.5657\text{ }\mu\text{m}$. This means that an image using the PSW band is actually a composite image comprising a weighted average of the wavelengths imaged across such that the emergent image is effectively an image taken at the effective band wavelength. As a result of this, any transmission effects had to be accounted for in the image composition, such that Equation 2.1 was used to construct the image intensities at each pixel.

$$I_{xpix,ypix} = \frac{\sum_{i=0}^{npix} I_{xpix,ypix,\nu_i} T_{\nu_i}}{\sum_{i=0}^{npix} T_{\nu_i}} \quad (2.1)$$

$I_{xpix,ypix}$ is the pixel intensity whilst $I_{xpix,ypix,\nu_i}$ and T_{ν_i} are the intensity at $(xpix, ypix)$ and transmission coefficient at frequency ν_i respectively. By performing the product of the intensity across all considered frequencies (i.e. ν_i) at pixel $(xpix, ypix)$ with the transmission coefficient across all considered frequencies and normalising by dividing by the sum of all of the transmission coefficients in the dataset, the transmission weighted intensity can be computed at each pixel. Figure 2.2 illustrates the output for a 3-dimensional isothermal cloud in RADMC-3D with perfect transmission (i.e. no transmission convolution). Figure 2.3 illustrates the same image but with transmission being accounted for according to Equation 2.1. These images are computed in the Herschel PSW band.

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

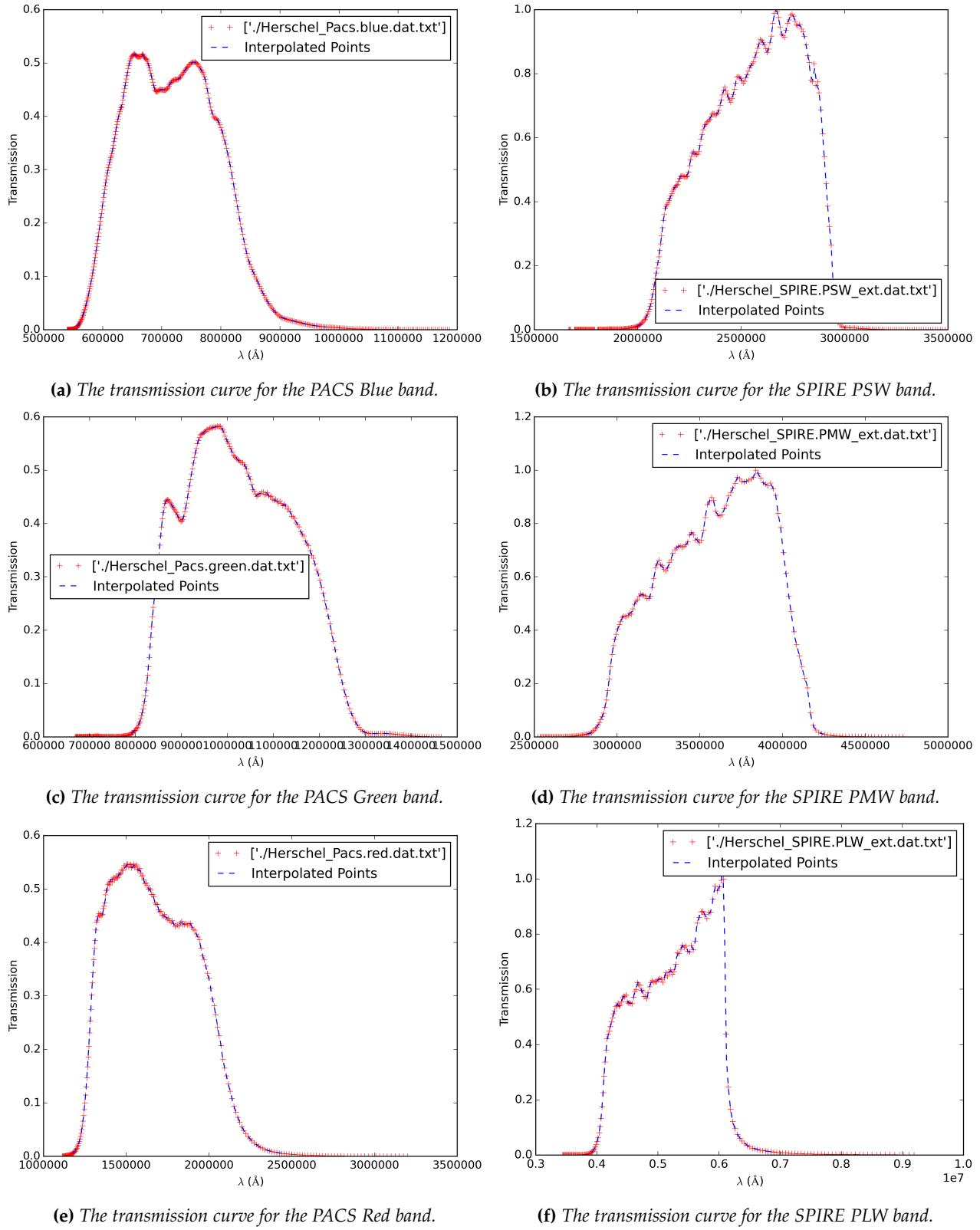


Figure 2.1: The transmission curves for each of the SPIRE PSW, PMW and PLW bands along with the PACS Blue, Red and Green bands.

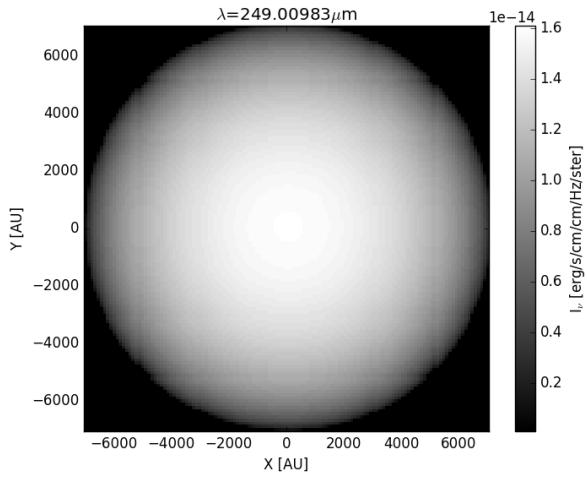


Figure 2.2: The emergent image (before transmission effects) of a ray-trace on a $1M_{\odot}$ spherical, 3D, isothermal ($T = 10K$) cloud.

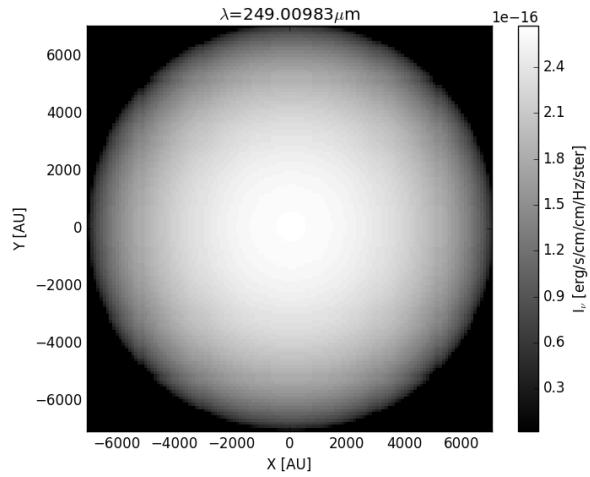


Figure 2.3: The emergent image (after transmission effects) of a ray-trace on a $1M_{\odot}$ spherical, 3D, isothermal ($T = 10K$) cloud.

2.2. ANALYSIS ROUTINE

The analysis routine used in the project can be split into 4 distinct stages:

1. Use the discrete data from RADMC-3D images to produce SEDs
2. Fit a modified blackbody to the SED on a pixel by pixel basis and utilise χ^2 minimisation to derive the best fit N and T
3. Construct maps of N and T from both the χ^2 derived values as well as the initial input values
4. Extract prestellar core locations and masses from these maps using dendrogrammatic analysis

2.2.1. Spectral energy distributions (SEDs)

Each individual source of flux has an energy distribution; that is, a function of how that point radiates as a function of frequency/wavelength. This function is continuous, and in the case of a blackbody source the SED can be described by the Planck function. Within the synthetic observations used in this project, each pixel within the computed image (rather than a discrete point) has an energy distribution. The SEDs in these cases are approximated using Equation 1.6 as opposed to the unaltered Planck function². When imaging an object however this continuous SED is not observed. Instead, the telescope ‘sees’ one, discrete intensity for each resolved point owing to the telescope’s averaging over all observed frequencies as discussed in Section 2.1.5. Figure 2.4 illustrates this.

Figure 2.4 shows the SED for the central pixel (0,0) of Figure 2.3 imaged in the PSW band. This SED is sampled directly from `image_trans.out` by taking the intensity of the pixel (in this case the (0,0) pixel) at each discrete frequency/wavelength considered. Plotting the intensity as a function of frequency produces the red continua in the left hand panel. Convolving the theoretical SED with the transmission curve for the band imaged in produces the SED

²Whilst the Planck function does not describe dust emission in this instance, a modified version of the Planck function does

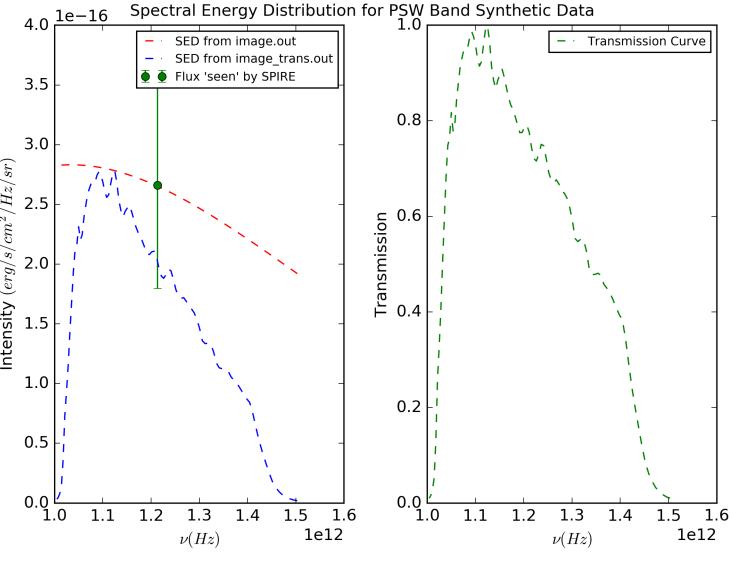


Figure 2.4: A figure showing the continuous SED sampled from `image_trans.out`, along with the SED after convolving with the transmission curve (right hand panel). Also illustrated is the discrete point that Herschel would see were it to observe this source in its PSW band.

that falls on to Herschel’s filter. This is displayed as the blue curve in the left hand panel. The green point represents the intensity incident upon the detector, as determined by Equation 2.2.

$$I_{pix} = \frac{\sum_v I_{pix,v}}{\sum_v T_v} \quad (2.2)$$

I_s is the averaged intensity as measured by Herschel, whilst $I_{pix,v}$ is the image intensity at frequency v and T_v is the transmission coefficient at frequency v .

Repeating this procedure for each band on board SPIRE and PACS allows a dataset of points to be built up as a function of frequency. This dataset then acts as an ‘expected’ dataset that is described by Equation 1.6. As a direct result of this, fitting Equation 1.6 to the ‘expected’ dataset using a χ^2 minimisation routine allowed the 2 fit parameters, N and T to be determined.

2.2.2. Parameter estimation

A χ^2 minimisation test is a statistical test to assess the ‘goodness-of-fit’ between observed data and a theoretical model. The value of χ^2 is determined via Equation 2.3 (Taylor 1997).

$$\chi^2 = \sum_i \left(\frac{O_i - E_i}{\sigma_i} \right)^2 \quad (2.3)$$

O_i is an observed point, and E_i is the expected point. σ_i is the standard deviation of O_i . If O_i is a given dataset and E_i is the model that is expected to fit that dataset, then χ^2 is therefore a measure of how close that model fits the data within one standard deviation.

By applying this χ^2 test to assess how well the modified blackbody given in Equation 1.6 fits the observed data, recovery of both N and T is possible. We perform this on a pixel by pixel basis such that N and T can be determined for each pixel in the data. The observed points used here, O_i , are 6 discrete points, however the function being fitted

$k - \sigma$	P	$M = 1$	$M = 2$	$M = 3$	$M = 4$
$1 - \sigma$	68%	1	2.30	3.53	4.72
$2 - \sigma$	95.4%	4	6.17	8.02	9.70
$3 - \sigma$	99.73%	9	11.8	14.2	16.3

Table 2.1: A table showing the values of $\Delta\chi^2$ for fit parameters in varying M dimensions. Bold parameters are those used in this study (Press, Flannery, and Teukolsky 1986)

for, E_i , is continuous across a frequency range ν . As the continua produced by the model is evaluated over the same frequency range ν , the index of each of the observed points in ν is also the index of the intensity in the continua. By then evaluating Equation 1.6 using given values of N and T the expected continua can be determined. Subsequently pulling out the points within this continua that match the indices from the discrete points and applying the χ^2 test given in Equation 2.3 allowed the ‘goodness-of-fit’ of that given curve to the observed data to be evaluated. Systematically repeating this process across all values of N and T produces a χ^2 landscape. The minimum value of χ^2 within this landscape corresponds to the theoretical best fit; by locating the values of N and T that produced this minimised curve, the best fit parameters can be determined. This procedure was then applied to the next pixel in the image until all pixels had been analysed and their best fit parameters estimated, therefore producing best fit values of N and T at each pixel.

The χ^2 fitting routine is also the major source of error in N and T estimation and so these errors were accounted for by analysing the $\Delta\chi^2$ landscape. By expressing χ^2 as $\chi^2 = \chi_{min}^2 + \Delta\chi^2$, a region around χ_{min}^2 can be isolated by drawing lines of constant χ^2 characterised by $\Delta\chi^2$. These regions contain a varying percentage of the fit parameters depending upon the value of $\Delta\chi^2$ described in Table 2.1.

This project required 2 fit parameters in N and T . The values of $\Delta\chi^2$ used in this study are shown in bold in Table 2.1. The corresponding confidence interval is shown in the P column. All errors quoted hereafter correspond to the $1 - \sigma$ interval.

2.3. IMAGE CONSTRUCTION

2.3.1. 2D Maps of N and T

The result of the χ^2 minimisation is a best fit value of N and T at each pixel. 2-dimensional heat maps can be plotted that graphically illustrate the recovered values. This process is also repeated for the reconstructed values of N and T from the initial input data files `dust_density.inp` and `dust_temperature.dat` by Equations 2.4 and 2.5.

$$N_{col} = \sum \frac{\rho_d d_{cell}}{\mu m_p} \quad (2.4) \qquad T_{pix} = \frac{\sum_i T_i \rho_i}{\sum_i \rho_i} \quad (2.5)$$

Within Equation 2.4, N_{col} is the column density of the pixel, ρ_d is the dust density at that pixel, μ is the mean molecular weight, μ is the mean molecular weight, m_p is the mass of the proton and d_{cell} is the size of the RADMC-3D cell in the Z dimension. This is summed over the pixels along the line of sight.

Within Equation 2.5, T_{pix} is the temperature of the pixel, T_i and ρ_i are the column-weighted temperature and dust density at i th pixel along the line of sight. The RADMC-3D temperature inputs at each cell are largely independent of the density at that cell, however when observed the varying column density along the line of sight causes the temperature observed to vary as well. This is not the case at the source of emission, and Equation 2.5 accounts for that such that it determines the temperature of the dust as it was emitted and not as it was observed. It is this column-weighted temperature that the χ^2 routine fits for.

The temperature variation along the line of sight, $\sigma_{T_N}^2$ can also be computed using Equation 2.6.

$$\sigma_{T_N}^2 = \frac{\sum (T_{data} - T_{col})^2 N_{inp}}{\sum N_{inp}} \quad (2.6)$$

T_{data} is the initial data input temperature, T_{col} is the data derived temperature whilst N_{inp} is the data derived N . Computing maps of N and T from both the χ^2 recovered values as well as the theoretical predictions based on Equations 2.4 and 2.5 from the input data allows both data to be compared and contrasted.

Along with maps of N and T , we also produce probability density functions (PDFs). These allow the distribution of N and T within a given map to be analysed. It also allows a further measure of comparison between χ^2 and theoretical predictions.

2.3.2. Dendrogram analysis

As Section 1.1.2 stated, molecular clouds have a hierarchical structure according to their fragmentation and collapse. This produces a complex structure within the cloud, often with regions of having substructure according to any hierarchical fragmentation that it may undergo during collapse. This structure can be analysed using maps of N but it can also be probed with dendrogrammatic analysis. All dendograms within the study are plotted using `astrodendro` (Robitaille et al. n.d.).

A dendrogram is a graphical representation that shows the relationship between individual datapoints in a set by way of a tree diagram. An example dendrogram is shown Figure 2.5.

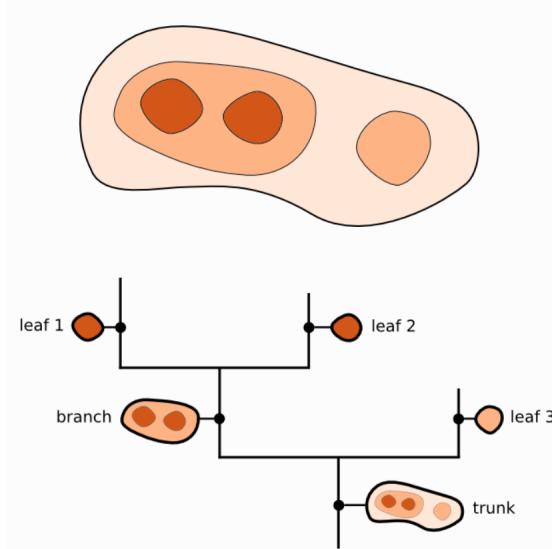


Figure 2.5: An idealised molecular cloud containing some substructure defined by sequentially darker regions. Below, the dendrogrammatic representation of the cloud (astrodendro 2016).

As Figure 2.5 shows, the dendrogram highlights the relationships between areas in the image. If Figure 2.5 represents an idealised molecular cloud then the fragmented structure represents regions of consecutively greater density. The dendrogram highlights this in a tree diagram format.

The initial vertical line of the dendrogram in Figure 2.5 is the trunk: the first representation of structure, and the parent from which all structures within it derive. From here the dendrogram probes the next level of structure, illustrated by the first vertex. A leaf represents a structure that has no further substructure, whilst a branch represents an area that has further structure. The ‘taller’ the dendrogram the greater the number of individual levels of substructure.

`astrodendro` allows the extraction of the leaf indices as well as the leaf values. This permits, in the case of the column density maps, the value of N to be found at each leaf. The mass in the given structure, M_{core} , is then given by

Equation 2.7.

$$M_{core} = N_{H_2} \mu m_p \pi R_{core}^2 \quad (2.7)$$

R_{core}^2 is the radius of the core. In reality, this core would be spherical, presenting to the observer as a circle. As such, $R_{core} = \sqrt{\frac{A_{tot}}{\pi}}$. A_{tot} is the total area of the pixels that make up the core. To discriminate between regions of high N and those that form a gravitationally bound core, the Jeans' mass and radius of a typical core ($n = 10^4 \text{ cm}^{-3}$ and $T = 15\text{K}$) can be used to determine a column density that the given leaves should have if they form a core, N_{core} . This was determined using Equation 2.8.

$$N_{core} = \frac{M_J}{\mu m_p R_J^2} \quad (2.8)$$

M_J is the Jeans mass (defined in Equation 1.1) and R_J is the Jeans radius (defined in Equation 1.2). A measure of whether a core is gravitationally bound or not can be observed in the Virial Ratio: $Q = T / -\Omega$, where $T = \frac{3/2 M_{core} k T}{\mu m_p}$ and $\Omega = -\frac{3/5 G M_{core}^2}{R_{core}^3}$. If $Q < 1$ then the core is bound, as the gravitational potential energy is greater than the thermal energy (the condition for collapse to occur).

Chapter 3

Results

3.1. ISOTHERMAL SPHERICAL CLOUD

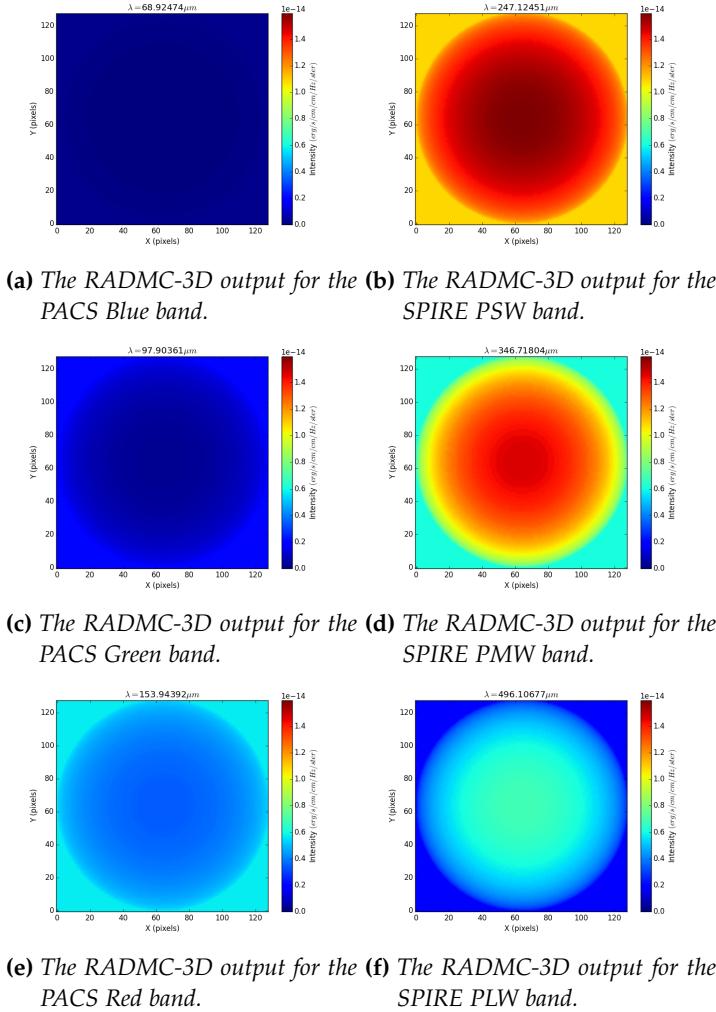


Figure 3.1: The RADMC-3D intensity outputs for the isothermal source at $T = 10\text{K}$ imaged in all 6 passbands considered.

Figure 3.1 shows the transmission weighted intensities for an idealised isothermal source of $T = 10\text{K}$ and $n_{\text{dust}} = 10^3 \text{cm}^{-3}$ in a $T = 15\text{K}$ and $n_{\text{dust}} = 1\text{cm}^{-3}$ background¹. The intensities vary depending on the filter used. This is

¹These values of n place the gas density at 10^5cm^{-3} , above that of the coupling threshold of 10^4cm^{-3} when gas begins to heat the dust. This scenario is ideal however, such that dust temperature was fixed and therefore independent of heating via gas

to be expected, as each filter's passband allow photons of given frequency - and therefore photons of given energy - to be detected. As the transmission curves in Figure 2.1 illustrates however, the intensities observed here are not the intensities that the source emits at owing the transmission coefficients. By comparing Figures 3.1a through to 3.1f, we observe the expected behaviour: the higher the frequency, the greater the intensity owing to the greater photon energy observed.

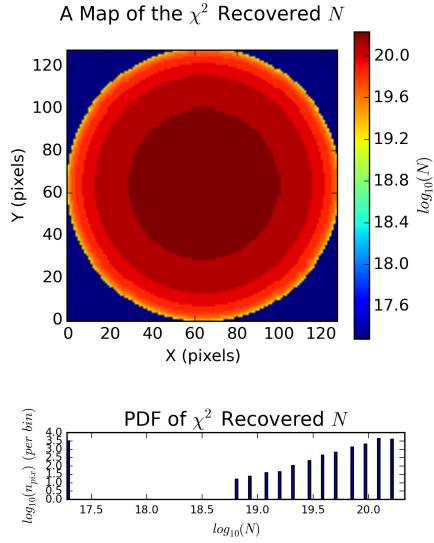


Figure 3.2: The recovered values of N from the χ^2 minimisation routine.

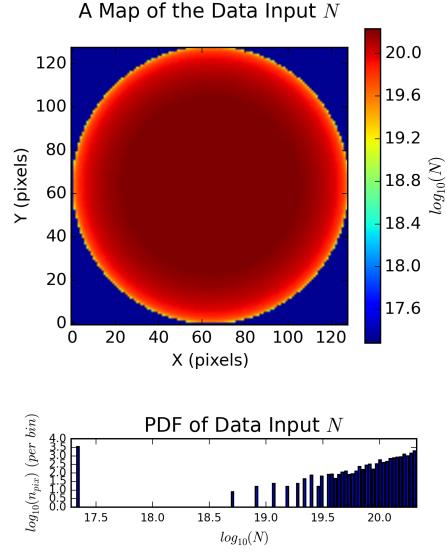


Figure 3.3: The reconstructed values of N from the initial input data.

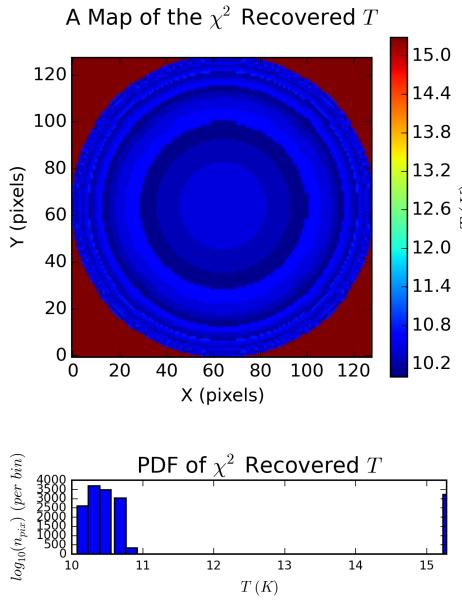


Figure 3.4: The recovered values of T from the χ^2 minimisation routine.

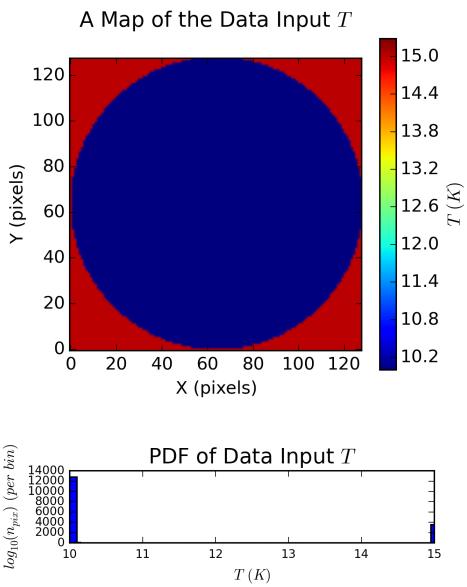


Figure 3.5: The reconstructed values of T from the initial input data.

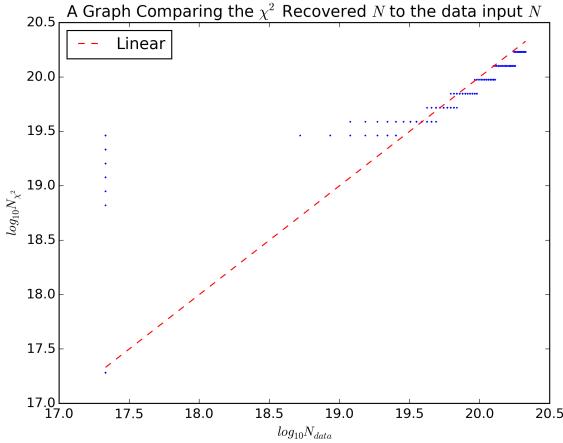


Figure 3.6: A representation of how the χ^2 derived N values compare to the initial data input N .

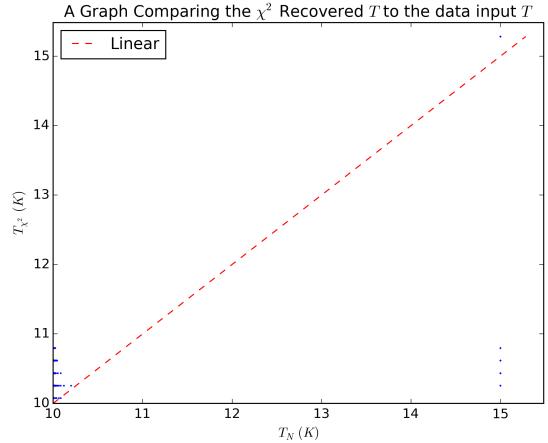


Figure 3.7: A representation of how the χ^2 derived T values compare to the initial data input T .

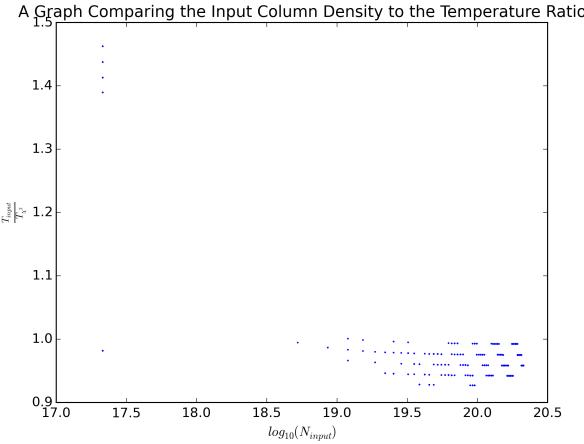


Figure 3.8: A plot showing how the ratio of $T_{\text{input}}/T_{\chi^2}$ varies with the input T .

Figure 3.1 is an isothermal sphere, meaning that very few line of sight temperature variations are present. As such, a single T assumption along the line of sight is accurate, implying that the modified blackbody in Equation 1.6 is a good fit to the data. The χ^2 recovered values of N and T (by fitting Equation 1.6), along with the data derived values, are shown in Figures 3.2, 3.3, 3.4 and 3.5. The χ^2 recovered values were derived from fitting to the SPIRE bands *only*, as the PACS bands were found to be inaccurate descriptors of dust emission intensity at the temperatures considered. Figures 3.2, 3.3, 3.4 and 3.5 also show PDFs of the data where n_{pix} is the number of pixels per bin.

By direct comparison of Figure 3.2 with Figure 3.3 it is observed that the χ^2 recovered N values match the derived N values well at large N (though the χ^2 values are underestimated at low N). This is also shown in Figure 3.7, which compares the χ^2 N to the data input N . Within Figure 3.7, the red line represents the linear relationship expected were the χ^2 recovered values to match the data derived value exactly. It clearly shows a wide degree of scatter across both input N and $\chi^2 N$. Figure 3.7 shows that χ^2 minimisation produces one estimate of N for a large degree of input N , especially at lower values of N . Of note here is the fact that this is especially true of high N , where the values differ by

a smaller degree relative to the low N regime where the spread is greater. This is also illustrated in the ratios shown in Figures 3.8.

Comparing Figure 3.4 with Figure 3.5 we see that χ^2 largely overestimates T relative to the data derived T . It also finds that the temperature decreases radially from the centre of the sphere, whereas Figure 3.5 shows T to be constant. This is also evident in the PDFs in both figures: Figure 3.5 shows 2 discrete bins at 10K and 15K, whereas Figure 3.4 displays a peak above 10K that decreases at 11K, as well as another peak above 15K. The overestimate of T is evident here, though the shape of PDFs in Figures 3.2 and 3.3 show similar shapes even with the underestimate evident. Likewise, the PDFs in Figures 3.4 and 3.5 exhibit the same behaviour. Whilst the shape of the PDF themselves remain similar, the χ^2 recovered quantities suffer from the discretisation of the values used to generate them, as evidenced by the discrete peaks in the PDFs. This is not evident in the data derived quantities, which show a smoother PDF. Furthermore, the spheres themselves are generated with discrete pixels, resulting in the column density (and temperature along the line of sight as this quantity is column weighted) increasing in steps rather than transitioning smoothly. This effect is exaggerated in the χ^2 recovery, as both N and T show banding.

3.2. AREPO DATA

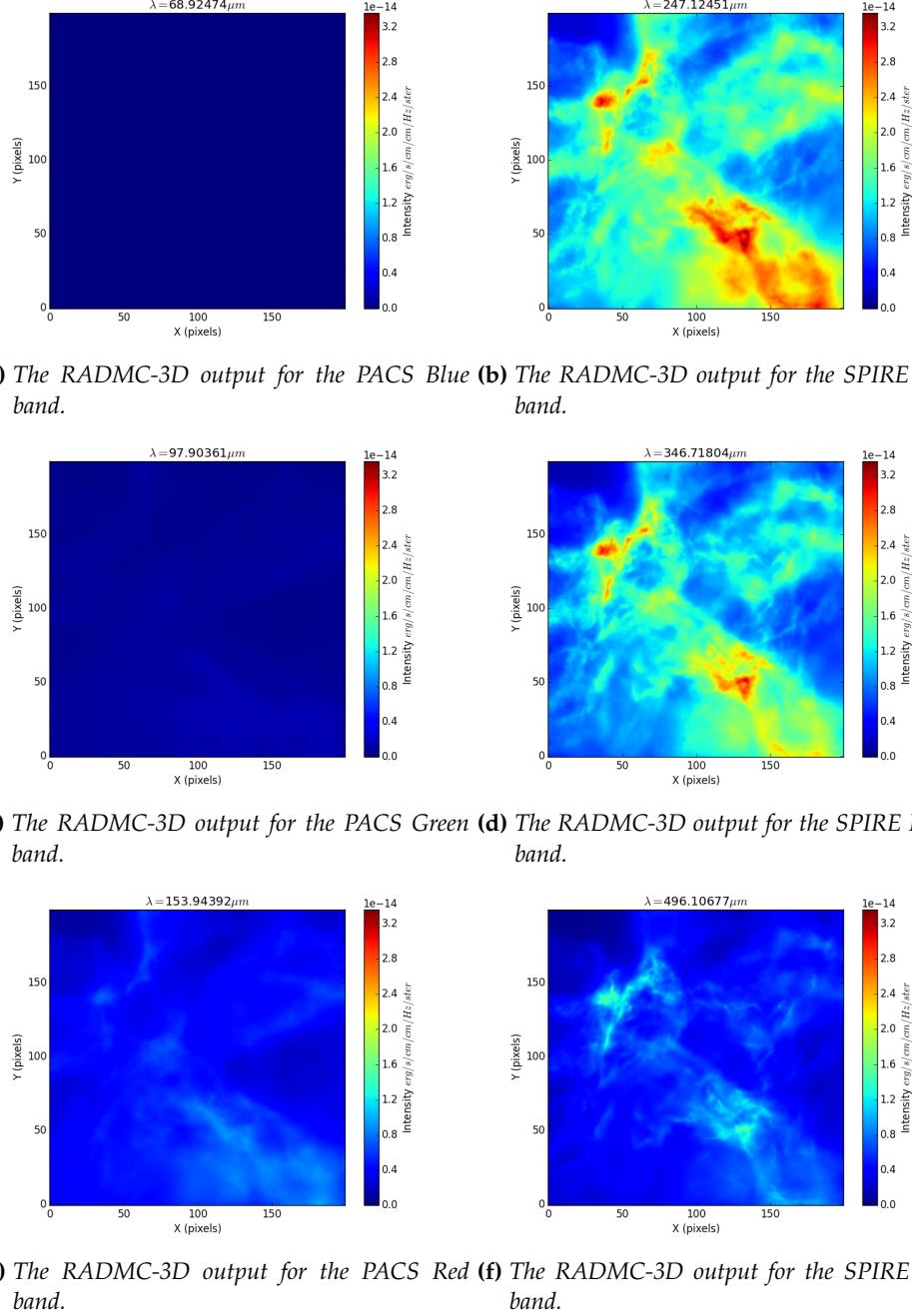


Figure 3.9: The RADMC-3D intensity outputs for the Arepo data imaged in all 6 passbands considered.

Figure 3.9 shows the RADMC-3D image outputs for the Arepo data. The same behaviour is observed here, as was observed in Figure 3.1. Figures 3.10, 3.11, 3.12 and 3.13 illustrate the χ^2 recovered values of N and T , as well as the data

input derived equivalents.

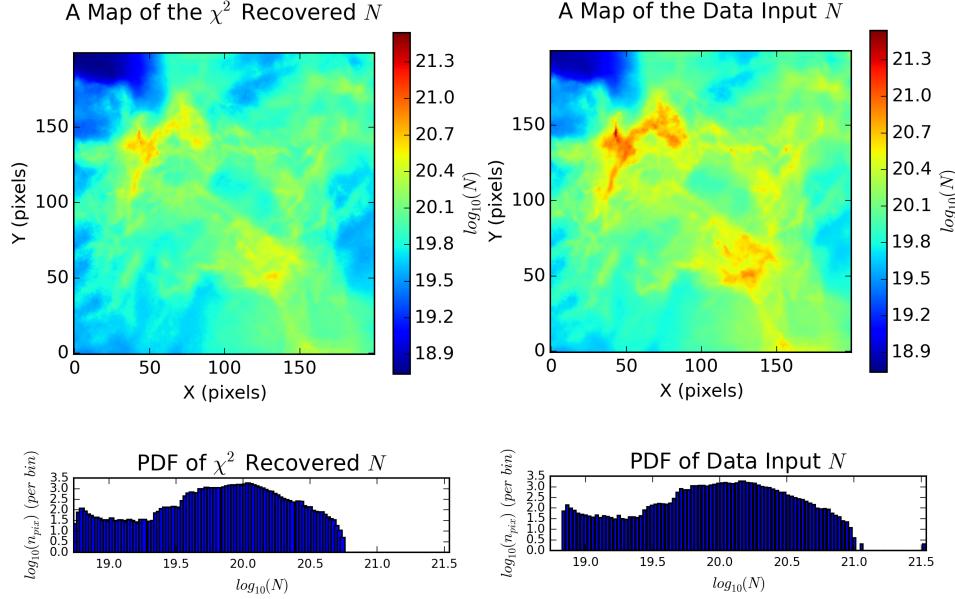


Figure 3.10: The recovered values of N from the χ^2 minimisation routine.

Figure 3.11: The reconstructed values of N from the initial input data.

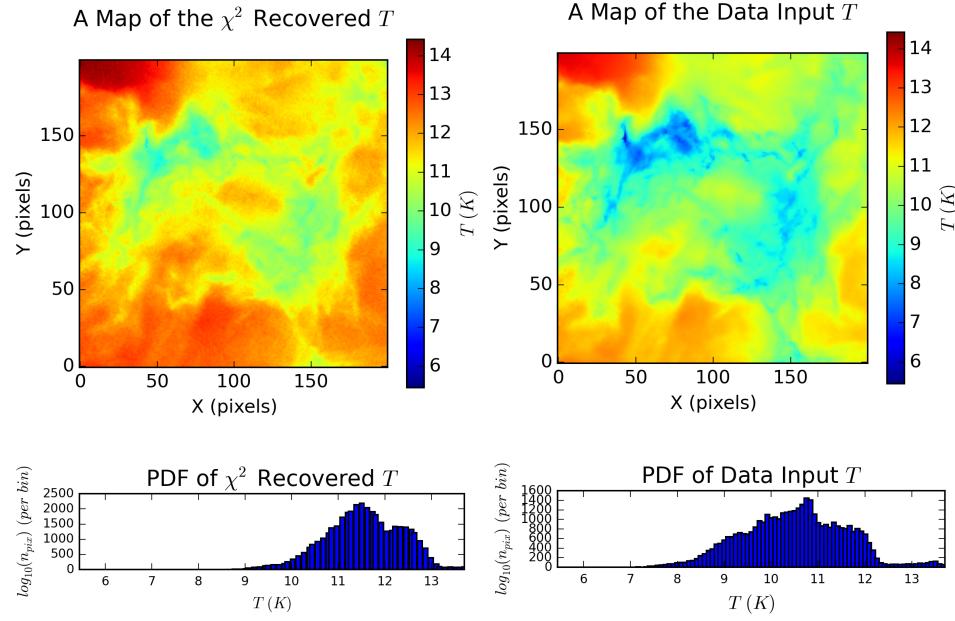


Figure 3.12: The recovered values of T from the χ^2 minimisation routine.

Figure 3.13: The reconstructed values of T from the initial input data.

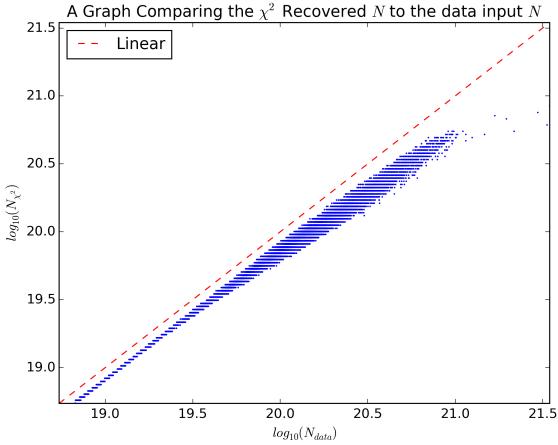


Figure 3.14: A representation of how the χ^2 derived N values compare to the initial data input N .

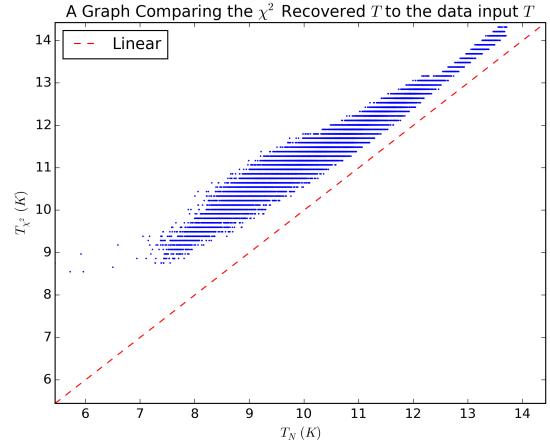


Figure 3.15: A representation of how the χ^2 derived T values compare to the initial data input T .

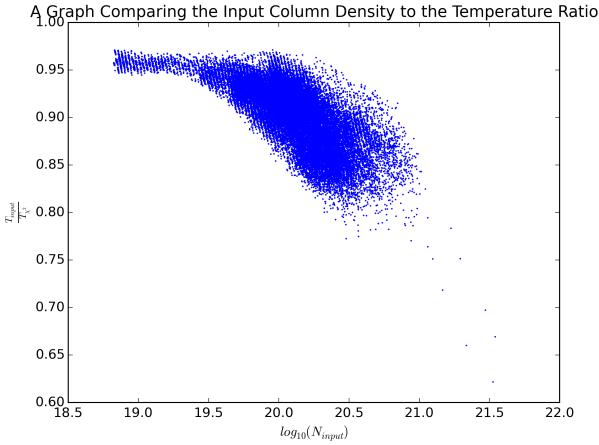


Figure 3.16: A plot showing how the ratio of $T_{\text{input}} / T_{\chi^2}$ varies with the data derived N .

We notice here, owing to the more intricate structure within the image, that χ^2 minimisation recovers the majority of that structure, despite failing to recover a number of obvious high density regions. Of particular interest here is the small peak at $\log_{10}(N) = 21.5$ that the χ^2 routine fails to detect. Similarly to the isothermal case in Figure 3.2, these values are underestimated to a small degree relative to the data derived N , as shown by the distribution existing further towards low N in the χ^2 case. We also note that far less of the higher N material is recovered, as evidenced by the maximum of the χ^2 N PDF falling below that of the data derived N ; data derived N extends continuously to $\log_{10}(N) = 21$, whilst the χ^2 continues to $\log_{10}(N) = 20.7$. However, the shape of the PDFs remain largely intact, similar to the isothermal case. One feature that is emergent here and not in the isothermal sphere however is the sharp peak in $\log_{10}(n_{\text{pix}})$ at small N in both χ^2 N and data derived N . This highlights the relative abundance of low column material within the image. The PDF also features an extended ‘tail’ such that the expected lognormal form is departed from at both low N as well as high N . In the high N case the ‘tail’ is best described by a power-law relation (Brunt 2015; Kainulainen et al. 2009). It is widely accepted that this power-law indicates the presence of gravitationally bound material.

Figures 3.12 and 3.13 illustrate that, again like the isothermal case, $\chi^2 T$ is overestimated relative to the data derived T . Also apparent is the fact that the χ^2 minimisation fails to recover a lot of the colder material in the image, as evidenced from the minimum of the $\chi^2 T$ PDF beginning above that of the data derived T PDF. Similar to the isothermal case, the shape of the PDFs remains consistent between χ^2 and data derived.

Figures 3.14 and 3.15 show how data derived quantities varies with the χ^2 recovered quantity. The linear reference line is plotted as a red line. Figure 3.14 shows that the χ^2 recovered N is now consistently underestimated with respect to the data derived N as shown by the data existing under the linear relationship line. This relationship is, however, linear in log space at low N though the disparity increases at higher data derived N . The scatter in data derived N that produces the same value of $\chi^2 N$ increases towards the middle of the data. Interestingly, this linear relationship is departed from at high N as evidenced by the data points falling away from the main grouping (and further away from the linear relationship line). Figure 3.15 shows the inverse relationship in T : the $\chi^2 T$ is now largely overestimated relative to the data derived T , though this overestimate is much larger at low data derived T . The same departure in datapoints is observed at low data derived T in this case (as opposed to high data derived quantity as was the case in Figure 3.14).

Figure 3.16 displays how the ratio of the determined temperatures $T_{\text{input}}/T_{\chi^2}$ varies with the data derived N . It clearly shows that as the data derived N increases the χ^2 recovered T is increasingly overestimated. Of note here is the fact that even in the low N regime, the χ^2 recovered N remains overestimated. This relationship continues into the high N regime, culminating in a sharp drop off that shows the χ^2 recovered T is almost 3 times the data derived T . Figure 3.16 also shows a large degree of scatter that increases with N , implying that not only can multiple different N values produce the same estimate of T_{χ^2} but also that this scatter increases as N increases. To further illustrate this, Figure 3.17 shows how the χ^2 recovered values of N and T compare.

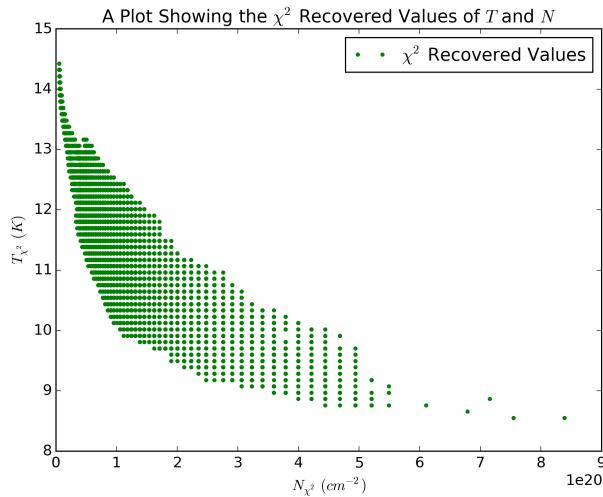


Figure 3.17: A plot showing the relationship between the χ^2 recovered values of N and T .

Figure 3.17 exhibits an elliptical shape, such that minimal scatter is observed in the minima and maxima of T and N . The largest scatter occurs in the centre of the data, where single T provides multiple differing values of N . Likewise, single N can provide multiple different values of T . A similar relationship is observed in Shetty et al. (2009a) and Shetty et al. (2009b) for β and T , who attribute the elliptical data to be a result of noise in the observations.

3.3. LINE OF SIGHT TEMPERATURE VARIATIONS

Figures 3.7, 3.6, 3.15, 3.14 as well as Figure 3.17 highlight underestimates of N and overestimates of T in the χ^2 minimisation test. Figures 3.18 and 3.19 show the line of sight temperature variations σ^2 as a function of the data derived N and T . The line of sight temperature variation is expressed as the standard deviation of the weighted mean temperature expressed in Equation 2.6.

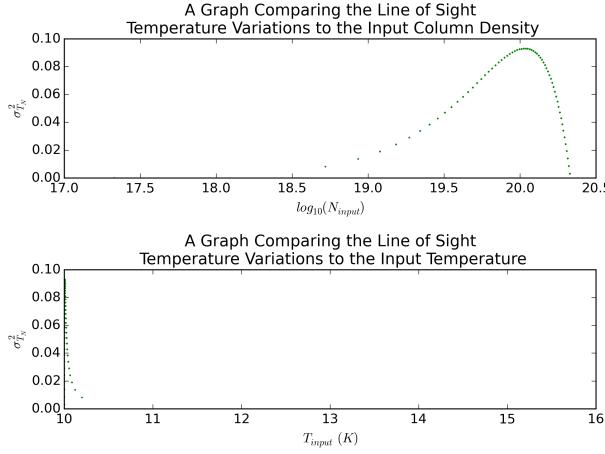


Figure 3.18: The line of sight temperature variations in the isothermal sphere case as a function of the data derived N and T .

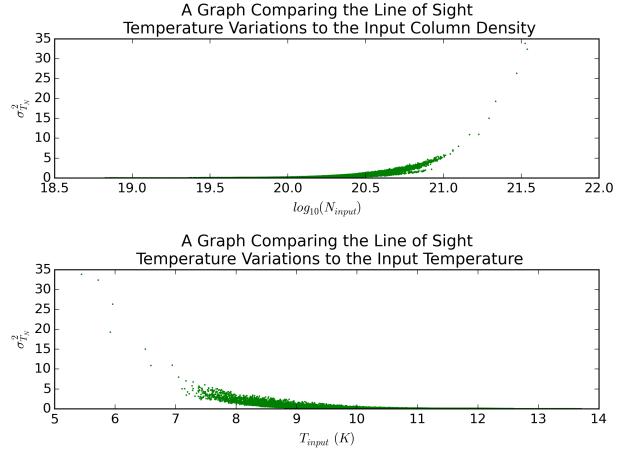


Figure 3.19: The line of sight temperature variations in the Arepo simulation case as a function of the data derived N and T .

Figure 3.18 clearly shows minimal line of sight temperature variations in the isothermal sphere. The greatest temperature variation occurs close to high N , whilst there is very little (if any) temperature variation at low N . Likewise the data derived temperature shows the inverse: there is far greater line of sight temperature variation at low T than there is at high T . Referring to Figure 3.3 we observe that the greatest column density is towards the centre of the sphere (likewise Figure 3.5 shows that the lowest temperature is in the same location) and the lowest column density is towards the image corners (greatest temperature is in these locations as well). These regions are of constant temperature (in this case, 15K) so minimal temperature variation is expected. The high column density region has a larger line of sight temperature variation owing to, as Figure 1.3 demonstrates, the larger number of boundary regions that the line of sight intercepts. Whilst the isothermal sphere does not have distinct intersections of changing dust density, it is spherical and therefore further away from the image centre ‘sees’ less of the sphere itself and more of the surrounding background. Of note here is the sharp drop off in σ^2 at very high N . To understand this, we must consider the image boundaries. By referring again to Figure 3.5, we see that the regions that lie within this N range essentially ‘touch’ the image boundaries. In 3-dimensional parameter space, this means that this point does not have any of the simulated background in front of (or behind it). This produces constant temperature along the line of sight, resulting in the sharp drop off observed. The curvature of the sphere results in points around this central ‘touch’ point having background material in front of and behind, therefore increasing the line of sight temperature variation. Figure 3.5 shows that a large portion (out to 100 pixels remains at a column density close to 10^{20} cm^{-2}). This corresponds to the peak in the temperature variation - the drop off in σ^2 then occurs from here as more of the background is seen along the line of sight. The other minimum in σ^2 is found to be at low N , where the background exists.

Repeating the same procedure on the Arepo data produces Figure 3.19. The same behaviour as was observed in Figure 3.18 is observed here: σ^2 increases with N and decreases with T . The constraining of σ^2 in N is remarkably minimal, whilst the scatter of σ^2 in T is far greater. This is to be expected, as the input temperature varies along the line

of sight for more in the Arepo instance than in the isothermal instance. We observe far greater temperature variations in Arepo data than in the isothermal sphere. This is to be expected, given the large range of temperatures (those derived from the input quantities) in the Arepo data.

Interestingly in Figure 3.19 we do not observe the sharp drop off in σ^2 at high N that is seen in Figure 3.18. This is to be expected however as the structure in Arepo data is transient and therefore not of uniform, spherical shape.

3.4. CORE EXTRACTION

Applying `astrodendro` to the column density maps in Figures 3.10 and 3.11 produces Figures 3.20 and 3.21.

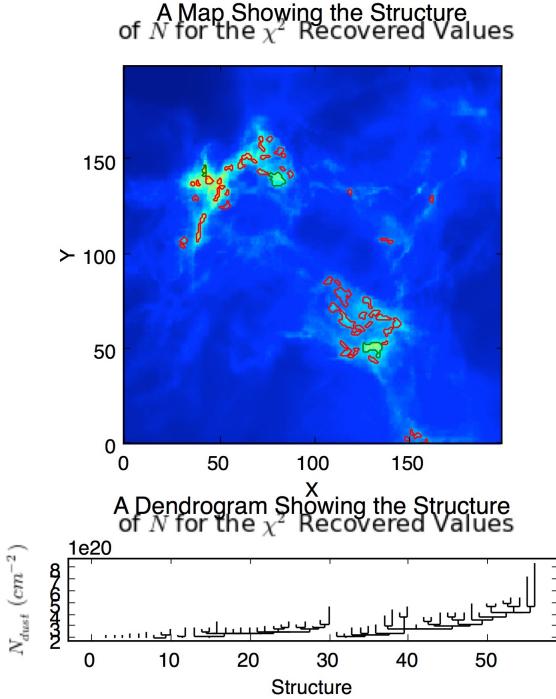


Figure 3.20: The χ^2 recovered N . The contours represent leaves of the dendrogram: a red contour represents an unbound core, whilst a green contour represents a bound core. The lower graph shows the dendrogram. In this case, it displays the structure of the leaves (cores).

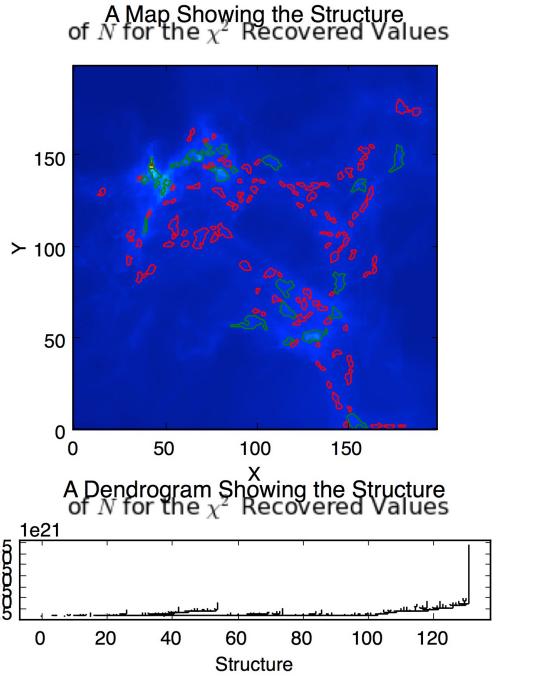


Figure 3.21: The data derived N . The contours represent leaves of the dendrogram: a red contour represents an unbound core, whilst a green contour represents a bound core. The lower graph shows the dendrogram. In this case, it displays the structure of the leaves (cores).

The conditions to extract these cores are discussed in Section 1.1.2. Figure 3.20 displays 3 bound cores, whilst Figure 3.21 displays 25 bound cores. The 3 bound cores extracted from the χ^2 N (Figure 3.20) are also extracted from the data derived N (Figure 3.21). Table 3.1 shows the masses for the cores recovered in both dendograms.

The larger number of core extractions from the data derived N is expected, as these values are larger than the χ^2 equivalents. This is also apparent in the mass estimates from Table 3.1, as the M_{χ^2} are consistently underestimated with respect to M_{data} . Again, this is to be expected as the column densities (and therefore mass) is lower in the χ^2 case relative to the data derived case.

The dendograms in Figures 3.20 and 3.21 show similar shapes - evidence that the χ^2 routine retains the substructure evident in the data. The dendrogram in Figure 3.21 extends to a greater N - this is due to the fact that the χ^2 routine

M_{χ^2}/M_{Sun}	M_{data}/M_{Sun}
0.3403 + / - 0.3616	0.7016
0.2799 + / - 0.3269	0.6241
0.1041 + / - 0.07105	0.7974

Table 3.1: A table showing the masses of cores extracting from the χ^2 N map and the data derived N map.

fails to pick up the small, dense regions of the map. However, Clark and Glover (2014) finds that the minimum density required for star formation to occur is $10^{21} cm^{-2}$. All of the regions highlighted in Figures 3.20 and 3.21 exceed this limit, implying that the cores observed are indeed sites of star formation.

Chapter 4

Discussion

Section 3.1 illustrates that whilst the χ^2 recovered values of N and T remain close to their data derived counterparts (especially in the low N and high T regime), the minimisation routine still underestimates N and overestimates T . In comparison, Section 3.2 shows largely the same: an accurate recovery of N and T in the low N and high T regime, though the same underestimate of N and overestimate of T is evident. Interestingly, Figure 3.17 displays the same relationship as observed in Shetty et al. (2009a) and Shetty et al. (2009b) who attribute this behaviour to noise in the observations. Shetty et al. (2009a) and Shetty et al. (2009b) does however note that this behaviour could be a property of the emitting dust and not, as is first stated, a noise effect. The study shown here would imply that the latter is true, given that the observations used here are based upon the Arepo simulation which has no traditional noise (e.g. Poission noise, dark currents etc.) as would be the case in ‘real’ observations. However, noise is still emergent in these results owing to the line of sight temperature variations, meaning the source of the elliptical χ^2 values could stem from here.

Interestingly the largest departure from the data derived value is when the column density N is high and the temperature T is low. Referring to Figure 1.3, high N along the line of sight occurs when there are a number of different boundaries, where a boundary is an increase/decrease in density from one medium to another. Given that high density regions are heated less, and therefore have a lower temperature T , we observe that regions of high N are also regions of large temperature variation. This is observed in Figure 3.18 and 3.19, which shows the standard deviation in temperature along the line of sight. The largest variation is observed to be at high N and at low T , with that variation exhibiting an exponential behaviour. Furthermore, the large temperature variations render the single T modified blackbody being fit to the data an inaccurate assumption. Even in the isothermal case where we observe very minimal temperature variations, T is still overestimated. Shetty et al. (2009a) and Shetty et al. (2009b) also find that this effect is worse when fitting to low λ data, hence the inaccurate values of N and T derived when using both the SPIRE and PACS bands to fit using χ^2 .

Whilst the PDFs in Figures 3.2 and 3.3 exhibit the same shape, their power-law tails are interesting. Brunt (2015) states that the PDF is best described by a lognormal in the turbulent region (central to the PDF), where gravity is yet to cause the gas to collapse. This relationship is observed here in Figures 3.2 and 3.3. This region is observed to have a power-law relationship (Kainulainen et al. 2009), which is thought to arise from the effect of gravity in the cold and molecular regions of the cloud. One could presume, therefore, that the presence of such a power-law here implies a contribution from gravitationally bound material. We observe from our dendrogrammatic extraction that gravitationally bound cores do indeed exist within the cloud, though the number of gravitationally bound cores extracted from both maps is inconsistent despite both extracting 3 of the same cores. The tail drop off in Figure 3.2 falling below the drop off in Figure 3.3 lends credence to this, in so far as significantly fewer cores are extracted from Figure 3.2 (i.e. the χ^2 recovered values) than from Figure 3.3 (i.e. the data derived values). Were χ^2 to recover more high N material, it may extract more cores thereby extending the power-law tail. As Brunt (2015) highlights, the peak in the PDF is attributable to the warmer, atomic material whilst the tail is attributable to the cold, molecular material. We find that the lack of high N and low T material - the material responsible for the power-law tail - severely limits a χ^2 test’s ability to extract cores that exist in this region of the PDF. Furthermore the underestimate of N results in a lower mass determined from the χ^2 (owing to Equation 2.7) relative to the data derived N .

Given the core masses the CMF could be plotted, however the lack of extracted cores in the project meant that this wasn’t possible. Further surveys (on either real data or larger synthetic data) would be required to build a set of cores large enough to plot an accurate CMF.

Chapter 5

Conclusion

In summary we have demonstrated, by constructing maps of column density N and temperature T from simulated molecular clouds, that the χ^2 recovered values are underestimated in N and overestimated in T with respect to the values of N and T determined from the initial input conditions used to produce the simulation. We have found that this is the case even in the absence of large line of sight temperature variations and observational noise. Furthermore we have generated dendograms of the constructed maps in order to extract the prestellar cores that exist within them, finding that both $\chi^2 N$ and data derived N contain differing numbers of gravitationally bound cores (3 in the χ^2 case and 25 in the data derived case), despite extracting 3 of the same cores across both values. This was attributed to the χ^2 minimisation failing to detect high N material, thereby halting the power-law tail observed in the $\log_{10}(N)$ PDFs therefore destroying evidence of cores at these densities. The core masses that were recovered from χ^2 were found to be underestimated relative to the data derived masses.

Chapter 6

Future work

Future work would focus on applying this technique to constraining the CMF, either through further simulated data or through real observations. Work would also commence on assessing how statistically valid the χ^2 recovered values of N T are in relation to their data derived (and therefore ‘true’) counterparts.

Future work would also focus on adding further variables to the simulation. This study considered 1 dust species: silicate grains. Adding further dust species, as well as gas, to the simulation would allow an analysis of dust and gas column density across both the image space itself as well as the sites isolated as cores by the dendrogram. When paired with a thermal Monte-Carlo simulation, the inclusion of gas would allow the effect of gas-dust temperature coupling on recovered N and T to be assessed. These proposals would also allow further study into the elliptically shaped $T - N$ graph as found by Shetty et al. (2009a) and Shetty et al. (2009b).

Chapter 7

Acknowledgements

I would like to thank Dr. Paul Clark for his guidance and encouragement as well as his proof-reading and advice (relating to both this project and matters external) throughout this academic year. I would also like to thank Dr. Sarah Ragan for her Python assistance and observational advice throughout the project. This project made heavy use of the Python programming language, RADMC-3D and astrodendro and their accompanying handbooks, web resources and tutorials.

Chapter 8

Appendix

CODE

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>. Below is the χ^2 fitting routine used in this project: chiv6.py. All further code used can be found at the GitHub repository above.

chiv6.py

```
#####
##### 4th Year Project Code #####
##### Chi Squared Routine v5 #####
#####

#####
##### Import statements #####
#####

# Import OS manipulation tools
import os

# Import radmc3dPy
import radmc3dPy
from radmc3dPy.natconst import *

# Import numpy
import numpy as np

# Import csv
import csv

# Import plotting tools
from matplotlib.pyplot import *

# Import glob
import glob

import random

#####
##### Define functions #####
#####

# Chi squared
def chi(O,E,sigma):
    """
    Returns simply the minimised chi squared value for a given dataset.

    O is the flux from the actual data.
    Sigma is the square root of the variance of O
    E is the data that is expected
    """
    return sum(((O-E)/sigma)**2)

# Modified blackbody
def mbb(N,dust_mass,opac,v,T):
    """
    Function that returns the modified black body law that best describes cold dust emission as stated in Kelly et al. (2012).

    This returns the intensity of the source over the beam.
    """

```

```

#a = (2*(hh)*(cc)**2)/(wav**5)
#b = ((hh)*(cc))/(wav*(kk)*T)
a = (2*(hh)*(v**3))/(cc**2)
b = (hh*v)/(kk*T)
return (N*muh2*mp*opac*(a*(1./np.exp(b)-1)))

#####
# Read the average data determined earlier #####
# Initialise a list to store the read in data
data, filenames = [], []

# Initiate counter to help store the passbands
count = -1

# Loop through all files that contain average data
for file in glob.glob('*_average_data.txt'):
    # Open them
    with open(file) as f:
        count += 1

        # Read in using loadtxt and then append the data to a list for use later
        contents = np.loadtxt(file, delimiter=",")

        # Append the name of the file to the position (count+len(data)) followed by
        # the file contents
        data.insert(count+len(data), file)
        data.append(contents)
        filenames.append(file)

# Assign to variables and put fluxes into an array (data plotted later)
blue = data[data.index(filenames[0])+1]
green = data[data.index(filenames[1])+1]
plw = data[data.index(filenames[2])+1]
pmw = data[data.index(filenames[3])+1]
psw = data[data.index(filenames[4])+1]
red = data[data.index(filenames[5])+1]

# Extract meaningful data from input files
v_data = np.array([psw[:,4],pmw[:,4],plw[:,4],blue[:,4],green[:,4],red[:,4]])
flux = np.array([psw[:,5],pmw[:,5],plw[:,5],blue[:,5],green[:,5],red[:,5]])
flux_error = np.array([psw[:,6],pmw[:,6],plw[:,6],blue[:,6],green[:,6],red[:,6]])

# Read the initial radmc3dPy output to get image dimensions and info
#imag = radmc3dPy.image.readImage('../..../data/blue/dust_project/image.out')

#####
# Determine (again) the opacity #####
# Query for reference wavelength
w_ref = 250e-4
v_ref = cc/w_ref
print 'The reference wavelength is taken to be', w_ref, 'cm which equates to a frequency of', v_ref, 'Hz.\n'

# Ask for reference opacity
kappa_0 = 4.0
print '\nThe k_0 opacity is taken to be', kappa_0, 'cm^2/g as per Ossenkopf and Henning, 1994. According to http://arxiv.org/pdf/1302.56

# Define the start and end points of the spectrum
lambda_init = 55.670637e-4
lambda_fin = 690.8139e-4
v_init = cc/lambda_init
v_fin = cc/lambda_fin
nlam = 600

# Create array of wavelengths from range given and reshape
w = np.linspace(lambda_init, lambda_fin, nlam)
v = np.linspace(v_init, v_fin, nlam)

# Find the index of the frequency that most closely matches the frequency of the band (i.e. find the difference between the two and find
psw_index = min(enumerate(v, key=lambda x: abs(x[1]-psw[0,4]))
pmw_index = min(enumerate(v, key=lambda y: abs(y[1]-pmw[0,4]))
plw_index = min(enumerate(v, key=lambda z: abs(z[1]-plw[0,4]))
blue_index = min(enumerate(v, key=lambda a: abs(a[1]-blue[0,4]))
```

```

green_index = min(enumerate(v), key=lambda b: abs(b[1]-green[0,4]))
red_index = min(enumerate(v), key=lambda c: abs(c[1]-red[0,4]))

# Solid angle of the beam
sigma_arc = 1768 # 465 square arcseconds
sigma_beam = sigma_arc*(1/4.24e10) # 465 square arcseconds in steradians

# Ask for spectral index
B = 2.0

# Evaluate opacities
opacity = kappa_0*(v/v_ref)**B

##### Determine the expected column density #####
print 'Entering the column density determination script\n'

# Dust to gas ratio
d2g = 0.01

# Read in the dust density information
dust_density = np.loadtxt('../data/blue/dust_project/dust_density.inp', skiprows=3)
dust_temperature = np.loadtxt('../data/blue/dust_project/dust_temperature.dat', skiprows=3)

# Create file to store the values
datafeed_store = open('datafeed.txt', 'w')
df = csv.writer(datafeed_store, delimiter='|')

# Because of the format of the dust_density file, create a list of the indices that correspond to the pixel values. The file is based on
imag = radmc3dPy.image.readImage('../data/blue/dust_project/image.out')
xpix, ypix, zpix = np.arange(0,imag.nx), np.arange(0,imag.ny), np.arange(0,(len(dust_density)/(imag.nx*imag.ny))) 

# Create list to store values of dust density summed along every x,y coordinate
dust_density_line, T_line, col_full, T_full = [], [], [], []

# Ask for distance to source
#d = input('At what distance is the source? This answer should be in parsecs.\n')
d = 150
D = np.float64(d)*pc # Convert to cm

# The mass of 1 dust grain is simply 1./100th the mass of Hydrogen
dust_mass = muh2*mp*d2g

# Determine solid angle of the pixel
#sigma_pix = (imag.sizepix_x*imag.sizepix_y)/D**2
sigma_pix = sigma_beam

# Instantiate a counter and a number of lists to store values (and for diagnostics)
count = 0
store_density, store_loc, store_temp = [], [], []

# Loop over the 2D image square cube
for i in range(0,len(xpix)*len(ypix)):

    # Add to the counter
    count += 1

    # Finds the locations of every each pixel's Z values
    loc = np.arange(i,len(zpix)*(len(xpix)*len(ypix)),len(xpix)*len(ypix))

    # Sums these values (i.e. sums along the line of sight) and stores the locations
    store_density.append(sum(dust_density[loc]))
    store_loc.append(loc)

    # The dust density is dust_density along the line of sight and so therefore the dust mass in one pixel along the line of sight is due
    #dust_mass_pixel = (sum(dust_density[loc]))*(imag.sizepix_x*imag.sizepix_y)*(len(dust_density[loc])*imag.sizepix_y)

    # Account for the column-weighted temperature
    col_T = np.sum((dust_temperature[loc]*dust_density[loc])/(np.sum(dust_density[loc])))

    # Repeat similar procedure for the temperature
    store_temp.append(col_T)

```

```

# Determine the number of dust grains in the pixel
#N_d = (dust_mass_pixel/(dust_mass*len(dust_density[loc])))
#N_d = dust_mass_pixel/dust_mass

# From Ward-Thompson and Whitworth, column density is the number of dust grains per unit area
#col = N_d/((D**2)*(sigma_pix))
col = sum((dust_density[loc])/(mu_h2*mp)*imag.sizepix_y)

# Assign all of the writable items to a variable for easier write
df_towrite = [i, col, col_T]

# Save to a file
df.writerow(df_towrite)

col_full.append(col)
T_full.append(col_T)

#N = np.linspace(np.log10(min(col_full)/100), np.log10(max(col_full)*100), 40)
N = np.linspace(18, 24, 40)

# T is independent of the column density in determination so this remains unchanged
#T = np.linspace(min(dust_temperature)-4,max(dust_temperature)+4,40)
#T = np.logspace(np.log10(5),np.log10(15),80,base=10)
T = np.linspace(5, 20, 40)

# Create 2 2D arrays of the data to track the progress of the loop
T_mesh, N_mesh = np.meshgrid(T,N)

datafeed_store.close()

print 'Column_densities_have_been_evaluated_and_have_been_saved_to_the_file_datafeed_store.txt,\n'

#N = np.linspace(np.round(col,-22),2*np.round(col,-22),10000)
#A = (imag.sizepix_x*imag.sizepix_y) # Area of one pixel in cm

##### Determine the modified black body #####
print 'Now_determining_the_modified_blackbody_curves.\n'

chi_min_index_all, chi_min_blackbody_all = [], []

# Save this to a file for storage
chi_store = open('chi_coarse.txt', 'w')
cs = csv.writer(chi_store, delimiter='')

# Save a line to allow better understanding of the columns
cs_towrite = ['Index', 'Column_Density', 'Temperature', 'Minimised_Chisquared', 'One-Sigma_N', 'One-Sigma_T', 'Two-Sigma_N', 'Two-Sigma_T']
cs.writerow(cs_towrite)

n = random.sample(range(0,imag.nx*imag.ny), 4)

for h in range(0,imag.nx*imag.ny):

    # Loop through each column density and determine modified black body curve
    mod, chivals, N_index, T_index = [], [], [], []

    # Loop through both N and T to determine the blackbody
    for i in range(0,len(N)):
        for j in range(0,len(T)):
            blackbody = mbb(10**N[i],dust_mass,opacity,v,T=T[j])

            # Append the value of the given blackbody to a list for storage
            mod.append(blackbody)

            # Append the given values of N and T
            N_index.append(10**N[i])
            T_index.append(T[j])

    # Define lists to store band fluxes
    to_fit_psw_list, to_fit_pmw_list, to_fit_plw_list, to_fit_blue_list, to_fit_green_list, to_fit_red_list = [], [], [], [], []

```

```

# Find the flux at the index determined earlier
to_fit_psw = blackbody[psw_index[0]]
to_fit_pmw = blackbody[pmw_index[0]]
to_fit_plw = blackbody[plw_index[0]]
to_fit_blue = blackbody[blue_index[0]]
to_fit_green = blackbody[green_index[0]]
to_fit_red = blackbody[red_index[0]]

# Put the fitting fluxes into an array
#to_fit = np.array([to_fit_plw, to_fit_pmw, to_fit_psw, to_fit_red, to_fit_green, to_fit_blue])
#to_fit = np.array([to_fit_psw,to_fit_pmw,to_fit_plw,to_fit_blue])
to_fit = np.array([to_fit_plw, to_fit_pmw, to_fit_psw])

# Takes the 6 data points for each pixel and puts them on a list to allow easier assignment
points = np.array([flux[2][h],flux[1][h],flux[0][h]])
points_error = np.array([flux_error[2][h],flux_error[1][h],flux_error[0][h]])

# Append the chi squared value
chisquared = chi(to_fit,points,sigma=points_error)

if chisquared == np.inf:
    chivals.append(np.nan)
else:
    chivals.append(chisquared)
#chivals.append(chi(to_fit,ps,sigma=[psw[2],pmw[2],plw[2],blue[2]]))

#print str('Found the chi-squared landscape. Moving to the next values...\n')

if h == imag.nx*imag.ny/10:
    print '\r[=>_____]_10%'
elif h == 2*(imag.nx*imag.ny)/10:
    print '\r[==>_____]_20%'
elif h == 3*(imag.nx*imag.ny)/10:
    print '\r[====>_____]_30%'
elif h == 4*(imag.nx*imag.ny)/10:
    print '\r[=====>_____]_40%'
elif h == 5*(imag.nx*imag.ny)/10:
    print '\r[======>_____]_50%'
elif h == 6*(imag.nx*imag.ny)/10:
    print '\r[======>_____]_60%'
elif h == 7*(imag.nx*imag.ny)/10:
    print '\r[======>_____]_70%'
elif h == 8*(imag.nx*imag.ny)/10:
    print '\r[======>_____]_80%'
elif h == 9*(imag.nx*imag.ny)/10:
    print '\r[======>_____]_90%'
elif h == (imag.nx*imag.ny)-1:
    print '\r[======>_____]_100%'

# Determine the chi squared minimum
chi_min_index = chivals.index(min(chivals))
chi_min_blackbody = mod[chi_min_index]

##### Error analysis begins here #####
# Reshape the chivals to fit the format of N_mesh and T_mesh
chivals_mesh = np.reshape(chivals, (len(T),len(N)))

# Determine error in the value by finding location of points that lie within 3 sigma
# Start by finding the location in the mesh of the minimum chi-squared value
val = zip(*np.where(chivals_mesh == np.amin(chivals_mesh)))

# According to Data Analysis notes 2D distribution has the following values of delta-chi-squared
delta_1 = 2.3 #(one_sigma)
delta_2 = 6.17 #(two_sigma)
delta_3 = 11.8 #(three_sigma)

# Find the locations of the points within these bounds
one_sigma = zip(*np.where((chivals_mesh >= chivals_mesh[val[0]]-delta_1) & (chivals_mesh <= chivals_mesh[val[0]]+delta_1)))
two_sigma = zip(*np.where((chivals_mesh >= chivals_mesh[val[0]]-delta_2) & (chivals_mesh <= chivals_mesh[val[0]]+delta_2)))
three_sigma = zip(*np.where((chivals_mesh >= chivals_mesh[val[0]]-delta_3) & (chivals_mesh <= chivals_mesh[val[0]]+delta_3)))

```

```

T_temp, N_temp = [], []
T_error, N_error = [], []

# Pull the values of N and T at these points
for b in [one_sigma, two_sigma, three_sigma]:
    for a in b:
        T_temp.append(T_mesh[a])
        N_temp.append(10**N_mesh[a])

    T_error.append(max(T_temp) - min(T_temp))
    N_error.append(max(N_temp) - min(N_temp))

# Append this value to a list to store the values
chi_min_index_all.append(chi_min_index)
chi_min_blackbody_all.append(chi_min_blackbody)

cs_towrite = [h, N_index[chi_min_index], T_index[chi_min_index], min(chivals), N_error[0], T_error[0], N_error[1], T_error[1], N_err

# Write to file
cs.writerow(cs_towrite)
#print 'Writing row to datafile...\\n'

if h == n:
    # Plot the data
    figure(1)
    errorbar(psw[0][4], psw[0][5], yerr=psw[0][-1], fmt='co', label='SPIRE:PSW')
    errorbar(pmw[0][4], pmw[0][5], yerr=pmw[0][-1], fmt='yo', label='SPIRE:PMW')
    errorbar(plw[0][4], plw[0][5], yerr=plw[0][-1], fmt='mo', label='SPIRE:PLW')
    errorbar(blue[0][4], blue[0][5], yerr=blue[0][-1], fmt='bx', label='PACS:Blue')
    errorbar(green[0][4], green[0][5], yerr=green[0][-1], fmt='gx', label='PACS:Green')
    errorbar(red[0][4], red[0][5], yerr=red[0][-1], fmt='rx', label='PACS:Red')
    plot(v, chi_min_blackbody, label=str('\\chi^2') + str('Minimum:' + str(N_index[chi_min_index])) + str('+' + '-') + str(N_error[2]/2) +
    xlabel(r'$\nu$ / Hz')
    ylabel(r'Intensity $(erg/cm^2/s/Hz/ster)$')
    xscale("log", nonposx='clip')
    yscale("log", nonposx='clip')
    grid(True, which='both')
    legend(loc='best', prop={'size':8})
    title(str('The $\chi^2$ Minimised Best Fit SED for PACS and SPIRE Bands\nfor T < 11K at the') + str(h) + str('th Pixel\\n'))
    savefig(str('imgdump/coarse/averages_coarse') + str(h) + str('.png'), dpi=300)
    close()

# Also plot probability contour
figure(2)
#plot(10**N_mesh, T_mesh, 'b.')
plot(N_index[chi_min_index], T_index[chi_min_index], 'r.', label=str(r'$\chi^2$ Minimum=' + str(min(chivals))))
CS = contour(10**N_mesh, T_mesh, chivals_mesh)
clabel(CS)
xlabel(r'$N / (g/cm^{-3})$')
ylabel(r'$T / (K)$')
title(r'$\chi^2$ Contours')
legend(loc='best', prop={'size':8})
savefig(str('imgdump/coarse/contours_') + str(h) + str('.png'), dpi=300)
close()

# And the chi-squared landscape
figure(3)
subplot(2,1,1)
plot(T_index, chivals, 'bo')
plot(T_index[chi_min_index], chivals[chi_min_index], 'r.')
xlabel(r'$T / (K)$')
ylabel(r'$\chi^2$')

subplot(2,1,2)
plot(N_index, chivals, 'bo')
plot(N_index[chi_min_index], chivals[chi_min_index], 'r.')
xlabel(r'$N / (g/cm^{-2})$')
ylabel(r'$\chi^2$')

savefig(str('imgdump/coarse/landscape_') + str(h) + str('.png'), dpi=300)
close()

```

```

chi_store.close()

#####
# Repeat for a second time #####
# Reopen the chi-squared storage
chi_coarse = np.loadtxt('chi_coarse.txt', skiprows=1)

# Determine the minimum and maximum column densities and temperatures
min_N_loc, max_N_loc = np.where(chi_coarse[:,1] == chi_coarse[:,1].min())[0][0], np.where(chi_coarse[:,1] == chi_coarse[:,1].max())[0][0]
min_T_loc, max_T_loc = np.where(chi_coarse[:,2] == chi_coarse[:,2].min())[0][0], np.where(chi_coarse[:,2] == chi_coarse[:,2].max())[0][0]

min_N, max_N = chi_coarse[:,1][min_N_loc], chi_coarse[:,1][max_N_loc]
min_T, max_T = chi_coarse[:,2][min_T_loc], chi_coarse[:,2][max_T_loc]

min_N, max_N = np.min(chi_coarse[:,1]), np.max(chi_coarse[:,1])
min_T, max_T = np.min(chi_coarse[:,2]), np.max(chi_coarse[:,2])

print 'Now determining the (new) modified blackbody curves.\n'

# Define the new N and T
N = np.linspace(np.log10(min_N-min_N/4), np.log10(max_N+min_N/4), 100)
#T = np.logspace(np.log10(min_T-min_T/4),np.log10(max_T+min_T/4), 80,base=10)
T = np.linspace(min_T-min_T/4, max_T+min_T/4, 100)

# Create 2D arrays of the data to track the progress of the loop
T_mesh, N_mesh = np.meshgrid(T,N)

print 'I am now considering N from', min(N), 'to', max(N), 'and T from'
print min(T), 'to', max(T), '\n'

chi_min_index_all, chi_min_blackbody_all = [], []

# Save this to a file for storage
chi_fine = open('chi_fine.txt', 'w')
cs = csv.writer(chi_fine, delimiter=' ')

# Save a line to allow better understanding of the columns
cs_towrite = ['Index', 'Column_Density', 'Temperature', 'Minimised_Chi-Squared', 'One-Sigma_N', 'One-Sigma_T', 'Two-Sigma_N', 'Two-Sigma_T']
cs.writerow(cs_towrite)

n = random.randint(0,imag.nx*imag.ny)

for h in range(0,imag.nx*imag.ny):

    # Loop through each column density and determine modified black body curve
    mod,chivals,N_index,T_index = [],[],[],[]

    # Loop through both N and T to determine the blackbody
    for i in range(0,len(N)):
        for j in range(0,len(T)):
            blackbody = mbb(10**N[i],dust_mass,opacity,v,T=T[j])

            # Append the value of the given blackbody to a list for storage
            mod.append(blackbody)

            # Append the given values of N and T
            N_index.append(10**N[i])
            T_index.append(T[j])

        #print 'N=',10**N[i],'and T=',T[i]

        # Define lists to store band fluxes
        to_fit_psw_list, to_fit_pmw_list, to_fit_plw_list, to_fit_blue_list, to_fit_green_list, to_fit_red_list = [], [], [], [], []

        # Find the flux at the index determined earlier
        to_fit_psw = blackbody[psw_index[0]]
        to_fit_pmw = blackbody[pmw_index[0]]
        to_fit_plw = blackbody[plw_index[0]]
        to_fit_blue = blackbody[blue_index[0]]
        to_fit_green = blackbody[green_index[0]]
        to_fit_red = blackbody[red_index[0]]
```

```

# Put the fitting fluxes into an array
#to_fit = np.array([to_fit_plw, to_fit_pmw, to_fit_psw, to_fit_red, to_fit_green, to_fit_blue])
to_fit = np.array([to_fit_plw,to_fit_pmw,to_fit_psw])

# Takes the 6 data points for each pixel and puts them on a list to allow easier assignment
points = np.array([flux[2][h],flux[1][h],flux[0][h]])
points_error = np.array([flux_error[2][h],flux_error[1][h],flux_error[0][h]])
#points = np.array([flux[2][h],flux[1][h],flux[0][h],flux[5][h],flux[4][h],flux[3][h]])
#points_error = np.array([flux_error[2][h],flux_error[1][h],flux_error[0][h],flux_error[5][h],flux_error[4][h],flux_error[3][h]])

# Append the chi squared value
chisquared = chi(to_fit,points,sigma=points_error)

if chisquared == np.inf:
    chivals.append(np.nan)
else:
    chivals.append(chisquared)
#chivals.append(chi(to_fit,ps,sigma=[psw[2],pmw[2],plw[2],blue[2]]))

#print str('Found the chi-squared landscape. Moving to the next values...\n')

if h == imag.nx*imag.ny/10:
    print '\r[=>_____]_10%'
elif h == 2*(imag.nx*imag.ny)/10:
    print '\r[==>_____]_20%'
elif h == 3*(imag.nx*imag.ny)/10:
    print '\r[====>_____]_30%'
elif h == 4*(imag.nx*imag.ny)/10:
    print '\r[=====>_____]_40%'
elif h == 5*(imag.nx*imag.ny)/10:
    print '\r[======>_____]_50%'
elif h == 6*(imag.nx*imag.ny)/10:
    print '\r[=======>_____]_60%'
elif h == 7*(imag.nx*imag.ny)/10:
    print '\r[=======>_____]_70%'
elif h == 8*(imag.nx*imag.ny)/10:
    print '\r[=======>__]_80%'
elif h == 9*(imag.nx*imag.ny)/10:
    print '\r[=======>_]_90%'
elif h == (imag.nx*imag.ny)-1:
    print '\r[=======>]_100%'

# Determine the chi squared minimum
chi_min_index = chivals.index(min(chivals))
chi_min_blackbody = mod[chi_min_index]

# Append this value to a list to store the values
chi_min_index_all.append(chi_min_index)
chi_min_blackbody_all.append(chi_min_blackbody)

##### Error analysis begins here #####
# Reshape the chivals to fit the format of N_mesh and T_mesh
chivals_mesh = np.reshape(chivals, (len(T),len(N)))

# Determine error in the value by finding location of points that lie within 3 sigma
# Start by finding the location in the mesh of the minimum chi-squared value
val = zip(*np.where(chivals_mesh == np.amin(chivals_mesh)))

# According to Data Analysis notes 2D distribution has the following values of delta-chi-squared
delta_1 = 2.3 #(one_sigma)
delta_2 = 6.17 #(two_sigma)
delta_3 = 11.8 #(three_sigma)

# Find the locations of the points within these bounds
one_sigma = zip(*np.where((chivals_mesh >= chivals_mesh[val[0]]-delta_1) & (chivals_mesh <= chivals_mesh[val[0]]+delta_1)))
two_sigma = zip(*np.where((chivals_mesh >= chivals_mesh[val[0]]-delta_2) & (chivals_mesh <= chivals_mesh[val[0]]+delta_2)))
three_sigma = zip(*np.where((chivals_mesh >= chivals_mesh[val[0]]-delta_3) & (chivals_mesh <= chivals_mesh[val[0]]+delta_3)))

T_temp, N_temp = [], []
T_error, N_error = [], []

```

```

# Pull the values of N and T at these points
for d in [one_sigma, two_sigma, three_sigma]:
    for c in d:
        T_temp.append(T_mesh[c])
        N_temp.append(10**N_mesh[c])

T_error.append(max(T_temp) - min(T_temp))
N_error.append(max(N_temp) - min(N_temp))

# Append this value to a list to store the values
chi_min_index_all.append(chi_min_index)
chi_min_blackbody_all.append(chi_min_blackbody)

cs_towrite = [h, N_index[chi_min_index], T_index[chi_min_index], min(chivals), N_error[0], T_error[0], N_error[1], T_error[1], N_error[2]/2]

# Write to file
cs.writerow(cs_towrite)
#print 'Writing row to datafile...\\n'

if h == n:
    # Plot the data
    figure(1)
    errorbar(psw[0][4], psw[0][5], yerr=psw[0][-1], fmt='co', label='SPIRE:\u209cPSW')
    errorbar(pmw[0][4], pmw[0][5], yerr=pmw[0][-1], fmt='yo', label='SPIRE:\u209cPMW')
    errorbar(plw[0][4], plw[0][5], yerr=plw[0][-1], fmt='mo', label='SPIRE:\u209cPLW')
    errorbar(blue[0][4], blue[0][5], yerr=blue[0][-1], fmt='bx', label='PACS:\u209cBlue')
    errorbar(green[0][4], green[0][5], yerr=green[0][-1], fmt='gx', label='PACS:\u209cGreen')
    errorbar(red[0][4], red[0][5], yerr=red[0][-1], fmt='rx', label='PACS:\u209cRed')
    plot(v, chi_min_blackbody, label=str('$\chi^2$') + str('_Minimum:' + str(N_index[chi_min_index])) + str('+' + '-') + str(N_error[2]/2) +
    xlabel(r'$\nu/\text{Hz}$')
    ylabel(r'Intensity $(\text{erg/cm}^2/\text{s}/\text{Hz}/\text{ster})$')
    xscale("log", nonposx='clip')
    yscale("log", nonposy='clip')
    grid(True, which='both')
    legend(loc='best', prop={'size':8})
    title(str('The $\chi^2$ Minimised Best Fit SED for PACS and SPIRE Bands\nfor $T < 11K$ at the') + str(h) + str('th Pixel\n'))
    savefig(str('imgdump/fine/averages_coarse') + str(h) + str('.png'), dpi=300)
    close()

# Also plot probability contour
figure(2)
#plot(10**N_mesh, T_mesh, 'b.')
plot(N_index[chi_min_index], T_index[chi_min_index], 'r.', label=str(r'$\chi^2$ Minimum=' + str(min(chivals))))
CS = contour(10**N_mesh, T_mesh, chivals_mesh)
clabel(CS)
xlabel(r'$N/(g/cm^{-3})$')
ylabel(r'$T/(K)$')
title(r'$\chi^2$ Contours')
legend(loc='best', prop={'size':8})
savefig(str('imgdump/fine/contours_') + str(h) + str('.png'), dpi=300)
close()

# And the chi-squared landscape
figure(3)
subplot(2,1,1)
plot(T_index, chivals, 'bo')
plot(T_index[chi_min_index], chivals[chi_min_index], 'r.')
xlabel(r'$T/(K)$')
ylabel(r'$\chi^2$')

subplot(2,1,2)
plot(N_index, chivals, 'bo')
plot(N_index[chi_min_index], chivals[chi_min_index], 'r.')
xlabel(r'$N/(g/cm^{-2})$')
ylabel(r'$\chi^2$')

savefig(str('imgdump/fine/landscape_') + str(h) + str('.png'), dpi=300)
close()

chi_fine.close()
'''
```

```

#_Plot_the_data
figure(1)
errorbar(psw[0],psw[1],yerr=psw[2],fmt='co',label='SPIRE:_PSW')
errorbar(pmw[0],pmw[1],yerr=pmw[2],fmt='yo',label='SPIRE:_PMW')
errorbar(plw[0],plw[1],yerr=plw[2],fmt='mo',label='SPIRE:_PLW')
errorbar(blue[0],blue[1],yerr=blue[2],fmt='bo',label='PACS:_Blue')
errorbar(green[0],green[1],yerr=green[2],fmt='go',label='PACS:_Green')
errorbar(red[0],red[1],yerr=red[2],fmt='ro',label='PACS:_Red')
plot(v,chi_min_blackbody,label=str('$\chi^2$')+str('_Minimum:\n$N$') + str(N_index[chi_min_index]) + str('cm^{-2}$')) + str('T$=$') + str(xlabel(r'$\nu$/(Hz)'))
ylabel(r'Intensity_(erg/cm^2/s/Hz/ster)')
xscale("log",_nonposx='clip')
yscale("log",_nonposx='clip')
grid(True,which='both')
legend(loc='best')
title('The $\chi^2$ Minimised Best Fit SED for PACS and SPIRE Bands\n')
savefig('SPIRE_averages_v4.png',dpi=300)
close()

#_Plot_the_figure_and_save_for_reference
figure(2)
plot(N,chivals)
grid(True,which='both')
xlabel('N,(cm^{-2})')
ylabel('$\chi^2$')
title(str('\nThe $\chi^2$ Distribution for $N$') + str(N[0]) + str('cm^{-2}$to') + str(N[-1]) + str('cm^{-2}$'))
savefig('chisquared_N_v3.png',dpi=300)
close()
'''
```

DUST HEATING

Clark, Glover, and Klessen (2012) defines Γ_{ext} due to the ISRF alone to be:

$$\Gamma_{ext} = \chi \Gamma_{ext,0} \quad (8.1)$$

$\Gamma_{ext,0}$ defines an optically thin (i.e. $\tau \ll 1$) heating rate whilst χ is a dimensionless scaling factor that accounts for the attenuation of the ISRF by dust emission. $\Gamma_{ext,0}$ is defined as:

$$\Gamma_{ext,0} = 4\pi R_{dust-gas} \rho \int_0^{\infty} J_\nu \kappa_\nu d\nu \quad (8.2)$$

$dust-gas$ is the dust to gas ratio, whilst ρ is the gas density. J_ν is the mean specific intensity of the ISRF, and κ_ν is the dust opacity. The cooling rate is defined similarly:

$$\Lambda_{dust} = 4\pi R_{dust-gas} \rho \int_0^{\infty} B_\nu \kappa_\nu d\nu \quad (8.3)$$

The only emergent difference between the equations is the integrand: B_ν is the Planck function for a particle at temperature T_d .

Bibliography

- André P., Basu S., and Inutsuka S. (2009). "The formation and evolution of prestellar cores". *Structure Formation in Astrophysics*. Ed. by G. Chabrier. Cambridge University Press, p. 254.
- astrodendro (2016). *Astronomical Dendrograms in Python*. [Online; accessed April 21, 2016]. URL: https://dendrograms.readthedocs.org/en/latest/_images/schematic_structure_1.png.
- Brunt C. M. (2015). "Power-law tails in probability density functions of molecular cloud column density". 449, pp. 4465–4475. doi: 10.1093/mnras/stv471. arXiv: 1503.01306.
- Caltech. *IRAS Image of Rho Ophiuchi*. File: rho.jpg. URL: http://coolcosmos.ipac.caltech.edu/image_galleries/IRAS/roph.html.
- Clark P. C. and Glover S. C. O. (2014). "On column density thresholds and the star formation rate". 444, pp. 2396–2414. doi: 10.1093/mnras/stu1589. arXiv: 1306.5714.
- Clark P. C., Glover S. C. O., and Klessen R. S. (2012). "TreeCol: a novel approach to estimating column densities in astrophysical simulations". 420, pp. 745–756. doi: 10.1111/j.1365-2966.2011.20087.x. arXiv: 1109.3861.
- 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.
- Draine B. T. and Lee H. M. (1984). "Optical properties of interstellar graphite and silicate grains". 285, pp. 89–108. doi: 10.1086/162480.
- Dullemond C. et al. *Manual for RADMC-3D*. English. Version Version 0.39. Universität Heidelberg. 174 pp.
- Dullemond C. et al. *RADMC-3D: A multi-purpose radiative transfer tool*. Version 0.38.
- ESA. *Herschel Fact Sheet*. Fact-sheet. URL: <http://esamultimedia.esa.int/docs/herschel/Herschel-Factsheet.pdf>.
- Ferrière K. M. (2001). "The interstellar environment of our galaxy". *Reviews of Modern Physics* 73, pp. 1031–1066. doi: 10.1103/RevModPhys.73.1031. eprint: astro-ph/0106359.
- Garcia-Lario P. et al. *Herschel Observer's Manual*. English. Version Version 5.0.3. ESA. 124 pp.
- Glover S. C. O. and Clark P. C. (2012). "Is molecular gas necessary for star formation?". 421, pp. 9–19. doi: 10.1111/j.1365-2966.2011.19648.x. arXiv: 1105.3073.
- 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].
- Hayashi C. (1966). "Evolution of Protostars". 4, p. 171. doi: 10.1146/annurev.aa.04.090166.001131.
- 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>.
- Kainulainen J. et al. (2009). "Probing the evolution of molecular cloud structure. From quiescence to birth". 508, pp. L35–L38. doi: 10.1051/0004-6361/200913605. arXiv: 0911.5648.
- Keene J. et al. (1980). "Far-infrared observations of the globule B335". 240, pp. L43–L46. doi: 10.1086/183320.
- Kelly B. C. et al. (2012). "Dust Spectral Energy Distributions in the Era of Herschel and Planck: A Hierarchical Bayesian-fitting Technique". 752, 55, p. 55. doi: 10.1088/0004-637X/752/1/55. arXiv: 1203.0025 [astro-ph.IM].
- 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: 978-3-540-49035-7. 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.
- Pineda J. L. et al. (2010). "The Relation Between Gas and Dust in the Taurus Molecular Cloud". 721, pp. 686–708. doi: 10.1088/0004-637X/721/1/686. arXiv: 1007.5060 [astro-ph.GA].
- 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.
- Press W. H., Flannery B. P., and Teukolsky S. A. (1986). *Numerical recipes. The art of scientific computing*.
- Robitaille T. et al. *astrodendro: Python package for computation of astronomical dendograms*. Version 0.1.0.
- Salpeter E. E. (1955). "The Luminosity Function and Stellar Evolution." 121, p. 161. doi: 10.1086/145971.
- Shetty R. et al. (2009a). "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].
- Shetty R. et al. (2009b). "The Effect of Noise on the Dust Temperature-Spectral Index Correlation". 696, pp. 676–680. doi: 10.1088/0004-637X/696/1/676. arXiv: 0902.0636 [astro-ph.GA].
- Springel V. (2010). "E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh". 401, pp. 791–851. doi: 10.1111/j.1365-2966.2009.15715.x. arXiv: 0901.4107.
- SVO Filter Profile Service. *Filter 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>.
- Taylor J. (1997). *Introduction to Error Analysis, the Study of Uncertainties in Physical Measurements*, 2nd Edition. University Science Books. Chap. Chapter 12: The Chi-Squared Test of a Distribution.