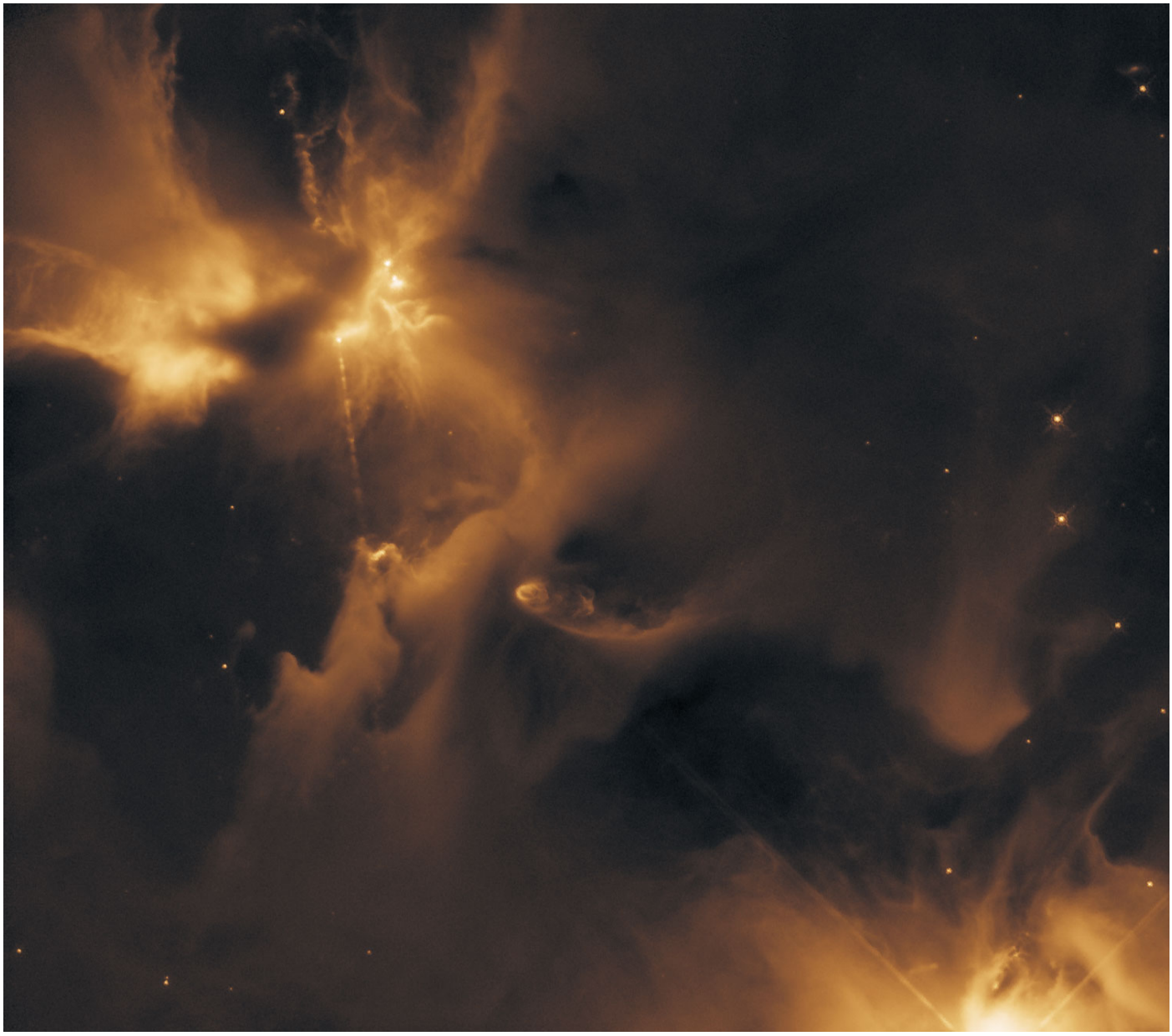
MSc Astrophysics
PXT206: Astrophysics Research Project

Numerical Modelling of Internal Radiative Feedback from Accreting Protostars in Moving-Mesh Code AREPO

Andrew M. Bostock

School of Physics & Astronomy, Cardiff University

4 September 2017

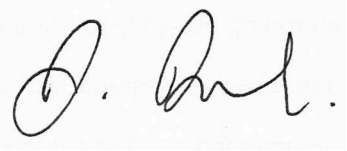


Hubble Space Telescope image of young stellar objects in the Orion B molecular cloud complex (https://apod.nasa.gov)

NOTICE OF SUBMISSION: POSTGRADUATE TAUGHT DEGREES

SECTION A: TO BE COMPLETED BY THE CANDIDATE AND SUBMITTED WITH THE DISSERTATION

Student ID Number:	**1537349**
Title:	**Mr**
Surname:	**Bostock**
First Names:	**Andrew Mark**
School:	**School of Physics and Astronomy**
Title of Degree:	**MSc Astrophysics**
Full Title of Dissertation submitted:	**Numerical Modelling of Internal Radiative Feedback from Accreting Protostars in Moving-Mesh Code AREPO**
Is this a resubmission?	**Yes** ☐ **No** ☑
Dissertation submitted for Examination:	**Electronically**
Supervisor's Name:	**Dr Paul Clark**
Address for receipt of your Result Letter, Degree Certificate and Details of the Graduation Ceremony:	**Colwinston House** **Colwinston** **Vale of Glamorgan** **CF71 7NJ**
Please indicated whether you wish to attend the Degree Ceremony:	**Yes** ☑ **No** ☐
Preferred Contact Telephone (with dialling code/s):	**07740 552024**
Email/s:	**AndrewMBostock@gmail.com**

CANDIDATE'S SIGNATURE: 	**DATE OF SUBMISSION:** **4 September 2017**

SECTION B: TO BE COMPLETED BY THE SCHOOL

Please tick the appropriate box to confirm that the following documents have been completed by the candidate and deposited in the School.

Dissertation Summary Sheet:	**Yes** ☐	**No** ☐
Statements and Declaration:	**Yes** ☐	**No** ☐
A signed statement regarding the availability of the Dissertation:	**Yes** ☐	**No** ☐
Where Candidate is resubmitting, School should attach cheque (or evidence that the fee has been paid):	**Yes** ☐	**No** ☐

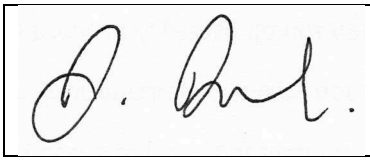
SCHOOL'S SIGNATURE:	**DATE:**
POSITION:	

SECTION C: DECLARATIONS AND STATEMENTS

CANDIDATE'S ID NUMBER	**1537349**
CANDIDATE'S SURNAME	**Bostock**
CANDIDATE'S FULL FORENAMES	**Andrew Mark**

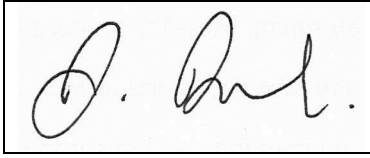
DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

	**4 September 2017**

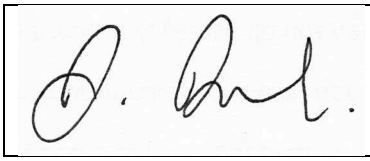
STATEMENT 1

This dissertation is being submitted in partial fulfilment of the requirements for the degree of MSc.

	**4 September 2017**

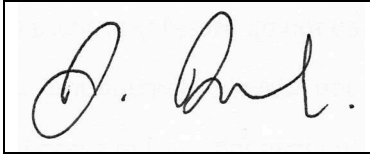
STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

	**4 September 2017**

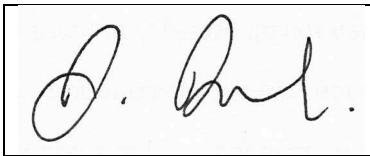
STATEMENT 3

I confirm that the electronic copy is identical to the bound copy of the dissertation.

	**4 September 2017**

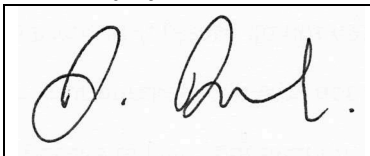
STATEMENT 4

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

	**4 September 2017**

STATEMENT 5

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the Graduate Development Committee**.

	**4 September 2017**

CONTENTS

ABSTRACT

The first phase of the development and implementation of a new 'internal' radiative feedback algorithm is presented for moving-mesh astrophysical hydrodynamic code 'AREPO'. This algorithm allows the effect of radiative feedback on a star-forming molecular cloud to be studied where multiple feedback sources, currently in the form of spherically accreting protostars, are generated in-situ as the cloud evolves. Previous numerical models have primarily focused on the impact of 'external' feedback from sources, typically giant OB stars, outside the cloud. Where numerical models of internal feedback have been reported, none have been identified that involve the use of state-of-the-art moving mesh codes. The veracity of the algorithm output has been demonstrated by comparing simulation predictions of dust temperature for some simple models with 'analytical' calculations of the same parameter.

It is found that in these internal feedback scenarios there is a potentially significant increase in accuracy to be gained by considering the directional nature of the feedback in an 'anisotropic' model on account of the fact that protostars will almost certainly be formed within regions of enhanced column density. Due to the high degree of attenuation to which this leads, significant alteration of radiation energy density from protostellar feedback, as described by the relative energy density parameter G_0, is only expected where the protostar is either in close proximity (within ~0.1 pc) and/or has high mass (> ~5 $M_\odot$) and even the highest mass protostars probably need to be within 0.5 pc to be of any significance in the environment in which protostars evolve. Although currently limited in run time, preliminary science investigations indicate that, on account of generally increased temperature and associated changes in chemistry, this model will, at a general level, confirm the findings from other models that radiative feedback leads to a reduction in star formation efficiency. However further work to improve this algorithm is identified and it is anticipated that this refined product, once developed, will ultimately lead to new insights into how internal radiative feedback controls star formation rate and mass function.

1 INTRODUCTION

1.1 PROJECT MOTIVATION, GOAL & OBJECTIVES

Why and how stars form and subsequently evolve is a fundamental building-block of astrophysics, underpinning many theoretical and observational areas of astrophysical research. Consequently, maximising our understanding of the factors which are important in determining, for example, star formation rates, characteristics and mass distribution are of crucial importance in this field.

In star formation studies, feedback is the group of mechanisms by which a star or other astronomical object affects the formation of new stars from a star-forming region with which it is in close proximity. Such mechanisms include radiative feedback alongside more dynamic mechanisms such as stellar winds and supernovae. These mechanisms appear to be a critical factor in the evolution of the molecular clouds from which stars form, since it is only possible to reproduce the population of stars observed in nature in star-formation models that include them. It is clear, then that any effective model of star formation should incorporate feedback mechanisms as far as practicable.

With the availability of more advanced computer hardware and the development of new numerical techniques, the last ten years or so have seen significant advances in numerical hydrodynamic codes. However, in a comprehensive review of simulation of feedback by such models, Dale (2015) concludes that despite some successes, significant shortcomings persist and there is still much to do before a comprehensive numerical description of stellar feedback exists.

The goal of this project was to make a contribution to addressing the need identified by Dale, specifically by developing and implementing algorithms into the state-of-the-art moving mesh hydrodynamic code 'AREPO' (Springel, 2010; www.illustris-project.org, 2017) that allow the effect of 'internal' radiative feedback from luminous sources that emerge in-situ within a molecular cloud to be taken into account. Previous numerical models have primarily focused on the impact of 'external' feedback from sources, typically giant OB stars, outside the cloud. Furthermore, where numerical models of internal feedback have been reported none have been identified that involve the use of moving mesh codes. As a first step within the time constraints of this project, this study has focused specifically on a simple numerical depiction of spherically accreting protostars as a prototype internal luminous source, this being the first significant luminous objects to emerge in the

evolution of a protostar towards the main sequence. It is anticipated that this model will be considerably enhanced in future phases of development.

The primary research question of the project was, therefore: can a model of internal radiative feedback from embedded accreting protostars be successfully incorporated within the existing coding architecture of AREPO? Confirming this was indeed possible and demonstrating the physical validity of the new code thus formed the key objectives of the project. A secondary objective was to conduct preliminary science investigations into the influence of radiative feedback in a turbulent molecular cloud model using the new code.

In terms of the relevance of this project within the ongoing research in this field, numerical models of internal radiative feedback have not been widely reported and certainly not using AREPO, which prior to this project only contained the capability to model an isotropic radiation field due to a general population of stars in the galaxy. AREPO is used by a number of research groups around the world to investigate astrophysical phenomena at the cosmological, galactic and interstellar scales and is due to become publically available in February 2018, less than 6 months after the end of this project. Consequently, the radiative feedback model initiated by this project will potentially be widely used by the astrophysical community, certainly once the identified further stages of development are complete.

1.2 Background Science Review

In order to successfully model the impact of radiative feedback on star formation processes, it is first necessary to understand the current state of knowledge regarding the role electromagnetic radiation plays in the physics and chemistry of the interstellar medium ('ISM') and what form internal radiative feedback sources might take within a molecular cloud. This is reviewed below.

1.2.1 Interaction between Matter and Radiation in the Interstellar Medium ('ISM')

At a fundamental level, the rate and nature of star formation from the ISM is driven by the interplay between gravitation on the one hand and the often complex heating and cooling mechanisms that control gas temperature and hence hydrostatically balancing pressure on the other (Ward-Thompson and Whitworth, 2011).

At the very cold temperatures associated with molecular clouds (~5 to 15 K), the most important gas heating process is the photoelectric emission that results from UV photons incident on dust grains, thereby transferring the photon energy to the kinetic energy of gas particles via collision with the liberated electrons. This process dominates any heating effect due to viscous dissipation, compressional heating, cosmic ray ionisation or equipartition of energy between dust grains and gas particles by at least an order of magnitude (Glover and Clark, 2012).

Radiation also affects the hydrostatic balance indirectly via photochemical reactions. These reactions ultimately determine the chemical state of interstellar gas (the relative proportions of chemical species present, degree of ionisation and degree of molecularisation) and hence its cooling capacity. In particular, C+ and CO concentrations are critical, being the main coolants once metallicity exceeds $\sim 10^{-3}$ that of solar metallicity i.e. C and O abundances $> \sim 10^{-7}$ relative to the total number of hydrogen nuclei (Dr. Paul Clark, private communication).

Direct interaction between interstellar *gas* and radiation generally results in destructive processes such as photodissociation of molecules and photoionisation of atoms. However the interactions between *dust* and radiation that determine dust temperature are also an important indirect factor in determining the efficiency of interstellar chemical processes. Most significantly, dust grains are considered essential for the formation of molecules in the first place (e.g. Hollenbach and Salpeter, 1971), providing the catalytic surface necessary in this low temperature, low density environment for H_2 and other molecules to form. Conversely, dust can also affect the thermal balance by binding with and thus locking-away important coolants such as C^+ and Si^+, thus suppressing their effect (Peters et al., 2017). The dust grain temperature is a key determinant of the efficiency of the type of chemical processes described above (Bron et al., 2014) and this temperature is almost entirely determined by the equilibrium between the radiation absorbed by a dust grain and the radiation emitted as thermal/IR radiation (Draine, 2011). Since a large proportion of incident UV radiation is absorbed by dust grains and reemitted as IR radiation (Soifer et al., 1987), the interaction between dust and radiation also plays a crucial part in 'shielding' the contents of an existing molecular cloud from photodissociation by UV photons, thus boosting the concentration of molecular species above what would otherwise be the case.

In summary, the interaction between dust and radiation is the most important source of ISM heating and a critical determinant of the chemical state and hence cooling capacity of interstellar gas. This interaction is thus ultimately key to the propensity of a molecular cloud to form stars.

Finally, as a result of radiation pressure, the interaction between interstellar matter and radiation can potentially result in dynamic effects, moving gas and dust around and thus potentially triggering or disrupting star formation processes (e.g. Murray et al., 2010; Sales et al., 2014).

1.2.2 The Nature of the Interstellar Radiation Field

The base case source of electromagnetic radiation in the Galactic ISM is the ambient interstellar radiation field ('ISRF'), an effectively isotropic energy density field due to the general population of stars in the galaxy augmented by the cosmic microwave background ('CMB'). In decreasing order of energy density (in galactic locations away from plasma regions), Draine (2011) summarised the dominant intragalactic contributions as follows:

- 'Starlight' – radiation from stellar photospheres;
- Dust – reprocessed starlight absorbed and reemitted in the IR and FIR part of the spectrum by dust grains;
- Nebular emission – bound-bound, bound-free and free-free ('Bremsstrahlung') plasma emission from regions with plasma temperatures of $\sim 10^4$ K (e.g. HII regions);
- X-rays – emission from hot plasma regions with temperatures of $\sim 10^5$ - 10^8 K such as the accretion discs surrounding X-ray binaries; and
- Radio wavelength synchrotron radiation – predominantly from relativistic electrons spiraling around galactic magnetic fields.

The energy densities associated with each of these ISRF contributions in the solar neighbourhood (well away from plasma regions) are shown in Table 1.1.

ISRF Component		u_{rad} / erg cm^{-3}
Radio synchrotron		2.7×10^{-18}
CMB		4.2×10^{-13}
Dust IR emission		5.0×10^{-13}
Nebular bremsstrahlung	4.5×10^{-15}	
Nebular Hα & other bound-bound	1.7×10^{-15}	
Nebular emission total		6.2×10^{-15}
Starlight (3,000 K)	4.3×10^{-13}	
Starlight (4,000 K)	3.2×10^{-13}	
Starlight (7,500 K)	2.4×10^{-13}	
Starlight (UV)	7.1×10^{-14}	
Starlight total		1.1×10^{-12}
Soft X-ray		1.0×10^{-17}
ISRF total		$2.\times10^{-12}$

Table 1.1. Energy densities associated with the components of the ISRF in the solar neighbourhood (Draine, 2011)

Table 1.1. demonstrates that the dominant contributions to energy density in the local ISRF are from starlight, dust and the CMB. The CMB spectrum is well known, being an almost perfect black body spectrum at a temperature of 2.73 K. The starlight and dust components have been described by Mathis et al. (1983) in the range 91 ηm to 1 mm, adopting previous work by Mezger et al. (1982) to describe the emission spectrum of dust in the range above 8 μm. Mathis et al. (1983) showed that at wavelengths in the range 245 ηm to 8 μm, the starlight contribution to energy density, u_ν, at frequency ν in the solar neighbourhood is approximated by the sum of the energy densities associated with three diluted black body spectra:

$$\nu u_\nu = \frac{4\pi}{c}\sum_{i=1}^{3}\nu J_\nu = \frac{4\pi}{c}\sum_{i=1}^{3} W_i \nu B_\nu = \frac{8\pi h}{c^3}\sum_{i=1}^{3} W_i \nu^4 \left(\exp\left(\frac{h\nu}{kT_i}\right) - 1\right)^{-1} \quad (1.1)$$

where J_ν is specific intensity, B_ν is the Planck function and W_i and T_i are the dilution factor and temperature associated with the three components. The first two components with temperatures of 4,000 K and 7,500 K and respective dilution factors of 1.65×10^{-13} and 1×10^{-14} account for the general population of main sequence stars while the third component with temperature of 3,000 K and dilution factor of 7×10^{-13} accounts for the population of red giants.

At wavelengths in the range 91 - 245 ηm, Mathis et al. (1983) found that the ISRF in the solar neighbourhood can be approximated by a fourth component due to UV emission from hot, high-mass stars with a broken power-law fit:

$$\nu u_\nu = 2.373\times10^{-14}\lambda^{-0.6678} \text{ for } \lambda = 0.134 - 0.245 \text{ μm} \quad (1.2)$$

$$\nu u_\nu = 6.825\times10^{-13}\lambda \text{ for } \lambda = 0.110 - 0.134 \text{ μm} \quad (1.3)$$

$$\nu u_\nu = 1.287\times10^{-9}\lambda^{4.4172} \text{ for } \lambda = 0.091 - 0.110 \text{ μm} \quad (1.4)$$

where λ is the wavelength in microns and u_ν is the specific energy density in erg $\text{cm}^{-3}\text{Hz}^{-1}$.

The spectrum resulting from the 4 components described above is shown in Figure 1.1 - reproduced from Mathis et al. (1983) - in units of $4\pi\lambda J_\lambda$ to allow more straightforward comparison between the various wavelength ranges.

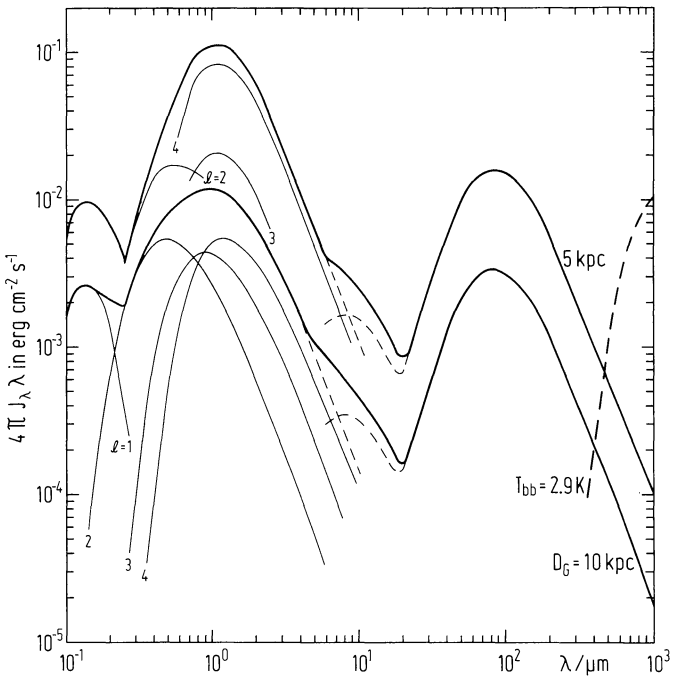


Figure 1.1. Reproduced from Mathis et al. (1983). The ambient ISRF at distances of 5 and 10 kpc from the Galactic centre. Curves labeled l = 1 – 4 relate to the 4 contributions from starlight in decreasing order of effective temperature.

It should be noted that an alternative description of the ISRF based on data from the Cosmic Background Explorer ('COBE') as reported by Black (1994) combined with a polynomial fit to the UV portion of the ISRF reported by Draine (1978) is also in widespread use in ISM models.

Recognising the importance of photoelectric heating of the ISM, the relative strength of the UV portion of the ISRF density, termed G_0, is conventionally referenced to an energy density of 5.29×10^{-14} erg cm^{-3} in the range 6 – 13.6 eV known as the 'Habing Field' (Draine, 2011), i.e.:

$$G_0 = \frac{u(6 - 13.6 \text{ eV})}{5.29\times10^{-14} \text{ erg cm}^{-3}} \quad (1.5)$$

This reference energy density is calculated from an early estimate of the intensity of the ISRF in the solar neighbourhood in the range 912 - 2400 Å reported by Habing (1968), with the rationale that photons with

energies greater than the ionisation energy of hydrogen (13.6 eV) will be largely absent, having been absorbed elsewhere by hydrogen atoms, and photons with energies < 6 eV will be below the photoelectric work function of a typical dust grain.

1.2.3 The Evolution and Associated Luminosity of Young Stellar Objects

The initial stage of the collapse of a molecular cloud core is isothermal since the cloud is optically thin to emission and compressional heating due to the increasing density and pressure is much less than the radiative cooling capacity, down to very small masses – approximately 0.015 $M_{\odot}$ for clouds with contemporary metallicity (Ward-Thompson and Whitworth, 2011). The isothermal condition will end when the optical depth of the cloud, mainly due to dust opacity, becomes approximately 1 and an adiabatic core forms at its centre. This marks the end of the 'first collapse' (Masunaga et al., 1998). The temperature of this 'pre-stellar core' can now increase until it reaches a state close to hydrostatic equilibrium; further contraction is now quasi-static ('Kelvin-Helmholtz' contraction) with half the released gravitational energy converted to the internal energy necessary to maintain hydrostatic equilibrium and the remaining half converted into luminosity as predicted by the virial theorem (Prialnik, 2009) i.e. given gravitational potential energy Ω at time t, the core luminosity due to contraction of the protostar:

$$L_{core} = \frac{1}{2}\left(-\frac{d\Omega}{dt}\right) \tag{1.6}$$

The second collapse (Masunaga and Inutsuka, 2000) occurs when the central temperature reaches ~2,000 K, at which point hydrogen molecules begin to thermally dissociate, consuming released gravitational energy which would otherwise go towards heating and supporting the pre-stellar core. The second collapse ends once dissociation is complete, leaving a core now sufficiently evolved to be considered a 'protostar' at the centre of the cloud.

At approximately the same temperature (~2,000 K), dust will evaporate and no longer be the main source of opacity within the protostar. The protostar continues to contract quasi-statically with its subsequent evolution on to the main sequence determined by whatever is now the main source of opacity within the protostar at any point in time. Except for the most massive stars, the main source of mean opacity, $\bar{\kappa}$, will initially be the H^- ion. This leads to a convective interior ($\bar{\kappa} \propto T^{9/2}$) and a protostar that will follow a 'Hyashi' track (on the conventional Herzsprung-Russell luminosity-temperature diagram) towards the main sequence, where hydrostatic equilibrium is now maintained by nuclear burning rather than gravitational collapse (Ward-Thompson and Whitworth, 2011).

As the star continues to contract and heat up, the opacity of the stellar interior changes to follow Kramer's opacity law ($\bar{\kappa} \propto T^{-7/2}$) and at very high temperatures or low densities is mainly due to electron scattering ($\bar{\kappa}$ constant). In either case, the interior is now predominantly radiative and the protostar will follow a 'Henyey' track (on the conventional Herzsprung-Russell luminosity-temperature diagram) on to the main sequence (Ward-Thompson and Whitworth, 2011). Once on the main-sequence, hydrostatic equilibrium is maintained and core luminosity generated by nuclear burning rather than gravitational collapse. Although not sufficient to halt collapse in protostars above the mass of brown dwarves, deuterium burning (which can begin at lower core temperatures of ~ 10^6 K and therefore earlier then main-sequence H-burning) can also play an important role in both establishing quasi-static equilibrium and generating core luminosity during the journey of a protostar to the main sequence (Krumholz et al., 2012).

During all of the above phases of core contraction (and even during the early part of main sequence life), the remaining 'envelope' of cloud matter from which the protostar formed continues to infall, accreting onto the surface of the protostar where it will both add mass to the protostar and be rapidly thermalized at an accretion shock front, generating an accretion luminosity given by:

$$L_{acc} = \frac{GM\dot{M}}{R} \tag{1.7}$$

where M is the mass of the protostar, $\dot{M}$ is the accretion rate on to the protostar and R is the radius of the accretion shock front.

In numerical modeling of systems where accretion is occurring onto collapsing protostars, Hosokawa and Omukai (2009) have found that the net behaviour of a such systems can be divided into four phases: three protostellar phases and a fourth main-sequence phase characterised by the onset of core hydrogen burning.

Transition between the three protostellar phases is fundamentally determined by the balance between the accretion timescale:

$$t_{acc} = \frac{M}{\dot{M}} \tag{1.8}$$

and the Kelvin-Helmholtz timescale:

$$t_{KH} = \frac{GM^2}{RL_{core}} \tag{1.9}$$

An initial 'adiabatic accretion' phase is characterised by $t_{acc} < t_{KH}$, such that the increase in protostellar radius due to mass accretion is faster than the decrease due to contraction; the radius thus increases and luminosity is dominated by accretion luminosity. Conversely, during the 'Kelvin-Helmholtz contraction' phase which immediately precedes entry on to the main sequence, $t_{acc} > t_{KH}$, the protostellar radius shrinks and luminosity will be predominantly due to the core luminosity. Between these two phases a transitional 'swelling' phase has been identified associated with the redistribution of entropy within the protostar following the onset of deuterium burning. During this phase the radius is believed to increase rapidly by a factor of around 3 and core luminosity to be temporarily dominated by deuterium burning.

The evolution of radius and luminosity associated with these four phases is illustrated in Figures 1.2 and 1.3, reproduced from Hosokawa and Omukai (2009).

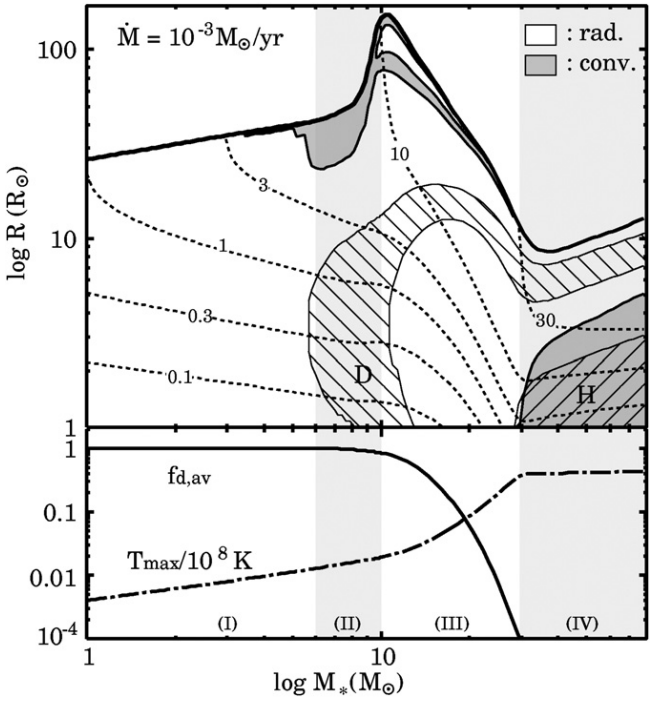


Figure 1.2. Reproduced from Hosokawa & Omukai (2009). Top panel: The evolution of the radius of the accretion shock front in an accreting protostar with $\dot{M} = 10^{-3}\, M_{\odot} yr^{-1}$; the dotted lines represent the loci of mass coordinates, convective areas are shaded grey and the hatched areas indicate where nuclear burning is occurring. Lower panel: Evolution of the mass averaged deuterium concentration and the maximum temperature within the star. In both panels the shaded background indicates the 4 evolutionary phases: I: adiabatic accretion; II: swelling; III: Kelvin-Helmholtz contraction and IV: main-sequence accretion.

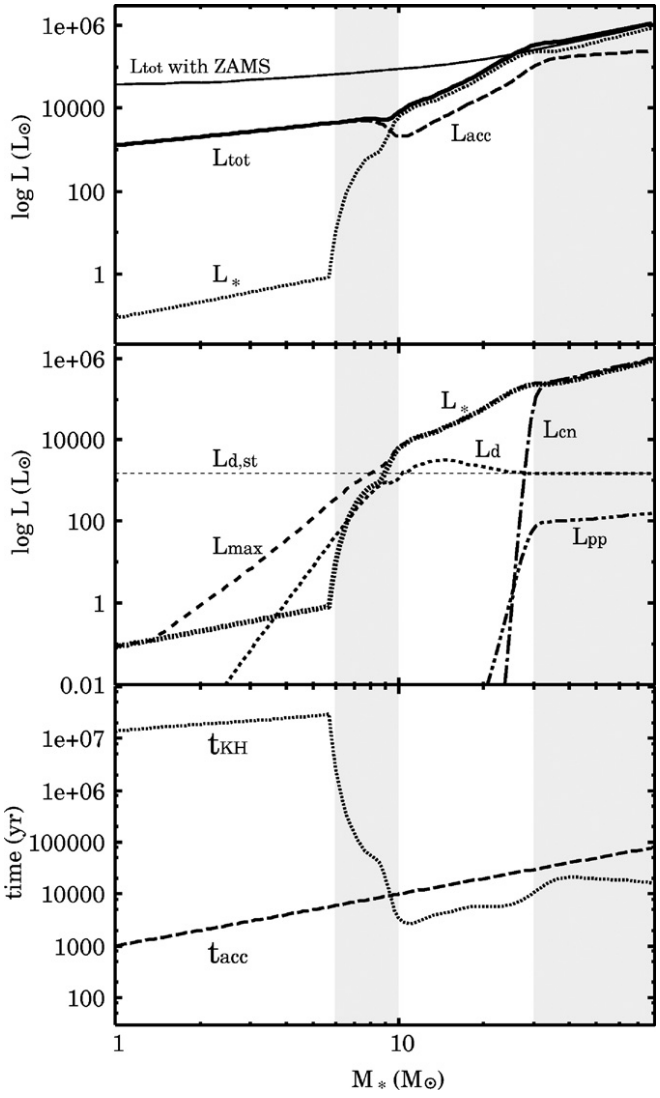


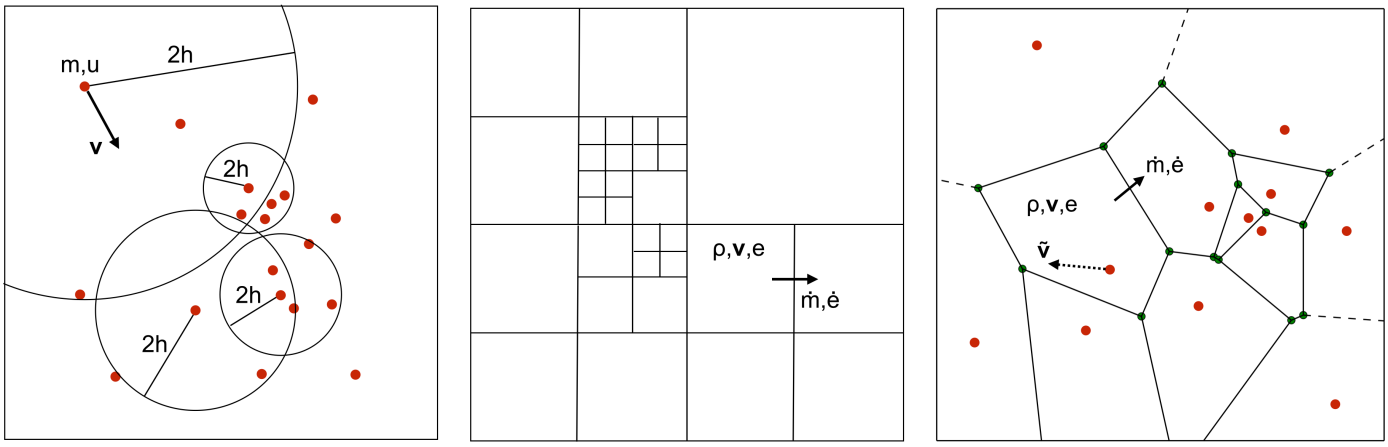
Figure 1.3. Reproduced from Hosokawa & Omukai (2009). Top panel: Evolution of the total, accretion and core luminosity of an accreting protostar with $\dot{M} = 10^{-3}\, M_{\odot} yr^{-1}$ compared to the main-sequence luminosity of an equivalent mass star. Middle panel: The contribution of D, H pp chain and H CN cycle to the core luminosity. Lower panel: Evolution of the accretion and Kelvin-Helmholtz timescales. For all panels the shaded background indicates the 4 evolutionary phases as in Figure 1.2.

1.3 Astrophysical Hydrodynamic Simulation Codes

To understand the environment into which the new algorithm is being implemented and appreciate where AREPO sits in the evolution of hydrodynamic numerical simulation codes, a brief outline of the basis of such models is given below together with an overview of associated radiative transfer models. For a more comprehensive review see Dale (2015).

1.3.1 Overview

The vast majority of complex numerical models of the ISM (and indeed the cosmos) are based on the assumption that the ISM is reasonably approximated as a smooth fluid. This assumption is justified on the basis that the mean free path of its constituent particles is significantly less than the size of the structures in which they move. As with all fluid dynamics problems, in order to make useful predictions from this starting point it is then necessary discretise the fluid in some manner. Classically, two approaches have been used: the Eulerian approach which considers the evolution of fluid parameters with time at a fixed spatial position and the Lagrangian approach which considers the evolution of fluid parameters with time in a fixed volumetric element comoving with the flow (Clarke and Carswell, 2007). These two classical approaches are mirrored in the two main methods that have evolved within astrophysical fluid dynamics codes together with the more recent hybrid approach which is the main tool of this project. These three methods are illustrated in Figure 1.4 - reproduced from Dale (2015) - and summarised below:



(a) SPH (b) AMR (c) Moving mesh

Figure 1.4. Reproduced from Dale (2015). Schematic illustration of the three main categories of astrophysical hydrodynamics code.

a. 'Particle' codes, for example 'SEREN' (Hubber et al., 2011), are Lagrangian in nature, describing the fluid as a collection of particles, each carrying mass, velocity and internal energy. After each time-step the position and trajectory of each particle is calculated from the forces exerted by the other particles. Denser regions of gas will be associated with higher concentrations of particles. Modern particle codes use 'Smooth Particle Hydrodynamics' ('SPH') to minimise the inherent noisiness of particle models. This is achieved by defining a 'smoothed' particle with a radius $2h$ enclosing an appropriate number of neighbouring particles (e.g. five in Figure 1.4).

b. 'Grid' codes, for example 'FLASH' (e.g. Fryxell et al., 2000), are Eulerian in nature, dividing the fluid up into a grid of volume cells then calculating the density, velocity and internal energy in each cell as the flow evolves by computing the exchange of mass, momentum and energy between cells. In 'Adaptive Mesh Refinement' ('AMR') schemes the grid is refined automatically in order to increase the resolution in the most interesting regions. Whilst grid codes model shocks and contact discontinuities more effectively than particle codes, they notoriously suffer from 'Galilean non-invariance', the phenomenon where different solutions result from initial conditions that vary only with respect to bulk velocities relative to the fixed computational grid.

c. 'Moving mesh' codes attempt to combine the relative advantages of the above two approaches to fluid dynamics. In moving mesh codes the simulation volume is divided into a dynamic grid of 'cells' drawn up according to the principle of Voronoi tessellation at each time step. Each cell is therefore associated with a 'particle' positioned at the centre of gravity of the cell and carrying velocity and acceleration vectors in addition to the scalar quantities such as mass, density and temperature associated with its cell. The cells exchange mass, momentum and energy with neighbouring cells as in a grid code; however the cells also move with the flow of their associated particles, altering their shape according to the principle of Voronoi tessellation after every time step. Cell volume will thus depend on the density of particles in the immediate surroundings, thus increasing resolution automatically in areas of interest. In theory, therefore, moving mesh codes retain the benefit of both grid codes and particle codes while eliminating their respective drawbacks. This project uses the state-of-the-art moving-mesh code 'AREPO' (Springel, 2010; www.illustris-project.org, 2017).

1.3.2 RADIATIVE TRANSFER

To connect a luminous source to a region of interest within a numerical model, it is necessary to understand how a spectral distribution of radiation will be altered and attenuated as it travels from its source to its destination and how this might best be numerically modeled. Except in the very simplest of cases, radiative transfer calculations usually require a numerical solution, since transfer between 2 non-neighbouring fluid elements, depends on the properties of all the intervening elements and will be frequency dependent. A number of algorithms have consequently been developed, which, broadly speaking, can be categorised as 'ray-tracing' methods, moment methods and Monte Carlo methods. A comprehensive overview of all of these methods is again provided by Dale (2015).

Arguably the most intuitive are the so-called 'ray-tracing' methods which, stated simply, just involve drawing a line between the source and the target then solving the radiative transfer equation:

$$\frac{\partial I_\nu}{\partial s} = \eta_\nu - \chi_\nu I_\nu \tag{1.10}$$

along the ray path, s, to determine the variation of specific intensity I_ν at frequency ν along a path with volume emissivity η_ν and opacity χ_ν. This has the general solution:

$$I_\nu = I_{\nu,0} e^{-\tau_\nu} + \int_0^{\tau_\nu} S_\nu \, e^{-\tau_\nu} \mathrm{d}\tau_\nu \tag{1.11}$$

where $I_{\nu,0}$is the specific intensity at frequency ν at the start of the ray, S_ν is the ratio η_ν / χ_ν known as the source function and τ_ν is the optical depth.

The first common approximation that allows a more straightforward solution of equation (1.11), both within and without numerical models, are that the medium is approximately uniform and hence that that S_ν is a constant. In this case, equation (1.11) simplifies to:

$$I_\nu = I_{\nu,0} e^{-\tau_\nu} + S_\nu (1 - e^{-\tau_\nu}) \tag{1.12}$$

A further, and often not unrealistic approximation, is that the emissivity of the medium is negligible, so that S_ν is zero and equation (1.12) simplifies to:

$$I_\nu = I_{\nu,0} e^{-\tau_\nu} \tag{1.13}$$

The problem has now become one of just determining optical depth, and since optical depth:

$$\tau_\nu = \int_0^s \sigma_\nu n \, \mathrm{d}s \tag{1.14}$$

where σ_ν is the mean cross-section and n is the volume number density ($\chi_\nu = \sigma_\nu n$), then if σ_ν is also assumed to be constant, the problem can be reduced further to the determination of column density:

$$\Sigma = \int_0^s n \, \mathrm{d}s \tag{1.15}$$

since now:

$$\tau_\nu = \sigma_\nu \Sigma \tag{1.16}$$

The ray-tracing algorithm adopted in this project is the 'TreeCol' method developed by Clark et al. (2012) and now implemented within AREPO (Smith et al., 2014). This method exploits information already determined by tree-based gravity solvers, which are widely deployed in numerical models, to determine a spherical map of column densities around any point in a simulation in a computationally 'cheap' manner.

The generic algorithm associated with tree-based gravity-solvers is illustrated in Figure 1.5 and proceeds as follows:

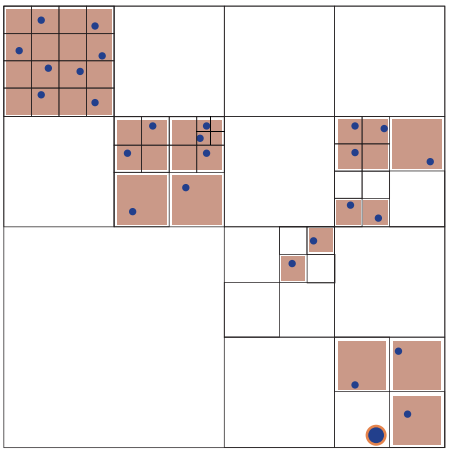


Figure 1.5: Reproduced from Clark et al. (2012). Schematic 2D illustration illustrating how a tree is constructed in a gravitational force calculation.

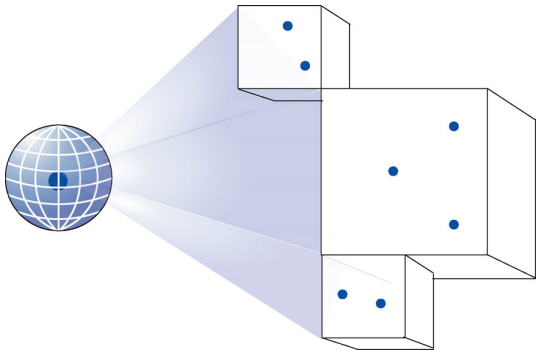


Figure 1.6: Reproduced from Clark et al. (2012). Schematic illustration of how projected column densities are mapped onto a spherical grid in TreeCol.

- The simulation volume is sequentially split into a hierarchical series of nested boxes or 'nodes' beginning with a 'root' node that contains the whole volume; this is then divided into 'daughter' nodes (typically 8).
- Any daughter node that contains multiple particles is subdivided as above until each node contains a single particle known as a 'leaf'.
- Once this 'tree' of nodes is built, it is 'walked' in a manner defined by an 'opening angle' from the particle of interest with the hierarchy of the node used in the gravitational calculation being the one whose angular diameter, as seen from from the particle of interest, is the first one to be less than the opening angle.
- If the selected node is a leaf, the position of the particle itself is used to calculate the gravitational force; however, if the selected node contains more than one particle the average mass and position are used.
- The opening angle is selected so that nodes with greater than one particle are far enough away that their masses and positions can be averaged within the gravitational calculation without any significant loss of accuracy, whilst facilitating considerably higher computational efficiency compared to calculating the gravitational effect of each particle individually.

The TreeCol scheme recognizes the fact that the information necessary to calculate the column density along a path between a particle and the edge of the simulation, namely the density, position and angular extent of each node is now available as a result of the gravitational tree building process. TreeCol then utilises the HEALPix algorithm (Górski et al., 2005; healpix.sourceforge.net, 2017) to assign this column density to a sphere of equal area 'pixels' surrounding the particle under consideration at the limits of the simulation (Figure 1.6) - see Clark et al. (2012) for a more detailed description of the assignment methodology.

Since its introduction, TreeCol has been implemented in SPH codes (e.g. Clark and Glover, 2015), grid codes (Gatto et al., 2017) and within AREPO (Smith et al., 2014). Additionally Hartwig et al. (2015) have introduced refinements to the TreeCol algorithm that are able to take account of the Doppler effect when particles contributing to the column density have significant relative velocities.

1.4 Existing Numerical Models of Internal Feedback

For a thorough review of simulations designed to evaluate the impact of radiative and non-radiative feedback mechanisms at molecular cloud scales or below up until mid 2015, the reader is referred to sections 5.1 – 5.5 of Dale (2015). In summary, Dale (2015) found that inclusion of internal (embedded) or external sources of radiative feedback when modeling collapsing turbulent clouds consistently reduces star formation rates and efficiency.

Whilst quite a number of studies of the potential effects of radiative feedback at molecular cloud scale have been reported, a majority of these consider external feedback sources, typically OB type stars (e.g. Dale et al., 2013). However, a number of studies within the period of Dale's review are of particular relevance to internal feedback from embedded protostars, the focus of this project, and are consequently worth highlighting.

This includes the work of Offner et al. (2009) who used an AMR grid code to determine that the energy from protostellar radiation exceeds that from viscous dissipation and compressional heating by at least an order of magnitude once star formation reaches a steady state and highlight that models that omit such radiative feedback from protostars are likely to significantly underestimate the gas temperature and overestimate star formation efficiency. The same authors also developed a useful algorithm for tracking the evolution and associated luminosity of 'star particles'.

Bate (2009), simulating accretion feedback in a 50 $M_{\odot}$ turbulent cloud using an SPH code and a flux-limited-diffusion (moment) radiative transfer method, found that the number of stars formed was reduced by a factor of around 4 compared to a base case without feedback and, interestingly, that the feedback had the effect of decoupling the mean stellar mass formed from the initial mass of the cloud. Subsequent work (Bate, 2014), which extended the cloud mass to 500 $M_{\odot}$, has demonstrated a statistically excellent fit between the mass-function produced from the simulation and those observed in nature.

Also of particular interest, Krumholz et al. (2011), simulating feedback from protostars in a 1,000 $M_{\odot}$ cloud using an AMR grid code, found that inclusion of feedback led to a smaller number of stars forming and even shut down fragmentation altogether. In later work, Krumholz et al. (2012) go on to suggest that star formation rate and mass function, which at first sight seem only weakly related become very closely coupled when feedback from protostars is included. Other relevant findings by this group include the suggestion that internal radiative feedback from low-mass stars may play a crucial role in the formation of high-mass stars (Krumholz and McKee, 2008) and the observation that luminosity originating from Kelvin-Helmholtz contraction and deuterium burning is an essential component of feedback in simulations of embedded protostars (Krumholz et al., 2010).

Since Dale's review, Lomax et al. (2014) and Lomax et al. (2015) in their modeling studies of the L1688 star-forming cloud in the ρ-Ophiuchi cloud complex using an SPH code, find that the temporal variability of radiative feedback is important, demonstrating that an acceptable reproduction of observed IMF and stellar multiplicity requires the radiative feedback to be episodic. Also of note, Howard et al. (2017) have used an AMR grid code to explore how radiative feedback affects the early evolution (< 5 Myrs) of molecular clouds, concluding that inclusion of radiative feedback lowers the efficiency of cluster formation by 20-50% and noting that good agreement between simulated cluster mass distributions and observed cluster mass functions is only achieved for simulations including radiative feedback..

Finally, Sales et al. (2014) have reported a previous use of AREPO in feedback studies, using it to examine the relative impact of radiation pressure and photoionisation feedback from young OB stars on the surrounding gas. However, this study used a moment-based radiative transfer scheme - as opposed to the TreeCol scheme to be used in this project - and again focused on the impact of external sources of radiation from high-mass main-sequence stars – as opposed to the effect of multiple embedded protostars with a range of mass which is the focus of this project.

2 METHODOLOGY

2.1 GENERAL CONSIDERATIONS

In order to observe and analyse the effect of introducing internal radiative feedback into AREPO, it is necessary to first develop a base case model using the existing code (without radiative feedback) into which the new algorithm can be introduced. This is described in terms of its key parameters below followed by a description of the isotropic and anisotropic feedback algorithms developed for this project and the analytical calculations performed to test the code.

Code amendments were focused on the ***sgchem.c*** source code within AREPO, written in the standard C programming language. This code already contains the complex rates and microphysics governing photo-ionisation/heating mechanisms and photochemical reactions (e.g. Bertram et al., 2016) and is assumed to be functioning as intended. Likewise the TreeCol radiative transfer code and, for that matter, the remainder of the AREPO code is assumed to be functioning 'correctly'.

Note that the unit of distance used by AREPO is 1×10^{17}cm, the unit of mass 1.991×10^{33}g and the gravitational constant, G, is set to 1. This results in a unit of time of 2.744×10^{12} s (87,000 years) and a unit of density of 1.991×10^{-18} g cm^{-3} . These units are hereinafter termed 'code units'.

2.2 BASE CASE MODEL

The base case model consists of a spherical overdensity ('cloud') of 200,000 cells with a mass of 1,000 $M_{\odot}$, a radius of approximately 1.6 pc (49.575 code units), an initial temperature, T, of 10 K and a gas particle mass, m_p, of 2.33 m_H. These parameters are designed to ensure that the cloud should collapse, by making the radius of the cloud significantly less than the Jeans radius at this temperature and the column density larger than the thresholds presented by Clark and Glover (2014). The Jean's radius (Ryan and Norton, 2010) for a uniform density sphere is calculated from:

$$R_{max} = \frac{2GM}{5a_0^2} \tag{2.1}$$

where G is the gravitational constant, M is the mass of the cloud and a_0 is the speed of sound in the cloud ($= (k_B T/m_p)^{1/2}$) For the above cloud parameters, this equates to approximately 48 pc – more than 30 times the cloud radius in the model.

The base case model additionally includes bulk turbulence, generated using the model developed by Girichidis et al. (2014) with a velocity field strength defined by a gravitational potential to kinetic energy ratio of 2. Finally, each side of the volume 'box' in which the cloud sits is given dimensions of 2.5 times the diameter of the cloud (~248 code units / 8 pc) and the volume outside the cloud is 'padded' with an additional 4,000 low-density cells. The cloud has no bulk rotation and no magnetic fields are included.

The adopted code treats the ISRF applicable to any cell as an isotropic sphere or external 'bath' of energy density surrounding that cell with a magnitude defined as a multiple of the Habing field, i.e. G_0 (see section 1.2.2). This project adopted the starlight and dust spectra of the ambient ISRF described by Mathis et al. (1983) at a distance of 10 kpc from the galactic centre as a base case. This has an associated ambient G_0 of 1.15. The existing code also includes an ambient background of cosmic rays.

Except in cases where it was necessary to assume that the cloud is optically thin (section 2.5), the existing TreeCol algorithm (see section 1.3.2) is used to determine column densities surrounding the cell under consideration – using a 48 pixel HEALPix sphere - and hence a mean attenuation coefficient, χ (Clark et al., 2012; Goldsmith, 2001) for the radiation reaching that particle.

A 'sink' particle code is included that forms sinks when the density of a cell exceeds 20 code units (~4×10^{-17}g cm^{-3}). Such sites are taken to be the centres of star formation activity; however within the non-feedback base case model, such 'protostars' interact with the rest of the model through gravity alone. Sink particles have a known mass within the model and continue to accrete mass from the surrounding cloud with time.

2.3 ISOTROPIC FEEDBACK MODEL

Working within the existing coding architecture of AREPO, a 'first-pass' feedback model involves calculating the overall impact on energy density of nearby luminous sources at the cell under consideration and hence determining a factor by which G_0 has increased relative to that associated with the ambient ISRF alone - in this case that described by Mathis et al., (1983) with a G_0 of 1.15. This factor is then applied by AREPO to the ambient ISRF spectrum, increasing the spectral intensity across its frequency range by this single factor and thus generating an enhanced ISRF which continues to be considered to be isotropically distributed around the cell under consideration.

As this work only considers the pre-stellar and adiabatic accretion phase of protostellar evolution - when the luminosity is totally dominated by accretion luminosity (Hosokawa and Omukai, 2009) - the algorithm used to determine the incremental G_0 due to a particular sink begins by determining the accretion rate onto the sink particle in question. Following the protostellar evolution model described above, in particular the observation by Masunaga et al. (1998) that the mass of the first pre-stellar core reaches a mass of ~0.05 $M_\odot$ regardless of the mass and initial density profile of the parent cloud, the pre-stellar phase is associated with sink particle masses < 0.05 $M_\odot$. Consequently sink particles with masses < 0.05 $M_\odot$ are assigned zero luminosity, noting that this is somewhat higher than the cut-off of 0.01 $M_\odot$ adopted by Offner et al. (2009). For masses ≥ 0.05 $M_\odot$, the work of Maschberger et al. (2014) is used. This study found that accretion rates appear to be effectively constant at around 5×10^{-6} $M_\odot$ yr^{-1} for sink particle masses <0.5 $M_\odot$ and that accretion rates for sink particle masses ≥ 0.5 $M_\odot$ follow a power law approximated by:

$$\dot{M} \sim 7.9\times10^{-6} M^{0.66} \ M_\odot \ yr^{-1} \tag{2.2}$$

The radius of the accretion shock is then calculated using the mass-radius relationship reported by Hosokawa and Omukai (2009) - as originally derived by Stahler et al. (1986) - on the basis that opacity in the underlying protostar is dominated by the H^- ion and heat transfer in the core is consequently convective:

$$R_{core} \approx 26 R_\odot \left(\frac{M}{M_\odot}\right)^{0.27} \left(\frac{\dot{M}}{10^{-3} M_\odot \ yr^{-1}}\right)^{0.41} \tag{2.3}$$

Making the assumption that the effective photospheric radius $R_{eff} \approx R_{core}$, the accretion luminosity of the sink particle is determined straightforwardly from:

$$L = \frac{GM\dot{M}}{R_{eff}} \tag{2.4}$$

and, assuming the accreting protostar emits as a black body, the effective photospheric temperature from:

$$T_{eff} = \left(\frac{L}{4\pi\sigma_{SB} R_{eff}^2}\right)^{0.25} \tag{2.5}$$

where σ_{SB} is the Stefan-Boltzmann constant.

As G_0 is defined as the ratio of electromagnetic radiation energy density in the range 6 – 13.6 eV relative to the Habing field (5.29×10^{-14} ergs cm^{-3} in the same energy range) it is next necessary to determine the fraction of the bolometric accretion luminosity which lies in this energy range, termed 'UV fraction' for the purposes of this project. Again assuming the accreting protostar emits as a black body, the UV fraction will be given by:

$$UV_{frac} = \frac{L_{uv}}{L_{bol}} = \frac{\int_{\nu_1}^{\nu_2} B_\nu \, d\nu}{\int_0^\infty B_\nu \, d\nu} \tag{2.6}$$

where ν_1 and ν_2 are the frequencies associated with the energy range above (approximately 1.4×10^{15} and 3.3×10^{15} Hz) and B_ν is the Planck function:

$$B_\nu = \frac{2h\nu^3}{c^2}\left(\exp\left(\frac{h\nu}{kT}\right) - 1\right)^{-1} \tag{2.7}$$

Using the standard substitution $u = h\nu/kT$, the expression for the UV fraction becomes:

$$UV_{frac} = \frac{\int_{u_1}^{u_2} \frac{u^3}{e^u - 1}\,\mathrm{d}u}{\int_0^{\infty} \frac{u^3}{e^u - 1}\,\mathrm{d}u} = \frac{15}{\pi^4}\int_{u_1}^{u_2} \frac{u_3}{e^u - 1}\,\mathrm{d}u \tag{2.8}$$

which has been analysed for temperatures between 6,000 K and 10,000 K as shown in Figure 2.1.

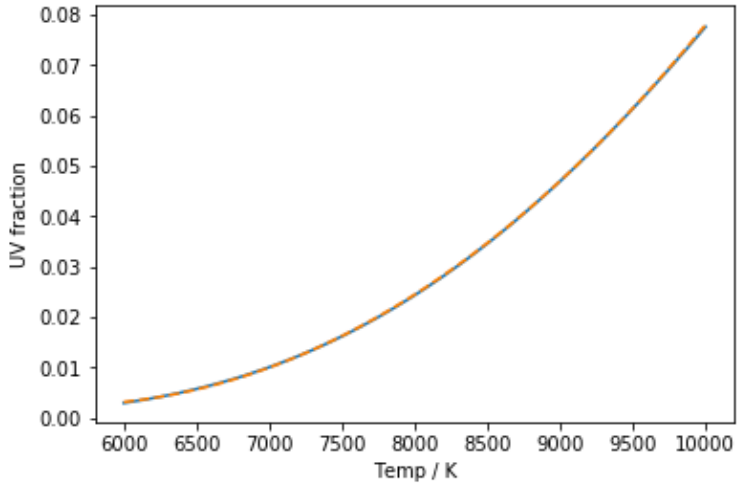


Figure 2.1. The variation of 'UV fraction' (6-13.6 eV) of a black body with the temperature of the black body (solid blue line) overlain (dashed orange line) by the polynomial fit $uv_{frac} = 0.1 - 3.3\times10^{-5}T + 2.4\times10^{-9}T^2 + 6.8\times10^{-14}T^3$

For reference, a 0.05 $M_\odot$ accreting protostar would have an effective temperature of approximately 7,850 K and a 100 $M_\odot$ accreting protostar an effective temperature of approximately 9,200 K assuming the above method. As is also shown in Figure 2.1 the UV fraction can be well fitted with the polynomial:

$$uv_{frac} = 0.1 - 3.3\times10^{-5}T + 2.4\times10^{-9}T^2 + 6.8\times10^{-14}T^3 \tag{2.9}$$

and to maximize computational speed this is the equation used to determine the UV fraction in the code.

With the UV luminosity (in the Habing standard energy range) known and the distance, D, between each sink and the cell under consideration straightforwardly calculated from their relative coordinate positions already in the code, the incremental flux, F_{uv}, in the Habing energy range at the cell due to a sink is calculated from:

$$F_{uv} = \frac{L_{bol}\times uv_{frac}}{4\pi D^2} \tag{2.10}$$

and this is equated to an incremental G_0 (energy density expressed as a fraction of the Habing energy density) acting isotropically on the particle via:

$$G_{0,incr} = \frac{4F_{uv}/c}{5.29\times10^{-14}\text{ergs cm}^{-3}} \tag{2.11}$$

The code loops around all the sinks to determine the total incremental G_0 due to all the sink particles which is then added to the ambient G_0 before being passed back to AREPO as the prevailing G_0 at the location of the cell under consideration.

2.4 Anisotropic Feedback Model

The isotropic ISRF model is not, intuitively, an accurate model of the feedback from nearby luminous sources as attenuation is averaged over the entire sphere of pixels within TreeCol. Continuing to work within the existing AREPO architecture, this situation can be improved by associating each sink particle with a particular pixel in the HEALPix sphere surrounding the cell under consideration and then calculating a mean attenuation coefficient, χ, *weighted* by the relative contribution to G_0 of each pixel.

The function for determining the pixel number given the position vector of a sink particle (***vec2pix_ring.c***) is already available within the HEALPix software package (Górski et al., 2005; healpix.sourceforge.net, 2017) and has been incorporated into the code developed for this project. The sink particles were confirmed as being assigned to the correct pixel using the complementary function ***pix2ang_ring.c***, which returns the spherical coordinates of an orthogonal vector through the centre of the pixel. This is compared against the spherical

coordinates of the vector from the cell under consideration to the sink with a tolerance given by the mean spacing between pixels of 29.3 degrees for a 48 pixel HEALPix sphere.

The concept behind both the isotropic and anisotropic feedback models is illustrated in Figure 2.2.

The resultant ***sgchem.c*** code is included as Appendix II with the amendments to the existing code highlighted in red.

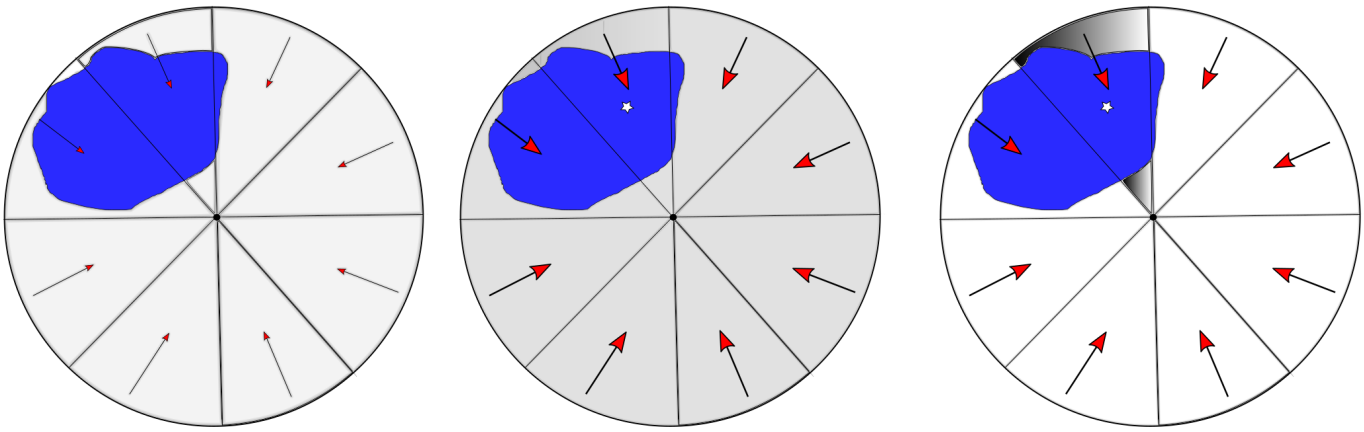


Figure 2.2. Schematic 2D illustration of the radiative feedback algorithm within sgchem.c. In the absence of any sinks within a denser region of the cloud (shown in blue), the radiation at a cell (at the centre of the circle of 2D pixels) is due to the ambient ISRF only and the mean attenuation is calculated based on an equal weighting of the column densities assigned to each pixel. In the isotropic feedback model, the radiation is enhanced by the contribution from the sink but the mean attenuation is still calculated based on an equal weighting of the column densities assigned to each pixel. In the anisotropic feedback model, the augmented radiation is still considered to be isotropic but the mean attenuation is now calculated based on the contribution to column density for each pixel weighted by the contribution to the total radiation field (ambient ISRF + sinks).

2.5 'Analytical' Dust Temperature Calculations

In order to gain confidence in the veracity of the AREPO output (section 3.2), calculations were performed from first principles to determine theoretical dust temperatures surrounding a 5 $M_\odot$ accreting protostar in an optically thin environment and compared against the dust temperature output from an equivalent AREPO simulation where sink particles were forced to a mass of 5 $M_\odot$, the environment is taken to be optically thin (TreeCol deactivated) and the simulation terminated once the first sink particle had formed. These calculations follow the method outlined by Clark et al. (2012) in which the optically thin heating rate of the dust is given by:

$$\Gamma_{d,0} = 4\pi D\rho \int_0^\infty J_\nu \kappa_\nu \, \mathrm{d}\nu \tag{2.12}$$

where D is the dust/gas ratio, ρ is the gas density, J_ν is the mean specific intensity of the incident radiation and κ_ν is the specific opacity (effective cross-section per unit mass) of the dust particles. The converse dust cooling rate, assuming that the dust particles radiate as black bodies, is given by:

$$\Lambda_d = 4\pi D\rho \int_0^\infty B_\nu(T_d)\kappa_\nu \, \mathrm{d}\nu \tag{2.13}$$

where B_ν is the Planck function at dust temperature T_d. At equilibrium, the dust temperature will be the temperature that ensures that $\Gamma_{d,0} = \Lambda_d$, i.e.:

$$\int_0^\infty B_\nu(T_d)\kappa_\nu \, \mathrm{d}\nu = \int_0^\infty J_\nu \kappa_\nu \, \mathrm{d}\nu \tag{2.14}$$

The calculations were performed using iterative numerical methods within Python 2 code ***dust_temp.py*** (Appendix III), assuming the Mathis et al. (1983) ambient ISRF and adopting the mean dust opacities given for non-coagulated and thick ice-mantle grains by Ossenkopf and Henning (1994) for wavelengths ≥ 1μm and the Mathis et al. (1983) opacities for wavelengths < 1μm. A cubic spline fit is made to both the ISRF and opacity available data points to facilitate calculation within the Python code, as shown in Figures 2.3 and 2.4. Note that

the ISRF data is presented in units of $4\pi\lambda J_\lambda$ to allow more straightforward comparison between the various wavelength ranges and to allow direct comparison with Figure 1.1.

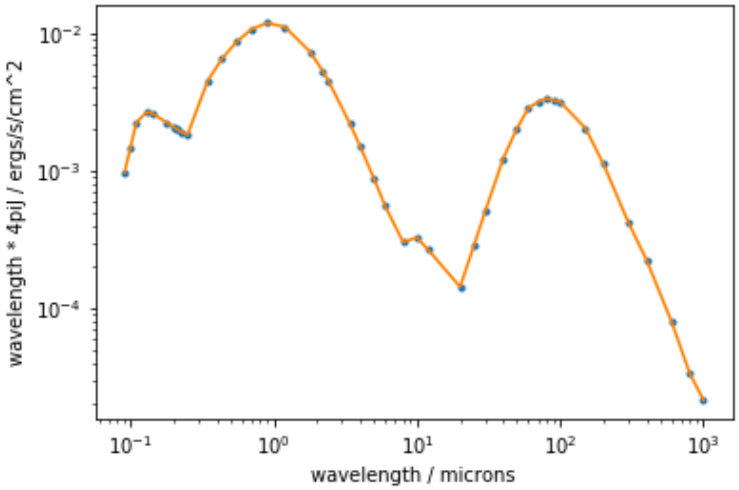


Figure 2.3. Python cubic spline-fit to the ISRF data of Mathis et al. (1983)

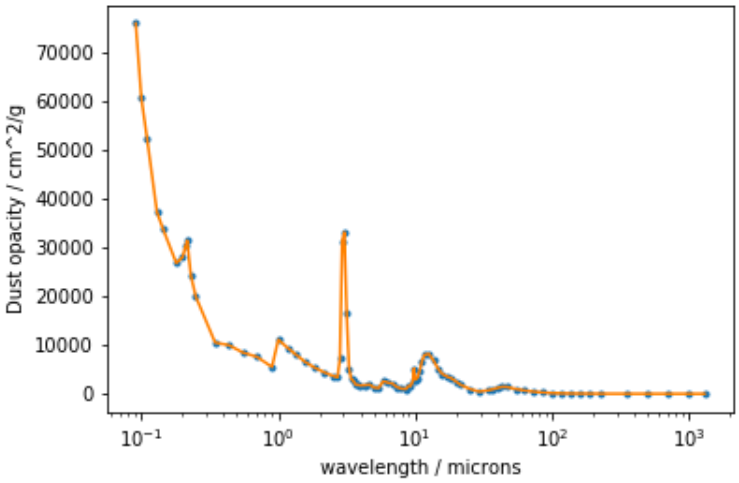


Figure 2.4. Python cubic spline-fit to the dust opacity data of Ossenkopf and Henning (1994)

The calculations were performed firstly assuming just the ambient Mathis et al. (1983) ISRF, yielding an expected dust temperature of 15 K. They are then repeated assuming a 5 $M_\odot$ accreting protostar at a range of distances from the dust position, firstly assuming the ambient ISRF is increased across its frequency range by a single factor equal to the incremental effect on G_0 of the protostar (as in the current AREPO model) and secondly, and intuitively more realistically, by the superposition the black body spectrum from the accreting protostar onto the ambient ISRF. The impact of the latter calculation is discussed further in section 4.

3 RESULTS & DISCUSSION

3.1 G_0 IMPACT OF ACCRETING PROTOSTARS

The analytically calculated impact on G_0 of an accreting protostar in an optically thin environment (i.e. unattenuated radiation) as a function of protostellar mass and distance from the source is illustrated in Figure 3.1 using the algorithm described in section 2.3.

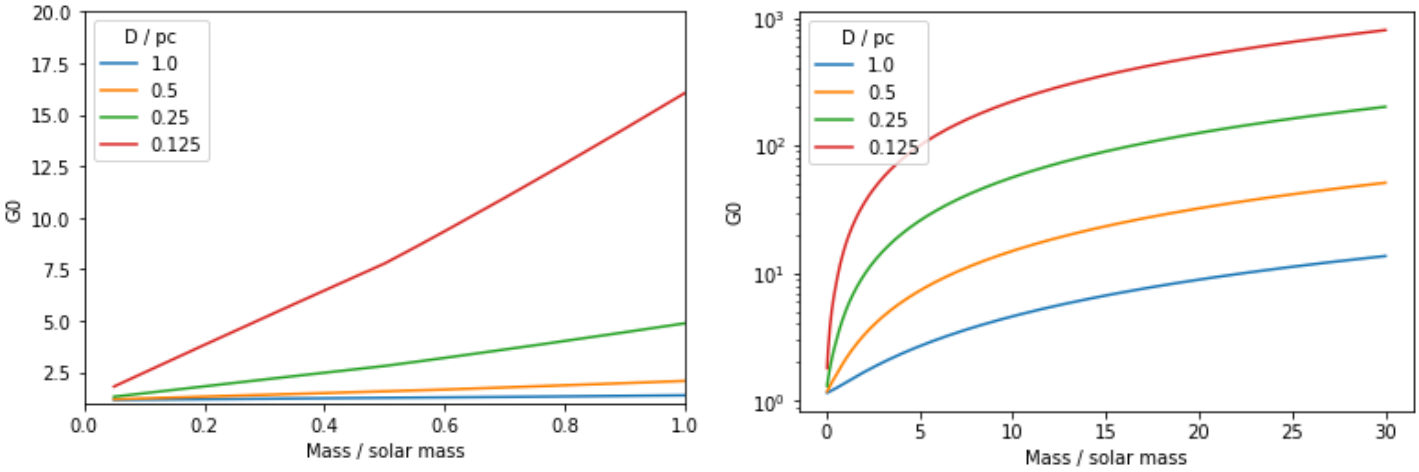


Figure 3.1. Analytically calculated total G_0 (including an ambient G_0 of 1.15) as a result of feedback from an accreting protostar at various distances in an optically thin environment. The right-hand panel is for masses < 1 $M_\odot$ and the left-hand panel for masses < 30 $M_\odot$. Note the change to log scale for G_0 for the higher mass diagram.

Bearing in mind that these calculations assume no attenuation of radiation between the source and the region of interest, Figure 3.1 illustrates that a significant impact from protostellar feedback (G_0 > 10 in an optically thin environment) is only expected where the protostar is either in close proximity (within ~0.1 pc) or has high mass (> ~5 $M_\odot$). However, even for very high mass stars, they probably need to be within 0.5 pc to be of significance in an optically thick environment.

3.2 FEEDBACK CODE TESTING

3.2.1 UNATTENUATED DUST TEMPERATURE

As discussed in section 2.5, the fundamental test of the new code is that the predicted dust temperatures surrounding the simple case of a sink with a fixed mass of 5 $M_\odot$ (and hence a sink with a distinct accretion rate, radius and luminosity) in an optically thin environment (i.e. with the TreeCol algorithm inactive) can be reproduced from first principles. The results of this test are shown in Figure 3.2, which indicates that the code can indeed be relied upon. Any small discrepancies between the numerical output and the analytical calculation are explained by the fact the dust heating and cooling mechanisms in AREPO are dependent on more sophisticated chemistry and micro-physics, such as dust-gas interaction and interactions with cosmic rays, and such effects will have a secondary impact on dust temperature. Note also that at large distances away from the sink, the dust temperature is seen to equilibrate to ~15 K as predicted by the analytical calculations.

3.2.2 ATTENUATED DUST TEMPERATURE

The impact of the inclusion of the TreeCol radiative transfer scheme using the same single sink model as above is illustrated in Figure 3.3 (isotropic radiation field) and Figure 3.4 (anisotropic radiation field). The dust temperature is now a function not only of the prevailing ISRF, as augmented by the sink, but also of the column density between the edge of the simulation and the cell under consideration. Since the dust temperatures in all cells now falls below the optically thin analytical line, Figure 3.3 indicates that no cells are now seeing the unattenuated impact of radiative feedback from the sink/protostar with many cells exhibiting significantly cooler temperatures due to the more complex cooling mechanisms in the AREPO model. Nevertheless, the impact of radiative feedback on dust temperature in the isotropic model is still significant, particularly close to the source.

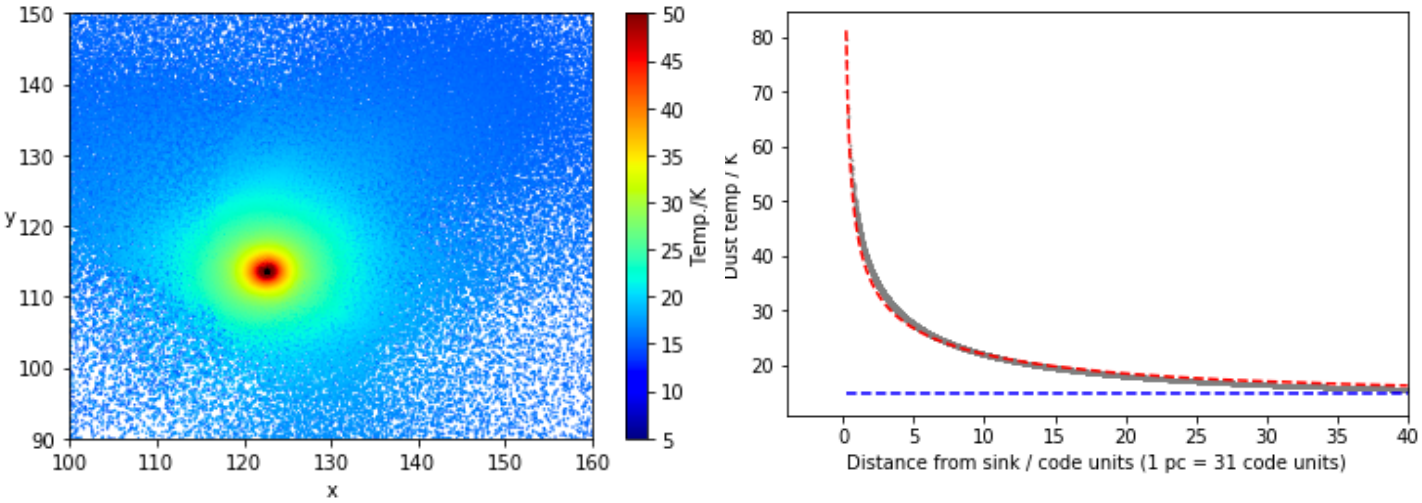


Figure 3.2. AREPO predicted dust temperature in the region of a single 5 $M_{\odot}$ sink particle with unattenuated isotropic feedback. In the right-hand plot, the AREPO predicted temperatures are shown in grey, the red dashed line is the analytically predicted dust temperature with unattenuated radiative feedback and the blue dashed line is the analytically predicted dust temperature without radiative feedback (unattenuated Mathis ISRF only).

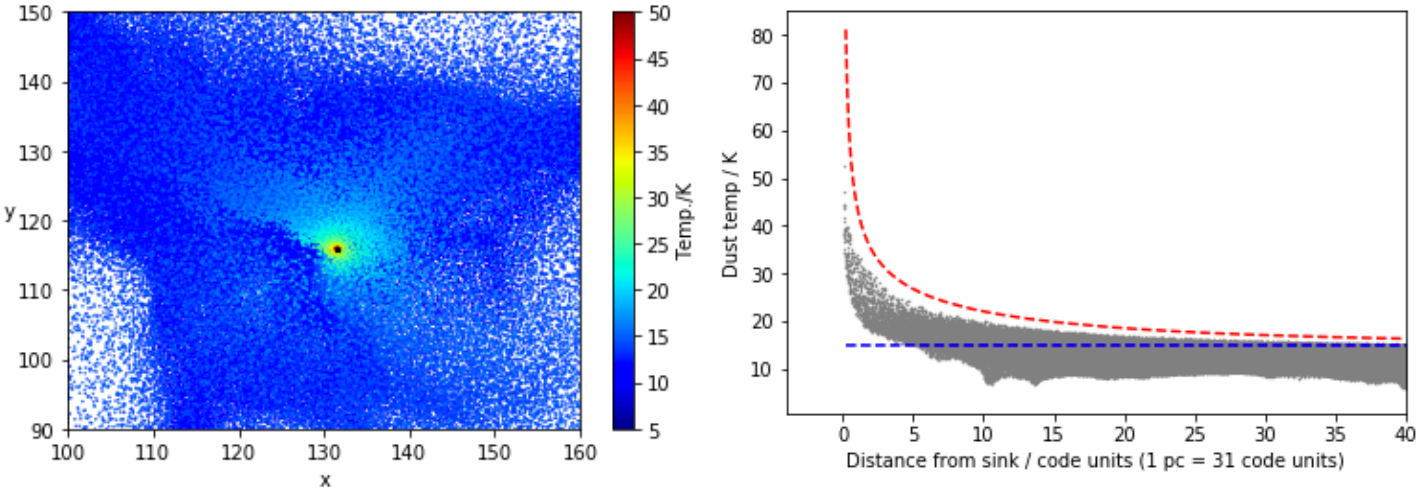


Figure 3.3. AREPO predicted dust temperature in the region of a single 5 $M_{\odot}$ sink particle with isotropic feedback attenuated by TreeCol. Colour scheme is as in Figure 3.2.

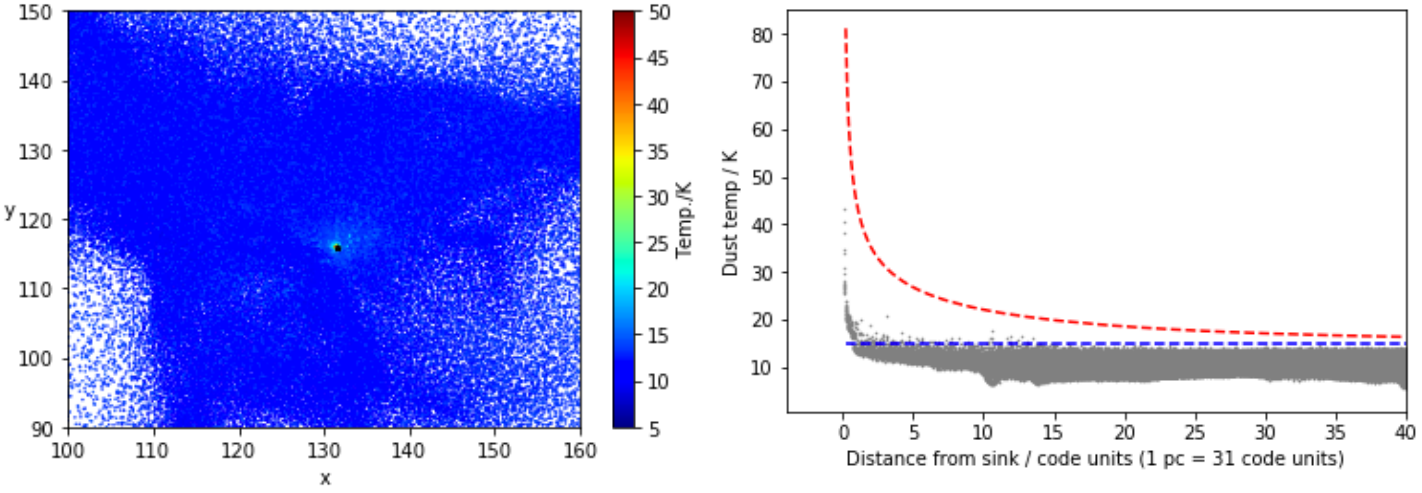


Figure 3.4. AREPO predicted dust temperature in the region of a single 5 $M_{\odot}$ sink particle with an isotropic radiation field but with the attenuation weighted to the column density in the direction of the source ('anisotropic' feedback). Colour scheme is as in Figure 3.2

Once the directional nature of the incremental ISRF is taken into account in the anisotropic model (Figure 3.4), the impact of radiative feedback from the protostar/sink is seen to be significantly reduced with only those cells within a distance of a few code units (~0.1 pc or 10^4 AU) seeing any clear effect. This makes sense, since sink particles will, by definition, form in regions of higher density and it would therefore be expected that if the calculation of the weighted mean column density is heavily weighted towards this region, the resultant column density will be significantly higher than if an equally weighted mean column density is calculated, as previously illustrated in Figure 2.2.

3.3 Preliminary Science Results

Three full simulations have been run with the model parameters described in section 2.2; a base case model with no feedback included, an isotropic feedback model including the new algorithm as described in section 2.3 and an anisotropic feedback model as described in section 2.4. The base case model was only able to run until a time of 12.4 code units (≈ 1.1 Myrs) before it was terminated due to exceeding the iteration limit for Voronoi tessellation. The first sinks form, in all cases, at a time of 8.9 code units, hence the longest life of any sink in the model at termination is around 300,000 yrs. Although this seems very short, given that this project is only exploring the impact of accretion luminosity from the adiabatic accretion phase of protostellar evolution, this time period corresponds to a final mass of ~3 $M_\odot$ for a mean accretion rate of 10^{-5} $M_\odot$/yr and so is not a completely pointless timescale to consider in a first pass analysis of the impact of the majority of embedded protostars.

3.3.1 Density

The impact on density of the inclusion of the feedback code is subtle over such a short timescale. Consequently the results are only shown (Figure 3.5) for the final anisotropic feedback case to allow an overview of how the model has evolved into clumps and filaments of overdensity - where protostars form - by the time it is terminated.

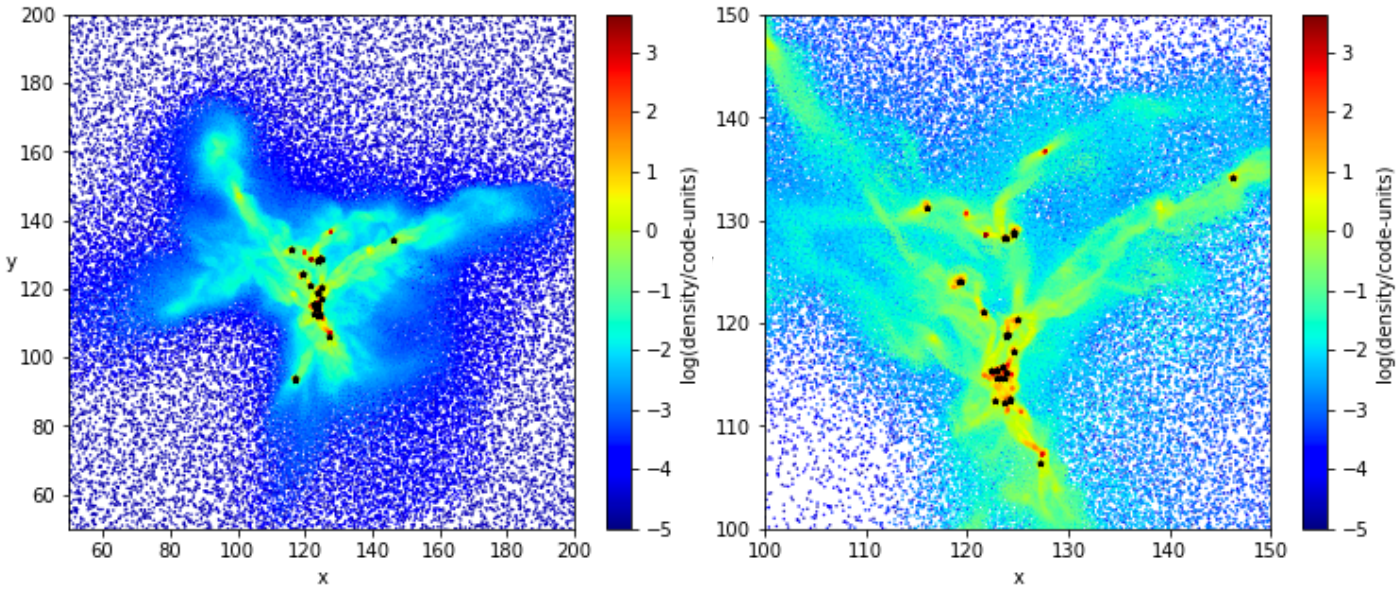


Figure 3.5. AREPO predicted maximum density along the z axis after ~1.1 Myrs for the anisotropic feedback case at two levels of zoom. The black dots mark the positions where sink particles/protostars have formed.

3.3.2 Temperature

The impact on dust temperature is more marked (Figure 3.6) with no heating in the base case, significant heating of a wide area in the isotropic case and, as intuitively expected, much more localised heating around regions of protostar formation in the anisotropic case.

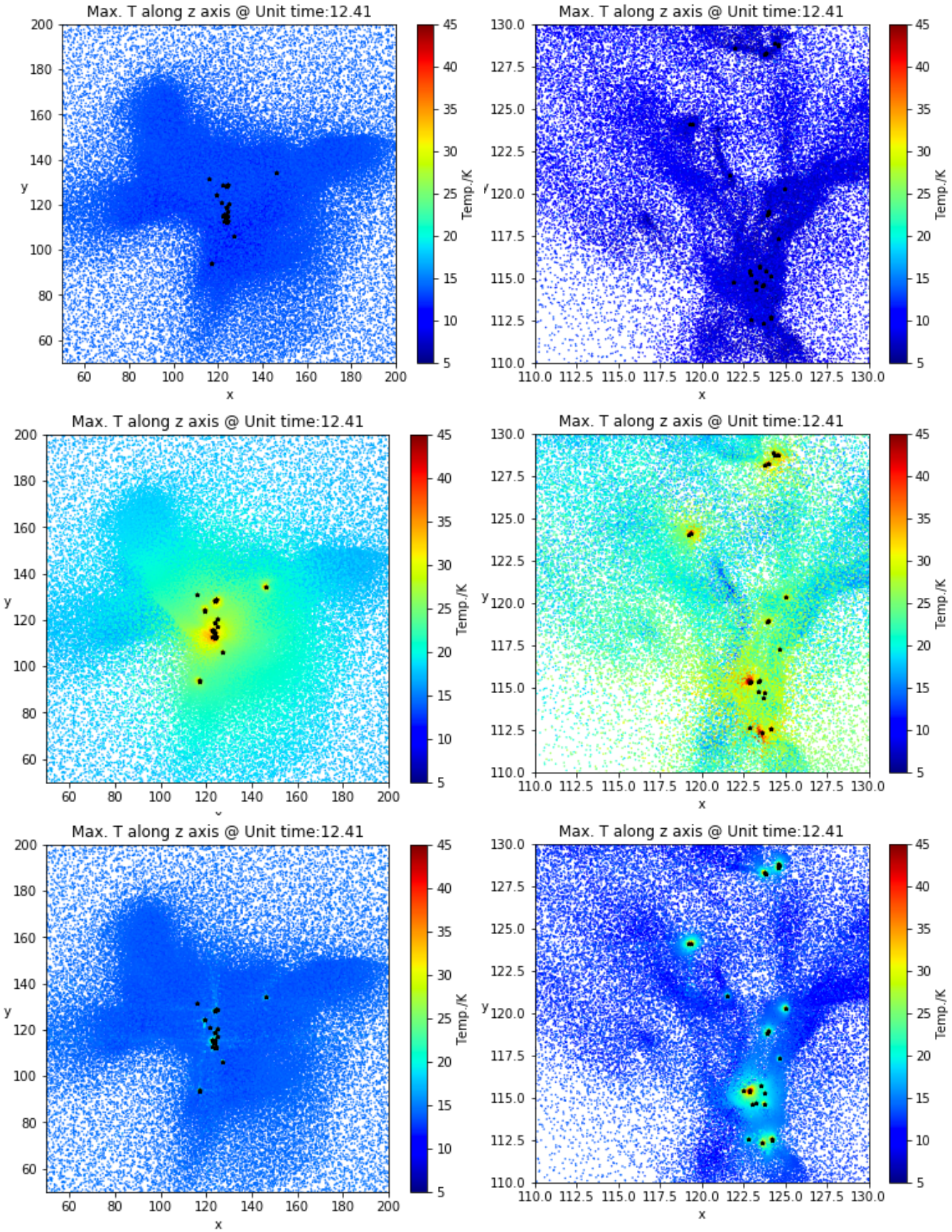


Figure 3.6. AREPO predicted maximum temperature along the z axis after ~1.1 Myrs for the base case (top row), isotropic feedback case (middle row) and anisotropic feedback case (bottom row) at two levels of zoom. The black dots mark the positions where sink particles/protostars have formed.

3.3.3 CHEMISTRY

This increase in dust temperature has some knock-on effect to the chemistry as illustrated in Figure 3.7 which shows the 100 bin probability density function of H2 concentration and CO concentration (relative to the total number of hydrogen nuclei) for all cells in the zoom-in region in Figure 3.6 (x, y = 110-130). There is no clearly discernible effect on the distribution of H2 concentration, the mean value of which is unaltered to 3 significant figures at 0.466. However, the proportion of cells with lower CO concentrations is clearly higher in the anisotropic feedback case, suggesting a general reduction of CO concentration overall which is confirmed by a reduction in mean value across all the cells in the region from 1.30×10^{-4} in the base case to 1.22×10^{-4} when the feedback model is effective. Since CO is a main coolant species, this implies that feedback is not only raising temperatures directly through heating but also by suppressing coolant species.

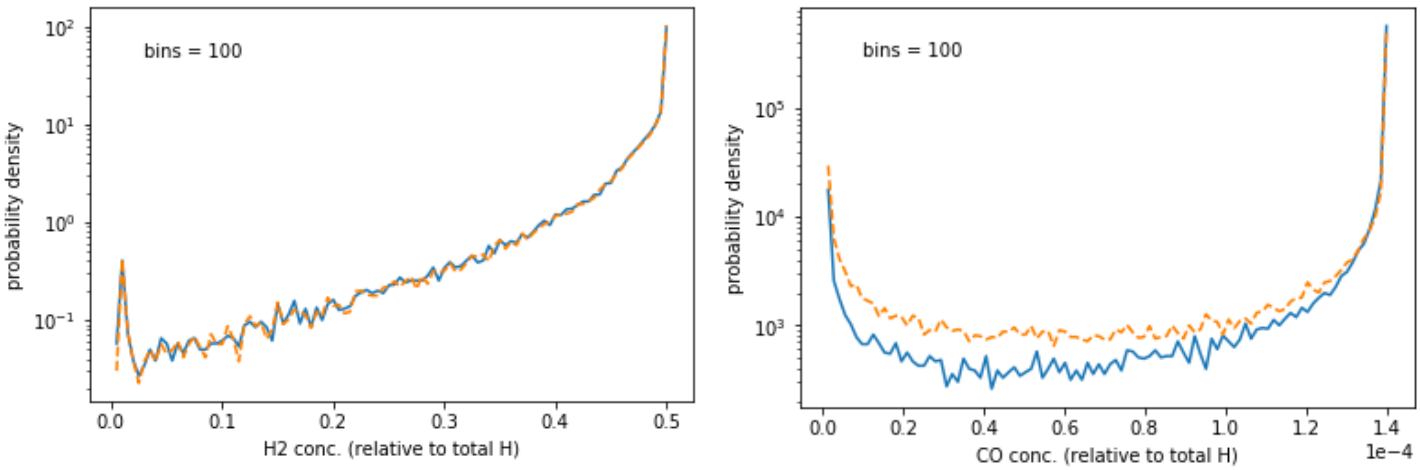


Figure 3.7. AREPO predicted 100 bin probability density function of H2 concentration and CO concentration (relative to the total number of hydrogen nuclei) for all cells in the zoom-in region in Figure 3.6 (x, y = 110-130). The solid blue line is base case data and the dashed orange line the anisotropic feedback case. The mean H2 concentration remains at 0.466 in both cases; the CO concentration drops from a mean value of 1.30×10^{-4} to 1.22×10^{-4} when the feedback model is effective

3.3.4 MASS FUNCTION

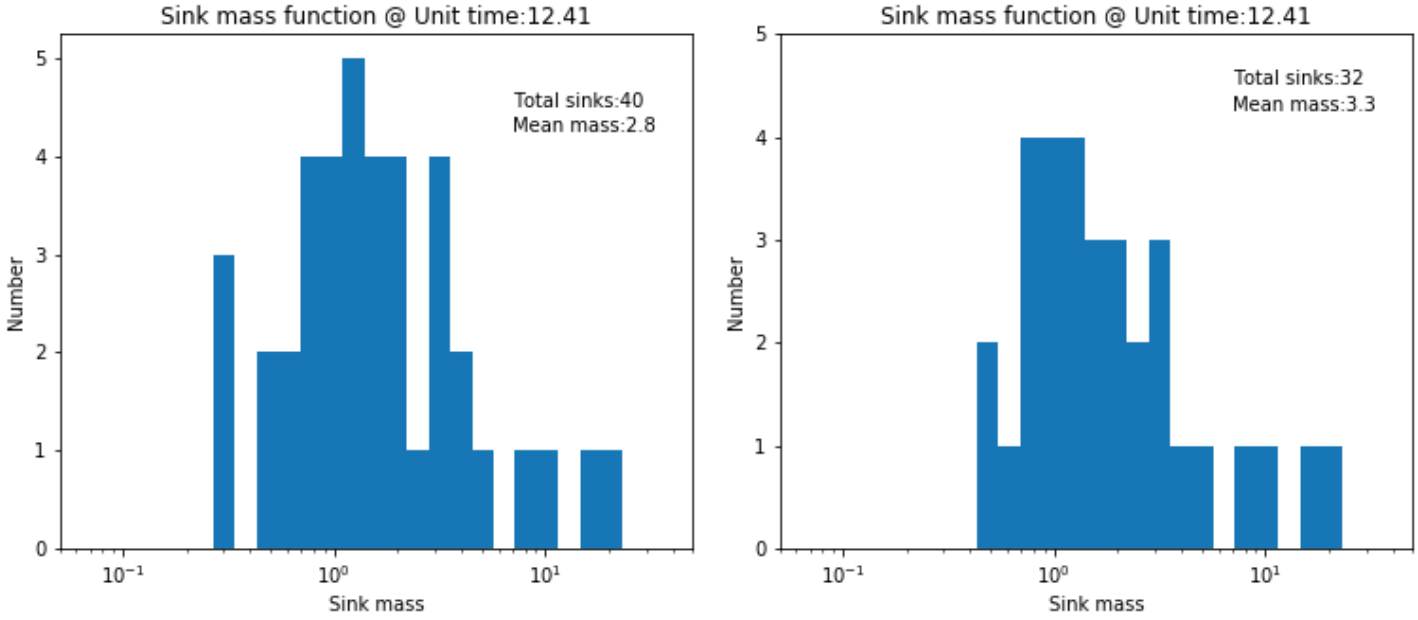


Figure 3.8. AREPO predicted sink mass function for the base case (left) and the anisotropic feedback case (right)

The impact of the (anisotropic) feedback model on sink mass function is illustrated in Figure 3.8, which shows both a reduction in the number of sinks/protostars created and an increase in the mean mass when comparing the feedback model to the base case model. Notwithstanding the fact that the sample size is statistically small,

both of these effects would be expected from increased temperature since the Jeans mass (the minimum mass that will theoretically collapse given a spherical cloud with a certain density and temperature) can be shown to be proportional to $T^{3/2}$ for a given density. Consequently, the Jean's mass increases at higher temperatures and we would thus expect less stars to form in general. Furthermore, the degree of fragmentation should also decrease so that those stars that do form will on average have a higher mass.

4 CONCLUSIONS

4.1 SUMMARY

This project has shown that it is possible to incorporate algorithms into the existing AREPO coding architecture that will effectively model 'internal' radiative feedback from accreting protostars embedded within a molecular cloud. The algorithm developed and implemented during this project should allow the effect of radiative feedback on a star-forming molecular cloud to be studied where multiple feedback sources, currently in the form of spherically accreting protostars, are generated in-situ as the cloud evolves. The veracity of the algorithm output has been demonstrated by comparing simulation predictions of dust temperature for some simple models with 'analytical' calculations of the same parameter.

The project has also determined that, using the TreeCol radiative transfer algorithm, there is a potentially significant increase in accuracy to be gained by considering the directional nature of the feedback in an 'anisotropic' model on account of the fact that the protostars will almost certainly be formed within regions of enhanced column density. Due to the high degree of attenuation to which this leads, significant alteration of radiation energy density from protostellar feedback, as described by the relative energy density parameter G_0, is only expected where the protostar is either in close proximity (within ~0.1 pc) and/or has high mass (> ~5 $M_\odot$) and even the highest mass protostars probably need to be within 0.5 pc to be of any significance in the environment in which protostars evolve.

Although limited in run time, preliminary science investigations indicate that, on account of generally increased temperature and associated changes in chemistry, at a general level the model is likely to confirm previous findings that radiative feedback reduces star formation efficiency.

4.2 CRITICAL PROJECT REVIEW

The primary objective of the project was to answer the research question: can a model of internal radiative feedback from embedded accreting protostars be successfully incorporated within the existing coding architecture of AREPO? This has been clearly answered in the affirmative. Furthermore, the secondary objective to conduct preliminary science investigations into the influence of radiative feedback in a turbulent molecular cloud model has also been met to the extent allowed by the available time.

Having acknowledged this success, it is acknowledged that this project only represents the first phase of development of the model with a number of areas for improvement and refinement identified as the project has progressed. These are specifically identified in section 4.3. Once these are addressed, it is anticipated that the internal feedback code within AREPO will become an important tool in furthering our understanding of the complex, interwoven mechanisms which lead to the population of stars we observe in nature.

4.3 SUGGESTED FUTURE WORK

4.3.1 THE NATURE OF THE SPECIFIC INTENSITY SPECTRUM

The current scheme within AREPO enhances the specific intensity spectrum by calculating the ratio of total G_0 to ambient G_0 (1.15 for the Mathis et al. ISRF) then increasing the intensity of the entire ambient spectrum by this factor across the frequency range. Whilst this may be an acceptable approach in relation to determining the magnitude of the photoelectric effect, which is the main driver of gas temperature, it is potentially inaccurate for the determination of dust temperature, a key driver of chemical state, which is determined by total incident energy regardless of its frequency range. This is illustrated in Figure 4.1 (calculated as described in section 2.5), which suggests that there may be significantly less heating due to longer wavelength radiation than currently modeled. It is therefore recommended that a method to better model the augmented radiation field across the spectrum be investigated.

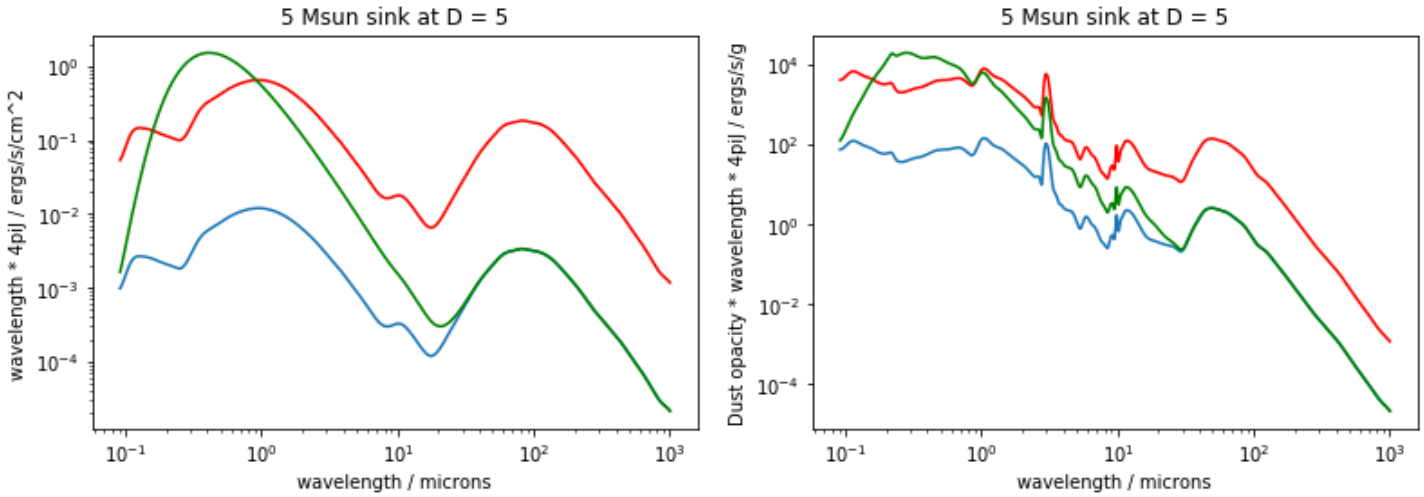


Figure 4.1. A comparison of the effect (for a 5 $M_{\odot}$ sink at ~0.16 pc) on the specific intensity spectrum of representing an ambient ISRF (blue) augmented by radiative feedback as a multiple of the ambient ISRF (red) with a more rigorous superposition of the black-body spectrum of the feedback source with the ambient ISRF (green). The right-hand panel is the same data multiplied by specific opacity for an indication of the net effect on dust heating.

4.3.2 Radiative Transfer

The existing TreeCol algorithm within AREPO assigns column densities to its HEALPix pixels based on cell volume densities within a particular pixel out to the edge of the simulation. In the anisotropic model described above, this entire column density is then used to provide the intensity weighting to determine the weighted average attenuation factor. Consequently, the impact of a sink particle, which will very rarely be located at the edge of the simulation will tend to be over-attenuated. This situation could be alleviated by 'walking the tree' to determine specific column densities for sinks taking into account only the volume densities out to the location of the sink.

4.3.3 Accretion Rate & Associated Luminosity

There are potentially a number of refinements that can be made to improve the determination of accretion rate and the associated luminosity.

a. The accretion rate in the current algorithm is calculated using an approximation function from work by Maschberger et al. (2014). However the AREPO model itself calculates is already calculating accretion rates onto sinks in order to determine their mass at any given time-step. Extracting this information and replacing the current approximation is suggested.

b. The effective photospheric radius of the accreting protostar is currently assumed to be the same as the accretion shock. This is almost certainly not the case as the innermost part of the accreting envelope nearest the accretion shock, termed the radiative precursor, is probably optically thick (Hosokawa and Omukai, 2009). Furthermore, a protostar will be surrounded by a dust cocoon which may serve to more or less completely reprocess UV and optical band light into the infrared. These features are illustrated schematically in Figure 4.2, reproduced from Hosokawa and Omukai (2009). Further consideration of the position of the photospheric radius in relation to the accretion shock and how the presence of a dust cocoon can be modeled is suggested.

c. The model currently assumes that accretion occurs in a spherically symmetric manner, yet conservation of angular momentum in collisional accreting systems is known to lead to the formation of accretion discs, and potentially bipolar outflows, which are currently not taken into consideration in the model. In particular, the temperature profile and associated emission spectrum of an accretion disc is well known (e.g. Ward-Thompson and Whitworth, 2011) and could be incorporated into the accretion luminosity model as an additional refinement.

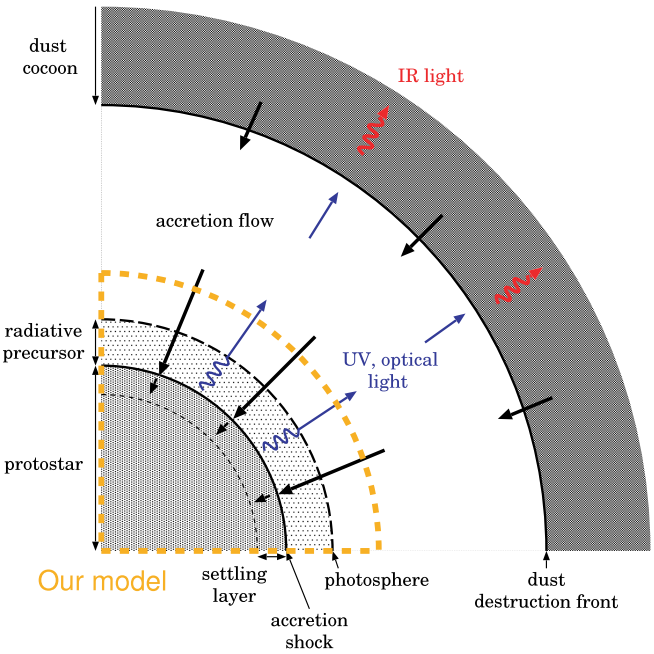


Figure 4.2. Reproduced from Hosokawa and Omukai (2009). Schematic diagram of an accreting protostar.

4.3.4 Protostellar Evolution & Associated Luminosity

As highlighted in section 1.2.3, the assumption that accretion luminosity dominates any other form of luminosity is only valid during the adiabatic accretion phase of protostellar evolution. Model realism would be significantly enhanced by tracking the evolution of each sink / protostar in a manner similar to that described by Offner et al. (2009). It follows that Kelvin-Helmholtz contraction luminosity and any relevant nuclear burning luminosity (including pre main-sequence deuterium burning), which become more important during later evolutionary phases, should then also be calculated and added to the accretion luminosity.

4.3.5 Other Considerations

Obviously, a more comprehensive science review of the effect of embedded protostars on molecular cloud evolution will require significantly longer run times than so far achieved. This will require investigation into the reasons for the current termination after a time of 12.4 code units and a solution thereto.

Finally, as other areas of AREPO are improved these should be incorporated where relevant. For instance, an improved sink particle model has been developed which will benefit this area of research once incorporated.

ACKNOWLEDGEMENTS

I would like to thank Dr. Paul Clark, the supervisor of this project, for his teaching, encouragement and patience, always delivered with humanity and good humour. I would also like to thank my MSc project mentor, Dr. Richard Lewis, for patient counsel on general matters surrounding project execution and the joys and pitfalls of research in general.

REFERENCES

Bate, M.R., 2014. The statistical properties of stars and their dependence on metallicity: the effects of opacity. Mon. Not. R. Astron. Soc. 442, 285–313. doi:10.1093/mnras/stu795

Bate, M.R., 2009. The importance of radiative feedback for the stellar initial mass function. Mon. Not. R. Astron. Soc. 392, 1363–1380. doi:10.1111/j.1365-2966.2008.14165.x

Bertram, E., Glover, S.C.O., Clark, P.C., Ragan, S.E., Klessen, R.S., 2016. Synthetic observations of molecular clouds in a galactic centre environment - I. Studying maps of column density and integrated intensity. Mon. Not. R. Astron. Soc. 455, 3763–3778. doi:10.1093/mnras/stv2619

Black, J.H., 1994. Energy Budgets of Diffuse Clouds. Presented at the The First Symposium on the Infrared Cirrus and Diffuse Interstellar Clouds, p. 355.

Bron, E., Le Bourlot, J., Le Petit, F., 2014. Surface chemistry in the interstellar medium. II. H2 formation on dust with random temperature fluctuations. Astron. Astrophys. 569, A100. doi:10.1051/0004-6361/201322101

Clark, P.C., Glover, S.C.O., 2015. Does the CO-to-H2 conversion factor depend on the star formation rate? Mon. Not. R. Astron. Soc. 452, 2057–2070. doi:10.1093/mnras/stv1369

Clark, P.C., Glover, S.C.O., 2014. On column density thresholds and the star formation rate. Mon. Not. R. Astron. Soc. 444, 2396–2414. doi:10.1093/mnras/stu1589

Clark, P.C., Glover, S.C.O., Klessen, R.S., 2012. TreeCol: a novel approach to estimating column densities in astrophysical simulations. Mon. Not. R. Astron. Soc. 420, 745–756. doi:10.1111/j.1365-2966.2011.20087.x

Clarke, C., Carswell, B., 2007. Principles of Astrophysical Fluid Dynamics, 1 edition. ed. Cambridge University Press, Cambridge ; New York.

Dale, J.E., 2015. The modelling of feedback in star formation simulations. New Astron. Rev. 68, 1–33. doi:10.1016/j.newar.2015.06.001

Dale, J.E., Ercolano, B., Bonnell, I.A., 2013. Ionizing feedback from massive stars in massive clusters - III. Disruption of partially unbound clouds. Mon. Not. R. Astron. Soc. 430, 234–246. doi:10.1093/mnras/sts592

Draine, B.T., 2011. Physics of the Interstellar and Intergalactic Medium. Princeton University Press, Princeton, N.J.

Draine, B.T., 1978. Photoelectric heating of interstellar gas. Astrophys. J. Suppl. Ser. 36, 595–619. doi:10.1086/190513

Fryxell, B., Olson, K., Ricker, P., Timmes, F.X., Zingale, M., Lamb, D.Q., MacNeice, P., Rosner, R., Truran, J.W., Tufo, H., 2000. FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. Astrophys. J. Suppl. Ser. 131, 273–334. doi:10.1086/317361

Gatto, A., Walch, S., Naab, T., Girichidis, P., Wünsch, R., Glover, S.C.O., Klessen, R.S., Clark, P.C., Peters, T., Derigs, D., Baczynski, C., Puls, J., 2017. The SILCC project - III. Regulation of star formation and outflows by stellar winds and supernovae. Mon. Not. R. Astron. Soc. 466, 1903–1924. doi:10.1093/mnras/stw3209

Girichidis, P., Konstandin, L., Whitworth, A.P., Klessen, R.S., 2014. On the Evolution of the Density Probability Density Function in Strongly Self-gravitating Systems. Astrophys. J. 781, 91. doi:10.1088/0004-637X/781/2/91

Glover, S.C.O., Clark, P.C., 2012. Is molecular gas necessary for star formation? Mon. Not. R. Astron. Soc. 421, 9–19. doi:10.1111/j.1365-2966.2011.19648.x

Goldsmith, P.F., 2001. Molecular Depletion and Thermal Balance in Dark Cloud Cores. Astrophys. J. 557, 736–746. doi:10.1086/322255

Górski, K.M., Hivon, E., Banday, A.J., Wandelt, B.D., Hansen, F.K., Reinecke, M., Bartelmann, M., 2005. HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. Astrophys. J. 622, 759–771. doi:10.1086/427976

Habing, H.J., 1968. The interstellar radiation density between 912 A and 2400 A. Bull. Astron. Inst. Neth. 19, 421.

Hartwig, T., Clark, P.C., Glover, S.C.O., Klessen, R.S., Sasaki, M., 2015. A New Approach to Determine Optically Thick H2 Cooling and its Effect on Primordial Star Formation. Astrophys. J. 799, 114. doi:10.1088/0004-637X/799/2/114

healpix.sourceforge.net [WWW Document], 2017. . Heal. - Featur. URL http://healpix.sourceforge.net/ (accessed 8.25.17).

Hollenbach, D., Salpeter, E.E., 1971. Surface Recombination of Hydrogen Molecules. Astrophys. J. 163, 155. doi:10.1086/150754

Hosokawa, T., Omukai, K., 2009. Evolution of Massive Protostars with High Accretion Rates. Astrophys. J. 691, 823–846. doi:10.1088/0004-637X/691/1/823

Howard, C.S., Pudritz, R.E., Harris, W.E., 2017. Simulating radiative feedback and star cluster formation in GMCs - II. Mass dependence of cloud destruction and cluster properties. Mon. Not. R. Astron. Soc. 470, 3346–3358. doi:10.1093/mnras/stx1363

Hubber, D.A., Batty, C.P., McLeod, A., Whitworth, A.P., 2011. SEREN - a new SPH code for star and planet formation simulations. Algorithms and tests. Astron. Astrophys. 529, A27. doi:10.1051/0004-6361/201014949

Krumholz, M.R., Cunningham, A.J., Klein, R.I., McKee, C.F., 2010. Radiation Feedback, Fragmentation, and the Environmental Dependence of the Initial Mass Function. Astrophys. J. 713, 1120–1133. doi:10.1088/0004-637X/713/2/1120

Krumholz, M.R., Klein, R.I., McKee, C.F., 2012. Radiation-hydrodynamic Simulations of the Formation of Orion-like Star Clusters. II. The Initial Mass Function from Winds, Turbulence, and Radiation. Astrophys. J. 754, 71. doi:10.1088/0004-637X/754/1/71

Krumholz, M.R., Klein, R.I., McKee, C.F., 2011. Radiation-hydrodynamic Simulations of the Formation of Orion-like Star Clusters. I. Implications for the Origin of the Initial Mass Function. Astrophys. J. 740, 74. doi:10.1088/0004-637X/740/2/74

Krumholz, M.R., McKee, C.F., 2008. A minimum column density of 1gcm-2 for massive star formation. Nature 451, 1082–1084. doi:10.1038/nature06620
Lomax, O., Whitworth, A.P., Hubber, D.A., Stamatellos, D., Walch, S., 2015. Simulations of star formation in Ophiuchus - II. Multiplicity. Mon. Not. R. Astron. Soc. 447, 1550–1564. doi:10.1093/mnras/stu2530
Lomax, O., Whitworth, A.P., Hubber, D.A., Stamatellos, D., Walch, S., 2014. Simulating star formation in Ophiuchus. Mon. Not. R. Astron. Soc. 439, 3039–3050. doi:10.1093/mnras/stu177
Maschberger, T., Bonnell, I.A., Clarke, C.J., Moraux, E., 2014. The relation between accretion rates and the initial mass function in hydrodynamical simulations of star formation. Mon. Not. R. Astron. Soc. 439, 234–246. doi:10.1093/mnras/stt2403
Masunaga, H., Inutsuka, S., 2000. A Radiation Hydrodynamic Model for Protostellar Collapse. II. The Second Collapse and the Birth of a Protostar. Astrophys. J. 531, 350–365. doi:10.1086/308439
Masunaga, H., Miyama, S.M., Inutsuka, S., 1998. A Radiation Hydrodynamic Model for Protostellar Collapse. I. The First Collapse. Astrophys. J. 495, 346–369. doi:10.1086/305281
Mathis, J.S., Mezger, P.G., Panagia, N., 1983. Interstellar radiation field and dust temperatures in the diffuse interstellar matter and in giant molecular clouds. Astron. Astrophys. 128, 212–229.
Mezger, P.G., Mathis, J.S., Panagia, N., 1982. The origin of the diffuse galactic far infrared and sub-millimeter emission. Astron. Astrophys. 105, 372–388.
Murray, N., Quataert, E., Thompson, T.A., 2010. The Disruption of Giant Molecular Clouds by Radiation Pressure & the Efficiency of Star Formation in Galaxies. Astrophys. J. 709, 191–209. doi:10.1088/0004-637X/709/1/191
Offner, S.S.R., Klein, R.I., McKee, C.F., Krumholz, M.R., 2009. The Effects of Radiative Transfer on Low-Mass Star Formation. Astrophys. J. 703, 131–149. doi:10.1088/0004-637X/703/1/131
Ossenkopf, V., Henning, T., 1994. Dust opacities for protostellar cores. Astron. Astrophys. 291, 943–959.
Peters, T., Zhukovska, S., Naab, T., Girichidis, P., Walch, S., Glover, S.C.O., Klessen, R.S., Clark, P.C., Seifried, D., 2017. The turbulent life of dust grains in the supernova-driven, multiphase interstellar medium. Mon. Not. R. Astron. Soc. 467, 4322–4342. doi:10.1093/mnras/stx341
Prialnik, D., 2009. By Dina Prialnik An Introduction to the Theory of Stellar Structure and Evolution, 2nd Edition edition. ed. Cambridge University Press.
Ryan, S.G., Norton, A.J., 2010. Stellar Evolution and Nucleosynthesis, 1 edition. ed. Cambridge University Press, Cambridge ; New York.
Sales, L.V., Marinacci, F., Springel, V., Petkova, M., 2014. Stellar feedback by radiation pressure and photoionization. Mon. Not. R. Astron. Soc. 439, 2990–3006. doi:10.1093/mnras/stu155
Smith, R.J., Glover, S.C.O., Clark, P.C., Klessen, R.S., Springel, V., 2014. CO-dark gas and molecular filaments in Milky Way-type galaxies. Mon. Not. R. Astron. Soc. 441, 1628–1645. doi:10.1093/mnras/stu616
Soifer, B.T., Sanders, D.B., Madore, B.F., Neugebauer, G., Danielson, G.E., Elias, J.H., Lonsdale, C.J., Rice, W.L., 1987. The IRAS bright galaxy sample. II - The sample and luminosity function. Astrophys. J. 320, 238–257. doi:10.1086/165536
Springel, V., 2010. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. Mon. Not. R. Astron. Soc. 401, 791–851. doi:10.1111/j.1365-2966.2009.15715.x
Stahler, S.W., Palla, F., Salpeter, E.E., 1986. Primordial stellar evolution - The protostar phase. Astrophys. J. 302, 590–605. doi:10.1086/164018
Ward-Thompson, D., Whitworth, A.P., 2011. An Introduction to Star Formation. Cambridge University Press, Cambridge.
www.illustris-project.org [WWW Document], 2017. . AREPO — Arepo 4125473 Mon May 22 194902 2017 Doc. URL http://www.illustris-project.org/w/arepo_docs/ (accessed 8.25.17).

APPENDICES

APPENDIX I: RISK ASSESSMENT FORM

IMPORTANT: before carrying out the assessment, read the Guidance Notes

Department	None	Building	Home	Room No	Study
Name of Assessor	A. M. Bostock	Date of Original Assessment	12 June 2017	Assessment No	1

Status of Assessor: Supervisor ☐ Postgraduate ☑, Undergraduate ☐ Technician ☐ Other: ..... (Specify)

Brief Description of Procedure/Activity including its Location and Duration

Undertaking a theoretical MSc Astrophysics project sitting at a desk in a domestic house and either reading, writing or using a PC or laptop.

Persons at Risk	Are they...	Notes:
Staff ☐	Trained ☐	Only the assessor is involved in this project at this location. The assessor has previous experience of undertaking risk assessments from previous experience in industry
Students ☑	Competent ☑	
Visitors ☐	Inexperienced ☐	
Contractor ☐	Disabled ☐	

Level of Supervision	Notes
None ☑ Constan☐ Periodi☐ Training Required ☐	Not applicable

Will Protective Equipment Be Used? Please give specific details of PPE

Head ☐	Eye ☐	Ear ☐	Not applicable
Body ☐	Hand ☐	Foot ☐	

Is the Environment at Risk?	Notes
Yes ☐ No ☑	

Will Waste be generated?	If 'Yes' please give details of disposal
Yes ☑ No ☐	Waste paper will be shredded and recycled via the usual council doorstep service

COSHH Assessment

Substance (name or formula)	Quantity (g or cm^3)	Exposure Time	Hazard	Likelihood (0 to 5)	Severity (0 to 5)	Disposal Method
None relevant						

Other Hazards involved in the Procedure/Activity

Hazard and how it can cause harm.	Control Measures in place and Consequence of Failure	Level of Risk (Lik x Sev=Risk)
Fatigue, eye strain, upper limb problems and backache from overuse or improper use of display screen equipment ('DSE')	User will take regular breaks and has ensured workstation is set up properly in accordance with HSE leaflet INDG36(rev4), published 04/13.	2 x 1 = 2
Fire & electric shock risk from use of personal computer & laptop.	All electrical connections installed by properly accredited electricians with modern trip protection.	0 x 3 = 0

Slip & trip hazards	Minor slip & trip hazards exist but no more so than in a normal tidy domestic environment. There is no history of significant slips & trip events in this environment.	1 x 1 = 1

Scoring Criteria for Likelihood (chance of the hazard causing a problem)
0 – Zero to extremely unlikely, 1 – Very Unlikely, 2 – Unlikely, 3 – Likely, 4 – Very Likely, 5 – Almost certain to happen

Scoring Criteria for Severity of injury (or illness) resulting from the hazard
0 – No injury, 1 – First Aid is adequate, 2 – Minor injury, 3 – "Three day" injury, 4 – Major injury, 5 – Fatality or disabling injury

Source(s) of Information used to complete the above

1. Cardiff University Risk Management Procedure version 2 2016; available at https://intranet.cardiff.ac.uk/students/your-study/your-rights-and-responsibilities/procedures-and-guidance/risk-assessment/generic
2. Cardiff University Display Screen Equipment Guidelines; available at https://intranet.cardiff.ac.uk/students/your-study/your-rights-and-responsibilities/procedures-and-guidance/display-screen-equipment
3. HSE leaflet INDG36(rev4) "Working with Display Screen Equipment (DSE)", published 04/13; available at http://www.hse.gov.uk/pubns/indg36.pdf
4. Personal experience, previous industry training & common sense

Training. Please give details of any specific training requirements for persons carrying out this procedure/activity.

Status of Trainer	Training Requirement
	None required

Further Action

Highest Level of Risk Score	Action to be taken
0 to 5	No further action needed
6 to 11	Appropriate additional control measures should be implemented
12 to 25	Additional control measures **MUST** be implemented. Work **MUST NOT** commence until such measures are in place. If work has already started it must **STOP** until adequate control measures are in place.

Additional Control Measures Required

Hazard and how it can cause harm.	Additional Controls needed to Reduce Risk	How is Likelihood or Severity reduced?	Risk Now

After the implementation of new control measures the procedure/activity should be re-assessed to ensure that the level of risk has been reduced as required.

Action in the Event of an Accident or Emergency

First aid is available within 10 m in another room in the house. Two members of the household are first aid trained.

Arrangements for Monitoring the Effectiveness of Control

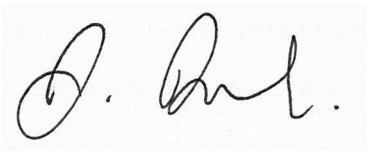
Any accidents or incidents to be reviewed at the end of the MSc project

Review this assessment must be reviewed by 4 September 2017

Name of Reviewer	Review Date	
Have the Control measures been effective in controlling the risk?		
Have there been any changes in the procedure or in information available which affect the estimated level of risk?		
What changes to the Control Measures are required?		

Signatures for printed copies:

Assessor: Date:



12 June 2017

Approved by: Date:

Reviewed by: Date:

This copy issued to: Date:
(print name and sign)

APPENDIX II: CODE LISTING: SGCHEM.C

New code is highlighted in red

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <mpi.h>
#include "../allvars.h"
#include "../proto.h"
#include "sgchem_proto.h"
#include "f2c.h"

/* Intensity weighting array defined outside functions for now */
double intensity_weighting[NPIX];

/* See README file for a description, authors list and usage policy*/

void init_chemistry(void) {
  /* Initialize parameters stored in coolr and cooli common blocks */
  INIT_CHEMISTRY_PARAMETERS(&All.DeutAbund,
#ifndef SGCHEM_VARIABLE_Z
                            &All.CarbAbund, &All.OxyAbund, &All.MAbund, &All.ZAtom,
#endif
                            &All.InitDustTemp,
#ifndef SGCHEM_VARIABLE_ISRF
                            &All.UVFieldStrength,
#endif
                            &All.LWBGType, &All.LWBGStartRedsh,
#ifndef SGCHEM_VARIABLE_Z
                            &All.DustToGasRatio,
#endif
#ifndef SGCHEM_VARIABLE_CRION
                            &All.CosmicRayIonRate,
#endif
                            &All.InitRedshift, &All.ExternalDustExtinction,
&All.H2FormEx,
                            &All.H2FormKin, &All.PhotoApprox, &All.ISRFOption,
&All.AtomicCoolOption);

  /* Initialize chemical rates */
  COOLINMO();
  CHEMINMO();

  /* ODE integrator tolerances */
  INIT_TOLERANCES();

  /* MCMA setup, if using NL99 network */
#ifdef MCMA
#if CHEMISTRYNETWORK == 15
  CMA_INIT();
#endif
#endif

  return;
}

void evolve_chemistry(void) {
  double dt;
  int idx, i;

   mpi_printf("SGCHEM: Looping over the chemistry for active cells \n");
  /* Loop over active particles, and do chemical evolution for each one */
  for(idx = 0; idx < TimeBinsHydro.NActiveParticles; idx++)
```

```
    {
      i = TimeBinsHydro.ActiveParticleList[idx];
      if(i < 0)
        continue;
      dt = (P[i].TimeBinHydro ? (((integertime) 1) << P[i].TimeBinHydro) : 0) *
All.Timebase_interval;

      evolve_chemistry_for_single_cell(dt, i);

    }

  return;
}

/* Chemical evolution, radiative heating & cooling */
void evolve_chemistry_for_single_cell(double dt, int index) {
  double time, rho, energy, divv, dl, yn;
  double rho_cgs, redshift, dust_temp;
  double non_eq_abundances[SGCHEM_NUM_SPECIES];
  double column_density_projection[NPIX];
  double column_density_projection_H2[NPIX];
  double column_density_projection_CO[NPIX];
  double a, a3inv, hubble_a, hubble_param, hubble_param2;
  double column_correction_cosmic, column_correction_cgs;
  double column_correction_NH, column_correction_NH2, column_correction_NCO;
  double Utherm_diff, Utherm_new;
  double dust_to_gas, carb_abund, oxy_abund, m_abund, Z_atom;
  double breakdown_of_thermal_rates[SGCHEM_NUM_THERMAL_RATES];
  int i, id;

#if NO_CHEMISTRY
  return;
#else

  /* Check size of timestep */
  if (dt < 0)
    {
      printf("Error: negative timestep in evolve_chemistry\n");
      endrun();
    }
  else if (dt == 0)
    {
      return;
    }

  /* Set current index and ID in cool.h */
  id = P[index].ID;
  SET_INDEX_ID_FOR_CHEMISTRY(&index, &id);


  /* Starting values */
  time = dt;
  rho = SphP[index].Density;
  divv = SphP[index].DivVel;
#ifdef STATIC_CHEMISTRY_TEST
  /* Allow direct comparison with one-zone model */
  divv = 0;
#endif

  /* Estimate of typical distance from centre to edge of local grid-cell, used for
computing
   * local contribution to shielding.
   *
   * XXX: should we make it possible to switch this on or off?
```

```
   */
  dl = 0.5 * pow(SphP[index].Volume, 1. / 3.);

  for (i = 0; i < SGCHEM_NUM_SPECIES; i++)
    {
      non_eq_abundances[i] = SphP[index].TracAbund[i];
    }

  /* Get local element abundances and dust-to-gas ratio, if they're spatially
variable */
#ifdef SGCHEM_VARIABLE_Z
  carb_abund = SphP[index].CarbAbund;
  oxy_abund  = SphP[index].OxyAbund;
  m_abund    = SphP[index].MAbund;
  Z_atom     = SphP[index].ZAtom;
  SET_LOCAL_ELEMENT_ABUNDANCES(&carb_abund, &oxy_abund, &m_abund, &Z_atom);

  dust_to_gas = SphP[index].DustToGasRatio;
  SET_LOCAL_DUST_ABUNDANCE(&dust_to_gas);
#else
  /* Setting this here avoids lots of messy conditional code later. We could also
set the other
     values (oxy_abund, m_abund, etc.) in the same way, but right now they're not
used later on */
  carb_abund = All.CarbAbund;
  oxy_abund  = All.OxyAbund;
  m_abund    = All.MAbund;
#endif

/* Run variable ISRF function & set intensity weightings. If variable ISRF flag not
defined set just set intensity weightings at 1/NPIX */
#ifdef SGCHEM_VARIABLE_ISRF
  find_and_set_local_ISRF(index);
#else
  for (i = 0; i < NPIX; i++) {
    intensity_weighting[i] = 1.0/48; //XXX should us NPIX instead of 48.0 but
returns array of 0.333!
// Debugging code
    if(id == 1000) {
      printf("Intensity weightings for particle id 1000: ");
      for (i = 0; i < NPIX; i++)
        printf("%.3g ;",intensity_weighting[i]);
// Debugging code ends
    }
  }

#endif



#ifdef SGCHEM_VARIABLE_CRION
  find_and_set_local_CRION(index);
#endif


#ifdef TREE_RAD
  for (i = 0; i < NPIX; i++)
    {
      column_density_projection[i] = SphP[index].Projection[i];
      /*    if (isnan(column_density_projection[i])){
         printf("SGChem: NaN coldensity i %d index %d id %d treecol
projection%g\n",i,index,id,SphP[index].Projection[i]);
        } */
#ifdef TREE_RAD_H2
```

```
      column_density_projection_H2[i] = SphP[index].ProjectionH2[i];
      if(isnan(column_density_projection[i]))
        {
          printf("SGChem: NaN H2 coldensity i %d index %d id %d treecol
projection%g\n", i, index, id, SphP[index].Projection[i]);
        }
#else
      column_density_projection_H2[i] = 0.0;
#endif
#ifdef TREE_RAD_CO
      column_density_projection_CO[i] = SphP[index].ProjectionCO[i];
#else
      column_density_projection_CO[i] = 0.0;
#endif

    }
#else
/* If TREE_RAD and friends are not in use, then these are all single-element arrays
*/
  column_density_projection[0] = column_density_projection_H2[0] =
column_density_projection_CO[0] = 0.0;

#endif /* TREE_RAD */




  /* Convert from comoving to physical units */
  if (All.ComovingIntegrationOn)
    {
      a = All.Time;
      a3inv = 1 / (a * a * a);
      hubble_a = hubble_function(a);
      hubble_param = All.HubbleParam;
      hubble_param2 = hubble_param * hubble_param;
      redshift = (1 / a) - 1;
    }
  else
    {
      a = a3inv = hubble_a = hubble_param = hubble_param2 = 1;
      redshift = All.InitRedshift;
    }

  time *= 1 / hubble_a / hubble_param;
  rho *= a3inv * hubble_param2;
  energy = SphP[index].Utherm * rho;    /* Energy here is energy density, AREPO
evolves specific energy */
  divv *= 1.0 / a;
  if (All.ComovingIntegrationOn)
    divv += 3 * hubble_a;
  dl *= a;

  column_correction_cosmic = hubble_param / (a * a);
  for (i = 0; i < NPIX; i++)
    {
      column_density_projection[i] *= column_correction_cosmic;
      column_density_projection_H2[i] *= column_correction_cosmic;
      column_density_projection_CO[i] *= column_correction_cosmic;
    }

  /* Convert from AREPO code units to cgs */
  time *= All.UnitTime_in_s;
  rho_cgs = rho * All.UnitDensity_in_cgs;       /* We keep rho in code units as we
need it later */
```

```
  energy *= All.UnitEnergy_in_cgs / pow(All.UnitLength_in_cm, 3);
  divv *= 1.0 / All.UnitTime_in_s;
  dl *= All.UnitLength_in_cm;

  column_correction_cgs = All.UnitDensity_in_cgs * All.UnitLength_in_cm;
  for (i = 0; i < NPIX; i++)
    {
      column_density_projection[i] *= column_correction_cgs;
      column_density_projection_H2[i] *= column_correction_cgs;
      column_density_projection_CO[i] *= column_correction_cgs;
    }

/* Convert from mass surface density to particle column
   * density
   */
  column_correction_NH = 1.0 / ((1.0 + 4.0 * ABHE) * PROTONMASS);
  column_correction_NH2 = 1.0 / (2.0 * PROTONMASS);
  column_correction_NCO = 1.0 / (28.0 * PROTONMASS);
  for (i = 0; i < NPIX; i++)
    {
      column_density_projection[i] *= column_correction_NH;
      column_density_projection_H2[i] *= column_correction_NH2;
      column_density_projection_CO[i] *= column_correction_NCO;
    }


  /* Compute derived units */
  yn = rho_cgs / ((1.0 + 4.0 * ABHE) * PROTONMASS);

  /* Check that the chemistry has sane values after the advection. */
  for (i = 0; i < SGCHEM_NUM_SPECIES; i++)
    {
      if (non_eq_abundances[i] < -1e-12)
        {
          printf("sgchem.c negative abundance from advection, species = %d
abundance = %g\n", i, non_eq_abundances[i]);
          printf("sgchem.c: Setting abundance to +1e-20 (this might not help!)\n");
          non_eq_abundances[i] = 1e-20;
        }
      if (non_eq_abundances[IH2] > 0.5)
        non_eq_abundances[IH2] = 0.5;
      if (non_eq_abundances[IHP] > 1)
        non_eq_abundances[IHP] = 1;
#if CHEMISTRYNETWORK != 1
      if (non_eq_abundances[ICO] > carb_abund)
        non_eq_abundances[ICO] = carb_abund;
#endif
    }

  EVOLVE_ABUNDANCES(&time, &dl, &yn, &divv, &energy, &redshift, non_eq_abundances,
breakdown_of_thermal_rates, column_density_projection,
column_density_projection_H2, column_density_projection_CO, intensity_weighting,
&dust_temp, &id);


  /* Convert from cgs to AREPO code units */
  energy = energy * pow(All.UnitLength_in_cm, 3) / All.UnitEnergy_in_cgs;       /*
Energy density in Arepo code units */

  /* Set new dust temperature */
  SphP[index].DustTemp = dust_temp;

  /* Update evolved values */
  Utherm_new = energy / rho;
```

```
  Utherm_diff = Utherm_new - SphP[index].Utherm;
  SphP[index].Utherm += Utherm_diff;
  SphP[index].Energy += a * a * Utherm_diff * P[index].Mass;

#ifdef TREE_RAD_VEL
  /* Compute square of thermal velocity of H atom. Note: used for velocity-
dependent molecular shielding.
   * Note: value here is in code units */
  SphP[index].Vth2 = 2.0 * SphP[index].Utherm * GAMMA_MINUS1 * (1.0 + 4.0 * ABHE) /
(1.0 + ABHE - SphP[index].TracAbund[IH2] + SphP[index].TracAbund[IHP]);
#endif

  /* Update cooling time */
#ifdef SGCHEM_OUTPUT_COOLTIME
  SphP[index].CoolTime = SphP[index].Utherm * dt / Utherm_diff;
#endif

#ifdef USE_ENTROPY_FOR_COLD_FLOWS
  SphP[index].A = GAMMA_MINUS1 * SphP[index].Utherm / pow(SphP[index].Density *
a3inv, GAMMA_MINUS1);
  SphP[index].Entropy = log(SphP[index].A) * P[index].Mass;
#endif



  for(i = 0; i < SGCHEM_NUM_SPECIES; i++)
    {
      SphP[index].TracAbund[i] = non_eq_abundances[i];
      SphP[index].MassTracAbund[i] = P[index].Mass * SphP[index].TracAbund[i];
    }
#if CHEMISTRYNETWORK == 4 || CHEMISTRYNETWORK == 5
  SphP[index].TracAbund[IHATOM] = 1.0 - 2.0 * SphP[index].TracAbund[IH2] -
SphP[index].TracAbund[IHP];
  if (SphP[index].TracAbund[IHATOM] < 0.0)
    {
      SphP[index].TracAbund[IHATOM] = 0.0;
    }
  SphP[index].MassTracAbund[IHATOM] = SphP[index].TracAbund[IHATOM] *
P[index].Mass;
#endif
#if CHEMISTRYNETWORK == 5
  SphP[index].TracAbund[ICP] = carb_abund - SphP[index].TracAbund[ICO];
  if (SphP[index].TracAbund[ICP] < 0.0)
    {
      SphP[index].TracAbund[ICP] = 0.0;
    }
  SphP[index].MassTracAbund[ICP] = SphP[index].TracAbund[ICP] * P[index].Mass;

  if(index == 10000)
    printf("sgchem:Final single cell %d abundances %g %g %g\n", index,
SphP[index].TracAbund[0], SphP[index].TracAbund[1], SphP[index].TracAbund[2]);
#endif
#if CHEMISTRYNETWORK == 1
  SphP[index].TracAbund[IHATOM]     = 1.0 - 2.0 * SphP[index].TracAbund[IH2] -
SphP[index].TracAbund[IHP] - SphP[index].TracAbund[IHD];
  if (SphP[index].TracAbund[IHATOM] < 0.0) {
    SphP[index].TracAbund[IHATOM] = 0.0;
  }
  SphP[index].MassTracAbund[IHATOM] = SphP[index].TracAbund[IHATOM] *
P[index].Mass;

  SphP[index].TracAbund[IHEATOM]     = ABHE - SphP[index].TracAbund[IHEP] -
SphP[index].TracAbund[IHEPP];
  if (SphP[index].TracAbund[IHEATOM] < 0.0) {
```

```
    SphP[index].TracAbund[IHEATOM] = 0.0;
  }
  SphP[index].MassTracAbund[IHEATOM] = SphP[index].TracAbund[IHEATOM] *
P[index].Mass;

  SphP[index].TracAbund[IDATOM]     = All.DeutAbund - SphP[index].TracAbund[IDP] -
SphP[index].TracAbund[IHD];
  if (SphP[index].TracAbund[IDATOM] < 0.0) {
    SphP[index].TracAbund[IDATOM] = 0.0;
  }
  SphP[index].MassTracAbund[IDATOM] = SphP[index].TracAbund[IDATOM] *
P[index].Mass;
#endif
#if CHEMISTRYNETWORK == 15
  SphP[index].TracAbund[IHATOM] = 1.0 - 2.0 * SphP[index].TracAbund[IH2] -
SphP[index].TracAbund[IHP] - SphP[index].TracAbund[ICHX]
    - SphP[index].TracAbund[IOHX] - SphP[index].TracAbund[IHCOP];

  SphP[index].TracAbund[ICATOM] = carb_abund - SphP[index].TracAbund[ICP] -
SphP[index].TracAbund[ICHX]
    - SphP[index].TracAbund[ICO] - SphP[index].TracAbund[IHCOP];

  SphP[index].TracAbund[IOATOM] = oxy_abund - SphP[index].TracAbund[IOHX] -
SphP[index].TracAbund[ICO]
    - SphP[index].TracAbund[IHCOP];

  SphP[index].TracAbund[IHEATOM] = ABHE - SphP[index].TracAbund[IHEP];

  SphP[index].TracAbund[IMATOM]  = m_abund - SphP[index].TracAbund[IMP];

  for (i = 0; i < SGCHEM_NUM_ADVECTED_SPECIES; i++) {
    if (SphP[index].TracAbund[i] < 0) {
      SphP[index].TracAbund[i] = 0.0;
    }
    SphP[index].MassTracAbund[i] = SphP[index].TracAbund[i] * P[index].Mass;
  }
#endif

#ifdef SGCHEM_DUMP_THERMAL_RATES
  /* Pass out the heating / cooling rate block to the main code for writing to
snapshots */
  for(i = 0; i < SGCHEM_NUM_THERMAL_RATES; i++)
    SphP[index].HeatCoolRates[i] = breakdown_of_thermal_rates[i];
#endif

  /* Update pressure */
  set_pressure_of_cell(index);

  return;
#endif /* No CHEMISTRY */
}

#ifdef SGCHEM_VARIABLE_ISRF
void find_and_set_local_ISRF(int index)
{
  double local_ISRF;

#ifndef SGCHEM_VARIABLE_ISRF_STELLAR_SOURCES
/* Compute local strength of ISRF, using a simple Galactic model from Wolfire et
al 2003 */
  double xi = P[index].Pos[0] - boxHalf_X;
  double yi = P[index].Pos[1] - boxHalf_Y;
  double zi = P[index].Pos[2] - boxHalf_Z;
```

```
  double R = sqrt(xi*xi + yi*yi) * All.UnitLength_in_cm / KILOPARSEC;

  /* Radial dependence of the ISRF is taken from Wolfire+ 2003 */
  double R1 = 4.;
  double HRJ = 4.1;
  double R0 = 8.;

  if (R < R1)
    local_ISRF = All.UVFieldStrength * exp(-(R1-R0) / HRJ);
  else
    local_ISRF = All.UVFieldStrength * exp(-(R-R0) / HRJ);

#else
/* Compute local strength of ISRF incl. accreting sinks */

  #define Rsun_in_cm 6.96e10
  #define secs_per_yr 3.16e7
  #define Stef_Boltz 5.67e-5  //cgs units
  #define light_speed 3e10 //cm/s
  #define Habing_field 5.29e-14 //ergs/cm^3

  double acc_rate[NSinksAllTasks], sinkR[NSinksAllTasks], sinkL[NSinksAllTasks],
sinkT[NSinksAllTasks], UVfrac[NSinksAllTasks], sinkUVL[NSinksAllTasks],
sinkD[NSinksAllTasks], sinkUVflux[NSinksAllTasks], incr_G0[NSinksAllTasks];

  int i, isink;
  long sinkpix;
  double sink_dist, sink_theta, sink_phi, sink_lat, sink_long;
  double pixel_G0[NPIX];
  for (i = 0; i < NPIX; i++)
    pixel_G0[i] = All.UVFieldStrength/48;

  /* Loop through all sinks calculating incremental G0 and pixel location */
  for(isink = 0; isink < NSinksAllTasks; isink++) {
    /* Calculate accretion rate (Maschberger wt al 2014, Offner et al 2009) */
    /* SinkP[isink].Mass = 5; */ //Forces sink mass to 5 for testing
    if (SinkP[isink].Mass < 0.05)
      acc_rate[isink] = 0;
    else if (SinkP[isink].Mass < 0.5)
      acc_rate[isink] = 5e-6;
    else
      acc_rate[isink] = 7.9e-6*pow(SinkP[isink].Mass,0.6667);

    /* Calculate adiabatic accretion radius (Hosokawa and Omukai 2008) */
    sinkR[isink] = 26*pow(SinkP[isink].Mass,0.27)*pow((acc_rate[isink]/1e-
3),0.41)*Rsun_in_cm;

    /* Calculate bolometric luminosity from accretion */
    if (!sinkR[isink])
      sinkL[isink] = 0;
    else
      sinkL[isink] =
GRAVITY*SinkP[isink].Mass*All.UnitMass_in_g*acc_rate[isink]*All.UnitMass_in_g/secs_
per_yr/sinkR[isink];

    /* Calculate effective photopsheric temperature */
    if (!sinkR[isink])
      sinkT[isink] = 0;
    else
      sinkT[isink] =
pow(sinkL[isink]/(4*M_PI*sinkR[isink]*sinkR[isink]*Stef_Boltz),0.25);

    /* Calculate UV fraction and luminosity (polynomial fit to Planck function in
range 6 - 13.6 eV) */
```

```
    if (sinkT[isink] < 6e3)
      UVfrac[isink] = 0;
    else
      UVfrac[isink] = 0.1-(3.3e-5*sinkT[isink])+(2.4e-
9*sinkT[isink]*sinkT[isink])+(6.8e-14*sinkT[isink]*sinkT[isink]*sinkT[isink]);
    sinkUVL[isink] = UVfrac[isink]*sinkL[isink];

    /* Calculate distance to sinks */
    double dx = SinkP[isink].Pos[0] - P[index].Pos[0];
    double dy = SinkP[isink].Pos[1] - P[index].Pos[1];
    double dz = SinkP[isink].Pos[2] - P[index].Pos[2];
    sinkD[isink] = sqrt(dx*dx + dy*dy + dz*dz) * All.UnitLength_in_cm;

    /* Calculate UVflux & incremental energy density/G0 at particle from individual
sinks */
    sinkUVflux[isink] = sinkUVL[isink]/(4*M_PI*sinkD[isink]*sinkD[isink]);
    incr_G0[isink] = 4*sinkUVflux[isink]/light_speed/Habing_field;

    /* Assign incremental energy density from sink to a pixel */
    double sink_pos_vector[3] = {dx, dy, dz};
    vec2pix_ring(NSIDE, sink_pos_vector, &sinkpix);
    pixel_G0[sinkpix] = pixel_G0[sinkpix]+incr_G0[isink];

    /* Calculate the angle from a typical particle to the sink */
    if(P[index].ID == 1000) {
      sink_dist = sqrt(dx*dx + dy*dy + dz*dz);
      pix2ang_ring(NSIDE, sinkpix, &sink_theta, &sink_phi);
      sink_lat = sink_theta*57.29578;
      sink_long = sink_phi*57.29578;
      printf("sgchem: Pixel containing sink %d around particle id 1000: %ld \n",
isink, sinkpix);
      printf("sgchem: D, theta; phi to sink: %.1f %.2f; %.2f; Lat/long of sink:
%.1f; %.1f \n", sink_dist, sink_theta, sink_phi, sink_lat, sink_long);
    }

  }

  /* Calculate total G0 at particle position */
  double incr_G0_total = 0;
  for (isink = 0; isink < NSinksAllTasks; isink++)
    incr_G0_total += incr_G0[isink];
  local_ISRF = All.UVFieldStrength + incr_G0_total;


/* Calculate intensity weighting by pixel if anisotropic flux option defined*/
#ifdef TREE_RAD_ANISOTROPIC_FLUX
  for (i = 0; i < NPIX; i++)
    intensity_weighting[i] = pixel_G0[i]/local_ISRF;
/* ... or set intensity weighting as 1/NPIX for isotropic flux */
#else
  for (i = 0; i < NPIX; i++)
      intensity_weighting[i] = 1.0/48; //XXX should be 1/NPIX but returns array of
0.333!
#endif

/* Debugging Code */
  if(P[index].ID == 1000) {
    double iwtot = 0.0;
    printf("sgchem: Intensity weightings per pixel for particle id 1000: \n");
    for (i = 0; i < NPIX; i++) {
      printf("%.3g; ",intensity_weighting[i]);
      iwtot += intensity_weighting[i];
    }
    printf("\nsgchem: Total intensity weighting = %.3f \n", iwtot);
```

```
/* Debugging code ends */
  }
#endif //SGCHEM_VARIABLE_ISRF_STELLAR_SOURCES

  SET_LOCAL_ISRF(&local_ISRF);
  return;
}
#endif //SGCHEM_VARIABLE_ISRF

#ifdef SGCHEM_VARIABLE_CRION
/* Compute local cosmic ray ionization rate. If we're using the COSMIC_RAYS option,
we scale this according to the
 * cosmic ray energy density; otherwise, we use a simple Galactic scaling model
from Wolfire et al 2003
 */
void find_and_set_local_CRION(int index)
{
  double local_CRIR, cr_energy_density;

#ifdef COSMIC_RAYS
  cr_energy_density = SphP[index].CR_SpecificEnergy*SphP[index].Density;
  cr_energy_density *=
All.cf_a2inv*All.cf_a2inv*All.HubbleParam*All.HubbleParam*All.UnitEnergy_in_cgs/pow
(All.UnitLength_in_cm, 3);
  local_CRIR = 0;
#else
  double xi = P[index].Pos[0] - boxHalf_X;
  double yi = P[index].Pos[1] - boxHalf_Y;
  double zi = P[index].Pos[2] - boxHalf_Z;

  double R = sqrt(xi*xi + yi*yi) * All.UnitLength_in_cm / KILOPARSEC;

  /*radial dependence of the CRIR is taken from Wolfire+ 2003*/
  double R0 = 8.;
  double HROB = 3.5;

  local_CRIR = All.CosmicRayIonRate * exp(-(R-R0) / HROB);
  cr_energy_density = 0;
#endif
  SET_LOCAL_CR_ION_RATE(&cr_energy_density, &local_CRIR);
}
#endif
```

APPENDIX III: CODE LISTING: DUST_TEMP.PY

```
#!/usr/bin/python
import numpy as np
from scipy.integrate import simps
from scipy.interpolate import interp1d
import matplotlib.pyplot as plt

# Define mass and distance parameters
m_star = 5 #code units (solar masses)
D_min = 0.2 #code units (31 code units ~ 1pc)
D_max = 250 # code units
D_delta = 0.1 #code units

D_plot = 5 # distance to plot (code units)

ISRF_G0 = 0 #set to zero to calculate from ISRF data

#Set required constants
hP = 6.63e-27
kB = 1.38e-16
cL = 3e10
SB = 5.67e-5
Habing_Field = 5.29e-14

#Load ISRF data and convert to easily usable arrays
ISRF = np.loadtxt('Mathis_ISRF.txt',skiprows=2)
ISRF_wave = ISRF[:,0]
ISRF_J4pi = ISRF[:,1]
ISRF_J = ISRF_J4pi/4/np.pi

#Cubic spline fit of ISRF data
ISRF_fit = interp1d(ISRF_wave, ISRF_J, kind='cubic')

#Calculate G0 if not defined above using Simpson's rule in range 6-13.6 eV
if ISRF_G0 == 0:
    G0_waves = np.linspace(9.141e-2,0.2072,100)
    ISRF_u = ISRF_fit(G0_waves)*4*np.pi/cL
    ISRF_G0 = simps(ISRF_u,G0_waves)/5.29e-14

#Plot ISRF spline fit
fig = plt.figure()
ax=fig.add_subplot(111)
ax.plot(ISRF_wave,ISRF_wave * ISRF_J4pi,ls='', marker ='.')
ax.plot(ISRF_wave, ISRF_wave * ISRF_fit(ISRF_wave)*4*np.pi)
ax.set_xscale('log')
ax.set_yscale('log')
ax.set_xlabel('wavelength / microns')
ax.set_ylabel('wavelength * 4piJ / ergs/s/cm^2')
plt.show()

#Load opacity data and convert to easily usable arrays
DATA = np.loadtxt('dustkappa_OH94plusMMP83.inp', skiprows=9, usecols=(0,1))
Opac = DATA[:,1] #cm^2 / g
Opac_wave = DATA[:,0] #microns

#Cubic spline fit of opacity data
Opac_wavefit = interp1d(Opac_wave, Opac, kind='cubic')

#Plot opacity spline fit
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(Opac_wave,Opac,ls=' ',marker='.')
ax.plot(Opac_wave,Opac_wavefit(Opac_wave))
```

```
ax.set_xscale('log')
ax.set_xlabel('wavelength / microns')
ax.set_ylabel('Dust opacity / cm^2/g')
#ax.set_title('10 Msun sink - Feedback code ON')
plt.show()

#Set log-scale wavelength range & calculate warming = integrated opacity *
intensity from ISRF using Simpson's rule
Wave = np.logspace(-1.04,3,10000)
ISRF_kapJ = Opac_wavefit(Wave) * ISRF_fit(Wave)
ISRF_kapJ_sum = simps(ISRF_kapJ,Wave)

#Convert opacity wavelength parameters to frequency
Opac_freq = 3e14/Opac_wave #Hz
Opac_freqfit = interp1d(Opac_freq, Opac, kind='cubic')

#Set log-scale frequency range and linear temp. range
Freq = np.logspace(11.5,15.5,10000)
T = np.linspace(5,100,10000)

#Calculate dust radiated energy (cooling) at range of temperatures
Dust_kapB_sum = np.zeros_like(T)
for i in range(len(T)):
    Dust_kapB = Opac_freqfit(Freq) * 2*hP*Freq**3/cL**2/(np.exp(hP*Freq/(kB*T[i]))-
1)
    Dust_kapB_sum[i] = simps(Dust_kapB,Freq)

#Find temperature at which heating = cooling for ambient ISRF
for i in range (len(T)-1):
    if Dust_kapB_sum[i] < ISRF_kapJ_sum and Dust_kapB_sum[i+1] > ISRF_kapJ_sum:
        T_dust_ISRF = T[i]

#Algorithm for determining incremental G0 at range of distances
acc_rate = 7.9e-6*m_star**(2.0/3.0) #solar masses pa
r_star = 26*m_star**0.27*(acc_rate/1e-3)**0.41 #solar radii
lum_star = 6.67e-8*m_star*1.99e33*acc_rate*1.99e33/3.16e7/(r_star*6.96e10) #ergs /s
Teff_star = (lum_star/4/np.pi/(r_star*6.96e10)**2/SB)**0.25 #K

D_star = np.linspace(D_min,D_max,int((D_max - D_min)/D_delta +1))

UVfrac_star = 6.8e-14*Teff_star**3+2.4e-9*Teff_star**2-3.3e-5*Teff_star+0.1
UVflux_star = lum_star * UVfrac_star/4/np.pi/(D_star*1e17)**2

incr_G0 = 4*UVflux_star/cL/Habing_Field
Total_G0 = ISRF_G0 + incr_G0
G0_factor = Total_G0/ISRF_G0

# Scale up ambient ISRF warming by G0 factor
ISRF_kapJ_sum_G0 = ISRF_kapJ_sum * G0_factor

#Find temperature at which warming = cooling for scaled-up G0 at range of distances
T_dust_G0 = np.zeros_like(D_star)
for h in range(len(D_star)):
    for i in range (len(T)-1):
        if Dust_kapB_sum[i] < ISRF_kapJ_sum_G0[h] and Dust_kapB_sum[i+1] >
ISRF_kapJ_sum_G0[h]:
            T_dust_G0[h] = T[i]

#Find incremental warming exactly from superposition of black-body curve from star
Star_kapJ_sum = np.zeros_like(D_star)
Total_kapJ_sum = np.zeros_like(D_star)
T_dust_Total = np.zeros_like(D_star)
for i in range(len(D_star)):
```

```
    Star_J = 2*hP*Freq**3/cL**2/(np.exp(hP*Freq/(kB*Teff_star))-1) *
(r_star*6.96e10)**2 / (D_star[i]*1e17)**2
    Star_kapJ = Opac_freqfit(Freq) * Star_J
    Star_kapJ_sum[i] = simps(Star_kapJ, Freq)
    Total_kapJ_sum[i] = ISRF_kapJ_sum + Star_kapJ_sum[i]

#Find temperature at which warming = cooling for exact solution
    for k in range (len(T)-1):
        if Dust_kapB_sum[k] < Total_kapJ_sum[i] and Dust_kapB_sum[k+1] >
Total_kapJ_sum[i]:
            T_dust_Total[i] = T[k]

#Calculate total ISRF exeactly for plotting purposes at selected D
Star_J = 2*hP*cL**2/(Wave*1e-4)**5/(np.exp(hP*cL/(kB*Wave*1e-4*Teff_star))-1) *
(r_star*6.96e10)**2 / (D_plot*1e17)**2 *1e-4
Total_J = Star_J + ISRF_fit(Wave)

for i in range(len(D_star)):
    if D_plot >= D_star[i] and D_plot < D_star[i+1]:
        index = i
        break

#Plot ISRF intensity for various options
fig = plt.figure()
ax=fig.add_subplot(111)
ax.plot(Wave,Wave * 4*np.pi*ISRF_fit(Wave))
ax.plot(Wave,Wave * 4*np.pi*ISRF_fit(Wave) * G0_factor[index],c = 'red')
ax.plot(Wave,Wave * 4*np.pi*Total_J,c='green')
ax.set_xscale('log')
ax.set_yscale('log')
ax.set_xlabel('wavelength / microns')
ax.set_ylabel('wavelength * 4piJ / ergs/s/cm^2')
ax.set_title('%d Msun sink at D = %d' %(m_star, D_plot))
plt.show()

#Plot intensity * opacity for various options
fig = plt.figure()
ax=fig.add_subplot(111)
ax.plot(Wave,Wave * 4*np.pi*ISRF_fit(Wave) * Opac_wavefit(Wave))
ax.plot(Wave,Wave * 4*np.pi*ISRF_fit(Wave) * G0_factor[index]* Opac_wavefit(Wave),
c='red')
ax.plot(Wave,Wave * 4*np.pi*Total_J* Opac_wavefit(Wave), c='green')
ax.set_xscale('log')
ax.set_yscale('log')
ax.set_xlabel('wavelength / microns')
ax.set_ylabel('Dust opacity * wavelength * 4piJ / ergs/s/g')
ax.set_title('%d Msun sink at D = %d' %(m_star, D_plot))
plt.show()

#Print key G0 data
print 'G0 = %.1f' %Total_G0[index]
print 'G0 factor = %.1f (ISRF = %.2f)\n' %(G0_factor[index],ISRF_G0)

print 'T_dust (ISRF only) = %.1f\n' %T_dust_ISRF
```