Molecular line transfer calculations in star-forming regions

David Rundle

Submitted by Mr David Rundle to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Physics, July, 2010.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signed: .....
David Rundle

Date: .....

Abstract

This thesis describes the development, benchmarking and application of a non-LTE, co-moving frame Monte Carlo molecular line radiative transfer module for TORUS. Careful attention has been paid to the convergence, acceleration and optimisation of the code.

I present the results of the application of the code to various benchmarking scenarios, including a collapsing cloud, a circumstellar disc and a very optically thick cloud of interstellar water. Benchmarking is an essential step in verifying the accuracy and efficiency of the code which is vital if it is to be used to analyse real data. In all cases, the code was able to accurately reproduce either the expected analytical solution or (in the absence of such a solution) was able to produce results commensurate with the results of other codes.

In order to facilitate the motivating radiative transfer calculations of a star-forming cluster simulated using smoothed particle hydrodynamics (SPH) performed in this thesis, it was first necessary to devise and test an algorithm that efficiently maps an irregular distribution of smoothed particle hydrodynamics (SPH) particles onto a regular adaptive mesh. Whilst the algorithm was designed with this in mind it has also been used to study the effects of radiative feedback in circumstellar discs as well create a synthetic survey of a simulated galaxy.

Bate et al.'s particle representation was resampled onto an adaptive mesh to enable me to use TORUS to obtain non-LTE level populations of multiple molecular species throughout the cluster and create velocity-resolved datacubes by calculating the emergent intensity using raytracing. I compared line profiles of cores traced by N_2H^+(1-0) to probes of low density gas (^{13}CO and $C^{18}O$(1-0)) surrounding the cores along the line-of-sight. The relative differences of the line-centre velocities were found to be small compared to the velocity dispersion, matching recent observations. The conclusion is that one cannot reject competitive accretion as a viable theory of star formation based on observed velocity profiles.

Contents

List of Figures

List of Tables

Declaration

Much of the work described in this thesis has been published in scientific journals. Specifically, significant portions of chapters 2, 3, 4 and 5 have been accepted for publication in the Monthly Notices of the Royal Astronomical Society (MNRAS), arXiv e-print: 1005.1648 (doi:10.1111/j.1365-2966.2010.16982.x).

Chapter 4 also contains the results of work performed by Acreman, Harries and Rundle which is published in MNRAS, 2010, volume 403, pp. 1143 – 1155 as well as work performed using parts of the TORUS radiative transfer code written by me by Acreman, Douglas, Dobbs and Brunt (10.1111/j.1365-2966.2010.16858.x) and Douglas, Acreman, Dobbs and Brunt (doi:10.1111/j.1365-2966.2010.16906.x). The work was facilitated by the SPH to AMR algorithm written by me and described in the same chapter.

Both are available online at the MNRAS Early View website.

Acknowledgements

Well, I did it! Well... I've done something!

Throughout my Ph. D. I have made many mistakes; some enlightening, some embarrassing but all necessary on my path towards today, where I stand on the precipice of becoming *Doctor Dave*. However, I could not and, truthfully, would not have done it if it weren't for the support and love of my family and friends.

It is my greatest wish that my grandfather could have read this work. He always encouraged me with my studies, especially those of maths and science and with each step further I take, I regret more and more that he is not here to witness my success. So I dedicate this thesis to his memory.

My wife, Hannah, with whom I have been together for nine years today, will be pleased that this is finally over 45 months after it began – that I have surely run out of excuses for why I haven't...Why I can't...Why I didn't... She has been a constant source of ~~funding~~ encouragement and motivation and I know that without her I would not be writing this now. I cannot imagine what I *would* be doing instead but then, I wouldn't want to – 'Thanks my love!'

My parents, Jayne & Allan, have been amazing of course. They've supported me through everything – from spelling tests and doing my art homework through to the necessary pecuniary aid required to make it through my undergraduate years!

I'd like to thank the rest of my family, both natural and 'inherited', especially my grandmother, 'Aggie', and my parents- and sisters-in-law who have had to put up with my ups and downs for the past three-and-a-bit years.

Thanks to all of my friends within the Astrophysics department, most of whom have had to endure me for 3+ years and some of whom, mainly by virtue of starting later than me, did *not*, submit before me:

- Ben, my work wife after Suzie left me, for so much - his humour, his rage, his wildly fluctuating weight and (un)correlated moods :) but mainly the science we did talk about occasionally and the greater and greater frequency of nagging to get this thing done towards the end (between crosswords).

- Aude, who suffered our crude, crass office shenanigans with good humour and bright orange earplugs.
- Cameron, our tea-b*tch and all-round good guy who was always able to bring about my gibbering rage within 5 minutes of his arrival.
- Cátia, for whom we (her 'boys') have been a generally positive force in her life; opening her eyes to such things as 'Camping on a hillside, under a tree in a storm' and 'general sub-temperate high-precipitation climates'. She, in return gave us 'Chicken-in-d'oven' and 'Feijoada' for which we are eternally grateful.
- Eric, Nathan, Rob, Farzana, Suzie (you are missed) as well as the fresher faces of the fifth floor (Lee, Rob dR, Eli, Jo, Nawal!) who have known me perhaps for longer than they should have but are hopefully better for it. At least they can put a face to the name that those I leave behind will speak of.

Penultimately, my special thanks go to Dave Acreman, who primarily by virtue of being in the adjacent office became my daily go-to-guy, and who regularly sat quietly whilst I rambled out my half-thought-out problems and then suggested an elegant solution to whatever currently ailed me. His zen-like ways have been an inspiration to me and my next employer will benefit greatly from the more regimented person I will be then (if only I couldn't have been better sooner!)

Finally, I'd like to thank Tim Harries, who took on a young(-ish) failed maths teacher and allowed him to flourish into the confident author who is now penning his great treatise on astrophysics. In all seriousness, I genuinely thank Tim for providing me with the opportunity to become a successful independent researcher, for giving me a fascinating research project and an awesome codebase with which to work, some amazing observing opportunities (!) and some first-class support when it really mattered. Cheers Tim!

My final final thanks in advance go to my internal and external examiners, Drs Jenny Hatchell and Serena Viti respectively for their thorough reading of my thesis and what I can only hope will be their positive comments on the relevance and timeliness of my work. Moreover, I thank the tame admin staff in Northcote who I trust will ease the passage of this tome through the University officialdom to them – Thanks Vicky! Thanks Katrina! I know you're reading :)

I must now turn my attentions to the next stage of my life... whatever that may be.

–David Rundle, *Midsummer*
1st July 2010

On the subject of stars, all investigations which are not ultimately reducible to simple visual observations are...necessarily denied to us... We shall never be able by any means to study their chemical composition.

- Auguste Comte, *Cours de la Philosophie Positive* (1835)

Chapter 1

Introduction

In 1835, the French philosopher Auguste Comte predicted that '...we shall never be able by any means to study [the] chemical composition [of stars]'. As soon as this thought had been articulated it was wrong. More than thirty years earlier, in 1802, William Hyde Wollaston had observed dark lines in the optical spectrum of the Sun, heralding the inception of stellar spectroscopy. However, numerous developments in theoretical and experimental physics would be necessary before it would be possible to fully explain the presence and significance of these lines.

With the celebrated *début-de-siecle* discoveries in the physical sciences came the requisite quantum and atomic theory (e.g. Planck 1901), the Planck function (Planck 1909), stellar atmosphere models (Eddington 1916) and the theory of line formation and broadening (e.g. Lorentz 1905; Bohr 1913). Later on came generalised, rigorous radiative transfer theory of which Chandrasekhar was a leading proponent (Chandrasekhar 1950). In addition to these theoretical prerequisites, corresponding advances in experimental laboratory spectroscopy occurred; like the ability to accurately measure line wavelengths, dipole moments and oscillator strengths (for a deeper historical review, see Hearnshaw 2010).

Thus, by training their telescopes towards the stars and observing the properties of this radiation, astronomers of the mid-twentieth century were able to quantitatively

analyse the spectra of starlight, permitting them to study their temperature, density and chemical composition in direct contradiction of Auguste Comte's asseveration.

The creation of stars is a fundamental process in the universe. Indeed, stars are the only places in the universe where the heavier elements required for life are created. Naturally, understanding this process gives great insight into the formation and evolution of other astrophysical entities from the very large, to the very small. Furthermore, by revealing the secrets of star formation we fill in one more piece of the puzzle of the creation of the universe.

Regions of star formation have been observed since the late eighteenth century (Herschel 1785) but it was not until much later that the dark, opaque, nebulous 'clouds' clustered towards the galactic midplane were known to be such sites (Barnard 1919). The nebulae were shown to consist of dust but it was not known until much later that they also contain gas, indeed far more gas than dust. Although the wider interstellar medium (ISM) was known to contain atomic gas, from the observation of 21-cm radiation from HI (Lilley 1955), no such emission was detected in these 'dark' clouds. It was concluded by Bok et al. (1955) that any gas that existed must be molecular.

Unfortunately, the primary component of these clouds, molecular hydrogen, is virtually undetectable, owing to its homonuclear structure (see Section 1.2). It was not until the early 1970s that any molecular component of the ISM was discovered (Wilson et al. 1970) and these dark clouds suddenly shone when they were observed in emission at a wavelength associated with the transition of a CO molecule from an excited rotational state to its ground state. Today, CO and other molecules, although far less common in the ISM than molecular hydrogen, are used as proxies to inform our understanding of these molecular clouds.

The radiation from these molecules has a characteristic spectrum. They emit and absorb radiation at definite frequencies across the electromagnetic (EM) spectrum, depending on the mode of excitation. Moreover, each species emits or absorbs differently depending on the local conditions; that is, the behaviour of the same molecule in disparate regions of temperature and density can vary strongly – conversely, different species will also behave differently in the same region. It is through the interpretation of this observed radiation that it is possible to gain a deeper understanding of the astrophysical objects one wishes to observe.

In order to interpret the radiation observed it is necessary to understand how the transport of radiation from one location to another is affected by the intervening medium. This is the study of radiative transfer (RT). The mathematics of radiative

transfer have been understood for centuries and countless solutions to simplified problems have been solved exactly, however real astrophysical objects are three-dimensional, have no symmetry and exhibit structure that necessitates a more precise treatment of the radiation field. They contain both gas and dust, which behave very differently, but which are coupled to each other; chemically, kinematically and radiatively. The mere fact that they are dynamic objects, in constant motion, adds a degree of complexity that makes it impossible to obtain an exact solution. Furthermore, the gas and dust are made up of many different species that have their own unique radiative 'fingerprint' and they may overlap throughout the electromagnetic spectrum. Thus it has become necessary to employ numerical models of increasing complexity to accurately model the physical phenomena we can now observe.

Our knowledge of the physical conditions in these molecular clouds and their effect on the molecular gas is critical to our understanding of star formation processes and consequently it is important to have access to powerful RT codes that are able to explain what we can observe. The rest of this introductory chapter is given over to explaining in greater detail the physical conditions necessary to form stars (Section 1.1); the fundamental physical processes governing the creation and destruction of electromagnetic radiation by molecules (Section 1.2); the physics and mathematics of radiative transfer (RT) necessary for quantifying the physical processes that are observed (Section 1.3) and the computational aspects of modern-day radiative transfer codes that are critical to our knowledge of the regions we seek to understand (Section 1.4).

The aims of this thesis and an explanation of each subsequent chapter are set out in Section 1.6 at the end of this chapter.

1.1 The interstellar medium

The 'space' between stars is not empty; closer inspection shows that it is filled with diffuse gas and dust. This mixture is known as the *interstellar medium* (ISM) and provides the environment for all astrophysical events that occur inside our galaxy. It is composed of a heterogeneous mixture of gas and dust of varying thermal and chemical phases. The main components are summarised in Table 1.1.

Hydrogen and helium make up the bulk of the medium with heavier elements, or *metals* as they are referred to in astrophysics, comprising the remaining fraction. Temperatures range from just above the cosmic microwave background (CMB) in

Table 1.1: Phases of the ISM

Phase / Cloud	Gas temperature (K)	n(H_2) (cm^{-3})
Hot Ionised Medium	$\sim 10^6$	$\sim$ few$\times 10^{-3}$
Warm Ionised Medium	$\sim 10^4$	$\sim 10^{-2}$
Warm Neutral Medium	$\sim 10^3$–10^4	~ 0.1
Atomic Cold Neutral Medium	~ 100	$10 - 100$
Molecular Cold Neutral Medium	$\sim 10 - 50$	10^3 – 10^5
Diffuse cloud	100	100 – 300
GMC	10–20	1000
Dark cloud	10	$\sim 500 - 10^3$
Molecular Cloud Cores	$\sim 10 - 30$	$> 10^4$

cold, dark 'cores' of molecular clouds where ionising radiation cannot penetrate, to many thousands of Kelvin in the diffuse, tenuous regions suffused with stellar radiation. Similarly, the medium may be ionised or neutral depending on its exposure to photoionising radiation from a nearby stellar radiation field (UV and higher energy photons), leading to a wealth of chemical combinations.

This thesis focuses primarily on molecular clouds and their constituent cores as it is in these locations where molecules are able to form and subsequently emit their characteristic line radiation.

1.1.1 Molecular clouds

Molecular clouds are regions of greater density of matter embedded within the ISM. The cold, neutral conditions in the bulk of the cloud are predicated on the attenuation of radiation that would otherwise break the intra-molecular bonds. They comprise one of the three distinct phases of the classical 3-phase ISM posited by McKee & Ostriker (1977) (see Table 1.1) and are distinct from the rest of the ISM in that they are not in pressure balance with the other phases.

In order to understand star formation we must observe and understand molecular clouds as they are known to be active sites of star formation (e.g. Reddish 1975; Shu et al. 1987). They are fundamental to the formation of every other structure in the universe and have thus been studied extensively. Although their present-day kinematic and emission profiles are known, owing to their long 'gestation' period (on anthropological timescales) their formation and evolution remains an enigma.

Whether, they are long-lived on a cosmological timescale or whether they are transient features is unclear. Similarly it is not well understood if the molecular gas

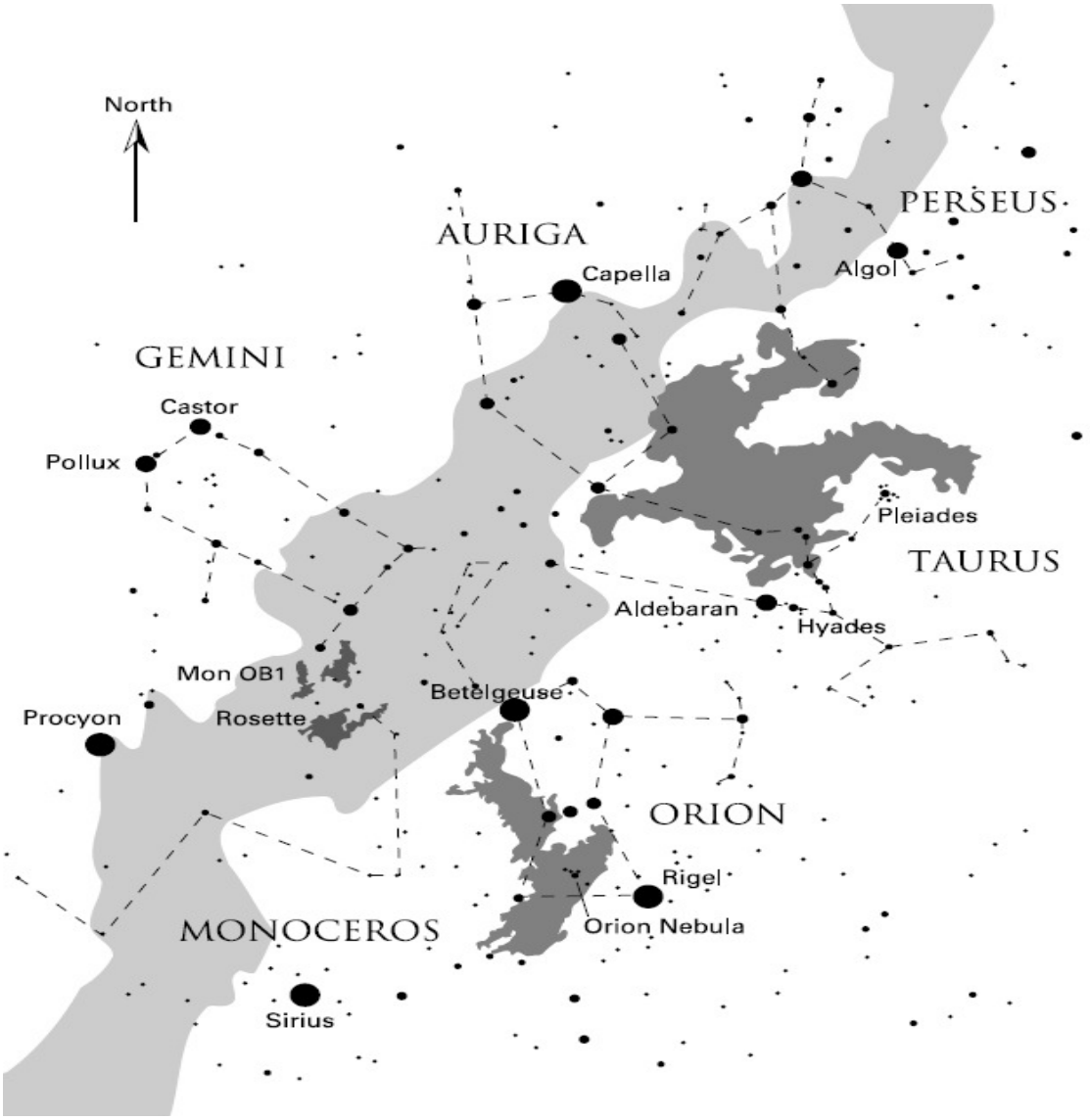


Figure 1.1: Schematic diagram of molecular cloud sites in the galactic plane. The galactic plane is depicted in light grey, while the darker patches indicate local star-forming regions. Constellations and brighter stars are also shown for reference. This figure is reproduced from 'The Formation of Stars' by Stahler & Palla (2005).

is created from an atomic precursor or whether the gas is pre-existent. If they are static features of the galaxy then *how* can they survive so long? Are they perhaps just very inefficient at forming stars (Elmegreen 2000)? Theories for the formation of molecular clouds such as collisional agglomeration of smaller clouds, gravithermal instability and the pressurised accumulation in shocks, either in supernovae or in galactic shocks have been posited (e.g. Elmegreen 1990; Blitz & Williams 1999; Dobbs & Bonnell 2007; Ballesteros-Paredes et al. 2007).

Molecular clouds are most commonly observed in the galactic plane where the concentrations of gas and dust are highest (see Figure 1.1). They exist on a number of

scales; from giant molecular clouds (GMCs) in Orion and Taurus-Auriga that have masses in excess of 10^6 $M_\odot$ and span hundreds of parsecs, to 'Bok globules' that have tens of solar masses of gas within a region often less than a parsec across (Larson 2003).

Molecular hydrogen constitutes most of the gaseous matter in these clouds, although helium represents a significant fraction ($\sim 20\%$ by mass, but is chemically inert) as well as other heavier elements present in trace amounts. These trace elements are responsible for most of the chemistry and nearly all of the molecular line radiation we can observe. Along with molecular gas, dust is the other primary constituent of molecular clouds Goldsmith et al. (1997). Dust grains have sizes ranging from nanometres up to 0.1mm and by the process of elemental depletion are inferred to be variously composed of silicates, carbonaceous compounds, ice and even iron and titanium (Williams 2005), depending on the relative abundances of the elements in their local environment. Whilst dust is vital to interstellar chemistry, its existence has historically frustrated observers until relatively recently because the size of the constituent grains makes them highly effective at absorbing and scattering the light emitted by protostars deeply embedded within the cloud (known as *extinction*). Furthermore, as radiation is absorbed by the dust and re-radiated primarily in the infrared region of the spectrum, the study of molecular clouds using the infra-red and sub-millimetre region of the spectrum is of crucial importance.

Figure 1.2 illustrates the effect of this extinction and reddening highlighting the importance of infra-red and sub-millimetre astronomy. The panels show the increasing transparency of the constituent dust with increasing wavelength. In the visible bands (B and V) and the near-infrared band I (top panels) the cloud absorbs background light and appears as a region of strong absorption. Longer wavelengths (bottom panel) are better able to penetrate the cloud. The top middle panel is a composite image of B, V(not shown) and I and the bottom middle panel is a composite image of B, I and K that fully illustrates the large degree of reddening of the background stars.

As the molecular cloud collapses further in on itself and a protostar begins to form it is observable in the infra-red long before it can be seen visibly. Thus, the very thing that thwarted the astronomers of antiquity's observations of the stars is the same thing that allows us to better understand their formation process today. In fact, modern infra-red and sub-millimetre sky surveys like 2MASS (Skrutskie et al. 2006) and the JCMT legacy survey (Ward-Thompson et al. 2007b) clearly show that the regions of sky that have a marked extinction of visible light are in fact active regions of star formation.

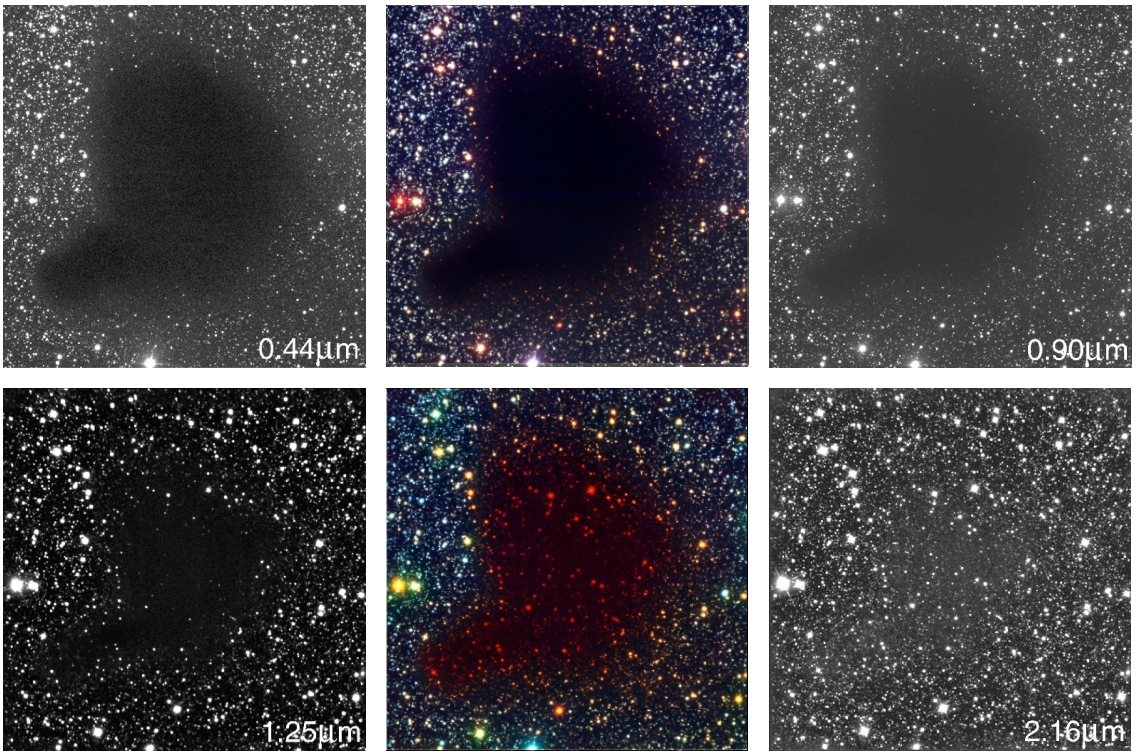


Figure 1.2: Barnard 68 imaged in 6 different wavelengths (4 shown) from 0.44 – 2 micron. Adapted from Alves et al. (2001). From top-left to bottom-right panels, the filters used are: B, BVI composite, I, J, BIK composite, K.

1.1.2 Dense cores

Molecular clouds contain a large proportion of the molecular mass of our own galaxy. As previously stated, they can fragment into smaller regions or 'clumps' which then go on to collapse independently of each other giving rise to star-forming 'cores'. The observation that the structure of molecular clouds is hierarchical gave rise to a number of theories suggesting their fractal nature (Scalo 1990). Moreover, the mass distribution of these clumps follows a form similar to that of the stellar initial mass function (IMF) (e.g. Kroupa 2002; Chabrier 2003) warranting further study in their own right. The expected mass of these cores was first analysed by Jeans (1902). Assuming an infinite, isotropic medium it can be shown that the maximum mass that is stable against gravitational collapse is,

$$M_J = \left(\frac{4\pi\rho}{3}\right)^{-1/2} \left(\frac{5kT}{G\mu m_{\rm H}}\right)^{3/2} \tag{1.1}$$

where ρ is the mass density of the cloud, μ is the mean molecular weight and $m_{\rm H}$ is the mass of atomic hydrogen. While this is a useful first approximation, molecular cloud cores have been observed with a much greater mass than this, suggesting that

the core is supported against collapse by more than thermal pressure. Magnetic fields and turbulence are the most probable candidates for this (e.g. Ballesteros-Paredes et al. 2007; Krumholz et al. 2007).

The locations of these cores within the clouds that spawn them can be detected as areas of increased intensity of emission in the sub-millimetre (and longer wavelength) region of the electromagnetic spectrum (e.g. Myers et al. 1983; Myers & Benson 1983). By using the relative strengths of molecular line emission in these regions it was possible to infer that the density had to be at least 10^4 H_2 cm^{-3}.

Observationally, if one of these enhancements coincides with an infra-red source (indicative of the presence of a protostar) then the core is said to be *protostellar*, otherwise it is *starless*. Later, Ward-Thompson et al. (1994) showed that a further distinction can be made in starless cores if evidence of further density enhancement is seen; these are known as *prestellar* cores.

Dramatic increases in temperature and density that result from gravitational collapse effect changes in not only the physical composition of the cores but also, crucially, their chemical composition. Whereas carbon– and oxygen-bearing molecules (e.g. CO, CS, H_2O etc.) are found throughout the molecular cloud, they appear to be depleted in the inner regions of these dense cores. There is evidence that the molecules 'freeze out' (i.e. they are adsorbed) onto the mantles of the dust grains where the gas density is high and the temperature is low (e.g. Caselli et al. 1999; Tafalla et al. 2004; Walmsley et al. 2004). Later in the evolution of a dense core (should it continue to condense and form a protostar) the core warms up to $\sim$ 100 K and the frozen out molecules sublimate back in to the gas phase, further changing the relative abundances of the chemical species present in the core. Figure 1.3 shows a schematic diagram depicting the relative locations of different chemical species contained within a starless dense cloud core.

Today's (sub-)millimetre observatories facilitate the observation of these cool, chemically rich regions of the galaxy. Observations with modern sub-millimetre instruments like the earthbound James Clerk Maxwell Telescope (JCMT), the Caltech Submillimeter Observatory (CSO) and the Submillimeter Array (SMA) have given high spatial resolution observations of these progenitors of young stellar objects. These terrestrial instruments have recently been joined by a space-based counterpart, Herschel. It is hoped that Herschel will '...provide new insights on interstellar chemical processes [and] their role in the chemodynamics of star and planet formation...' (Harwit 2004) through the greater spectral resolving power of the HIFI spectrometer. Herschel also carries SPIRE and PACS, two highly sensitive imaging cameras and low-resolution spectrometers providing spectral coverage from the

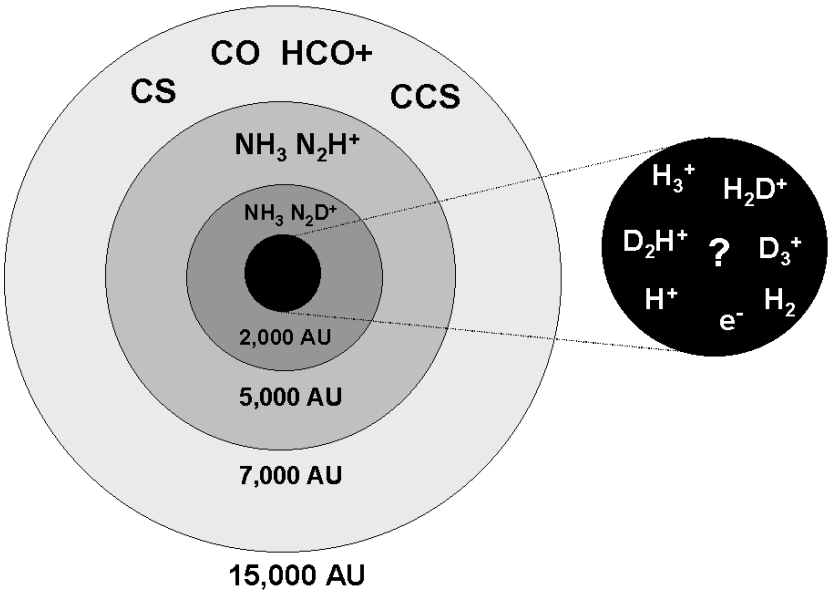


Figure 1.3: A schematic diagram showing the relative locations of different chemical species contained within a starless dense cloud core. Reproduced from di Francesco et al. (2007).

far-infrared into the sub-millimetre (55 – 672 microns). This is a vital region of the spectrum for determining key astrophysical parameters for astrophysical objects such as dust temperature and density (and hence mass) amongst many other things. Not until the Atacama Large Millimeter Array (ALMA) sees first light some time in 2012 will our resolving power be bettered, bringing with it the promise of unprecedented resolution of these objects.

However, improved observations can only advance our understanding so far, therefore it is vital that the interpretation of the data is sound, necessitating the use of non-symmetric models incorporating complex physics. Each new generation of equipment must be matched by appropriate numerical modelling and indeed it has been thus far. By Moore's law, transistors continue to halve in size every 18 months and the available processing power doubles. Thus in the last ten years processing power has increased by two orders of magnitude and no doubt will do so again in the next ten years. This enhanced power will, by permitting the inclusion of more detailed physics, unlock regimes that were previously inaccessible, reducing the necessity of approximations and facilitating broader parameter searches.

1.1.3 Star formation

As an individual, self-contained process, the act of star formation is qualitatively well understood (e.g. Shu 1977; Shu et al. 1987). An initially tenuous cloud of cold

molecular gas with small perturbations in its density will begin to fragment into smaller 'clumps' as the gas and dust collapses upon itself. Gravitational potential energy is converted into thermal energy which increases the frequency of the collisions between gas molecules whereby they become excited. The molecules cool by emitting radiation which initially escapes from the cloud. As the cloud continues to contract, the density continues to rise in the core of the clump and radiation can no longer be radiated away as it is reabsorbed and continuously scattered over very short lengthscales; the cloud is said to have become *optically thick*. The temperature rises dramatically and a protostar is born, however it is still embedded deep within its natal envelope; this is known as a 'class 0' young stellar object (YSO) using the morphological classification scheme based on spectral energy distributions (SEDs) of Lada & Wilking (1984) (later extended by Andre et al. 1993).

The protostar continues to evolve by accretion, collecting material that has accumulated onto a disk from the protostellar envelope (class I). Classes I and II are distinguished by their bolometric temperature and spectral slope, α, between 2 and 25 μm (Lada 1987). Class I stars have a positive gradient, or 'infrared excess' owing to the strong extinction of visible light by the dusty envelope surrounding the protostar. Material from the envelope is accreted on to the protostar and accretion disc until the envelope disappears entirely and all that remains is the disc. In this phase the star will gain most of its mass and will continue to do so until the accretion disc is dissipated. These objects are known as class II objects and typically have a negative value for α unless observed edge-on, when the optically thick dust in the disc can affect the SED strongly. As the temperature of the protostar continues to rise, hydrogen burning commences and the newly-formed star becomes a main-sequence star.

This picture is complicated as stars rarely form in isolation. One mechanism that exists to explain this phenomenon is the competitive accretion model where protostars are theorised to form in dense clusters competing to accrete material from their surroundings (e.g. Bonnell et al. 2001a). The most 'successful' will go on to become the most massive, most luminous objects and may even eject their smaller competitors from the cloud entirely (Bate et al. 2003a), truncating their accretion and prematurely consigning them to existence as a brown dwarf, unable to attain a great enough mass to start hydrogen burning. A more recent study by Price & Bate (2007) studied the effect of magnetic fields on star formation. They showed that in their simulations the extra pressure support provided by magnetic fields inhibits disc and binary formation and crucially, showed that the star formation rate is reduced.

A more detailed discussion on clustered star-formation can be found in Chapter 5.

1.2 Interstellar molecules

Molecular astrophysics is the study of line emission and absorption of radiation from molecules in space. Molecular lines are excellent probes of the key parameters of molecular clouds and cloud cores and can provide more information than dust emission alone. The information encoded in the radiation received from any given molecular line pertains not only to the physical conditions (temperature, density, kinematics etc.) but also the chemistry of the object along the line-of-sight of the observation. Yet, despite this, molecular line data must be interpreted with great care for precisely the same reason as it is so powerful. Because so much information is encoded in a very narrow range of frequencies and because the observed intensity is potentially a function of temperature, density, velocity, chemistry and optical depth, it is necessary to make many more assumptions about the state of the core than with dust observations. In addition, because the intensity can be sensitive to any one of these parameters, strong constraints are required to ensure the validity of the conclusions obtained using line data. Fortunately, multiple lines can be used, in addition to continuum data to provide these constraints.

As discussed earlier, molecular hydrogen is the primary component of molecular clouds. The possible rotational (and vibrational) states for molecular hydrogen have been known for sixty years (Herzberg 1950), therefore they should be excellent candidates for observation in space. However, the electrons are tightly coupled and the electron spins cancel completely so no electric dipole is present to absorb or scatter radiation, making it very difficult to detect despite its abundance. The spontaneous emission of light from the $J = 1 - 0$ state of molecular hydrogen is 'forbidden' (in the quantum sense), thus only very low-probability electric quadrupole transitions may occur.

The most commonly observed molecular transition in molecular clouds is the CO $J = 1 - 0$ line where the molecule falls from its first excited state back into the ground state. The characteristic wavelength for this transition is 2.6 mm and is readily observed owing to its high relative abundance (typically 1 part in 10,000 by number density relative to H_2) and its dipole moment, facilitating absorption and emission of EM radiation. In general, one of the most useful observational diagnostic tools available in the pursuit of knowledge about star-formation is the analysis of radiation emitted by the transition of one rotational state of a molecule to another. The applications of molecular line radiation as probes of astrophysical parameters are discussed in Section 1.5.

However, CO is only one of over 100 molecules known to be present in the interstellar medium and in circumstellar environments, each one with a number of observable

transitions. The transitions of many of the most abundant molecules have been catalogued and put into various online databases such as the JPL (Jet Propulsion Laboratory) catalogue (Pickett et al. 1998) and CDMS (Cologne Database for Molecular Spectroscopy) (Müller et al. 2005).

In order to quantify the radiation received from these gaseous objects it is necessary to understand how the molecules initially formed and the mechanism by which the radiation is produced. The former question is one of chemistry addressed below; the latter is one of molecular microphysics which is addressed in Section 1.2.2.

1.2.1 Chemistry

Astrochemistry is the study of interactions between atomic and molecular species; typically at very low densities and/or temperatures compared to a terrestrial environment. The field has its roots in observations of CH and CN in absorption in diffuse clouds (Swings & Rosenfeld 1937) and subsequent observations of water, ammonia and hydroxyl radicals in the 1960s and 1970s (e.g. Dieter 1964). As previously discussed, chemistry is a powerful diagnostic, both of the current and the past physical conditions of the forming protostar. Owing to the wide variation in the physical parameters in the field, it is very complex. Temperatures, densities, radiation fields, timescales, spatial scales, molecular abundances and many more variables all vary over many orders of magnitude. Although, similar structures present broadly similar environments, their subsequent evolution (both chemical and stellar) can be very different from cloud to cloud depending on mass contained within (e.g. the formation of stars in Orion vs Perseus, see for example, Johnstone & Bally (1999); Hatchell et al. (2005)).

While the overall density of matter throughout the galaxy is very low, there are some relatively dense regions which are extremely active sites of star formation. The basic building blocks of stars contained within these dense molecular clouds are molecular hydrogen (H_2) and helium (He), but these regions also contain over 100 other detected molecular species; carbon monoxide (CO), ammonia (NH_3) and water (H_2O) being some of most readily identified. Hydrogen and helium are the most abundant in the ISM, making up 98% by number of protons. The remaining fraction of heavier elements (or *metals*) are created by nucleosynthesis. After hydrogen and helium, the next most abundant elements are carbon and oxygen, both more than one-thousand times less prevalent than hydrogen (Stahler & Palla 2005), so it comes as no surprise that the next most abundant molecule is $^{12}C^{16}O$ with a relative abundance of $\sim 10^{-4}$ compared with that of H_2.

Table 1.2: Molecular formation and destruction mechanisms. Reproduced from Hogerheijde (2005)

Radiative Association	$X + Y \leftrightarrow XY* \leftrightarrow XY + \nu$
Photodissociation	$XY + \nu \rightarrow X + Y$
Associative detachment	$\begin{cases} X + e \rightarrow X^- + \nu \\ X^- + Y \rightarrow XY + e \end{cases}$
Dissociative recombination	$X^+ + e \rightarrow X + \nu$
	$XY^+ + e \rightarrow \begin{cases} XY + \nu \\ X + Y \end{cases}$
Gas-phase ion-neutral reaction	$X^+ + YZ \rightarrow XY^+ + Z$
neutral-neutral reaction	$X + YZ \rightarrow XY + Z$
Charge transfer	$X^+ + YZ \rightarrow X + YZ^+$
Grain-surface reaction	$X + G : Y \Rightarrow X : G : Y \Rightarrow G : XY \Rightarrow G + XY$

Immersed in a homogeneous radiation field such as that in the vicinity of a star, isolated molecular lifetimes can be as short as 10^2 – 10^3 years (Draine 1978) before photodissociation occurs, however deep inside molecular clouds with their associated large column densities which block this ambient radiation field, the only mechanism that is capable of effectively ionising molecules is collision with cosmic rays. Thus, excluding their ionised envelopes (remnants of condensed atomic clouds or photodissociated gas from the cloud, Blitz & Williams 1999), molecular clouds, opaque to radiation, are denser ($n_{H_2} \sim 10^4$ cm^{-3}) and cooler than their surroundings.

On Earth, the high ambient temperature and gas density can often provide sufficient activation energy and opportunity for chemical reactions to occur but in these tenuous environments, mechanisms other than two–(or more) body radiative association must be utilised. Table 1.2 demonstrates the main pathways by which chemistry can occur. On-line databases such as the UMIST06 database for Astrochemistry (Woodall et al. 2007) list the known rates of these reactions between the 420 species in their network. The different chemical processes are inter-dependent and it is possible to create (or destroy) more and more complex molecules. Moreover, it is clear that dust plays a critical role as a catalyst in gas chemistry as well as being used as a diagnostic tool (e.g. Andre et al. 1996).

Today, time-dependent models (e.g. Bergin & Langer 1997; Viti et al. 2000; Le Petit et al. 2006) are used to understand the inter-dependence of chemical abundances in astrophysically interesting regions. These models simulate large reaction networks which are evolved with realistic initial conditions and approach a steady-state if the simulation timestep is shorter than the chemical timescale. It is then possible to

determine abundances of some molecular species which agree well with observations (e.g. Bergin et al. 2002) when the column densities of the model are compared with observations. However, many uncertainties in these models persist and often the abundances provided by chemical models can differ from observations by orders of magnitude. By incorporating a fuller treatment of radiative transfer into the chemical networks (e.g. treating photodesorption/photodissociation of some species that are sensitive to Lyman α radiation), matched to improved constraints provided by future observations, we can increase our cognisance of the chemical evolution of many important tracer elements in these chemically active regions.

1.2.2 Molecular microphysics

Energy can be stored in molecules in four ways: by increasing their kinetic energy through heating and collisions; by exciting electronic transitions; by exciting vibrational modes of the molecular bonds and by exciting rotational modes in the bonds. The energy required to promote molecules to these excited states ranges from the UV in electronic excitation to the sub-millimetre in the rotational case. Typically, infrared radiation stimulates vibrational states. Figure 1.4 illustrates the range of transitions that molecules may undergo. In this section I will focus primarily on rotational modes, as these are the modes that observers of cool molecular clouds are most interested in, although (ro-)vibrational transitions are important in hotter environments like protoplanetary discs and shock fronts. The concepts discussed in this chapter are equally valid for both rotational and vibrational lines but the associated energy levels are orders of magnitude different owing to the different mechanism of excitation.

The positive and negative charges do not completely overlap in the covalent bonds that hold together most molecules. Polar molecules possess a permanent dipole moment, a time-averaged imbalance of charge relative to the centre of mass. Molecules with reflection symmetry like carbon dioxide and molecular hydrogen have no permanent dipole moments and thus do not readily radiate by electric dipole radiation. They may radiate by electric quadrupole radiation but this is a very slow, low-probability process compared to electric dipole transitions. It is possible, however, to induce a dipole moment by applying an external electric field or by inducing electronic excitations which lead to asymmetric charge distributions creating a net dipole moment inside the molecule. Electromagnetic waves can then excite the molecules into higher rotational levels by exerting a torque on the bond about a rotational axis. Energy is stored in this state analogously to a classical rigid rotor,

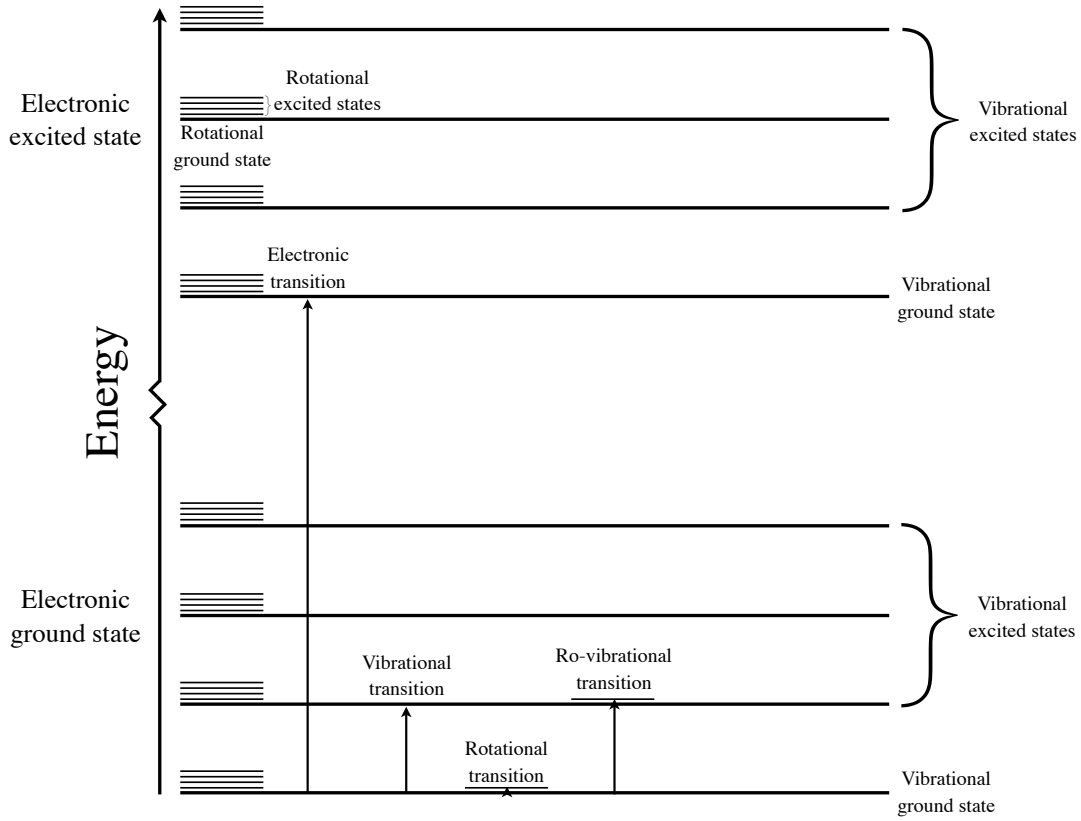


Figure 1.4: A schematic molecular energy level diagram. Electronic, vibrational, rotational transitions and ro-vibrational transitions (a combination of both rotational and vibrational transitions) are shown.

however quantum mechanics dictates that the rotational energy stored in a molecule must be quantised.

The frequency required to promote a molecule into an excited state is typically on the order of $100-1000$ GHz, the corresponding wavelengths being $3-0.3$ mm, which is in the microwave region of the spectrum. This gives rise to the moniker millimetre astronomy that is used to probe star-forming regions at these frequencies.

Diatomic molecules are the simplest possible molecules; they must have a linear geometry and may be either homonuclear (i.e. consist of two identical atoms, e.g. H_2, O_2) or heteronuclear (e.g. CO or CS). Owing to their cylindrical symmetry, one of the moments of inertia is negligible, reducing the degrees of freedom to one. Consequently, they exhibit the simplest spectra. The separation between energy levels is almost directly proportional to the bond length of the molecule because a slight reduction is observed as the centrifugal force of the rotating molecule acts to pull the atoms apart. The measurement and identification of one spectral line permits the direct calculation of the moment of inertia and the bond length. It is possible to ascertain the same information for linear molecules with three or more

atoms (e.g. HCN, HNC, HCO^+) with little additional work.

Only certain energies are permitted as the solutions to the Schrödinger equation for a rigid rotor:

$$E_{\text{rot}} = \frac{\hbar^2}{2I} J(J+1), \tag{1.2}$$

where I is the classical moment of inertia and where J is the dimensionless *rotational quantum number*. This quantum number can be used to quantify the change from one energy level to another (a *transition*) and the notation $J = i - j$ describes the energy levels that the molecule has transitioned from, i, and to, j. The *rotational constant*, B, is measurable using spectroscopy and is defined as

$$B = \frac{h}{8\pi^2 I} \tag{1.3}$$

where $I = \mu R^2$, the classical moment of inertia of the molecule, μ is the reduced mass of the molecule $\mu = \left(\frac{1}{M_{\text{A}}} + \frac{1}{M_{\text{B}}}\right)^{-1}$ and R is the bond length.

For a diatomic molecule, the selection rules for electric dipole rotational transitions are $\Delta J = \pm 1$ so from equation 1.2, the energy liberated from the de-excitation of a molecule from J to $J-1$ is:

$$E_{J \to J-1} = 2BhJ. \tag{1.4}$$

The allowed transitions for the diatomic molecule are regularly spaced at intervals of $2Bh$ for increasing J creating a 'ladder' with equally spaced rungs. Molecules with large moments of inertia will have a small rotational constant and their levels will be less widely separated. The moment of inertia for a CO molecule, $I_{\text{CO}} = 1.46{\times}10^{-39}$ is large, so the levels are close together making it a good tracer of temperature. In comparison, $I_{\text{H}_2} = 4.7{\times}10^{-41}$ is extremely small and thus the levels are spaced much further apart. Owing to the large amount of energy required to excite H_2 and its ubiquity in molecular clouds it is an excellent tracer of shocks; any emission at all is indicative that the temperature is at least 500 K below which the gas is usually undetectable.

The rate at which a molecule radiates is characterised by the Einstein coefficient, A_{ul} and is unique to each transition. These rates are described in this section.

Intuitively, more complex molecules have more complex spectra but can typically be grouped into two distinct groups: symmetric rotors, where some symmetry may be exploited to ascertain the physical parameters of the molecule and two moments of inertia are the same, as is the case with NH_3 and asymmetric rotors (e.g. H_2O and NO_2), where the moments of inertia along all three axes are different and no symmetry can be exploited. These non-linear molecules have more complex spectra that exhibit many features.

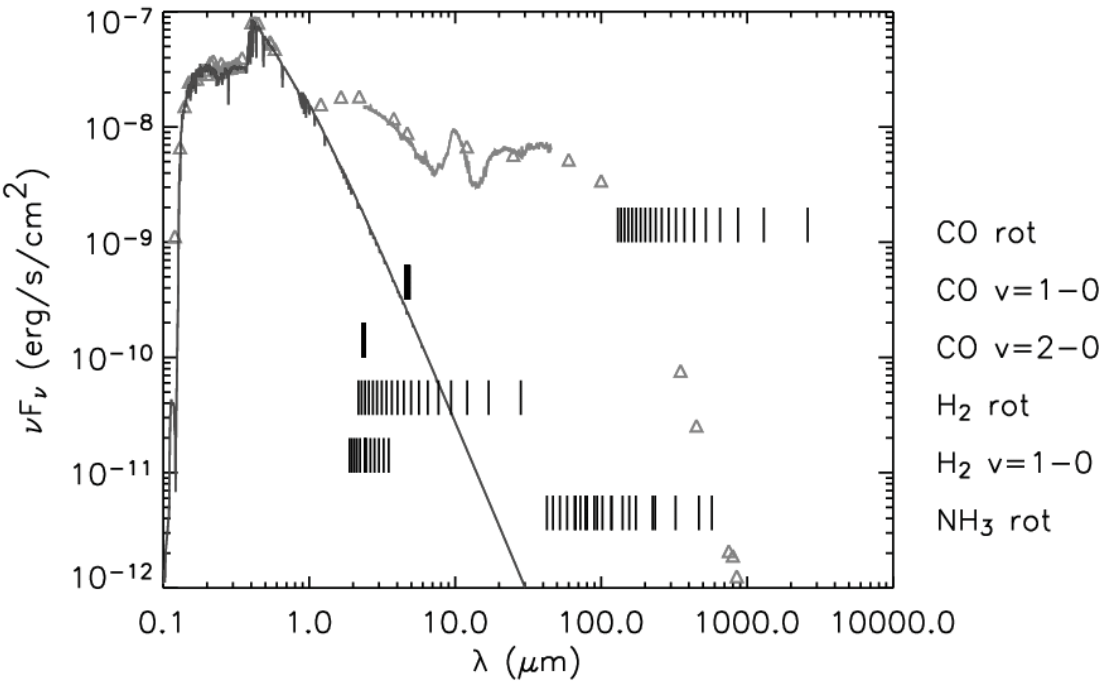


Figure 1.5: Location in the electromagnetic spectrum of some common molecular transitions, superimposed on an SED of a dusty envelope with extensive flux emission at long wavelengths (light grey) and a typical stellar SED. Molecular rotational transitions can be used to probe regions of the EM spectrum where the continuum emission is very low.

Figure 1.5 illustrates the location in the electromagnetic spectrum where these lines are present. Coverage of the electromagnetic spectrum from ultraviolet far into the radio by spectral lines is extensive and where lines from different species do not overlap they provide an excellent diagnostic tool that is sensitive to many intrinsic properties of a YSO.

Einstein co-efficients and radiative rates

The Einstein coefficients are fixed probabilities associated with each atom or molecule and do not depend on the state of the gas of which the atoms are a part. The subsequent derived relationships assume thermal equilibrium will be valid universally. When all levels are in equilibrium, the net change between any two levels will be balanced, because the probabilities of transition are not affected by other excited atoms/molecules.

Spontaneous emission is the radiative process by which a state 'spontaneously' (i.e. without any outside influence) decays from a higher energy level, u, to a lower one,

l. The Einstein coefficient, A_{ul}, quantifies the probability per unit time that level u will decay spontaneously to level l, emitting a photon with an energy $E = h\nu_{ul}$. The rate of increase in the number density of atoms in level l per unit time due to spontaneous emission will be:

$$\frac{dn_l}{dt}_{A_{ul}} = A_{ul} n_u, \tag{1.5}$$

where n_u is the number density of atoms in level u.

The classical value for the oscillator strength for a damped harmonic oscillator is:

$$\gamma^{\mathrm{rad}} = \frac{\mu^2 \nu^3}{3\pi\epsilon_0 \hbar c^3}, \tag{1.6}$$

which, by applying the dipole approximation, can be shown to be equivalent to the Einstein A-coefficient.

Similarly, stimulated emission is the process by which an excitation is induced by the presence of electromagnetic radiation at (or near) the frequency of the transition. As above, the Einstein coefficient B_{ul} gives the probability per unit time per unit spectral radiance of the radiation field that an excited molecule in state u will decay to state l. As above, the change in the number density of atoms in l per unit time due to induced emission will be:

$$\frac{dn_l}{dt}_{B_{ul}} = B_{ul} n_u \rho(\nu), \tag{1.7}$$

where

$$\rho(\nu) = \frac{2h\nu^3}{c^2(e^{h\nu/kT} - 1)} \tag{1.8}$$

and $\rho(\nu)$ is the radiation density of the radiation field at the frequency of the transition.

Stimulated absorption can be viewed as the inverse process of stimulated emission and is governed by the coefficient B_{lu} and

$$\frac{dn_l}{dt}_{B_{lu}} = -B_{lu} n_l \rho(\nu). \tag{1.9}$$

By invoking statistical equilibrium, it can be shown that the other Einstein coefficients are not uniquely defined. The pair of stimulated absorption and emission

coefficients can be determined from the spontaneous emission coefficient,

$$B_{ul} = \frac{c^2}{2h\nu^3} A_{ul} \tag{1.10}$$

$$B_{lu} = \frac{g_u}{g_l} B_{ul}. \tag{1.11}$$

g_u and g_l are the statistical weights of the states – the number of ways a particular state can be populated. These values typically increase with increasing quantum number. In the case of linear molecules, $g = 2J + 1$.

These intrinsic rates govern the macroscopic radiation field that is produced by the cloud through the coupling of the global radiation field and the local level populations.

Molecular Hyperfine Structure

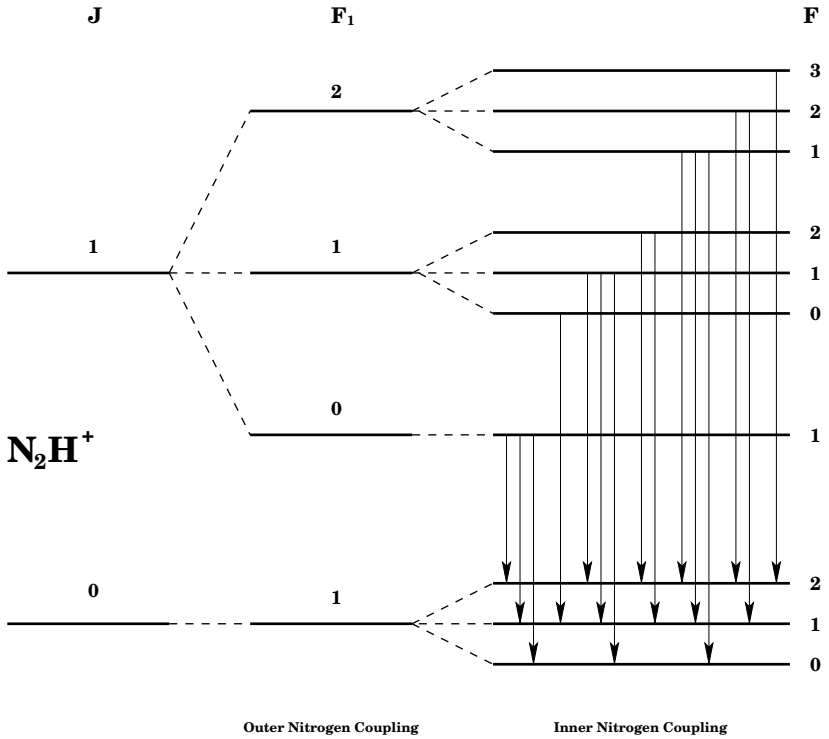
In addition to the normal structure that is observed in rotational line spectra, further interactions between electrons and/or nucleons cause *hyperfine* structure to be observed in the spectra. The shifts in energy are an order of magnitude smaller than the rotational energies they are superimposed upon, but can have significant effects on astrophysical observations (e.g. Daniel et al. 2006) by providing alternative pathways for a molecule to de-excite rather than saturating a line that would exist in the absence of hyperfine splitting.

In the case of N_2H^+, hyperfine splitting is observed because the quadrupole moments of the two nitrogen atoms and the molecular spin are coupled (Shirley et al. 2005). Figures 1.6(a) and 1.6(b) illustrate that a single transition splits into 7 distinct although potentially overlapping levels each with a distinct strength (although 15 exist mathematically).

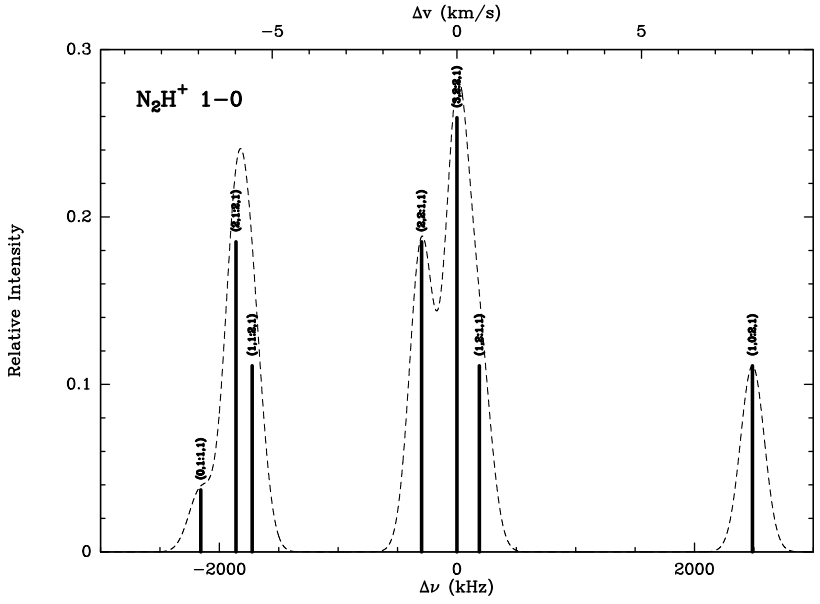
This additional splitting means that more information about the temperature and density structure of the core can be encoded in an observation (by examining the ratios of the hyperfine levels, e.g. Tatematsu et al. 2004) however the increased complexity in the spectrum requires special analysis of observations and additional physics in molecular line transfer codes.

1.3 Radiative transfer

The field of radiative transfer is the bridge between the microscopic – interactions of photons with matter – and the macroscopic – the flux of energy observed from the



(a) Schematic energy level diagram of the first two rotational levels of N_2H^+ including hyperfine splitting.



(b) Synthetic spectra for N_2H^+ $J = 1 - 0$ assuming a 100kHz gaussian linewidth (dashed). Reproduced from 'Molecular Column Density Calculation' (Mangum & Shirley 2008)

Figure 1.6: Schematic energy level diagram of the first two rotational levels of N_2H^+ and the relative line strengths of the observed transitions.

matter. Our understanding of all astrophysical objects (e.g. molecular clouds, YSOs, disks, photon-dominated regions (PDRs) etc.) necessarily requires both observation and detailed theoretical modelling. It is impossible to physically acquire the intrinsic (intensive) parameters (temperature, density, molecular abundance or (turbulent) velocity and magnetic fields for example) of these objects by direct manipulation as it would be possible to in a laboratory. Thus, the ability to examine conditions passively, using only the emitted radiation is a key diagnostic tool, not only in astrophysics, where the benefits are manifest, but in other fields too, such as plasma physics and medicine.

It has truly existed as a field of study in its own right since 1950 when Subrahmanyan Chandrasekhar published his treatise on 'Radiative Transfer' (Chandrasekhar 1950) focussing strongly on plane-parallel stellar and planetary atmospheres which rapidly became the seminal work in the field. Much subsequent work has been accomplished in the field which builds on the foundations of Chandrasekhar's work (e.g. Rybicki & Lightman 1979).

By making some simple assumptions like the ability of radiation to freely escape the medium in the case of large velocity gradients (e.g. Sobolev 1960) (see Section 1.4.1) and assuming that stimulated emission is totally redistributed in space and frequency it has been possible to derive analytical solutions to situations that can approximate with some degree of accuracy what is observed. Naturally, representing real media with its complex interplay of physical effects is beyond a purely analytical approach and requires computer modelling, but it is the same basic equations that must be solved in order to determine the flux of energy emitted from these objects.

Throughout this section, I shall assume time-independent radiative equilibrium. If one is considering the kinematical effects of radiation on its environment then one may have to consider the fact that radiation travels at a finite speed, which, in the case of optically thick media will effectively be considerably less than the speed of light *in vacuo*. Radiative equilibrium is a dynamic equilibrium and time-dependence breaks this assumption as it is no longer necessary for every point in space to instantaneously satisfy this criterion.

In common with the most popular texts on radiative transfer, the derivation of the equations in this section use the CGS unit system.

1.3.1 The equations of radiative and statistical equilibrium

The propagation of radiation through a medium is affected by emission, absorption and scattering. The radiation field can be defined by its *specific intensity*, I_ν, defined

as the energy emitted per unit time with frequency between ν and $\nu + d\nu$ that propagates normal to the area $d\mathcal{A}$ and within the solid angle $d\Omega$,

$$dE = I_\nu d\mathcal{A} d\Omega d\nu dt. \tag{1.12}$$

The radiative transfer equation mathematically describes the rate of change of I_ν, along a path, $\mathbf{s}$, as the energy is added to or subtracted from its path. In the absence of scattering, as this does not apply to molecular line radiation, this becomes:

$$\frac{dI_\nu}{ds} = -\alpha_\nu I_\nu + j_\nu, \tag{1.13}$$

where α_ν and j_ν are the local absorption and spontaneous emission coefficients respectively.

Defining the frequency specific *optical depth*, τ_ν, as

$$\tau_\nu = \int \alpha_\nu ds, \tag{1.14}$$

equation 1.13 becomes

$$\frac{dI_\nu}{d\tau_\nu} = -I_\nu + S_\nu, \tag{1.15}$$

where S_ν is the local source function, the ratio of the emission to the absorption:

$$S_\nu = \frac{j_\nu}{\alpha_\nu}. \tag{1.16}$$

From the definition of optical depth come two more terms, *optically thin* and *optically thick*. An optically thin medium does not strongly attenuate incident radiation and emission from 'behind' the object can still be observed. An optically thick medium does attenuate incident radiation and it is not possible to obtain information about material emitting from behind an object where $\tau \gtrsim 1$; the distinction between thick and thin is consequently made where $\tau = 1$, recalling that this point may vary with frequency.

Equation 1.15 is the differential version of the radiative transfer equation along a path. The more computationally useful integral of this equation along the path from $\tau = 0$ to $\tau = \tau'$ is obtained by formally solving equation 1.15:

$$I_\nu(\tau) = I_\nu(0)e^{-\tau_\nu} + \int_0^\tau S_\nu(\tau'_\nu)e^{\tau'_\nu - \tau_\nu} d\tau'_\nu, \tag{1.17}$$

where $I_\nu(0)$, is the background radiation entering the medium.

The above equations are valid for both line radiation and broadband continuum.

Indeed it is possible and often necessary to take into account radiative mechanisms for both gas and dust simultaneously, depending on the region of the EM spectrum under consideration.

Dust absorbs many frequencies of radiation with different efficiencies. Neglecting scattering, the opacity due to dust is, at its simplest, a function of frequency and temperature, $\kappa_\nu(T_{\rm dust})$ but it can also be a function of space as well, where larger dust grains sediment into the midplane of a disc (e.g. Dubrulle et al. 1995; Dullemond et al. 2007). The units of dust opacity are g cm^{-2}, which when scaled by the dust density returns the total absorption per cm for the particular dust model being used,

$$\alpha_\nu^{\rm dust} = \kappa_\nu \rho_{\rm dust}. \tag{1.18}$$

The accepted canonical mass ratio of gas-to-dust is 100 (e.g. Mathis 1993). The emission by the dust is then given by assuming all absorbed radiation is re-radiated as a blackbody at the dust temperature, $T_{\rm dust}$:

$$j_\nu^{\rm dust} = B_\nu(T_{\rm dust})\alpha_\nu^{\rm dust}, \tag{1.19}$$

where

$$B_\nu(T_{\rm dust}) = \frac{2h\nu^3/c^2}{e^{h\nu/kT_{\rm dust}} - 1}. \tag{1.20}$$

is the Planck function expressed in terms of frequency.

The effect of frequency on the dust absorption depends on both the chemical composition of the dust and the distribution of grain sizes in the object (Draine & Lee 1984; Ossenkopf & Henning 1994). The functional form of grain size distribution is usually assumed to follow a power-law (Mathis et al. 1977), $f(a) \propto a^{-q}$, where $q = 3.5$.

For the specific molecular line transfer problem, the emission and absorption coefficients can be determined by the Einstein transition rates between rotational/vibrational levels of the molecule and the relative populations of these levels. However, the treatment of absorption and emission due to molecular gas is more complex because systematic velocity fields make it anisotropic:

$$\alpha_{\nu_{ul}}^{\rm gas} = \frac{h\nu_0}{4\pi}\left(n_l B_{lu} - n_u B_{ul}\right)\phi_\nu \tag{1.21}$$

$$j_{\nu_{\rm ul}}^{\rm gas} = \frac{h\nu_0}{4\pi}(n_u A_{ul})\phi_\nu, \tag{1.22}$$

where A_{ul}, B_{ul} and B_{lu} are the Einstein coefficients of spontaneous emission, stimulated emission and stimulated absorption associated with the transition from level

u to level l respectively (Equations 1.10 and 1.11); similarly n_u and n_l are the respective number densities of molecules populating those levels.

The systematic velocity field enters both the gas emission and absorption coefficients through the *line profile function*, ϕ_ν. The line profile is discussed in more detail in Section 1.3.1. Technically, the line profile functions associated with spontaneous and stimulated emission/absorption may differ but, by invoking the assumption of complete angular and frequency distribution of emitted photons, we treat them as being the same, vastly simplifying the radiative transfer equation without significantly reducing its validity.

As stated previously, the absorption and emission due to dust and gas may be added to find a total source function which, if known along the entire length of the path joining two points, may be used to determine the intensity of radiation incident at one point from the the other.

In order to predict the mean intensity at a point from all other points in space, the intensity must be integrated over the entire sphere:

$$J_\nu = \frac{1}{4\pi} \int I_\nu d\Omega. \tag{1.23}$$

Equation 1.23 can only be solved exactly if the level populations are known at all points in the cloud. Under the assumption of LTE (Section 1.3.1), the level populations are solely a function of temperature but this is rarely the case in astrophysical situations where the densities are too low. However, even when LTE is not valid, it is usually possible to assume *statistical equilibrium* where the rates of excitation and deexcitation in to each level, n_i, are in dynamic equilibrium (it is usually assumed that the chemical time scale is long compared to the radiative timescale so that this holds):

$$\frac{dn_i}{dt} = 0. \tag{1.24}$$

By taking into account radiative and collisional processes, equation 1.24 may be re-written as

$$n_i \left[\sum_{i>j} A_{ij} + \sum_{i \neq j} (B_{ij} J_{ij} + C_{ij}) \right] = \sum_{i<j} n_j A_{ji} + \sum_{i \neq j} n_j (B_{ji} J_{ij} + C_{ji}), \tag{1.25}$$

where A, B and C are the spontaneous and simultaneous radiative coefficients and the collisional rate coefficient respectively and where the LHS gives the rate of destruction and the RHS gives the rate of formation in level i.

The first term on the LHS gives the rate of spontaneous emission (deexcitation) from level i and the second term gives the rate from stimulated emission and deexcitation from the level caused by collisions, which it is important to note, permits the molecule to enter any rotational state. Radiative transitions are constrained to change the rotational state such that $\Delta J = \pm 1$. The RHS contains the equivalent terms for the formation of molecules in level i. The collisional coefficients are complex and are discussed in context in Chapter 2.

Solving statistical equilibrium radiative transfer is non-trivial because the problem is no longer local because the mean radiation field and the level populations are coupled to the global radiation field through equations 1.17, 1.23 and 1.25. In all but the simplest cases it is necessary to solve these equations iteratively using radiative transfer codes that discretise the problem and solve the problem at each discrete point in the grid. This is the subject of the next section of this chapter.

Local Thermodynamic Equilibrium

Thermodynamically, exchanges within a system are controlled by intensive parameters, like temperature. In general, thermodynamic equilibrium (TE) means that the intensive parameters are homogeneous to some degree. If the equilibrium is global, then these parameters are the same everywhere in the system; if the equilibrium is 'local', as in LTE then the intensive parameters may vary in both space and time but are so slowly varying that TE may be assumed in the neighbourhood about that point.

In LTE, the relative fraction of molecules in each state, $n_i / \sum n_i$ is characterised by a Boltzmann distribution characterised by a single parameter, $T_{\rm gas}$:

$$\frac{n_i}{\sum_i n_i} = \frac{g_i e^{-E_i/kT_{\rm gas}}}{Z(T_{\rm gas})} \tag{1.26}$$

where $\mathrm{Z}(T_{\rm gas})$ is the partition function, $\sum_i g_i e^{-E_i/kT_{\rm gas}}$; g_i, is the statistical weight, the number of states having energy, E_i; and n, being the total number of particles in the ensemble.

When the density is sufficiently great, LTE is mediated by collisions between particles – the volume is said to be *collision-dominated* or *thermally excited*. However, for lower densities, radiative processes effect more change in the local conditions and so the simplification breaks down.

The *critical density* is defined as the point at which radiative and collisional processes are equal to each other, i.e. where the collisional timescale, K_{ul} and the characteristic

radiative timescale, A_{ul}, are equivalent,

$$C_{ul} = K_{ul} n_{\mathrm{H_2}}^{\mathrm{crit}} = A_{ul}$$
$$\Rightarrow n_{\mathrm{H_2}}^{\mathrm{crit}} = \frac{A_{ul}}{K_{ul}}. \tag{1.27}$$

If the gas is optically thick then the probability that a photon can escape, $\beta \propto 1/\tau$, reduces. This effectively reduces the rate of spontaneous emission and thus the critical density is reduced by a factor of β. This effect explains why two molecules can have a similar critical density but trace different densities before becoming thermalised.

Spectral line broadening

In any gas, thermal motions will cause radiation to be emitted over a distribution of frequencies dependent on temperature which is sharply peaked at $\nu = \nu_0$, the rest frequency in the absence of bulk motions. The non-relativistic Doppler equation gives a relationship between velocity and frequency:

$$v = \frac{c(\nu - \nu_0)}{\nu_0}. \tag{1.28}$$

Recalling that the number of atoms or molecules with mass m having velocities between v and $v + dv$ along the line-of-sight is proportional to the 1-dimensional Maxwell distribution:

$$n(v) \propto \exp\left(\frac{-mv^2}{2kT_{\mathrm{gas}}}\right) dv, \tag{1.29}$$

then by combining equations 1.28 and 1.29 the emission between frequencies ν and $\nu + d\nu$ is expected to be proportional to

$$\exp\left(\frac{-mc^2(\nu - \nu_0)^2}{2\nu_0^2 kT_{\mathrm{gas}}}\right) d\nu. \tag{1.30}$$

Normalising this distribution so that the integral over all possible frequencies is unity gives

$$\phi_\nu = \frac{1}{\Delta\nu_D\sqrt{\pi}} e^{-}((\nu - \nu_0)/\Delta\nu_D)^2 \tag{1.31}$$

where

$$\Delta\nu_D = \frac{\nu_0}{c}\sqrt{\frac{2kT_{\rm gas}}{m}}, \quad \text{or by applying Eq. 1.28} \tag{1.32}$$

$$v_{\rm Thermal} = \sqrt{\frac{2kT_{\rm gas}}{m}} = \sqrt{\frac{2\mathcal{R}T_{\rm gas}}{\mu}} \tag{1.33}$$

where $\mathcal{R}$ is the gas constant and μ is the molecular mass in amu. As an example the thermal linewidth for $^{12}C^{16}O$ gas at 10 K is 0.077 km s^{-1}.

Except in very cold cloud cores, observed linewidths in molecular clouds are actually on the order of 1 km s^{-1} (e.g. Larson 1981; Goodwin et al. 2004). This is typically ascribed to unresolvable stochastic gas motions at a microscopic or macroscopic level (*microturbulence* and *macroturbulence* respectively). Many effects act to 'stir up' the gas but their individual effects have generally eluded a closed formula and the turbulent velocity is described by adding the parameter $v_{\rm NT}$ in quadrature to the thermal broadening parameter to give the line profile function:

$$\phi_\nu = \frac{c}{v_{\rm turb}\nu_0\sqrt{\pi}}\exp\left(-\frac{\Delta v^2}{v_{\rm turb}^2}\right) \tag{1.34}$$

$$v_{\rm turb} = \sqrt{v_{\rm T}^2 + v_{\rm NT}^2}. \tag{1.35}$$

In this definition of the line profile function, the intrinsic (natural) line broadening has been neglected as in most cases it is many orders of magnitude smaller than the Doppler broadening. Where this is not the case, the line profile can be expressed as a Voigt profile; a convolution of the gaussian line function and the Lorentzian (natural) line function.

Temperatures

Various measures of temperature are used in radiative transfer. Each one has a different meaning but they are all used in some way to equate the observed conditions with a blackbody emitting at an equivalent temperature.

By setting the magnitude of the source function to that of a blackbody emitting at the same frequency, one can write the source function in terms of an *excitation temperature*, $T_{\rm ex}$:

$$S_\nu = B_\nu(T_{\rm ex}) \tag{1.36}$$

where

$$T_{\rm ex} = \frac{-h\nu}{k\ln\left(\frac{g_l n_u}{g_u n_l}\right)}. \tag{1.37}$$

The excitation temperature is useful for characterising the ratio of two levels and can be used to determine where the level populations deviate from LTE, where $T_{\rm ex} = T_{\rm kin}$. The excitation is said to be super-thermal if the excitation temperature exceeds the kinetic temperature; otherwise it is sub-thermal. In the optically thin limit, it can be shown that the observed intensity is proportional to $T_{\rm ex}$.

The *brightness temperature* is extensively used in sub-millimetre and radio astronomy to express the temperature that a blackbody would have to have in order to match the observed intensity, $I_\nu = B_\nu$. In radio astronomy, where $h\nu \ll kT$ the Rayleigh-Jeans law is used, so that:

$$\begin{aligned} B_\nu(T) &= \frac{2kT_B\nu^2}{c^2} \\ \Rightarrow T_B &= \frac{c^2 I_\nu}{2k\nu^2}. \end{aligned} \tag{1.38}$$

This measure of intensity is often extended to observations at shorter wavelengths even though it is no longer technically valid.

More relevant measures of intensity for comparison with observations are *radiation temperature*, T_R, and the associated *antenna temperature*, T_A:

$$T_R = \frac{c^2}{2kv^2}(I_\nu - I_\nu^{\rm bg}) = T_B - T_B^{\rm bg} \text{ and} \tag{1.39}$$

$$T_A = \eta T_R, \tag{1.40}$$

where $T_B^{\rm bg}$ is the Rayleigh-Jeans temperature of the background and where η takes into account various corrections for atmospheric attenuation, radiative loss, scattering, spillover and main-beam efficiency (Kutner & Ulich 1981).

1.4 Computational approach to radiative transfer

Historically, poor spatial and spectral resolution of astrophysical objects combined with insufficient computational power has led to the widespread use of approximate methods, including the assumption of LTE throughout the object which is rarely correct and using so-called 'escape probabilities' (see below).

As shown in the previous sections, in order to interpret molecular line data it is necessary to accurately determine the level populations of each required molecule for the entire object. The solution to the coupled equations of radiative transfer and statistical equilibrium is both non-trivial and non-linear, necessitating the use

of iterative schemes that converge on a set of global level populations that are consistent with the mean radiation field that they generate (Pavlyuchenkov et al. 2007).

In this section I address some of these approximate and so-called 'exact' methods.

1.4.1 Approximate (local) methods

The simplest source model in line radiative transfer is that of LTE. In LTE, the source function is assumed to be a blackbody radiating at the local temperature of the system. Moreover, the molecular level populations are a function of a single temperature only and are not coupled to locations elsewhere in the object. This approximation is permissible in certain scenarios like the dense, collisionally-dominated midplane of a circumstellar disc. However, as the gas becomes less dense (or, more generally where radiative processes become significant) this approximation rapidly breaks down and masses derived assuming LTE often underestimate the true mass of an object (Pavlyuchenkov et al. 2007).

The Sobolev approximation permits the very fast solution of the RT equation in media with large velocity gradients, hence it is also known as the LVG approximation. It is an excellent approximation when dealing with stellar winds or rapidly expanding envelopes, where velocities can be many hundreds of kilometres per second. Physically, it takes advantage of the fact that photons that are emitted locally are only likely to be absorbed locally; their line profiles being shifted too far away from that of other areas to interact further away. Mathematically, the integration of the transfer equation over space can be replaced with an integration over frequency, substantially simplifying the problem (Wehrse & Kalkofen 2006). This method is one of many 'escape probability' approaches that have been used to rapidly determine the basic physical parameters of molecular clouds using molecular line data. Another similar method is the 'one-zone' escape probability method which can be used in isothermal and homogeneous media (e.g. Mihalas 1978). The probability in this case indicates the chance of a photon escaping a cloud without undergoing scattering or absorption. However, LVG analyses suffer when applied to real molecular clouds that do not exhibit large velocity gradients and which are not isothermal and homogeneous, indeed, they are often far from it.

All other local approximations are variants on this probabilistic theme. 'Full' escape probability codes assume that the photon will definitely escape instantly (assuming the line is very optically thin) (e.g. van der Tak et al. 2007); VEP (vertical escape probability) and VOR (vertical only radiation transport) methods utilise the

known velocity profiles of a disc undergoing Keplerian rotation in order to speed up their level population calculations but they are still not true non-local methods (Pavlyuchenkov et al. 2007).

1.4.2 Exact (non-local) methods

More recently, sophisticated non-local line radiative transfer codes have been developed. Initially these were one-dimensional Lambda Iteration methods but their evolution has culminated in the development of three-dimensional accelerated Monte Carlo methods as the available computational resources have increased. Their necessity as analytical tools has arisen as it has become possible to resolve the more complex kinematic and chemical environments of, for example, dense cloud cores (e.g. Hogerheijde & van der Tak 2000, and many others) and more recently protoplanetary discs (e.g. Semenov et al. 2008).

In this section I present a broad overview of the methods that have been used in molecular line radiative transfer following van Zadelhoff et al. (2002) which provides an excellent summary of the implementation of some contemporary codes.

The Lambda Iteration method

The Lambda Iteration is the most intuitive of all non-local methods. It is eponymously titled for its construction and use of the Lambda operator, Λ, which represents the process of determining the mean radiation field impinging on each point in space from the source function across the whole region as calculated in the previous iteration:

$$J_\nu^{\rm new} = \Lambda \left(S_\nu^{\rm old} \right). \tag{1.41}$$

Mathematically, it involves the formal integration of equation 1.23 over the entire spatial-frequency domain. The Lambda operator can be expressed as a matrix that connects each grid point and each level. By assuming some initial set of level populations, the Lambda operator is constructed by solving 1.23 and using it to find updated level populations which are then used as the next trial solution. In subsequent iterations, it is expected that the relative change in level populations reduces and that a self-consistent numerical solution for J in the cell has been reached when, after the final iteration, the relative change in the level populations of successive iterations is less than some predetermined limit (since the solution is not known *a priori*). This does however, rely on the assumption that when the mean intensity

and the level populations are far from their equilibrium solution, the change between iterations is large which is not generally the case in optically thick media.

High optical depth has been identified as a factor affecting convergence and Hogerheijde & van der Tak succinctly surmised that '...emission passing through an opaque cell will rapidly lose all memory of its incident intensity and quickly tend towards the local source function. The distance over which the information about changes in excitation can travel is one mean free path so the required number of iterations grows $\propto \tau^2$ characteristic of a random walk'. Thus, in regions of high optical depth, the level populations may change very slowly and care must be taken to ensure that the 'convergence' towards the solution is real, as iterative processes do not converge perfectly monotonically and fluctuations in their convergence ratios may trigger the convergence criteria prematurely. This is discussed in greater detail in Chapter 3.

The accelerated Lambda Iteration method

Lambda iteration methods converge very slowly for high optical depths. It is possible to accelerate the convergence of the method by considering an approximate operator, Λ^* (Rybicki & Hummer 1991) constructed from the full operator, thus:

$$\Lambda = (\Lambda - \Lambda^*) + \Lambda^*. \tag{1.42}$$

Equation 1.41 can be written as,

$$J_\nu^{\rm new} = (\Lambda - \Lambda^*)[S_\nu^{\rm old}] + \Lambda^*[S_\nu^{\rm new}]. \tag{1.43}$$

An ALI iteration proceeds similarly to an LI iteration except that whilst equation 1.41 is fully solved in LI, only the second approximate term is solved in ALI. In ALI, sub-iterations are performed so that the level populations are consistent with the approximate operator and then only once all the cells have been updated in this way is the full operator updated.

The approximate operator is typically chosen to be easily inverted making the problem readily soluble. By happy coincidence Olson et al. (1986) demonstrated that the local (diagonalised) Λ operator was almost optimal. This approximate operator is trivial to invert and easy to construct by using only the conditions within the cell. Because the dependency on data outside the cell is greatly reduced, it is possible to converge to a solution in optically thick media.

(Accelerated) Monte Carlo methods

The main difference between Lambda Iteration codes and Monte Carlo codes is in the sampling of the mean radiation field that affects the level populations. LI codes use a fixed grid yielding a solution without the noise associated with random sampling at the risk of potentially undersampling the radiation field in other directions.

There are a number of advantages in using Monte Carlo methods over conventional methods. By definition one does not need to consider every possible configuration in the ensemble but merely sample from it, reducing memory and time requirements, allowing more consideration to be given to the physics of a model. Specifically, instead of performing a formal integration of some variable over all space, one evaluates a finite sum approximating an integral:

$$\frac{1}{4\pi}\int_0^\infty f(\mathbf{x})dV = \sum_{i=1}^{N} f(\mathbf{x}_i), \tag{1.44}$$

where N is the number of samples (with index i) to be taken of the function, f. Furthermore, it is possible to quantify the random errors present in the solution by monitoring the variance in the solution.

Moreover, a major advantage of Monte Carlo codes is the ease of their extension to non-regular grids (e.g. adaptively refined meshes, see Chapter 2) which are able to be more flexible in their coverage over many spatial scales, reducing the number of cells required to cover a space.

Although Lambda Iteration numerical radiative transfer codes had been developed earlier in the 1970's it was Bernes (1979) who first used the flexibility of Monte Carlo methods to solve non-LTE line radiative transfer in an astrophysical context, recognising that similar accomplishments had been made using the MC method in radiative transfer in other areas of physics. He demonstrated that a well-converged non-local solution could be arrived at within a feasible timescale (at least for one-dimensional source models). Although Bernes used a simplified spherical, collapsing cloud, his work laid the foundations for many efforts to come.

Bernes determined the level populations by using randomly sampled discrete photons that are emitted and absorbed in the material; this however can lead to poor convergence in undersampled cells. An alternative formulation considers each cell and considers the mean radiation field that impinges upon each cell, ensuring that no cell is undersampled (although depending on the number of rays used, it is still possible that the radiation field is not well sampled in either direction or frequency).

One of the well-documented problems with MC simulations come from their slow 'square-root' convergence (Press et al. 1992), although this can be overcome by employing acceleration techniques (e.g. Ng 1974) and variance reduction techniques (e.g. Juvela 1997; Juvela & Padoan 2005).

Also, analogously to the ALI method, it is possible to split the contribution of the radiation field internal and external to the cell. By using sub-iterations to consider only the local radiation field, a similar increase in performance as seen in the ALI method over the LI method in optically thick media can be achieved.

1.5 Applications of molecular tracers

Since the discovery of cosmic CO in 1951 (Ewen & Purcell 1951; Muller & Oort 1951), it has yielded more information about star-forming regions than any other molecule. Moreover, for over 40 years, molecular line radiation has regularly been used to trace key evolutionary parameters.

The lowest rotational levels of Carbon monoxide are used as tracers of low density gas, because its critical density is low ($n_{\rm cr} \approx 10^3$ cm^{-1}, see Section 1.3.1). At higher densities, because the rotational constant of CO is relatively high compared to other molecules the levels are more closely spaced, the transitions become readily thermalised and the line ratios become sensitive to temperature. However, owing to the high relative abundance of the most abundant isotopologue of CO, ^{12}C^{16}O, lower transitions are rapidly saturated for typical column densities associated with molecular clouds. Less abundant isotopologues like ^{13}CO and C^{18}O must be used to probe conditions deeper within the interior of the cloud. In an extra-galactic context the molecule is even more important as its abundance is its strength as it is readily observable in extra-galactic sources. CO is thought to be a good probe of the entire mass of an object because its abundance relative to H_2 can be well constrained. The observed intensity of this line is strongly correlated to the hydrogen column density (e.g. Frerking et al. 1982) (although it should be noted that a direct determination found the ratio to be three times higher, Lacy et al. 1994).

Conversely, the CS molecule has a large dipole moment and its emission is not very sensitive to temperature. Consequently, by examining the ratio between different emission lines, say (2-1)/(3-2), and assuming some relative abundance to H_2, it can be used to determine the H_2 density more accurately than many other molecules. This is because the degeneracy between gas temperature and density make it difficult

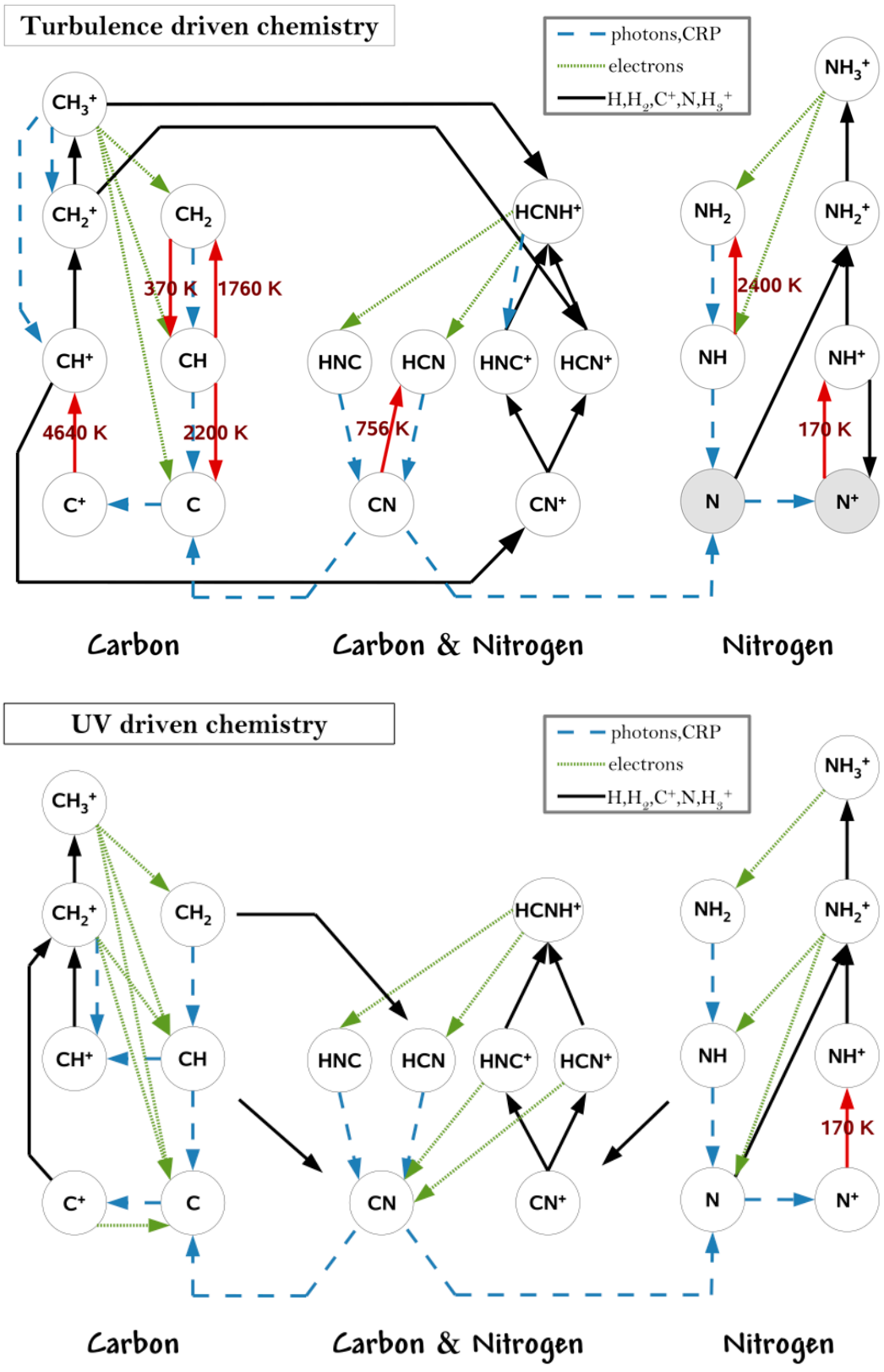


Figure 1.7: Top panel: Chemical network of turbulence-dominated chemistry. Bottom panel: Chemical network of UV-dominated chemistry. Reproduced from Godard et al. (2010).

to disentangle the two parameters. van der Tak et al. (2000) have used their radiative transfer code to demonstrate this effect for some molecular species (Figures C1-C5).

More generally, for any molecule, higher frequency transitions will probe high temperatures and densities as the critical densities of molecular lines scale $\propto \mu^2\nu^3$ (Equation 1.6). Moreover, non-linear molecules such as H_2CO and NH_3 are advantageous because it is possible to probe both temperature and density within the same range of frequencies. Line ratios from different J states tend to trace density whereas different K states (on the rotational backbone) trace temperature (e.g. Stutzki & Winnewisser 1985).

While it is critical to understand the temperature and density profiles of an object as they broadly define the physical conditions of an astrophysical object, secondary characteristics may be used to separate two otherwise similar objects. Molecular tracers have been used to determine magnetic field strengths, ionisation and deuteration fraction (e.g. Bourke et al. 2001).

The utility of molecular lines as tracers however is tempered by the fact that the relative abundances of a particular molecule must be well constrained. As a first-order approximation some species can be assumed to have an approximately constant abundance throughout the cloud and this can be quite accurate. Some species however are sensitive to the abundances of other species and also to the evolution of the cloud. More complex chemical networks attempt to model this but they often assume very simple conditions (e.g. homogeneous temperature and density field) and more sophisticated treatments couple these networks to radiative transfer codes. A simplified example of a chemical network showing the interconnection of some species with others is depicted in Figure 1.7. It should be noted that the full networks contain hundreds of species and typically contain thousands of reaction pathways.

1.6 Thesis aims

For the reasons outlined already in this chapter it is clear that molecular line data from observations is an extremely powerful tool for analysing the properties of many astrophysical objects, from giant molecular clouds spanning many 100s of parsecs to dense clouds cores to protoplanetary discs around young stars. Historically, the use of approximate transfer methods to analyse this data has been out of necessity; today, with the ready availability of parallel processing facilities it is possible to use non-LTE models to recover more accurate estimates of the physical parameters

governing an object. Moreover, as radiative transfer provides a gateway from theory to observation, it can be used, for example, to create synthetic maps of hydrodynamic simulations permitting their comparison with observational data. In effect, radiative transfer codes hold the key to attacking the discrepancy between theory and observation from both ends – their critical importance cannot be understated. Consequently, I have developed a fast, efficient, accurate line transfer code that can be used in all these respects.

In Chapter 2 I present an implementation of a non-LTE molecular line RT module using the AMC method for determining molecular line populations for TORUS, a multi-purpose AMR radiative transfer code. In Chapter 3, I demonstrate agreement with other line transfer codes in some of the benchmark problems set out in van Zadelhoff et al. (2002) and verify the accuracy of the raytracing routines used for generating synthetic intensity maps. In Chapter 4, I present an efficient method for mapping irregular SPH data on to an AMR grid as used in Douglas et al. (2010), Acreman et al. (2010b) and Rundle et al. (2010) and in Chapter 5, I present the results of its application to the hydrodynamic clustered star-formation simulation of Bate et al. (2003a).

Finally, in Chapter 6 I conclude on the significance of the work contained in this thesis and report on the future work that it will be possible to do using the code that I have developed.

I consider the work presented in [Rundle, Harries, Acreman & Bate (2010)] to be a major step forward in providing a tool to understand the star formation process in all its complexity. After having seen years and years of simulation presentations lacking attempts to make a relation to observations, it is refreshing to see ... meaningful analysis tools for their simulations. In view of the expected Herschel and ALMA data, the manuscript is excellently timed to strengthen a communication between astrophysicists working on observations and simulations.

— Anonymous referee for Rundle, Harries, Acreman & Bate (2010)

Chapter 2

Molecular Line Radiative Transfer using TORUS

TORUS (an acronym, according to the webpage of the original author, Dr Tim Harries for, Transport of Radiation Under Sobolev or Transport of Radiation Using Stokes) is a Monte Carlo radiative transfer code that was originally designed to model atomic line transfer in a moving medium in order to compute polarisation images and spectra on a generalised three-dimensional opacity grid. It is written in FORTRAN-2003 and has been designed to be as general in application as possible. To this end, the code is highly modular and has incorporated techniques such as AMR (Berger & Colella 1989) to enable the adequate resolution of, for example, dust nebulae (Harries et al. 2004) where densities can change by many orders of magnitude. The code is fully parallelised using the Message Passing Interface (MPI) libraries and as each Monte Carlo process can be run independently, the code benefits greatly from running in parallel on multiple processors.

TORUS has been used to examine astrophysical phenomena that occur over many orders of magnitude; from massive and hot Wolf-Rayet and O stars (Harries et al. 2000, 2004), to YSOs of all classes and their encircling discs/envelopes (Symington

et al. 2005a,b; Kurosawa et al. 2006; Hatchell et al. 2007). Now, using the molecular line radiative transfer module I developed for TORUS, large scale cluster calculations have been performed (Rundle et al. 2010) and even galaxy-scale calculations (Acreman et al. 2010a; Douglas et al. 2010).

Until relatively recently, only spherically or axially symmetric core collapse simulations of line transfer had been performed (e.g. Tsamis et al. 2008; Pavlyuchenkov et al. 2008) however, due to recent advances in computational power, it is now possible to compute three-dimensional line transfer for these complex density structures in star-forming regions. As a result, my primary Ph. D. project was to develop, benchmark and apply a molecular line transfer module for TORUS, the result of which is now known as `molecular_mod` in the TORUS code. As the name suggests, the module requires the main TORUS subroutines for grid creation and various 'bookkeeping' tasks but is self-contained insofar as it only contains all the routines for, e.g., determining level populations and creating synthetic images. Currently, the module comprises about 5500 lines of FORTRAN. Figure 2.1 captures the salient details of the determination of the non-LTE level populations for a user-specified model. By supplying some criteria or accepting sensible program defaults, the user ultimately obtains the end product of a converged grid which can be used to create synthetic line maps/profiles. An abundance of diagnostic information is available to the developer or keen user.

This chapter describes how I have implemented the code in TORUS. It is split broadly into 3 main sections; the creation of an appropriate AMR grid (Section 2.1), the determination of non-LTE level populations (Section 2.2) and the generation of synthetic line maps/line profiles from a converged grid of level populations (Section 2.3). Convergence, acceleration and optimisation are also discussed towards the end of this chapter (Section 2.4) but the results from the benchmarking application are presented in chapter 3.

The TORUS code is currently undergoing substantial revision in order to make it more user-friendly; this will be known as TORUS version 2. This chapter documents the functionality of the molecular line transfer module for TORUS version 1.2 at the time of submission of this thesis (July 2010).

2.1 Grid generation

Problems in star formation typically span many orders of magnitude in both space and density. AMR naturally resolves the fine structure of these problems well

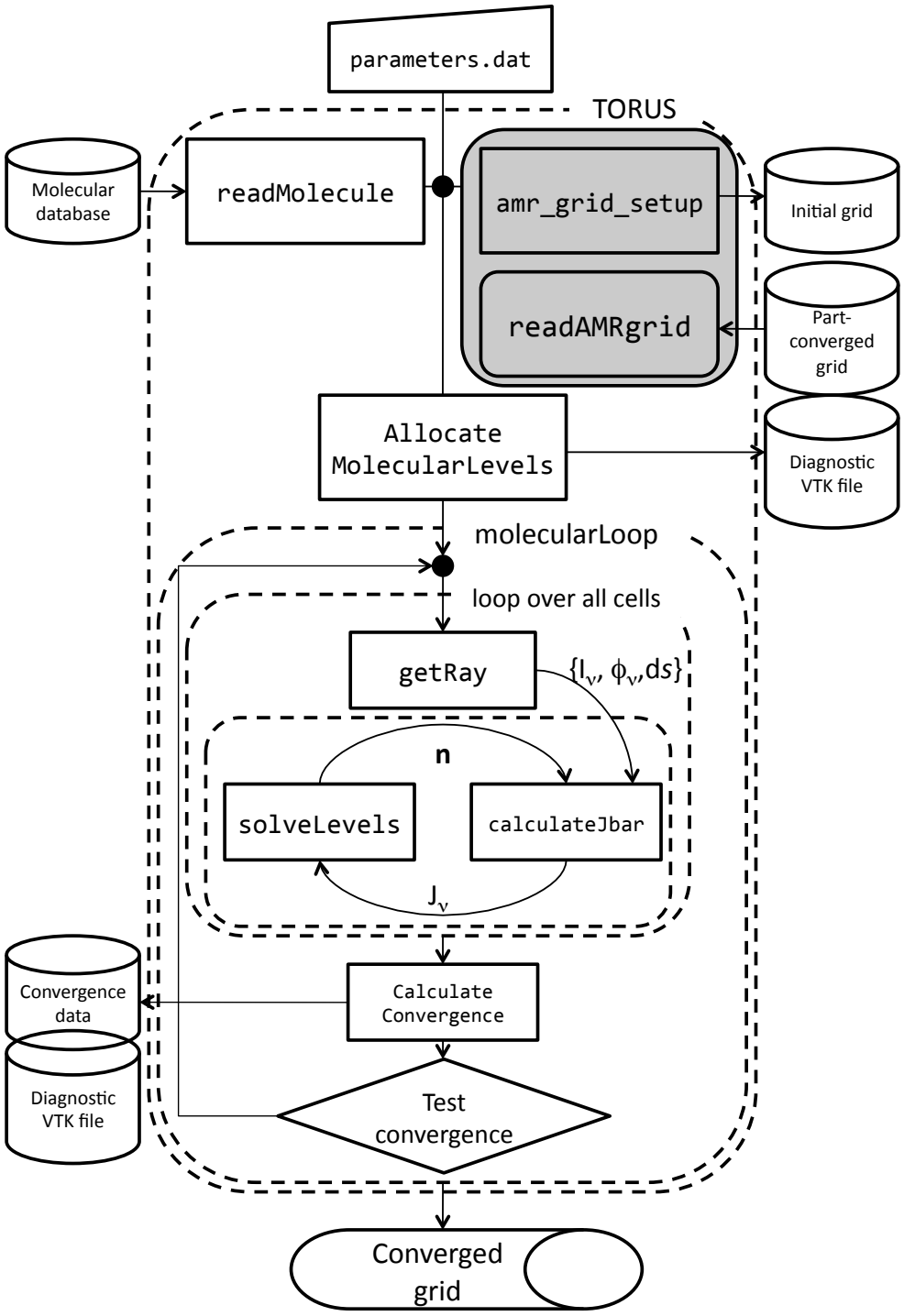


Figure 2.1: Abridged programme flowchart of TORUS when calculating non-LTE molecular level populations. Highlighted in grey are important TORUS routines that facilitate the execution of the code but I did not contribute greatly to the development.

whereas a fixed, regular grid will often be insufficient to capture crucial information like peaks in the density profile where a protostellar core far smaller than the size of a grid cell has begun to accrete material. Moreover, the resolution of a fixed grid may be unnecessarily fine in large, low density regions that are unimportant in a star formation context. Nested grids can ameliorate this problem somewhat (D'Angelo et al. 2002) but they often need to be hand-crafted to fit each individual problem. The flexibility of AMR is that it will adapt the resolution given to a region depending on the criteria the user applies. Typically, one is interested in resolving the temperature and density structure in a region (ensuring that the optical depth of a cell is not too high), although when calculating line radiative transfer accurately it is necessary to accurately resolve variations in the velocity field as well, owing to the anisotropy in absorption introduced by the line profile function.

Three-dimensional grids have become a more popular tool for modelling objects without any axes of symmetry as the computational power available to numerical modellers has increased. Many astrophysical fields are expected to act to disturb the symmetry of an object placed within them (e.g. anisotropic magnetic fields) and thus these grids are essential for accurately modelling the effect of these phenomena.

In order to create an AMR grid it is necessary to define how the simulation space will be split and how each physical parameter for each cell will be determined. User-specified criteria are used to control the resolution of the grid at any given point in space. In TORUS this is achieved by the repeated subdivision of an initial unrefined cell centred on the region of interest. This *parent* cell is bisected once in each dimension, thus, in Cartesian coordinates, each *child* has a volume $2^{-\nu}$ times that of its parent, where ν is the dimensionality of the simulation. The simulation region may also be specified (and split) by cylindrical coordinates. Each cell is recursively refined until it no longer satisfies the splitting criteria or it has reached some pre-determined depth, that is, it has been split as many times as the user has determined to be necessary (in order to limit the amount of memory used by the grid). More detail of the AMR implementation used in TORUS is given in Harries (2000) and Symington et al. (2005a) and a comprehensive discussion of the mechanism by which TORUS executes this can be found in Symington (2006).

The initial splitting criteria to be used in a simulation depends on what is known *a priori* while subsequent adaptive refinement criteria may change as the calculation proceeds. These initial criteria depend on the *geometry* being used by TORUS to model the physical situation. Geometry is a key input parameter in TORUS that defines the physical parameters of the simulation. In the theoretical benchmark model examined in chapter 3, all the salient parameters are tabulated. In this

case, the splitting criteria ensure that the parameters for each octal can be uniquely defined by one datapoint. When 'gridding' an SPH simulation (see Chapter 4) it is necessary to refine the grid by mass per cell, density gradient and the range of velocities contained with a cell. Further details on these criteria can be found in their respective chapters.

The final grid may undergo further processing if it is determined to be necessary. In some geometries it is possible that large cells may neighbour a number of very small cells if, in the case of an accretion flow geometry or a jet, the local density gradient is very high. This can cause numerical instabilities in the determination of physical parameters especially in cases where it is necessary to perform interpolation. The grid may be smoothed after this initial refinement to ensure that the difference in cell depth of adjacent cells is no greater than the maximum permissible size ratio between cells. In geometries where the opacity is simply proportional to the path length through the cell (e.g. continuum radiative transfer), a further refinement can be performed where the grid is refined to ensure that the optical depth through one cell at some frequency does not exceed some user-defined limit. The grid is stored in a tree structure. In three dimensions, this is called an *octree* to reflect the fact that each branch of the tree can spawn 2^3 nodes recursively. In one and two dimensions each branch spawns fewer nodes but for brevity, in this thesis each node may be called a cell, or *octal*.

Once the structure of the grid *octree* is known, it is necessary to fill each octal with the values that will be used to perform the calculation. Each octal is implemented as an instance of a Fortran 95 derived type, akin to a structure in the C-family of languages, greatly improving the readability, maintenance and development of the code. Position, linear size, density, temperature and velocity are the only parameters that are statically allocated for every octal. As the functionality of TORUS has been enhanced, the number of grid variables required to perform the requisite calculations has increased and in order to save memory, almost every other quantity is initialised as a null pointer that can be allocated *post hoc*. Allocation of memory for variables associated with molecular line radiative transfer is handled by the `allocateMolecularLevels` subroutine and is discussed in Section 2.1.2.

The `calcvaluesAMR` subroutine recursively traverses the octree assigning values to the fundamental physical quantities according to a combination of user-defined parameters and/or tabulated data/analytical formulae. From here on, I shall only discuss what is necessary to perform molecular line radiative transfer. Other modes of operation will usually be broadly similar.

It must be stressed here that the line code described in this chapter requires a gas

temperature field as an input, be it tabulated or self-consistently determined assuming radiative equilibrium. In the latter case, an appropriate geometry containing at lease one radiation source must be used. TORUSuses an iterative algorithm described by Lucy (1999) to determine the gas temperature for each cell. Owing to the negligible emission in the sub-millimetre region of the spectrum of any 'point-like' source (e.g. a protostar), it is only through their contribution to the temperature of the region of interest that their radiation field is accounted for in the line transfer code.

As it is critical when considering line transfer, to have an accurate representation of the velocity field, it is often necessary to perform sub-grid calculations where it might be impractical or impossible to split the grid sufficiently to account for the changes in velocity; the most illustrative example being the existence of a large radial velocity gradient in one direction and no gradient perpendicular to that as is found in a rotating disc. Consequently, velocity vectors are stored at up to 27 points within the parent octal corresponding to the corners of the 8 subcells (in the three-dimensional case) facilitating the performance of up to quadratic interpolation without access external octals. For certain geometries, an identical scheme has been implemented for storing extra density information on the parent octal corners, resulting in improved quality images, but currently only linear interpolation is possible because simple higher-order interpolation schemes do not guarantee that the interpolated value lie within the range of the fiducial values. Obviously, the potential for the grid to contain negative densities is highly undesirable and there are diminishing returns from increasing the interpolation order.

For some geometries, like the `cluster` geometry used in chapter 5, the process of populating the grid is non-trivial and can be time-consuming. At the end of the process TORUS saves an intermediate grid filled with the fundamental physical quantities but with no molecular data superposed on top.

The current mode of operation for the line transfer module is to calculate the level populations for each molecule of interest separately, so this intermediate grid may be determined and reused for multiple molecular species. This is a valid assumption where the lines of different species do not overlap. Similarly, the probability of molecular species other than H_2 colliding with each other is vanishingly small and this may be neglected. Consequently, the user chooses the molecule that they wish to calculate level populations and/or a synthetic line map for and the corresponding data is fetched from the file in the TORUS data directory using the `readMolecule` subroutine (described below). The level populations and other relevant parameters are then superposed onto the grid using `allocateMolecularLevels`. Once the

molecular data are known, the AMC code can begin to iterate on the initial grid.

2.1.1 The `readMolecule` subroutine

TORUS uses molecular data from the LAMDA database (Schöier et al. 2005), a repository of molecular and atomic data for many common species found in molecular clouds. The data is stored in the RADEX file format (van der Tak et al. 2007) which contains data pertaining to the energy levels, E_i, statistical weights, g_i, Einstein A-coefficents, A_i and collisional rate coefficients, C_{ij}, for whichever collision partners have been calculated; all required inputs for performing statistical equilibrium calculations. The authors of the database have compiled the data from a number of sources; energy levels, transition frequencies and Einstein A-coefficients are usually taken from the Jet Propulsion Laboratory (JPL) and/or CDMS whereas collisional data is collated from various reference sources (e.g. Green & Chapman 1978; Monteiro 1985).

The `readMolecule` routine reads the data for each level or transition and stores it in a global `moleculetype` data structure containing many dynamically allocated fields (because it is not known *a priori* which molecule will be modelled and consequently how many transitions/levels need to be stored).

The rest of the file is given up to describing how the collision rates vary with specific collision partner and temperature. Unlike radiative transitions where specific transition rules apply between levels (e.g. $\Delta J = \pm 1$ for dipole transitions), collisions may excite or de-excite a molecule from any level to another. Furthermore, because the collision cross-sections vary as a function of temperature, the array needed to store the collision rates between partners is far larger.

The RADEX format allows for multiple collision partners for each molecule. Typically this will either be for ortho– and para-H_2 which have different collision cross-sections, or some combined values for H_2 as a whole which are usually taken from [molecule]-He data which have been scaled. In TORUS it is assumed that the ratio of ortho– to para-hydrogen, [ortho-H_2]/[para-H_2] = 5×10^{-5} for most astrophysical situations, under the assumption that the $J = 0$ and $J = 1$ levels of H_2 are not thermalised even at high densities (Walmsley et al. 2004). Other collision partners such as electrons, neutral and ionised hydrogen and helium may also be considered.

Temperature dependent collision rates are tabulated and range from $2 - 3000$ K in 25 unequally spaced data points in the case of CO (Yang et al. 2010) to 8 points covering a range of temperatures from $15 - 300$ K for NH_3 (Danby et al. 1988), depending on the current level of understanding of the specific molecule.

As a module variable, the `molecule` structure is accessible by all subroutines contained within `molecular_mod`.

2.1.2 The `allocateMolecularLevels` subroutine

The `allocateMolecularLevels` subroutine allocates the memory required to store all of the variables necessary for the calculation of the non-LTE level populations. Like many other subroutines it uses the same code to fully traverse the grid's octree structure; allocating memory only to those octals who do not have any child cells. The code fragment which controls this is given in Appendix C.

In the non-LTE case, where the relative population of molecules in a given state is no longer solely a function of temperature, the relative populations of each level in each cell is unique. This means that the arrays required to store this data can become very large. Furthermore, because of the large dynamic range of values that must be accurately stored, each array must be stored as a `double precision` variable, doubling the size of the array. In the three-dimensional case, the storage requirement can easily become as much as 160 MB (40 levels $\times$ 10^6 cells $\times$ 4 bytes per level per cell). Today, this might not seem like very much memory but combined with the other large arrays that are required to store previous levels to test for convergence and the temporary arrays that store estimates of the external radiation field the memory requirement can rapidly exceed 2 gigabytes, the maximum memory per core on a standard node on *zen*, the University of Exeter supercomputer (the most efficient use of *zen* achievable using MPI).

Often, in the kind of astrophysical scenarios where molecular chemistry can occur, the gas temperature is very low. It is rarely necessary to take into account, for example, the $J = 40 - 39$ transition of some molecule even if the `molecule` structure contains that much information. It is possible, using the `setmaxlevel` input parameter or allowing the heuristic algorithm in `findMaxLevel` to set the maximum level (and hence upper level for a transition and the overall number of transitions to be considered, `maxtrans`) for which it is necessary to store level populations and the mean radiation field within the cell, $\bar{J}_{\rm ul}$. This can drastically reduce memory consumption and increase computation speed at the expense of reduced accuracy in the uppermost levels of the simulation.

As an initial condition for the iterative calculation, the molecular level populations can be initialised such that they are in LTE and so follow a Boltzmann distribution (see equation 1.26) or so that all levels are unpopulated except for the $J = 0$ level.

The choice of initial conditions is user-specified depending on the scenario; if the solution for the model is known to be close to LTE then LTE is a good choice as it saves many unnecessary code iterations. Naturally, for a solution to be truly converged, the final solution should be independent of initial conditions. Values for $\bar{J}_{\rm ul}$ are stored in `jnu` for each cell and are initialised such that they are consistent with the level populations within the cell (equation 1.16).

In addition to these variables, the following must also be allocated and initialised for each octal if they have not been already: the microturbulent velocity $v_{\rm turb}$, `microturb` and its reciprocal; the molecular abundance relative to H_2, `molAbundance` and where it is necessary to simulataneously account for gas and dust in the radiative transfer calculations, the blackbody emission at each transition rest-frequency, `bnu`.

The reciprocal of the microturbulence is stored explicitly because it is used so frequently in the calculation of the line profile. The memory footprint is only 4MB for 10^6 cells and the speedup is significant.

The `allocateOther` routine is similar to the main subroutine except that it allocates and initialises grid variables that are not so critical to the calculation as those allocated in `allocateMolecularLevels`. This primarily includes the self-explanatory `newMolecularLevel`, `oldMolecularLevel` and `oldestMolecularLevel` which are used to save the level population data from previous iterations principally for the purposes of determining convergence but also to facilitate the Ng Acceleration step (described in more detail in Section 2.4.3) which can accelerate the convergence of the solution considerably, especially in very optically thin and optically thick scenarios.

A useful facility offered by TORUS is the ability to restart a partially converged grid – a so-called 'warm start' – if for instance, a calculation exceeded its allotted time on a supercomputer. If the `restart` flag is set then it is only necessary to call `allocateOther`; everything else is read in at run-time.

2.2 Determining non-LTE level populations

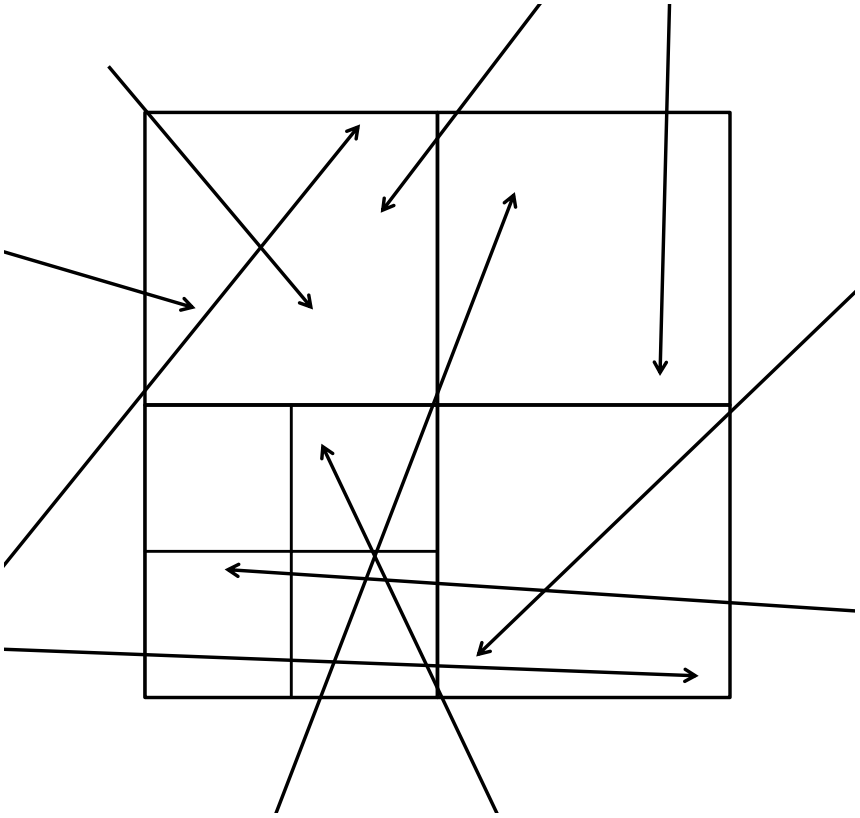
In order to determine the non-LTE populations within a cell it is necessary to calculate the mean radiation field for all molecular line transitions both within and external to the cell. As the level populations and the local radiation field are dependent upon each other, they must be found self-consistently through multiple iterations of the Lambda operator, Λ (equation 1.41).

TORUS adopts an accelerated Monte Carlo (AMC) method similar to that described in Hogerheijde & van der Tak (2000). The primary innovation in the AMC scheme is the splitting of this operator into a local and external component. The separation of local and external contributions to the radiation field in a cell facilitates convergence even in optically thick regimes ($\tau > 100$). This is analogous to the operator-splitting step in Accelerated Lambda Iteration (ALI) methods (see Rybicki & Hummer 1991). In particular, Hogerheijde & van der Tak's method uses rays, or *long characteristics* to sample the external radiation field as opposed to tracking photon packets (c.f. Bernes 1979; Lucy 1999), which can become trapped or fail to penetrate some regions leading to poor convergence or sometimes failure to converge at all. Using this method, each grid cell is sampled individually meaning that no cell is undersampled. Figures 2.2(a) and 2.2(b) highlight the fundamental differences between the two representations.

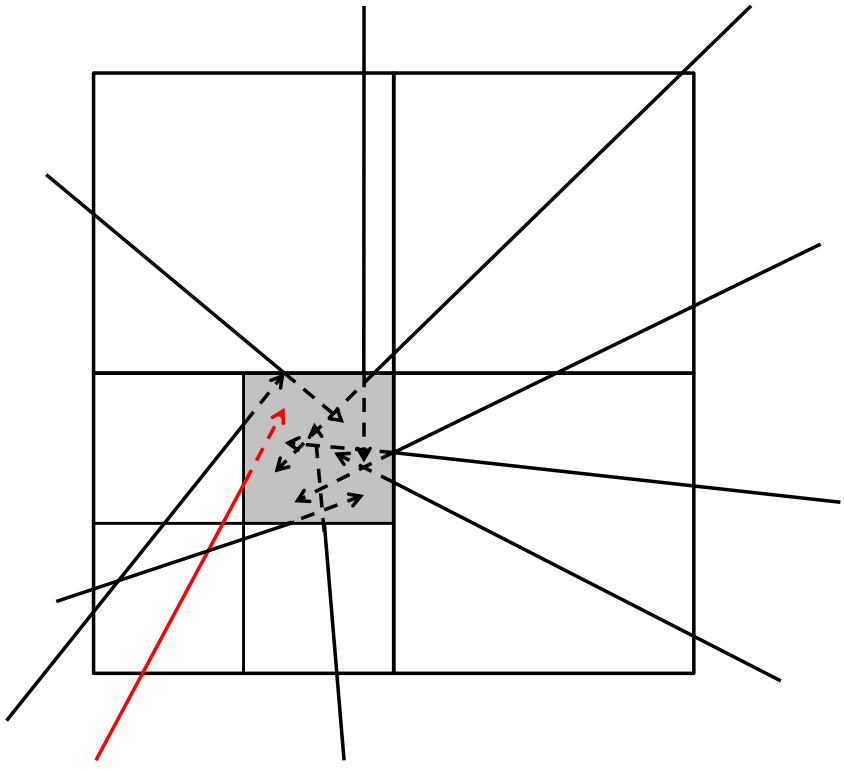
In the photon packet representation of the Monte Carlo method for solving radiative transfer (Figure 2.2(a)), the radiation field is represented by individual photon packets that emanate from an arbitrary position in the grid (corresponding to spontaneous emission) and travel through the grid until they are absorbed or escape. The packets convey information about cells that they have traversed and contribute to the cells that they pass through. A potential disadvantage is illustrated in the figure where it demonstrated that it is possible that some cells may be poorly sampled or completely unsampled.

Using the method of Hogerheijde et al. an estimate of J_ν is determined by choosing a certain number of rays that will enter the cell from the grid edge. The path that the ray took is then retraced (from the cell towards the grid edge along the long characteristic) along the random direction vector in order to find the ray's contribution to the field within the cell. There is no issue of undersampling because each cell is evaluated using the same number of rays. The main issue is the computation of potentially using so many more rays to sample some cells which will not require such good sampling. Furthermore, it is possible that the set of rays will still fail to convey information between two resonant cells (by not passing through both).

In the AMC/ALI schemes, as opposed to regular MC/LI schemes, only the local component is updated and iterated until the radiation field and the level populations within the cell are self-consistent. The external component is determined only once per cell by sampling the external radiation that would impinge on the cell from the other cells in the grid by means of random samples of intensity incident at the cell surface over 4π steradians. This is where the Monte Carlo element presents itself



(a) Photon-centric model.



(b) Cell-centric model.

Figure 2.2: Schematic representations of photon-centric and cell-centric methods of determining the mean radiation field felt by a cell.

although it is not technically necessary to use random numbers at all if care is taken to mitigate against systematic errors.

The AMC scheme is split into 2 consecutive stages; the first stage samples the intensity of radiation incident upon each cell systematically using a small set of rays each possessing a unique direction, $\mathbf{d}$, origin within the cell and frequency, ν. The direction and frequency of the set of rays do not vary from iteration to iteration, nullifying the random fluctuations in coverage of the radiation field and consequently converging rapidly towards a self-consistent solution. However, because the radiation field is poorly sampled both spatially and in frequency, the solution will be strongly dependent upon the choice of initial random seed.

The second stage of the scheme reduces systematic errors by allowing the direction and frequency of the rays to vary from iteration to iteration, improving coverage of the spatial-frequency parameter space, reducing systematic errors. Random errors are reduced by doubling the number of rays used to sample the radiation field per cell per transition in each successive iteration. It is trivial to show that the noise in a solution from a Monte Carlo simulation diminishes proportional to $\sqrt{N}$ asymptotically, where N is the number of rays used to sample the field, so it is quite possible to have to sample a cell with in excess of 10^5 rays in order to achieve global convergence on the order of 1%.

At the end of each iteration over all cells, the new level populations are tested against those of the previous iterations to quantify the convergence of the algorithm. The iteration is repeated again and again until subsequent levels are within a user-specified tolerance. This cycle of improvement is illustrated in Figure 2.3.

In TORUS, the process is executed by the `molecularLoop` subroutine and the helper routines, `getray`, `calculateJbar` and `solveLevels`. I document the method of action of each routine henceforth.

2.2.1 The `molecularLoop` subroutine

The `molecularLoop` subroutine controls the whole non-LTE level population determination process. It is this subroutine that is called from within the main TORUS code. The filled `grid` octree structure and the `molecule` structure are the only parameters passed to it explicitly although it uses other public variables stored within other modules. These variables are set at run-time in the parameters file, `parameters.dat`. A full description of these parameters can be found

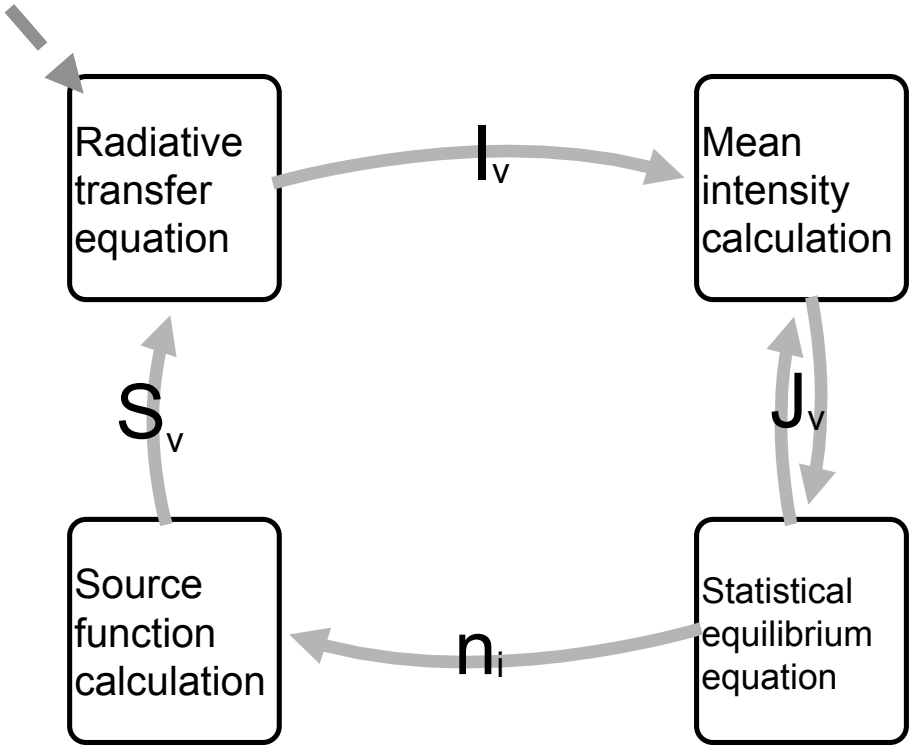


Figure 2.3: Iterative cycle of improvement for determining a self-consistent solution to the system of equations germane to the molecular line transfer problem when using an AMC method.

in Appendix D. Currently, the following parameters are used in `molecularLoop`: `tolerance`, `usedust`, `amr1d`, `amr3d`, `debug`, `restart`, `isinlte`, `quasi`, `doNgstep`. Their individual action is described where necessary.

The `molecularLoop` subroutine performs various 'house-keeping' tasks peripheral to the main calculation but nevertheless essential to the successful execution of the code, such as determining the minimum number of levels to be converged to the user-specified tolerance, opening file handles for output data and, if the code is being run in parallel, handling the MPI communication between threads. Specifically, there are two important events that must be coordinated at the start of the calculation. First, each MPI thread must have the same seed for the random number generator in order that the fixed-ray stage of the calculation run as intended - this means that it must be inherited by all MPI threads at the start of the calculation and kept synchronised throughout the first stage. Secondly, in order to attain the nearly-linear calculation speed increase afforded by the Monte Carlo method, it is necessary to ensure that the computational burden is shared equally over each thread. This is most simply done by allocating an equal share (N_{octals}/N_{proc}) of the octals in the grid to each thread. For this reason, an array of pointers to each cell, `octalarray` is constructed and broadcast to each thread along with the start and end indices of

the octals under consideration for each thread.

This method is simple but does not take into consideration the relative complexity of the subsequent calculation of the level populations for the cells in each thread; it is possible to contrive a scenario where one thread might have large quiescent cells at the grid edge where there is little work to be done and finish its workload relatively quickly whereas another thread consisting entirely of cells covering an optically thick turbulent area of the grid will take longer to complete its calculations.

In TORUS, to keep the code as broadly applicable to as many geometries as possible, storing the octal in this way (a flat array) will remain the method for the foreseeable future; however, it is possible to improve the scalability of the code by creating a worker thread that allocates smaller fractions of the grid to each thread and then, upon completion of that 'workunit' by a thread, 'hands out' more. Using this mechanism a thread only becomes idle when there are no more workunits to hand out. Alternative methods of decomposing the problem are possible; a common alternative is to split the grid so that no spatial domain is duplicated in memory. This is common in hydrodynamics calculations but is undesirable in a ray-based code because each time the ray crosses a domain boundary its properties must be communicated between threads. This method may ultimately become necessary if grids exceeding the available RAM of an individual computing node are to be considered.

Once each thread has its allotted octals to work on the calculation of the non-LTE level populations in each octal proceeds, in serial, following the flowchart given in Figure 2.3. I present an overview of the loop below and explain the mechanism of action for each subroutine in depth in the relevant subsections.

First, a Monte Carlo sample of the external radiation field at all the requisite line transition frequencies is obtained by repeating the `getray` subroutine `nray` times using different initial conditions (see 2.2.2 for details). This information is passed into `calculateJbar` which outputs the mean radiation field felt within the cell to `solveLevels` which in turn generates the non-LTE level population estimate which alters the mean radiation field. `calculateJbar` and `solveLevels` are iterated until the relative RMS change in the level populations between successive iterations is less than 10^{-10}.

If the user has elected to use Ng acceleration, every fifth sub-iteration will be an extrapolation based on Ng's vector sequence acceleration method. This can help to drastically reduce the number of sub-iterations required to reach the tolerance level. Details of this method can be found in Section 2.4.3.

Once each cell has converged or reached a maximum number of sub-iterations, each thread broadcasts its updated level populations and the global convergence properties of the grid are tested in `calculateConvergence` (discussed in Section 2.4.1). As with the sub-iteration acceleration, every fifth iteration may also be accelerated using the same technique. Although the technique permits every fourth iteration to be accelerated, in practice I have determined that the first two iterations after the acceleration show an increase in the RMS convergence parameter and that at least three more iterations are required to notice the benefit of the acceleration.

If the grid is determined to be insufficently converged when compared to the user-specified tolerance then `molecularLoop` starts a new iteration with the updated level populations in each cell. It is also at this stage that any convergence data or diagnostic data is written, along with a part-converged grid (so that the calculation may be restarted if interrupted). At the end of each iteration, TORUS can output a VTK file which can be used to visualise the current status of the grid. The VTK file contains normalised relative level populations and convergence information for each cell and the number of iterations it took to solve equations 2.12 and 1.25 in the previous iteration. This diagnostic information can be useful to determine if there are pockets of slower convergence in an otherwise converged grid. This data can be useful for analysing the individual convergence characteristics of each cell as opposed to the summary statistics provided by the code at this stage. A full discussion of the convergence criteria used in the calculation is given in section 2.4.

Once the grid is determined to be converged it is written to disk and the TORUS will either exit or proceed to calculate synthetic maps of the grid in any user-specified molecular transition.

2.2.2 Determining the external radiation field

The `getray` subroutine stores the specific intensity along the ray path for each transition in the array `i(1:maxtrans)`, the optical depth attained by each transition as the grid is traversed in the `tau(1:maxtrans)` array as well as scalar `phi` (ray weight) and `ds` (path length through current cell) in global arrays defined in `molecularLoop` for each ray for use by `calculateJbar` to find the local radiation field, $\bar{J}_{\rm ul}$ in the cell.

While determining the level populations, TORUS spends an increasingly large fraction of computational effort calculating the external radiation field depending on the number of rays used to sample it – in fact, the computational burden can be in

excess of 95% of the runtime as the number of rays exceeds 10^5. Significant effort was devoted to optimising this function.

The `getray` subroutine takes as primary inputs, a particular grid cell, position within that cell, $\mathbf{p(x)}$, and a direction, $\mathbf{d}$, from which the ray will emanate as well as the velocity shift from the line centre (in fractions of the turbulent line width). The particular grid cell is simply an element in the **octalArray** array. A position and direction is chosen based on the user's choice of whether to use the quasi-random number generator or not (controlled by the `quasi` parameter). Either way, a 'random' position within the cell is obtained, appropriate for the dimensionality of the calculation. In the one– and two-dimensional cases a uniform radius (and height) within the cell is chosen in the plane and then the position vector is rotated such that it lies within the cell projected by some random angle (or pair of angles as appropriate). In three Cartesian dimensions, the random position is trivial to obtain.

Currently, the direction is described by a random unit vector created so that no direction on the unit sphere is more likely than any other - although this may change if it is decided to implement weighted spatial sampling. The Cartesian unit direction vector $(u, v, w)^T$ is constructed from a pair of uniform random deviates $\{r_1, r_2\} \in [0, 1)$ according to the equations shown in Figure 2.4.

Finally, the velocity shift of the ray from the line centre is determined. Two effects must be considered. The first is the systematic velocity field at $\mathbf{x}$, which will Doppler shift the line function by $\mathbf{v}(\mathbf{x}) \cdot \mathbf{d}$. The shift is proportional to the cosine of the angle that the ray makes with the local velocity vector, $\mathbf{v}(\mathbf{x})$. TORUS accepts an analytical function for evaluation of $\mathbf{v}(\mathbf{x})$. However, in the absence of one, or if the velocity is computationally expensive to calculate, it is possible to quadratically interpolate over values stored at the grid cell corners.

Secondly, the frequency shift caused by Doppler broadening of the line profile caused by microturbulent (thermal and non-thermal) motions of the molecules must be accounted for. Noting the relationship between (non-relativistic) velocity and frequency (Equation 1.28) the Maxwellian distribution of velocities of the molecules relative to a stationary observer is transformed into a distribution of frequencies centred about the transition rest frequency ν_0. This quantifies the broadening as a function of the known turbulent velocity field; the so-called *line profile function* which is given here again:

$$\phi_\nu = \frac{c}{v_{\mathrm{turb}} \nu_0 \sqrt{\pi}} \exp\left(-\frac{\Delta v^2}{v_{\mathrm{turb}}^2}\right). \tag{2.1}$$

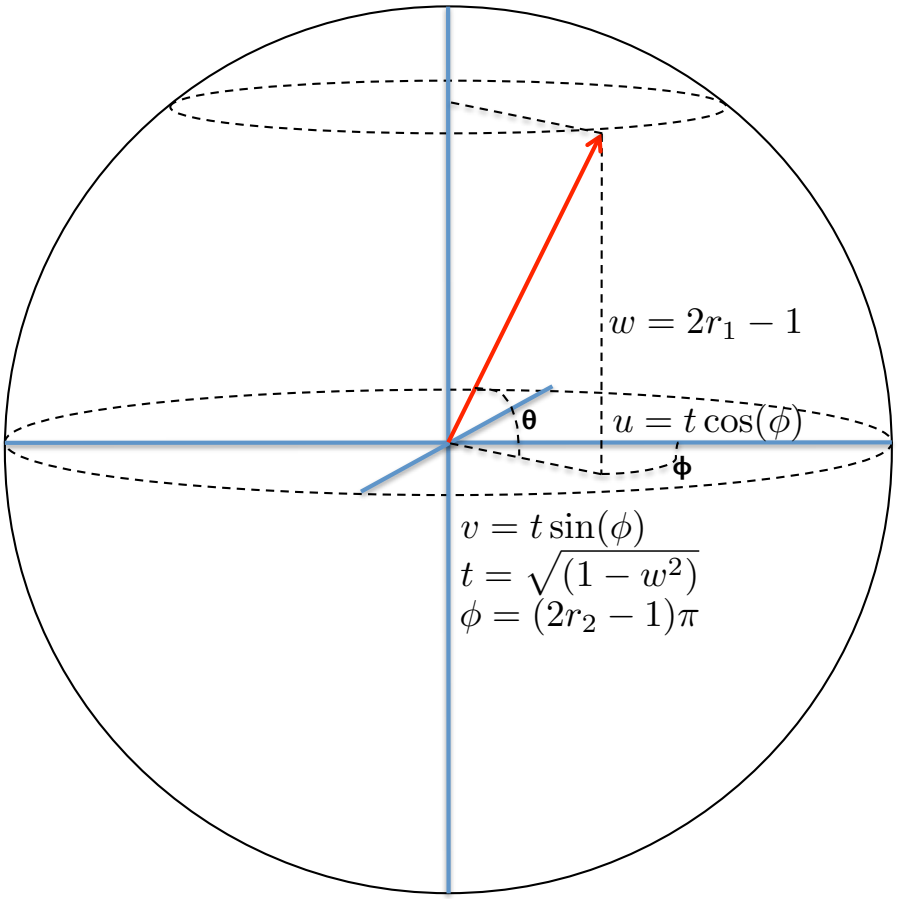


Figure 2.4: The random unit vector is efficiently derived from two uniform random deviates using the equations shown in the figure.

where $v_{\rm turb}$ is calculated as in equation 1.35.

Observers will note that this definition is less easily determined than the full-width at half maximum (FWHM) of an observed line profile. Assuming that the line profile is gaussian then $v_{\rm turb}$ is related to the FWHM by the relationship $v_{\rm turb} = {\rm FWHM}/2\sqrt{\ln(2)}$.

Thus, each frequency has a weighting depending on its likelihood of occurring, ϕ_ν. Because the line optical depth is greater in the line centre, the effective path length over which information can travel ($\propto 1/\tau^2$) is reduced compared to the line wings and it is computationally efficient to devote as many photons to the line wings as to the line centre. Consequently, a variance reduction technique is employed to reduce the total number of rays required to sample the radiation field. Instead of sampling the ray frequencies from a gaussian distribution, in practice, the deviate is picked from a uniform distribution with a width of 4.3 turbulent line widths, $v_{\rm turb}\nu_0/c$, centred on the rest frequency for the transition, ν_0. This ensures good sampling in the optically thin line wings.

Having picked a random frequency for the ray, the total velocity shift, Δv, is the sum of these two effects:

$$\Delta v = c\frac{\nu - \nu_0}{\nu_0} + \mathbf{v} \cdot \mathbf{d}. \tag{2.2}$$

The subroutine's primary role is to return an array of intensities incident on the cell for each transition being considered, `i(1:maxtrans)`. This natural vectorisation is one of the most valuable design choices in the code. By considering all transitions at once, rather than looping over each transition individually, the grid need only be traversed once per transition. Furthermore, a secondary but not negligible speedup is obtained by vectorising the calculations along the ray; i.e. it is relatively quicker by using SIMD processor instructions (single instruction multiple data) to find the exponential of an array of, say, 50 values than it is to perform 50 separate exponentials. This is discussed in greater detail in section 2.5.

Once the position, direction and ray frequency shift are known it is necessary to find the distance to the cell boundary along the ray direction, `ds`. This kind of ray-surface 'collision' detection can be very time-consuming but is unavoidable. In the three-dimensional Cartesian case, the algorithm is quite simple, but, paradoxically, the mathematical burden becomes greater in lower dimensions necessitating the solution of multiple trigonometric quadratic equations.

Having found `ds` and `phi` for the ray, all that remains is to integrate the specific intensity in each transition, $I_{\rm ul}$, over all line segments in a piecewise manner along the ray direction towards the edge of the grid.

The bulk of the computational time is usually taken up by traversing the grid, i.e. finding the grid cell that contains the current position along the ray. In theory, given the tree-like structure in which the octals are stored, up to half the time the current position will be in the same parent octal as the previous position but lie within a different subcell - there is very little penalty for this in terms of tree-traversal. However, the advantage of this tree-like structure becomes a distinct disadvantage when crossing more fundamental boundaries. This is because spatially adjacent grid cells are not contiguous in memory.

All quantities on the grid are assumed to be constant within a cell except for the systematic velocity vector, $\mathbf{v(x)}$, which enters both the gas emission and absorption coefficients through the line function, ϕ_ν:

$$j_{\rm ul}^{\rm gas} = \frac{h\nu_0}{4\pi}(n_u A_{ul})\phi_\nu \tag{2.3}$$

$$\alpha_{\rm ul}^{\rm gas} = \frac{h\nu_0}{4\pi}(n_l B_{lu} - n_u B_{ul})\phi_\nu. \tag{2.4}$$

As previously stated, A_{ul}, B_{ul} and B_{lu} are the Einstein coefficients of spontaneous emission, stimulated emission and stimulated absorption associated with the transition from level u to level l respectively; similarly n_u and n_l are the respective number densities of molecules populating those levels.

In order to determine the contribution of the cell to the ray it is necessary to calculate the source function, S_ν, in the cell. If not considering dust then equation 1.16 reduces to

$$S_{ul}^{\rm gas} = \frac{j_{ul}^{\rm gas}}{\alpha_{ul}^{\rm gas}} = \frac{n_u A_{ul}}{n_l B_{lu} - n_u B_{ul}} \tag{2.5}$$

by cancelling the line profile function in both terms. Otherwise one must determine $\alpha_\nu^{\rm dust}$ and $j_\nu^{\rm dust}$, where ν is the frequency associated with the particular line transition u to l and calculate the ratio of emission and absorption from the combined gas and dust terms. This is handled in TORUS by the `returnKappa` subroutine which when scaled by the dust density returns the total opacity due to absorption for the particular dust model being used (scattering is negligible at rotational frequencies). The emission by the dust in the cell is then given by assuming all absorbed radiation is re-radiated as a blackbody at the dust temperature, $T_{\rm dust}$. This is calculated in the cell using equations 1.18 and 1.19.

It is also necessary to calculate the transition specific optical depth through the cell. The absorption is then weighted by line profile (equation 1.34) depending on difference in ray velocity and velocity within the cell at the current position. The differential optical depth within the cell for a transition at frequency, ν, is then given by

$$\Delta\tau_\nu = \alpha_\nu {\rm d}s, \tag{2.6}$$

where α without superscripts is the combined gas and dust absorption coefficient. This optical depth attenuates the ray by a factor of $e^{-\Delta\tau}$.

The line absorption can vary strongly with velocity gradient across a cell when the velocity gradient is large compared to the local magnitude of the turbulent velocity field. Variations in ϕ_ν are accounted for by subdividing the integration into smaller steps. To ensure good resolution of the velocity structure within a cell, we use the condition

$$n_{\rm split} = \max\left(2, \lfloor 5\frac{|\mathbf{v}_e - \mathbf{v}_s|}{v_{\rm turb}}\rfloor\right), \tag{2.7}$$

where $\mathbf{v}_e$ and $\mathbf{v}_s$ are the velocities at the cell boundaries where the ray intersects the cell. Often, there is very little velocity gradient across a cell regardless of direction. If, during the fixed ray stage $n_{\rm split}$ never exceeds 2 then the cell is determined to be quiescent and $n_{\rm split}$ is fixed at 2 for the cell for the rest of the calculation. By

pre-determining this in the relatively quick fixed-ray stage, it is possible to reduce by two-thirds the number of calls to the velocity subroutine.

The contribution of the cell to the total ray intensity in the transition from u to l, ΔI_ν, is therefore by integrating equation 1.15:

$$\Delta I_\nu = \left(1 - e^{-\Delta\tau}\right) S_\nu, \tag{2.8}$$

where

$$\Delta\tau_\nu = \alpha_\nu \mathrm{dT} \tag{2.9}$$

and dT is the differential distance that the ray has been subdivided into within the cell. Different notation for ray pathlength has been used for clarity; *ds* is only used to describe the initial pathlength of the ray within the originating cell, not outside the cell.

In the code, two running counters keep track of total attenuated intensity and optical depth for each transition as the ray is tracked along its path:

$$\tau_\nu = \tau_\nu + \Delta\tau \tag{2.10}$$

$$I_\nu = I_\nu + e^{-\tau_\nu} \Delta I_\nu. \tag{2.11}$$

The current position along the ray is advanced to the next cell by adding on the distance as calculated in `distanceToCellBoundary` and the calculation is repeated until the ray reaches the edge of the grid.

By default, the code assumes that the simulated region is surrounded by a vacuum pervaded by cosmic microwave background (CMB) radiation; consequently, at the edge of the grid, the attenuated contribution of the CMB, $\mathrm{B}_\nu(T_{\mathrm{CMB}})$, is added to the intensity counter.

The code fragment that tracks I_ν (for brevity, the gas-only scenario) is included in Appendix B.

2.2.3 Determining $\bar{J}$ and solving the equations of statistical equilibrium

Given a set of {`phi`, `ds`, `i(1:maxtrans)`} for each of the rays from `getray` and the most recent estimate of the level populations, `npops`, the `calculateJbar` subroutine calculates a Monte Carlo estimate for $\bar{J}_\nu$ by summing the two contributions to the field. Mathematically, this is determined by averaging the contribution of samples

of the specific intensity incident at a point in the cell over 4π steradians. Thus the equation

$$J_\nu = \frac{1}{4\pi}\int I_\nu d\Omega,$$

becomes

$$\bar{J}_\nu = J_\nu^{\rm ext} + J_\nu^{\rm int} = \frac{\sum_i^{\rm all\ rays} I_\nu^i e^{-\tau_\nu^i}\phi_\nu^i}{\sum_i^{\rm all\ rays_i}\phi_\nu^i} + \frac{\sum_i^{\rm all\ rays} S_\nu[1-e^{-\tau_\nu^i}]\phi_\nu^i}{\sum_i^{\rm all\ rays}\phi_\nu^i}. \quad (2.12)$$

Equation 2.12 highlights the necessity of knowing `ds` and hence τ inside the cell. Unfortunately, as the level populations are iterated to a solution consistent with the radiation field, the optical depth within the cell will change and consequently both the local and external contributions will change. The line profile weights, their sum and the external intensities remain constant until the next `getray` iteration for the cell and so do not need to be recomputed. Thus, compared to `getray`, the `calculateJbar` subroutine is quick to execute even when summing over thousands of rays.

The newly updated $\bar{J}_\nu$ data are used to update the level populations by solving the statistical equilibrum system of equations.

The `solveLevels` subroutine

In order to obtain the relative fractional level populations, $\mathbf{n}_i$, for a cell it is necessary to solve a system of equations describing the statistical equilibrium that is to be attained where the rates of excitation and de-excitation to and from each level reach dynamic equilibrium - the so-called equations of detailed balance, viz.,

$$n_i\left[\sum_{i>j}A_{ij} + \sum_{i\neq j}(B_{ij}J_{\rm ij} + C_{ij})\right] = \sum_{i<j} n_j A_{ji} + \sum_{i\neq j} n_j(B_{ji}J_{\rm ij} + C_{ji}), \quad (2.13)$$

where $\bar{J}_\nu$ is input from `calculateJbar` and C_{ij} is the rate coefficient of collision from level i to j for the molecule and a particular collision partner at a specific temperature. To obtain a total collision rate at a specific temperature, linear interpolation is performed on the tabulated data described in Section 2.1.1 and the collision rates per unit volume are scaled by the collision partner density, normally some fraction of n(H_2), to give the rate for that unique collision. Each collision rate is summed linearly to obtain the total rate.

Collision rates are only stored for de-excitations, however excitation rates are trivial to calculate. The excitation rate can be obtained from the de-excitation rate using the Boltzmann distribution as follows:

$$C_{ji} = C_{ij}\frac{g_i}{g_j}e^{-|E_i - E_j|/kT} \tag{2.14}$$

Neither the temperature nor the density in a cell changes at any time during the non-LTE calculation, so the elements of the collision matrix are calculated in advance (even if not stored owing to the large memory requirements).

In addition to the system of equations (1.25) an additional constraint is imposed:

$$\sum_{i}^{\text{all levels}} n_i = 1, \tag{2.15}$$

to ensure that the relative level populations are normalised in each cell. In order to recover the actual number of molecules in any given state, the level populations may be multiplied by the molecular number density, `molAbundance`.

In practice, the system of equations is represented as a matrix equation where each element, $\mathbf{M}_{ij}$, represents the net rate of emission (radiative and collisional) from level i to level j. In the `solveLevels` subroutine, the matrix of radiative rates is constructed and added to the matrix of collisional rates. The radiative matrix is comprised of only tri-diagonal terms due to quantum selection rules. Consequently, only collision terms populate matrix elements that lie off the diagonal or its neighbouring elements. The constructed matrix is illustrated below:

$$M_{ij} = \begin{pmatrix} B_{01}J_{10} + \Sigma C_{0i} & -B_{10}J_{10} - A_{10} - C_{10} & -C_{20} & \cdots & -C_{n0} & 0 \\ -B_{01}J_{10} - C_{01} & B_{10}J_{10} + A_{10} + B_{12}J_{21} + \Sigma C_{1i} & B_{21}J_{21} - A_{21} - C_{21} & \cdots & -C_{n1} & 0 \\ -C_{02} & -B_{12}J_{21} - C_{12} & B_{21}J_{21} + A_{21} + B_{23}J_{32} + \Sigma C_{2i} & \cdots & -C_{n2} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -C_{0n} & -C_{1n} & -C_{2n} & \cdots & B_{n,n-1}J_{n,n-1} + A_{n,n-1} + \Sigma C_{n,i} & 0 \\ 1 & 1 & 1 & \cdots & 1 & 0 \end{pmatrix}$$

Each row represents one of the equations of statistical equilibrium (Equation 1.25), where the diagonal term represents the total emission out of a level and the off-diagonal terms represent emission in to the level.

The resultant system of $\mathrm{N}_{\mathrm{maxlevel}} + 1$ linear equations,

$$\mathbf{Mn} = (0, 0, ..., 1)^T, \tag{2.16}$$

is readily solved using a matrix method. In `solveLevels`, the system of equations is solved using LU-decomposition. This method is easy to implement, numerically

stable and moderately efficient (see Press, Teukolsky, Vetterling, & Flannery (1992) for details).

2.3 Creating synthetic maps

As well as determining the non-LTE level populations for a geometry, TORUS can create three-dimensional, velocity-resolved spatial maps of the emergent intensity (hereinafter described as *datacubes*) from an AMR grid containing molecular level data for each cell. TORUS generates datacubes in a similar way to that in which it calculates the intensity in a cell, except that when creating datacubes the point from which the ray emanates lies outside the grid, on an image plane centred on the hypothetical observer's position and the frequency/transition of the observation is chosen directly.

A datacube is comprised of many velocity channels which the observer may scan through in order to create images of intensity at each corresponding frequency. This simulated three-dimensional dataset is similar in structure to 'real' observations but free from instrument noise, bias and point spread function. When comparing theory and observation it is not sufficient to determine the level populations, temperature or H_2 density profiles of an astrophysical object. Observers must make do with an image projected upon the sky from which they can infer the properties of the object. By convolving the synthetic map with an instrument function, observational data can be compared with synthetic data, enabling the salient parameters of an object to be quantified by recovering the model parameters that best fit the observational data through a fitting process.

Each datacube contains a wealth of information. By integrating over the velocity channels when considering line molecular data it is possible (using an optically thin tracer) to estimate the column density, as would be done for real observations. By integrating over the spatial dimensions it is possible to obtain information on the velocity structure of the object. TORUS produces synthetic observations as a datacube of intensities; conversion to flux by beam convolution can be achieved using one of the many tools that exist that will emulate real instruments, e.g. CASA[1] for ALMA, or perform arithmetic on datacubes like MIRIAD (Sault et al. 1995), CIAO (Fruscione et al. 2006) or IDL.

In this section I describe the routines that are called when generating a datacube. The process is much more linear than determining the level populations and consequently I present the subroutines that I wrote to create the synthetic maps in order

[1] http://casa.nrao.edu/

of highest to most elemental function (amplifying as appropriate where it is obvious that a loop will be necessary, e.g. determining the intensity in each pixel).

The first step is to create a hypothetical observer and the corresponding image plane (Section 2.3.1) then to divide the image into individual elements, or pixels, that represent a discrete sample of the intensity that would be incident on that region of the image plane (Section 2.3.2). The final stage is the population of each pixel with the appropriate values (Section 2.3.3).

2.3.1 Defining the image plane

When creating synthetic maps of an object the first issue is the definition of the image plane. The plane onto which the image is projected is defined by the vector passing through the point of observation and an arbitrary point inside the grid (usually the centre). This is handled by the `setObserverVectors` subroutine.

The `setObserverVectors` subroutine

Observer coordinates can be specified in TORUS in two ways. In the first mode the observer coordinates are input directly from a file called `observerpositions.dat`. The file contains the normalised direction of the observer from the centre of the grid. Orthonormal image basis vectors, $\mathbf{b}_1$ & $\mathbf{b}_2$ are chosen such that one is perpendicular to $\hat{\mathbf{z}}$ (in the case where the observer is looking along $\hat{\mathbf{z}}$, the bases are chosen as $\pm\hat{\mathbf{x}}$ and $\pm\hat{\mathbf{y}}$). Using this method it is possible to increment the `observerpos` parameter in `parameters.dat` to create a whole series of maps with minimal effort.

The second way is to specify up to three angles through which an observer initially at -$\hat{\mathbf{y}}$ should be rotated. The horizontal and vertical image basis vectors are chosen as $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$ respectively and are fixed relative to the rotated observer position.

It is not a requirement that a map be centred on the grid centre. An offset may be applied if the user is interested in a feature elsewhere in the grid. A zoom effect can be obtained by reducing the image size while maintaining the resolution and picking a different coordinate in the grid to centre on.

The default assumption in the code is that the observer is far away from the object being imaged and that it is appropriate to sample the intensity in the image by using a set of paraxial rays normal to the plane. Douglas et al. (2010) have extended my original code to render images in solid angle (as opposed to rectilinear coordinates) in the near field where the observer is close to or actually inside the object, in their case, simulating the spiral structure of a galaxy in HI from within the galaxy itself.

The `createImage` subroutine

Once the image plane is defined, a map is parameterised by its extent, resolution and the frequency range of the observation to simulated. If the user wishes to simulate an observation in a particular molecular transition (by setting `lineimage` to *true* in `parameters.dat`) then they must supply a grid with the appropriate molecular data. They must also specify a particular transition for which they they wish to calculate the emergent intensity (the `itrans` parameter corresponding to the transition number in the molecular data file) and the range of velocities they wish to scan over, determined by `maxvel`. If the user wishes to create a continuum map of dust emission/absorption at a particular wavelength then they must instead set `lamline` in `parameters.dat` to the required wavelength in microns.

Datacubes of any size may be created, limited only by the amount of memory and the spatial resolution of the original AMR grid. Each image is subdivided into $N_{\rm pixel}^2$ square pixels of equal area so that the intensity per pixel may be stored in the datacube structure.The number of pixels per row/column of the image is controlled by `npixel` and the number of velocity channels by `nv` in parameters. For a typical datacube, the memory footprint can be as much as 400 MB (4 bytes per pixel $\times$ 1024 pixels $\times$ 1024 pixels $\times$ 100 velocity channels) so this must be taken into account along with the size of the grid when assessing the desired resolution of the datacube.

The datacube is stored as a derived type, `cube`, containing arrays of the intensity and optical depth (if required) attained for each velocity channel as well as the actual column density. The axis labels and quantity units are also stored so that the quantities may be scaled at a later time.

Like the level population code, the datacube creation routines also benefit from parallelisation by MPI. Each image is decomposed into strips which can be computed independently and summed together once all threads have computed the intensities in the pixels in their region. Each strip has an area of $N_{\rm pixels} \times N_{\rm pixels}/N_{\rm proc}$. Currently, no load balancing is performed but because of the magnitude of the problem compared to the determination of the level populations there would be little practical reduction in time taken by including this functionality.

The final task before calculating the intensities in each pixel is to pre-compute the emission and absorption (and hence the source function) in each grid cell using the level populations, which are known in advance. This step can be omitted if the `lowmemory` flag is set in `parameters.dat`. This trade-off can result in significantly slower computation times however. As in `allocateMolecularLevels` the reciprocal of the microturbulent velocity is also calculated to save time.

This subroutine then calls the `makeImageGrid` subroutine to populate a temporary intensity array for each velocity channel. The calls are looped using as index variable, `iv`, corresponding to a specific Doppler shift for each velocity channel stored in the datacube, $\Delta v = \frac{\texttt{cube\%vAxis(iv)}}{c}$. It is also necessary to pass the column indices that each MPI thread should process.

Once the intensity array has been populated by each thread (using a call to the MPI library routine `MPI_ALLREDUCE` to combine each thread's contribution) for a particular velocity channel, the values are saved to the `cube` structure as `single` precision real variables to save memory. The extra intermediate precision is unnecessary as the numerical errors in determining the intensities will be far greater than the round-off error.

Once the loop has been completed, the final datacube is written to disk in a standard FITS format using routines provided by the CFITSIO library (Pence 1992). At this stage, the labels stored in the structure are added to the World Coordinate System (WCS) stored in the FITS file so that the scale of the output file is known by the FITS reading tool. By default, intensity has units of erg s^{-1} cm^{-2} Hz^{-1} sr^{-1} and linear scale in 10^{10} cm and velocity in km s^{-1}. As a standard file format it is then possible to view, analyse and manipulate the file using familiar observational tools, e.g. DS9, GAIA, KVIS etc.

2.3.2 Populating the datacube

The `makeImageGrid` and `pixelIntensity` subroutine

The function of the `makeImageGrid` subroutine is to pass the pixel coordinates to a worker subroutine `pixelIntensity` and to populate the corresponding element in the array with the returned value. Each pixel is described by its linear extent and the physical position on the image plane of its top-left corner.

The image is stored so that the intensity in the bottom-left corner of the image is stored in pixel (0,0). The unit image basis vectors are now scaled so that one unit corresponds to one pixel length ($|\mathbf{b}| = N_{\rm pixel}|\mathbf{b}|$). Thus, considering the observer position as the origin, the top-left pixel in the grid (where the imaging process starts) is located at $\mathbf{O} + 0.5N_{\rm pixel}(\mathbf{b}_2 - \mathbf{b}_1)$. This is illustrated in Figure 2.5. Note that a further shift along $\mathbf{b}_1$ is required for MPI threads to ensure that they start in the correct pixel column.

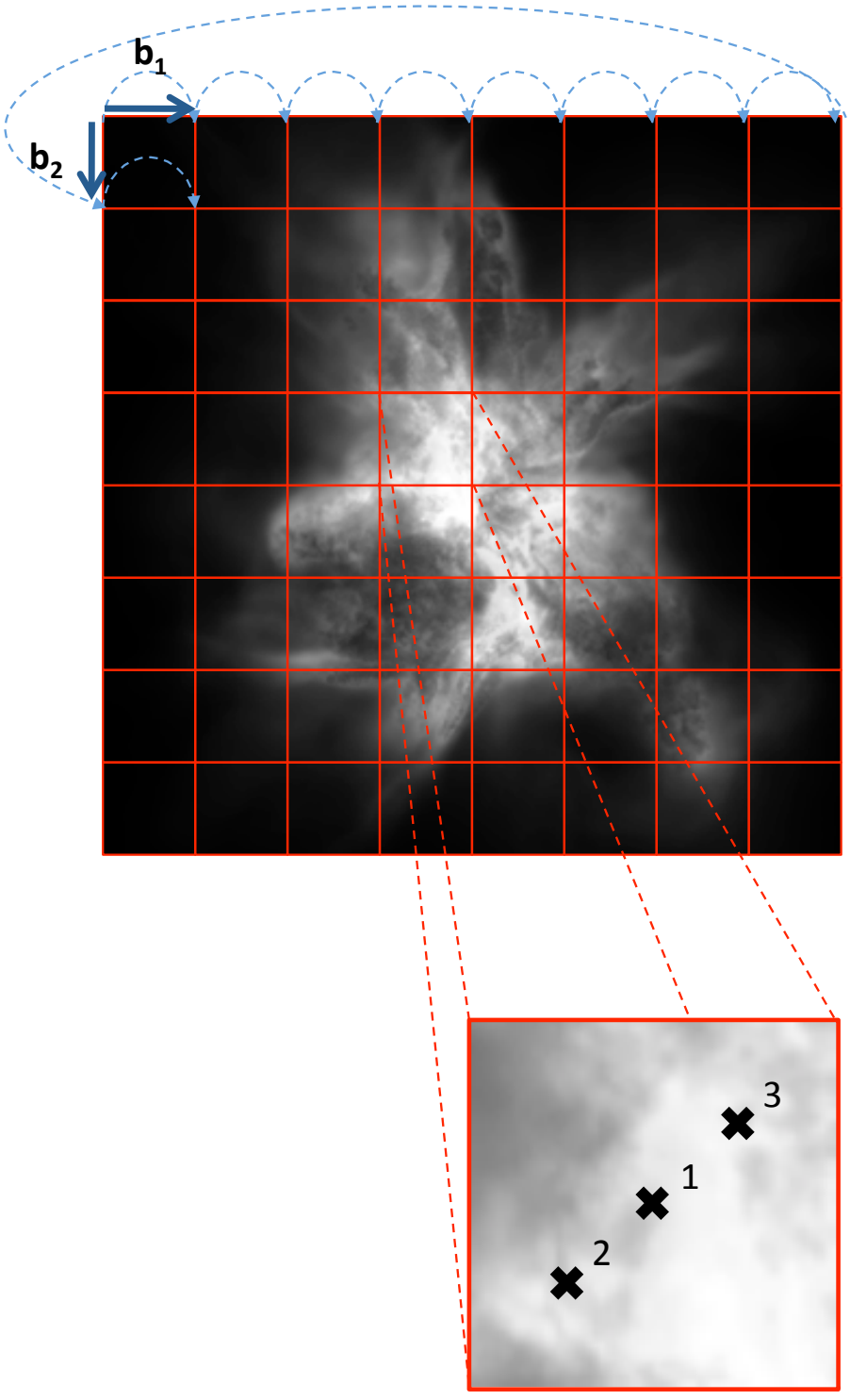


Figure 2.5: `makeImageGrid` controls where each pixel intersects the grid. The image is created by rastering along each row by a pixel size and calling `pixelIntensity` to find the Monte Carlo average of the intensities incident on the image plane within the pixel using one or more paraxial rays.

Once the pixel position in the image has been determined, control is handed over to the worker routine, `pixelIntensity`, which returns the average incident intensity over the area covered by the pixel.

In the simplest, fastest mode of operation, the intensity in a pixel is assumed to equal that of a single ray emanating from the centre of the pixel towards the grid along the viewing vector multiplied by the area of the pixel. The resolution of the image will usually be considerably less than that of the grid being imaged, i.e. the width of a pixel will be greater than that of the smallest subcells contained within the grid. This inevitably leaves many grid cells entirely unsampled (and likely *all* cells undersampled if there is a large velocity gradient across a cell in the direction of the observer) and it is possible that features on the order of one pixel width or smaller will be missed entirely. This necessarily means that some information about the structure will be lost.

TORUS utilises *sub-pixel sampling* to ameliorate this effect, by keeping a running average of the intensity sampled over quasi-randomly allocated rays bound within the pixel:

$$I^{avg}_{new} = \frac{(n-1)I^{avg}_{prev} + I_n}{n} \tag{2.17}$$

$$\sigma^2_{new} = \frac{(n-1)\sigma^2_{prev} + (I_n - I^{avg}_{new})(I_n - I^{avg}_{prev})}{n} \tag{2.18}$$

until the normalised standard error, is less than some user-specified tolerance, `tol`, or until the number of rays exceeds a pre-specified maximum:

$$\frac{\sigma_{new}/I^{avg}_{new}}{\sqrt{n}} < \texttt{tol} \quad \text{or computationally,} \tag{2.19}$$

$$\sigma^2_{new} < n(\texttt{tol} \times I^{avg}_{new})^2. \tag{2.20}$$

The advantage of the second representation being that the variance, σ^2_{new} is calculated directly (Equation 2.18) and that no square-roots or divisions need be performed.

The origin of the ray inside the pixel is chosen using a Sobol' quasi-random number generator that is reset for each new pixel. The 2-dimensional quasi-random sequence is initialised and the output is used immediately; the first quasi-random pair being (0.5,0.5). These are the fractions of each basis vector added to the top-left corner used to define the pixel's position in the image. If `nsubpixels` is set to 1 then no further rays are used and this recovers the simple mode described above.

The advantages over pseudo-randomly generated ray-origins are two-fold; the ray-origins are created so that they avoid each other as best possible within the confines

of the pixel thereby avoiding clumpiness (mathematically, a property known as *low discrepancy*), providing a truer representation of the range of intensities in the region being sampled. Also, because these low discrepancy sequences are extendable in a pointwise fashion until some criterion is met, they are considerably more flexible than grid-based subsampling techniques that require an exponential increase of samples. A more in-depth discussion of quasi-Monte Carlo methods is given in section 2.4.2.

The final task in `makeImageGrid` is to track the indices of the pixel that are sent to `pixelIntensity`. These are the indices over which `pixelIntensity` loops to populate the region of the image for which a particular thread is responsible for. The thread returns its region to `createImage` and the full image for that velocity bin is stored in the datacube.

2.3.3 The `intensityAlongRay` subroutine

The main part of `intensityAlongRay` is much the same as `getray`, their aims being similar, however this time only one transition is considered and the velocity shift is determined in advance. Furthermore, because the level populations are no longer changing, the absorption and emission are known in advance and can be stored, thereby reducing the computational effort.

In geometries where the density can be evaluated analytically or is stored on the corners of each cell in addition to the cell centres, TORUS uses *density subsampling*. In the iterative level population solver, a ray is subdivided within a cell to ensure good resolution of the velocity field. Analogously, using this technique it is possible to use a more accurate value for the density field at a point along the ray than that stored at the cell centre alone; producing both smoother and more accurate images - especially at lower grid resolutions. Furthermore, whilst errors in determining the intensity from very optically thick regions have little effect upon the level populations in their neighbourhood, they can strongly affect the appearance of an object because they affect the shape of the boundary that can be probed by radiation of a particular wavelength.

`intensityAlongRay` will also return the optical depth, τ, along a ray and the true column density. Both these variables can be stored in the datacube. In the optically thin limit, the intensity should map the column density well and this can then be verified.

An experimental piece of diagnostic code has been added recently. For each velocity bin, the average contribution to the intensity, dI, along each ray that passes through

a cell in the grid is written to disk (in VTK format). This will potentially allow a spatial map of where radiation might be expected to be created for some geometry. Of course, this is anisotropic but averaged over enough angles may provide an interesting theoretical insight into the emission/absorption profiles of complex physical structures.

Finally, `intensityAlongRay` can run in two other modes. The first returns the depth that a ray penetrated before being attenuated by some optical depth in a velocity channel. The second returns either the maximum density attained for a user-specified fraction of the maximum intensity along the ray or the mean density along the ray integrated over all intensities. This facility may be used to explore the relationship between emergent intensity and density. This is discussed in more depth in Chapter 6.

2.4 Convergence and acceleration

At the heart of the stochastic AMC method there must be some definite endpoint where it is determined that further iteration is unnecessary for the level of accuracy required by the user. The random nature of this method should be viewed as a strength and not as a weakness as it becomes possible to readily quantify the numerical errors present in the simulation.

In the absence of an analytic solution to compare against it must be assumed that an iterative solution tends to the true solution for a sufficiently large number of samples. That is to say that the parameter space is well sampled and that no systematic bias is present in the solution. The level of convergence of the calculation therefore can be approximated by the examining the change in the level populations since the previous iteration.

2.4.1 Convergence in `molecularLoop`

When determining the level populations there are two places in the code where it is necessary to quantify the relative difference between two iterations. Once, when iteratively solving equations 1.25 and 2.12 and again when finding the global convergence properties of the calculation. Each instance is examined here and detailed numerical results are presented in Section 3.1.2.

Sub-iteration convergence

For each cell, the system of equations is solved iteratively to obtain a self-consistent solution vector of relative level populations, **n**. TORUS uses a user-defined parameter to specify the point at which successive iterations of this system are determined to have converged. By default, the criterion is set so that the mean-square (MS) error between the two most recent iterations over all but the uppermost level is less than 10^{-12}, i.e.

$$\sum_{i}^{N_{\mathrm{level}-1}} \left(\frac{n_i^{\mathrm{new}} - n_i^{\mathrm{old}}}{n_i^{\mathrm{new}}} \right)^2 < 10^{-12}. \tag{2.21}$$

Typically, the uppermost level is chosen so that its population has no significant impact on the quality of the solution of lower levels. Due to its low population and underestimated rate at which molecules de-excite in the level from above, the uppermost level is often subject to a high degree of noise and its convergence is not representative of the convergence of the other levels. Furthermore, given the very low absolute abundance of molecules in this level round-off error may become significant, although `double precision` variables are used throughout the code to safeguard against this.

The mean-square is used because it is a more stringent test than the sum of the absolute deviates. Large deviations in any level are weighted more strongly and are a sign that more iterations are required to reach proper convergence. Normalised level populations are used instead of absolute populations because each level is dependent on every other so the input from each level into every other is potentially relevant however sparsely populated it is. It is possible that an unacceptably large change in a single sparsely populated level from one iteration to the next may still pass a weaker criterion. Criterion 2.21 ensures that every important level is converged to at least 10^{-6} and many will be converged to a much higher degree.

Global convergence

Once each cell has an updated set of level populations, they are compared with those from the previous iteration. The global convergence of the grid is characterised by a single parameter, ξ, which must be less than the user-specified tolerance for the grid to be defined as converged;

$$\xi = \frac{\max_{\text{all levels}} \left(\sqrt{\sum_{\text{all cells}} \left(\frac{\mathbf{n}^{\mathrm{new}} - \mathbf{n}^{\mathrm{prev}}}{\mathbf{n}^{\mathrm{new}}} \right)^2} \right)}{\sqrt{N_{\mathrm{cells}}}}. \tag{2.22}$$

In the case where all levels and all cells have relative differences of, say, 1% then equation 2.22 will return a value of 1%. If all bar one cell is perfectly converged then the maximum error in any level can be as much as $\xi^{1/2} N_{\text{cells}}^{1/4} \sim 30\%$ for $\xi = 1\%$ and 100 grid cells. Whilst it is not ideal that one cell might be so far from convergence, this extreme case is highly unlikely. However, it does illustrate the fact that the value obtained is the best case for the most poorly converged level. This in itself can be problematic and the most insidious scenario occurs when the sum over all cells is equal for each level and that the error is spread so the error in each cell is all contained within one level. The maximum error may then be as much as $\xi^{1/2} N_{\text{levels}}^{1/2} N_{\text{cells}}^{-1/4}$ in every cell for any particular level.

Obviously, it is crucial to examine the relative change in the final iteration of the converged grid to see where, spatially and in which level, the fractional change is greatest. If the error is concentrated in the least important parts of the grid (perhaps at the boundary of the grid or where the density is very low) then it is safe to assume that the level populations are mainly converged and will not vary much in subsequent iterations. For this reason, TORUS outputs this diagnostic data in VTK format so that it can be visualised in programs like VisIt - an interactive parallel visualisation and graphical analysis tool for viewing scientific data.[2] However, it is the parameter defined in equation 2.22 that defines when the code should move from the fixed-ray stage to the random ray stage or should stop running entirely. Once this criterion is satisfied, one further iteration is performed and if the convergence parameter is still better than the tolerance then the grid is considered to be converged.

One important point that must be stressed is that a converged grid is not necessarily a correct grid. Only comparison with analytic solutions can verify the veracity of a solution. This is discussed in Chapter 3.

2.4.2 Acceleration

Juvela (1997) describes a number of schemes for reducing the variance and accelerating the convergence of Monte Carlo methods. I have implemented low-discrepancy Sobol' sequences (Sobol' 1967, 1976) to optimally cover the direction/frequency parameter space offering the potential to exceed the $N^{-1/2}$ convergence of random Monte Carlo methods (this section). I have also implemented antithetic variates in an attempt to reduce the variance associated with random sampling, the theory being that a ray that is the antithesis of another (that is, has the 'opposite' characteristics) breaks the assumption that the covariance between two rays is, on average, 0 (this section).

[2] https://wci.llnl.gov/codes/visit/

Both methods appear to offer a small gain in convergence. Owing to the random nature of the method and the complexity of even simple benchmark cases it has so far been difficult to show a statistically significant effect, however this work is ongoing.

True spatial weighting to reduce the variance between iterations for the code has not yet been implemented and as such it is possible that the contribution of a small bright grid cell to another cell at some distance will often be missed when the number of rays used per cell to sample the field is low. However, in the scenarios considered in this thesis, it has not been necessary to implement such a strategy.

Significant gains in expediency, however, have been achieved by implementing Ng acceleration (Ng 1974) with weights from Olson, Auer, & Buchler (1986) to improve the convergence properties of the code without any apparent loss of generality (c.f. Hogerheijde & van der Tak 2000), especially in very high optical depth calculations as found in Section 3.3. An outline of the method is presented in Section 2.4.3.

(Quasi-) Monte Carlo methods and low discrepancy sequences

Numerical integration lies at the heart of Monte Carlo methods. By sampling some N-dimensional space it is possible to construct an estimate of the average function value over the space. Most simply, in one-dimension:

$$\int f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i), \tag{2.23}$$

which is readily extended to more dimensions, when x becomes a vector quantity.

The choice of the set of $\{x_i\}$ becomes key to the accuracy with which the integral can be calculated. If the points are chosen such that they are regularly spaced then one recovers the rectangle-rule. If the points are chosen at random (or pseudorandomly) then a Monte Carlo method is being used.

Both methods have their strengths and weaknesses. It is trivial to show that the error between the approximation using the simple rectangle rule and the true integral of the function improves as the square of number of points used to sample the space. However, the method is inflexible if greater accuracy is required; the number of points required to re-sample the space scales exponentially as previous function evaluations cannot, in general, be re-used. Conversely, it is always possible to generate another random number which can be used to reduce the error term.

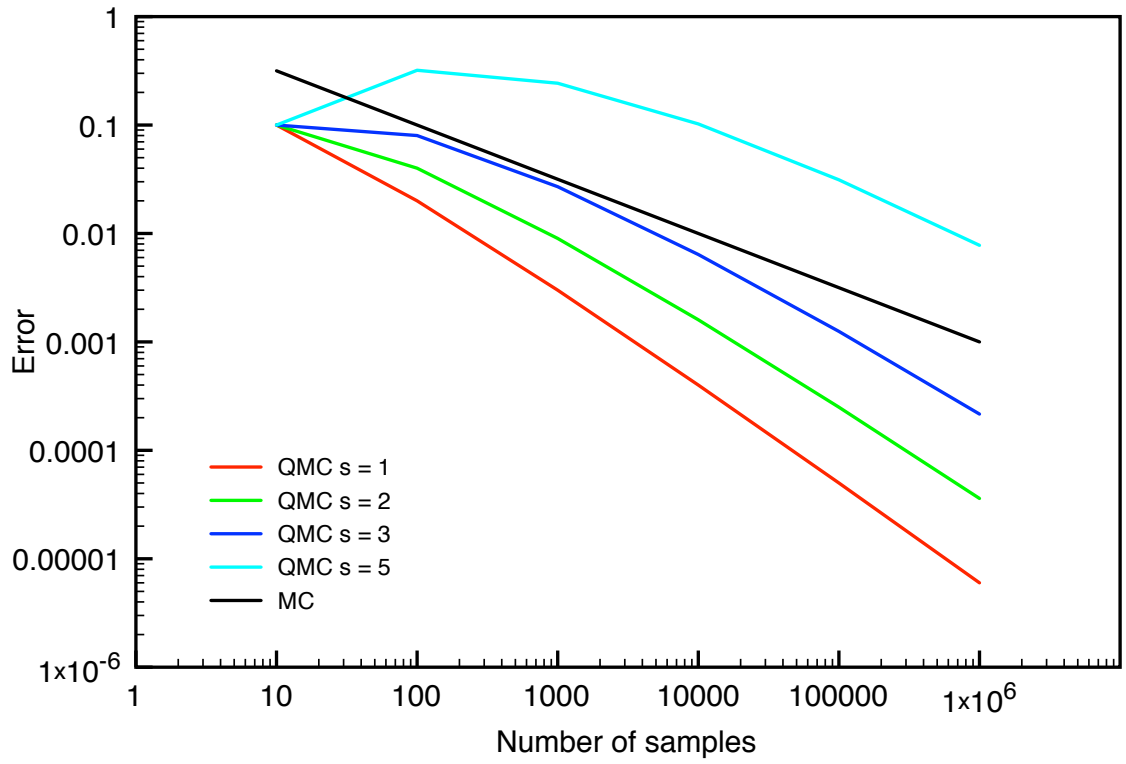


Figure 2.6: Quasi-random integration (QMC) vs. pseudo-random integration (MC)

For this flexibility, asymptotic accuracy is sacrificed; the improvement now goes as $\sqrt{N}$, necessitating many more samples than the regular-sampling method.

A third way, using low discrepancy sequences (or quasi-random sequences) represents a compromise between the two methods, achieving better asymptotic convergence than Monte Carlo whilst maintaining its flexibility.

The discrepancy of a sequence is a measure of how many points fall inside an arbitrary region compared with the average expected for such a region. A 'clumpy' sequence will exhibit high discrepancy whereas a regular grid will exhibit very low discrepancy.

It can be shown, using the Koksma-Hlawka inequality (e.g. Niederreiter & Sloan 1996) that the error is dependent on the product of two terms; one dependent on the smoothness of the function (Morokoff & Caflisch 1994, 1995) and the other on the discrepancy of the set of $\{x_i\}$. Combined, these terms bound the error in the integration from below by a constant times

$$\frac{(\log N)^s}{N}, \tag{2.24}$$

where s is the dimensionality of the integration. Figure 2.6 compares equation

2.24 with $s = 1, 2, 3, 5$ with $1/\sqrt{N}$ convergence. Asymptotically, quasi-Monte Carlo integration outperforms regular Monte Carlo methods for low-dimensional spaces (i.e. $s \leq 3$) for relatively low N but for higher dimensions, this advantage is rapidly lost. However, Morokoff & Caflisch (1995) cite examples of problems in which the advantage of the quasi-Monte Carlo is less than expected theoretically.

In TORUS , the Sobol' sequence is used (Sobol' 1967), following the implementation found in Numerical Recipes (Press et al. 1992). To ensure that the convergence is typically at least as good as regular MC, a maximum of three dimensions are requested at any time. In the non-LTE solver, 2 random uniform deviates are used to generate a direction for a ray and a further one is used for the frequency (see Section 2.2.2). In the image generation, two deviates are used as normalised (x, y) coordinates within a pixel (see Section 2.3.2).

Antithetic variates

The method of antithetic variates is a variance reduction technique that acts to improve the sampling error in a Monte Carlo simulation. The underlying principle of the method is to take, for each sample vector, $\mathbf{x} = \{x_1, ..., x_n\}$, the antithetic path; the very opposite set of variables, $-\mathbf{x} = \{-x_1, ..., -x_n\}$.

Fundamentally, an unbiased estimate of the expected value of an unknown distribution, θ, from which two samples, Y_1 and Y_2, have been drawn will simply be:

$$\hat{\theta} = \frac{Y_1 + Y_2}{2} \tag{2.25}$$

and the variance will be

$$\mathrm{Var}(\hat{\theta}) = \mathrm{Var}(Y_1) + \mathrm{Var}(Y_2) + 2\mathrm{Cov}(Y_1, Y_2)/4. \tag{2.26}$$

Where Y_1 and Y_2 are independent and identically distributed variables (iid), the covariance is 0 and $\mathrm{Var}(Y_1) = \mathrm{Var}(Y_2)$ and so $\mathrm{Var}(\hat{\theta}) = \mathrm{Var}(Y_1)/2$. The method of antithetic variates breaks the assumption of independent and identically distributed (i.i.d.) random variates by seeking a second sample such that $\mathrm{Cov}(Y_1, Y_2) < 0$ thereby reducing the variance of the estimate.

In the simplest case of choosing random variates from a uniform distribution $U(0, 1)$ with a view to determining the integral of some one-dimensional scalar function $f(x)$, the first sample will be x_1. The antithetic sample $x_1' = 1 - x_1$ will be chosen as the second variate. Clearly this is anti-correlated with the first variate and the

Covariance is readily calculated as $\frac{-1}{12}$. Note that this variance reduction is only guaranteed if the function being evaluated is monotonic on the interval of U, otherwise it is possible that the variance may increase.

In TORUS, I have only used the antithetic sampling for the ray-direction generation and not for the ray-frequency generation because the line profile function is non-monotonic over the interval of the random deviates.

2.4.3 Vector sequence acceleration using Ng's method

As with many problems that are solved numerically by repeated function evaluation it is often possible to obtain a more accurate solution more rapidly by employing vector sequence acceleration. Ng acceleration is a popular method for accelerating vector sequences in astrophysics (e.g. Steinacker et al. 2003; Hauschildt & Baron 2006; Daniel & Cernicharo 2008); the original method however, was used to accelerate the numerical solution of the hypernetted chain equation; an equation that occurs in the field of plasma physics. I have implemented and extensively tested a vector acceleration technique developed by Ng (1974) to extrapolate an updated set of relative level populations from the previous four iterations of n_i.

In a complex iterative scheme such as this one where function evaluation can be expensive, we find that these acceleration techniques, used appropriately, not only reduces the time required to find an accurate solution but may, in cases of extremely high optical depth, be used to determine a solution that might not be possible to obtain by fixed iteration at all.

The simplest way to solve a system of equations is to re-write them such that

$$f_{n+1} = Af_n \tag{2.27}$$

where f is a potential solution vector, n is the iteration number and A is a (non–) linear operator, like the Lambda Operator. Starting with an initial vector, f_1, it is then possible to generate a sequence, f_1, f_2, f_3 ... by repeated iteration of equation 2.27. If certain conditions are satisfied regarding diagonal dominance of A, then for sufficiently large n, f_n is indeed a solution of 2.27. However, the fate of such a scheme is not guaranteed and often this naïve method fails to converge and either oscillates and/or diverges around the solution. A better choice is to use the following process

$$f_{n+1} = \omega Af_n + (1-\omega)f_n. \tag{2.28}$$

This is the basis of the *successive over-relaxation method*. However, the optimal choice of the relaxation parameter, ω must be determined empirically. Effectively, for $\omega > 1$, the most recent estimate is assumed to be closer to the true solution than expected and more weight is given to it; the older estimate is trusted less. For appropriate values of ω, convergence can be accelerated considerably. Values of $\omega < 1$ are used to damp oscillations of a process that would otherwise converge slowly towards a solution.

It is noted that, in general, A is not linear and that is especially true in the case of Λ operator in radiative transfer. However, A may be approximated by a linear operator in the neighbourhood of f_n for each iteration step, n. This is similar to the method of false position for solving non-linear equations.

Ng's innovation was to extend this method by taking into account more of the previous iterations and quantifying the relaxation parameters. Thus the form of the solution vector becomes,

$$\bar{f} = (1 - c_1 - c_2)f_n + c_1 f_{n-1} + c_2 f_{n-2} \tag{2.29}$$

where $\bar{f}$ is the extrapolated solution vector. In Ng (1974), Ng derives forms for c_1 and c_2 that minimise the absolute difference between successive iterations $A\bar{f}$ and $\bar{f}$ in the case where A is linear.

One drawback of this method is that, in order to be applied to the global convergence problem, it requires the storage of three previous level populations making it impractical for very large grids as the memory requirements can run into hundreds of megabytes. It is possible to apply the same algorithm to the sub-iteration where it is only necessary to keep the level populations for one cell at a time.

A useful corollary of this method is that because the accelerated solution vector uses previous iterations as basis vectors it assumes the properties of those vectors. Chiefly, this means that because the problem imposes the condition that for any iteration, $\sum \mathbf{n}_i = 1$ then the accelerated vector will also have this property. I have not used Aitken's Δ^2-method or the Shanks transform because they do not satisfy this necessary criterion.

In the study of radiative transfer, Olson, Auer, & Buchler (1986) comment that weighting each level with $W_i = \bar{J}_{ul}^{-1}$ ensures that weak transitions with small J are adequately represented. I have implemented these weights in the code and note that they do indeed lead to acceleration in all cases where they have been used.

2.5 Compilers, profiling and optimisation

As a result of increased processing power and the rise of massively parallel, affordable supercomputing facilities, three-dimensional molecular line radiative transfer at the kinds of resolutions discussed later on in this work has only recently become possible. Consequently, it is important to reduce the computational time where possible. I have therefore dedicated significant time in optimising the code. In this section I give an overview of some of the techniques I have used to speed up the code. I have quantified the effect of some of these optimisations in Chapter 3.

Optimisation can occur at a number of levels, some of which the developer has more control over than others. The most important stage of optimisation should happen before the code is even written, at the design stage. Some algorithms are intrinsically more efficient than others (e.g. quicksort vs. bubblesort or LU matrix reduction vs. Gauss-Jordan matrix reduction). The efficiency of an algorithm is described using so-called 'big O notation' which describes the limiting behaviour of a function when the argument tends towards a particular value. Algorithms that solve a problem in quasi-linear time scale as $n \log(n)$ whereas those that solve in quadratic time scale as n^2. Clearly as n becomes large this exponential increase is highly undesirable and this must be born in mind when writing a piece of code incorporating these algorithms. I have considered this when writing my code and make reference to this principle in the text where appropriate. One must also be aware that it might be necessary to trade algorithmic efficiency for scalability. If an algorithm requires that MPI threads communicate their state to one another regularly then it may perform less in parallel well than an algorithm where each thread can be executed independently.

The next level of optimisation occurs at source code level. Writing good quality code that avoids redundancy, unnecessary loops and avoids where possible 'slow' operations like division or transcendental functions (trigonometric or exponential functions) will naturally be quicker than naïve code. At this level, the developer should look to balance memory against execution time. Whilst developing the `molecular_mod` code I have strived to provide a choice of execution modes, noting that it is often possible to pre-compute or store variables that would otherwise have to be re-calculated repeatedly. The most important optimisations I have made have been those that vectorise code that was previously running in a loop. Not only does this reduce the number of function calls made but also invokes the Vector Math Library (when using the IFORT compiler) which can take advantage of SIMD instructions that efficiently operate on multiple data that have a quicker overall execution time (in CPU ticks) than the equivalent serial instructions.

Optimising a piece of code includes finding bottlenecks. These are critical parts of the code that are frequently executed or where the code spends most of its time. Profiling a code enables the developer to find these 'hotspots' and take steps to optimise them as a priority. By Amdahl's law (Amdahl 1967), optimising these code segments will result in the greatest speed increase. I have used SHARK, a profiling tool available on Mac OS X to profile TORUS at various points in the development cycle to pinpoint parts of the code that might benefit from optimisation. I discuss Shark more in the benchmarking chapter.

Although this optimisation has been undertaken on a single processor machine, good single processor performance tends to lead to good multi-processor performance.

Compilers: portability and optimisation

The final step in the optimisation process is taken at compile time. Use of an optimising compiler ensures that the code is optimised at a lower-level than the previous stages. The compiler must take advantage of the specific architecture that the code will run on.

TORUS has been designed to be as portable as possible. Today, the code runs primarily on 32– and 64-bit LINUX and Mac OS platforms but it has run on many other platforms in the past. The portability of TORUS is achieved by using only FORTRAN features that are part of the current standard. Therefore TORUS can be compiled by any compiler that supports all the features of current standard (FORTRAN 2003) that TORUS uses. Furthermore, TORUS does not need any external libraries (outside those used by the compiler but these may be statically linked if necessary) to perform any task, except the CFITSIO library, which may be included at compile time, if required.

On the Mac OS platform, the list of supported compilers includes GFORTRAN, G95 and NAGFOR as well as IFORT. I have developed the `molecular_mod` on a mid-2007 iMac using a combination of the G95 compiler and the Intel proprietary FORTRAN compiler, IFORT. IFORT can automatically create vectorised code that take advantage of SIMD instruction set extensions like SSE(1-4). Moreover, because Intel tunes its compiler for its own processors it can take advantage of specific details of the processor architecture. The main reason I have used and recommend IFORT over other compilers is its support for high-level optimisation techniques like, for example, inter-procedural optimisation (IPO) and profile-guided optimisation (PGO) and is thus able to produce very fast code. I have extensively tested the combinations of compiler flags that can be safely used to improve the execution time of the code.

I describe the compiler flags I have found to give the fastest stable code in the benchmarking chapter.

Inter-procedural optimisation takes a holistic approach to optimisation by examining the inter-dependence between functions, subroutines and modules. This kind of optimisation is very difficult to do by hand and the details are beyond the scope of this work; the IFORT 11 manual[3] is an excellent reference however. Profile-guided optimisation (PGO) is a two-stage process where the first stage is given to profiling the code at runtime to determine which parts of the code are accessed most frequently. The second stage incorporates the profiling data gathered in the first stage to improve branch prediction etc., leading to fewer cache misses which increase the time taken to perform a calculation. Both IPO and PGO appear to reduce execution time by 10% and $15-20\%$ when combined.

Other reasons for using IFORT are that it is also the primary compiler on *zen*, the University of Exeter supercomputer and that the compiler suite comes with its own MPI library for parallelising the code to take advantage of Monte Carlo methods inherent parallelism and the Math Kernel library, a library of highly optimised math routines that are applicable to many scientific calculations.

[3]http://software.intel.com/sites/products/documentation/hpc/compilerpro/en-us/fortran/lin/main_for_lin.pdf

故曰：知彼知己，百戰不殆；不知彼而知己，
一勝一負；不知彼，不知己，每戰必殆。

'So it is said that if you know your enemies and know yourself, you can win a hundred battles without a single loss.
If you only know yourself, but not your opponent, you may win or may lose.
If you know neither yourself nor your enemy, you will always endanger yourself.'

— Sun-Tzu, *The Art Of War*

Chapter 3

Benchmarking

With continuing exponential improvement in processing power, numerical simulations are becoming increasingly important in astrophysics. However, with increased value being placed on their results, the scrutiny under which the results are placed must be enhanced and their limitations must be recognised. Today, computational models are used in all areas of science. In all these applications, accuracy, reliability (reproducibility) and speed have important ramifications.

It is common to test the output of these codes against each other to verify the numerical quality of the obtained solution and indeed that of the ensemble against which it is tested. This is known as *benchmarking* and is a necessary step in the development of a numerical code. Without this benchmarking step, any conclusions drawn from the results of a calculation would be potentially unsound.

The assessment of a numerical simulation can be split into two parts; verification and validation. Verification tests a code's implementation of the particular physics contained within. Potential errors here include, but are not limited to; flawed algorithms, poor quality numerics, finite resolution effects, code errors, unexpected compiler behaviour and poor, or worse, incorrect convergence. Comparison of observations with insufficiently converged or systematically incorrect solutions can range

from preposterous to subtly wrong. It is the goal of this chapter to demonstrate that I have taken all reasonable steps to ensure that TORUS is as free from these defects as possible whilst remaining widely applicable to many astrophysical scenarios.

Validation assesses the ability of a specific model to describe the true physical situation. In this chapter it is assumed that the models that have been created to represent their physical counterparts are sufficiently accurate and reflect the complexities of reality. As a result I have not discussed in depth the detail of the physics contained within each model except when concluding on the application of the code in non-benchmark scenarios.

Each key subroutine of the molecular line radiative transfer code, as described in Chapter 2, has been extensively tested and debugged. A daily test is run on a suite of benchmark problems to ensure the consistency of the code from revision to revision and to identify any regressions that may have been introduced so that they may be fixed in a timely manner.

This chapter focuses primarily on models that test a particular subsystem of the code (e.g. non-LTE level populations, datacube generation or vector sequence acceleration). Any individual model will have a certain degree of physical and geometrical complexity but by isolating and testing each subsystem it is more likely that when combined to tackle a complex scenario (such as that described in Chapter 5) requiring all the facilities of the code, the resultant output is as expected.

The first test described in this chapter (Section 3.1) is a test of all the major subsystems of the molecular-line radiative transfer code; the propagation of rays (and hence non-local information) across the grid, the determination of the local radiation field, the statistical equilibrium solution and the ability of the algorithm to converge on a solution that is accurate and reproducible with sufficient rapidity to permit the code's use on larger scales. The model contains complex physics but is simplified geometrically by the assumption of spherical symmetry. The relative abundance of the molecular tracer used in the model (HCO^+) can be varied to make the test more or less optically thick (and hence more or less difficult to converge). The output from TORUS is compared with an average of other codes.

Another model (Section 3.2) tests the other main capability of TORUS; datacube generation. An optically thick circumbinary disc assumed to be in LTE, the source function is assumed but the frequency-dependent emergent intensity must be calculated. In this case, the more complex two-dimensional geometry requires many more cells. This time the results are compared against results obtained by MCFOST (Pinte et al. 2006).

Finally, TORUS has successfully solved an analytical benchmark model; this is discussed in Section 3.3. The two-level molecule employed in this model allows for the comparison of the level populations derived by TORUS with an analytical solution and despite the many simplifying assumptions provides a stiff test of a code's performance owing to the extreme optical depths ($\tau > 1000$) attained.

3.1 One-dimensional collapsing cloud model

In order to verify the accuracy of a code, it is necessary to compare its output against either analytical solutions or the results of other codes. Whilst comparison with analytical solutions can provide useful sanity checks, they rarely provide insight into astrophysically relevant situations. I have tested TORUS against a one-dimensional collapsing cloud model to benchmark the iterative solver used to ascertain the non-LTE level population solution along the radius of the cloud.

This problem was first presented as a robust test of a molecular line radiative transfer code's ability to deal with as many astrophysical phenomena as possible at a 1999 workshop on 'Radiative Transfer in Molecular Lines' held in Leiden.[1] As a prototypical example of a problem in sub-millimetre astronomy, it has become a standard test of a molecular line transfer code's ability to reproduce level populations in a complex physical situation. Although the spherically symmetric model is quite straightforward to implement, the velocity and temperature gradients, variable turbulent line widths and multiple levels provide a stern test of a code's accuracy, especially at higher optical depths. It is noted that whilst no analytic solution exists for this benchmark, multiple codes have reached a broad consensus (within 20%, though often much better depending on the fractional abundance of HCO^+ and the specific level being verified) as to what the level populations should be along the radius of the cloud. The results of this test were published in van Zadelhoff et al. (2002).

Eight codes constituted the ensemble: 4 ALI codes (Wiesemeyer 1997; Dullemond & Turolla 2000; Ossenkopf et al. 2001), 3 (A)MC codes (Juvela 1997; Hogerheijde & van der Tak 2000; Schöier 2000) and a MULTI-type code (Rawlings & Yates 2001). These are all non-local codes; codes utilising local escape probability methods would fail the optically thick version of this test due to the radiative trapping that occurs in the centre of the cloud.

[1] http://www.strw.leidenuniv.nl/astrochem/radtrans/

As well as comparing the final output of TORUS with the other codes I have used the model to test the convergence characteristics, the convergence speed and accuracy of the code and present them in this section.

3.1.1 Model

The problem is based on a model by Rawlings et al. (1992) to analyse HCO^+ data for an infalling envelope around a protostar; a typical event in the evolution of a class 0 YSO. The model is similar to the inside-out collapse model theorised by Shu (1977) where a pressure-balanced isothermal sphere is perturbed causing the cloud to collapse from the cloud centre outwards but includes more realistic physics.

At the point of cloud collapse, a rarefaction wave travels at the local sound speed creating an enhancement in density in the interior. In this model, the collapse is represented by the negative radial velocity in the cloud interior. Because the collapse propagates outward at the sound speed in the cloud, the outer shell of the cloud is assumed to be static (see Figure 3.1). The density profile follows a piecewise power-law $n(r) = n_0(r/r_0)^{-m}$ where $r_0 = 10^{17}$ cm and $m = 1.5$ for $r < r_0$ and $m = 2$ beyond r_0; n_0 denotes the number density of H_2 at 10^{17} cm. All other parameters are specified at 50 logarithmically spaced points between 10^{16} cm and 4.6×10^{17} cm except the relative abundance of HCO^+ to H_2 which is constant. In this section, two otherwise identical models are studied with either $[HCO^+] = 10^{-9}$ (model (a)) or $[HCO^+] = 10^{-8}$ (model (b)).

In this problem, where the input parameters are tabulated, TORUS uses an AMR grid which is refined such that no cell contains more than one datum point as defined in the source model. In each octal, the salient parameters, temperature, $n(H_2)$, systematic velocity field and local turbulent velocity are assigned assuming conditions at the centre of the cell apply throughout its extent. Where the cell centre does not coincide with a datum point the parameters are logarithmically interpolated where an analytical value is not available. In the one-dimensional case, each cell represents a spherical shell. In this problem both thermal and turbulent velocities are combined.

To maintain consistency with the other codes, a common molecular data file was supplied with the problem set; a database of Einstein coefficients and collisional rate coefficients for HCO^+ with H_2 in the ground state (Green 1975; Monteiro 1985). For radii inside and outside the region covered by the data file a vacuum at $T_{\rm CMB}$ was assumed. In this model, emission and opacity due to dust is neglected as the dust

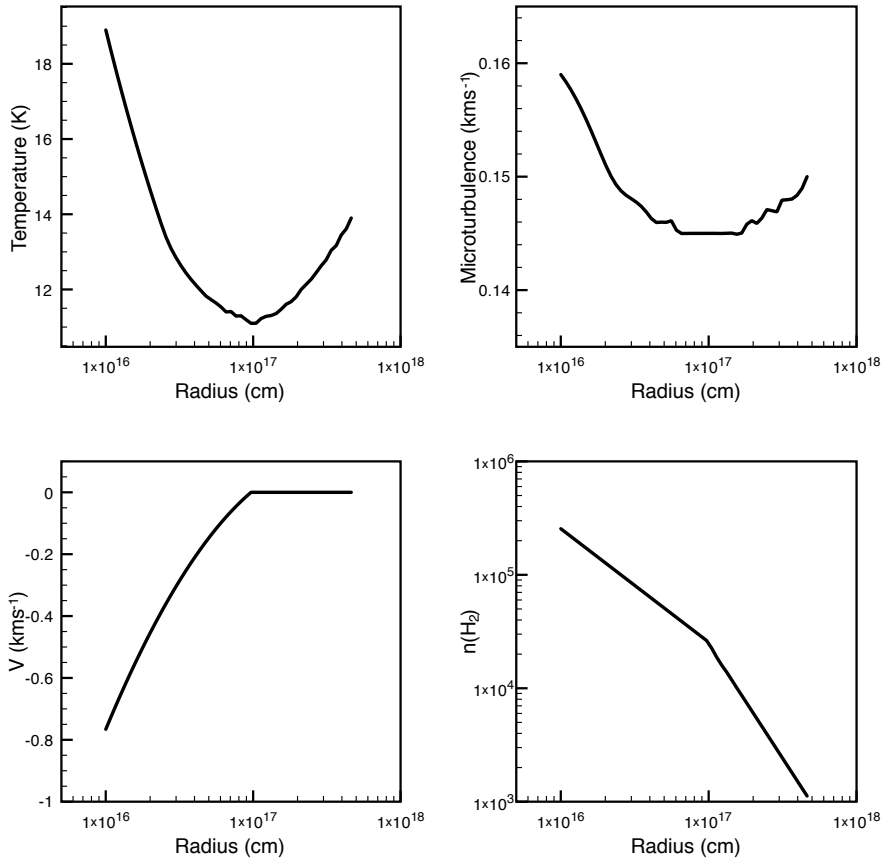


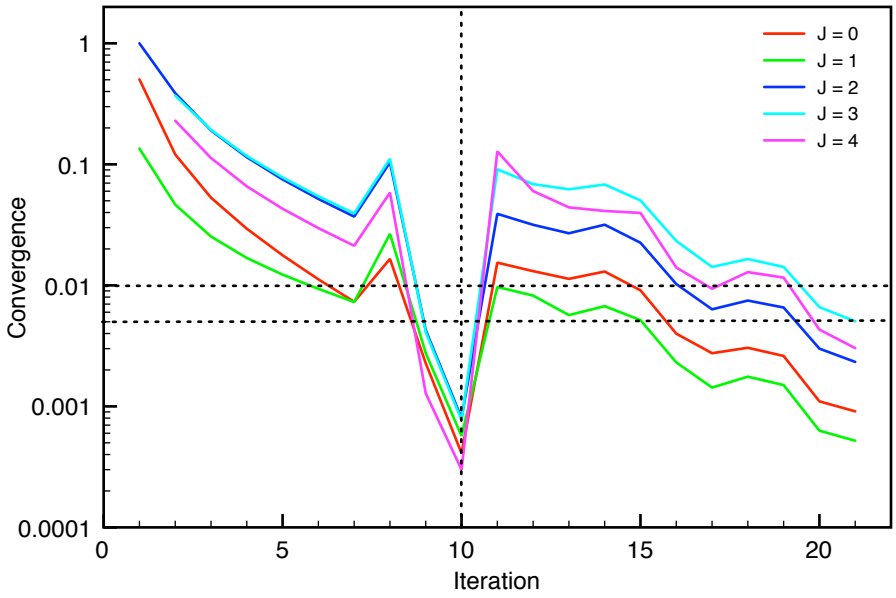
Figure 3.1: Model parameters as a function of radius along the cloud. The stark change in $n(H_2)$, temperature and velocity at 10^{17} cm is evidence of the collapse front having reached this radius. The non-monotonic kinetic temperature and microturbulence profiles complicate the model.

emits only weakly at the (sub-) millimetre wavelength associated with the HCO^+ line emission.

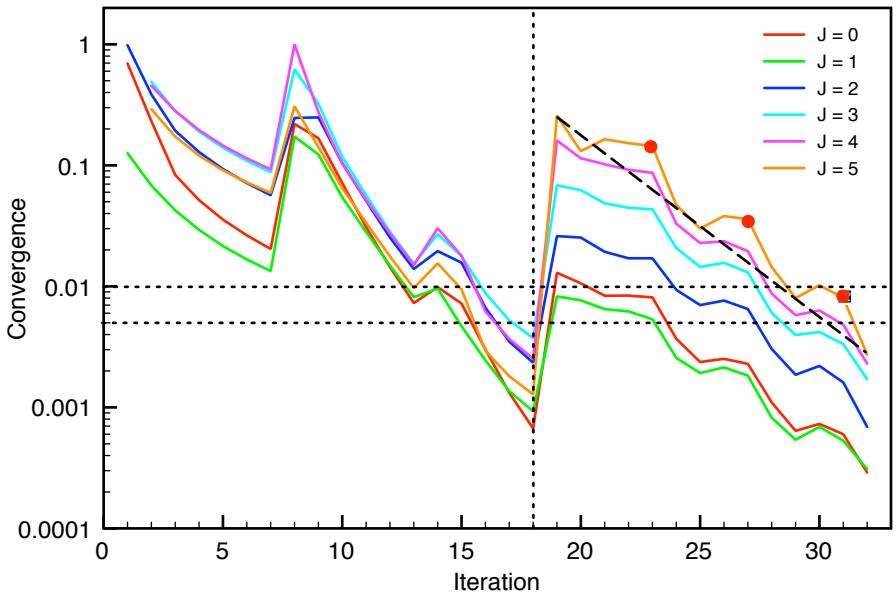
Both models were used to test the convergence and accuracy of TORUS. The grid was split into 89 cells covering the simulation volume which is sufficient to avoid spurious finite resolution effects. Only the first 8 rotational levels of the molecule were used as the neglecting higher levels made no appreciable difference to the final solution. The salient results of the simulation are presented in the sections below.

3.1.2 Convergence

As expected, model (a) converges rapidly in both time and number of rays required; just 5 minutes and 21 iterations (32768 rays per cell) gives a convergence value



(a) Optically thin model.



(b) Optically thick model.

Figure 3.2: Graph of the convergence parameter as a function of iteration number for the benchmark models. The vertical dashed line represents the transition from fixed rays to random rays. The horizontal dashed lines denote the 1% and 0.5% convergence level. In model (b) the red dots denote iterations where global Ng acceleration was performed in the non-fixed ray case. The long-dashed line overlaid represents $\sqrt{N}$ convergence.

(equation 2.21) of approximately 0.5%. Model (b) needs 32 iterations (262144 rays per cell).

In the absence of an analytic solution to compare against it must be assumed that an iterative solution tends to the true solution for a sufficiently large number of samples. That is to say that the parameter space is well sampled and that no systematic bias is present in the solution. The true level of convergence of the calculation therefore can be approximated by examining the change in the level populations from the previous iteration. This picture is complicated by the fact that convergence is a complex function of model geometry, initial conditions, space and level populations.

For the initial iteration, it was assumed that all molecules were in the ground state. The tolerance was set at 1% and Ng acceleration was used. In both models, the fixed ray stage took between 10 and 15 seconds to converge, however, as discussed in chapter 2, this solution is not free from systematic bias; the error can be greater than 100% in the higher levels compared to a solution converged to 0.1%. Each fixed ray iteration took less than 1 second on a single processor of a 2.4 GHz MacBook Pro. The random rays are used to reduce this bias but increase the random fluctuations substantially. The optically thin case converged to better than 1% in about 5 minutes having reached 16384 random rays. Within 10 minutes this improves to better than 0.5% using 32768 rays to sample the radiation field. While each level converges at its own rate, in this model the highly populated $J = 0$ and $J = 1$ levels converge much faster than the other levels having converged to better than 0.1% after 32768 rays (Figure 3.2(a)).

The optically thick model (Figure 3.2(b)) does not exhibit such rapid convergence as model (a) but it is still able to converge to the same level in just over 1 hour. The final iteration used 262144 rays and took just under 35 minutes.

The Ng acceleration is only moderately effective in these models. It is very effective in the fixed ray stage, typically reducing by five or six, the number of iterations required to reach convergence. Its effect is most striking in the optically thin case where the model converges 3 iterations after the first Ng acceleration. However, for this model this only saves a few seconds. It is more effective in the very optically thick cases where the abundance was set to be very high. In fact convergence was only possible by enabling the Ng acceleration.

The most heavily populated levels were the quickest to converge for both models although all levels seemed to converge at approximately the same $\sqrt{N}$ rate, as would be expected for a Monte Carlo method.

The effect of the dimensionality for this particular geometry has ramifications for not only the speed of execution but also the convergence rates of the problem. For

the one-dimensional case of the collapsing cloud, 89 cells are required if a minimum AMR depth of 6 is requested. By reducing the axes of assumed symmetry, the same space is split into 4882 cells and in the fully 3-dimensional cartesian space, 192172 cells are required.

The qualitative effect on the time taken to complete an iteration is obvious. However, quantitatively, the effect is ameliorated somewhat by simpler ray-cell intersection calculations and it only takes 20 minutes to perform a single iteration in the 3-dimensional case, half as long as would be expected by scaling up the spherically symmetric case. Furthermore, fewer iterations are required in higher dimensions - this may be simply because the 'local' conditions surrounding any given cell are 'more' local and/or that a greater fraction of cells are allotted for regions further out in the cloud where convergence is observed to be slower in the one-dimensional case.

Figure 3.3 illustrates some of the convergence data recorded by TORUS. Extra summary data is written to the disc if requested (through the `outputconvergence` flag), including historical data for each iteration since the start of the calculation so that trends can be ascertained. The most useful summary data are the fraction of cells converged to different tolerances (5, 2, 1 and 0.5 times the user-specified tolerance) for each level ($J = 0$ on the left). Clearly, if all the cells are converged to at least some degree then, assuming monotonic convergence of the levels towards the true solution, the entire grid is also likely to be well converged.

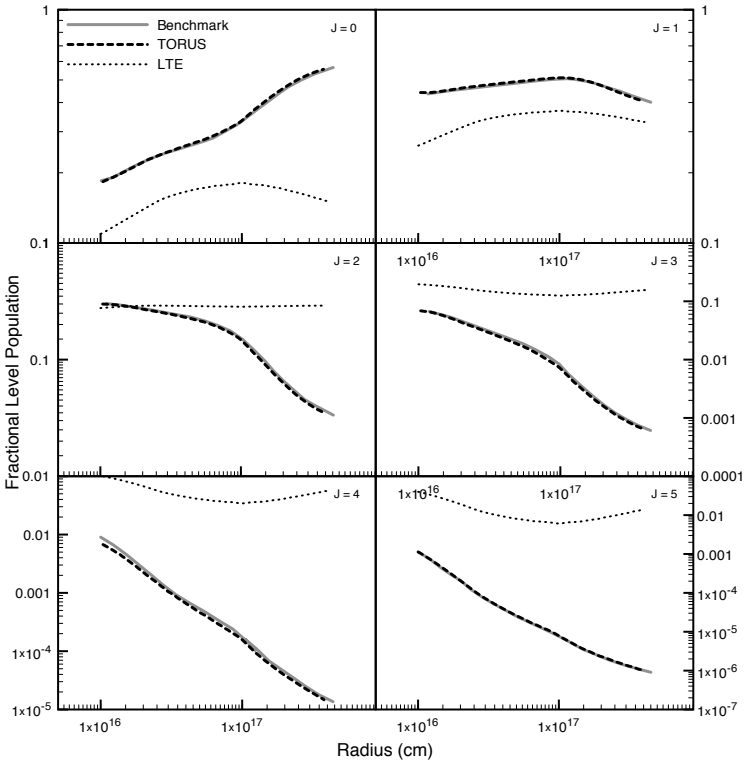
3.1.3 Results and comparison

For comparison with the ensemble of codes tested during the workshop, TORUS produced results for both models (a) and (b) where the first 8 levels were converged to better than 0.1%; no data were available for the levels above $J = 5$ so only the first 6 levels were compared. Moreover, higher levels are more sparsely populated and are subject to significantly more Monte Carlo noise.

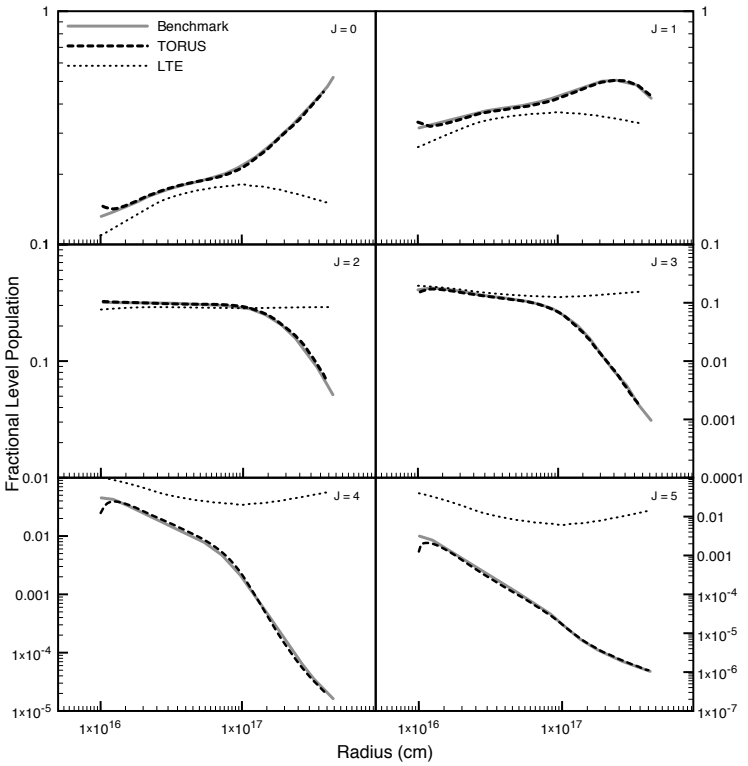
The level populations as calculated by TORUS were compared for the optically thin scenario (model (a), Figure 3.4(a)) and for the optically thick scenario (model (b), Figure 3.4(b)) with the ensemble average of other codes that completed this test. It is clear that the results obtained by TORUS are consistent with the other codes. Note that the deviation from the benchmark of the level populations at the inner boundary reflects differences in the chosen boundary conditions of the models that were combined to produce the benchmark curves.Without the exact values obtained

```
! Now trying    2560 Rays
!
! Iteration  36
! Minimum important level  8
!
! Number of unconverged cells      0
! Maximum fractional change this iteration    0.2238079 tolerance 0.005 fixed rays  F nray   2560
! Average fractional change this iteration    0.0258778 in level 5
! RMS fractional change this iteration        0.0432809 in level 5
! Std Dev                                     0.0346926
!
! Individual levels converged @ 0.0250 | 1.0000  1.0000  0.9888  0.8652  0.7079  0.6517  0.8315
! Individual levels converged @ 0.0100 | 1.0000  1.0000  0.8427  0.6292  0.3933  0.3820  0.7753
! Individual levels converged @ 0.0050 | 0.8427  0.9438  0.7079  0.4382  0.2022  0.2584  0.6404
! Individual levels converged @ 0.0025 | 0.6517  0.7978  0.6067  0.2809  0.1236  0.1573  0.3708
!
! Writing AMR grid file to:  HCO+_grid.grid
!  Written Molecular Grid to HCO+_grid.grid
!
! Dumping results
!
!
! Now trying    5120 Rays
!
! Iteration  37
! Minimum important level  8
!
! Number of unconverged cells      0
! Maximum fractional change this iteration    0.1431973 tolerance 0.005 fixed rays  F nray   5120
! Average fractional change this iteration    0.0178468 in level 5
! RMS fractional change this iteration        0.0314392 in level 5
! Std Dev                                     0.0258828
!
! Individual levels converged @ 0.0250 | 1.0000  1.0000  0.9888  0.9326  0.8202  0.8315  0.8315
! Individual levels converged @ 0.0100 | 0.9888  1.0000  0.8427  0.7079  0.5393  0.5281  0.7978
! Individual levels converged @ 0.0050 | 0.9326  0.9663  0.7865  0.5169  0.2360  0.3034  0.7416
! Individual levels converged @ 0.0025 | 0.7640  0.8652  0.6966  0.3258  0.1573  0.1685  0.5281
!
! Writing AMR grid file to:  HCO+_grid.grid
!  Written Molecular Grid to HCO+_grid.grid
!
! Dumping results
!
!
! Now trying   10240 Rays
!
! Iteration  38
! Minimum important level  8
!
! Number of unconverged cells      0
! Maximum fractional change this iteration    0.0555113 tolerance 0.005 fixed rays  F nray  10240
! Average fractional change this iteration    0.0124468 in level 5
! RMS fractional change this iteration        0.0190180 in level 5
! Std Dev                                     0.0143791
!
! Individual levels converged @ 0.0250 | 1.0000  1.0000  1.0000  0.9888  0.8652  0.8202  0.8427
! Individual levels converged @ 0.0100 | 1.0000  1.0000  0.9663  0.7640  0.6067  0.6517  0.8202
! Individual levels converged @ 0.0050 | 1.0000  1.0000  0.8652  0.6517  0.3596  0.3596  0.8090
! Individual levels converged @ 0.0025 | 0.9438  0.9775  0.7079  0.4944  0.1685  0.2135  0.6517
!
! Writing AMR grid file to:  HCO+_grid.grid
!  Written Molecular Grid to HCO+_grid.grid
!
! Dumping results
```

Figure 3.3: TORUS outputs a range of summary convergence data to standard output.



(a) Optically thin model.



(b) Optically thick model.

Figure 3.4: Relative level populations for $J = 0$ to $J = 5$ for the benchmark model.

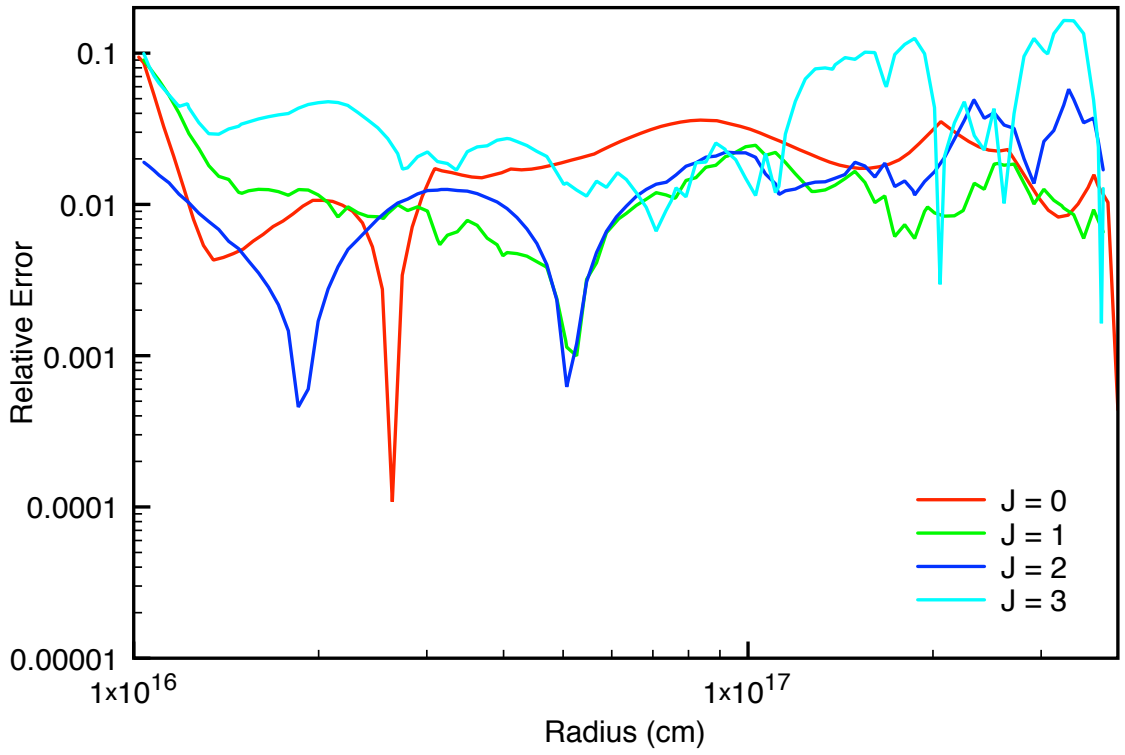


Figure 3.5: Relative error of TORUS level populations compared with ensemble average.

for each code, however, it is not possible to precisely compare our output with the ensemble average but our code varies by no more than 5% for $J = 0$ in the optically thick case (Figure 3.5). Agreement in the optically thin case is even better. It should be noted that the average of the results of the codes is not a good quantitative measure of accuracy and the inclusion of results from some of the codes may create a systematic bias (c.f. Figure 6 in van Zadelhoff et al. 2002). It is expected that the results from TORUS would be most similar quantitatively to the AMC codes in the previous reference.

The relative error of the TORUS level populations compared with ensemble average exhibits a similar trend to that of TORUS compared with itself although the errors are somewhat higher.

The relatively low abundance of HCO^+ in the model creates a scenario where the cloud is quite optically thin. The $J = 1 - 0$ transition ($\nu_0 = 89.1885$ GHz) has an optical depth of ≤ 5 from the centre to the edge of the cloud for $[HCO^+] = 10^{-9}$. The critical density for this transition is $\sim 4.6 \times 10^4$ cm^{-3}. Consequently, throughout most of the cloud radiative processes are dominant over collisional processes therefore the assumption of LTE is inappropriate. In the optically thick scenario, the

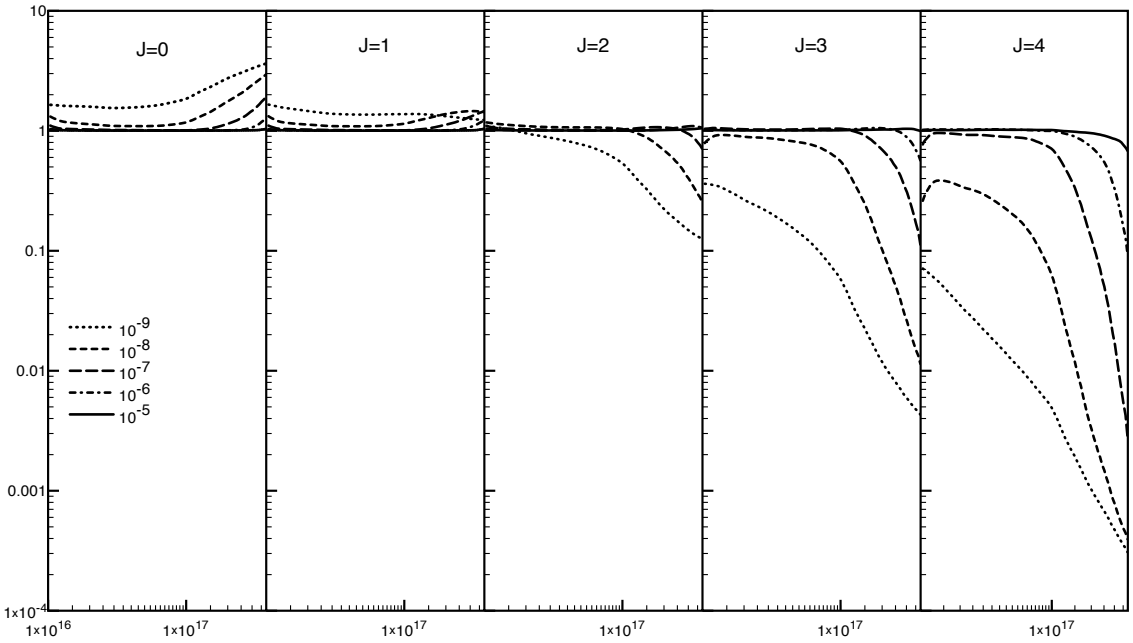


Figure 3.6: Departure coefficients for 5 different abundances from $[HCO^+] = 10^{-5}$ to 10^{-9} (see legend).

departure from LTE is still significant. Furthermore, the presence of a systematic velocity field allows radiation emitted at the line centre of a particular transition to escape the denser cloud centre more easily, pushing the solution further from LTE.

As the cosmic microwave background radiation field is able to readily penetrate the edge of the cloud, $J = 0$ and $J = 1$ are heavily overpopulated relative to LTE at larger radii. The $J = 1$ level has a characteristic temperature of 4.3 K and so the CMB is able to efficiently excite the transition. At smaller radii, $J = 2$ and $J = 3$ are closer to their LTE values because the greater density is able to thermalise the transition. However, the density in the outer regions of the cloud is too low to strongly excite higher energy transitions and the CMB is far weaker at these shorter wavelengths.

Figure 3.6 shows the departure coefficients from LTE, defined as n_i^{nLTE}/n_i^{LTE}, the ratio of the calculated relative level population for a specific rotational energy level at some radius to that predicted by assuming the cloud is collision dominated. Both model cases are far from LTE. As expected, for $[HCO^+] > 10^{-7}$ it can be seen that the local density of HCO^+ has become sufficiently great to permit collisions to dominate. The cloud has become sufficiently optically thick in most lines that these levels are well approximated by an LTE solution everywhere except the cloud periphery.

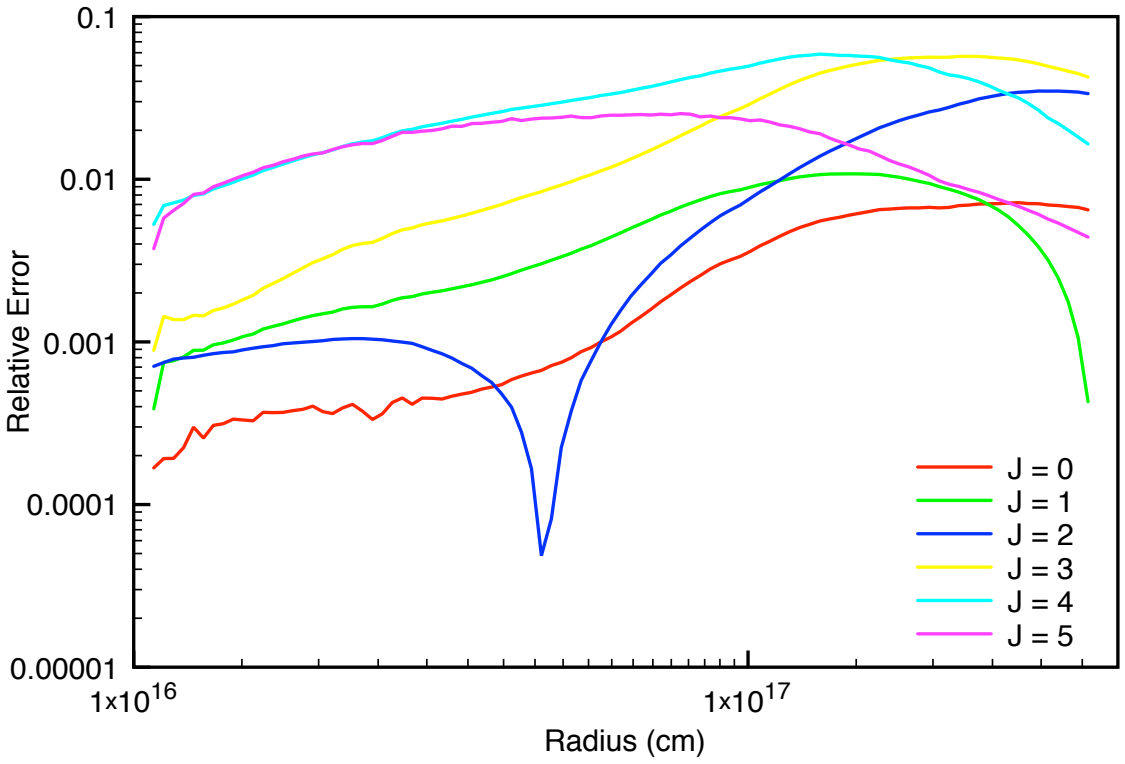


Figure 3.7: The relative error between a benchmark calculation converged to 1% and one converged to 0.1%.

3.1.4 Accuracy

The random nature of the Monte Carlo method employed in TORUS imparts significant variance in the level populations. Figure 3.7 compares two independent benchmark calculations performed at 1% and 0.1% tolerance and the actual difference is far higher than the convergence levels would suggest. This highlights the critical and often overlooked point that convergence to 1% does not guarantee 1% accuracy. The difference in this case is around 5% here at worst but can be far higher. This is a clear strength of the Monte Carlo method; deterministic methods will return the same results time after time and are dependent on perturbations in the input conditions to determine the true error. It is clear that less well populated (higher) levels exhibit greater errors but the dependence on radius can be complex for all levels.

The approximations used in local escape probability codes break down in regions of very high optical depth and the deviation from the true level populations can become severe. The authors of RADEX (van der Tak et al. 2007), an escape probability code, acknowledge that their code will be inaccurate in this regime but state that 'even if some lines are highly optically thick, RADEX may well be used to analyse other

lines which are optically thin.' It is unclear how an optically thin transition can have the correct source function if the contributing level populations are incorrect which must be the case if other level populations (contributing to the optically thick line) are incorrect.

3.1.5 Code profiling and execution time

Optimising any code that is expected to perform long calculations can be an exceptionally efficient thing to do. Premature or misinformed optimisation can lead poor results. That is, there is no point in optimising a piece of code that runs for a minute out of a 1000-hour calculation, even an order-of-magnitude only saves 54 seconds out of that hour. Conversely, a modest 10% improvement in a subroutine that runs for, say, 900 hours will save 90 hours of processing time. By profiling a code to establish which is the most frequently executed code, a developer can focus their efforts on optimising that part of the code to achieve the greatest results in the shortest time.

Because three-dimensional line radiative transfer can involve calculations using hundreds of thousands of path integrals over millions of cells, it was essential not only that the code I wrote worked, but worked *quickly*. The cluster calculation in chapter 5 took approximately 100000 CPU hours; however, if I had not profiled the code it would have taken closer to 1 million hours rendering it impractical to run at all.

Profiling using SHARK has allowed me to pinpoint the bottlenecks in the molecular module of TORUS. Figure 3.8 illustrates this well; 27% of the runtime is taken up by the `getray` subroutine. This is greatly reduced from earlier versions of TORUS where this figure would have been over twice as high. As can be seen, many other subroutines take $\sim 5\%$ of the process time which means that optimising individual subroutines would provide little further benefit. Furthermore, calls to square-roots and exponentials are unavoidable in the course of integrating the intensity along a ray however, it does illustrate that in order to improve the execution times of the `molecularLoop` subroutine significantly will require a change in design (see Chapter 6). It must be noted that this is only the case in the fixed ray case; when in the ray-doubling phase of the code, `getray` takes up proportionally more time. Moreover, in other geometries, more or less time may be spent calculating the ray trajectories and in other scenarios more or less transitions will be evaluated and the amount of time given over to solving the level population equation will vary.

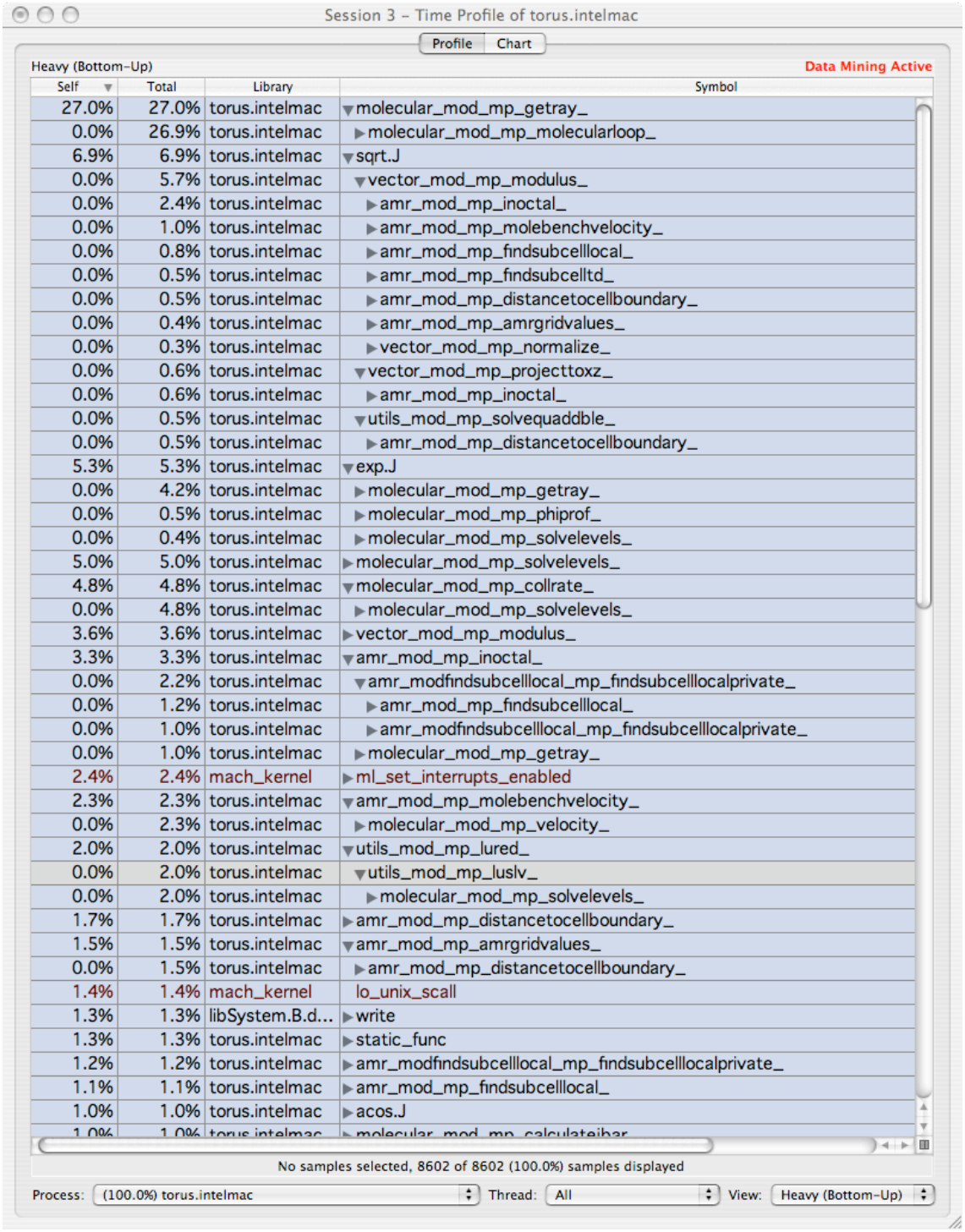


Figure 3.8: This screen capture of Shark profiling tool output at subroutine level shows the time spent in each subroutine during a sample of 30 seconds from the fixed ray stage.

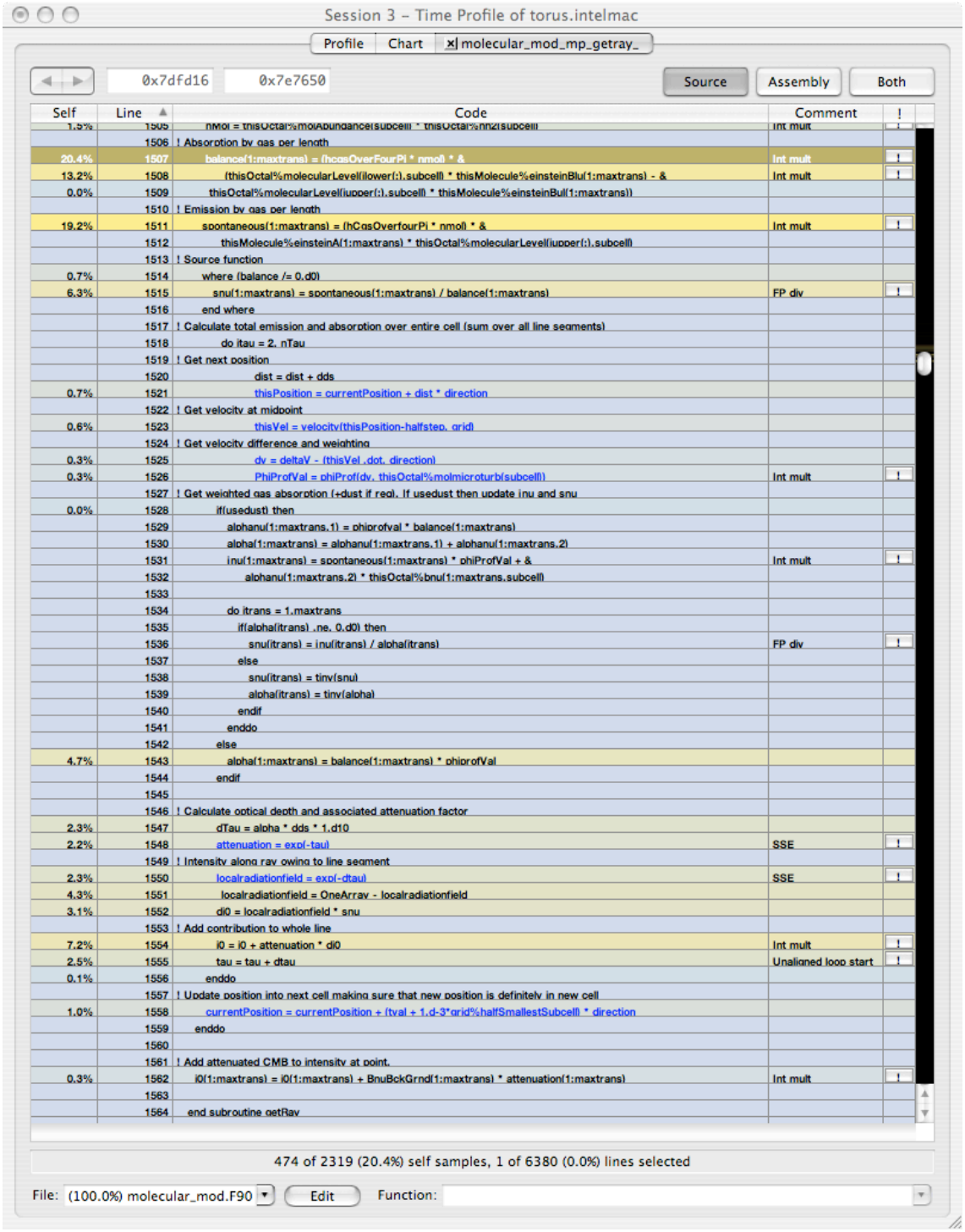


Figure 3.9: Understanding which lines of code are responsible for the greatest proportion of time taken to execute a code is the first step to effectively optimising a numerical code such as TORUS.

Using SHARK, it is also possible to attribute each sampled instruction to a specific line of source code, allowing the developer to see if, for example, a particular loop within a subroutine can be optimised. Figure 3.9 illustrates 60 lines of code in `getray` that are responsible for approximately 80% of the run time. Much of the time is spent calculating j_ν (L1511) and α_ν (LL1507-1509). The comment that an integer multiplication is occurring is indicative of the molecular level populations not being stored consecutively in memory which may be an area where the code might be improved. In line 1536 a floating-point division occurs but this is also unavoidable. Finally, SHARK shows that the compiler has managed to vectorise the optical depth calculation and this is reflected in the relatively small amount of time it takes to perform the transcendental exponential calculations.

Judicious choices of compiler flags can generate faster code than that created by the default options. The 'fast' profile uses the following flags to improve the execution time:

```
-O3 -xT -no-prec-div -fp-model fast=2 -mdynamic-no-pic
```

Each flag instructs the compiler to make more assumptions or reduce the precision with which a calculation is performed. Typically the loss of precision is negligible compared to the numerical errors associated with the Monte Carlo method and have been assuaged where precision is important (for example, in the ray-cell boundary intersection calculation). A brief description of each flag is given below:

- `-O3` instructs the compiler to pursue the most aggressive optimisation policy. It performs many optimisations which could be done by hand but which aren't for code readability reasons such as function inlining, constant propagation and loop unrolling as well as padding the size of array to allow more efficient cache use and performing more data dependency analysis to enable better data prefetching.

- `-xT` instructs the compiler to generate code that can only run on a minimum specified architecture (Intel Core2 Duo processors with SSSE3). This option is useful when it is known in advance which processor architecture will be used to execute the code. The compiler then able to use certain instruction set extensions (e.g. SSE etc.) which can improve efficiency of the code and is able to tailor the code to best utilise the known features of the processor.

- `-no-prec-div` instructs the compiler to use a less precise but faster division algorithm which is useful if full precision is not necessary. Where numerical errors are far greater in magnitude it is not necessary. `-no-prec-sqrt` enables similar optimisations for square-roots but this is included in `-O3`.

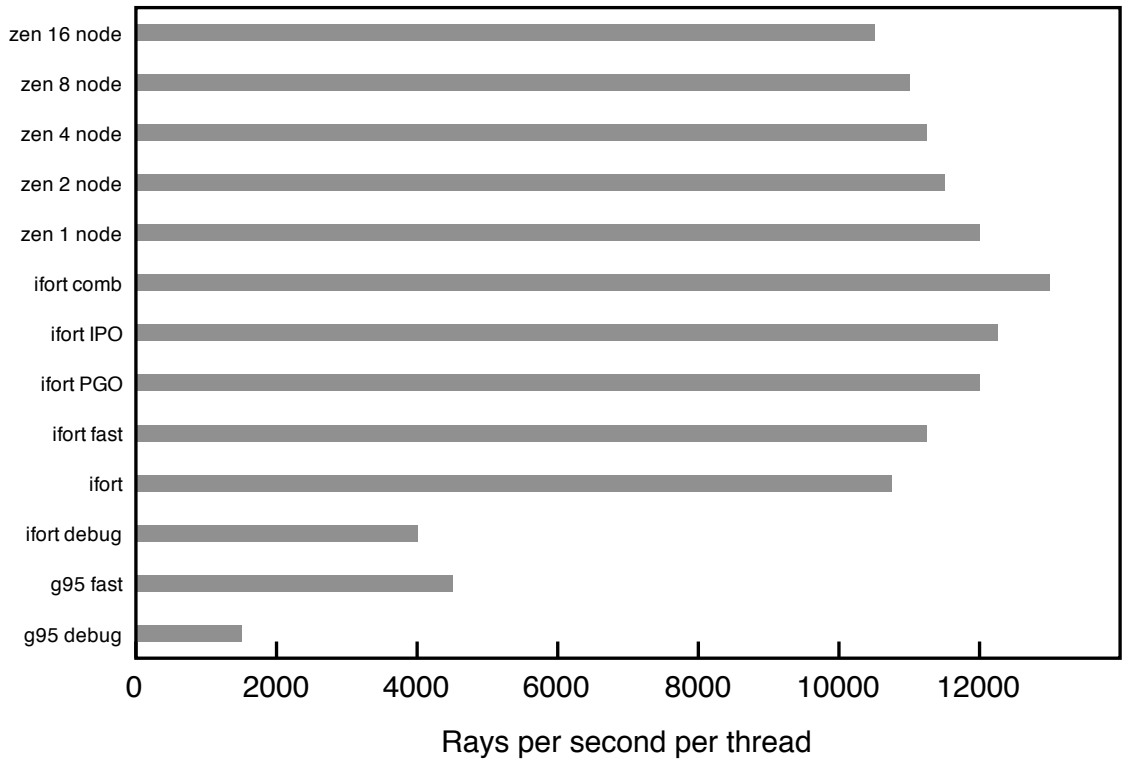


Figure 3.10: Graph of speeds attained by TORUS using different compilers / flags. Debugging flags typically reduce the execution speed by a fraction of 3.

- `-fp-model fast=2` specifies that the compiler should again optimise for speed over accuracy.
- `-mdynamic-no-pic` specifies that the compiler should generate code that is not position-independent but has position-independent external references. This makes a small difference in the execution time of the code.

Figure 3.10 clearly shows that the performance of TORUS is sensitive to compiler choice and flags. The commercial IFORT compiler generates code that is around 5 times faster that G95. Other optimisations such as inter-procedural optimisation (IPO) and profile-guided optimisation (PGO) can improve baseline performance by around 30%. All the G95 and IFORT times were obtained using a single thread (core) of a 2.4 GHz Intel MacBook Pro. Values obtained on the *zen* supercomputer were obtained by using 1 thread per core per node (e.g. 8 threads on 1 core, 16 threads on 2 cores etc.)

When executed on parallel systems, the non-LTE code scales very well. Because the simulation space is distributed over many processors and it is not necessary

to know the state of any other region, the calculation scales almost linearly as is the case with many Monte Carlo codes. The small reduction in performance as the calculation is performed on greater numbers of nodes is due to the increased amount of communication that occurs at the end of each iteration as well as taking into account the approximately constant amount of time it takes the single worker thread to calculate the convergence for the benchmark.

3.2 A circumbinary disc

In order to test the synthetic image / line profile generation routines used by TORUS, an optically thick circumbinary disc exhibiting Keplerian rotation is imaged in ^{13}CO. The model parameters are similar to those used to model GG Tau, a young multiple star system surrounded by a circumbinary disc. In this case, the more complex two-dimensional geometry requires many more cells. The results of the raytrace are compared against results obtained by MCFOST (Pinte et al. 2006).

3.2.1 Model

The Dutrey, Guilloteau, & Simon (1994) model of GG Tau provides an excellent test of the imaging capabilities of TORUS. The circumbinary disc is assumed to be in LTE and the temperature, H_2 number density and velocity profiles are all given in analytical form unlike the tabulated data for the collapsing cloud in Section 3.1. This allows a comparison of the predictions of the expected flux from the object by different RT codes that is free from discrepancies in the physical parameters and level populations.

The model parameters and scaling laws used in the model are given in equations (3.1–3.4) and in Table 3.1. Note that the velocity is assumed to be Keplerian and the turbulent linewidth is calculated as per equation 1.35.

$$n(r,z) = n_0 \left(\frac{r}{r_0}\right)^{-\frac{11}{4}} \exp\left(\frac{-z}{H(r)}\right)^2 \text{cm}^{-3} \tag{3.1}$$

$$H(r) = H_0 \left(\frac{r}{r_0}\right)^{\frac{5}{4}} \text{AU} \tag{3.2}$$

$$T(r) = T_0 \left(\frac{r}{r_0}\right)^{-\frac{1}{2}} \text{K} \tag{3.3}$$

$$V(r) = V_0 \left(\frac{r}{r_0}\right)^{-\frac{1}{2}} \text{km s}^{-1}. \tag{3.4}$$

Table 3.1: GG Tau model parameters

Parameter	Value	(unit)
n_0	6.3×10^9	cm^{-3}
$^{13}CO/H_2$	1.76×10^{-8}	
r_0	100	AU
H_0	14.55	AU
T_0	30	K
V_0	3.3	km s^{-1}
v_{NT}	0.2	km s^{-1}
Inclination	43	$^\circ$
r_{in}	180	AU
r_{out}	800	AU

As the model is symmetric about the rotational axis, a cylindrical coordinate system (r, ϕ, z) is used. TORUS takes advantage of rotational symmetry by projecting any coordinate with $\phi \neq 0$ onto the $(r, 0, z)$ plane. Naturally, this reduces the number of grid cells required to represent the space from $O(h^3)$ to $O(h^2)$. In order to resolve the disc well it is necessary to devise criteria that ensure the grid is split sufficiently to capture all its features. The following conditions were used to decide if a cell should be split, where d is the cell width and $H(r)$ is the characteristic scaleheight of the disc at radius r:

$$\frac{z}{H(r)} < 5 \quad \text{and} \quad \frac{d}{H(r)} > 0.1 \tag{3.5}$$

$$\frac{z}{H(r)} \geq 5 \quad \text{and} \quad \frac{z}{d} > 5 \tag{3.6}$$

$$r > 0.99r_{in} \quad \text{and} \quad r < 1.01r_{in} \quad \text{and} \quad d > 0.1r_{in} \tag{3.7}$$

If any of these conditions are true then the cell should be split. The cell is not split if $r < 0.99r_{in}$ or $r > 1.01r_{in}$ even if another condition is satisfied as the model is not defined outside these radii. The minimum cell depth is 3 and the maximum depth is set to 20, although this is never reached. The first two conditions cover the entire vertical extent of the disc and stipulate that at least 50 cells must be used to cover the first 5 scaleheights above the disc midplane. Beyond this, where the disc is far more tenuous, the criterion is relaxed so that the maximum cell size is less than one-fifth of the $z-$coordinate of the cell. The third criterion ensures the adequate resolution of the inner-edge of the disc.

When the disc is split according to these criteria, a region with diameter of 6×10^{16} cm is split into 23782 cells, the smallest of which is 0.12 AU across (representing a maximum AMR depth of 15). Figure 3.11 depicts the discretised density profile.

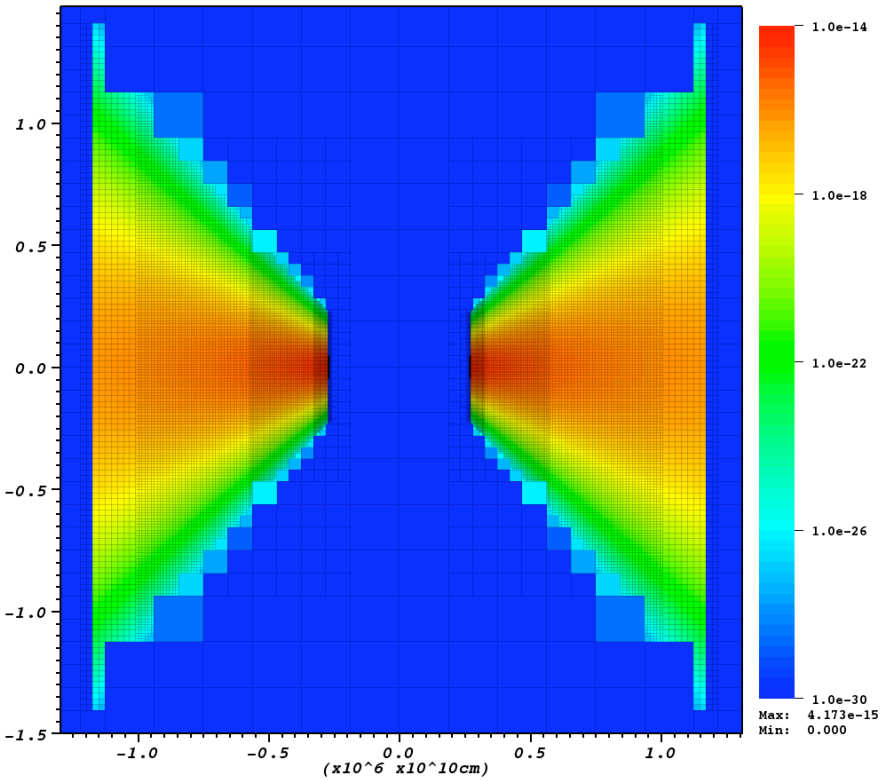


Figure 3.11: The discretised density profile of GG Tau in g cm^{-3}. The log scale captures the steep decline in material away from the disc midplane.

3.2.2 Raytracing test

The geometry and simplifying assumptions used in this model make it possible to test the capabilities of the rendering code without introducing additional numerical uncertainties inherent in the level population solver. By making these assumptions and ensuring that the same physical parameters are used by TORUS and MCFOST it is possible to isolate any potential differences in the resultant line profiles. In the scenario presented in this chapter, TORUS utilises an adaptive mesh, but the regular grids of TORUS and of MCFOST have also been compared and we find no significant difference between the AMR and the regular grid representation. Figure 3.12 shows a three colour image of the disc imaged in ^{13}CO(1-0).

Having ruled out any differences in discretisation, we assume that any differences in the line profiles can be explained by differences in raytracing strategy. The codes were created entirely independently of each other which adds to the credence that can be attached to this benchmark. Another reason that this test of the imaging code is

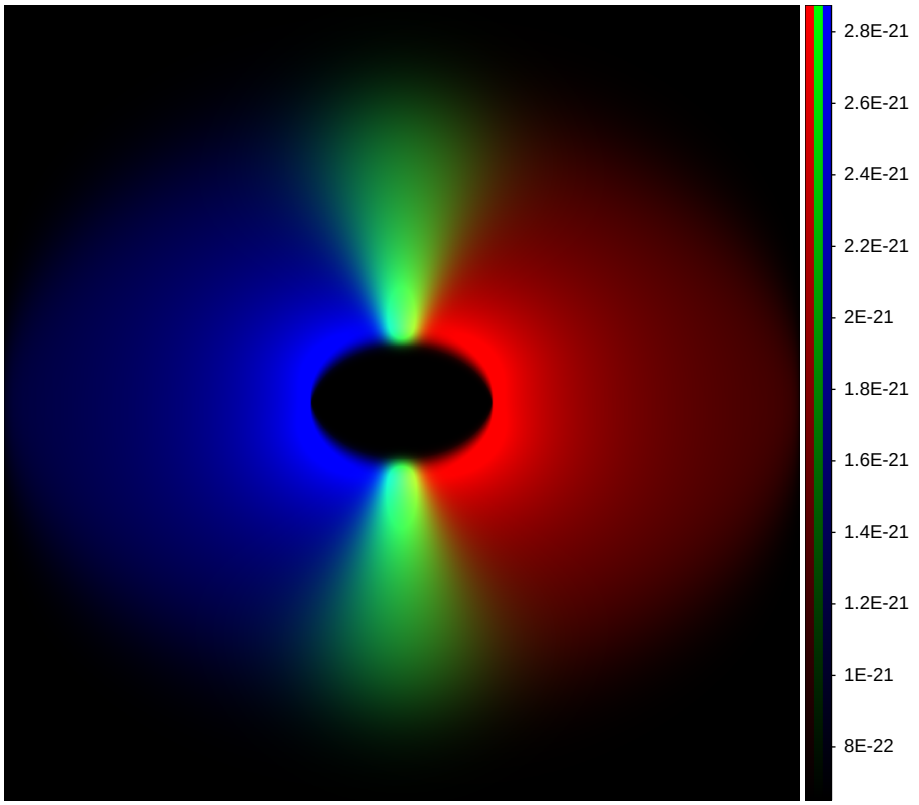


Figure 3.12: This three colour plot of GG Tau in ^{13}CO(1-0) shows that the observer is inclined to the midplane (as evidenced by the elliptical evacuated region in the centre of the disc). The red component denotes all material moving away from the observer, blue towards and green has no radial velocity component. The intensity scale is given in erg s^{-1} cm^{-2} sr^{-1} Hz^{-1}.

preferred to that of the benchmark in section 3.1 is that in van Zadelhoff et al. (2002), the level populations from all the codes are passed through the same raytracing code, SKY. This obfuscates any potential differences between codes making comparisons of line profiles not generated using SKY more difficult. Figure 3.13 illustrates the excellent agreement between MCFOST and TORUS line profiles in this test case. The agreement is well within observational error and discrepancies may be put down to minor differences in discretisation strategy. Having passed both tests we are confident in attempting a far more complex geometry that does not exhibit any symmetry and where no physical parameters are defined analytically.

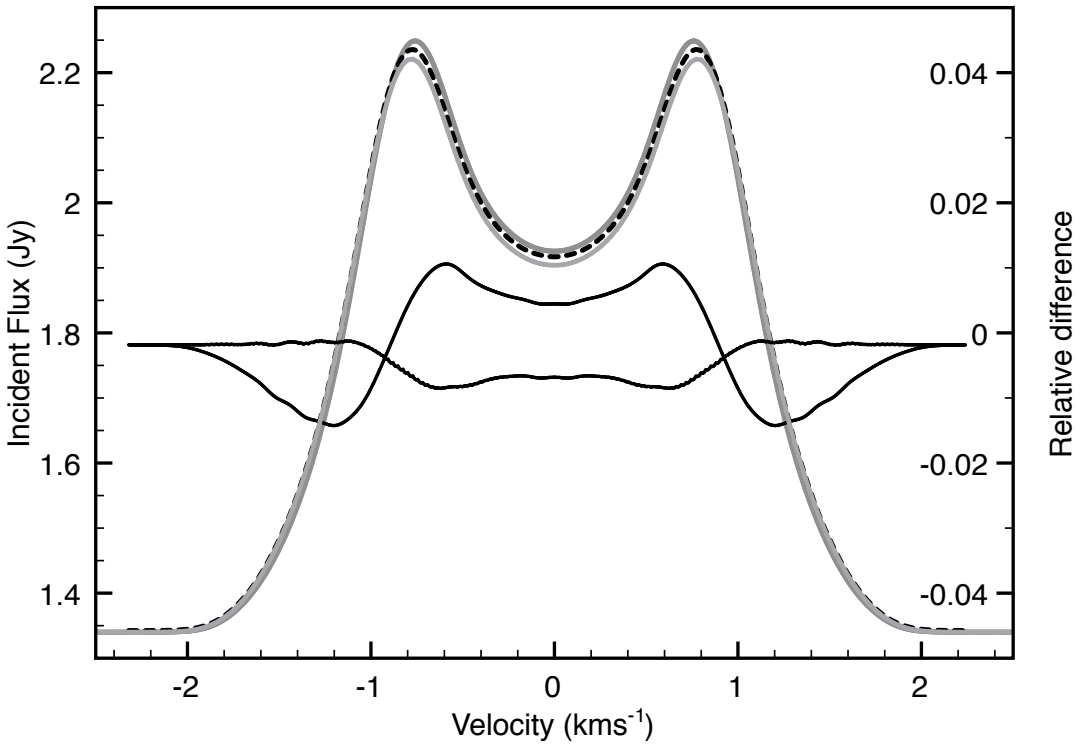


Figure 3.13: This line profile shows the excellent agreement between MCFOST (dashed black line) and TORUS (solid grey lines). A distance of 150 parsecs was assumed to convert intensity into flux. The two TORUS lines represent models with (lower peak) and without (higher peak) subpixel sampling using a resolution of 512×512 pixels. The two black lines denote the relative difference between the MCFOST solution and the TORUS solutions on the right-hand scale.

3.3 Benchmarks for H_2O radiative transfer

Water has been observed in many astrophysical sites (e.g. Melnick et al. 2000; Nisini et al. 2010), both in our galaxy and in others. It plays an important role in the chemistry of interstellar environments and observations of water lines can provide important diagnostics of the conditions in the chemically active shells surrounding the sites of star formation (i.e. photon-dominated regions, or, PDRs). Accurate RT modelling of these often optically thick lines is consequently an essential precursor for providing constraints on the conditions in these regions.

In order to further test the quality of the results of their codes, many of the participants of the 1999 workshop developed another series of benchmarks that test the codes' compliance with analytical solutions in the case of a fictive two-level model of ortho-water (developed by David Neufeld) and the complex case of an AGB star con-

Table 3.2: Water model parameters

Parameter	Value	(unit)
n_{H_2}	10^4	cm^{-3}
H_2O/H_2	$10^{[-4..-11]}$	
T_0	40	K
α^1	0	km s^{-1} pc^{-1}
^{2}	100	km s^{-1} pc^{-1}
v_{turb}	0	km s^{-1}
r_{min}	0.001	pc
r_{max}	0.1	pc

tributed by Jeremy Yates. In this section, I present the results obtained by TORUS for the two analytic models. The results of the other codes are available online.[2]

3.3.1 Model

The first two problems discussed at the workshop use a fictive molecule that uses the two lowest energy levels of ortho-H_2O, the so-called $(1_{10} - 1_{01})$ transition. This transition is important because it can be used to trace density enhancements in shock-heated clouds where the H_2O abundance is significantly enhanced by the return of water-ice previously frozen out onto dust grain mantles to the gas phase. Indeed, as the temperature continues to rise (above ∼300 K) other formation pathways become feasible (Bergin et al. 2003) and the abundance can be enhanced by many orders of magnitude (upto 10^{-4} relative to H_2).

In the analytical models, an isothermal, homogeneous cloud is presented containing only the model two-level ortho-H_2O molecule. We run two sets of models for two values of radial velocity gradient ($\alpha = 0$ and 100 km s^{-1} pc^{-1}). For each value of α the relative abundance of H_2O is varied from 10^{-4} to 10^{-11} representing increasingly optically thin scenarios. Tables 3.2 and 3.3 state the salient simulation parameters for the molecule and the cloud respectively. Note that γ_{21} and γ_{12} are the collisional de-excitation and excitation rates respectively and α is the radial velocity gradient given in km s^{-1} pc^{-1}, where a positive value indicates expansion. The parameters, u and s in the Table 3.3 are substituted for the Boltzmann factor and the critical density ratio model of the model respectively in the analytic derivation given in Section 3.3.2.

[2]http://www.sron.rug.nl/∼vdtak/H2O/

Table 3.3: Two-level molecule parameters

Parameter	Value	(unit)
E_{21}	556.936	GHz
γ_{21}	2.18×10^{-10}	cm ^{3} s^{-1}
γ_{12}	1.12×10^{-10}	cm ^{3} s^{-1}
A_{21}	3.458×10^{-3}	s^{-1}
n_{cr}	1.59×10^{7}	cm ^{3}
$\exp(-\Delta E/kT)$	0.512	
$\frac{n_{crit}}{n_{H_2}}$	1588	

The models are nominally defined over 50 logarithmically-spaced radial points from $r_{\rm min}$ to $r_{\rm max}$ but as TORUS splits the simulated region in same fashion as described in the HCO^+ benchmark above and the values are constant or analytically defined this is ignored.

As it is possible to derive an analytical solution in the optically thin and thick limits, the models are a good test of a code's ability to reproduce exact values. Codes utilising local approximations like the escape probability/LVG method are unable to do so in conditions of high optical depth as in the static $[H_2O] = 10^{-8}$ case, highlighting the importance of full radiative treatments in arbitrary media.

The analytic and numerical solutions for each model are given in the sections below.

3.3.2 The static cloud

Analytical predictions can be made for the critical abundance of H_2O at which the population of the upper level becomes significant.

The equation of detailed balance can be formulated in terms of an escape probability, β – the probability that a photon is not reabsorbed before leaving the cloud,

$$n_1 C_{12} - n_1 C_{21} - n_2 A_{21} \beta = 0 \tag{3.8}$$

or equivalently, as a ratio of the two levels,

$$\frac{n_2}{n_1} = \frac{C_{12}}{C_{21}} \left(1 + \frac{A_{21}}{C_{21}} \beta\right)^{-1}. \tag{3.9}$$

The ratio of collisional excitation to deexcitation is determined by Boltzmann distribution,

$$u = \frac{C_{12}}{C_{21}} = \frac{g_2}{g_1} e^{-\Delta E/kT}, \tag{3.10}$$

so equation 3.9 can be written as

$$\frac{n_2}{n_1} = \frac{g_2}{g_1} e^{-\Delta E/kT} \left(1 + \frac{A_{21}}{C_{21}}\beta\right)^{-1}. \tag{3.11}$$

Furthermore, by reformulating $\frac{A_{21}}{C_{21}}$ as the ratio of the critical density to the model density,

$$s = \frac{A_{21}}{C_{21}} = \frac{n_{\rm cr}}{n_{\rm H_2}}, \tag{3.12}$$

equation 3.11 becomes

$$\frac{n_2}{n_1} = \frac{g_2}{g_1} e^{-\Delta E/kT} \left(1 + \frac{n_{\rm cr}}{n_{\rm H_2}}\beta\right)^{-1} \tag{3.13}$$

or

$$\frac{n_2}{n_1} = \frac{u}{1 + s\beta} \tag{3.14}$$

using some simplifying substitutions set out in Table 3.3.

For this fictive two-level molecule, $n_2 = 1 - n_1$, so the relative population of the emitting level can be determined by the equation,

$$n_2 = \frac{1}{1 + \frac{1}{u}(1 + s\beta)}. \tag{3.15}$$

Equation 3.15 is dependent on β which is in turn a function of optical depth. For a static cloud with constant density, the optical depth across the cloud at the rest frequency of the transition can be written as

$$\tau_0 = \alpha_\nu \int ds = \frac{h\nu_0}{4\pi} \left(n_1 B_{12} - n_2 B_{21}\right) \phi(\nu_0) L, \tag{3.16}$$

where, L is the diameter of the cloud.

For $n_2 \ll n_1$ and $v_{\rm turb} = 0.32$ km s^{-1} $\Rightarrow \Delta\nu_D = 3.57158 \times 10^5$ Hz and recalling the relationship between Einstein coefficents (Equations 1.10 and 1.11), equation 3.16 simplifies to

$$\tau_0 = 0.194 \times \frac{X[\rm H_2O]}{10^{-10}} \tag{3.17}$$

For relative abundances of $\rm H_2O \lesssim 10^{-9}$, the optical depth is negligible and thus the radiation is not trapped and does not affect the population of the upper level. At low optical depths $\beta \approx 1$ and the population of the upper level can be directly calculated for this model as

$$n_2 = \frac{1}{1 + \frac{1}{0.512(1+1588)}} = 3.22 \times 10^{-4}. \tag{3.18}$$

For higher abundances, where radiative trapping and molecular collisions dominate, the population of the upper level tends to that expected when the two levels are in LTE, i.e. from equation 3.10, $n_2 = 0.3388$.

The formal escape probability for a uniform sphere (e.g. Osterbrock & Ferland 2006) is given as:

$$\beta_{\text{sphere}} = \frac{3}{2\tau}\left(1 - \frac{2}{\tau^2} + \left(\frac{2}{\tau} + \frac{2}{\tau^2}\right)e^{-\tau^2}\right). \tag{3.19}$$

Assuming significant trapping occurs at $\beta \simeq 0.01$, $\tau \simeq 150$ and from equation 3.16, the 'critical' abundance can be written as

$$X_{\text{crit}}[\text{H}_2\text{O}] = \frac{8\pi^{3/2}\tau_0\nu^2\Delta\nu_D}{(1-n_2)c^2A_{21}n_{\text{H}_2}L} = 1.17 \times 10^{-7}. \tag{3.20}$$

Figure 3.14 illustrates the predicted fractional upper level populations for both the static and the expanding cloud. The TORUS derived values for the innermost radial points very closely match (i.e. $\ll 1\%$) the analytical values which are valid only for small radii where any radiative trapping is most effective.

Figure 3.15 shows the population of the upper level as a function of radius across the cloud for a range of abundances from 10^{-5} to 10^{-10}. It clearly illustrates that at the cloud edge, radiation is able to escape more easily and because no background radiation field is present as an external boundary condition, the upper level populations decay freely towards the low abundance solution.

By plotting the level ratio as an excitation temperature (equation 1.37), it can be seen that for $\text{X}[\text{H}_2\text{O}] \gg \text{X}_{\text{crit}}$, $T_{\text{ex}} \rightarrow T_{\text{kin}} = 40\ K$ implying that the levels have been thermalised. Conversely, for $\text{X}[\text{H}_2\text{O}] \ll \text{X}_{\text{crit}}$, $T_{\text{ex}} \ll T_{\text{kin}}$. The excitation temperatures for $\text{X}[\text{H}_2\text{O}] = 10^{-4}$, 10^{-8} and 10^{-10} are shown in Figure 3.16 .

TORUS seems to perform very well in this benchmark. It is comparable with the other codes that took part in this test and does not appear to suffer from the same degree of Monte Carlo noise that the other MC codes did, even at very high abundances. I believe that this is explained by the use of Ng acceleration. Using this acceleration method, TORUS is able to converge to a solution at a level of $\sim 1\%$ in around 10 minutes on a single processor even when initialised assuming all molecules are initially in the ground state. Interestingly, TORUS is not able to achieve this without Ng acceleration in the very high optical depth case ($\text{X}[\text{H}_2\text{O}] = 10^{-4}$).

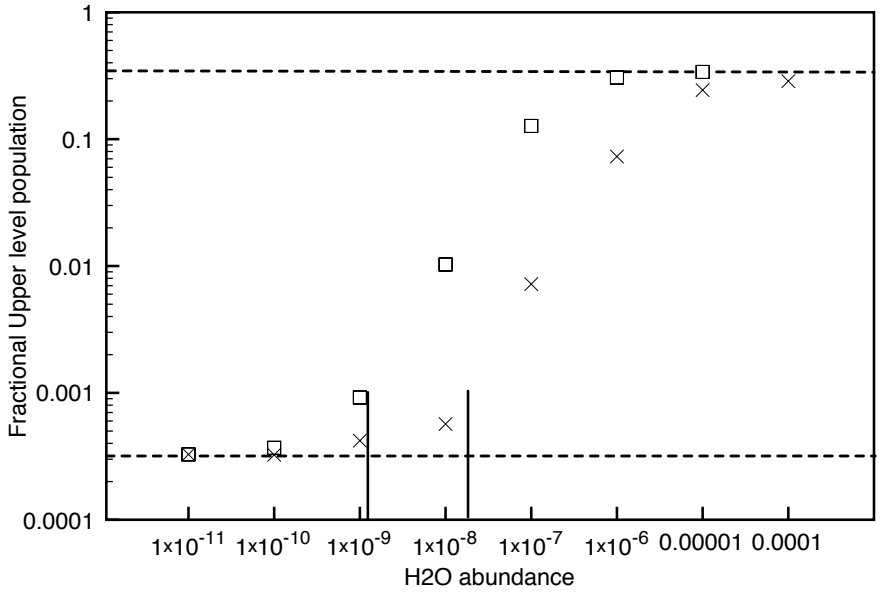


Figure 3.14: Graph of the upper level populations as a function of abundance. Squares denote the static cloud; crosses denote the expanding cloud. The lower dashed line is the low abundance limit whereas the upper dashed line represents the LTE equilibrium state. The black vertical lines represent where the optically thin approximation is no longer valid.

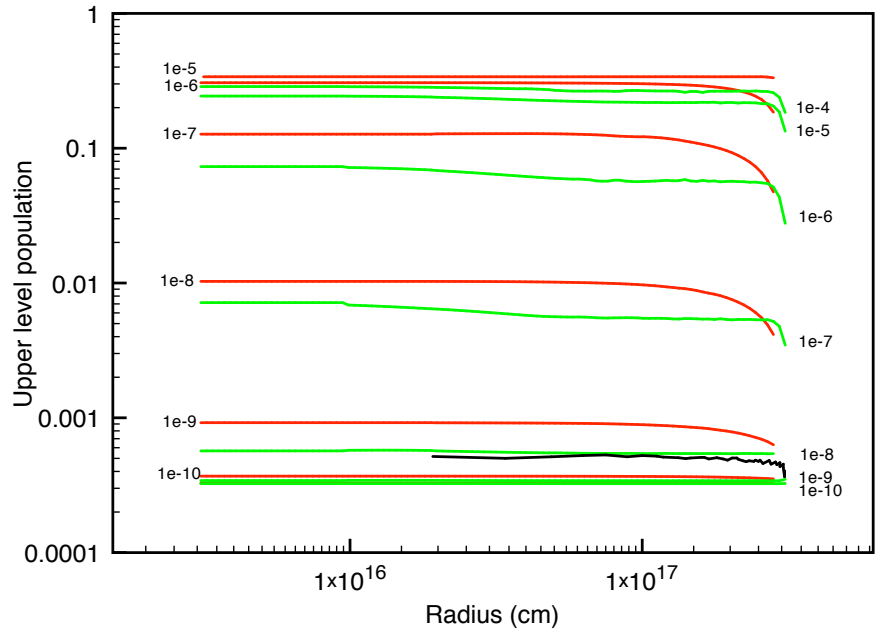


Figure 3.15: Graph of the upper level populations as a function of radius and abundance. The red lines represents the static cloud (abundances on left); green lines represent the expanding cloud (abundances on right). The black line represents the solution of the Schöier AMC code.

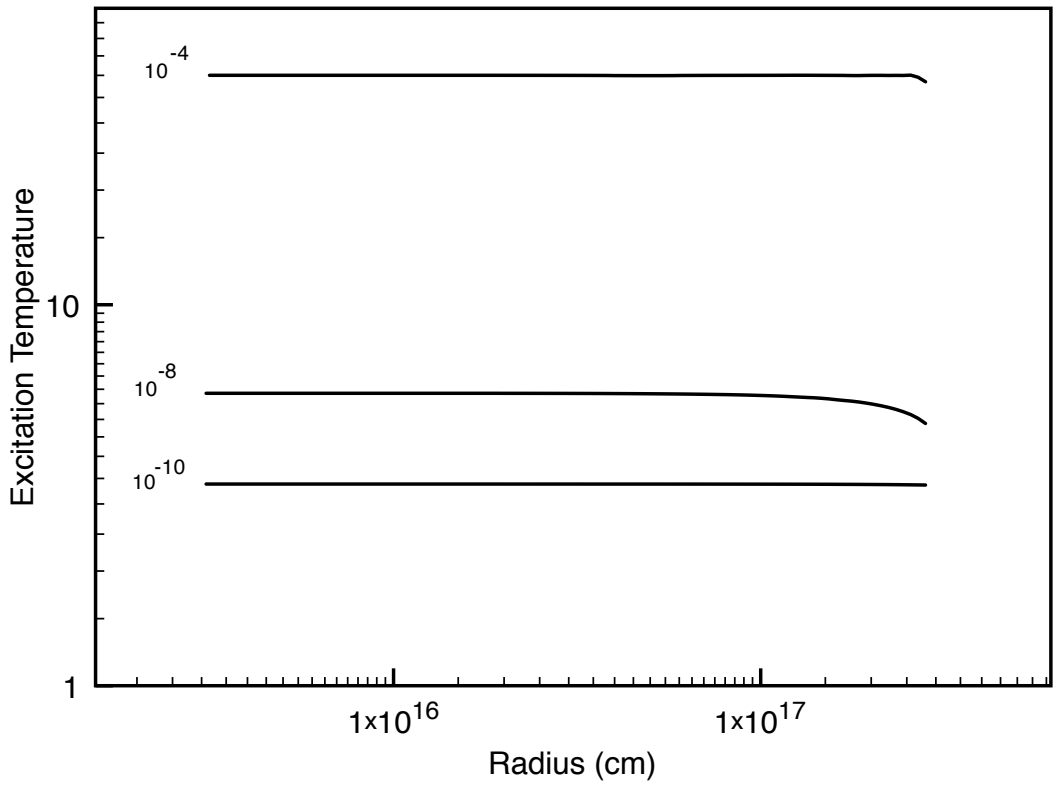


Figure 3.16: Graph of the excitation temperature as a function of radius for three different H_2O abundances

3.3.3 An expanding cloud

Up to equation 3.16, the derivation of the previous section was applicable to any homogeneous, isothermal cloud. A large velocity gradient increases the probability of the escape of a photon, effectively decoupling different layers of the cloud, reducing the effective optical depth. The optical depth in the case of the large velocity gradient was derived by Sobolev 1960,

$$\begin{aligned}\tau_{\mathrm{LVG}} &= \frac{hc}{4\pi\alpha}\left(n_1 B_{12} - n_2 B_{21}\right) \\ &= \frac{c^3 A_{21}}{8\pi\alpha\nu^3}\left(1 - 2n_2\right) \\ &= \frac{X[\mathrm{H_2O}]}{1.51\times10^{-8}}(1 - 2n_2) \end{aligned} \tag{3.21}$$

where equation 3.21 is specific for this expanding cloud model.

This critical abundance is an order of magnitude smaller than that determined for the static cloud, (equation 3.16). This too is illustrated in Figure 3.14 where higher abundances are required for the expanding cloud to attain the same upper level

population. Moreover it is expected that the level populations are constant over the cloud except close to the cloud edge. Figure 3.15 again corroborates TORUS compliance with this analysis.

The workshop organisers noted in their conclusions that the excitation temperature should increase towards the cloud centre. Codes that are not able to fully resolve the innermost cells underestimate the optical depth of the excitation and consequently show a decrease in excitation temperature towards the centre of the cloud. This effect seemed to affect a number of the codes that participated in this benchmark but closer inspection of the TORUS level populations does shows that the well-converged TORUS grids do not incur this spurious phenomenon.

Finally, TORUS seems quantitatively better than the other codes that undertook the expanding cloud test. Comparing the AMC code that was used at the workshop (black line in Figure 3.15) with that of TORUS (green) for the high abundance ($X[H_2O] = 10^{-8}$) case shows that not only is the AMC code subject to a significant amount of noise but it also is not able to predict the expected upper level populations, even at the cloud centre. TORUS is able to reproduce the analytical value of 5.6×10^{-4}. This is likely to have a significant effect on the total line luminosity where the organisers state that even the most effective code (the AMC code) was only able to recover 80% of the expected flux.

Science is what we understand well enough to explain to a computer. Art is everything else we do.

– Donald Knuth, *A=B* (1996)

Chapter 4

Transforming particle-based representations to grid-based representations

Problems in star formation typically span many orders of magnitude in both space and density. SPH and AMR naturally resolve the fine structure of these problems well, whereas fixed, regular grids will often be insufficient to capture crucial information such as peaks in the density profile where a protostellar core far smaller than the size of a grid cell has begun to accrete material. Moreover, the resolution may be unnecessarily fine in large, low density regions that are unimportant in a star formation context. Nested grids can ameliorate this problem somewhat but they often need to be hand-crafted to fit each individual problem (e.g. D'Angelo et al. 2002; Ercolano et al. 2007). The flexibility of the AMR method is that it adapts the resolution given to a region depending on the criteria the user applies. Typically, one is interested in resolving the temperature and density profile in a region, although when calculating line radiative transfer accurately it is also necessary to accurately resolve variations in the velocity field, owing to the anisotropy in the absorption introduced by the line profile function.

An alternative family of techniques that seek to overcome the issue of resolution are the particle-based 'Lagrangian' methods such as 'Smoothed Particle Hydrodynamics' (SPH), originally developed to describe astrophysical fluid dynamics problems

(e.g. Gingold & Monaghan 1977; Lucy 1977). As in grid-based methods, each unique particle contains information about the density, mass, velocity at its location. In addition, to these fundamental quantities, others may be stored such as the magnetic field strength or the abundance of a tracer molecule; these are astrophysically important parameters but the SPH is utilised in wider fluid dynamics scenarios.

Owing to the Lagrangian nature of the SPH method, calculations are made between particles and by design the particles move to represent the areas of highest density. Thus, the method is capable of dealing with simulation volumes that contain large variations in density making it a natural choice for star formation simulations. By considering additional properties stored on the SPH particles it is possible to study many diverse topics in astrophysics such as the cosmological expansion of space (e.g. Springel 2005), the formation of spiral structure in galaxies (e.g. Dobbs & Bonnell 2007), clustered star formation (e.g. Bate 2009a) and planet formation in protoplanetary discs (e.g. Ayliffe & Bate 2009b).

While SPH produces 3D temperature, density and velocity fields, radiative transfer is required in order to simulate the interaction of radiation with this material and produce artificial 2D maps or position-position-velocity cubes like those measured by observers. By comparing the simulated maps with the real observations, the strengths and weaknesses of the models can be identified.

Because it is necessary to perform many path integrals, radiative transfer, is most naturally treated using grid-based methods; it is particularly time consuming in particle methods, as the interaction between each particle and each path is complex and slow to compute. As a result, should one wish to perform an accurate treatment of radiative transfer, or model the effects of radiative feedback in an SPH simulation, a natural strategy is to interpolate from the values of the salient parameters stored on the particles on to the grid cells (and in the latter case *vice-versa*). Moreover, in order to facilitate a self-consistent radiative treatment, many timesteps of both the hydrodynamical code and the radiative transfer code must be performed. As a result, the algorithm must be efficient enough to readily convert one representation to another without using too much processing time.

In this chapter I discuss the basic formulation of the SPH method (Section 4.1). More details on the technical aspects of implementing an astrophysical hydrodynamics code can be found in Monaghan (1992) and in more recent references (e.g. Bate & Bonnell 1997; Price 2005; Price & Federrath 2010). The kernel smoothed particle-to-grid algorithm used in TORUS is detailed in Section 4.2. Specifically the section is split into two parts; creating a grid that accurately reflects the SPH particle distribution and efficiently populating the grid cells with values commensurate with

those stored on the particles. Finally, published applications of this algorithm are described in Section 4.3.

4.1 Smoothed particle hydrodynamics

To estimate the value of a continuous scalar function, A, at some point, $\mathbf{P}$, in the simulation space, the SPH method uses an integral interpolant. Formally,

$$A(\mathbf{P}) = \int A(\mathbf{p}) W(|\mathbf{P} - \mathbf{p}|, h) d\mathbf{p}, \tag{4.1}$$

where W is known as the *kernel function* and h is the *smoothing length*.

A good kernel is smooth and continuous over the entire simulation space. Furthermore, in the limit as the smoothing length tends to zero, $A(\mathbf{P}) = A(\mathbf{p})$ when $\mathbf{P} = \mathbf{p}$, i.e. it should act as a delta function. However, a kernel with finite width ($h \neq 0$) no longer satisfies this criterion and the relationship in equation 4.1 becomes an approximate one. Furthermore, whereas the function is continuous, the points at which the function is to be sampled, *particles*, are discrete. Thus, the integral representation given above is approximated using a summation over all particles, the volume element $d\mathbf{p}$ having been replaced with its discrete equivalent, m_i/ρ_i:

$$A(\mathbf{P}) \approx \sum_{i=1}^{N_{\text{part}}} A_i(\mathbf{p}_i) W(|\mathbf{P} - \mathbf{p}|, h_i) \frac{m_i}{\rho_i}, \tag{4.2}$$

where m_i, ρ_i and h_i are a particle's mass, density and smoothing length respectively and N_{part} is the number of particles being used to sample the continuous function.

The smoothing length of each particle determines the characteristic size scale that defines its 'sphere of influence'. Thus each particle has its own smoothing length. However, it is useful to nondimensionalise the kernel in such a way that it written in terms of a single dimensionless parameter, q:

$$W(q, h_i) = \frac{\mathcal{W}(q)}{h_i^{\nu}} \tag{4.3}$$

where

$$q = \frac{|\mathbf{P} - \mathbf{p}_i|}{h_i}. \tag{4.4}$$

Assuming a closed form for the particle smoothing length relating it to the true density of the medium being described (e.g. Springel & Hernquist 2002; Price &

Monaghan 2004, 2007), h_i can be written as:

$$h_i = \eta\left(\frac{m_i}{\rho_i}\right)^{1/\nu}, \tag{4.5}$$

where η is a constant controlling the approximate number of nearest neighbours a particle has and where ν is the dimensionality of the simulation. It is thus possible to succinctly re-write equation 4.2 as follows:

$$A(\mathbf{P}) \approx \sum_{i=1}^{N_{\rm part}} A_i(\mathbf{p}_i)\frac{\mathcal{W}(q_i)}{h^\nu}\frac{m_i}{\rho_i}, \tag{4.6}$$

$$\approx \sum_{i=1}^{N_{\rm part}} A_i(\mathbf{p}_i)\eta^{-\nu}\mathcal{W}(q_i). \tag{4.7}$$

If h is permitted to vary in such a way as to keep the number of neighbouring particles constant at each timestep (e.g. Bonnell & Bate 2006; Bate 2009a), then it necessary to use an iterative scheme (e.g. Newton-Raphson iteration) to determine a self-consistent smoothing length that satisfies the neighbour constraint. In any case, either equation 4.6 or 4.7 is used to determine the SPH interpolated value of the function to be evaluated.

The weight of each particle's contribution is thus determined by the smoothing kernel. One mathematically appropriate choice of smoothing kernel would be a spherically symmetric gaussian function, however there are major computational disadvantages that preclude its use in hydrodynamical calculations, primarily that the computation of exponentials is very slow compared to, say, polynomial functions and that the contribution of particles greater than a few smoothing lengths away will be negligible. A common smoothing kernel is the cubic spline function (Monaghan & Lattanzio 1985), defined below,

$$W(q,h) = \frac{\mathcal{W}(q)}{h^\nu} = \frac{\sigma}{h^\nu}\begin{cases} 1-\frac{3}{2}q^2+\frac{3}{4}q^3 & \text{for } 0 \leq \text{q} < 1, \\ \frac{1}{4}(2-q)^3 & \text{for } 1 \leq \text{q} < 2, \\ 0 & \text{otherwise,} \end{cases} \tag{4.8}$$

where σ is the normalisation constant for the dimensionless kernel $\mathcal{W}(q)$ defined such that

$$\int_0^\infty \mathcal{W}(q)dV = 1. \tag{4.9}$$

For $\nu = 3$, $\sigma = 1/\pi$. This kernel is popular due to its ease of computation and compact support over $2h$. By utilising the property of compact support possessed

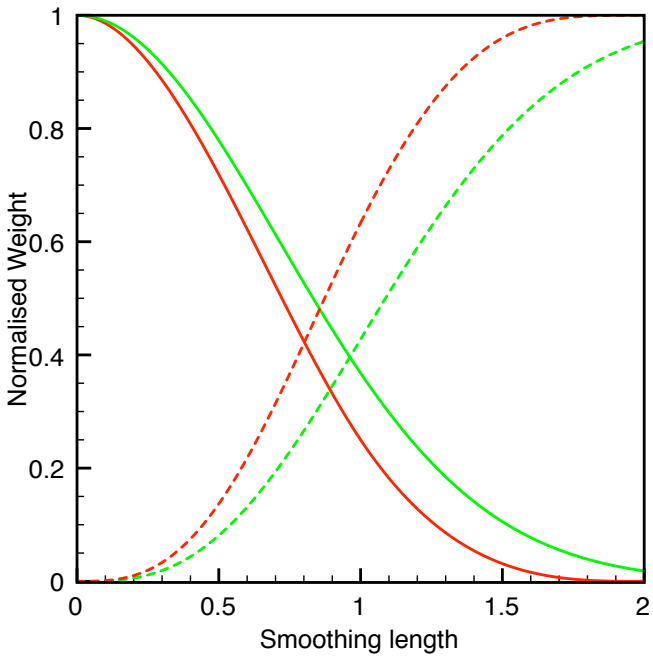


Figure 4.1: Two SPH smoothing kernels (solid line) and their normalised integrals (dashed line). Green is a truncated gaussian and red is a cubic spline.

by the kernel smoothing function it is possible to discard those particles with $q \geq 2$ without having to evaluate the kernel function. Figure 4.1 highlights the numerical differences between these two kernels. Other smoothing kernels are discussed in greater detail in Monaghan (1992).

By dividing the mass to be modelled between all the particles, the conservation of mass is a guaranteed property of the SPH method. Moreover, it can be shown that because the derivative of a scalar function approximated in this way is easy to calculate, linear and angular momentum are naturally conserved making it trivial to satisfy the hydrodynamical continuity, momentum and energy equations.

4.2 An efficient particle-to-grid algorithm

In order to map the density structure of a particle-based representation onto the adaptive mesh employed by TORUS it is necessary to use a particle-mesh algorithm that interpolates data stored at irregularly spaced points onto the grid. Many methods exist that already do this; the simplest is to use only the particles within a cell to determine a parameter value. For example, the density is taken to be the average density over all the particles within the cell or the sum of the particle masses divided by the cell volume. However, the variation between these two calculations

may be large and neither takes into account the concept of smoothing inherent in the SPH technique. Furthermore, whilst this technique tends to give an accurate result in high density regions, where the particle density is naturally greatest, it is not capable of giving an answer in regions where no particles exist and further provision must be made to evaluate parameters in empty cells. More sophisticated algorithms take into account a particle's contribution to its associated cell and its nearest neighbours using a linear or quadratic kernel, namely the cloud-in-cell algorithm (see Laux 1996, for details), however even this algorithm fails to take into account the variable radius over which each individual particle contributes to its environs. In fact, in order to accurately determine information about a parameter at any given point in space from a particle ensemble, it is necessary, in theory, to calculate and sum an appropriately weighted contribution from each particle.

4.2.1 Creating the grid

To create an AMR grid from an SPH particle distribution it is necessary to test each cell against a mass criterion, a density criterion and where calculating line transfer, a velocity criterion. Where these criteria are not met, a cell is recursively split until the conditions in each child cell are met. Handling complex astrophysical structures necessitates the use of a full 3-dimensional representation of the space so each parent has 8 child cells, or octals.

The mass per cell criterion determines the maximum number of particles that may occupy a cell (at least for equal mass particles), thus ensuring that no cell has more than a certain fraction of the entire mass of the simulation within it. Typically, this parameter is set to a value that balances the need for accurate determination of the conditions within a cell and sufficient grid resolution within memory constraints. The mass per cell condition allows a wide range in the total number of octals while maintaining a total mass which is correct to within a few percent.

The density criterion dictates the range of particle densities within a cell. This criterion facilitates the resolution of density gradients which are both physically and computationally important in radiative transfer calculations (e.g. treatment of shocks, jets, etc.). Large changes in density from cell to cell can have deleterious effects on the convergence of the iterative algorithm used to determine the relative level populations of a molecular species within the cell so it is critical that the volume over which they change is well resolved. Typically, this criterion is set so that the maximum density stored on a particle within the cell is not greater than twice that

of the minimum density or that the parameter

$$f_{\rm split} = \frac{\rho_{\rm max} - \rho_{\rm min}}{\rho_{\rm max} + \rho_{\rm min}} < \frac{1}{3} \tag{4.10}$$

The velocity criterion regulates the variation in the magnitude of the velocity within a cell. It is vital to have good velocity resolution over a cell when studying the effects of line radiation in an object, to obtain the correct line profile shape since absorption and emission of line radiation is a strong function of Doppler-shifted frequency. Therefore, where possible we use this criterion to ensure that the range of velocities within a cell is less than 5 times that of the turbulent line velocity.

In summary, the density condition is effective at giving extra resolution in regions of high density contrast which are likely to be important in radiative transfer calculations. Used in conjunction with the mass limit per cell condition it is possible to add extra resolution to the grid where required whilst maintaining an accurate total mass.

If any of the conditions in the parent cell meet the splitting criteria then the cell is split. This process is iterated until each cell no longer triggers any of the criteria or a maximum cell depth is reached. This is to ensure that the simulation does not run out of memory. A further condition is imposed that the minimum cell size is no smaller than the smallest inter-particle separation ($\sim 0.25 h_{\rm min}$), which is commensurate with the natural resolution of the SPH simulation. Figure 4.2 shows a cut-through version of a grid created from an SPH particle distribution representing a star forming cluster using the above splitting criteria; the minimum cell size in this grid is ~ 0.05 AU and the spatial extent of the grid is 1 pc. The number of cells used in this case was one-third that of the number of SPH particles.

4.2.2 Determining cell parameters

Having created a grid it is then necessary to populate each cell with parameters pertinent to the calculation that is to be performed. The method outlined here is optimised to take advantage of geometry where possible but is not prejudiced against the general case where the distribution of particles is not known *a priori*.

For each cell in the grid, the centre is chosen to be the point in space where the conditions that pervade the cell are to be determined. Whilst it is trivial to determine the region over which a particle acts (because the distance at which a particle

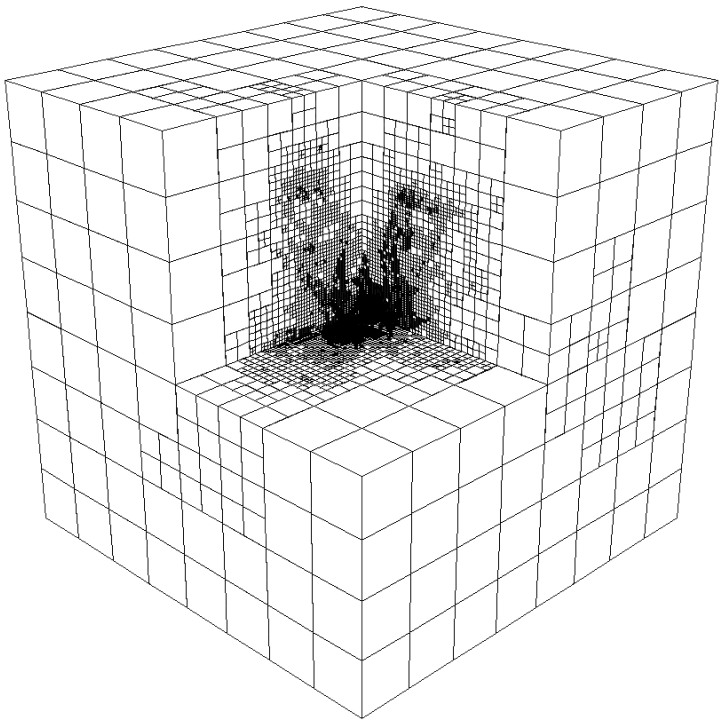


Figure 4.2: A cutaway of an adaptively refined grid created using the particle-to-grid method.

ceases to contribute is fixed at $2h_i$ in the case of the cubic spline kernel used here), the inverse problem of determining which particles contribute to the grid cell centre is less trivial. That is, some may lie within the cell but some may not. Furthermore, some may lie within the cell yet not contribute to it at all; Figure 4.3 illustrates this. Naïvely, one might assume that by finding the particle with the greatest smoothing length, $h_{\max}$ and subsequently checking for all particles that lie within $2h_{\max}$ of a point one is guaranteed to recover the most accurate answer possible. Whilst this is certainly true, it turns out to be so inefficient that determining the contents of an entire grid like this would take an impractically long time. Not only does $h_{\max}$ overestimate the smoothing lengths of all but one particle in the ensemble, but also, due to the direct relationship between smoothing length and density it is unrepresentative of the entire physical region; very low density regions being the exception rather than the rule in circumstellar discs and star-forming clusters. Moreover, these very low density regions play a negligible role in radiative transfer and are naturally

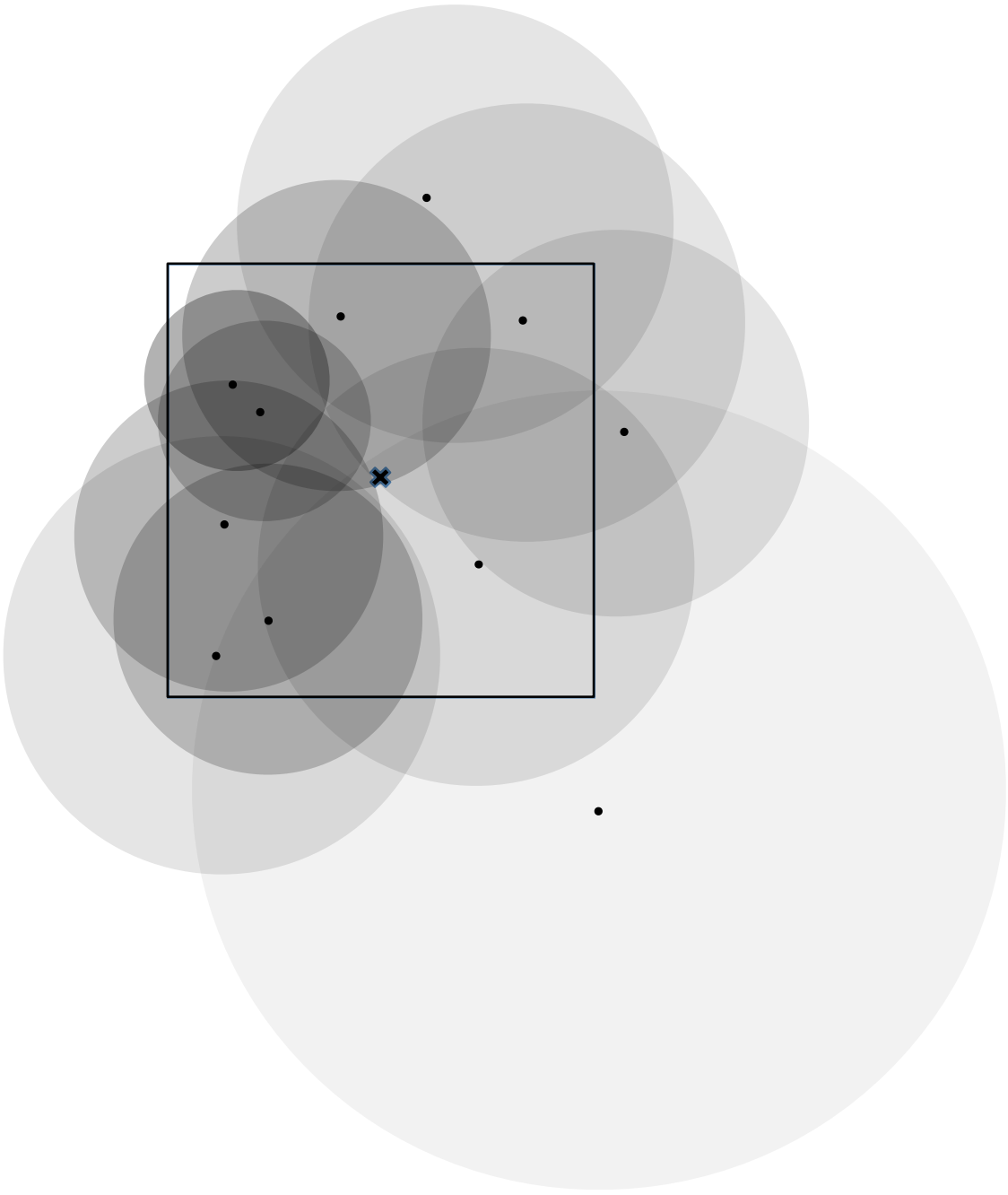


Figure 4.3: Schematic diagram of the contribution of particles to a cell. Darker grey denotes higher densities. In actuality, the density contributed to a point decays as the distance from the particle increases.

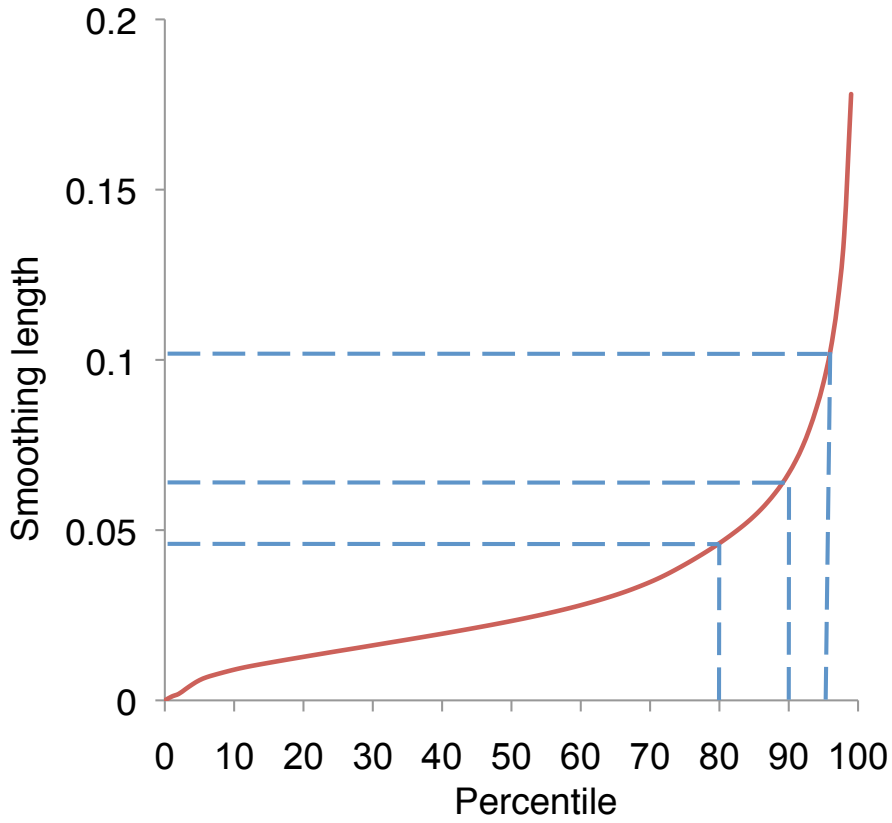


Figure 4.4: Cumulative graph of all particle smoothing lengths (in 1/10s of a parsec) for an SPH cluster geometry. $h_{\max}$ (not shown) ≈ 0.43 pc whereas $h_{95\%} \approx 0.01$ pc.

undersampled by the SPH technique.

In this method, the parameter $h_{\rm crit}$ is defined as the smoothing length of the particle lying at some critical point in the distribution of smoothing lengths of all particles; by default this is set at 95%. This length is often as much as an order of magnitude less than that of $h_{\max}$ but still greater than that of all but the 5% of particles representing the least well-populated areas of the physical space (see Figure 4.4). However, as the adaptive mesh is already split so that the cell size is commensurate with the smoothing lengths of the particles it contains, a better estimate for the critical lengthscale over which to search for nearby particles is

$$r_{\rm crit} = \min(4d, h_{\rm crit}), \tag{4.11}$$

where d is the cell width. The factor of 4 has been empirically derived as the smallest factor that retains the furthest most particles that are likely to contribute whilst keeping computational speed high. Equation 4.5 shows that for each ten-fold reduction in r, 3 orders of magnitude of density resolution are lost. However, as the

radius over which contributing particles are searched for reduces, the volume (and hence the computational effort) reduces as r^3, making the problem more tractable. Furthermore, as it is the least dense regions that are no longer resolved, the impact on the solution is minimised. If however there are no particles in a point's vicinity then the code makes another attempt to locate a nearby particle that can be used to derive the local density. The algorithm is repeated with a wider search volume whose radius is determined by

$$r_{\rm crit} = \min(\max(4d, 2h_{\rm crit}), 0.5h_{\rm max}). \tag{4.12}$$

This second condition covers almost all cases where the initial condition does not suffice but if even this radius is not sufficient then one last attempt is made using $r_{\rm crit} = h_{\rm max}$; beyond this the cell is justifiably declared empty and its parameters set to some global minimum.

Sorting the particle list by ascending $x-$coordinate facilitates the reduction of the number of candidates that can contribute to a point, $\mathbf{P}$. This is achieved by removing all particles that lie outside 2 $r_{\rm crit}$ in the abscissal dimension, leaving only particles whose $x-$coordinate lies within $\mathbf{P}_x \pm r_{\rm crit}$. In practice, by utilising *a priori* knowledge of the approximately constant number of nearest neighbouring particles this can be rapidly achieved by testing the $x-$coordinate of a particle 32 elements away from the particle whose $x-$coordinate is closest to that of the cell. If that cell lies outside 2 $r_{\rm crit}$ then the a particle that lies 16 elements away is tested until a range of particles satisfying this criterion is found (following the lines of a standard binary search).

In the case where all the particles are equally spaced this reduces the number of comparisons necessary to $4\sqrt[3]{N_{\rm part}}$. Each particle in this reduced list is then tested against the same criterion in y, i.e. $|\mathbf{P}^y - \mathbf{p}_i^y| < 2r_{\rm crit}$. For all those particles where this criterion is satisfied a further test for the $z-$component is made. This process discovers all particles that lie within a cube of side length $4r_{\rm crit}$ centred about the point. Whilst this means that only approximately half the particles are expected to lie within a sphere of radius 2 $r_{\rm crit}$, it is still more efficient than directly calculating the pairwise distance for each particle in the entire ensemble. The logic is such that, at each step in the algorithm, it is necessary to do less work than the stage before. Furthermore, the order in which the criteria are applied has been selected in such a way as to reject the greatest number of particles as early on in the process as possible; that is, culling by x then y then z is the most efficient method of locating nearby particles for a disc with its semi-major axis in the xy-plane. By storing the differences calculated when ascertaining the proximity of a particle to the point it is possible to speed up this calculation slightly; it is then only a matter of summing them in

quadrature to find the pairwise separation, $r_i^2 = (\mathbf{P}^x - \mathbf{p}_i^x)^2 + (\mathbf{P}^y - \mathbf{p}_i^y)^2 + (\mathbf{P}^z - \mathbf{p}_i^z)^2$.

Once the reduced list of particles has been created it is then a matter of calculating the contribution of each particle to the point, determined by the kernel smoothing function (equation 4.8).

In a homogeneous, isotropic distribution of 10^6 particles, using a smoothing kernel with compact support of $2h_i$ reduces the number of particles able to contribute to a point by approximately five orders of magnitude. Specifically, the number of particles able to contribute is independent of volume or number, being constant at ~ 60 for $\eta = 1.2$.

As stated in section 4.1, the value of a scalar parameter at a point in space, $\mathrm{A}(\mathbf{P})$, is the weighted sum of the stored parameter values, A_i, over all nearby particles (equation 4.6). This formulation assumes that $\sum \mathcal{W}(q_i) \approx 1$ which is typically true for points in the bulk of the simulation surrounded by a full complement of neighbours. Where it is true, it is possible to normalise $A(\mathbf{r})$ by dividing by $\sum \mathcal{W}(q_i) \approx 1$, reducing the variation associated with kernel smoothing. The superposition of many spherically bound kernels on to a regular 3D grid will not recover a constant field but rather it will tend to oscillate around the true value; the field will be over-estimated at the particle position (where the weighting function at that point is 1 and the sum over all other particles is positive) and will similarly be underestimated at the point in space centred at the midpoint of surrounding kernels. This normalisation can certainly improve the appearance of an image, (see Price (2007)) but can cause problems at free surfaces where a smooth decay is preferred.

Particles with $h > \sqrt[3]{V_{\mathrm{sim}}/N_{\mathrm{part}}}$, the average linear separation of particles in the simulation volume, V_{sim}, are expected to lie close to a free surface and are initially classed as 'hull' particles. The algorithm adaptively decides whether to normalise or not depending upon one of two strategies:

- If $\sum \mathcal{W}(q_i) > W_{\mathrm{norm}}$ then normalise; *or*
- Normalise only if all contributing particles are determined to be 'bulk' particles, (as opposed to hull particles).

Either strategy relies upon an empirically determined parameter, W_{norm}, that is typically set between $0.3 - 0.5$. In the first case, normalising when $\sum \mathcal{W}(q_i) > 0.3$ typically recovers total grid masses most similar to the SPH representation for a cluster, whereas $W_{\mathrm{norm}} = 0.5$ seems to work well for disc-like geometries (Acreman et al. 2010b). In the second case, the designation of a hull particle may change if it is

subsequently found to belong to a set of particles where $\sum \mathcal{W}(q_i) > W_{\rm norm}$, whereby it is classed again as a bulk particle. Similarly, it is possible, though unlikely, that a bulk particle may become a hull particle if it belongs to a set that fails to meet the criterion. In any case, it is possible for a user to override this strategy as it is not obviously desirable to let the temperature or velocity decay at a free surface even in the apparent absence of matter.

Once the contributing particles have been located and their weights determined they can be reused for each parameter that needs to be found. A vector such as velocity must be split into its 3 components and each one must be determined before outputting the resultant vector. Moreover, as this process is the slowest part of the algorithm, if the distance between grid cells is small then it is possible to reuse the list of contributing particles and just recalculate the weights. This can substantially reduce the amount of time taken to perform the interpolation.

The algorithm is repeated for each grid cell until the grid has been populated with data. The original serial version of this code took ∼12 hours to create a $\sim 10^6$ cell AMR grid using the default parameters on an SPH cluster containing 3.5×10^6 particles. By using a smaller $h_{\rm crit}$, this time could be reduced substantially at the cost of significantly decreased accuracy at the cloud edge. Similarly, runtime can be reduced by relaxing the splitting criteria to produce a less refined grid. However, the code has subsequently been parallelised by Dr Tim Harries; as the values in one cell are not dependent on another, the improvement in time is approximately linear so a grid with comparable resolution to that of the SPH representation can be created in minutes, given enough processors.

Interpolation

Despite the efficiency of the particle-to-grid algorithm, it is still many orders of magnitude slower to calculate the contributions from each appropriate particle to a point on the grid than it is to perform polynomial interpolation on a regular grid. TORUS takes a pragmatic approach to determining the values of parameters that do not lie on the grid cell centre, for example, the velocity of the gas at a point that is used to determine the Doppler-shifted line centre for a ray-trace.

By using the kernel interpolation algorithm to populate values for the velocity (or density) at the octal corners, of which there are 27, it is possible to linearly or quadratically interpolate to obtain a good approximation to the kernel smoothed value. For each cell in Figure 4.2, the relative fractional difference between the stored grid cell centre value for the magnitude of the velocity was compared with

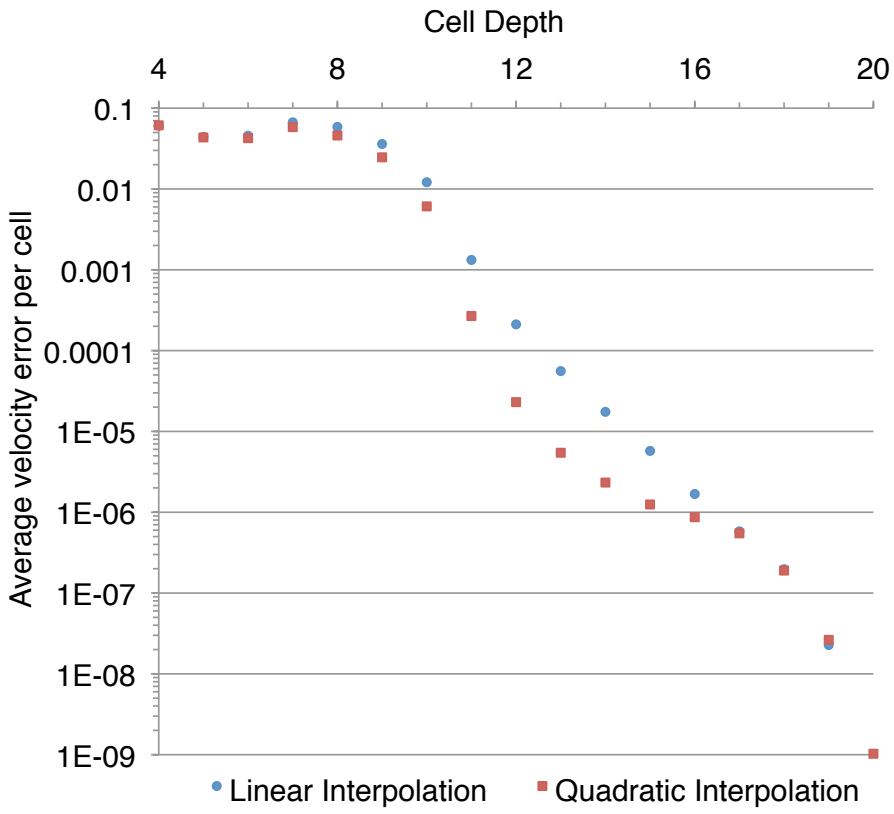


Figure 4.5: This graph demonstrates the efficacy of the quadratic interpolation routine used in TORUS and the general efficacy of the polynomial interpolation routines employed.

the interpolated value determined from the cell corners. The average error was recorded for each level of cell recursion. The results are shown in Figure 4.5. The quadratic interpolation routine gives very similar results to the linear interpolation routine for very small and very large grid cells but is more accurate by up to an order of magnitude for cell depths between 10 and 16. Approximately 2.25×10^6 out of the 3.5×10^6 cells lie between these depths which cover length scales between $\sim 10 - 650$ AU.

Note that it is not mathematically sound to quadratically interpolate the density field, as this may result in unphysical (negative) densities. However, using linear interpolation to sub-sample the density field during the synthetic line mapping in conjunction with sub-pixel intensity sampling (described in Section 2.3.2) can produce far higher quality images than images that are created without using these features (Figures 4.6 and 4.7).

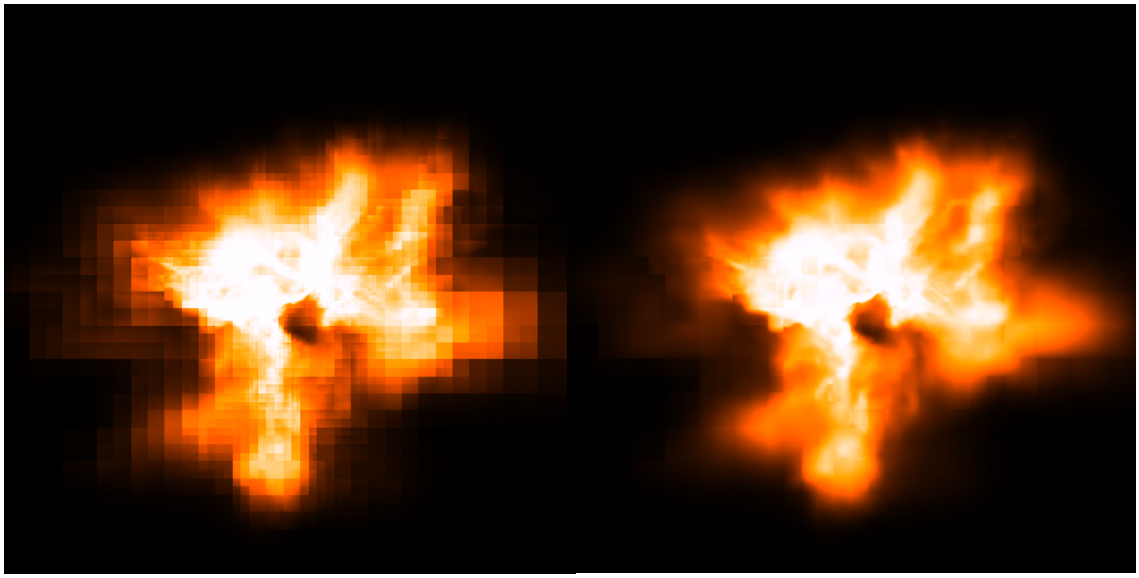


Figure 4.6: Comparison of two images produced without (left panel) and with density and pixel sub-sampling (right panel).

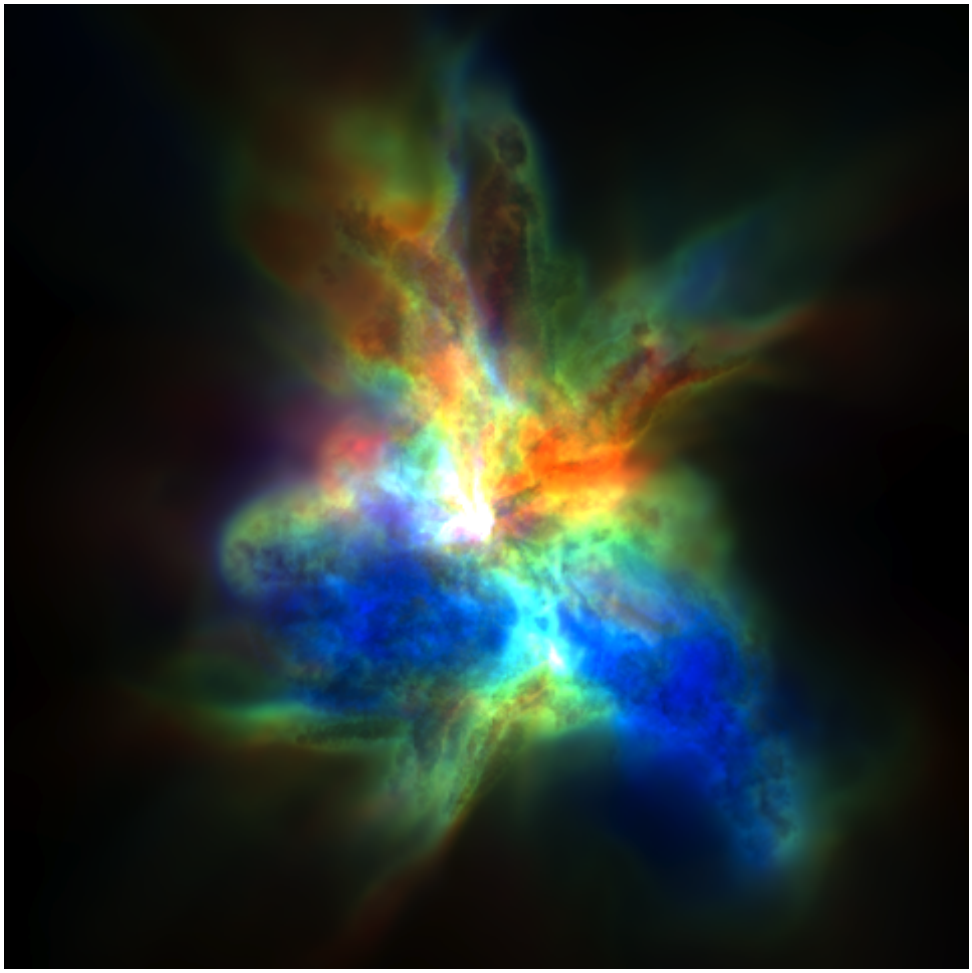


Figure 4.7: A three-colour image of a cluster observed in HCO^+ (1-0) produced using TORUS.

4.3 Application to hydrodynamical models

This section explains how the particle-to-grid algorithm detailed in the previous section has been used in different astrophysical scenarios. The algorithm has been used to convert an SPH representation of a circumstellar disc around a T Tauri Star into an AMR grid which was used to facilitate self-consistent radiative feedback calculations. It has also been used to facilitate the synthetic observation of an SPH representation of a galaxy in HI both from within the galaxy and exterior to it. The other use is fully described in Chapter 5. In all cases, the line radiative transfer code was also used to create synthetic images of the resultant grids.

4.3.1 A circumstellar disc around a T Tauri Star

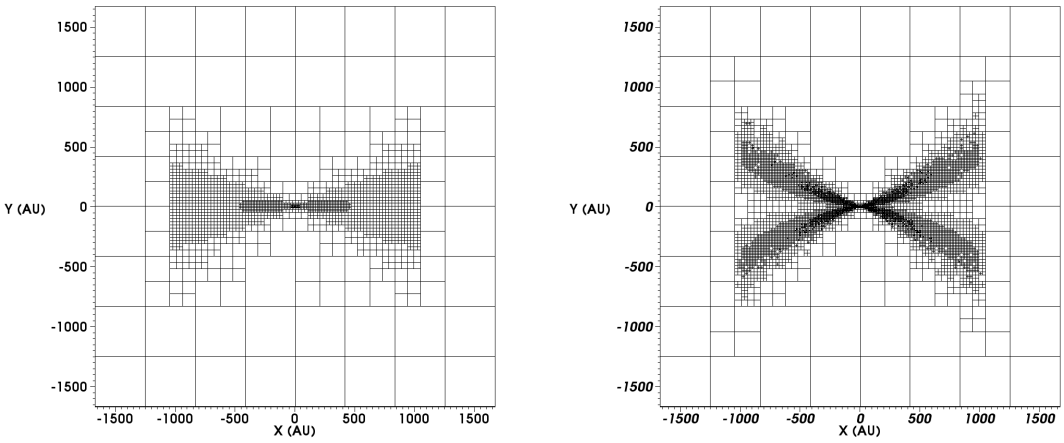
Acreman, Harries, & Rundle (2010b) presented the first use of the algorithm for coupling the grid-based TORUS radiative transfer code with the SPH code of Bate (2009a) to incorporate fully self-consistent polychromatic radiative feedback into the hydrodynamic simulation of a realistic circumstellar disc. The grid creation criteria were used to test the accurate creation of three-dimensional cartesian and cylindrical polar discs which were tested using the continuum radiative transfer benchmark of Pascucci et al. (2004). The hybrid code was subsequently used to evolve an initially vertically isothermal disc into a more physical flared disc in radiative and hydrostatic equilibrium.

Grid creation tests

In order to accurately resolve the disc it is necessary to ensure that the regions of interest are well-sampled, both by the SPH particle distribution and by the grid, that is, those regions that will contribute significantly to the shape of the disc SED.

The mass and the density splitting criteria were tested to find the optimal setting for each parameter, balancing the required number of grid cells with fidelity to the known model solution. For each splitting condition a number of AMR grids were generated using different values of the mass per cell limit or $f_{\rm split}$. This was repeated for discs represented by 10^5, 10^6 and 10^7 SPH particles.

The effects of the different splitting criteria described in Section 4.2.1 are demonstrated in Figure 4.8. A grid constructed from using a mass per cell limit of $5{\times}10^{26}$ g is depicted in Figure 4.8(a) whereas a grid with $f_{\rm split} = 0.1$ is plotted in Figure 4.8(b).



(a) Mass per cell condition of 5×10^{26} g.

(b) Density condition of $f_{\rm split} = 0.1$

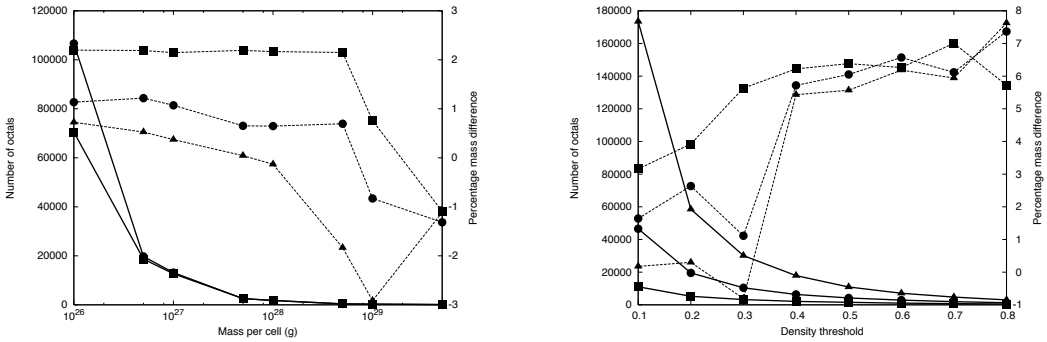
Figure 4.8: Example AMR grids constructed from 10^7 SPH particles. The grids shown are slices perpendicular to the disc mid-plane.

As expected, in Figure 4.8(a) increased resolution is seen towards the disc centre and mid-plane where the density is higher. Similarly, in Figure 4.8(b) the resolution increases towards the edge of the disc, where there are large density gradients.

Figure 4.9 plots the number of octals generated as a function of mass per cell or $f_{\rm split}$ (solid line), and the percentage error in the disc mass (dashed line). The number of octals as a function of mass per cell limit was found to be quite consistent between runs with different numbers of SPH particles. For mass per cell limits of 10^{28} g and less the total mass error was found to vary only slowly as a function of mass per cell and to be accurate to within a few percent (Figure 4.9(a)).

For the density contrast condition, the total number of octals required by the disc was a stronger function of the number of particles. In order to ensure a total mass accurate better than 1%, the density contrast condition was found to require at least 10^7 particles and a density contrast threshold of no more than 0.3.

It was noted that relatively few particles, $\sim 10^6$, were needed to obtain an accurate temperature distribution compared to the benchmark disc, offering a substantial improvement over the temperatures derived from the SPH equation of state when radiation transport dominates, however, in order to accurately model the disc SED many more particles are required – in excess of 10^8.



(a) Mass per cell condition

(b) Density contrast condition

Figure 4.9: Number of octals (solid line) and percentage error in total disc mass (dashed line) for the benchmark disc represented on Cartesian AMR grids. The grids were generated using either 10^5 (squares), 10^6 (circles) or 10^7 particles (triangles).

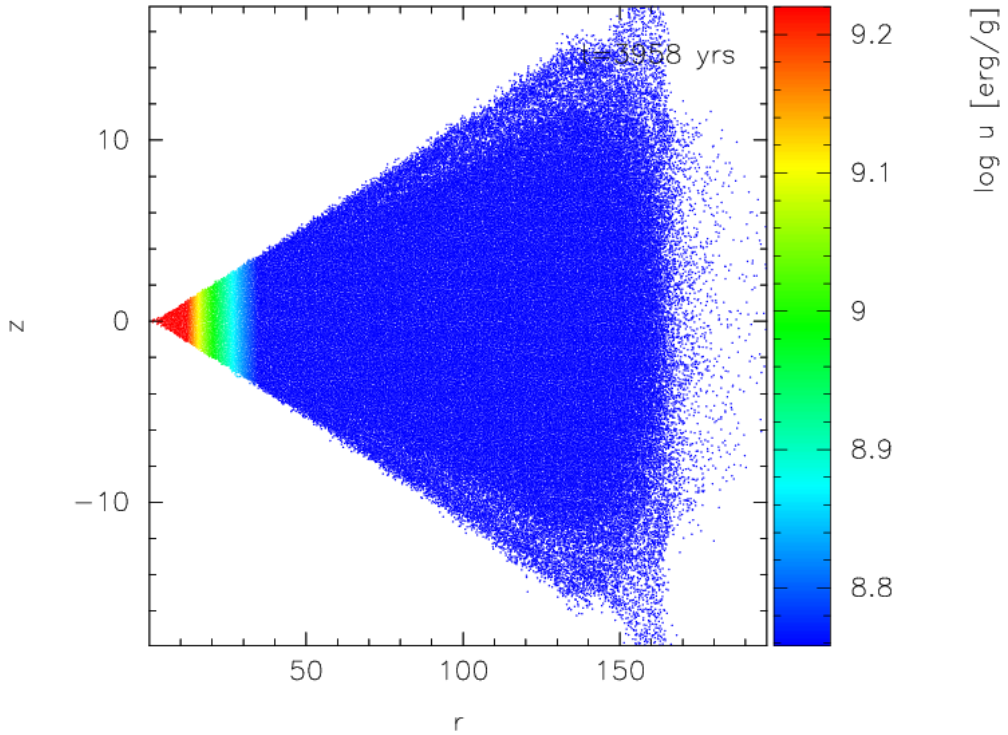
A radiation hydrodynamics simulation of a T Tauri star

Acreman, Harries, & Rundle demonstrated that not only was it possible to faithfully reproduce the SPH density distribution using AMR but also that it was possible to pass temperatures back that were sufficiently accurate for hydrodynamic calculations (under the assumption of radiative equilibrium) to the SPH code to update the gas pressure and perform the next hydrodynamical step.

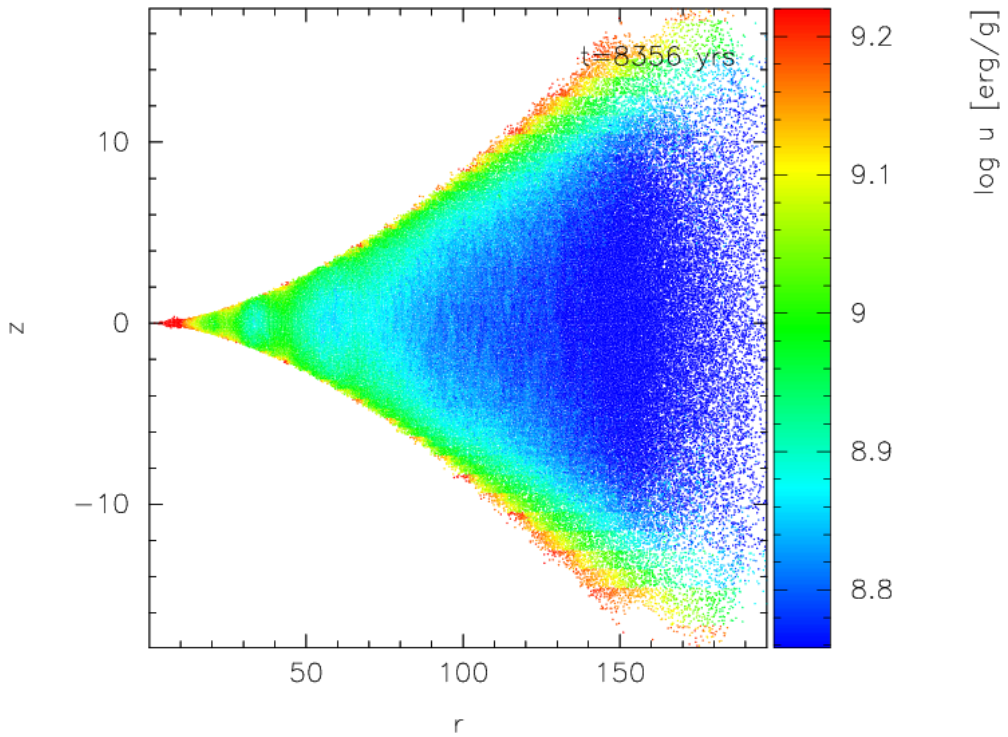
A 'realistic' test problem was devised to test and showcase the hybrid code. A model system was constructed to have physical parameters similar to those observed in the prototypical T Tauri star, AA Tau. The resultant midplane optical depths are considerably higher than those considered in the classical radiative equilibrium benchmark model of Pascucci et al. (2004), providing a stiffer test in terms of both computation time and required numerical accuracy.

A 1 $M_\odot$ star was placed in the centre of a 0.02 $M_\odot$ disc which was initialised to have an outer radius of 150 AU and an inner radius of 1 AU. Inside the inner radius particles were assumed to have accreted on to the sink particle representing the star, which was assumed to have a radius of 2 $R_\odot$, a blackbody temperature of 4000 K and a luminosity of 0.9 $L_\odot$.

10^6 SPH particles were evolved using only the SPH code to bring the disc to dynamical equilibrium. The internal energies of the SPH particles were determined using an isothermal equation of state in the vertical axis and having a 1/r radial dependence (Figure 4.10(a)). The star was then 'switched on' and the hybrid code



(a) Vertically isothermal initial state.



(b) Disc evolved self-consistently satisfying radiative and hydrostatic equilibrium.

Figure 4.10: Internal energy distribution in the initially vertically isothermal model disc (top panel) and in hydrostatic and radiative equilibrium (bottom panel).

was used to determine the radiative equilibrium temperatures (and hence particle internal energies) at every fourth hydrodynamical timestep (Figure 4.10(b)). The Monte Carlo RT calculation dominated the total run time of the model (440 hours out of 560 hours) but overall the calculation was well within feasible timeframes for such a calculation on the Exeter supercomputer and will be even more so in future. The particle-to-grid algorithm was able to transform the particles in approximately 280000 cells in around one minute so it can be concluded that it is not a bottleneck in this calculation.

The scope for use of this hybrid model is limited to geometries where faster approximations, like flux-limited diffusion are no longer valid, i.e. optically thin regions where the diffusion 'speed' may exceed the speed of light as the mean-free-path tends to infinity and an essentially arbitrary flux-limiter must be used. Moreover, flux-limited diffusion is often implemented as a grey method, typically using the Rosseland mean opacity tabulated as a function of temperature; this certainly neglects the importance of line radiation as a cooling/heating mechanism.

Finally, we concluded that although a method for constructing an AMR grid from SPH particles which was robust, in terms of conserving the total mass on the grid had been created, we found that it was necessary to add additional grid refinement in the central region of the disc to allow the inner edge to be well represented which had a large effect on the temperature distribution in the disc and on the emergent SEDs.

4.3.2 An SPH spiral galaxy simulation

In Acreman et al. (2010a) (Paper I) and Douglas et al. (2010) (Paper II), the particle-to-grid code is used to transform an 8 million particle SPH representation of a spiral-arm galaxy into an extremely high resolution AMR grid consisting of up to 6 million grid cells (Figures 4.11 and 4.13) (in Paper II the required grid resolution was greater in order to provide greater resolution in areas close to the observer).

In Paper I, a derivative of the ray-tracing algorithm presented in Chapter 2 of this thesis was used to produce synthetic spectra in the 21 cm HI line for arbitrary observing positions which were compared to observations of M31 and M33. Perturbations in constant intensity contours in each velocity channel associated with the spiral structure of the galaxy similar to those seen in the grand design spiral galaxy M81 (Rots & Shane 1975) were seen in the synthetic HI images (Figure 4.12). However,

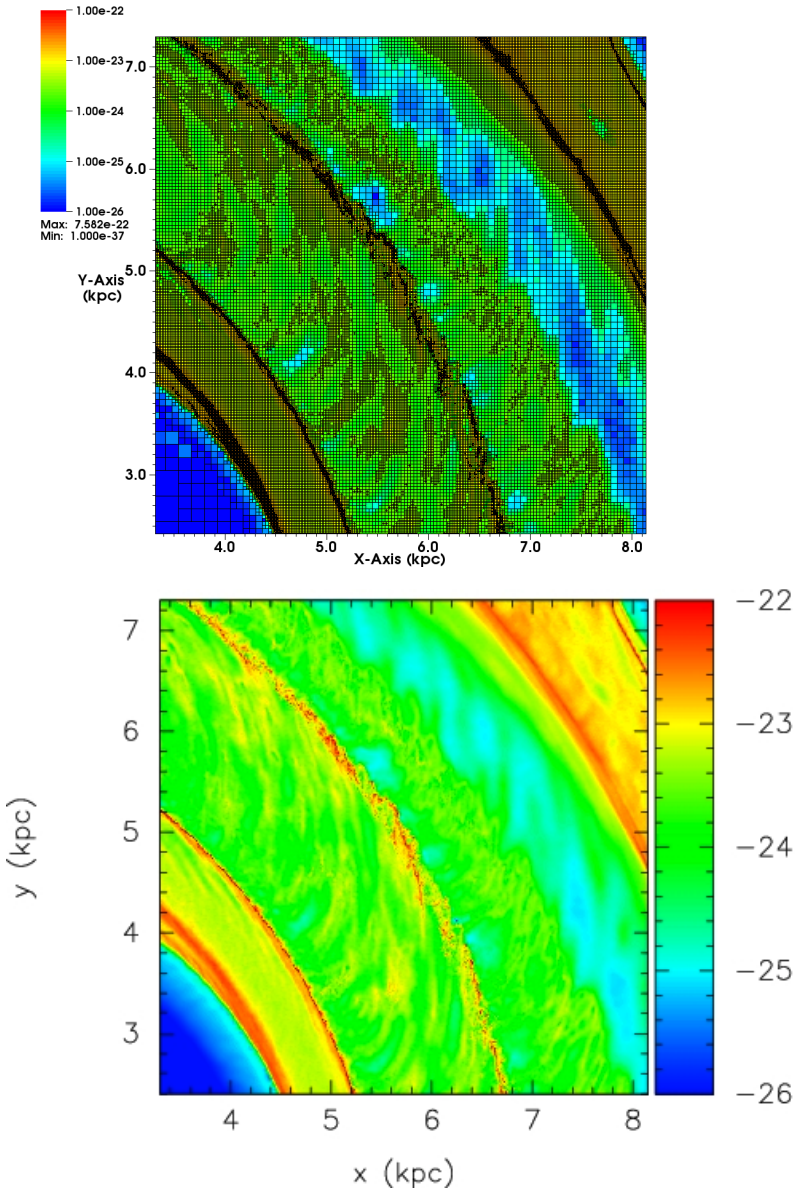


Figure 4.11: HI density (in g cm^{-3}) in the galaxy midplane, as represented on the TORUS AMR grid (top panel) and as represented with SPH particles (bottom panel). The AMR representation has the grid overplotted to illustrate the small-scale structures that can be captured using the splitting criterion.

these features are not observed in the flocculent M33 galaxy, indicating that the formation mechanism for the

The method of generating synthetic observations can also be applied to simulations in which velocity structure is generated by other mechanisms. Other mechanisms, including self-gravity and stellar feedback, can also generate velocity structure, and indeed are dominant in flocculent spiral galaxies (e.g. M33). Moreover, external influences are known to affect the HI morphology of a spiral galaxy (e.g. interactions

with a companion galaxy or high velocity clouds). As the majority of galaxies reside in groups or clusters, it is likely that external environmental effects will influence the HI structure of most real galaxies; the model galaxy simulated in Paper I was entirely isolated and the results showed idealised behaviour in the absence of external influences.

Artificially degraded images exhibited much less detail but displayed a broadening of the contours in the region of the spiral arms. The authors concluded that the simulated data showed what would be observed in an idealised case where a strong spiral perturbation dominates the structure of the galaxy.

In Paper II, the observer was placed inside the galaxy (Figure 4.13), facilitating the simulation of Galactic HI observations. A very high-resolution grid was created from 8 million particles in the region of interest, the so-called second quadrant, $l = 90 - 180°$ defined from a point 7.1 kpc from the galactic centre; this choice corresponds to a position in an inter-arm 'spur' of HI, similar to the Orion Spur of the Milky Way. By exploiting the cylindrical geometry of the galaxy, the process took less than one hour. The authors modified the ray-tracing code in TORUS in the near-field case where the assumption of paraxial rays used to generate the image breaks down. The resultant synthetic spectral cube (partly depicted in Figure 4.14) was compared to real Canadian Galactic Plane Survey data (CGPS) (Taylor et al. 2003). Moreover, by reversing the direction of the ray-trace (so that it was done from the object to the source) they were able to separate the absorption and emission contribution for each cell, facilitating the investigation of HISA (HI self absorption) in the overdense spiral arms.

The main purpose of the work was to investigate whether factors such as geometry, kinematics, chemistry and thermodynamics of a spiral shock can produce HISA, as has been observed in the Perseus arm of our Galaxy. In comparing the synthetic and observed data sets, it was possible to discern what characteristics the model galaxy had in common with Galactic H I observations.

The primary difference observed was the apparent confinement of HI gas to the plane in the synthetic datacubes, compared to the widespread and more diffuse distribution of atomic gas seen in the CGPS data. This was expected as the lack of any feedback mechanisms in the simulation mean that it would not be possible to 'stir up' the ISM. Another notable difference is the difference in scaleheights. The synthetic galaxy had a far great column density of atomic hydrogen in the mid-plane leading to greater brightness temperatures being observed (200 K (synthetic)

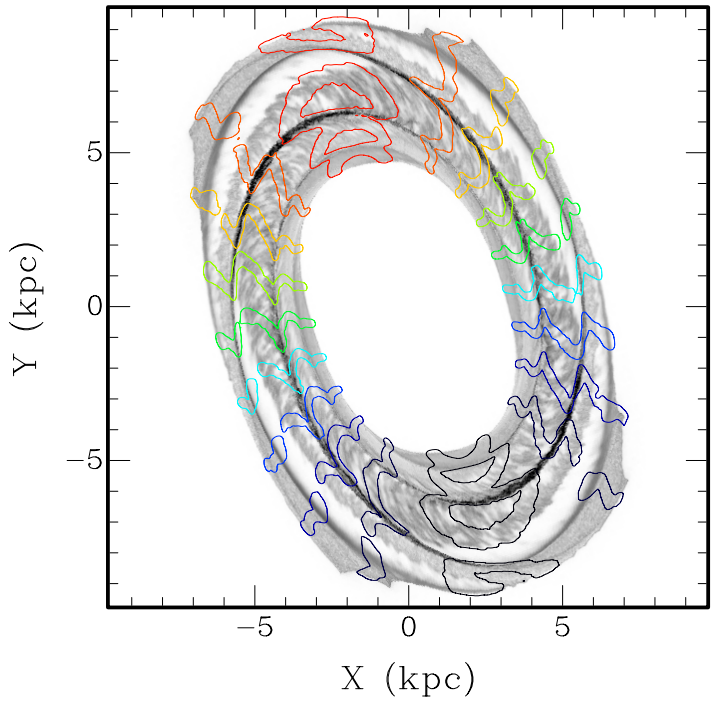


Figure 4.12: Contours of constant intensity in different velocity channels (colours) are overlaid on summed intensity of emission in simulated HI for the synthetic galaxy (greyscale).

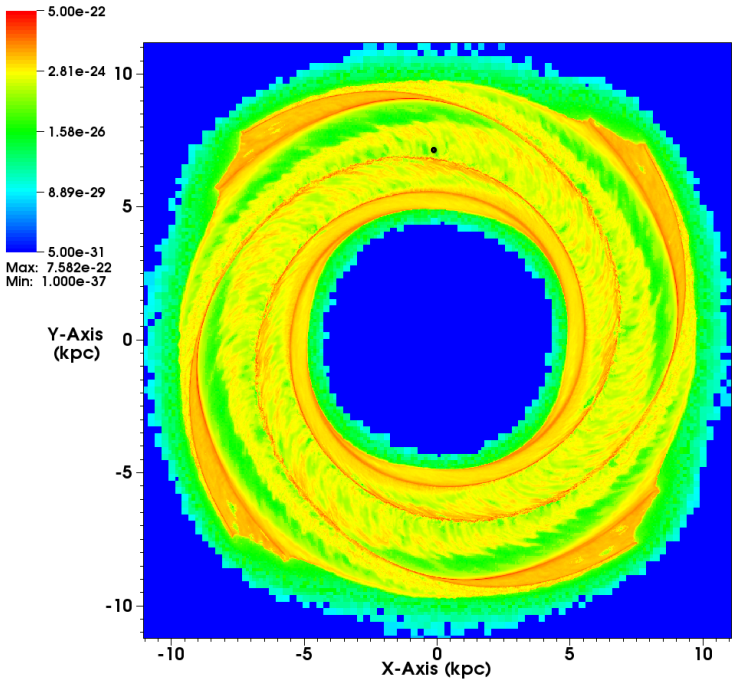


Figure 4.13: A midplane density slice of the simulated galaxy. The position of the observer in Paper II is denoted by the black circle situated between two spiral arms on the y-axis.

vs. 150 K (CGPS)) and also exhibited more HISA. The simulation did replicate the spiral structure of the Milky Way well and Δv_r (line of sight velocity) of 40 km s^{-}1 was observed (as seen in the Milky Way between local gas and the Perseus arm). It was noted that the future inclusion of feedback is likely to ameliorate both the differences seen in the original work and the authors are working to include this now.

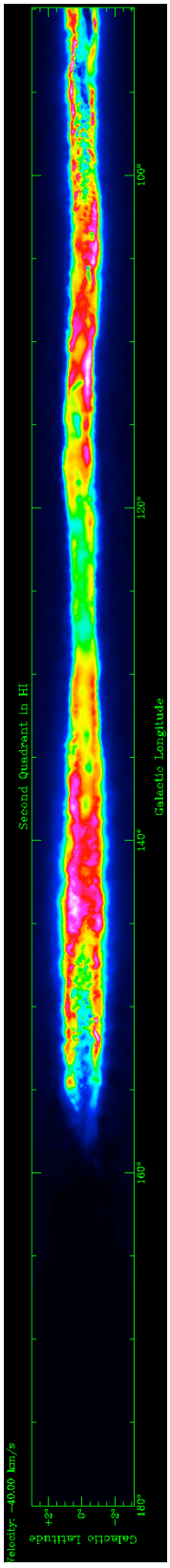


Figure 4.14: Simulated HI brightness temperature for synthetic 'Perseus Arm' gas at $v_r = -45$ km s^{-1} for the second quadrant.

To try to write a grand cosmical drama leads necessarily to myth.
To try to let knowledge substitute ignorance in increasingly larger regions of space and time is science.

– Hannes Alfven,
'Cosmology: Myth or Science?', *Journal of Astrophysics and Astronomy* (1984), **5**, 79-98

Chapter 5

Imaging a simulated star formation cluster

Large-scale hydrodynamic collapse calculations of turbulent clouds (e.g. Bate, Bonnell, & Bromm 2003a) demonstrate the fragmentation of molecular clouds into clusters which in turn typically fragment into high-density cores and subsequently fragment into protostars. These simulations predict the statistical properties of the distribution of young stellar objects such as the initial mass function, the binary fraction and the star formation rate (e.g. Zinnecker 1984; Bonnell et al. 2003; Bate et al. 2003a; Bate 2009a). It is expected that the fate of these objects is strongly affected by their evolution history, specifically the competitive accretion of gas as they move through their environs. Star formation in isolation necessarily neglects the dynamic interactions between these objects, which can cause them to be ejected at great velocities from their nascent gas-rich core, truncating their accretion (Klessen, Burkert, & Bate 1998; Bate, Bonnell, & Bromm 2002b). Consequently, it is posited that if ballistic motions are common for the dense cores likely to make stars, or that are forming stars, then it may be possible to measure these motions with precise Doppler spectroscopy using suitable millimeter-wavelength molecular lines (Walsh, Myers, & Burton 2004).

Although these simulations provide a physical explanation for many observed phenomena and enhance our understanding of cloud complexes and associated pre– and

protostellar cores, the conclusions that can be inferred from them are not readily verifiable directly from observational data. In order to make comparisons between theory and observation, given the intractability of the inverse problem, it is necessary to transform the temperature and density characteristics of a simulated cloud into a synthetic intensity map comparable with those obtained by observation.

To investigate the theory of Walsh et al. (2004), I took snapshots of an SPH star formation simulation by Bate et al. (2003a). The simulation used 3.5×10^6 particles to trace the evolution of a 50 $M_\odot$ molecular cloud as it collapsed to 50 proto-stellar objects, some still deeply embedded in their dense cores.

The paper published on this work, Rundle et al. (2010), aimed to improve upon the analysis of the relative motions of cores to their nascent envelopes in the cluster models of Bate et al. (2003a) by Ayliffe et al. (2007). Both Ayliffe et al. (2007) and Walsh et al. (2004) looked for supersonic ballistic motions of cores relative to their inchoate envelopes as evidence against competitive accretion as a clustered star formation theory. Both papers compared differences in the line of sight velocities of cores and envelopes as determined from line profiles of high– and low-density gas tracers. In Ayliffe et al. (2007) and Rundle et al. (2010) each timestep was examined for statistically significant differences in the profiles.

By performing a non-LTE radiative transfer calculation of the molecular level populations of tracers of high– and low-density gas within the cluster using TORUS (see Harries 2000; Harries et al. 2004; Symington et al. 2005a for details), I sought to improve upon the comparison between simulation and observation carried out by Ayliffe et al. (2007) and reconcile some of the differences between observation and theory.

Most of the contents of this chapter have been accepted for publication in Rundle et al. (2010). The outline of this chapter is broadly similar to Sections 5-7 of that paper. In addition to this, I have included a more in-depth consideration of the nature of clustered star formation and the use of molecular line data to observe the cores associated with clustered star formation (Section 5.1)

5.1 Star formation in clustered environments

Deep within giant molecular clouds, dense filaments of gas fragment and collapse to form star-forming clumps. These clumps are the precursors of stellar clusters and the most massive stars. They have masses ranging from 10 – 1000 $M_\odot$.

The specific formation mechanism of stars that form in clusters is not well-understood. Competitive accretion proposed by Bonnell et al. (1997) has been shown to be an effective way of simulating the formation of both low– and high-mass stars. Over a decade's worth of numerical simulations by Bate and Bonnell have shown that many statistical properties of star-forming regions can be reproduced using numerical simulations adopting this theory. Briefly, the accretion process that determines the ultimate mass of a star is split into two regimes. Initially, the graviational potential is dominated by cold gas and low-mass stars accrete their mass (slowly). Later, stars dominate the gravitational potential and high-mass stars are able to accrete mass rapidly. This dynamic stage has been demonstrated as being responsible for truncating the accretion of mass by brown dwarfs which can be ejected from the cluster following an interaction with a larger mass star.

The mechanism is able to create stars with masses up to around 10 $M_{\odot}$ but beyond this, the radiation pressure is too great to accrete more material. The mechanism can be extended by proposed stellar mergers (Bonnell et al. 1998) whereby stars greater than 10 $M_{\odot}$ can accrete mass very rapidly even where the expected radiation pressure associated with such stars becomes too great for regular accretion to occur. This mechanism has been used to successfully model the stellar IMF and multiple-star systems.

An alternative theory suggests that supersonic turbulence persists even on the scales where individual stars form (McKee & Tan 2003). Recent observational studies (Kirk 2009), however, suggest that even in dynamic cluster-forming clouds, the conditions for protostellar collapse are relatively free of supersonic turbulence.

The clusters show a lot of filamentary structure and ultimately fragment in to prestellar cores however many regimes not seen in isolated star-formation regions are observed in these dense environments, including 'hot cores', HII regions and OB-associations. Prestellar cores in clustered environments tend to be more compact ($\sim$ 0.02 pc vs $\sim$ 0.1 pc) and more dense ($n_{\mathrm{H_2}} > 10^6$ cm^{-3} vs. $n_{\mathrm{H_2}} < 10^5$ cm^{-3}) compared to isolated prestellar cores (e.g. Ward-Thompson et al. 2007a, and references therein). Their gravitationally bound masses are thought to be on the order of a few solar masses. However, they are all observed to be very cold exhibiting temperatures ranging from around 8 – 12 K.

Prestellar cores are important because they most closely represent the physical conditions prior to star formation. These cores are considered either *isolated* or *clustered* depending on the surface density of other nearby dense objects (cores or protostars).

Naturally, observations of isolated cores (e.g. Walsh et al. 2004) are simpler to interpret since the environments they inhabit are less confused. However, such

observations are not representative of star formation within embedded clusters (e.g. Walsh et al. 2007).

The use of line maps to identify and understand these cores is difficult. As discussed earlier in this thesis, line optical depths and chemical evolution can strongly affect the observed profiles. Even rare isotopolgues of CO (and others) have lines that can be too opaque to sample the inner regions of these cores. Moreover, all carbon-bearing molecules are affected by significant depletion making it harder to ascertain the relationship between the derived column densities of the tracers and of H_2. Nitrogen-bearing species are known to be better able to trace the denser parts of the cores. To partially workaround the problem of being unable to penetrate deeper into the cloud (where the gas is more dense), lines with high critical densities and rarer species such as N_2H^+which suffer little depletion are observed and interpreted with non-LTE codes. Coupled with chemical models, these codes can provide insight into the velocity differences between the protostellar cores and the surrounding cloud, to test whether cores accrete locally or move significant distances through the surrounding cloud over their lifetime.

Continuum maps are less complex but are still able to trace the cores; however, they are not able to map the kinematic profile of the cores. The determination of core kinematics is fundamental to understanding many of the dynamic processes that occur in star-formation. Clearly, the ultimate solution is to use line and continuum data synergistically to complement each other to provide better constraints for theoretical models of cores (e.g. Jessop & Ward-Thompson 2001).

5.2 The SPH cluster calculation

The SPH model used to demonstrate the efficacy of competitive accretion in replicating certain statistical features of observations by Bate et al. (2003a) can be thought of as being similar to part of the ρ Ophiuchus dark cloud, which has 6 identified dense cores (Motte et al. 1998). The SPH model forms dense cores with equivalent densities to those observed. The mass of the largest SPH core ($\sim 5.3 M_\odot$) is commensurate with the smallest core (F) in ρ Ophiuchus ($\sim 8\ M_\odot$). The model is also representative of part of the Orion Trapezium Cluster although far less massive (Bate et al. 2002a).

A 0.375 parsec, 50 $M_\odot$ cloud was created to be initially spherically symmetric with a uniform density and temperature (10 K). A supersonic turbulent velocity field ($\mathcal{M} = 6.4$) was imposed on the cloud to create anisotropies which ultimately led to

the formation of a dense pre-stellar core after approximately 1 free-fall time. Here, $t_{\rm ff} \approx 1.9 \times 10^5$ yr. The velocity field was normalised so that the kinetic energy of the turbulence equalled the magnitude of the gravitational potential of the cloud.

During the early stages of collapse, the relatively tenuous gas allows energy to be radiated with ease. At this stage the gas was assumed to be isothermal. If the rate at which energy is released exceeds the rate at which the gas can cool then the local environment begins to heat rapidly, because of the dependence of this point on gas opacity this is the so-called 'opacity limit for fragmentation'. Masunaga & Inutsuka (2000) showed that the gas begins to heat significantly at a density of $\rho > 10^{-13}$ g cm^{-3} depending on gas opacity (and initial temperature). To avoid performing a computationally expensive full radiative treatment, Bate et al. used a barotropic equation of state $p = K\rho^{\eta}$. The barotropic exponent, η, equals unity where the gas is isothermal (for $\rho \leq 10^{-13}$ g cm^{-3}) and increases to 7/5 for $\rho > 10^{-13}$ g cm^{-3}. This equates to a temperature-density relationship of

$$T_{\rm gas} = \max\left(10, 10\left(\frac{\rho}{10^{-13}}\right)^{2/5}\right). \tag{5.1}$$

As the gas heats up, a previously Jeans-unstable collapsing clump (in the isothermal regime) becomes stable once again forming a pressure-support fragment. Following the coalescence of these fragments becomes computationally impractical once their density increases beyond a certain level owing to the short dynamical timescales involved in following their evolution further. Consequently, when the local simulated density in a particle exceeds 10^{11} g cm^{-3}, a *sink particle* is created with a mass equivalent to the sum of the masses of all particles within a 5 AU radius (e.g. Bate et al. 1995). The sink particle inherits a weighted average of all the properties of the contributing particles and they are destroyed and interacts with the gas only via gravity and accretion. This ameliorates the computational slowdown caused by resolving the fragments as they continue to accrete more mass.

The minimum mass for a sink particle is $\sim$ 10 Jupiter masses (M_J). They will continue to accrete gas particles throughout the simulation to become brown dwarfs (< 75 M_J) or stars (> 75 M_J).

Finally, although the cloud initially contained enough turbulent energy to support itself against gravity, turbulence decays on the dynamical time-scale of the cloud and star formation began after approximately one global free-fall time. This rapid decay of the turbulence is consistent with other numerical studies of turbulence in molecular clouds (e.g. Offner et al. 2008a) but studies using driven turbulence are also starting to be performed (ibid).

The calculation was evolved for $\approx$ 69000 years after star formation began until $t = 1.40t_{\rm ff}$. At the time it was one of the largest SPH simulations ever performed. It required approximately 95000 CPU hours on the SGI Origin 3800 at the United Kingdom Astrophysical Fluids Facility (UKAFF). A similar calculation was repeated on the University of Exeter supercomputer in 2008 which took just weeks, highlighting the improvement of modern-day computational facilities over past facilities.

5.3 Creating synthetic images of an SPH cluster

Five snapshots were used to investigate the effect of time on the core velocities. They are taken at 1.0, 1.1, 1.2, 1.3 and 1.4 $t_{\rm ff}$; the first snapshot occurs just before the start of star formation, the second is taken during the first star formation episode, while the third and forth are taken during an interstitial period during which a second and third star-forming core is formed. The final snapshot is taken at the end of the simulation when over 50 objects have been formed, some of which are still accreting. At the end of the simulation, approximately 1/5 of the SPH gas particles (≈ 10 M$_\odot$) have been accreted onto the sink particles.

Each SPH snapshot was discretised using the algorithm outlined in the previous chapter using the same splitting criteria for each timestep; no more than 50 particles were allowed within one cell, the ratio of the maximum density to the minimum density must be no greater than 2 and the range of velocity magnitudes must be less than $5v_{\rm turb}$.

The process of discretisation will necessarily introduce some further uncertainty in the exact distribution of mass throughout the cloud, however, as discussed above, the algorithm has been designed to minimise any systematic error which may arise through using too coarse a grid to represent the space. The total mass stored on the grid in most cases is found to lie within 5% of the true mass.

In this application, the mass discretisation error is better than 3% and in the absence of overdense regions in later timesteps, better than 1% (see Table 5.1). Furthermore, a comparison of the number of cells in each level of refinement against a histogram of the smoothing lengths of the particles (Figure 5.1) shows broad agreement, except in the highly refined tail of the AMR cell distribution. This is because of the additional constraints required for adequate radiative transfer, additional velocity splitting in particular. Over all 5 frames, the maximum level of refinement was 22, resulting in

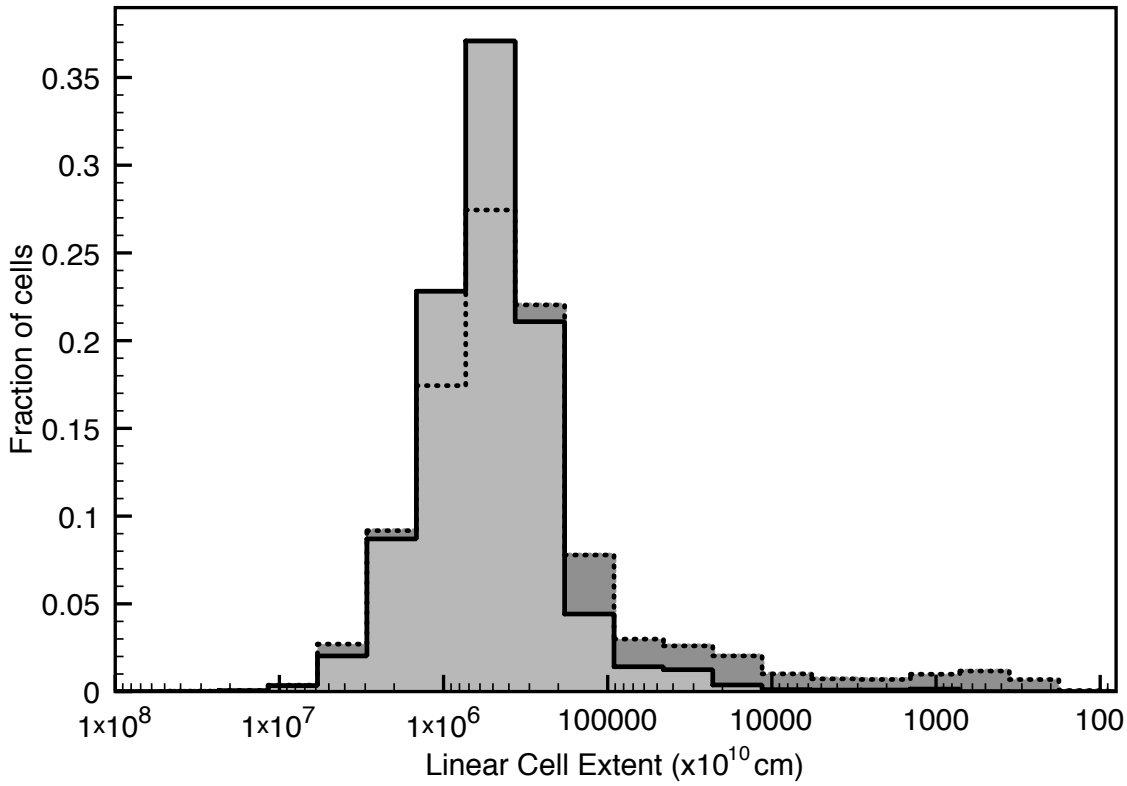


Figure 5.1: Histogram of the relative fraction of AMR cells (dark grey) at each depth for $t = 1.1t_{\rm ff}$. The smoothing lengths of the SPH particles (light grey) were binned according to their closest association in log-space. Note that the x-axis is reversed to show the proportion of cells at lower depths on the left.

the smallest octal having a linear dimension of ~ 0.08 AU, commensurate with the shortest inter-particle spacing. On average, this discretisation policy resulted in a ratio of particles:cells of approximately 1/3.

Owing to the complex, filamentary structure of the SPH particle distribution, the particle-to-grid algorithm took longer than in the disc applications described earlier. Each snapshot took between 12 and 24 hours to generate on a single processor

Table 5.1: Table of SPH properties and corresponding AMR grid properties for each timestep.

	SPH properties			Grid properties				
Timestep	$N_{\rm part}$	$N_{\rm pt}$	$\rho_{\rm max}$	$N_{\rm octals}$	Depth	Mass ($M_\odot$)	Mass Error %	$N_{\rm part}$ /$N_{\rm octals}$
1.0 $t_{\rm ff}$	3500000	0	2.18(-16)	1097622	14	50.43	-0.850	3.19
1.1 $t_{\rm ff}$	3455362	8	7.65(-10)	1118601	21	50.18	-0.369	3.09
1.2 $t_{\rm ff}$	3287092	18	2.49(-09)	1112357	23	49.42	1.16	2.96
1.3 $t_{\rm ff}$	3248783	27	2.92(-09)	1174818	21	49.14	1.73	2.77
1.4 $t_{\rm ff}$	3087719	50	3.17(-09)	1151781	20	48.67	2.66	2.68

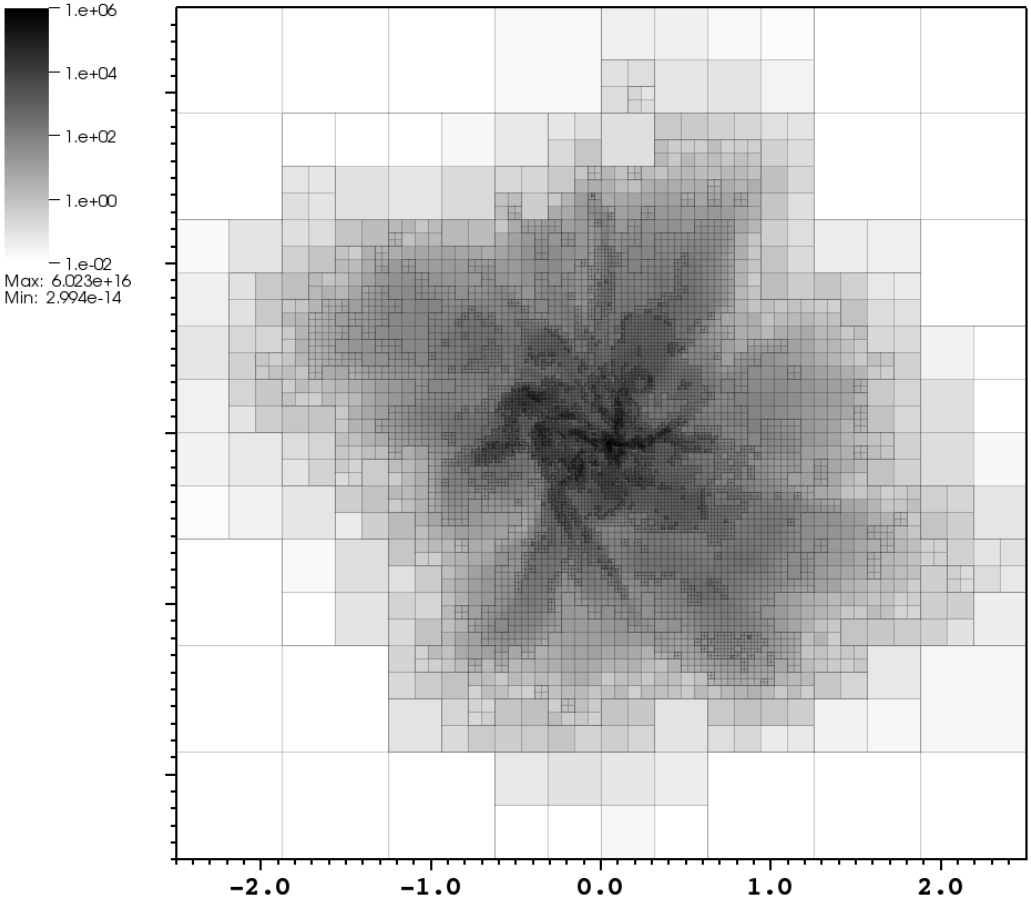


Figure 5.2: A slice through the $xy-$plane illustrating the dynamic range of densities that can be captured within the cloud using AMR. Thirty orders of magnitude are present in the simulation with densities reaching as high as 2×10^{-7}g cm^{-3} and as low as 5×10^{-36}g cm^{-3} (although very little material by mass is actually present below $\sim 10^{-25}$g cm $^{-3}$). The greyscale for this plot is given in $n(H_2)$.

machine depending on the complexity of the SPH particle distribution. In order to produce accurate gridded velocities and densities, $h_{\rm crit}$ (see Section 4.2.2) was taken at the 95% level. It is expected that this could have been reduced with minimal effect on the final conclusions of this work.

Figure 5.2 illustrates how an SPH representation of density can be discretised onto a mesh.

5.3.1 Molecular line analysis

Our analysis of this cluster simulation seeks to examine the conclusions of Ayliffe et al. (2007), that the dynamical picture of star formation provided by the SPH hydrodynamics simulation and others like it are not invalidated by the observations of Walsh et al.

We do this by following the analysis methods employed by Walsh et al. (2004). Accordingly, the ^{13}CO (1-0) transition (ν_0 = 110.2013542798 GHz) was used to trace low density envelopes. Initially, an abundance of [^{13}CO] = 10^{-6} relative to H_2 was assumed. To trace gas closer in to the cores within the cluster, the C^{18}O (1-0) transition (ν_0 = 109.7821734 GHz, [C^{18}O] = 10^{-7}) was used and to define cores the N_2H^+ (1-0) transition (ν_0 = 93.1737 GHz) was used because of its high critical density ($n_{\rm crit} \approx 1.4 \times 10^5$). For N_2H^+ a constant relative abundance of 1.5×10^{-10} was assumed in line with values obtained for well-studied cores in the Taurus-Auriga molecular complex (Tafalla et al. 2004). For N_2H^+ this is a good first-order approximation for the core but is not necessarily true for the wider envelope (although this is ameliorated by the fact that much of the gas outside the core is too diffuse to emit strongly in any case); a more sophisticated study would take into account the spatial variation noted by Tafalla et al. (2004) as well as considering the variation in N_2H^+ abundance for low and high mass cores (Zinchenko et al. 2009).

For all molecules, a non-thermal turbulence parameter with a fullwidth of 0.3 km s^{-1} $\sim c_s$ for H_2 at the simulated temperatures was assumed.

The ^{13}CO, C^{18}O and N_2H^+ molecular data was superimposed on to the discretised grid for a total of 15 distinct grids (3 molecules and 5 timesteps). A non-LTE solution was derived for each grid using the molecular line transfer code described in Chapter 2. Table 5.2 shows the mean error of the least well converged level over all grid cells. A further condition that every cell had to have an RMS error of less than 1% in $J = 0$ and $J = 1$ was imposed to reduce any pixel-to-pixel variance in intensity in the subsequent raytracing of the grid.

The computation of the level populations for all the grids took approximately 100000 CPU hours. Each individual grid was converged to the required level within 240 real hours (the maximum allotted wall time for each simulation) using upto 48 processors. Depending on the tracer, between 12800 rays and 102400 rays were used to converge the grid in the random-ray stage of the code. The quickest calculation took approximately 6000 minutes. The longest took almost exactly 15000 minutes (240 hours).

Table 5.2: Error (in percent) of the least well converged level in each grid for each timestep.

	Species		
	^{13}CO	$C^{18}O$	N_2H^+
1.0 t_{ff}	0.30	0.14	1.03
1.1 t_{ff}	0.27	0.19	1.98
1.2 t_{ff}	0.25	0.07	0.54
1.3 t_{ff}	0.16	0.12	0.59
1.4 t_{ff}	0.14	0.15	0.71

To improve the statistics of our analysis, each of the 15 grids were observed from 20 equally spaced positions on a sphere with radius 140 pc. Care was taken to maximise the separation of each observer position giving equal coverage of the viewing sphere from each perspective. Although these viewing positions do not provide independent observations, they are necessary to perform a statistical analysis of the velocity differences on a large enough sample set. It is common to take the six cardinal directions as viewing locations for simulated data (e.g. Offner & Krumholz 2009) and this is an extension of this method of extending datasets that are 'expensive' to acquire. By extension, the same caveats that apply to the conclusions based on that method for creating multiple datasets from one apply to this work also.

A $1024 \times 1024 \times 80$ element datacube was created for each observer position-tracer-timestep triplet. This work therefore contains 300 distinct datasets upon which to perform statistical analysis. The maps have a linear spatial resolution of 2×10^{15} cm or $\sim$133 AU per element. This is equivalent to a spatial resolution of ~ 1 arcsec at 140 pc. From the known velocity distribution of the SPH particles, the emission was expected to come from a range of velocities within 3.2 km s^{-1} of the line centre giving a velocity resolution of 0.08 km s^{-1}.

In these grids, molecular abundance has been assumed to remain constant in both time and space. Without a detailed evolution history, it is not possible to predict with any certainty the local chemistry in any region of the grid. In Section 5.3.2, a simple chemical model to take into account the expected CO freezeout in the cloud is described. In the absence of better chemical modelling it is assumed that the abundance of N_2H^+ remains constant, which is appropriate given the age of the cluster (Bergin & Langer 1997).

5.3.2 Chemistry

Molecular adsorption onto dust grain surfaces is known to be an effective depletion mechanism in many carbonaceous species at densities of $n(H_2) = 3 \times 10^4$ cm^{-3}

(Bacmann et al. 2002). At higher densities, the depletion is more than an order of magnitude greater (Caselli et al. 1999; Bergin et al. 2002) – Crapsi et al. (2005) found the average depletion of CO in starless cores to be ~ 10. Notable absentees in this list are N-containing species including N_2H^+. $C^{18}O$ is expected to trace a larger scale than N_2H^+ not only because of its lower critical density but also because it effectively freezes out at densities above $\sim 3 \times 10^4$ cm^{-3}. Therefore, in the cluster, $C^{18}O$ intensity is weighted towards the outer regions while N_2H^+ traces the denser inner parts.

Drop profile

A 'drop' profile was used to test the effect of CO chemistry, as described in Jørgensen (2004). In its simplest implementation, the model assumes a constant, undepleted abundance of CO relative to that of H_2,X_0, where $T_{\rm dust/gas} \geq T_{\rm evap}$ or $n(H_2) < 3\times10^4$ cm^{-3} combined with a depleted abundance of X_D, elsewhere. Using equation (5.1), it is possible to reformulate the model solely in terms of the local gas density, ρ,

$$X(\rho) = \begin{cases} X_D & \text{for } 10^{-19} < \rho < 1.56 \times 10^{-12} \text{ g cm}^{-3}, \\ X_0 & \text{elsewhere,} \end{cases} \tag{5.2}$$

Empirically, the drop profile has been used to improve the fit of gaussian line profiles to low J observational $C^{18}O$ data of various class 0, class 1 and pre-stellar objects. The improvement in the goodness of fit is found to be sufficiently significant as to provide strong evidence for its acceptance over the constant abundance model. Moreover, the model has its basis in observations (Caselli et al. 1999; Tafalla et al. 2002) where a freeze-out zone has been directly imaged.

Physically, the profile is explained in terms of radii. The material within the inner radius is expected to be gas-phase owing to its proximity to the warm core. The temperature at which CO is expected to start returning to the gas-phase is around 30 K with a non-linear increase in desorption rate occurring up to $\sim 160K$ (Collings et al. 2004). Beyond the outer radius, at densities lower than 3×10^4 cm^{-3}, the depletion timescale due to freezeout is thought to exceed the lifetime of the core and is thus assumed to remain at an undepleted abundance, to reduce the number of free parameters.

Since the critical densities of N_2H^+ are higher than those of the CO isotopologues, they are less sensitive to the outer region of the envelope where depletion and contribution from the surrounding cloud may be important. It has been posited (Jør-

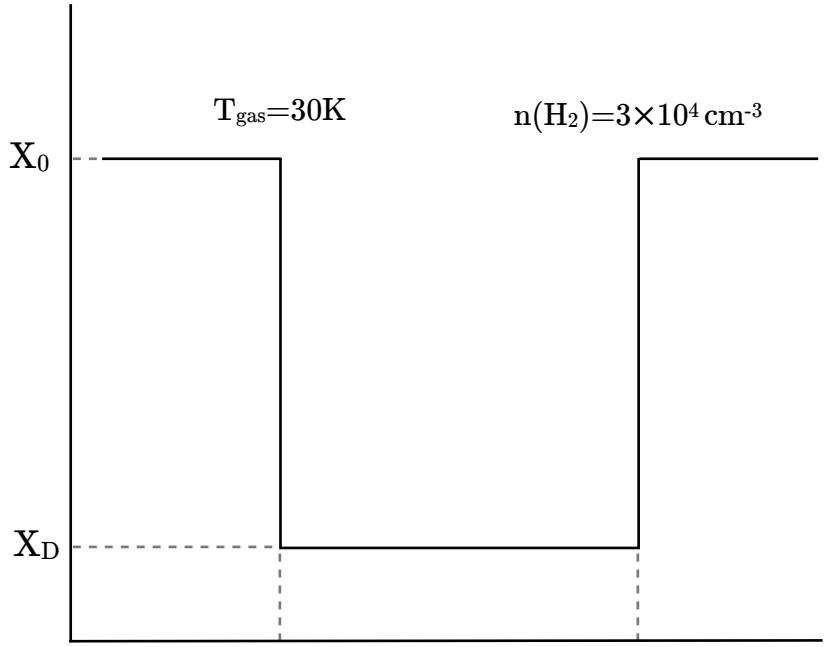


Figure 5.3: CO chemistry drop profile.

gensen 2004) that this may explain why these transitions can be modelled assuming a constant density.

In the runs taking chemistry into account, the same method as outlined in Section 5.3 but with depleted densities one order of magnitude lower in the intermediate region, viz. $[^{13}\mathrm{CO}]_D = 1 \times 10^{-7}$ and $[\mathrm{C}^{18}\mathrm{O}]_D = 1 \times 10^{-8}$ was used.

5.4 Results

In order to compare the core and cloud velocities, line profiles for each each molecular species were extracted. For each synthetic observation the intensity datacubes had their background intensity subtracted, $B(\nu_0, T_{\mathrm{cmb}})$, and were summed over all velocity channels to give an integrated intensity map. The maps were converted to brightness temperature, T_b using the Rayleigh-Jeans approximation (see equation 1.38).

We used WAVDETECT, part of the CIAO data analysis system written for the Chandra X-ray Observatory, to detect cores in the N_2H^+ intensity maps. The WAVDETECT tool (Freeman et al. 2002) correlates the image with wavelets of different, user-defined scales and then searches the results for significant correlations. This method is very similar to the difference of Gaussians (DoG) method, where a difference map is created by convolving an image with two Gaussian distributions with

different spatial variances and subtracting the resultants, preserving structure with lengthscales between the two variances. WAVDETECT produced very similar results but was slightly better at detecting objects that it was not possible to resolve using DoG and was more robust against the converse case of rejecting objects that upon later inspection had distinct line profiles (often indicative of containing protostars).

We looked for cores with lengthscales of $2-64$ pixels ($\sim 0.001-0.04$ pc) and applied a threshold of 50% of the global maximum to the transformed image to determine the membership of a pixel to a core. The range of lengthscales over which core-like features were searched for was determined by testing different values (limited by the fact that the limits must be a power-of-2) and examining the results. In actuality, the number of objects found by this method was approximately constant for most ranges. The limits that were settled upon are smaller compared to typical beam sizes at typical distances and thus smaller than the size of observationally extracted cores (upto 0.1 pc).

A recursive blob counting algorithm was invoked to determine the number of separate regions identified as containing a local maximum within the map; for N_2H^+, each region is a potential core and for ^{13}CO and $C^{18}O$ each region defines the envelope which surrounds the core. This is similar to the method of Offner, Krumholz, Klein, & McKee (2008b) in that we generate a background map and then detect local intensity peaks above that.

Any core candidate that lay wholly or partially outside the enveloping material as determined by the low density contour was excluded from the analysis. In this constant abundance analysis, it is expected that the regions should be strict subsets of each other. For each core candidate, a mask was applied to the original datacube corresponding to the region defined by the 50% contour and line profiles for that region were found by integrating over all pixels in the region for each molecular tracer.

Note that in this work, we have performed the spectrum extraction on the original, non-degraded synthetic observations. Convolution with a beam and the addition of a source of noise may be added in a future analysis of the data. It is expected that by convolving the data with an instrument function, the regions identified as having high-density gas will be smeared out over a wider area and that the point of peak intensity will have a reduced magnitude. This leads to an additional increase in the identified core size, as the 50% threshold is reduced compared to the non-degraded case and the emission from more lower-density gas is identified as belonging to the dense core.

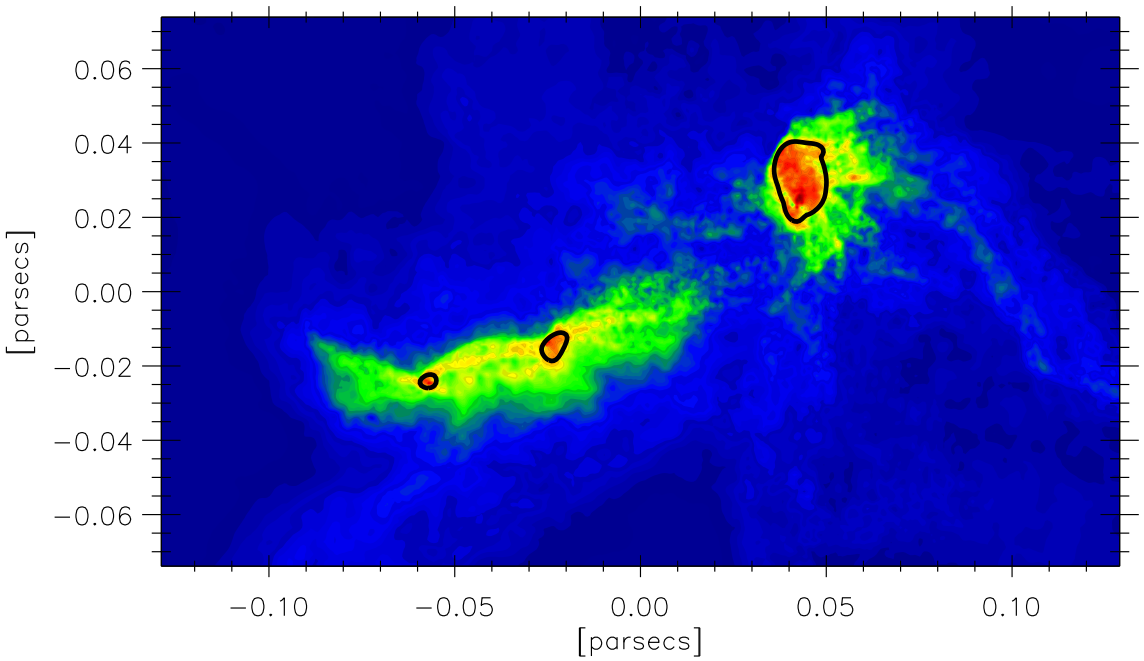


Figure 5.4: Integrated intensity map in N_2H^+ (1-0). Contours have been added to demarcate each of the cores identified. The line profiles of the two largest cores and those of the enveloping gas are illustrated in Figures 5.5 and 5.6 respectively. Note that this image only shows the region of interest in the cloud. The extent of the entire cloud is ≈ 0.6 pc.

Two methods were used to define the centre and spread of the extracted line profiles. In one method, the line profile was fitted with a Gaussian distribution using a Levenberg-Marquardt algorithm. The initial Gaussian was set to have a peak brightness temperature of 5 K, $\sigma = 1$ km s^{-1} and to be centred at $v = 0$ km s^{-1}. Each datum point was assumed to have equal weighting. The centre of the distribution was taken to be the mean of the fitted Gaussian and the spread was parameterised by the standard deviation of the Gaussian.

In addition to characterising the line profile using parameters derived from the fit, the mean and median of the distribution were directly calculated, providing a robust method to verify the position of the peak in the Gaussian distribution determined by fitting. Similarly, the full-width at half-maximum (FWHM) of the profile was recorded to parameterise the spread. The same process was repeated for the low density tracer and the resultant difference in line centre velocities was assumed to represent the systematic shift in line-of-sight velocity. Figures 5.4, 5.5 and 5.6 illustrate the results of the analysis for one particular observation at 1.4 t_{ff}.

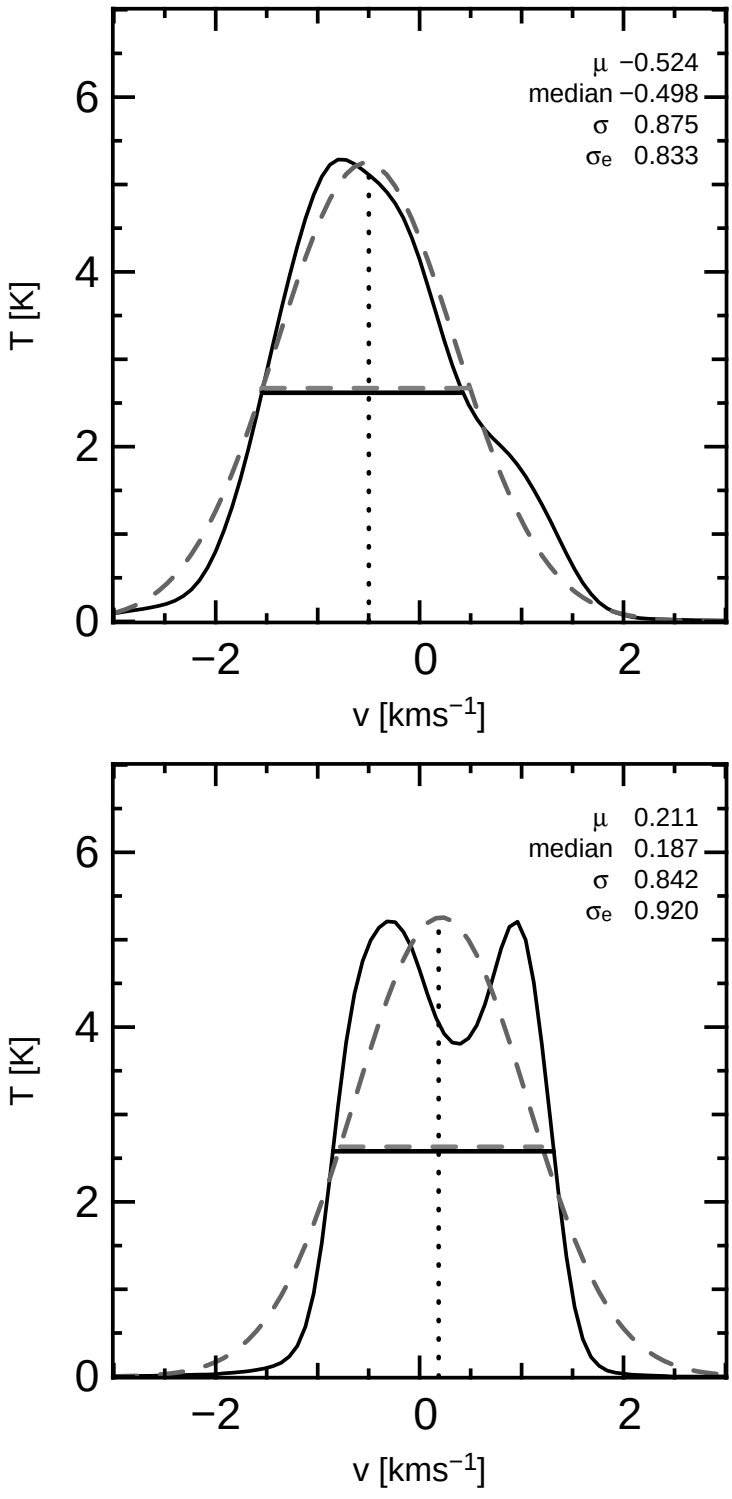


Figure 5.5: Typical high-density core line profiles traced using N_2H^+. Dashed lines indicate the Gaussian profile fit and FWHM. Dotted line indicates position of the Gaussian line centre. The legend also displays median line centre and the equivalent standard deviation, σ_e, from the FWHM of the profile. (b) shows a profile that it would be possible to fit two Gaussians to (assuming two cores along the line-of-sight) however by examining the SPH particle distribution this profile is known to be caused by rotating gas surrounding a multiple system.

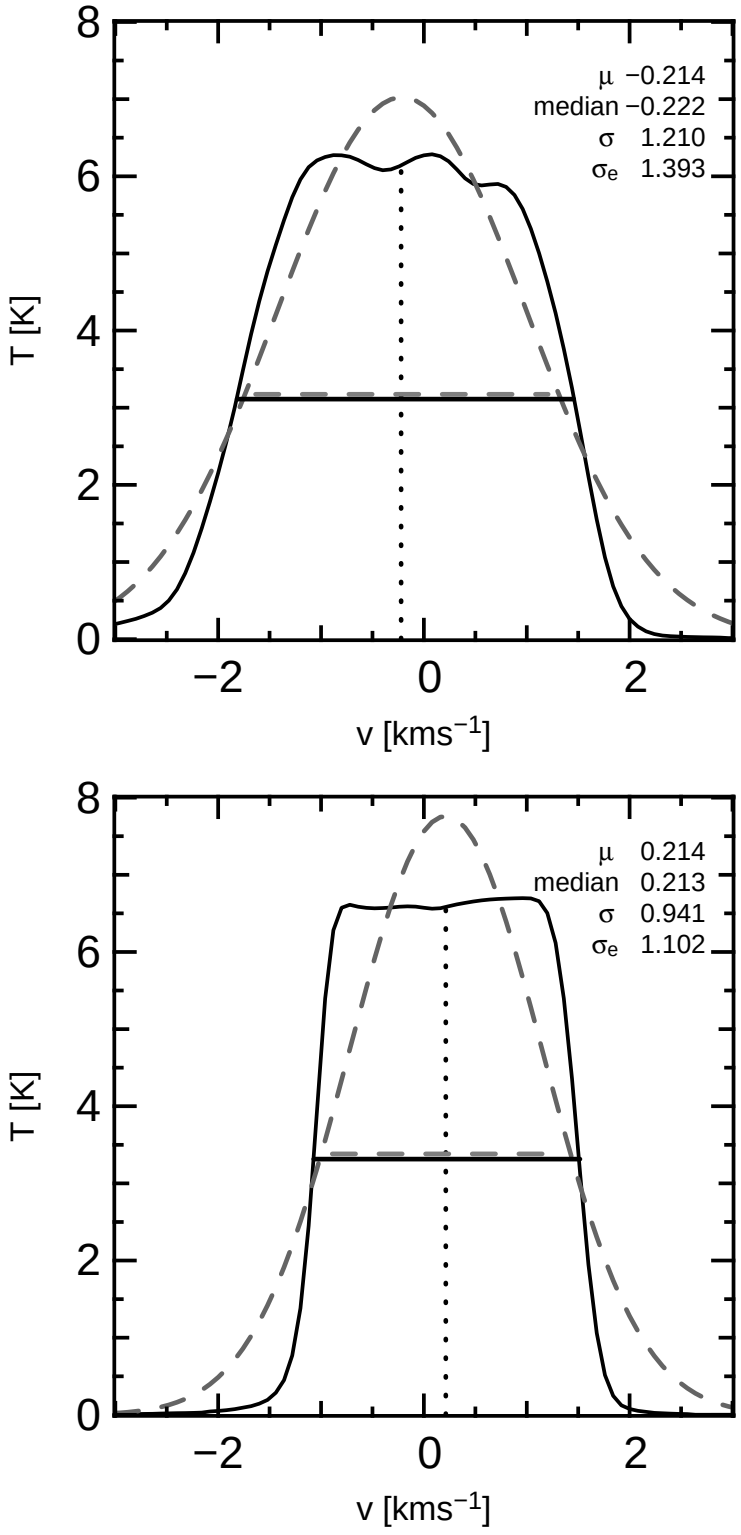


Figure 5.6: Typical low-density envelope profiles traced using ^{13}CO. Both profiles exhibit flat peaks associated with the saturation of optically thick ^{13}CO (1-0) line.

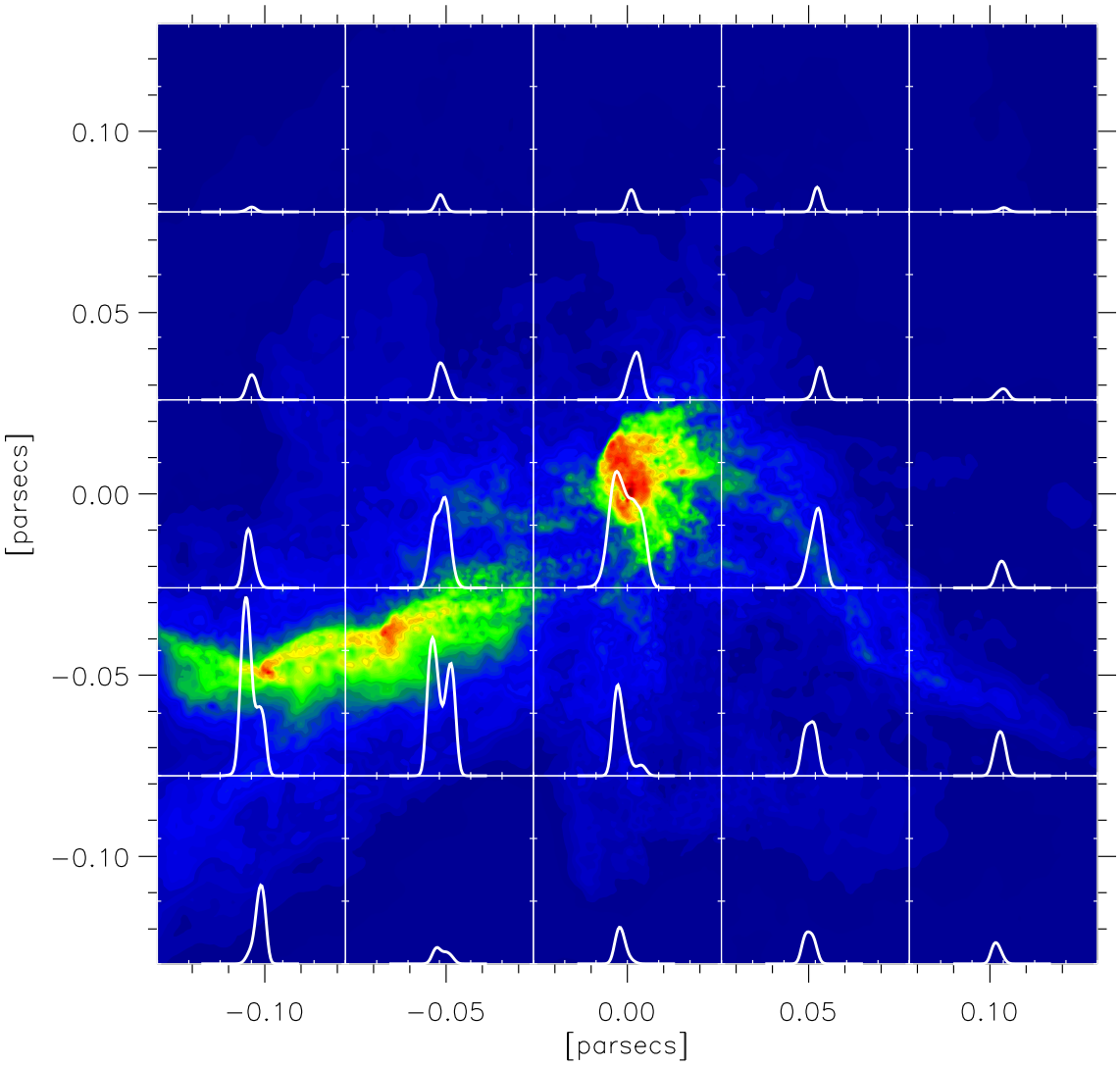


Figure 5.7: The line profiles of the square regions are superimposed onto an N_2H^+ (1-0) integrated intensity map (see Figure 5.4). The regions containing the three cores have somewhat broader line profiles and exhibit more structure (i.e. double peaks caused by rotation and infall) than the optically thin gas surrounding them.

Having removed the two 'cores' identified by WAVDETECT that did not lie within the 50% low-density envelope we also removed 11 cores (over the 5 timesteps) which were judged to be spurious detections which were typically characterised by very small areas (< 10 pixels), similar velocity profiles to a larger nearby detection and often did not contain any sink particles. This left 169 cores over the 5 timesteps; the fewest detected cores (22 from 20 observing positions) being in the first timestep before the creation of any sink particles and the most (58) being detected in the last timestep, where three unique star-forming regions were detected by Bate et al. (2003a). Every remaining valid core-envelope pair was analysed and the results were collated to allow us to perform a statistical analysis according to the strategy

outlined in Section 5.3.1. Cores that contained one or more sink particles were judged to be protostellar cores while cores without sink particles are categorised as being prestellar (or starless).

Following Ayliffe et al. (2007), we do not present the results of Gaussian fits in this work. All analysis was done using both methods and the differences between methods of analysis were sufficiently small that they did not affect any conclusions drawn upon the results of the work. Moreover, the profiles of ^{13}CO data were rarely found to be Gaussian, exhibiting flat, wide peaks (owing to their large optical depth). As such, we characterise the dispersion using the FWHM / $2\sqrt{2\ln(2)}$ of any given line profile and take the line centre to be the median, m, of the distribution ($\int_{-\infty}^{m} f(x)\ \mathrm{d}x = 0.5\ \int_{-\infty}^{\infty} f(x)\ \mathrm{d}x$). As a non-parametric estimator of the line centre, the median does not assume normally distributed data; as a result the median is less influenced by shoulders, long tails, noise etc. and may give a less biased estimate of the true line centre. Consequently we did not attempt to fit multiple Gaussians to the data (with the assumption that a double-peaked distribution indicated two cores along the line-of-sight) primarily because it was possible to confirm from the SPH particle distribution that this was not the case but also because it allows us to more directly compare our results with previous studies. The presence of a doubly (or more) peaked profiles is caused by the velocity shift associated with the rotation and/or infall of gas. In order to illustrate these effects we have calculated profiles from 25 areal bins around the cores shown in Figure 5.4 and plotted them over an integrated intensity map (Figure 5.7). Whilst the optically thin gas is associated with single-peaked profiles, the denser cores display a more complex line profile morphology. A similar variety of profile shapes have been found in one-dimensional simulations of collapsing and rotating cores (Pavlyuchenkov et al. 2008).

Figure 5.8 shows a histogram of the distribution of relative velocity difference, defined as the difference between the medians of the line profiles for ^{13}CO and N_2H^+, for each of the 300 observing positions of the cores and their associated low density gas envelopes. The standard deviation of the histogram for all cores (both pre– and protostellar) is $\sigma = 0.16$ km s^{-1}. In the same figure, we plot three normalised Gaussian distributions whose standard deviations are equal to that of the equivalent standard deviations as derived from the mean FWHM of the line profiles for each tracer molecule. They are notably more broad than either the relative velocity difference or the sound speed.

Figure 5.9 illustrates values for the mean velocity dispersions of each tracer for each timestep. The trend for the CO isotopologues is the same; that the mean velocity

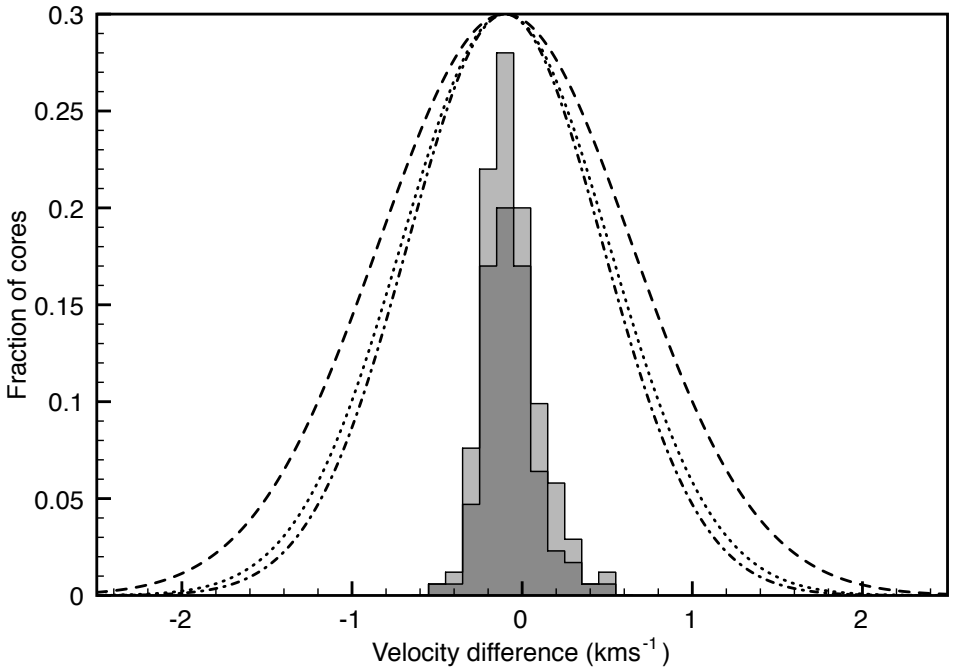


Figure 5.8: Histogram of velocity difference of all cores (combined starless and protostellar cores) identified at all time steps (light grey, $\sigma = 0.16$ km s^{-1}) and protostellar cores (dark grey, $\sigma = 0.18$ km s^{-1}). Also plotted are the velocity dispersions for N_2H^+ (dot, $\sigma = 0.861$ km s^{-1}), $C^{18}O$ (dot-dash, $\sigma = 0.808$ km s^{-1}), ^{13}CO (dash, $\sigma = 1.05$ km s^{-1}).

dispersion is relatively low at 1.0 t_{ff}, rises until a peak at 1.2 t_{ff} and then turns over and levels off towards the end of the simulation. The same effect can be seen in the N_2H^+cores except that the effect seems to be delayed until 1.3 t_{ff}. We discuss this trend in section 5.5.

The relative differences in line-centre velocities between the $C^{18}O$ envelope and the N_2H^+ cores are lower throughout the cluster. This is due to $C^{18}O$ emission being more optically thin than the ^{13}CO emission. Consequently, $C^{18}O$ (1-0) is able to trace a region closer to the core. It must be stressed that in these constant abundance calculations CO is not frozen-out and therefore will trace a higher density region – the opposite behaviour of what is known to occur. Consequently, until this is addressed any conclusions drawn from this work must be used cautiously. This effect is explored further in Section 5.5.

5.4.1 The effect of LTE and chemistry

The statistical equilibrium calculations performed in this work are very computationally-intensive and it is important to justify the complexity of the calculations. We investigated a synthetic observation from the last timestep to examine the effect that the assumption of LTE had on the N_2H^+ tracer. Owing to the high overall average density of the cluster (above the critical density for the (1-0) transition in both CO isotopologues), ^{13}CO (1-0) and C^{18}O (1-0) maps showed differences of only a few percent in integrated emission. However, the N_2H^+ map (Figure 5.10) shows a large deviation from the assumption of LTE, happily providing *a posteriori* justification for solving statistical equilibrium throughout the cluster.

We repeated the analysis on the same timestep with the 'drop' model outlined in Section 5.3.2. Although minor differences in line profile intensity and shape are observed, the velocity dispersions did not change significantly and no systematic differences with the constant abundance models were detected. It appears that the optical depths of the transitions we have studied are such that the effective photo-surface of the cores is outside the volume where freeze-out occurs. Naturally a more sophisticated treatment of chemistry, involving a full time-dependent calculation, is required to confirm our preliminary results. However such a calculation is beyond the scope of this thesis.

5.5 Discussion

We performed a detailed radiative transfer analysis of a self-gravitating hydrodynamics simulation in order to examine the relationship between the core and envelope velocity dispersions and make a more direct comparison between the hydrodynamical models and millimetre observations. We used information about specific molecular tracers and determined the non-LTE level populations and used these to calculate the emergent intensity from different observation angles around the cluster. We compared the core-envelope velocity dispersion for different observation directions of the same cluster and found good qualitative agreement with observational results (e.g. Walsh et al. 2004, 2007; Kirk et al. 2007), our main conclusion being that one cannot reject competitive accretion as a viable theory of star formation based on observed velocity profiles.

Ayliffe et al. (2007) found that the majority of their sources had velocity differences (between high and low density gas) less than the sound speed at all timesteps but

that they had a significant tail out to large velocity differences (in this case > 0.5 km s^{-1}). However, they also found that their high-density gas linewidth became larger than their low-density gas linewidth at times greater than 1.1 t_{ff}, contradicting the observations of Walsh et al. and Kirk, Johnstone, & Tafalla (2007). We believe this may be because the study done by Ayliffe et al. (2007) did not take into account optical depth effects and that their density cut for the high density material was too high. By reducing their assumed critical density by a factor of 3 they were able to reduce the velocity dispersion of high-density material by a significant amount so that the high-density gas linewidth was smaller than the low-density linewidth, when averaged over all timesteps. This work does not show the trend of larger high-density linewidths at any time, although Figure 5.9 shows that the velocity dispersion turns over with increasing time. We believe that this effect can be explained by two opposing processes. The original simulation did not drive turbulence so it is expected that in the absence of dynamic interactions between protostellar objects the velocity dispersion of the gas will ultimately tend to decrease. However, as we are observing regions undergoing complex interactions that act to stir up the gas, thereby increasing the velocity dispersion, we see the two effects combined; each star formation episode acting to increase the velocity dispersion and decay over time acting to reduce it. Like Ayliffe et al. and the observational surveys, we see no clear trend in the relative motions of core/envelope pairs and note that the standard deviations of the velocity differences are always far smaller than the velocity dispersion.

The original hydrodynamic simulation did not include any form of driven turbulence. For the radiative transfer calculation, we assumed a global constant non-thermal broadening parameter of 0.3 km s^{-1} in addition to the expected thermal broadening. This is similar to that of the mean non-thermal turbulent line width of the driven case analysed in Offner et al. (2008a) but at odds with 'the notion that cores are supported primarily by thermal pressure, large non-thermal motions having disappeared on small scales (< 0.1 pc) and at high densities ($> 10^4$ cm^{-3})' (Johnstone et al. 2010). As a result we found larger velocity dispersions than have been observed in some studies (Offner et al. 2008b). However, not only are our observed core motions smaller than the mean velocity dispersion but also they are often less than the local sound speed (Figure 5.8). This is true for both starless and protostellar cores and in fact, our results mirror those of Kirk et al. in that they too show no indication that the core-to-envelope motions significantly change between the starless and protostellar stages of evolution. By performing a more detailed radiative transfer study of Bate et al.'s cluster simulation, our findings go some way towards reconciling the results previously obtained by Ayliffe et al. (2007) with those found

by others who had performed similar studies.

Although relatively sophisticated we intend to incorporate additional physics in future. Specifically we have not yet taken into account N_2H^+ hyperfine splitting. Tafalla et al. (2004) studied the effect of neglecting hyperfine splitting (which maximises radiative trapping). They found that the difference in obtained level population was on the order of tens of per cent which is comparable to the uncertainties in collision parameters. Neglecting hyperfine splitting can have a significant effect on the optical depth of any N_2H^+ transition and consequently the emergent intensity. In effect, we have systematically overestimated the line optical depth and the inclusion of hyperfine splitting would shift the line formation regions to higher density. In the future we wish to incorporate hyperfine splitting, although previous work (Daniel et al. 2006) incorporating this microphysics has demonstrated that whilst it is likely to have a significant effect it will not be large enough to affect our overall conclusions.

Another important factor is depletion of CO by adsorption onto dust grain surfaces. For the results presented in this chapter, we have assumed that all tracer molecules occur with constant abundance and many observational studies have shown that this is demonstrably not the case. Chemical networks (Bergin & Langer 1997; Glover & Mac Low 2007) exist that allow us to predict the abundance of many molecular species in molecular clouds. We used a simple chemical model to estimate the effect that CO chemistry would have on our observations, and have discovered that a simple freeze-out model does not significantly affect our results. Nonetheless we recognise that a more complete treatment is desirable and a natural extension of this work would be to couple TORUS with a chemical network to understand more fully where line emission comes from in molecular clouds.

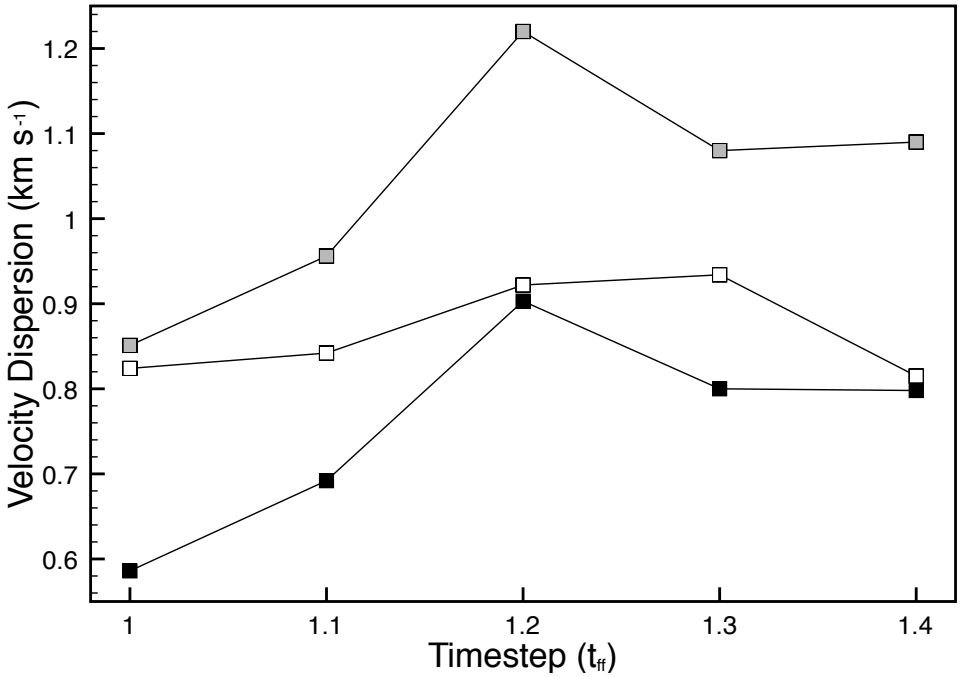


Figure 5.9: The velocity dispersion (km s^{-1}) of each molecular species for each timestep. White squares denote N_2H^+, grey squares denote ^{13}CO and black squares denote $C^{18}O$.

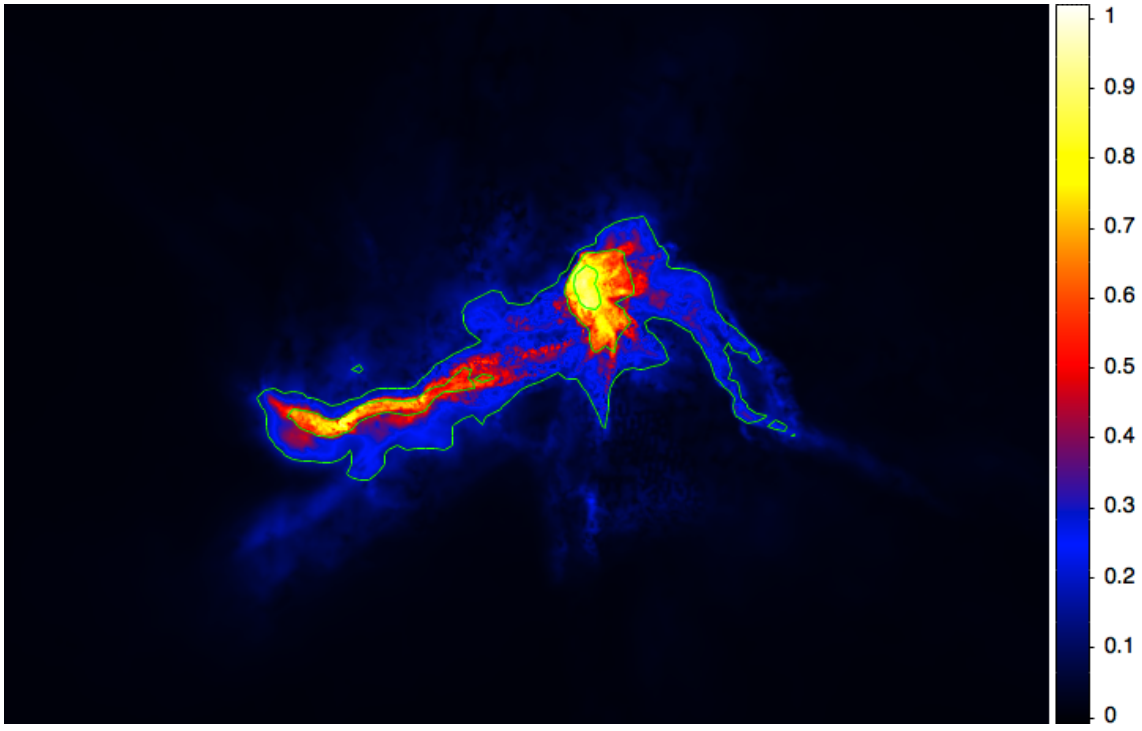


Figure 5.10: The ratio (nLTE:LTE) of integrated intensity of N_2H^+(1-0). The contours are at 0.25, 0.5, and 0.75.

Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

—Douglas Hofstadter,
Gödel, Escher, Bach: An Eternal Golden Braid

Chapter 6

Conclusions and future work

In this section I will summarise the work that has been presented in this thesis. I conclude upon the work done in each chapter of this work individually (Sections 6.1 – 6.4). In each section I also briefly discuss the future work that can be done that is relevant to the section.

6.1 The molecular line transfer module for torus

This chapter documents the non-LTE molecular line radiative transfer code module that has been developed for TORUS. The code is fast, highly parallelised and applicable to realistic astrophysical conditions. The AMR grid enables TORUS to fully resolve geometries that span many spatial orders of magnitude. It is possible to model any molecular line transitions for which the salient data are known, although the code has been extensively validated for microwave rotational transitions in linear molecules that do not exhibit hyperfine splitting. Excitation by dust, which is important for transitions at shorter wavelengths is also treated fully. It is my intention that TORUS become a popular tool for the analysis of observational line data from Herschel and ALMA as well as a tool for predicting the observed line emission expected from simulations (see Chapter 5) and to that end, the code has been written for future maintainability and ease of use. Moreover, now that desktop PCs are

sufficiently powerful, it is now possible to conduct full non-LTE analyses using one- and two-dimensional models in close to real-time. Consequently, full non-local RT codes may be used in χ^2-minimisation routines to extract physical parameters from observed objects without having to resort to LTE or LVG methods.

I hope to add more functionality in the near future to extend its applicability to a wider range of scenarios and to further increase the speed of calculation to offset the enhanced complexity of scenarios that TORUS might deal with.

Although TORUS has been tuned to efficiently calculate line data for microwave rotational transitions in linear molecules and runs most quickly in this scenario, due to inherent generality of the code design it will be very easy to extend the code to deal with hyperfine splitting and overlapping levels in general which add an additional degree of complexity.

6.1.1 Future work

In terms of speed improvements, much work has already been done. Where possible, the code has been vectorised and optimised for single processor performance. As a Monte Carlo method, it scales exquisitely with $N_{\rm proc}$ as the communication overhead is low (see Figure 3.10). However, while the code is currently parallelised with MPI, it does not take advantage of OpenMP which can be used in conjunction with MPI to take advantage of shared memory on machines that have large amounts of RAM. Specifically, in instances where the TORUS grid has a large memory footprint, i.e. in three-dimensional geometries ($> 10^6$ cells) with very many molecular levels to solve SE for, each MPI thread must retain a copy of the grid and all temporary arrays (e.g. `i(1:maxray,1:maxtrans)` can be in excess of 100 MB); this can be up to 8 copies on *zen*. In the so-called 'Hybrid mode' inter-thread communication would be reduced and, crucially, each node would only need one grid and one copy of the working arrays, potentially reducing the memory overhead by tens of gigabytes allowing far larger arrays to be considered (assuming CPU time is available).

Algorithmically, there are several methods that can be implemented to accelerate the code. TORUS uses so-called long characteristics to sample the external radiation field where the closer a cell is to the source (of the rays), the more rays pass through virtually the same part of the cell, resulting in a large number of redundant calculations. Whilst this method is intrinsically the most accurate for finding the intensity along a ray and very intuitive it is not fast. The method of short characteristics, where the salient parameters are interpolated from earlier (upstream) cells

does not suffer from this redundancy but introduces some degree of numerical diffusion. Furthermore, in order to perform the integration along short characteristics, all the relevant values for upstream cells must have already been calculated, imposing the condition that the grid must now be swept outwards from the source. Due to this dependence, the method of short characteristics is intrinsically serial making it impractical for implementation on a parallel computing platform. Conversely, long characteristics methods are inherently parallel and any slowdown can be ameliorated somewhat by parallelising the ray-tracing process because each thread is able to independently work on a different ray. Rijkhorst et al. (2006) suggest a third way, that combines the desirable properties of both methods, excellent parallelisation and computational efficiency, called *hybrid characteristics* where 'ray sections are traced through the cells of each patch and it is these local contributions which are combined through interpolation by performing another ray trace, this time not over cells but over patches'. It is anticipated that implementing this method would greatly accelerate the code when the number of rays being used to sample the radiation field becomes very large. Currently this stage can take many hours even when executed on many nodes.

Through extensive code profiling, it has been determined that TORUS spends a significant proportion of time calculating ray-boundary intersections and traversing the oct-tree. Currently, the routines that perform these calculations are very generalised and do not take into account the discrete nature of the AMR grid. The GADGET-2 code (Springel 2005) utilises Peano-Hilbert curves in order to decompose the simulation space. It may be possible to implement some kind of similar structure to improve the average-case ray-tracing speed through the grid but it is not clear yet how this might be implemented in FORTRAN.

I have recently noted that if the memory were available, `makeImageGrid` would not have to loop over all velocity channels as it does currently. As in `getray`, an array of intensities in each channel could be returned along the same ray, reducing the amount of time spent traversing the grid. This would halve the execution time when dealing with many velocity channels.

Finally, it is often possible to exploit the geometry of a model in order to gain significant execution time reductions. When considering disc structures Pavlyuchenkov et al. (2007) introduced the concept of interaction areas for each cell, where only radiation from within the area is likely to affect the level populations for that cell. For a disc structure exhibiting Keplerian rotation it is possible to analytically determine these zones in advance but it may be possible to generate them numerically for more complex geometries. The other innovation introduced in the same work was that

of thermalised cells where, if it is clear a cell is likely to be in thermal equilibrium with its surroundings there is no need to determine the level populations within as it will naturally be determined by the Boltzmann equation.

The simple, fundamental objective in this project is to improve techniques used to estimate the masses of gas and dust in several typical regions, using a wide range of data from the optical, infrared, sub-millimetre/millimetre spectral regions, which can critically constrain radiative transfer modelling.

6.2 Benchmarking

This chapter showcases the exemplary performance of TORUS in some challenging astrophysical scenarios. The models used in this chapter contain much of the physics necessary to model a collapsing cloud and a circumstellar disc with realistic temperatures and densities. TORUS compares well with other RT codes that have undertaken the same benchmark tests and in the case of the H_2O benchmark, quantitatively improves on the other codes efforts (in that in recovers more of the flux expected from the analytic solution).

With a preponderance of processing power now available to astronomers, it would seem that the days of local approximations are numbered. I have shown that some complex models take very little time even today. Even a more complex two-dimensional disc model with tens of thousands of cells takes little more than a day to converge to a non-LTE solution and, moreover, the ray-tracing routines are able to produce high-resolution datacubes in just a few hours.

As the push towards multi-core processors takes sway, it will become possible to use codes like TORUS to interpret observational data in something very close to real time on a regular desktop machine. It is vital then that these codes can be relied upon to produce accurate results.

6.2.1 Future work

The benchmark tests that were discussed in Chapter 3 are quite old now (8 years for the collapsing cloud and 6 years for the water benchmark). Whilst they are still very relevant, we must now look ahead to more robust, demanding benchmarks that will test codes' abilities to resolve the high-resolution we expect to be able to see with ALMA and other observatories. Moreover, it is now very clear that accurate

chemical modelling is essential for providing high-quality modelling of dense cores and discs. The benchmarks discussed in this chapter all assume constant abundances of their molecular tracers. We should now look to work like Tsamis et al. (2008) and others to create standard scenarios that all codes should be able to reproduce.

In this vein, I am working on a gas and dust disc model that was designed by Glauser et al. (2008) to model IRAS 04158+2805, a young source with spectral type M5-M6 in the Taurus star-forming region. The cylindrically symmetric model is specified by 60 radial points and 40 vertical points. The gas and dust are at different temperatures and because the molecular abundance (of ^{13}CO) has been calculated using a chemical network exhibits complex structure. I hope to present the results of this in the near future.

6.3 An efficient particle-to-grid method

The particle-to-grid method presented in this chapter has proved to be a useful tool for rapidly transposing a set of irregularly ordered SPH particles on to a fixed AMR grid. A number of parameters can be used to control the resolution of the grid and to control the fidelity with which the temperature or density or velocity field is reproduced.

In clustered star-forming environments, gas that is dense enough to emit with sufficient intensity may only exist in the small volumes occupied by dense cores compared with the size of the whole cloud. Consequently, fine discretisation is only required over a small volume. Thus, I devised an efficient method of mapping irregular SPH data onto an AMR grid with minimal discretisation error. This technique permitted the study of the expected molecular line emission from a hydrodynamical simulation of molecular cloud collapse using SPH with our grid-based radiative transfer code.

As well as enabling us to accurately model line profiles, the mapping of density distributions from the particle-based representation onto an AMR grid has important implications for coupling radiation transfer and hydrodynamics. For example, using the technique outlined in Chapter 4, it is now possible to conduct a full, polychromatic multiple scattering treatment of the transport of radiation in an SPH simulated circumstellar disc using high-accuracy grid-based techniques (Acreman et al. 2010b) without having to resort to the flux-limited diffusion approximation employed in earlier radiative transfer simulations (e.g. Whitehouse & Bate 2004; Krumholz et al. 2007; Bate 2009b).

The code has also been used to rapidly discretise AU-scale circumstellar discs through to kiloparsec-scale galactic hydrodynamic simulations so that a full treatment of radiative feedback can be performed. Currently, the diffusion approximation is still used in areas of high optical depth but the surface is determined self-consistently, as opposed to the grey flux-limited diffusion method commonly used in non-polychromatic SPH codes. In a disc configuration the volume covered by the diffusion approximation is large, reducing the computational effort. In contrast, the volume that can be approximated in a tenuous molecular cloud is far smaller than the cloud itself; thus extending such calculations to a cluster collapse model is currently not tractable given the very large dynamic range of linear scales necessary to accurately treat the radiation transport of optically thin-to-thick boundaries, but, with further advances in computational power this kind of calculation will become a reality.

6.3.1 Future work

In the future, I hope to modify the code so that the grid points are filled in an order such that the particle list that is currently created for each grid parent cell is created less frequently, relying more heavily on the property of compact support of the kernel to cull the contributions of any particle in the list that is too far away from the particle.

Soon, I will implement a monotonic cubic interpolation routine following Fritsch & Carlson (1980) that will guarantee the sign of the density while hopefully providing the benefits of improved interpolation order. By utilising the density and velocity gradient information on the particles in a manner identical to that already done for the density and velocity information it is possible to perform this more accurate interpolation without using data from outside the grid structure. This will lead to even smoother images and potentially more accurate physics where the density structure is complex and the use of additional cells is not practical. Conversely, it may be possible to use fewer cells to model the radiation feedback which may be important in calculations that are barely tractable.

Using the rate equations prescribed in Dobbs et al. (2008) it is possible to estimate the dynamic relative fractional abundance of CO to H_2 across the galaxy for each timestep of the SPH model. Douglas et al. intend to complement their existing HI study with a synthetic CO galactic plane survey. Whereas the simplicity of the two-level 21-cm transition makes the production of the synthetic HI spectra straightforward, detailed balance calculations will be necessary for other molecular tracers including CO. As TORUS been used to perform a similar study in a star-forming cluster, it is a natural extension to the TORUS to perform this calculation.

6.4 Creating synthetic images of SPH cluster calculations

This chapter demonstrates the full capabilities of the molecular line radiative transfer module and the particle-to-grid algorithm used in TORUS.

Having obtained non-LTE level populations throughout the cluster for tracers of high– and low–density gas over 5 timesteps covering the creation of 50 protostellar objects we were able to produce and analyse synthetic observations. We have shown in this chapter that clusters exhibiting competitive accretion are able to reproduce the properties of relative core motions found in observation.

The conclusions of this work are discussed in Section 5.5 so here I will discuss future work that can be done on hydrodynamic simulations.

6.4.1 Future work

In 2003, Bate et al. demonstrated that competitive accretion was a likely formation mechanism for stars in the kinds of environments that can be observed. In order to render these hydrodynamic calculations tractable, their original simulations did not consider radiative transfer and were limited to quite a small cluster. This is the work we have tested our code on today. Since 2003, Bate has run a 500 $M_{\odot}$ simulation with 10 times the number of SPH particles (Bate 2009a). This simulation created over 25 times as many protostellar objects as the original simulation and very many more star-forming cores. The calculation has also been run with a realistic gas equation of state and radiative transfer in the flux-limited diffusion approximation rather than the original barotropic equation of state (Whitehouse & Bate 2004). This simulation Bate (2009b) produced a ratio of brown dwarfs to protostars that better matches the ratio seen in observations.

As well as providing an excellent proof of concept for the particle-to-grid algorithm, the work contained in Rundle et al. (2010) provided confirmation of many of the conclusions reached by Ayliffe et al. (2007).

The next logical step for any future synthetic line mapping would be to incorporate the enhanced chemical modelling and hyperfine splitting of N_2H^+ and to more closely compare how the SPH particle distribution compares with the RT analysis. Moreover, having created accurate synthetic maps, we should examine other physical parameters of the simulation, such as the identified core masses and density

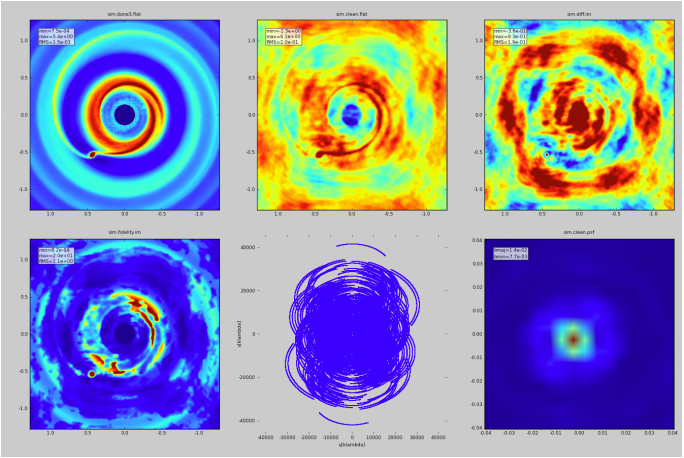


Figure 6.1: Output of the ALMA simulation package, CASA.

profiles. The effects of convolution should also be examined in more detail and it may be possible to investigate what might be seen with future observatories like ALMA.

6.5 Current work

In this section I will briefly discuss the work that is currently being done using TORUS. The application of the code given below is in the early stages but I will discuss the preliminary work here.

6.5.1 Predicted ALMA observations of a proto-planetary disc

Ben Ayliffe and I are currently working on a project to investigate whether a planet formed in an SPH simulation (Ayliffe & Bate 2009a,b), very close to the inner edge of a proto-planetary disc, can be resolved using ALMA. We will carry out multi-wavelength continuum and molecular line calculations to image the disc. An 850 micron image from TORUS has been read into CASA, the ALMA simulation software package, which produces output as shown in Figure 6.1. The figure shows the input data (top-left) and predicted output (top-middle) as well as a number of other diagnostic images. Although other studies have investigated the detectability of planets forming in discs (e.g. Wolf et al. 2002; de Gregorio-Monsalvo et al. 2004), this will be the first investigation based on radiation hydrodynamical simulations.

Appendix A

Good coding practice

Below, I have listed some of the coding practices I have tried to adhere to whilst writing the code.

1. The bulk of the work done by the program should be done in subroutines contained within modules that group similar subroutines together. The main program should primarily consist of calls to these subroutines. This improves readability and code maintenance.

2. Where appropriate, code is commented. Typically, functions and subroutines start with a description of their utility and there is a running commentary through the main program explaining what's going on.

3. All variables, functions and subroutines have descriptive names; loop variables are similarly descriptive, e.g. `itrans` vs. `i`. We use 'CamelCase' to improve code readability.

4. Code is fully indented, demarcating code branches and loops.

5. Calculations that consist of multiple steps are split over as many program lines.

6. We define all mathematical and physical constants in a module, `constants_mod`. This eliminates the need for repetitive calculation of constant values and improves readability. Furthermore, if the accepted value of Planck's constant, h, or π should change then it is only necessary to change it in one location.

7. Commonly used variables are declared as public rather than being passed in function calls to clarify the pertinent input and output of a call.

Appendix B

getray code fragment

```
1  ! Follow long characteristic of ray until edge of grid
2       do while(inOctal(grid%octreeRoot, currentPosition))
3
4  ! Find distance to edge of cell
5          call findSubcellLocal(currentPosition, thisOctal, subcell)
6          call distanceToCellBoundary(grid, currentPosition, direction, tVal, soctal
               = thisoctal)
7
8          CellEdgeInterval = OneOverNtauArray(ntau) ! if ntau = 2 then the segment is
                taken in one chunk
9          dds = tval * cellEdgeInterval ! determine line segment length
10         halfstep = dds * 0.5 * direction ! find point halfway between start and end
                to take representative velocity
11         dist = 0.d0
12 ! Calculate number density of molecules
13         nMol = thisOctal%molAbundance(subcell) * thisOctal%nh2(subcell)
14 ! Absorption by gas per length
15         balance(1:maxtrans) = (hcgsOverFourPi * nmol) * &
16              (thisOctal%molecularLevel(ilower(1:maxtrans),subcell) * thisMolecule%
                    einsteinBlu(1:maxtrans) - &
17              thisOctal%molecularLevel(iupper(1:maxtrans),subcell) * thisMolecule%
                    einsteinBul(1:maxtrans))
18 ! Emission by gas per length
19         spontaneous(1:maxtrans) = (hCgsOverfourPi * nmol) * &
20              thisMolecule%einsteinA(1:maxtrans) * thisOctal%molecularLevel(iupper
                    (:),subcell)
21 ! Source function
22         where (balance /= 0.d0)
23            snu(1:maxtrans) = spontaneous(1:maxtrans) / balance(1:maxtrans)
24         end where
25 ! Calculate total emission and absorption over entire cell (sum over all line
     segments)
26         do itau = 2, nTau
27 ! Get next position
28            dist = dist + dds
29            thisPosition = currentPosition + dist * direction
30 ! Get velocity at midpoint
31            thisVel = velocity(thisPosition-halfstep, grid)
32 ! Get velocity difference and weighting
```

```
33            dv = deltaV - (thisVel .dot. direction)
34            PhiProfVal = phiProf(dv, thisOctal%molmicroturb(subcell))
35 ! Get weighted gas absorption (+dust if req). If usedust then update jnu and snu
36               alpha(1:maxtrans) = balance(1:maxtrans) * phiprofVal
37            endif
38
39 ! Calculate optical depth and associated attenuation factor
40            dTau = alpha * dds * 1.d10
41            attenuation = exp(-tau)
42 ! Intensity along ray owing to line segment
43            localradiationfield = exp(-dtau)
44            localradiationfield = OneArray - localradiationfield
45            di0 = localradiationfield * snu
46 ! Add contribution to whole line
47            i0 = i0 + attenuation * di0
48            tau = tau + dtau
49         enddo
50 ! Update position into next cell making sure that new position is definitely in new
       cell
51         currentPosition = currentPosition + (tval + 1.d-3*grid%halfSmallestSubcell)
               * direction
52      enddo
53
54 ! Add attenuated CMB to intensity at point.
55      i0(1:maxtrans) = i0(1:maxtrans) + BnuBckGrnd(1:maxtrans) * attenuation(1:
            maxtrans)
```

Listing B.1: Source code fragment from getray.

Appendix C

Recursion code

```
1       do subcell = 1, thisOctal%maxChildren
2          if (thisOctal%hasChild(subcell)) then
3             ! find the child
4             do i = 1, thisOctal%nChildren
5                if (thisOctal%indexChild(i) == subcell) then
6                   child => thisOctal%child(i)
7                   call calculateOctalParams(grid, child, thisMolecule)
8                   exit
9                end if
10            end do
11         else
12 ...
```

Listing C.1: Source code fragment that controls the recursion through the octree

Appendix D

Table of TORUS input parameters

Table 7: Required parameters are in **bold**

Parameter	Type	Comment	Range	(default)
moleculefile	char	Name of the molecule to image the grid in (need the .mol). Can be any from the Leiden LAMDA database		
densitysubsample	logical	Interpolate density where possible in order to make smoother image	T/F	F
molAbundance	real	Relative abundance of molecule : H_2 Overridden by doChemistry	$0-1$	
vturb	real	Non-thermal turbulent velocity (km s^{-1}) Currently constant throughout grid See equation ... for usage	$0 \sim 1$	0.2
isinlte	logical	I nitialise molecular level populations assuming LTE Useful for geometries that fulfil LTE criteria (almost) globally	T/F	T
dongstep	logical	Perform Ng acceleration steps at specified intervals current global step is 6 current local step is 5	T/F	T
quasi	logical	Use Sobol quasi-random number generator to determine ray direction and frequency	T/F	F
setmaxlevel	integer	Manually set maximum number of levels to consider during convergence This can reduce the computation time for the grid at the obvious cost of loss of information about upper levels and inaccuracy in levels around them. Set to 0 or not set specifies that the code should automatically determine the maximum number of levels to take into account	2 - n_{levels}	0
plotlevels	logical	Write VTK file of diagnostic data + level populations in ./plots	T/F	F
usedust	logical	Includes continuum emission/absorption from dust (requires mie at the moment)	T/F	F
restart	logical	Restart process from part-converged grid Useful if the job is killed. Old level populations are not stored so convergence information cannot be used for first subsequent iteration	T/F	F
addnewmoldata	logical	Add molecular data to a secular grid Currently used for molcluster geometry where a temperature/density is stored on grid but nothing else	T/F	F
tolerance	real	RMS Convergence level (see eqn ...) for writemol Standard error per pixel (eqn...) for readmol with adaptive subsampling (see subpixels	$(0-0.1]$	0.01
lowmemory	logical	[broken?] Do not store constants for grid but recalculate each time (slower)	T/F	F
distance	real	Distance to grid (in pc)	[1 - 10^{10}]	10
maxVel	real	Range of velocities over which to scan (km s^{-1}). Currently this is symmetric about 0 and assumes the object does not possess any systematic velocity	$(0-1000]$	
lineimage	logical	Create a datacube in a molecular line determined by itrans rather than a continuum wavelength	T/F	T
lte	logical	Create datacube using LTE grid rather than nLTE grid	T/F	F
itrans	integer	Determines line transition for raytracing See molecularfile file for details	[1-n_{trans}]	
lamline	real	Determines wavelength of continuum image (in microns)		850
imageside	real	Determines spatial extent of datacube		

npixels	integer	Controls number of spatial pixels per dimension If using multiple threads, make this a multiple of $n_t hreads$	[1-4096]	100
nSubpixels	integer	Controls number of rays used to sample object per pixel Setting this to 0 turns on adaptive pixel sub-sampling. The region is sampled with paraxial rays until the standard error drops below tolerance	[0-4096]	1
nv	integer	Controls number of velocity channels in datacube	1-10000	100
wanttau	logical	output FITS file also contains average ray optical depth per velocity channel	T/F	F
centrevecX [x/y/z]	real	Image centre coordinates in 10^{10} cm.		0
rotateviewaboutX [x/y/z]	real	Angle (degrees) through which image plane is rotated about X-/Y-/Z- axis	[-180–180]	0
observerpos	integer	If present, read line XX from observerpos.dat to get observer coordinates		
maxrhocalc	logical	FITS file contains maximumdensity encountered in pixel instead of intensity	T/F]	F
SPH & splitting Parameters				
limitscalar	real	Mass (g) limit above which cell is split	[1]	1
limittwo	real	Ratio of minimum and maximum densities Negative value: $\rho_{\max}/\rho_{\min} \leq X$ Positive value: $(\rho_{\max} - \rho_{\min})/(\rho_{\min} + \rho_{\max}) \leq X$	[2]	2
vturbmultiplier	real	If present, number of turbulent linewidths that velocity in cell exceeds before splitting (km s^{-1})		-999
mindepthamr	integer	Minimum AMR depth	[2-30]	2
sphdatafilename	char	SPH data filename		
hcritpercentile	real	Fraction of particle smoothing lengths that $h_{\rm crit}$ should exceed	(0-1]	0.9
hmaxpercentile	real	Fraction of particle smoothing lengths that $h_{\rm max}$ should exceed	(0-1]	0.99
kerneltype	integer	Kernel used for discretisation (1 is cubic spline, 0 is truncated exponential)	0/1	1

Bibliography

Acreman, D. M., Douglas, K. A., Dobbs, C. L., & Brunt, C. M. 2010a, MNRAS, 856

Acreman, D. M., Harries, T. J., & Rundle, D. A. 2010b, MNRAS, 403, 1143

Alves, J. F., Lada, C. J., & Lada, E. A. 2001, Nature, 409, 159

Amdahl, G. M. 1967, in AFIPS '67 (Spring): Proceedings of the April 18-20, 1967, spring joint computer conference (New York, NY, USA: ACM), 483–485

Andre, P., Ward-Thompson, D., & Barsony, M. 1993, ApJ, 406, 122

Andre, P., Ward-Thompson, D., & Motte, F. 1996, A&A, 314, 625

Ayliffe, B. A. & Bate, M. R. 2009a, MNRAS, 397, 657

Ayliffe, B. A. & Bate, M. R. 2009b, MNRAS, 393, 49

Ayliffe, B. A., Langdon, J. C., Cohl, H. S., & Bate, M. R. 2007, MNRAS, 374, 1198

Bacmann, A., Lefloch, B., Ceccarelli, C., et al. 2002, A&A, 389, L6

Ballesteros-Paredes, J., Klessen, R. S., Mac Low, M., & Vazquez-Semadeni, E. 2007, Protostars and Planets V, 63

Barnard, E. E. 1919, ApJ, 49, 1

Bate, M. R. 2009a, MNRAS, 392, 590

Bate, M. R. 2009b, MNRAS, 392, 1363

Bate, M. R. & Bonnell, I. A. 1997, MNRAS, 285, 33

Bate, M. R., Bonnell, I. A., & Bromm, V. 2002a, MNRAS, 332, L65

Bate, M. R., Bonnell, I. A., & Bromm, V. 2002b, MNRAS, 336, 705

Bate, M. R., Bonnell, I. A., & Bromm, V. 2003, MNRAS, 339, 577

Bate, M. R., Bonnell, I. A., & Price, N. M. 1995, MNRAS, 277, 362

Berger, M. J. & Colella, P. 1989, Journal of Computational Physics, 82, 64

Bergin, E. A., Alves, J., Huard, T., & Lada, C. J. 2002, ApJL, 570, L101

Bergin, E. A., Kaufman, M. J., Melnick, G. J., Snell, R. L., & Howe, J. E. 2003, ApJ, 582, 830

Bergin, E. A. & Langer, W. D. 1997, ApJ, 486, 316

Bernes, C. 1979, A&A, 73, 67

Blitz, L. & Williams, J. P. 1999, in NATO ASIC Proc. 540: The Origin of Stars and Planetary Systems, ed. C. J. Lada & N. D. Kylafis, 3–+

Bohr, N. 1913, Nature, 92, 231

Bok, B. J., Lawrence, R. S., & Menon, T. K. 1955, PASP, 67, 108

Bonnell, I. A. & Bate, M. R. 2006, MNRAS, 370, 488

Bonnell, I. A., Bate, M. R., Clarke, C. J., & Pringle, J. E. 1997, MNRAS, 285, 201

Bonnell, I. A., Bate, M. R., Clarke, C. J., & Pringle, J. E. 2001, MNRAS, 323, 785

Bonnell, I. A., Bate, M. R., & Vine, S. G. 2003, MNRAS, 343, 413

Bonnell, I. A., Bate, M. R., & Zinnecker, H. 1998, MNRAS, 298, 93

Bourke, T. L., Myers, P. C., Robinson, G., & Hyland, A. R. 2001, ApJ, 554, 916

Caselli, P., Walmsley, C. M., Tafalla, M., Dore, L., & Myers, P. C. 1999, ApJL, 523, L165

Chabrier, G. 2003, PASP, 115, 763

Chandrasekhar, S. 1950, Radiative transfer, ed. Chandrasekhar, S.

Collings, M. P., Anderson, M. A., Chen, R., et al. 2004, MNRAS, 354, 1133

Crapsi, A., Caselli, P., Walmsley, C. M., et al. 2005, ApJ, 619, 379

Danby, G., Flower, D. R., Valiron, P., Schilke, P., & Walmsley, C. M. 1988, MNRAS, 235, 229

D'Angelo, G., Henning, T., & Kley, W. 2002, A&A, 385, 647

Daniel, F. & Cernicharo, J. 2008, A&A, 488, 1237

Daniel, F., Cernicharo, J., & Dubernet, M. 2006, ApJ, 648, 461

de Gregorio-Monsalvo, I., Gómez, J. F., & D'Alessio, P. 2004, Ap&SS, 292, 445

di Francesco, J., Evans, II, N. J., Caselli, P., et al. 2007, Protostars and Planets V, 17

Dieter, N. H. 1964, Nature, 201, 279

Dobbs, C. L. & Bonnell, I. A. 2007, MNRAS, 376, 1747

Dobbs, C. L., Glover, S. C. O., Clark, P. C., & Klessen, R. S. 2008, MNRAS, 389, 1097

Douglas, K. A., Acreman, D. M., Dobbs, C. L., & Brunt, C. M. 2010, MNRAS, 896

Draine, B. T. 1978, ApJS, 36, 595

Draine, B. T. & Lee, H. M. 1984, ApJ, 285, 89

Dubrulle, B., Morfill, G., & Sterzik, M. 1995, Icarus, 114, 237

Dullemond, C. P., Henning, T., Visser, R., et al. 2007, A&A, 473, 457

Dullemond, C. P. & Turolla, R. 2000, A&A, 360, 1187

Dutrey, A., Guilloteau, S., & Simon, M. 1994, A&A, 286, 149

Eddington, A. S. 1916, MNRAS, 77, 16

Elmegreen, B. G. 1990, in Astronomical Society of the Pacific Conference Series, Vol. 12, The Evolution of the Interstellar Medium, ed. L. Blitz, 247–271

Elmegreen, B. G. 2000, ApJ, 530, 277

Ercolano, B., Barlow, M. J., & Sugerman, B. E. K. 2007, MNRAS, 375, 753

Ewen, H. I. & Purcell, E. M. 1951, Nature, 168, 356

Freeman, P. E., Kashyap, V., Rosner, R., & Lamb, D. Q. 2002, ApJS, 138, 185

Frerking, M. A., Langer, W. D., & Wilson, R. W. 1982, ApJ, 262, 590

Fritsch, F. N. & Carlson, R. E. 1980, SIAM Journal on Numerical Analysis, 17, 238

Fruscione, A., McDowell, J. C., Allen, G. E., et al. 2006, in Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, Vol. 6270, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series

Gingold, R. A. & Monaghan, J. J. 1977, MNRAS, 181, 375

Glauser, A. M., Ménard, F., Pinte, C., et al. 2008, A&A, 485, 531

Glover, S. C. O. & Mac Low, M. 2007, ApJS, 169, 239

Godard, B., Falgarone, E., Gerin, M., Hily-Blant, P., & De Luca, M. 2010, ArXiv e-prints

Goldsmith, P. F., Bergin, E. A., & Lis, D. C. 1997, ApJ, 491, 615

Goodwin, S. P., Whitworth, A. P., & Ward-Thompson, D. 2004, A&A, 423, 169

Green, S. 1975, ApJ, 201, 366

Green, S. & Chapman, S. 1978, ApJS, 37, 169

Harries, T. J. 2000, MNRAS, 315, 722

Harries, T. J., Babler, B. L., & Fox, G. K. 2000, A&A, 361, 273

Harries, T. J., Monnier, J. D., Symington, N. H., & Kurosawa, R. 2004, MNRAS, 350, 565

Harwit, M. 2004, Advances in Space Research, 34, 568

Hatchell, J., Fuller, G. A., Richer, J. S., Harries, T. J., & Ladd, E. F. 2007, A&A, 468, 1009

Hatchell, J., Richer, J. S., Fuller, G. A., et al. 2005, A&A, 440, 151

Hauschildt, P. H. & Baron, E. 2006, A&A, 451, 273

Hearnshaw, J. B. 2010, in Bulletin of the American Astronomical Society, Vol. 41, Bulletin of the American Astronomical Society, 214–+

Herschel, W. 1785, Royal Society of London Philosophical Transactions Series I, 75, 213

Herzberg, G. 1950, Molecular spectra and molecular structure. Vol.1: Spectra of diatomic molecules, ed. Herzberg, G.

Hogerheijde, M. R. 2005, Ap&SS, 295, 179

Hogerheijde, M. R. & van der Tak, F. F. S. 2000, A&A, 362, 697

Jeans, J. H. 1902, Royal Society of London Philosophical Transactions Series A, 199, 1

Jessop, N. E. & Ward-Thompson, D. 2001, MNRAS, 323, 1025

Johnstone, D. & Bally, J. 1999, ApJL, 510, L49

Johnstone, D., Rosolowsky, E., Tafalla, M., & Kirk, H. 2010, ApJ, 711, 655

Jørgensen, J. K. 2004, A&A, 424, 589

Juvela, M. 1997, A&A, 322, 943

Juvela, M. & Padoan, P. 2005, ApJ, 618, 744

Kirk, H., Johnstone, D., & Tafalla, M. 2007, ApJ, 668, 1042

Kirk, H. M. 2009, PhD thesis, University of Victoria (Canada)

Klessen, R. S., Burkert, A., & Bate, M. R. 1998, ApJL, 501, L205+

Kroupa, P. 2002, Science, 295, 82

Krumholz, M. R., Klein, R. I., & McKee, C. F. 2007, ApJ, 665, 478

Kurosawa, R., Harries, T. J., & Symington, N. H. 2006, MNRAS, 370, 580

Kutner, M. L. & Ulich, B. L. 1981, ApJ, 250, 341

Lacy, J. H., Knacke, R., Geballe, T. R., & Tokunaga, A. T. 1994, ApJL, 428, L69

Lada, C. J. 1987, in IAU Symposium, Vol. 115, Star Forming Regions, ed. M. Peimbert & J. Jugaku, 1–17

Lada, C. J. & Wilking, B. A. 1984, ApJ, 287, 610

Larson, R. B. 1981, MNRAS, 194, 809

Larson, R. B. 2003, Reports on Progress in Physics, 66, 1651

Laux, S. E. 1996, IEEE Trans. on CAD of Integrated Circuits and Systems, 15, 1266

Le Petit, F., Nehmé, C., Le Bourlot, J., & Roueff, E. 2006, ApJS, 164, 506

Lilley, A. E. 1955, ApJ, 121, 559

Lorentz, H. A. 1905, Koninklijke Nederlandse Akademie van Weteschappen Proceedings Series B Physical Sciences, 8, 591

Lucy, L. B. 1977, AJ, 82, 1013

Lucy, L. B. 1999, A&A, 344, 282

Mangum, J. & Shirley, Y. 2008

Masunaga, H. & Inutsuka, S. 2000, ApJ, 531, 350

Mathis, J. S. 1993, Reports on Progress in Physics, 56, 605

Mathis, J. S., Rumpl, W., & Nordsieck, K. H. 1977, ApJ, 217, 425

McKee, C. F. & Ostriker, J. P. 1977, ApJ, 218, 148

McKee, C. F. & Tan, J. C. 2003, ApJ, 585, 850

Melnick, G. J., Ashby, M. L. N., Plume, R., et al. 2000, ApJL, 539, L87

Mihalas, D. 1978, Stellar atmospheres /2nd edition/, ed. Hevelius, J.

Monaghan, J. J. 1992, ARA&A, 30, 543

Monaghan, J. J. & Lattanzio, J. C. 1985, A&A, 149, 135

Monteiro, T. S. 1985, MNRAS, 214, 419

Morokoff, W. & Caflisch, R. E. 1994, SIAM J. Sci. Comput., 15, 1251

Morokoff, W. J. & Caflisch, R. E. 1995, J. Comput. Phys., 122, 218

Motte, F., Andre, P., & Neri, R. 1998, A&A, 336, 150

Muller, C. A. & Oort, J. H. 1951, Nature, 168, 357

Müller, H. S. P., Schlöder, F., Stutzki, J., & Winnewisser, G. 2005, Journal of Molecular Structure, 742, 215

Myers, P. C. & Benson, P. J. 1983, ApJ, 266, 309

Myers, P. C., Linke, R. A., & Benson, P. J. 1983, ApJ, 264, 517

Ng, K. 1974, J. Chem. Phys., 61, 2680

Niederreiter, H. & Sloan, I. H. 1996, Mathematical and Computer Modelling, 23, 69

Nisini, B., Benedettini, M., Codella, C., et al. 2010, ArXiv e-prints

Offner, S. S. R., Klein, R. I., & McKee, C. F. 2008a, ApJ, 686, 1174

Offner, S. S. R. & Krumholz, M. R. 2009, ApJ, 693, 914

Offner, S. S. R., Krumholz, M. R., Klein, R. I., & McKee, C. F. 2008b, AJ, 136, 404

Olson, G. L., Auer, L. H., & Buchler, J. R. 1986, Journal of Quantitative Spectroscopy and Radiative Transfer, 35, 431

Ossenkopf, V. & Henning, T. 1994, A&A, 291, 943

Ossenkopf, V., Trojan, C., & Stutzki, J. 2001, A&A, 378, 608

Osterbrock, D. E. & Ferland, G. J. 2006, Astrophysics of gaseous nebulae and active galactic nuclei, ed. Osterbrock, D. E. & Ferland, G. J.

Pascucci, I., Wolf, S., Steinacker, J., et al. 2004, A&A, 417, 793

Pavlyuchenkov, Y., Semenov, D., Henning, T., et al. 2007, ApJ, 669, 1262

Pavlyuchenkov, Y., Wiebe, D., Shustov, B., et al. 2008, ApJ, 689, 335

Pence, W. D. 1992, in Astronomical Society of the Pacific Conference Series, Vol. 25, Astronomical Data Analysis Software and Systems I, ed. D. M. Worrall, C. Biemesderfer, & J. Barnes, 22–+

Pickett, H. M., Poynter, I. R. L., Cohen, E. A., et al. 1998, J. Quant. Spec. Radiat. Transf., 60, 883

Pinte, C., Ménard, F., Duchêne, G., & Bastien, P. 2006, A&A, 459, 797

Planck, M. 1901, Annalen der Physik, 309, 553

Planck, M. 1909, Annalen der Physik, 336, 758

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical recipes in FORTRAN. The art of scientific computing (Cambridge University Press)

Price, D. 2005, ArXiv Astrophysics e-prints

Price, D. J. 2007, Publications of the Astronomical Society of Australia, 24, 159

Price, D. J. & Bate, M. R. 2007, MNRAS, 377, 77

Price, D. J. & Federrath, C. 2010, MNRAS, 960

Price, D. J. & Monaghan, J. J. 2004, MNRAS, 348, 139

Price, D. J. & Monaghan, J. J. 2007, MNRAS, 374, 1347

Rawlings, J. M. C., Hartquist, T. W., Menten, K. M., & Williams, D. A. 1992, MNRAS, 255, 471

Rawlings, J. M. C. & Yates, J. A. 2001, MNRAS, 326, 1423

Reddish, V. C. 1975, MNRAS, 170, 261

Rijkhorst, E., Plewa, T., Dubey, A., & Mellema, G. 2006, A&A, 452, 907

Rots, A. H. & Shane, W. W. 1975, A&A, 45, 25

Rundle, D., Harries, T. J., Acreman, D. M., & Bate, M. R. 2010, ArXiv e-prints

Rybicki, G. B. & Hummer, D. G. 1991, A&A, 245, 171

Rybicki, G. B. & Lightman, A. P. 1979, Radiative processes in astrophysics, ed. Rybicki, G. B. & Lightman, A. P.

Sault, R. J., Teuben, P. J., & Wright, M. C. H. 1995, in Astronomical Society of the Pacific Conference Series, Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes, 433–+

Scalo, J. 1990, in Astrophysics and Space Science Library, Vol. 162, Physical Processes in Fragmentation and Star Formation, ed. R. Capuzzo-Dolcetta, C. Chiosi, & A. di Fazio, 151–176

Schöier, F. L. 2000, PhD thesis, University of Stockholm

Schöier, F. L., van der Tak, F. F. S., van Dishoeck, E. F., & Black, J. H. 2005, A&A, 432, 369

Semenov, D., Pavlyuchenkov, Y., Henning, T., Wolf, S., & Launhardt, R. 2008, ApJL, 673, L195

Shirley, Y. L., Nordhaus, M. K., Grcevich, J. M., et al. 2005, ApJ, 632, 982

Shu, F. H. 1977, ApJ, 214, 488

Shu, F. H., Adams, F. C., & Lizano, S. 1987, ARA&A, 25, 23

Skrutskie, M. F., Cutri, R. M., Stiening, R., et al. 2006, AJ, 131, 1163

Sobol', I. 1967, USSR Computational Math. and Math. Phys., 7, 86

Sobol', I. 1976, USSR Computational Math. and Math. Phys., 236

Sobolev, V. V. 1960, Moving envelopes of stars, ed. Sobolev, V. V.

Springel, V. 2005, MNRAS, 364, 1105

Springel, V. & Hernquist, L. 2002, MNRAS, 333, 649

Stahler, S. W. & Palla, F. 2005, The Formation of Stars, ed. Stahler, S. W. & Palla, F.

Steinacker, J., Henning, T., Bacmann, A., & Semenov, D. 2003, A&A, 401, 405

Stutzki, J. & Winnewisser, G. 1985, A&A, 144, 13

Swings, P. & Rosenfeld, L. 1937, ApJ, 86, 483

Symington, N. 2006, PhD thesis, University of Exeter, Exeter, Devon, EX4 4QL

Symington, N. H., Harries, T. J., & Kurosawa, R. 2005a, MNRAS, 356, 1489

Symington, N. H., Harries, T. J., Kurosawa, R., & Naylor, T. 2005b, MNRAS, 358, 977

Tafalla, M., Myers, P. C., Caselli, P., & Walmsley, C. M. 2004, A&A, 416, 191

Tafalla, M., Myers, P. C., Caselli, P., Walmsley, C. M., & Comito, C. 2002, ApJ, 569, 815

Tatematsu, K., Umemoto, T., Kandori, R., & Sekimoto, Y. 2004, ApJ, 606, 333

Taylor, A. R., Gibson, S. J., Peracaula, M., et al. 2003, AJ, 125, 3145

Tsamis, Y. G., Rawlings, J. M. C., Yates, J. A., & Viti, S. 2008, MNRAS, 388, 898

van der Tak, F. F. S., Black, J. H., Schöier, F. L., Jansen, D. J., & van Dishoeck, E. F. 2007, A&A, 468, 627

van der Tak, F. F. S., van Dishoeck, E. F., Evans, II, N. J., & Blake, G. A. 2000, ApJ, 537, 283

van Zadelhoff, G., Dullemond, C. P., van der Tak, F. F. S., et al. 2002, A&A, 395, 373

Viti, S., Williams, D. A., & O'Neill, P. T. 2000, A&A, 354, 1062

Walmsley, C. M., Flower, D. R., & Pineau des Forêts, G. 2004, A&A, 418, 1035

Walsh, A. J., Myers, P. C., & Burton, M. G. 2004, ApJ, 614, 194

Walsh, A. J., Myers, P. C., Di Francesco, J., et al. 2007, ApJ, 655, 958

Ward-Thompson, D., André, P., Crutcher, R., et al. 2007a, Protostars and Planets V, 33

Ward-Thompson, D., Di Francesco, J., Hatchell, J., et al. 2007b, PASP, 119, 855

Ward-Thompson, D., Scott, P. F., Hills, R. E., & Andre, P. 1994, MNRAS, 268, 276

Wehrse, R. & Kalkofen, W. 2006, A&A Rev., 13, 3

Whitehouse, S. C. & Bate, M. R. 2004, MNRAS, 353, 1078

Wiesemeyer, H. 1997, PhD thesis, AA(Institut de Radio Astronomie Millimétrique);

Williams, D. A. 2005, Journal of Physics Conference Series, 6, 1

Wilson, R. W., Jefferts, K. B., & Penzias, A. A. 1970, ApJL, 161, L43+

Wolf, S., Gueth, F., Henning, T., & Kley, W. 2002, ApJL, 566, L97

Woodall, J., Agúndez, M., Markwick-Kemper, A. J., & Millar, T. J. 2007, A&A, 466, 1197

Yang, B., Stancil, P. C., Balakrishnan, N., & Forrey, R. C. 2010, ArXiv e-prints

Zinchenko, I., Caselli, P., & Pirogov, L. 2009, MNRAS, 395, 2234

Zinnecker, H. 1984, MNRAS, 210, 43

Index